© 2015 IJSRSET | Volume 1 | Issue 4 | Print ISSN : 2395-1990 | Online ISSN : 2394-4099
Themed Section: Engineering and Technology

Optimizing the Search in Hierarchical Database using Quad Tree

Deepak Kumar Sharma, Sonia Vatta
School of Computer Science and Engineering Bahra University Waknaghat Solan, Himachal Pradesh, India

ABSTRACT

This research work is done to optimize the search in hierarchal database by using Quad Tree. Quad Tree is a tree which consists of a node having 4 sub child nodes.[1] Quad Tree is used in many fields such as computer graphics to find out a desired pixel in graphics and collision detection in 3-D gaming. There are various other applications of quad tree in different fields like weather forecasting, geographical survey and study of mutation rate with a change of environment. The main focus of this work was optimizing the search for which the main areas of concentration were various searching techniques using Quad Tree and finding out the best solution. To fulfill the purpose, a new approach is designed which results in better representation of data and enhancing the speed of search. This work is done on the basis of the value of the root node. Further based on this value, the ranges of nodes at each level are calculated. The results are also included with snapshots and comparison is given between earlier and proposed techniques.
Keywords: Quad Tree, Memory Management, Search, Result

I. INTRODUCTION

The quad tree is widely in practice in detecting collision in 2D and 3D games and pixel processing to detect image in different image resolution.[2] Many games require the use of collision detection algorithms to determine when two objects have collided, but these algorithms are often expensive operations and can greatly slow down a game.[3][4][5][6][7] As such the collision detection is very costly process in gaming . So in many games we use Quad Tree to detect collision. The Quad Tree is too used to process the polygon or images in graphics. The techniques used to process the polygons in graphics are point to point processing, edge to edge processing and region by region processing. Commonly used techniques for collision detection is spatial portioning and bit- interleaving [8][9][10][11]12].

In the research paper Quad tree: an approach of traversing and searching published in IJRITCC volume 3 issue 5 [74] the searching time of the information in database is reduce.

II. METHODS AND MATERIAL

A. Searching Algorithm

To search an element in a quad tree, the new design technique is as follows:

Steps for Searching Item

SEARCHING (ROOT)

1. Put ITEM in LOC. Take RANGE = ROOT
2. Calculate range of each sub node of level i
a. QLN has range 0 to RANGE/2^2*i
b. BLN has range RANGE/2^2*i+1 to 2*RANGE/2^2*i.
c. BRN has range 2*RANGE/2^2*i + 1 to 3*RANGE/2^2*i
d. QRN has all values lies between 3*RANGE/2^2*i+1 to RANGE
3. Selection of initial node to traverse for searching
3. Selection of initial node to traverse for searching
4. i=1
   If (0<ITEM< RANGE)

```
{
  If (ITEM < RANGE/2^2*i)
//    search QLN

            {
              If (LOC = = INFO NODE)
                    Return to ROOT,
                    i++,
Else                                    Traverse QLN
SUB NODE;
SEARCHING (ROOT);
                    }
Else    if(ITEM    <2*(RANGE/(2^2*i))    to
(ITEM>(RANGE/(2^2*i))))
           {
         If(LOC== INFO NODE)
              Return to ROOT.
                   i++;
        else
         Traverse BLN SUB NODE;
          SEARCHING (ROOT);
           }
Else    if(ITEM    <3*(RANGE/(2^2*i))    to
(ITEM>2*(RANGE/(2^2*i))))
        {
             If (LOC== INFO NODE)
              Return to ROOT.
               I++;
   Else Traverse BRN SUB NODE Searching
(ROOT);
         }
Else    if(ITEM    <4*(RANGE/(2^2*i))    to
(ITEM>3*(RANGE/(2^2*i))))
         {
             If(LOC==INFO NODE)
              Return to ROOT;
              i++;
             Else
               Traverse QRN SUB NODE
                Searching (ROOT)
          }
  Else
    {
           ITEM NOT FOUND;
    }
```

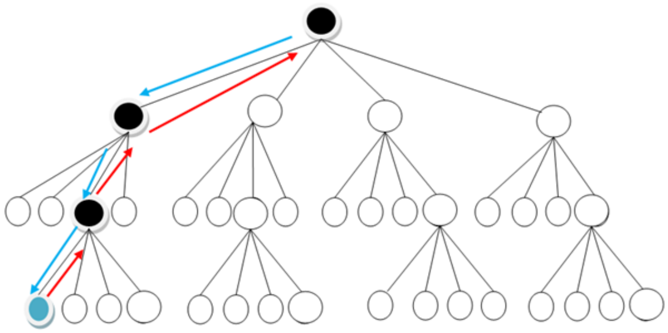


Figure 1: Showing Algorithm Approach Of Searching Node

B. Proposed Algorithm

According to the proposed technique, the nodes are traversed according to the ranges defined. Now when they require node containing the element is searched to find out the new node or for next search pointer backtrack from one previous node to another until root node arrived.

In this technique, the time requires in between one search and next search depends on the number of backtracking done to reach root node.

Suppose 1 millisecond time is requiring for comparison and 1 millisecond time for backtracking from one node to another.

C. Run Time Complexity of Searching Algorithm

The quad tree searching in this algorithm is based on the somewhat divide and conquer approach. In this approach, the nodes are divided into four groups. And the data of specific range is available in specific group. In this tree, the data is already in sorted manner and the ranges will be based on the value of root.

To find the ITEM in the tree, as in the algorithm, the range is checked and then the elements are further searched. In the algorithm, the tree node is divided into quadrants i.e. into four groups of nodes. Thus the recurrence relationship of the algorithm is as

$$T(n)=T(n/2^{2}*i)+\{T(n/2^{2}*i)+c0\}+c1+c2+c3+c4+c5+c6+c7$$

Where, T(n) is the time complexity and c0, c1, c2 , c3, c4 ,c5, c6, c7 is a cost.

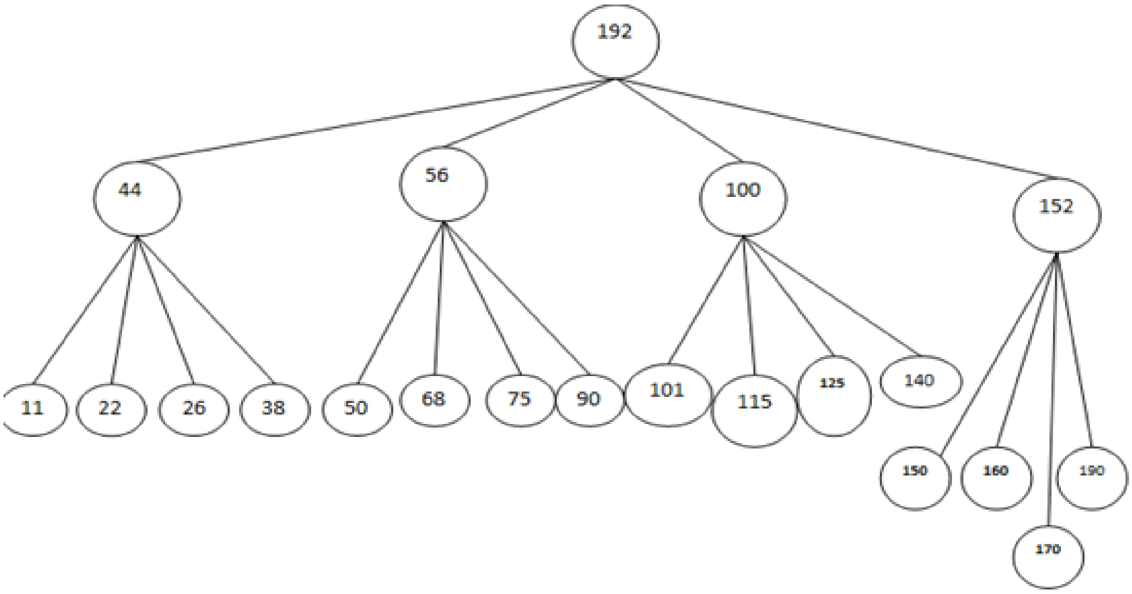
On simplifying the above equation we get $T(n)=T(n/2^2*i)+\{T(n/2^2*i)+c0\}+C$ where say, $C=c1+c2+c3+c4+c5+c6+c7$

On further simplifying the equation we have $T(n)= T(n/2^2*i)+S$, where we can say $S=\{T(n/2^2*i)+c0\}+C$ Now, on simplifying the equation the $T(n)=T(n/2^2*i)+S$ recursively where we assume nth times of ith term i.e. $2*i*n =k$ and we assume $n=2k$. As we find the complexity in upper bound of time space. Thus the complexity of the algorithm in run time is O (log2n)

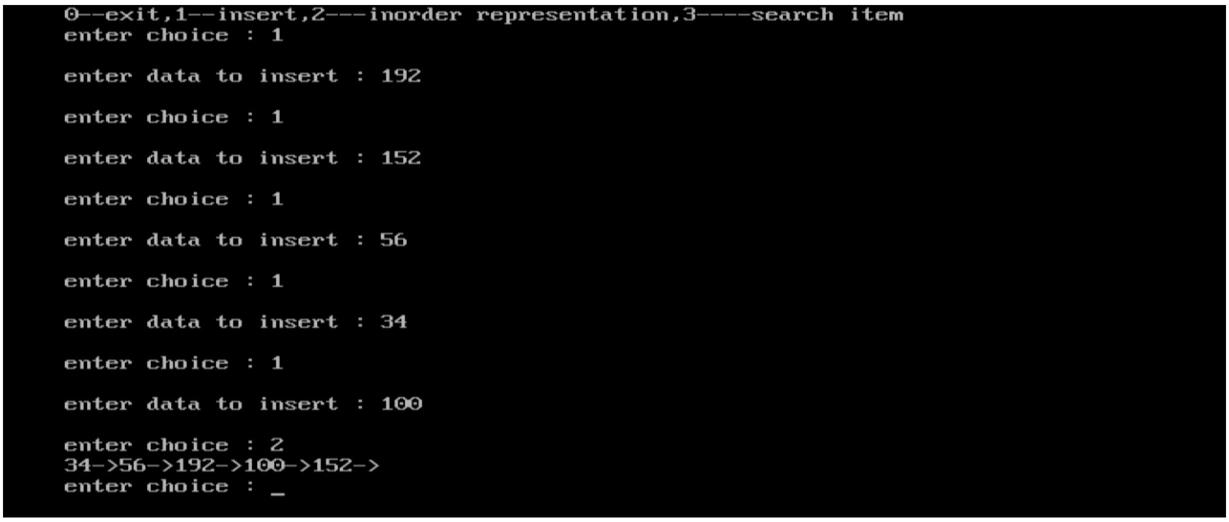
III. RESULTS AND DISCUSSION

Implementation of Algorithm

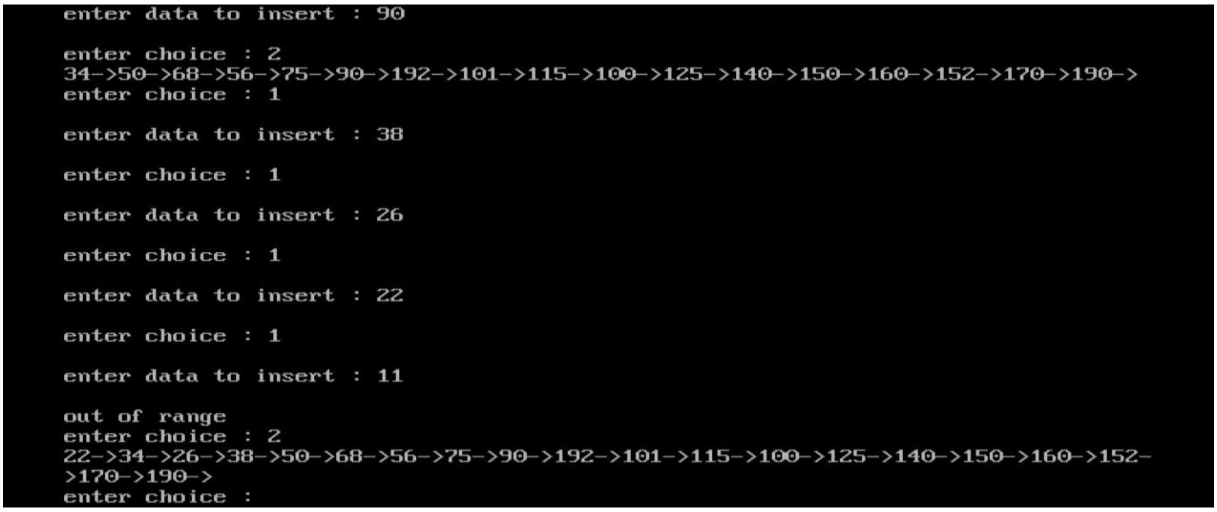
a) Example Quad Tree for Implementation



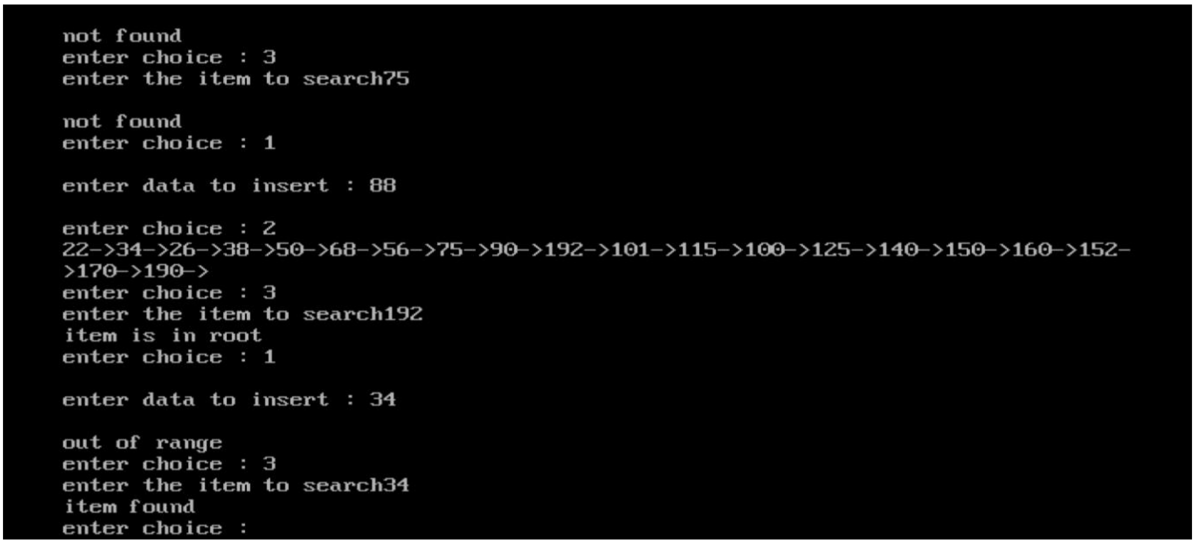
b) Implemented Code Result



```
0--exit,1--insert,2---inorder representation,3----search item
enter choice : 1

enter data to insert : 192

enter choice : 1

enter data to insert : 152

enter choice : 1

enter data to insert : 56

enter choice : 1

enter data to insert : 34

enter choice : 1

enter data to insert : 100

enter choice : 2
34->56->192->100->152->
enter choice : _
```



```
enter data to insert : 90

enter choice : 2
34->50->68->56->75->90->192->101->115->100->125->140->150->160->152->170->190->
enter choice : 1

enter data to insert : 38

enter choice : 1

enter data to insert : 26

enter choice : 1

enter data to insert : 22

enter choice : 1

enter data to insert : 11

out of range
enter choice : 2
22->34->26->38->50->68->56->75->90->192->101->115->100->125->140->150->160->152-
>170->190->
enter choice :
```



```
not found
enter choice : 3
enter the item to search75

not found
enter choice : 1

enter data to insert : 88

enter choice : 2
22->34->26->38->50->68->56->75->90->192->101->115->100->125->140->150->160->152-
>170->190->
enter choice : 3
enter the item to search192
item is in root
enter choice : 1

enter data to insert : 34

out of range
enter choice : 3
enter the item to search34
item found
enter choice :
```

IV. REFERENCES

[1] Data Structures, The McGraw Hill companies, G A V PAI

[2] http://searchoracle.techtarget.com/definition/quad-tree

[3] https://github.com/mbostock/d3/wiki/Quadtree-Geom

[4] http://cboard.cprogramming.com/c-programming/157784-help-me-understand-quadtree-traversal.html

[5] http://ibis.geog.ubc.ca/courses/klink/gis.notes/ncgia/u37.html

[6] www.ncbi.nlm.nih.gov/m/pubmed/21869244/

[7] Artificial Intelligence, Third Edition, Elaine Rich Kevin Knight, Shivashankar B Nair, The McGraw Hill companies

[8] The Introduction to Algorithm, Thomas H. Cormen, Charles E Leiserson, Ronald L. Reivst and Clifford Stein

[9] Data Structures and Algorithms: Annotated Reference with Examples First Edition Copyright Granville Barnett, and Luca Del Tongo 2008.

[10] Hanan Samet computer science Department University of Maryland college park, USA

[11] Sarah F. Frisken and Ronald N. Perry, Mitsubishi Electric Research Laboratories

[12] Kasturi Varadarajan May 2, 2013, journal of applied mathematics.

[13] H.Samet, C A Shaffer, R C Neison, Y. G huang, K . Fujimura and A hosenfeld, recent development in quad tree based geographical information system.

[14] Ivan •Sime•cek, Sparse Matrix Computations using the Quad tree storage format Department of Computer Science and Engineering, Czech Technical University, Prague

[15] P. Barrett ,Universities Space Research Association & Laboratory for High Energy Astrophysics, NASA/Goddard SFC, Greenbelt, MD 20771

[16] H. Samet, “Neighbor finding in images represented by octrees,” Comput. Vision, Graph. Image Process., vol. 46, no. 3, pp. 367–386, 1989.

[17] C.-Y. Huang and K.-L. Chung, “Faster neighbor finding on images represented by bincodes,” Pattern Recognit., vol. 29, no. 9, pp. 1507–1518, 1996.

[18] G. Schrack, “Finding neighbors of equal size in linear quadtrees and octrees in constant time,” CVGIP Image Underst., vol. 55, no. 3, pp. 221–230, 1992.

[19] J. Vörös, “A strategy for repetitive neighbor finding in images represented by quadtrees,” Pattern Recognit. Lett., vol. 18, no. 10, pp. 955– 962, 1997.

[20] R. A. Finkel and J. L. Bentley, “Quad trees a data structure for retrieval on composite keys,” Acta Inform., vol. 4, no. 1, pp. 1–9, 1974.

[21] H. Samet, “Neighbor finding techniques for images represented by quadtrees,” Comput. Graph. Image Process., vol. 18, no. 1, pp. 37–57, 1982.

[22] I. Gargantini, “An Effective Way to Represent Quadtrees,” Commun. ACM, vol. 25, no. 12, pp. 905–910, Dec. 1982.

[23] K. Aizawa, K. Motomura, S. Kimura, R. Kadowaki, and J. Fan, “Constant time neighbor finding in quadtrees: An experimental result,” in Communications, Control and Signal Processing, 2008. ISCCSP 2008. 3rd International Symposium on, 2008, pp. 505–510.

[24] K. Aizawa and S. Tanaka, “A Constant-Time Algorithm for Finding Neighbors in Quadtrees,” Pattern Anal. Mach. Intell. IEEE Trans., vol. 31, no. 7, pp. 1178–1183, Jul. 2009.

[25] A. Aho, J. E. Hopcroft and J. D. Ullman, The Desim and Analysis of Computer Algorithms, Addison-Wesley, Reading, Mass., 1974.

[26] A Klinger and C. R. Dyer, Experiments in picture representation using regular decomposition, Computer. Graphics Image Process. 5, 1976, 68-105.

[27] S. L. Tanimoto and T. Pavlidis, A hierarchical data structure for picture processing, Computer Graphics Image Process. 4, 1975, 104-119.

[28] N. Alexandridis and A. Klinger, Picture decomposition, tree data-structures, and identifying directional symmetries and node combinations, Comput. Graphics Image Process. 8,1978,43-47.

[29] G. M. Hunter and K. Steiglitz, Linear transformation of pictures represented by quadtrees, Computer Graphics Image Process. 5,1976, 68-105.

[30] C. R. Dyer, A. Rosenfeld, and H. Samet, Region representation: Boundary codes from quadtrees, Commun. ACM X3,1980,171-179.

[31] H. Same& Region representation: quadtrees from binary arrays, Comput. Graphics Image Process. 13,1980, 88-93.

[32] H. Samet, An algorithm for converting rasters to quadtrees, IEEE Trans. Pattern Anal. Mach. Intel.3,1981, 93-95.

[33] M. Shneier, Calculations fo geometric properties using quadtrees, Comput. Graphics Image Process16, 3, 1981, 296-302.

[34] H. Samet, The quadtree and related hierarchical data structures, ACM Comput. Surv. 16, 1984, 187-260.

[35] I. Gargantini, An effective way to represent quadtrees, Commun. ACM 25, 1982, 905-910.

[36] M. Tamminen, Comment on quad-and octtrees, Commun. ACM 27, 1984, 248-249.

[37] H. Samet, Data structures for quadtree approximation and compression, Commun. ACM 28, 1985, 973-993.

[38] C. H. Chien and J. K. Aggarwal, A normalized quadtree representation, Comput. Vision Graphics Image Process. X,1984,331-346.

[39] M. Li, W. I. Gorsky, and R. Jain, Normalized quadtrees with respect to translations, in Proceedings, PRIP-81, Dallas, Texas, 1981, pp. 60-62.

[40] T. Matsuyama, L. V. Hao, and M. Nagao, A file organization for geographic information systems based on spatial proximity, Comput. Vision Graphics Image Process. 26,1984, 303-318.

[41] D. M. McKeown and J. L. Denlinger, Map-guided feature extraction from aerial imagery, in Proceedings, Workshop on Computer Vision, New York, April 1984, pp. 205-213.

[42] A. Rosenfeld, H. !&met, C. Shaffer and R. E. Webber, Application of Hierarchical Data Structures for Geographical Information Systems, TR-1197, Computer Science Department, University of Maryland, June, 1982.

[43] A. Rosenfeld, H. &met, C. ShafTer, and R. E. Webber, Application of Hierarchical Data Structures for Geographical Information Systems Phase II, TR-1327, Computer Science Department, University of Maryland, September 1983.

[44] G. M. Hunter, Computer animation survey, Comput. Graphics 2,1977, 225-229.

[45] G. M. Hunter, Ejkient Computation and Data Structures for Graphics, Ph.D. thesis, 1979.

[46] J. R. Woodwark, The explicit quadtree as a structure for computer graphics, Comput. J. 25, No. 2, 1982, 235-238.

[47] R. 0. Duda and P. E. Hart, Pattern Classijication and Scene Analysis, Wiley, New York, 1973. 49. C. M. Eastman, Representations for space planning, Commun. ACM 13,1970, 243-250.

[48] F. T. Leighton and C. E. Leiserson, Wafer-scale integration for systolic arrays, in Proceedings. 23rd Ann. IEEE Sympos. Found. of Comput. Sci., 1982, pp. 297-311.

[49] N. Ahuja, Efficient planar embedding of trees for VLSI layout, private communication, October 1985.

[50] C. R. Dyer, A VLSI pyramid machine for parallel image processing, in Proceedings, PRIP-81, Dallas, Texas, 1981, 381-386.

[51] N. Ahuja, On approaches to polygonal decomposition for hierarchical image representation, Computer Vision Graphics Image Process. 24, 1983, 200-214.

[52] S. B. M. Bell, B. M. Diaz, F. Holroyd, and M. J. Jackson, Spatially referenced methods of processing raster and vector data, Image Vision Computer. 4, 1983, 211-220.

[53] F. C. Holroyd, The geometry of tiling heirarchies, Ars Combin. l&B, 1983, 211-240.

[54] F. Toth, Regtdar Figures, Macmillan, New York, 1964.

[55] A. V. Shubnikov and N. V. Belov, Colored Symmetry, Pergamon, Oxford, 1964.

[56] P. S. Stevens, Handbook of Regular Patterns, MIT Press, Cambridge, Mass., 1980.

[57] B. Grunbaum and G. C. Shepard, Tilings and Pattetns, Freeman, San Francisco, 1981.

[58] A. L. Loeb, Color and Symmetry Krieger, Huntington, N.Y., 1978.

[59] J. Bourgoin, Arabic Geometrical Patterns and Design, Dover, New York, 1973.

[60] P. Maaumder, Networks and Embedding Aspects of Cellular Structures for on-chip Parallel Processing.

[61] MSc. thesis, Department of Computer Science, University of Alberta, October 1984.

[62] S. W. Golomb, Tiig with polyominoes, .I. Combin. Theory. 2,1966,280-296.

[63] A. Rosenfeld, Adjacency in digital pictures, Inform. Control 26, 1974, 24-33.

[64] A. Rosenfeld and A. Kak, Digital Picture Processing, Academic Press, New York, 1982.

[65] B. Hayes, Computer recreations, Sci. Amer. 250, 1984, 12-21.

[66] A. Rosenfeld, Connectivity in digital pictures, .I. Assoc. Comput. Mach. 17, 1970, 146-160.

[67] S. W. Golomb, Poiyominoes, Scribner, New York, 1965.

[68] F. W. Barnes, Algebraic theory of brick packing 1, Discrete Math. 42, 1982, 7-26.

[69] F. W. Barnes, Algebraic theory of brick packing 2, Discrete Math. 42, 1982, 129-144.

[70] H. S. M. Coxeter, Introduction to Geometry, Wiley, New York, 1980.

[71] H. S. M. Coxeter, Regular Polytopes, Dover, New York, 1973.

[72] C. L. Jackins and S. L. Tanimoto, Ott-trees and their use in representing three-dimensional objects, Computer. Graphics Image Process. 14, 1980, 249-270.

[73] Quad Tree: New Approach of Representing and Traversing; International Journal on Recent and Innovation Trends in Computing and Communication; ISSN: 2321-8169; Volume: 3 Issue: 5; ISSN: 2562 – 2565.

[74] Applications of Quad tree: A Review; International journal of engineering research and general science; volume: 3 Issue: 4; ISSN: 2091-2730