إستخدام الأشجار الرباعية و تقسيم المسافة الهرم

أ.م مكيّة كاظم حمد
كلية العلوم - جامعة بغداد

م.م مها شاكر إبراهيم

مجلة الدنانير

Abstract

A quad tree :is a nonlinear data structure in which each internal node has exactly four children without information. The information will be stored only in leaves. Quad trees are most often used to partition a two dimensional space into four equal squares then recursively partition these squares into smaller squares until each square contains a suitably uniform subset of the input . In this research we describe the characteristic of the structure and algorithms to represent bitmap images by subdividing it until each square has the same color value.

المستخلص

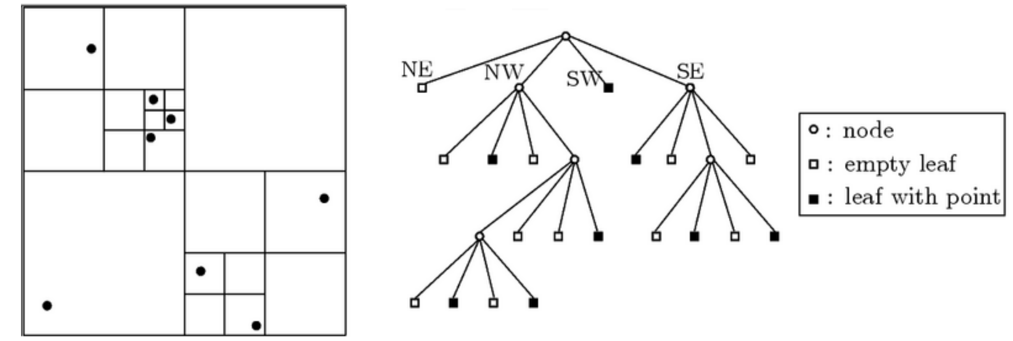
الشجرة الرباعية هي نوع من هياكل البيانات غير الخطية التي تكون العقد الداخلية لها تتكون من أربعة أبناء بدون معلومات. تكون المعلومات مخزونة في الأوراق فقط. تستخدم الأشجار الرباعية عادة لتقسيم المساحة الثنائية الأبعاد الى اربعة مربعات متساوية ثم يستمر التقسيم عن طريق الاستدعاء الذاتي لكل المربعات و تقسيمها الى مربعات أصغر إلى أن يحتوي كل مربع على مجموعة جزئية موحدة من المدخلات، في هذه البحث وصف لخصائص هذا النوع من الاشجار والخوارزميات المستخدمة لتمثيل الصور عن طريق تقسيم الصورة إلى مربعات الى ان يحتوي كل مربع منها على نفس القيمة اللونية.

Keywords:Quad tree, spatial data structures , image decomposition.

1.Introduction

A quad tree is a non linear data structure used to describe a class of hierarchical data structures. The common property is that they are based on the principle of recursive decomposition in which each internal node has up to four children, as in figure (1). Quad trees are most often used to partition a two dimensional space by recursively subdividing it into four quadrants or regions. The regions may be square or rectangular, or may have arbitrary shapes.

According to quad tree partitioning scheme a two dimensional space is divided using two orthogonal lines, into four areas. The areas are named: NE (north-east), NW (north-west), SE (south-east), and SW (south-west).This splitting process is recursively applied and a result is arranged in a hierarchical structure.[1][2][3]



Figure(1) Quad tree representation

Quad tree types

There are two common types of the quad tree data structures these are :

2.1 The region quad tree

The region quad tree represents a partition of space in two dimensions by decomposing the region into four equal quadrants, sub quadrants, and so on with each leaf node containing data corresponding to a specific sub region. Each node in the tree either has exactly four children, or has no children (a leaf node). Figure (2) represent region quad tree .[1][2]

2.2 The Point quad tree

The point quad tree is an adaptation of a binary tree used to represent two dimensional point data. It shares the features of all quad trees but is a true tree as the center of a subdivision is always on a point, Figure (3) represent point quad tree. The tree shape depends on the order data is processed. It is often very efficient in comparing two dimensional ordered data points, usually operating in O(log n) time.[4][5]

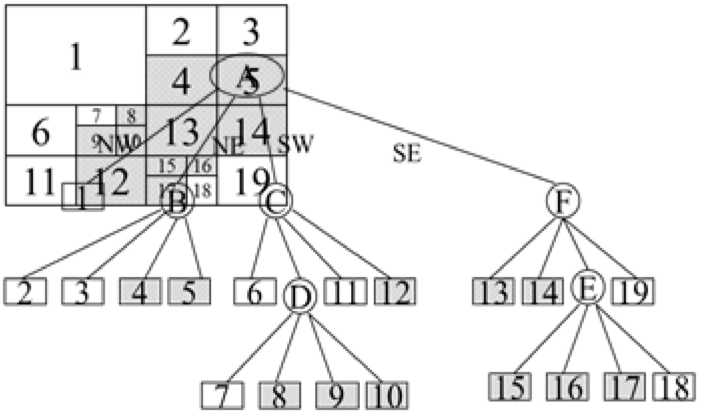


Figure (2) region quad tree [5]

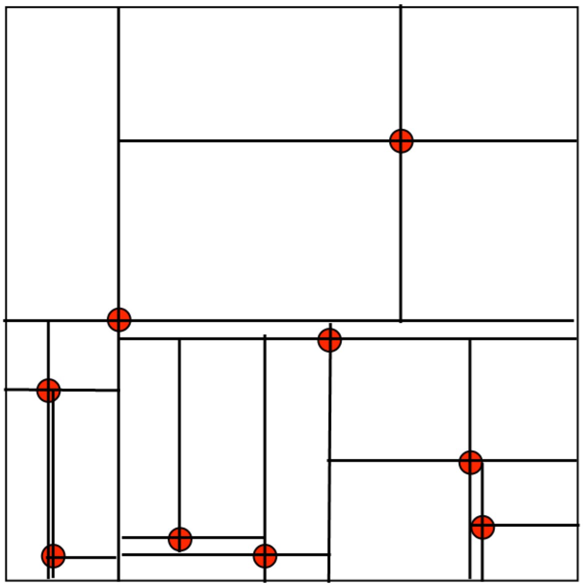


Figure (3) point quad tree

3. Node structure for a quad tree

A node of a quad tree is similar to a node of a binary tree, with the major difference being that it has four pointers (one for each quadrant) instead of two ("left" and "right") as in an ordinary binary tree. Also a key is usually decomposed into two parts, referring to x and y coordinates. Therefore a node contains following information:

- 4 Pointers: quad[‘NW’], quad[‘NE’], quad[‘SW’], and quad[‘SE’]
- point; which in turn contains:
- key; usually expressed as x, y coordinates.
- value; for example a name.[2]

4. Searching in quad tree

Searching for a record matching point Q in the quad tree is straight forward beginning at the root, we continuously branch to the quadrant that contain Q until our search reach a leaf node. If the root is a leaf, then just check to see if the node's data record matches point Q. if the root is an internal node, proceed to the child that contain the search point. For example, the NW quadrant of figure (4) contains points whose x and y values each fall in the range 0 to 63. The NE quadrant contains points whose x value fall in the range 64 to 127 and whose y value falls in the range 0 to 63. If the roots child is a leaf node then that child is checked to see if Q has been found. If the child is another internal node, The search process continues through the tree until a leaf node is found. If this leaf node stores a record whose position matches Q then the query is successful; otherwise Q is not is not in the tree. [1]

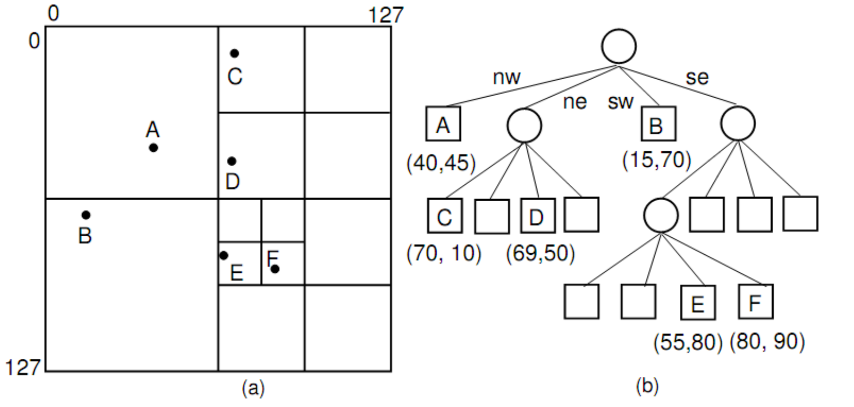


Figure (4) Example of a quad tree . (a) A map of data points with region size (128×128). (b) the quad tree for the points in (a).

5. Insertion in quad tree

Inserting record P into the quad tree is performed by first locating the leaf node that contains the location of P. if this leaf node is empty, then P is stored at this leaf. If the leaf already contains P (or a record with P's coordinates), then a duplicate record should be reported. If the leaf node already contains another record, then the node must be repeatedly decomposed until the existing record and P fall into different leaf nodes. Figure (4) show an example of such an insertion [5].

6. Deletion in quad tree

Deleting a record P is performed by first locating the node N of the quad tree that contains P, node N is then changed to be empty. The next step is to look at N's three siblings.N and it's siblings must be merged together to form a single node N' if only one point is contained among them. This merging process continues until some level is reached at which at least two points are continued in the sub trees represented by N' and its sibling. For example if point C was deleted from the quad tree in figure (4,b) the resulting node must be merged with its siblings and that larger node again merged with its siblings to restore the quad tree to the decomposition of figure(4,a).

7. Quad trees performance

There are some advantages that concluded from performing this data structure, these advantages were as follow,[6]:

1) Very efficient in certain spatial analysis functions (e.gnearest neighbor, point-in- polygon search).
2) High compression for relatively homogeneous data.
3) High processing time needed to create and modify the tree.
4) Non significant compression for relatively heterogeneous data.

8. Applications of Quad trees

Quadtrees have common uses in the different applications such as [2]:

- Image representation.
- Spatial indexing.
- Efficient collision detection in two dimensions.
- Storing sparse data, such as a formatting information for a spreadsheet or for some matrix calculations.
- Solution of multidimensional fields (computational fluid dynamics, electromagnetism) .
- Quad trees are also used in the area of fractal image analysis.

9. Image representation using quad trees

This research used Quad trees for binary images representation, such that each node in the quad tree is labeled either B (black), W (white) or G (gray) and the leaf nodes can only be B or W, such that if a quad tree with a depth of n wanted to be used to represent an image consisting of 2n × 2n pixels, each leaf must represent a uniform area of the picture. If the picture is black and white, we only need one bit to represent the color in each leaf; for example, 0 could mean black and 1 could mean white.[7]

9.1 Algorithm for construction of a quad tree for an (N × N) binary image

For image representation using quad trees, we proposed the following algorithm:

1) If the binary images contains only black pixels, label the root node B and quit.
2) Else if the binary image contains only white pixels, label the root node W and quit.
3) Otherwise create four child nodes corresponding to the 4 (N/4 × N/4) quadrants of the binary image.
4) For each of the quadrants, recursively repeat steps 1 to 3. (In worst case, recursion ends when each sub-quadrant is a single pixel). [2][8]

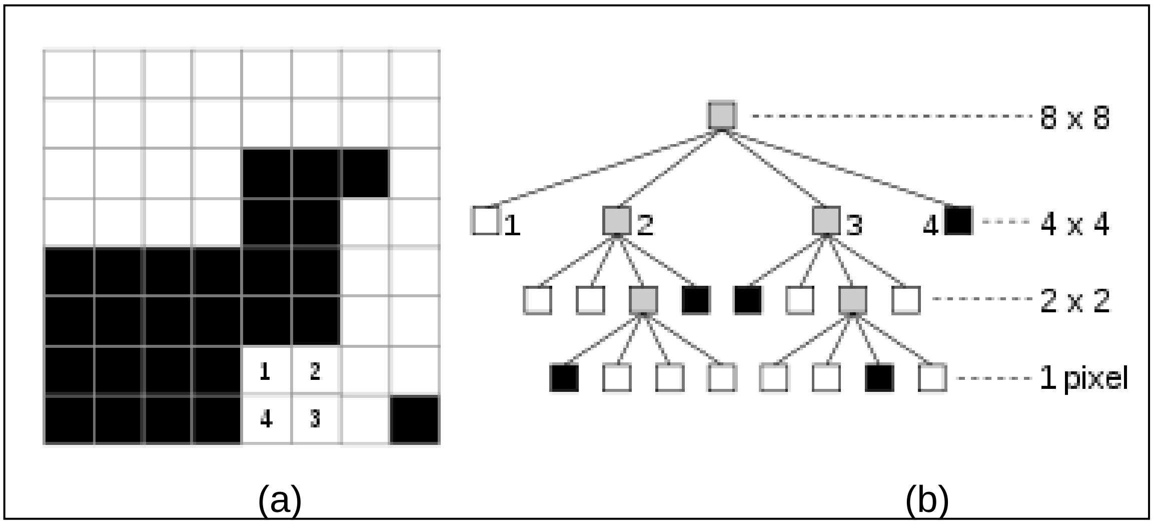


Figure (5) image representation using quad tree. (a) 8 x 8 pixel image . (b) The quad tree representation of the image in (a). The quadrants are shown in clockwise order from the top-left quadrant,[2]

9.2 Performance Of Quad Tree for Images Representation

Quad trees are used extensively in computer graphics; and this because they can be manipulated and accessed much quicker than other models. For that reason, quad trees are very popular in fractal graphics. Recursive images can be implemented easily using quad trees: the root of the quad tree has four children, where one of the children is the actual image and the other three point to the root. The advantages of using quad trees for image representation are:

- Erasing an image takes only one step. All that is required is to set the root node to neutral.
- Zooming to a particular quadrant in the tree is a one step operation.
- To reduce the complexity of the image, it suffices to remove the final level of nodes.
- Accessing particular regions of the image is a very fast operation. This is useful for updating certain regions of an image, perhaps for an environment with multiple windows.

The only disadvantage of quad trees is that they take up a lot of space.If a quad tree is implemented using links, most of the memory will be taken up by the links. Nevertheless, there are ways of compacting quad trees, which is important for transferring data efficiently.[8]

10. Conclusions

The quad tree data structure is appropriate to store composite key information, usually spatial data. Most often used to partition a region into 2d space, and this could be done by partitioning space recursively subdividing it into 4 disjoint quadrants. Quad tree implementations vary according to data represented including areas, points, lines (including poly lines /curves). The quad trees are efficient for binary image representation.

11. References

1) [1] Clifford A. Shaffer: A practical introduction to data structures and algorithm analysis,3'rd edition, January 2010.
2) [2] Wikipedia, the free encyclopedia.mht, quad trees article. http://en.wikipedia.org/wiki/quadtrees
3) [3] vorowiki encyclopedia , article "quad trees" www.voronoi.com/wiki/index.php?title=Quadtree
4) [4] Applications of Geometric / Spatial Data structures. www.cs.umd.edu/class/spring2008/cmsc420/L17-18.QuadTrees.pdf
5) [5] Snehal Thakkar and Hanan Samet Spatial Data Structures

6) [6] Geographic Database Systems Emmanuel Stefanakis Department www.dbnet.ece.ntua.gr/~stefanak/gdbs/Lecture7a.pdf

7) [7] lossless compression, CIS 658 Multimedia computing www.cis.csuohio.edu/~arndt/lossless.ppt

8) [8] Picture representation using quad trees, McGill University: School of Computer Science. Winter 1999.
www.cs.ubc.ca/pcarbo/cs251/welcome.html.