7. B. W. Kernighan and D. M. Ritchie, *The C Programming Language*. Prentice-Hall, New Jersey (1978).
8. C. J. Prosser, Graphical output methods and their relation to display systems design, *M.Sc. thesis*, University of Glasgow (1981).
9. C. J. Prosser and A. C. Kilgour, Generating conic sections using integer arithmetic, *Report CSC-82-R1*, University of Glasgow Computing Science Department (1982).
10. F. R. A. Hopgood and R. W. Witty, PERQ and advanced raster graphics workstations, *Computer Graphics and Applications* **2** (No. 7), 9–15 (1982).
11. R. Hutchings, Private communication (1982).

Received July 1982

Comment on 'The Explicit Quad Tree as a Structure for Computer Graphics'

Woodwark[1] has proposed an indexing scheme for representing a complete quad tree without using pointers. The resulting data structure may be used to store pictorial information with pixel values stored in leaves. We propose the use of an alternative indexing scheme which is more suitable for machines having virtual memory. In addition, we propose that average intensity values should be stored in higher level nodes. Storage and processing efficiency are considered.

The quad tree or exponential pyramid data structure has been proposed as a mechanism which records various resolution versions (from fine to coarse) of a picture.[2] The structure may be viewed as a balanced 4-ary tree defined as follows: the root of the tree or 'father' node contains information about the entire picture. The father node is partitioned into 4 (sub)quadrants or 'son' nodes which contain information about the subquadrants. By viewing each son node as a father node and repeating the process, previously defined quadrants can be further partitioned into 4 subquadrants up to the level where the 'leaf' nodes or pixels contain the information obtained by 'raster-scanning' devices.

Woodwark[1] has proposed an indexing scheme for representing a complete quad tree without using pointers. We have found that an alternative indexing scheme, described on p. 401 of Knuth,[3] offers several advantages over the scheme proposed by Woodwark.

In describing this alternative indexing scheme, we will assume that the original picture contains $N \times N$ pixels, where N is a power of two. It is easy to show by induction that a complete quad tree contains $(4N^2 - 1)/3$ nodes. The values associated with these nodes may be stored in an array of length $(4N^2 - 1)/3$ as follows. The value associated with the root is stored in the first array element. For other nodes, the children of the node associated with location i are associated with locations $4i - 2$, $4i - 1$, $4i$ and $4i + 1$. The father of the node associated with location i is associated with location $\lfloor (i + 2)/4 \rfloor$. Hence the children of the root are associated with the second to fifth elements of the array, the grandchildren are associated with the next sixteen array elements, and so forth.

As with Woodwark's scheme, each level of the pyramid is stored as a block in consecutive storage locations. However, with the scheme proposed here, the entire quad tree is treated as a single array and indices are of fixed length. Index calculations can be performed easily in a high level language without bit manipulation facilities.

The top levels of the quad tree are stored together, which is useful in the case of a virtual memory machine. In addition, all descendants at a given level of a given node are located together. (For example, all great grandchildren of a given node are located together.) If there is at least one page available for each of the lower levels of a deep tree (recalling that the higher levels can share a single page) then page turns may be considerably reduced when traversing a quad tree. Page turns also are likely to be reduced in local operations, such as following a boundary between two regions.

The overheads of storing and using a quad tree rather than just the bottom level pixels are small when the above indexing scheme is used. Storage requirements are increased by just under $\frac{1}{3}$ (since there are $(4N^2 - 1)/3$ array elements rather than N^2 pixels).

We have found it useful to store in each non-leaf node the average intensity of the four children of the node. In this way, we can obtain different versions of a picture, having different resolutions, by going to different depths in the quad tree. We may optionally use an additional bit to indicate that a node is the root of a constant subtree. (Woodwark used a special value, called *transparent*, for all nodes which were not roots of constant subtrees.)

The proposed data structure has a clear advantage over a pixel array when many picture operations may be performed on low resolution approximations to the original picture. In these cases, only the upper levels of the quad tree need be considered.

In many cases it may be reasonable to attempt to solve a problem by first considering a low resolution approximation and then going on to higher resolution approximation if necessary. The number of nodes in the top k levels of a quad tree is $(4^k - 1)/3$. (If $k = 1 + \log_2 N$, as is the case with the full tree, then $(4^k - 1)/3 = (4N^2 - 1)/3$.) Hence, if we look at the top level, then the top two levels, and so forth until we traverse the entire quad tree, then we must examine $1 + 5 + \cdots + (4N^2 - 1)/3 = 16/9N^2 - 7/9 - (\log_2 N)/3 < 16/9N^2$ nodes. So if we examine the picture at every possible resolution down to and including the resolution at which we are able to solve the problem, and traverse the entire tree down to the particular resolution in each case, then the number of array element accesses is less than $16/9N^2$. Looking at all lower resolution approximations increases our work by only about $\frac{1}{3}$. This is because the vast majority of nodes in a quad tree are leaves. If there is a reasonable chance that we can solve a problem with a low resolution approximation to a picture, it is often worth trying since the cost of failure is small relative to the cost of inspecting the entire picture.

F. WARREN BURTON
School of Computing Studies
University of East Anglia
Norwich NR4 7TJ
UK

J. G. KOLLIAS
Department of Computer Science
National Technical University of Athens
9 Heroon Polytechniou Ave
Genikes Edres
Zografou
Athens (621)
Greece

References

1. J. R. Woodwark, The explicit quad tree as a structure for computer graphics. *The Computer Journal* **25** (2), 235 (1982).
2. A. Klinger, Regular decomposition and picture structure. *Proceedings 1974 IEEE Systems, Man and Cybernetics Conference*, pp. 307–310 (October 1974).
3. D. E. Knuth, *The Art of Computer Programming, Fundamental Algorithms*, Vol. 1. Addison-Wesley, Reading, Massachusetts (1968).

Received September 1982