Constructing Orientation Adaptive Quadtrees

Bradford G. Nickerson
Sri Hartati
School of Computer Science
University of New Brunswick
P.O. Box 4400
Fredericton, N.B., Canada E3B 5A3

Abstract

A novel spatial data structure called the orientation adaptive quadtree (OAQ) is presented. The data structure takes advantage of large second-order moments of inertia of planar objects by aligning the quadtree axes with the object's principal axis of inertia. This provides a reduction in the number of leaf nodes necessary to store the quadtree compared to quadtrees aligned with the object coordinate axes. An efficient method for constructing OAQs for planar graphs described by edge vertices is given, along with a discussion of some test results.

Keywords: spatial data structures, quadtrees, adaptive encoding, optimal encoding.

Introduction

Quadtrees [Samet, 1984] provide a natural decomposition of two dimensional space which have been used in a wide variety of applications [Bentley and Stanat, 1975; Samet and Rosenfeld, 1980; Yerry and Shepard, 1983; Samet et al, 1985; Andersson, 1988]. An overview paper by Samet and Webber [1988] focusses on the uses of the quadtree and other hierarchical data structures for computer graphics.

The motivation for this work stems from using the quadtree structure to represent planar directed graphs (of even degree) embedded in the plane. Such graphs are commonly used to represent the boundary of planar objects. Edge-based data structures such as the doubly-connected edge list [Muller and Preparata, 1978] or winged-edge structure [Weiler, 1985] are commonly used to represent planar graphs, but quadtree representation is better suited if the processing expected is truly of an "area" nature. For example, if a "point-in-polygon" query is posed for an object represented using edge-based structures, the response time will be O(n), for n = number of edges in the planar graph, whereas the response time to the same query for a planar graph represented as a quadtree will be O(d), for d = maximum depth of the quadtree. For objects which are naturally aligned to the object space axes and whose eccentricity is small, the standard quadtree encoding is sufficient. When objects do not align themselves to the object space axes, and may have a large eccentricity (see the example in Figure 1), the standard quadtree encoding gives a large number of leaf nodes.

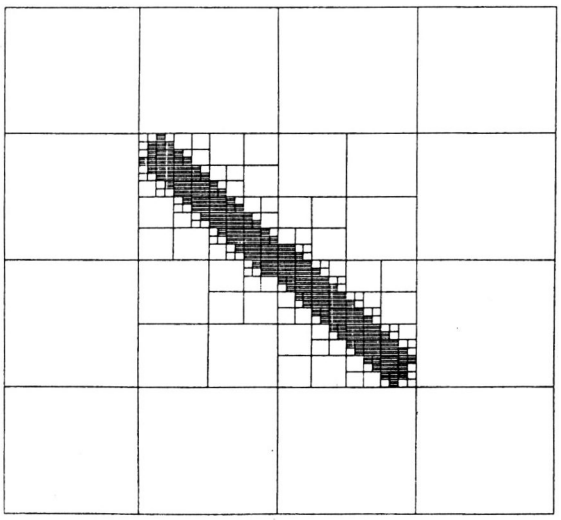


Figure 1. Example of standard quadtree encoding of an object not aligned to the quadtree axes. There are 4 original vertices, and the quadtree coordinate space is of size 64 by 64.

A better approach would be to position the quadtree axes centred at the object's centroid and aligned with the object's principal axis of inertia. This paper documents a general-purpose approach to generating these so-called

orientation adaptive quadtrees (OAQs).

Determining the Orientation of Planar Objects

In general, the two dimensional $(p+q)$th order moments of some function $f(x,y)$ defined on the Cartesian plane are given in terms of the following integral [Hu, 1962]:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p y^q f(x,y)\,dx\,dy \qquad (1)$$

for $p,q = 0,1,2,....$ For the case of an embedded planar graph of even degree, $f(x,y)$ represents the area inside the edges, and the moments are computed using $f(x,y) = 1$, with the integration taking place over the bounding rectangle of the graph.

The central moments are defined as

$$\mu_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} (x-\overline{x})^p (y-\overline{y})^q f(x,y)\,dx\,dy \qquad (2)$$

for $p,q = 0,1,2,...,$ and where $(\overline{x},\overline{y})$ represent the coordinates of the centroid. In terms of moments, the centroid is computed as

$$\overline{x} = \frac{m_{10}}{m_{00}},\quad \overline{y} = \frac{m_{01}}{m_{00}} \qquad (3)$$

where m_{00} is the area.

The central moments can also be expressed in terms of the ordinary moments [Hu, 1962; Beer and Johnston, 1962] as

$$\begin{aligned} \mu_{00} &= m_{00} = \mu \\ \mu_{10} &= \mu_{01} = 0 \\ \mu_{11} &= m_{11} - \mu\overline{x}\overline{y} \\ \mu_{20} &= m_{20} - \mu\overline{x}^2 \\ \mu_{02} &= m_{02} - \mu\overline{y}^2 \end{aligned} \qquad (4)$$

The orientation θ of the principal axis of inertia is computed as [Beer and Johnston, 1962]

$$tan2\theta = \frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \qquad (5)$$

In computer vision, the second order moments are usually computed using numerical integration. This numerical approach has the drawback of requiring significant computations to obtain accurate results. We decided to evaluate the integrals in equation (1) for m_{20} and m_{02} directly. Consider the planar graph depicted in Figure 2.

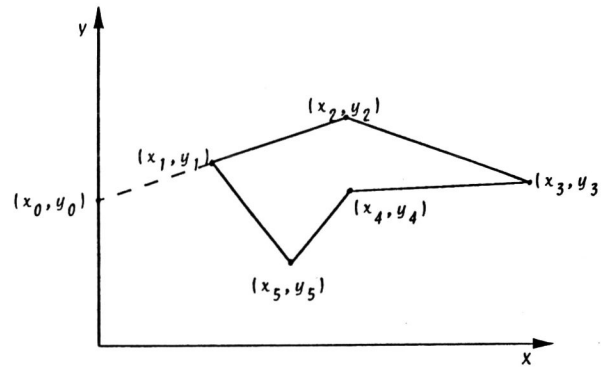


Figure 2. Simple planar graph embedded in the plane.

The second order moment m_{02} of the area under the line segment $(x_0,y_0) - (x_1,y_1)$ can be defined as

$$m_{02} = \int_0^{x_1}\int_0^{y_1} y^2\,dy\,dx = \int_0^{x_1} \frac{1}{3}y^3\,dx \qquad (6)$$

The equation of the line segment is $y = y_0 + bx$, where b is the slope. Assuming $t = bx$, then $x = \frac{t}{b}$, $dx = \frac{dt}{b}$, and for $x = 0, t = 0$; $x = x_1, t = bx_1$. Applying this substitution to the above equation, the line segment equation becomes $y = y_0 + t$, and the integral is evaluated between the limits $[0, bx_1]$. This leads to

$$m_{02} = \int_0^{bx_1} \frac{1}{3b}(y_0+t)^3\,dt \qquad (7)$$

$$m_{02} = \frac{1}{12b}((y_0+bx_1)^4 - y_0^4) \qquad (8)$$

Now, $y_0 + bx_1 = y_1$, and the second moment for the area under line segment $(x_0,y_0) - (x_1,y_1)$ is

$$m_{02}^1 = \frac{1}{12b}(y_1^4 - y_0^4) \qquad (9)$$

A similar derivation can be carried out to compute the second moment under line segment $(x_0,y_0) - (x_2,y_2)$, which results in

$$m_{02}^2 = \frac{1}{12b}(y_2^4 - y_0^4) \qquad (10)$$

The second order moment under line segment $(x_1,y_1) - (x_2,y_2)$ can be derived by taking the difference $m_{02}^2 - m_{02}^1$, which gives

$$m_{02}^{1,2} = \frac{1}{12b}(y_2^4 - y_1^4) \qquad (11)$$

This equation shows that the second order moments of an area under a line segment of the boundary of a planar graph is dependent only on the slope of the line segment and its end-point coordinates. In general, the second order moment for line segment $(x_i,y_i) - (x_{i+1},y_{i+1})$ can be stated as

$$m_{02}^{i,i+1} = \frac{1}{12b}(y_{i+1}^4 - y_i^4) \qquad (12)$$

If the slope b is zero or close to zero, as shown in Figure 3, an alternative formulation is used to avoid ill-conditioning.

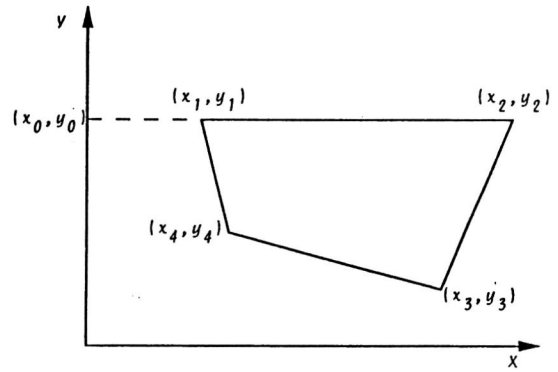


Figure 3. Planar graph embedded in the plane with $b = 0$.

Evaluating the second order moment here under line segment $(x_0, y_0) - (x_1, y_1)$ gives

$$m_{02} = \int_0^{x_1} \int_0^{y_1} y^2 \, dy \, dx = \int_0^{x_1} \frac{1}{3} y^3 \, dx \qquad (13)$$

Using the line segment equation $y = y_0 + bx$ with $b = 0$ results in

$$m_{02}^{1_h} = \frac{1}{3} y_0^3 x_1 \qquad (14)$$

A similar derivation for line segment $(x_0, y_0) - (x_2, y_2)$ leads to

$$m_{02}^{2_h} = \frac{1}{3} y_0^3 x_2 \qquad (15)$$

The difference $m_{02}^{2_h} - m_{02}^{1_h}$ gives the second order moment $m_{02}^{1,2_h}$ under the horizontal line segment $(x_1, y_1) - (x_2, y_2)$ as

$$m_{02}^{1,2_h} = \frac{1}{3} y_0^3 (x_2 - x_1) = \frac{1}{3} y_1^3 (x_2 - x_1) \qquad (16)$$

since $y_0 = y_1 = y_2$. The general expression for horizontal line segments $(x_i, y_i) - (x_{i+1}, y_{i+1})$ can now be written as

$$m_{02}^{i,i+1_h} = \frac{1}{3} y_i^3 (x_{i+1} - x_i) \qquad (17)$$

The second order moments for a complete planar object represented by an embedded graph of even degree is computed by adding the contributions of all the edges, taking into account whether or not the edge is horizontal or vertical. After similar derivations to those shown above, the complete second and first order moments for a planar object are as follows:

$$m_{02} = \sum_{i=1}^{n} \begin{cases} \frac{1}{12 b_i} (y_{i+1}^4 - y_i^4) & \text{for } b_i \neq 0 \\ \frac{1}{3} y_i^3 (x_{i+1} - x_i) & \text{for } b_i = 0 \end{cases} \qquad (18)$$

$$m_{20} = \sum_{i=1}^{n} \begin{cases} \frac{b_i}{12} (x_{i+1}^4 - x_i^4) & \text{for } b_i \neq \infty \\ \frac{1}{3} x_i^3 (y_{i+1} - y_i) & \text{for } b_i = \infty \end{cases} \qquad (19)$$

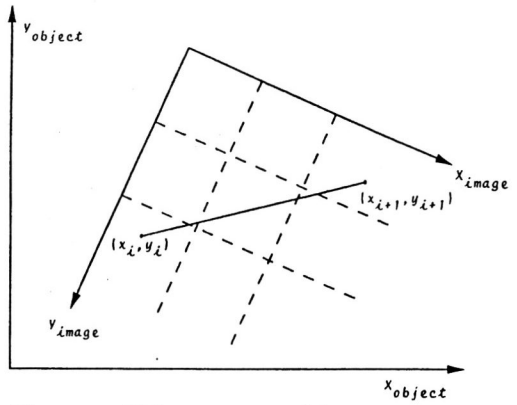


Figure 4. Object space and image space.

$$m_{01} = \sum_{i=1}^{n} \begin{cases} \frac{1}{6 b_i} (y_{i+1}^3 - y_i^3) & \text{for } b_i \neq 0 \\ \frac{1}{2} y_i^2 (x_{i+1} - x_i) & \text{for } b_i = 0 \end{cases} \qquad (20)$$

$$m_{10} = \sum_{i=1}^{n} \begin{cases} \frac{b_i}{6} (x_{i+1}^3 - x_i^3) & \text{for } b_i \neq \infty \\ \frac{1}{2} x_i^2 (y_{i+1} - y_i) & \text{for } b_i = \infty \end{cases} \qquad (21)$$

$$m_{11} = \sum_{i=1}^{n} \begin{cases} \frac{(y_{i+1} - y_{0_i})^2}{24 b_i^2} a_{i+1} - \frac{(y_i - y_{0_i})^2}{24 b_i^2} a_i & \text{for } b_i \neq 0 \\ \frac{y_i^2}{4} (x_{i+1}^2 - x_i^2) & \text{for } b_i = 0 \end{cases} \qquad (22)$$

where n = the number of vertices, vertex $n+1 \equiv$ vertex 1, $a_{i+1} = (y_{0_i}^2 + 2 y_{0_i} y_{i+1} + 3 y_{i+1}^2)$, $a_i = (y_{0_i}^2 + 2 y_{0_i} y_i + 3 y_i^2)$, and b_i is the slope $\frac{y_{i+1} - y_i}{x_{i+1} - x_i}$. Notice that the expression for m_{11} necessarily involves the intersection y_{0_i} of the ordinate axis with the line segment $(x_i, y_i) - (x_{i+1}, y_{i+1})$. The orientation of any planar object represented by the coordinates of its edge vertices can be computed using these equations in conjunction with equations (4) and (5).

Conversion from Edges to OAQs

Once the orientation θ has been determined, the edge-based structure is converted to a quadtree whose axes are aligned with this direction. Figure 4 depicts the two spaces involved; i.e. the original object space and the differently oriented image space. One edge of the original object is also shown. The dashed lines represent unit increments in the image space coordinate frame. Conversion of planar graphs of even degree to OAQs proceeds via the following five steps:

1. Convert
   all edge vertices from object coordinates to image coordinates using an orthogonal transformation.
2. Use the Bresenham algorithm [e.g. Foley and Van Dam, 1982; Rogers, 1985] to obtain a raster representation of the graph in image space.
3. Obtain the tightly closed boundary (TCB) representation of this raster represenatation

```
vertex = record
   x,y : real
end;
pqtree = ^qtree; {pointer to a quadtree}
qtree = record   {node of a quadtree}
   colour : (B,W,G);
   NW,NE,SW,SE : pqtree;
   parent : pqtree
end;
OAQ = record     {Orientation Adaptive}
   qt : pqtree;  {  Quadtree          }
   orientation : real;
   centroid : vertex
end;
```

Figure 5. Pascal record structure for an orientation adaptive quadtree.

using the augmentation algorithm of Merrill [1973]. This ensures topological consistency in that one can always infer directly from the TCB whether a point is interior to the object represented by the graph or not.

4. Sort the TCB lexicographically using the y coordinate as the principal sort field. This produces a run-length coded (rlc) description of the graph.
5. Using this rlc description, generate the quadtree corresponding to it using a modified Moreton matrix approach [Samet, 1980]. The modification made here is that the decision about whether an image space point is interior to the object is inferred from the rlc description. This obviates the requirement to store a binary array of size (m,m), where m is the maximum image space coordinate.

Once this process is complete, the OAQ structure is identical to the ordinary quadtree, except for the storage of the orientation (and centroid) at the root. The Pascal record structure for an OAQ is given in Figure 5. The ^ stands for the normal Pascal ↑ pointer type. The five steps listed above are also used to compute the standard quadtree, except that the transformation of step 1 does not require a rotation.

Test Results

The long skinny object shown in Figure 6 was encoded using both OAQs and standard quadtrees. This object was rotated by increments of -10° from 0 to -50°. Image space sizes of 16 by 16 up to 512 by 512 were used to encode the object. The results are shown in Table 1. Notice that the number of nodes for the OAQ is constant for varying image size and orientation, but the number of nodes for the standard quadtree basically doubles for every doubling of the image space coordinate range.

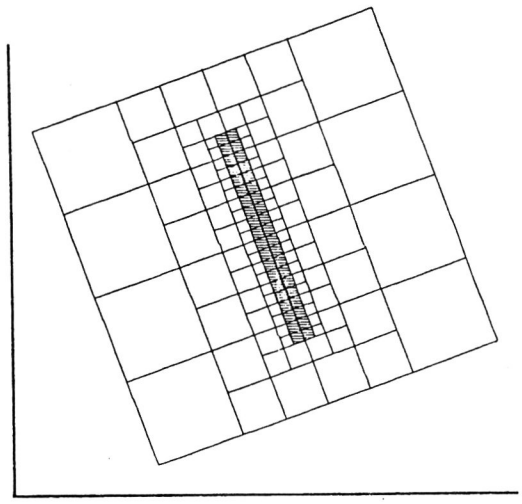


Figure 6. Example test object encoded as an OAQ (rotation 20°). The non-rotated corner coordinates are (-0.5,5), (0.5,5), (0.5,-5),(-0.5,5).

Table 1. Number of nodes for Figure 6 object encoded as an OAQ and as a standard quadtree (St.Q).

OAQ	Image Size					
θ	16x16	32x32	64x64	128x128	256x256	512x512
0	65	181	181	181	181	181
-10	65	181	181	181	181	181
-20	77	181	181	181	181	181
-30	77	181	181	181	181	181
-40	45	181	181	181	181	181
-50	65	181	181	181	181	181

St.Q	Image Size					
θ	16x16	32x32	64x64	128x128	256x256	512x512
0	53	181	181	181	181	181
-10	73	141	245	525	1053	2117
-20	65	131	285	573	1149	2341
-30	73	141	309	613	1237	2493
-40	61	101	333	621	1261	2597
-50	69	137	317	613	1285	2565

Table 2. Number of nodes for the road object encoded as an OAQ and as a standard quadtree (St.Q).

	Image Size				
	16x16	32x32	64x64	128x128	256x256
OAQ	73	133	365	681	1365
St.Q	85	185	401	857	1841

To evaluate the OAQ performance with real data, a section of the centerline of a road from a map was used. An even degree graph representing the road was established by forming a polygonal band around the centerline, and using the vertices of this polygon to form the graph. The OAQ orientation θ was computed as -22.52°, and the OAQ and standard quadtree encoding were done for various image sizes. Figure 7 shows both the OAQ and standard encoding of the road object for an image space size of 64x64.

The number of nodes required for different image sizes is shown in Table 2. Although the savings in nodes as not as dramatic as for the rectangle of Figure 6, there is, on average, a 25% savings in the number of nodes required for the OAQ compared to the standard quadtree.

Conclusions

The orientation adaptive quadtree takes advantage of large eccentricity in an object and accounts for any rotation of the principal axis of inertia. In this sense, it provides an optimal (in terms of space) quadtree encoding of a connected planar object. Computing the second order moments necessary to obtain the orientation requires O(n) time, for n = number of edges in the original object description. This extra time does not add significantly to the overall time of converting edge structures to quadtree structures which is $O(n^2)$, for n = image space coordinate range.

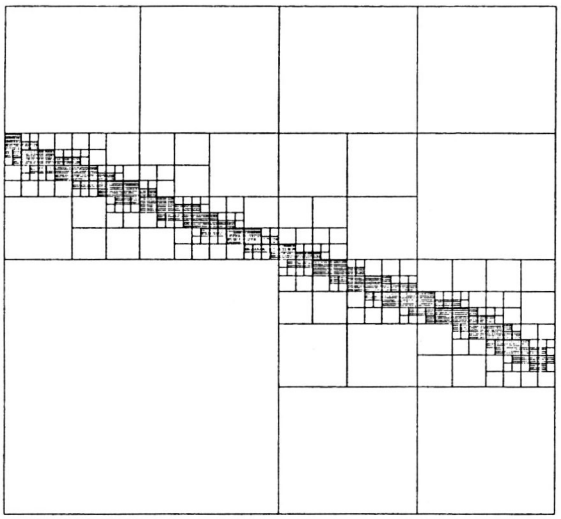
Acknowledgements

This research was supported, in part, by a grant from the Natural Sciences and Engineering Research Council (NSERC) of Canada. Facilities to carry out this research were largely provided by the University of New Brunswick, School of Computer Science.

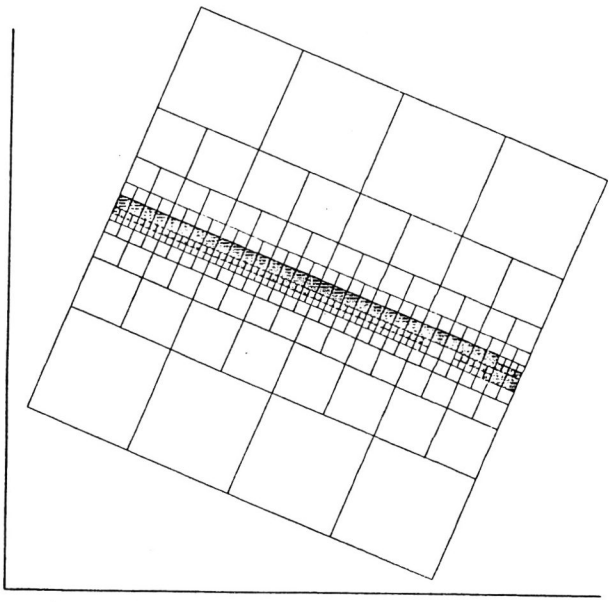
References

Andersson, R.L. *A Robot Ping-Pong Player*, MIT Press, Cambridge, MA, 1988.

Beer, F.P. and Johnston, E.R. *Mechanics for Engineers: Statics and Dynamics*, 2nd edition, McGraw-Hill, New York, 1962.



(a)



(b)

Figure 7. (a) Standard quadtree encoding of road object. (b) OAQ encoding of the same road object. An image space size of 64x64 was used for both cases.

Bentley, J.L. and Stanat, D.F. "Analysis of range searches in quad trees", *Information Processing Letters*, vol.3, no.6, 1975, pp.170-173.

Foley, J.D. and Van Dam, A. *Fundamentals of Interactive Computer Graphics*, Addison-Wesley, Reading, MA, 1982.

Hu, M.K. "Visual pattern recognition by moment invariants", *IRE Transactions on Information Theory*, vol.It-8, 1962, pp.179-187.

Merrill, R.D. "Representation of contours and regions for efficient computer search", *Comm. of the ACM*, vol.16, no.2, 1973, pp.69-92.

Muller, D.E. and Preparata, F.P. "Finding the intersection of two convex polyhedra", *Theoretical Computer Science*, vol.7, no.2, 1978, pp.217-236.

Rogers, D.F. *Procedural Elements for Computer Graphics*, McGraw-Hill, New York, 1985.

Samet, H. "Region representation: quadtrees from binary images", *Computer Graphics and Image Processing*, vol.13, 1980, pp.88-93.

Samet, H. "The quadtree and related hierarchical data structures", *ACM Computing Surveys*, vol.16, no.2, 1984, pp.187-260.

Samet, H. and Rosenfeld, A. "Quadtree structures for image processing", *Proc. of the 5th Int. Conf. on Pattern Recognition*, (Miami Beach, FL), Dec. 1980, IEEE, New York, pp.815-818.

Samet, H., Rosenfeld, A., Shaffer, C.A., Nelson, R.C., Huang, Y.G. and Fujimura, K. "Application of hierarchical data structures to geographical information systems phase IV", University of Maryland, Computer Vision Laboratory Technical Report CS-TR-1578, 1985.

Samet, H. and Webber, R.E. "Hierarchical data structures and algorithms for computer graphics", "Part I : Fundamentals" in *IEEE Computer Graphics & Applications*, May, 1988, pp.48-68; "Part II : Applications" in *IEEE Computer Graphics & Applications*, July, 1988, pp.59-75.

Weiler, K. "Edge-based data structures for solid modelling in curved-surface environments", *IEEE Computer Graphics & Applications*, Jan. 1985, pp.21-40.

Yerry, M.A. and Shepard, M.S. "A modified quadtree approach to finite element mesh generation", *IEEE Computer Graphics & Applications*, vol.3, no.1, 1983, pp.39-46.