12-29-00

A/Prov

Jc952 U.S. PTO 12/27/00

Jc714 U.S. PTO 60/258465 12/27/00

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

REQUEST FOR FILING A PROVISIONAL APPLICATION FOR PATENT

UNDER 37 CFR §1.53 (c)

INVENTOR(S)

1.	Isaac	Levanon	3 Nachal Besor St., Ramat Hasharn, Israel
2.	Yoni	Lavi	21 Bar Ilan St., Raanana, Israel

TITLE OF THE INVENTION

OPTIMIZATION T-JUNCTION CRACKING-PROBLEM OF IMAGE PARCELS BEING PACKET STREAMED BY UTILIZING QUADTREE SCHEME

X Direct all correspondence to Customer Number 23488.

Gerald B. Rosenberg, Esq. (Reg No.: 30,320)
NewTechLaw
285 Hamilton Avenue, Suite 520
Palo Alto, California 94301

Telephone: 650.325.2100
Facsimile: 650.325.2107

23488
PATENT TRADEMARK OFFICE

ENCLOSED APPLICATION PARTS *(check all that apply)*

X	Specification	No. of pages: 11		____	Small Entity Statement
X	Drawings	No. of sheets: 5		____	Power of Attorney
____	Declaration			____	Assignment and Cover Sheet
X	Other: Return-Receipt Post Card				

METHOD OF PAYMENT OF FILING FEES FOR THIS PROVISIONAL APPLICATION FOR PATENT

Provisional Basic Filing Fee: $ 150.00 (Small Entity: $75.00) Filing Fee Amount: $ 150.00

X A check is enclosed to cover the Filing Fees.

X The Commissioner is hereby authorized charge Filing Fees or credit any overpayment to: Deposit Account Number: 50-0890.

X This invention was not made by or under contract with a US Government agency.

____ US Government agency and Contract: ____

Signature: [signature] Date: December 26, 2000

Gerald B. Rosenberg
Reg. No.: 30,320

Application Docket No: FLVT3002

Express Mail Label No.: EL 661 534 274 US

Address To: Box Provisional Application, Assistant Commissioner for Patents, Washington, DC 20231

gbr/flvt/3002.002.prov.xmittal wpd

EFFICIENT CORRECTION OF T-JUNCTION CRACKING-PROBLEM OF IMAGE PARCELS BEING PACKET STREAMED BY UTILIZING QUADTREE SCHEME

Inventors:
Isaac Levanon
Yoni Lavi

Background of the Invention

The present invention is generally related to the delivery of high-resolution highly featured graphic images over limited and narrowband communications channels.

Summary of the Invention

The objective is to display a two-dimensional pixel map, a16-Bit RGB color image in the preferred embodiments, of very large dimensions and permitting the viewing of the image from a dynamic three-dimensional viewpoint. Multiple such images are remotely hosted for on-demand selection and transfer to a client system for viewing.

Images, as stored by the server, may individually range from gigabytes to multiple terabyte in total size. A correspondingly large server storage and processing system is contemplated. Conversely, client systems are contemplated to be conventional personal computer systems and, in particular, mobile, cellular, embedded, and handheld computer systems, such as personal digital assistants (PDAs) and internet-capable digital phones, with relatively limited to highly constrained network communications capabilities. For most wireless applications, conventional narrowband communications links have a bandwidth of less than approximately three kilobytes of data per second. Consequently, transmittal of entire images to a client system in reasonable time is infeasible as a practical matter.

Description of the Invention

Overview:

For purposes of the present invention, each image (Figure 1) is at least logically defined in terms of multiple grids of image parcels with various levels of resolutions (Figure 2) that are created through composition of information from all level of resolutions, and stored by the server to provide an image for transfer to a client system (Figure 3). Composed and separate static and dynamically created layers are transferred to client system in parcels in a program selectable order to optimize for fast quality build-up of the image presented to a user of the client system, particularly when the parcels are streamed over a narrowband communication link.

The multiple layers of an image allow the selectivity to incorporate topographical, geographical, orientational, and other terrain and mapping related information into the image delivered. Other layers, such as geographic grids, graphical text overlays, and hyperlink selection areas, separately provided or composed, aid in the useful presentation and navigation of the image as presented by the client system and viewed by the user.

Compositing of layers on the server enables the data transfer burden to be reduced, particularly in analysis of the requirements and capabilities of the client system and the connecting communications link. Separate transfer of layers to the client system allows the client system selectivity in managing and presentation of the data to the user.

The system and methods of the present invention are designed to, on demand, select, process and immediately transfer data parcels to the client system, which immediately processes and displays a low-detail representation of the image requested by the client system. The system and methods immediately continue to select, process and sequentially transfer data parcels that, in turn, are processed and displayed by the client system to augment the presented image and thereby provide a continuously improving image to the user.

Selection of the sequentially transferred data is, in part, dependent on the progressive translation of the three-dimensional viewpoint as dynamically modified on the client system during the transfer process. This achieves the above-stated objective while concurrently achieving a good rendering quality for continuous fly-over of the image as fast as possible, yet continuously building the image quality to the highest resolution of the image as stored by the server.

To optimize image quality build-up over limited and narrowband communication links, the target image, as requested by the client system, is represented by multiple grids of 64x64 image pixels (Figure 4) with each grid having some corresponding level of detail. That is, each grid is treated as a sparse data array that can be progressively revised to increase the resolution of the grid and thereby the level of detail presented by the grid. The reason for choosing the 64x64 pixel dimension is that, using current image compression algorithms, a 16-bit 64x64 pixel array image can be presented as a 2KByte data parcel. In turn, this 2KByte parcel is the optimal size, subject to conventional protocol and overhead requirements, to be transmitted through a 3KByte per second narrowband transmission channel. Using a smaller image array, such as 32x32, would create a 0.5KByte parcel, hence causing inefficiencies due to packet transmission overhead, given the nature of current wireless communications protocols.

Image array dimensions are preferably powers of two so that they can be used in texture mapping efficiently. Each parcel, as received by the client system, is preferably immediately processed and incorporated into the presented image. To do so efficiently, according to the present invention, each data parcel is independently processable by the client system, which is enabled by the selection and server-side processing used to prepare a parcel for transmission. In addition, each data parcel is sized appropriate to fit within the level-1 cache, or equivalent, of the client system processor, thereby enable the data processing intensive operations needed to process the data parcel to be performed without extended memory access delays. In the preferred embodiment of the present invention, data parcels are also processed for texture mapping and other image features, such as topographical detailing.

Currently, with regard to conventional client systems, a larger image array, such as 128x128, is too large to be fully placed within the level-1 cache of many of the smaller conventional current processors, such as used by personal digital assistants (PDAs) and cellular phones. Since access to cache memory is substantially faster than to RAM this will likely result in lower frame rate.

Different and larger data parcel sizes may be optimal as transmission protocols and micro-architectures of the client computers change. For purposes above, the data content was a pixel array representing image data. Where the data parcel content is vector, text or other data that may subject to different client system design factors, other parcel sizes may be used.

In the process implemented by the present invention, data parcels maybe selected for sequential transmission based on a prioritization of the importance of the data contained. The criteria of importance maybe defined as suitable for particular applications and may directly relate to the presentation of image quality, provision of a textual overlay of a low-quality image to quickly provide a navigational orientation, or the addition of topography information at a rate or timing different from the rate of image quality improvement. Thus, image data layers reflecting navigational cues, text overlays, and topography can be composed into data packets for transmission subject to prioritizations set by the server alone, based on the nature and type of the client system, and interactively influenced by the actions and commands provided by the user of the client system (Figure 5).

Progressive transmission of image parcels is performed in an iterative process involving selection of an image data grid within the target image of the client system, which is a portion of a potentially multi-layered source image stored by the server. The selection parameters are preferably dependent on the client navigation viewpoint, effective velocity, and height, and the effective level of detail currently presented in each grid. Once a grid is selected, the server selects the source data to be logically composed into the selected grid to complement the effective resolution of that grid, processing the grid data to produce the optimally sized size grid data parcels, and sequentially transmitting the parcels to the client system. Preferably, the detail of a grid array is sequentially enhanced by division of the grid into sub-grids related by a power of two (Figure 6). Thus, a given grid is preferably updated using four data parcels having twice the data resolution of the existing grid. Whatever number of parcels are used, each data parcel is rendered by the client system into the target image. Additional client system image data processing to provide texturing and three-dimensional representation of the data may be performed as part of the parcel rendering and integration into the target image.

Image Parcel Download Sequence:

The server of the present invention supports the download of parcel data to a client system by providing data parcels in response to network requests originated by client systems. Each requested data parcel is identified within a grid coordinate system relative to an image stored by the server.

A client system implementing the process of the present invention is responsible for identifying and requesting parcel data, then rendering the parcel data into the target image at the correct location. The client system is also responsible for managing navigational and other interaction with the user. In identifying the parcel data to be requested, the client system operates to select grids within the coordinate system, corresponding to portions of the target image, for which to request a corresponding data parcel. The requests are issued over the network to the server and rendering performed asynchronously as data parcels are received. The order of data parcel requests is defined as a sequence that will provide for the optimal build-up of the target image as presented to the user. The rate of optimal build up of the target image is dependent on the nature of the target image requested, such as the supported parcel size and depth of the target image that can be rendered by the client system.

The client identifies and requests the download of data parcels in the process as follows. Denote the target image as I_0 and its size in pixels as (X, Y). Let N be the smallest power of 2 that is equal or greater than max {X,Y}. Construct the grid of 64x64 pixel grid-images $I_{0,i,j}$ that together compose the target image I_0. The rectangle [64i,64i + 64] x [64j,64 j + 64] of I_0 is mapped to $I_{0,i,j}$.

In order to view a large portion of the image, the target image, without downloading the substantial bulk of the target image, mip-maps of I_O are created, representing a collection of images to be used as surface textures when rendering a two-dimensional representation of a three-dimensional scene, and which are defined recursively as:

$$I_{k+1}(i,j) = avg(I_k(2i,2j), I_k(2i+1,2j), I_k(2i,2j+1), I_k(2i+1,2j+1))$$

Such mip-maps are created up to $I_M, M = \log_2(N) - 6$. At this point, I_M is a 64x64 image containing the entire area of the original image, hence no further mip-mapping is required.

The methods of the present invention then proceed by constructing the respective grids or cells ($I_{k,i,j}$) for each mip-map. Each nonempty image cell $I_{k,i,j}$ now may be downloaded. Larger values of k cover more area within the original image but provide lower detail on that area. The task at hand is now to determine, given the viewing frustum and the list of previously downloaded image cells $I_{k,i,l}$, downloading which grids will improve the quality of the display as fast as possible, considering the download rate as fixed. The scheme used to implemented the downloading sequence of these cells is by constructing a tree, starting from $I_{N-6,0,0}$ and expanding a quadtree towards the lower mip-map levels. (Quadtrees are data structures in which each node can have up to four child nodes. As each 64x64 pixel image in the grid 1_k has exactly four matching 64x64 pixel images on the grid 1_{k-1} covering the same area, the data structure is built accordingly.)

For every frame that is rendered , begin with the cell that covers the area of the entire original image, $I_{N-6,0,0}$. For each cell under consideration, compute the principle mip-map level that should be used to draw it. If it is lower than the mip-map level of the cell, subdivide the cell to four smaller cells and use recursion. If this operation attempts to draw over areas that do not yet have image cells at a low enough mip-map level to use with them, the recursion stops.

If the principle mip-map level is equal or higher than the level of the cell, then the cell is rendered using the cell of the principle mip-map level, which is the parent of that cell in the Quad-tree, at the appropriate level. Then download the cells in which the difference between the principle mip-map level to the mip-map level of the image cell actually used is the highest. Downloading is asynchronous; the renderer maintains a priority queue of download requests, and separate threads are downloading images. Whenever a download is complete, another download is initiated immediately, based on the currently highest-priority request.

The principle mip-map level of an image cell is determined by the screen resolution, FOV (field of view) angle, the angle formed between the image's plane normal and the line connecting between the camera and the position within the cell that is closest to the camera, and a few other factors. The equation, which uses the above information, approximates the general mip-mapping level equation:

$$l = \max(0, \log_4 (T/S))$$

in which S is the surface of the cell as displayed on the screen during rendering (in pixels), and T is the surface of the cell within the texture being mapped (in pixels).

When rendering a cell of the grid I_k,

$$T = N^2 2^{-k}$$

and

$$S = xy\cos(a)\mathrm{ctg}^2(0.5FOV)t^2\ T / z^2$$

where x is the display's x-resolution, y is the display's y-resolution, FOV is the field-of-view angle, a is the angle between the image's plane normal and the line connecting the viewpoint and the point in the cell of shortest distance to it, t is the length of the square each pixel in the original image is assigned to in 3D, and z is the height of the camera over the image's plane.

This arrives at the equation:

$$l = \log_4 (z^2 /(xy\cos(a)\mathrm{ctg}^2(0.5FOV)t^2))$$

$$l = \max(0, \min(l, M))$$

For example, using a 64x64 target grid display to render the image from a view of height N with FOV angle of 90 degrees, with the length of each pixel in space being one, the entire target image can be fitted precisely to the display as demonstrated by:

$$l = \log_4 (N^2/(64^2 \bullet 1 \bullet 1 \bullet 1^2)) = M$$

Image Quality Management at T-Junctions:

Note that the geometry (polygons) generated by quadtree scheme is non-manifold, due to a problem shared among all adaptively subdivision triangulation schemes, known as the T-junction cracking problem, where an image parcel is adjacent to two smaller image parcels. In the case of the present invention, all parcels are 64x64 pixel arrays, where the parcels for smaller dimensioned grids represent a correspondingly higher resolution. The spatial discontinuity created by the difference in resolutions, specifically between one grid and the sub-grids of an adjacent grid, results in undesirable display artifacts.

The present invention provides a solution to this problem by converting the polygon, or in the present instance, grid mesh into a Manifold surface by adding vertices along edges connecting grids of different cell levels. The addition of new vertices, where necessary, is done efficiently, involving only constant time per vertex added.

The algorithm of the present invention works as follows: an 8-bit square map is created, in which the edge length is 2 + N/64. Where the target image size in pixels is (X, Y), N is the smallest power of 2 that is equal or greater than max {X,Y}. For each frame rendered, the contents of this map are reset to zero. Each cell that is rendered, is also drawn as a square on the map, corresponding to the area it occupies, using the number M - l as a color, where l is the level of the grid the polygon is upon, where M is $M = \log_2(N) - 6$.

The boundaries of the map remain set to zero while the cells are drawn. When each of the polygons is rendered, its boundaries on the map are checked. Pixels on the map are evaluated to check if any vertices should be added. Locations that can be predicted mathematically are not read from the map and are skipped. Consequently, the process implemented by the present invention is efficient and, in particular, more efficient than searching within traditional data structures such as the Quad-tree as an approach to preventing the occurrence of T-junction based artifacts.

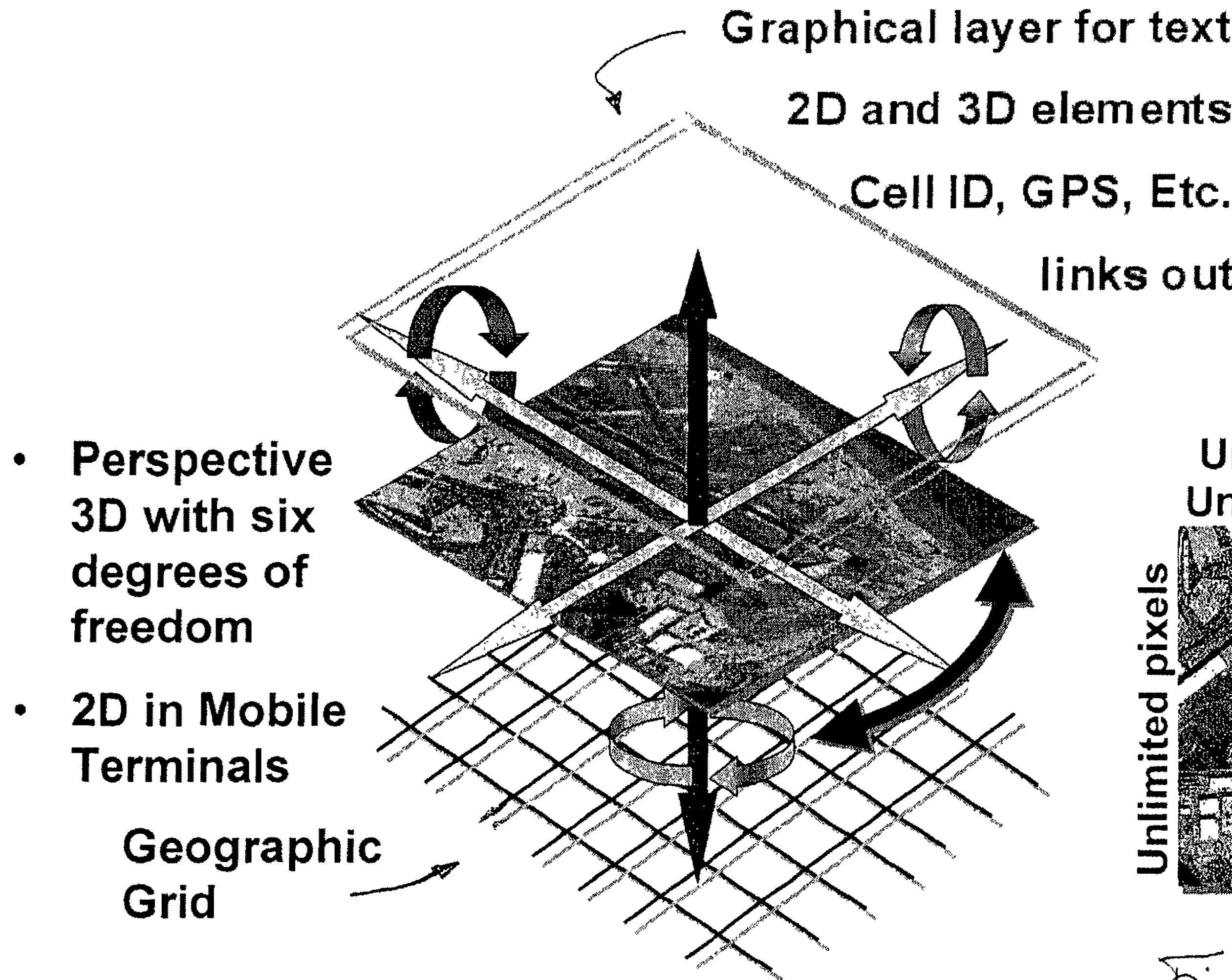

Figure 3

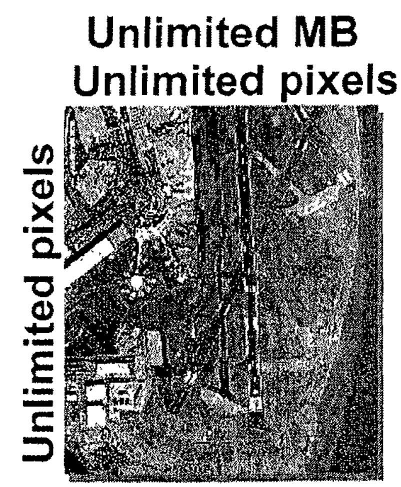

Figure 1.

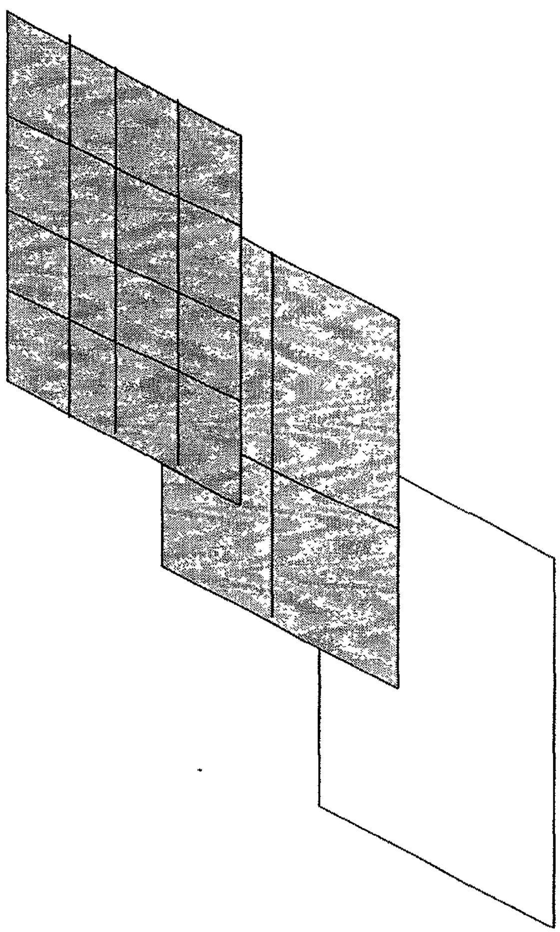

FIG.2

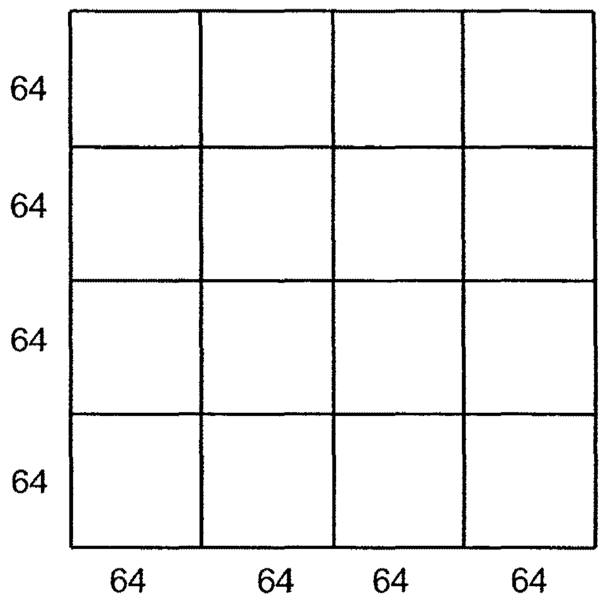

FIG. 4

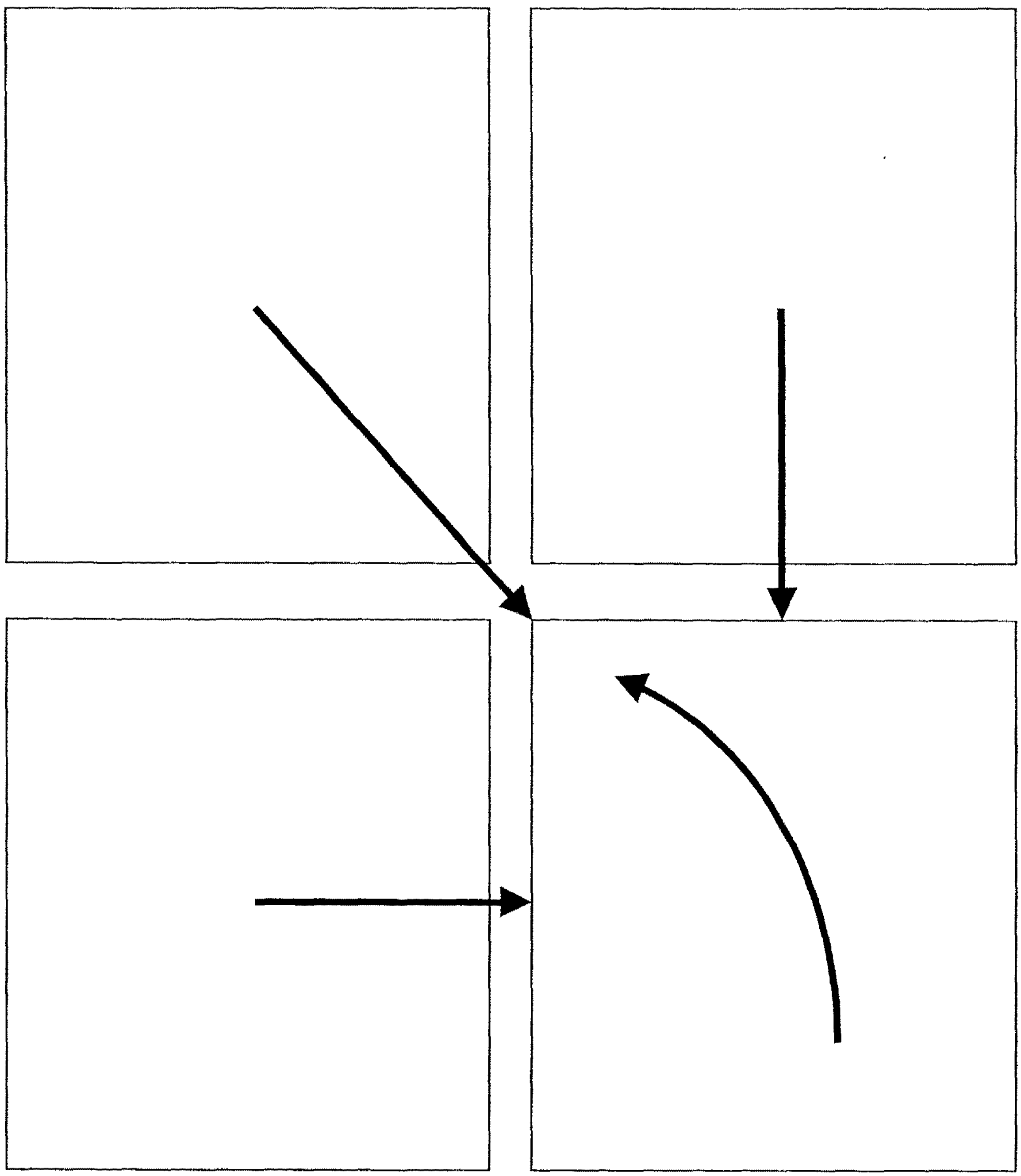

FIG.5

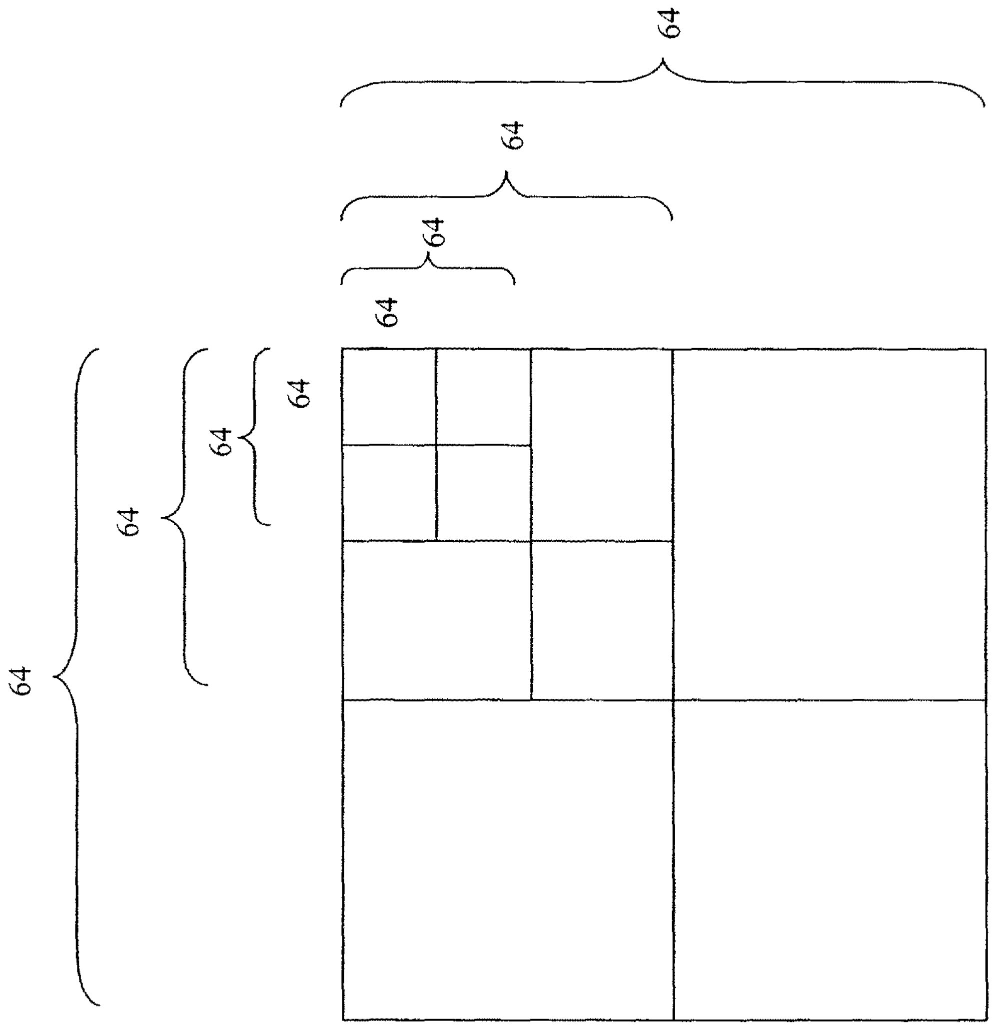

FIG. 6