On Balanced Geometric Point Set Partitioning Problems

A Thesis Presented

by

Yogesh M Srihari

to

The Graduate School

in Partial Fulfillment of the

Requirements

for the Degree of

Master of Science

In

Computer Science

Stony Brook University

May 2011

Stony Brook University

The Graduate School

Yogesh M Srihari

We, the thesis committee for the above candidate for the

Master of Science degree, hereby recommend

acceptance of this thesis.

Joseph S.B. Mitchell – Thesis Advisor
Professor, Dept. of Applied Mathematics & Statistics

Jie Gao
Assistant Professor, Dept. of Computer Science

This thesis is accepted by the Graduate School

Lawrence Martin
Dean of the Graduate School

Abstract of the Thesis

On Balanced Geometric Point Set Partitioning Problems

by

Yogesh Srihari

Master of Science

in

Computer Science

Stony Brook University

2011

Motivated by the airspace sectorization problem in air traffic management, we consider the following family of geometric partitioning problems: Given a finite set of points within a geometric domain, we want to partition the domain, and thereby the point set, in order that the resulting disjoint polygonal pieces are load-balanced, so that each piece contains a specified number of points and satisfies specified geometric constraints, such as "fatness" (aspect ratio), convexity of the pieces, etc. We consider different variations of the problem having various optimization criteria, including measures of the balance in terms of areas of pieces, number of points per piece, and total length of the edges defining the partition.

Contents

Acknowledgments

I would like to express my deep sense of gratitude to Prof. Joseph S.B Mitchell, for his invaluable help and guidance during the course of this work. I am highly indebted to them for constantly encouraging me by giving their critics on my work. I am grateful to them for having given me the support and confidence.

Yogesh Srihari

May 2011

Stony Brook University

Chapter 1

Introduction

Geometric partitioning and load balancing are well studied problems in various applications, such as resource allocation, political districting (geographic load balancing), minimum cost load balancing in sensor networks, multiple depot vehicle routing, etc.

1.1 Problem Motivation

This thesis has been motivated by studying the airspace sectorization problem that arises in air traffic management. The sectorization problem is to determine a decomposition of a given airspace domain into k sectors, σ_i , $1 \leq i \leq k$, according to various operational constraints and optimization objectives. Decomposition of airspace in 2D means geometric partitions of plane into k sectors or pieces and optimality here means amount of work load in each of those k pieces, which is usually quantified by numerical value indicating the amount of effort required to manage and control traffic in sector σ_i. The objective functions as in [5] are to minimize the maximum workload or to minimize the average workload across sectors, subject to some upper bound on the number of sectors.

The input to the problem is set of trajectories of airplanes, which can be seen as moving set of points from source point to destination point, so at certain instance of time the problem is just geometric partition /load balancing problem with some geometric constraints which requires each sector to be of certain aspect ratio, sectors are preferable to convex, degree of each node of partition are expected to be of three or less.

The challenge here is to load balance the sectors dynamically with changing time in addition also maintain fairly reasonable sector geometry with satisfying constraints .

Observing the complexity of the problem, this thesis considers several related geometric partition /load balancing problems over a static point set.

1.2 Overview

- Chapter 2 discusses the compressive survey over the related geometric static/moving point set partitioning problems.
- Chapter 3 discusses the initial problem definition and with insight in to certain variation of the problem.
- Chapter 4 introduces simple Quad Tree Decomposition Heuristics analyzed for load balancing and fixes empty Square problem created from the previous problem using Pie Cutting Heuristics combined with Quad Tree Decomposition.
- Chapter 5 visualizes few more snapshots Quad Tree and Pie Cutting Heuristics.
- Chapter 6 all experiment setup hardware, configuration, framework, platform used.

Chapter 2

Related Work

We are motivated by studying an airspace sectorization problem that arises in air traffic management. The sectorization problem is a geometric partition problem, with various operational constraints and objectives that come from the air traffic application area. This brief survey focuses on several related geometric static point set partition problems with results of the variations of the same problem and identifies some of the open problems among the different variations.

2.1 Introduction

Geometric load balancing/partition problems are well studied in resource allocation, map segmentation, airspace design and air traffic controller workload balancing etc.

In a usual geometric partition there is a region or a polygon S, which we must partition into a collection of n-smaller sub-regions Si, while optimizing certain specified criteria and satisfying certain geometric constraints. Geometric constraint can be a simple disjoint set of partitions, convex regions, having specified degree of the partition node etc. The optimizing criteria might be the area of sub-regions, shape of the region (fatness), number of points in the region, perimeter of the partition etc.

The results in the survey are reviewed on the basis of the following categories:

1) Variation / Definition of the problem :
2) Approach used :
3) Application / Result :
4) Time Complexity :

2.2 Static Point Set Partitioning Problems

This major section reviews basically the geometric point set partitioning problem for static point set as input while optimizing certain criteria such as area, points, fatness etc of the partitions and without violating certain geometric constraints.

Atsushi Kaneko et al [1] proposed an algorithm for finding a generalized area and boundary of the region partition of a given region. Following is the problem definition.

Problem 1: For a region X in the plane, we denote by area(X) the area of X and l(δ(X)) the length of the boundary of X . Let S be a convex set in the plane , n>=2 an integer and $\alpha_{1,}\alpha_2\ldots\alpha_n$ *positive real numbers such that* $\alpha_1+\alpha_2+\ldots\ldots\ldots\alpha_n = 1$ *, S can be partitioned into n disjoint subsets* $T_{1,}T_{2\,..}$ *so that each of* T_i *satisfies the following two conditions i) area* $(T_i) = \alpha_i * area(S)$ *ii)* $l(T_i \cap \delta(S)) = \alpha_i * l(\delta(S))$ *iii) Each partition has to be continuous curve .*

Atsushi Kaneko et al [1] deal with only the proof of existence of such a partition i.e.Region X can be partitioned in to such convex subsets with area and boundary of each partition satisfying the criteria parameter α_i, their paper does not deal with algorithm to find such a partition.

The paper starts of understanding how to achieve a perfect partition in to three convex subsets by three rays emanating from a point in a convex set in the plane and proves the existence of it and later proves generalized version of a perfect partition in to n subsets proceeding on the same lines.

Some of the Algorithmic Questions arise out of the papers

Algorithm Question 1: Find an algorithm to compute the perfect partitions in to n convex subsets on same constraints defined in the problem.

Algorithm Question 2: How efficiently or what would be time complexity of such an algorithm since we already know its existence by the results of the paper.

John Gunnar Carlsson et al [2] solves special version of the Problem 1 for $\alpha_i = 1 / n$ i.e , area of all the partition are equal to each other and there is an additional constraint of having a equal points in each of the partitions.

Problem 2: Collection of points X with |X| =g.n contained in a convex polygon C with m vertices. Partition C into g disjoint convex sub-regions, each having area Area(C)/g and containing n points.

John Gunnar Carlsson et al [2] results have been motivated by in the context of the multiple depot vehicles routing, and it has interesting relation to "makespan" problem. (Job Scheduling problems).

This paper actually solves an special case of *Problem1* with additional constraint of equitable point partition .John Gunnar Carlsson et al [2] propose an algorithm to find the equitable partition of the given convex polygon C and point set .They also provide worst case time complexity $O(m^2n^2g^5)$ where m is number of vertices of the polygon, n is the number of points in each partitions and g being the number of partitions.

Paper starts of dealing with existence of equitable 2-3 partitions through a topological lemma which guarantees the existence of such partitions and there by the algorithm works by constructing such equitable 2-3 partitions with equal point and area cells.

Good thing about the results of John Gunnar Carlsson et al [2] is that they provide algorithm for the problem unlike Atsushi et al [1] where existence of proof is major debate, and also [2] provides worst case upper bound for the algorithm which seems to be very good for practical applications.

[2]'s main application is the dynamic resource allocation problems [4] where there is need to allocate resources to the client in a load balanced manner, in their case it is allocation of vehicles to customers that it should service and also minimize the travel time which should be roughly equal in optimal solution.

S.Bespamyatnikh et al [3] propose an algorithm on slightly different variation of the problem which does not concern with the measure area of the partition but it is replaced with measure of different kind of point set.

In John Gunnar Carlsson et al [2]'s paper categorizes most of these variation of partition problems results in to balancing two important measures i.e $\beta 1$ = Point set or Area or Perimeter and β_2 = Point Set.

Problem 3 : Given gn red points and gm blue points in the plane in general position , there exists an equitable subdivisions of the plane into g disjoint convex polygons ,each of which contains n red points and m blue points .

In S.Bespamyatnikh et al [3] propose an algorithm for constructing an equitable subdivision of the plane into g disjoint convex polygons, each of which contains n red points and m blue points. If points sets are in powers of 2, then ham sandwich cut can be applied in divide and conquer fashion.

Approach is similar to other two criteria load balancing partition problems, proves the existence of 3-cutting which is again with help of topological lemma identifies existence of equitable 2-3 partitions , their results are not limited to discrete version of the problems , all their existence theorems of 3-cutting of discrete version are easily extended continuous 3-cutting subdivision.

In S.Bespamyatnikh et al [3] approach of red and blue point partitioning is very similar to approach of Carlsson et al[2] of area/point partitioning with respect proving the existence of 3-cutting and then proceeding to propose an algorithm to find the equitable partitions.

2.3 Moving Point Set Partitioning Problems

The variation of the problem is that partitioning data is not just a set of points but instead consist of trajectories of moving points, rather than static point set. The motivation for such moving point set partitioning problem is that of airspace sectorization problem.

Problem: The goal of airspace sectorization problem is to determine a decomposition of a given airspace domain D into a set of k-sectors (sector corresponds to a partition or subdivision in previous defined problem) in an optimal manner. Optimality is defined in terms of numerical value of workload which amount of effort needed to manage the airplanes in particular sector.

Amitabh Basu et al [5] propose a combination of exact solution for 1D problem, BSP and pie cutting heuristics for 2D airspace sectorization problem ,here 1D problem is actually 2D problem with space and time as parameters thereby 2D problem can thought of 3D (x,y,t) with additional time parameter .

Approach in [5] solve the 1D version of the problem exactly by reducing the trajectories as a line segments lying on (x,t) plane which is then solved greedily using sweeping rectangle and intersecting with set of line segments which gives the number of airplanes in x_i x_{i+1} i.e. in a particular sector , so optimizing parameter can be reducing the maximum workload which can be solved adding points on x_i (sectors) incrementally (greedily) minimizing the maximum or average workload .

In airspace sectorization, the two load balancing measures of generic partitioning problems defined above would be changed here to β1 = Aspect Ratio (Fatness of the piece), β2 = Number of points at time instant t.

Having equal area partition is not big requirement in airspace partitioning problem, other static point set partition results would not be appropriate because results of Carlsson [3] shows partitions which are really skinny and can have equal area which certainly will not be good aspect ratio pieces.

2.4 Open Questions

Questions 1: Is there exists such a partition where equitable point/aspect ratio partition exists for just the static point set partitions.

Questions 2: Is there exists such partition of all sectors such that the aspect ratio of all the sectors are approximately equal to each other within a given approximation bound.

Questions 3: Is there a polynomial time algorithm to find such an equitable partition?

Chapter 3

3.1 Problem Definition

We introduce various problem definitions in this chapter. We provide heuristics in the subsequent chapters for good subset of the problems introduced in this chapter.

Definition 1:

Input: I = {Set of trajectories with starting and final destination in 2D} , Each trajectory defined with respect to two parameters ; points (xi,yi,ti) and (xj,yj,tj) with initial and final (location ,time) .

Constraint: Bounded by k sectors, each of k sectors should have certain aspect ratio and preferably be convex pieces.

Output: At time instant t_i , set of k sectors {σ1, σ_2, σ_3………σ_k } , For each time instant the output consist of k load balancing sectors , hence dynamically the output changes with respect to the time .

Objective Function: It can be to minimize the maximum workload across all the k sectors or to minimize the average workload across all the k sectors.

Here we are considering moving set of points so output would be set of k-sectors changing dynamically with time, so there are many parameters which make the problem way too complex. Certainly one would ask questions such as:

1) How would a set of sectors at time instant t_i differ from set of sectors at time instant t_j?
2) Will aspect ratio of all the sectors be controlled well at all any time instant?
3) Would the sectors be not convex at some time instant t?
4) Are we able to maintain optimal partition at any given time instant?

Since there are too many parameters to keep in state in the above definition of the problem, in the next section we reduce moving point set as input to static point set which can seen as a snap shot of airspace at some time instant t.

3.2 Variations of the Problem:

Problem 2:

Input: I = {P1, P2…..Pn} in R^2

Output: O = Set of k sectors $\{\sigma_1, \sigma_2 \ldots \sigma_n\}$

Geometric Constraint: Sectors should have good aspect ratio. This can be quantified by specifying a certain numerical value for the aspect ratio and the aspect ratio of each piece should be within some defined lower and upper bound.

Description: Let m = n / k where n – number of points and k – number of sectors, each sector has m points. If there is choice of n = k*m points for the input:

1) What is a criterion for distribution of point set in order to have a good aspect ratio partition?

2) If good aspect ratio partition cannot be maintained for all distribution of the point set , is it possible to have an aspect ratio lie within the bound and allowing points in each sector to be within certain lower and upper bound while removing restriction of having exactly m points in a sector .

Problem 3: Equitable Aspect Ratio Partition:

Input:
Set of points in 2D, Let N be the number of points.

Output:
Set of K Convex Pieces, Each Convex Piece having M points each.

Constraints:
Each of the Convex Pieces should have equal Aspect Ratios.

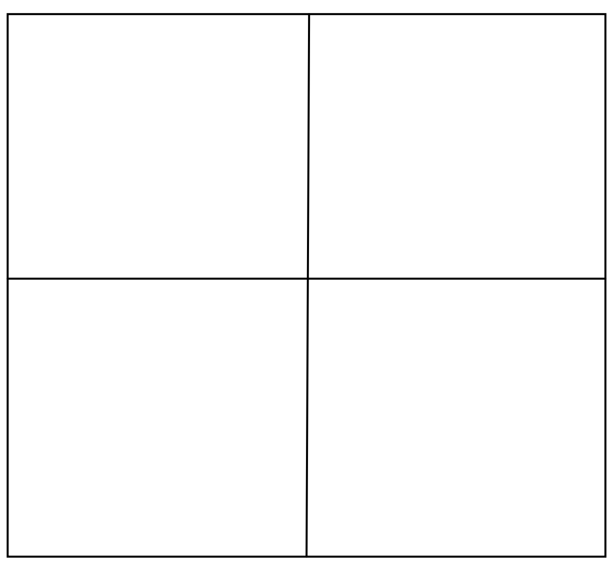
Existence of Equal Aspect Ratio Pieces with Squares:

Say we have a square Bounding Box over the point set, then if we can divide the bounding box in to K ≥ 6 squares, then we can have equitable aspect ratio square partition, n – Number of squares that a given square can be decomposed to.

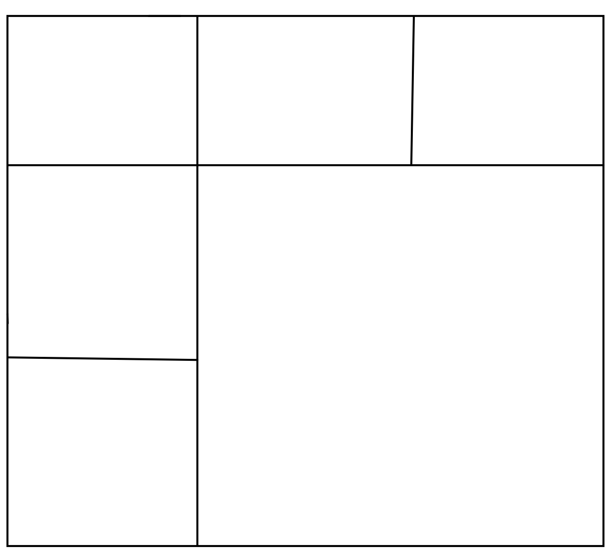
We know that a square can decomposed into any number of smaller squares for n≥6, and for n=4 (Quad tree), it is easy see that any square bounding box can be decomposed in to smaller squares, each time we have quad tree like decomposition of the square we increase the number of squares by three, now we can have squares 4, 7, 10, 13....

If we start with 6 squares (5 small squares + 1 big square) decomposition, then again each square can divided into four small squares so it again increases by 3 and We can have squares decomposition of 6, 9, 12...

Similarly for we can start with 8 squares decomposition (7 small squares + 1 Big Square) then again with same idea we can have squares of 8, 11, 14, 17.All squares have equal aspect ratio.



Basic Quad Tree, we can start with above decomposition if we want partition of 4, 7, 10...



If we can start with n = 6 decomposition, then get partition of squares 6, 9, 12, 15. This shows that equal aspect ratio partition exists at least for the convex pieces restricted to only to squares,

Questions:

1) Is there exists such an equal aspect ratio partition covering given distribution of the point set?

2) What is the restriction that, we can place on the distribution of the point set so that we can have equitable aspect ratio?

3) Can we have some bounded aspect ratio partition of given point set?

Chapter 4

Load Balancing Quad Tree Decomposition Heuristics

We consider the following version of the problem for applying the heuristics defined in the subsequent sections.

Problem 4:

Input: Set of random points in {P1, P_2……. P_n}

Output: Partition of set of points in to K point partitions , so there are N/K such Partitions, if N is divisible by K.

Constraints and Objective Function: Expected to have good shaped pieces, so the objective function can be defined t o maximize the ratio MinArea (σ) / MaxArea (σ) for all sectors i.e. minimize the difference between maximum area piece and minimum area piece.

4.1 Description of Quad Tree Decomposition:

For the above defined problem, square bounding box is created for set of input points. K is bound number of points in the sectors.

Quad decomposition works by dividing the current square in to 4 equal smaller squares based on some specified criteria. The criteria in this case are K point Quad decomposition i.e. if the squares contain more than k points, divide in to four more squares recursively applying the Quad decomposition.

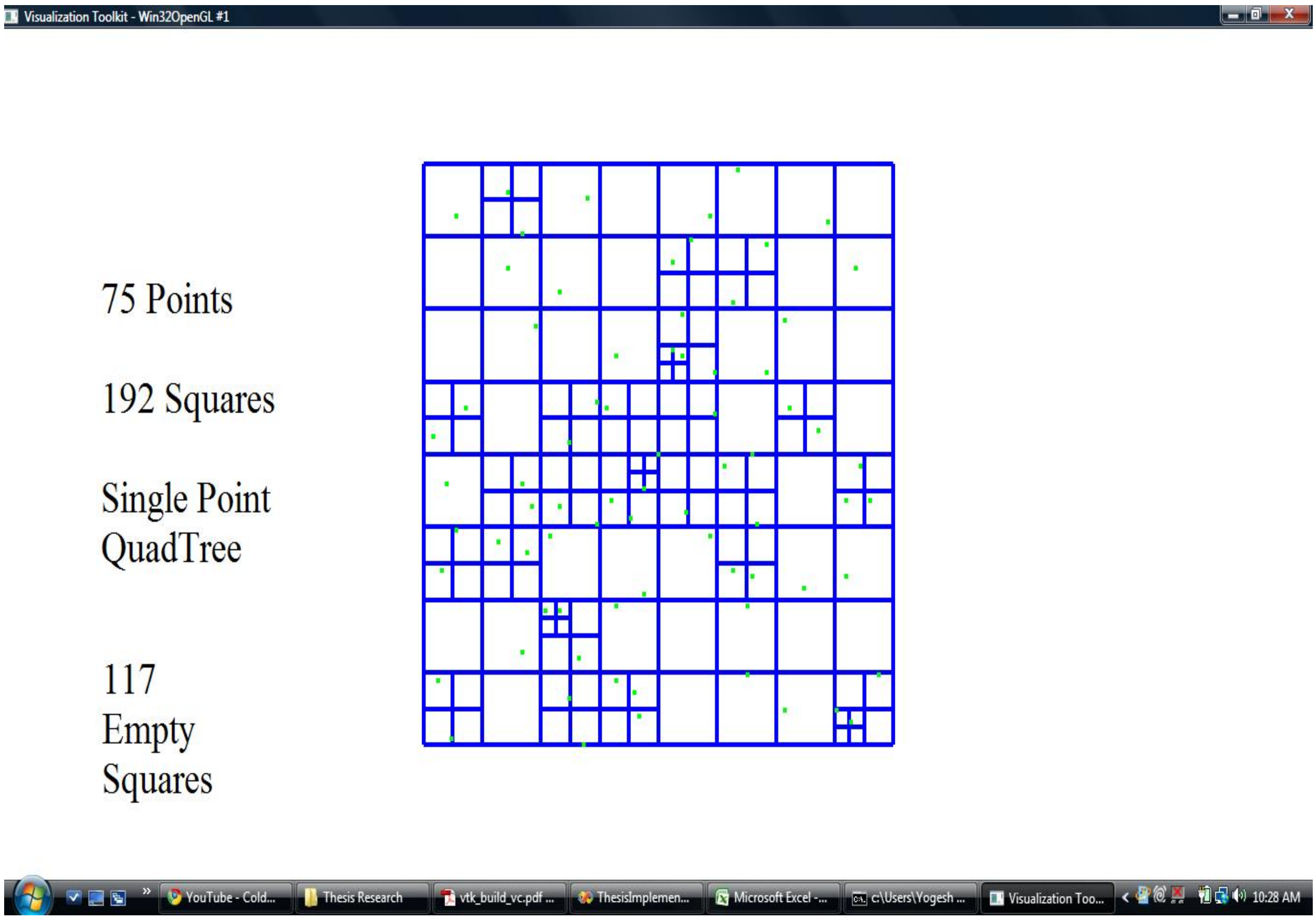
The pieces here are always good shaped because aspect ratios of squares are always equal.

4.2 Results of Quad Decomposition:

K-point load balancing Quad decomposition:

Suppose K = 1, 1-point Quad decomposition, where each piece has not more than single point in the final partition.

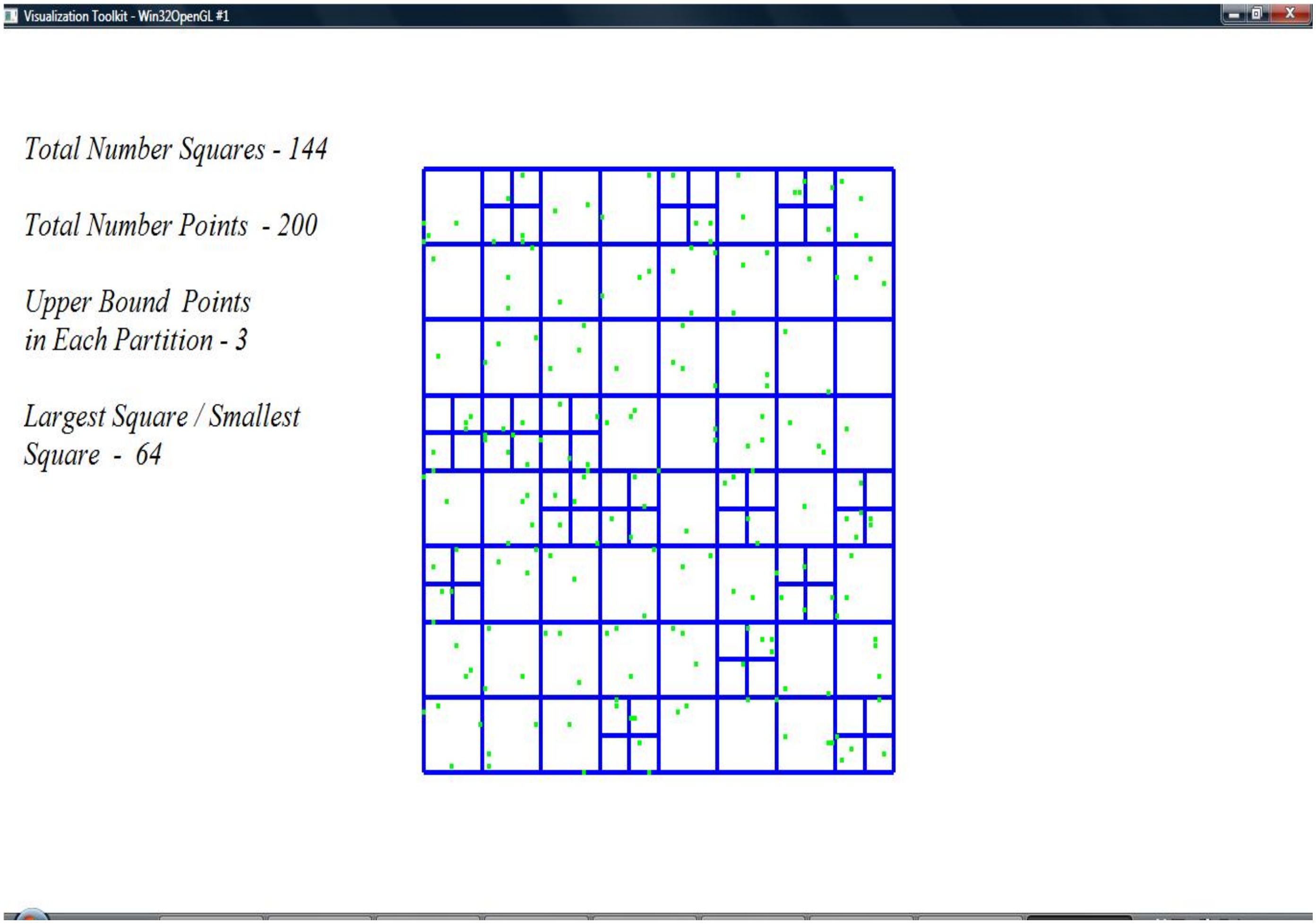
In figure below there are N = 75 points, K =1 , there should be basically N single point partitions.



One Point Quad Subdivision *Figure 4.1*

This sort of subdivisions results in lot of empty squares because of nature of Quad like decomposition. For 75 points in the input set, we have 117 empty Squares which are undesirable output partitions.

For K = 3, N = 200 Points Quad decomposition is shown in figure 2 below, as the point density increases in the input set ; the simple Quad decomposition performs better , number of empty squares reduces drastically .



3 Point Quad Subdivisions *Figure 4.2*

The above figure shows that none of the squares has more than 3 points and also there are not many empty squares in the partition.

Ratio of Largest Square / Smallest Square is 64; it shows in our experimental results that this ratio moves closer to 1 as K, the number of point in the sector increases.

4.3 Empty Squares Problem solved with Pie Cutting Heuristics within Quad Decomposition.

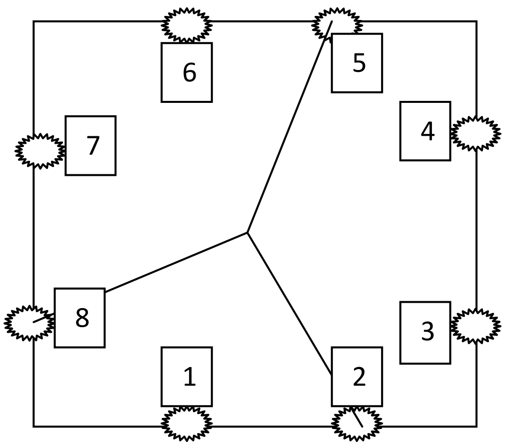
In previous section, Quad decomposition algorithm continues recursively until the current square has lesser than K points, the number of points specified in each partition.

In this section pie cutting is applied to a square at certain level of Quad decomposition.

Description: In the recursion of Quad subdivision, when the current square has less than 4*K points; instead of the continuing into further quad subdivision, pie cutting is applied at this stage of recursion.

Worst balanced sector or piece can have 4*K-1 points, empty sectors are almost completely avoided in this combination of Quad and Pie cutting heuristics.

Pie Cutting a Square:



On each side of the square, finite number of points is choosen; this is done for all the square sides. Now we have finite set of points on the sides of the square.

For Example in the figure 4.3, we define 8 points on the 4 sides of the square. Now we brute force all possible combination of a degree three pie cut , choose one among them with best load balancing partition .

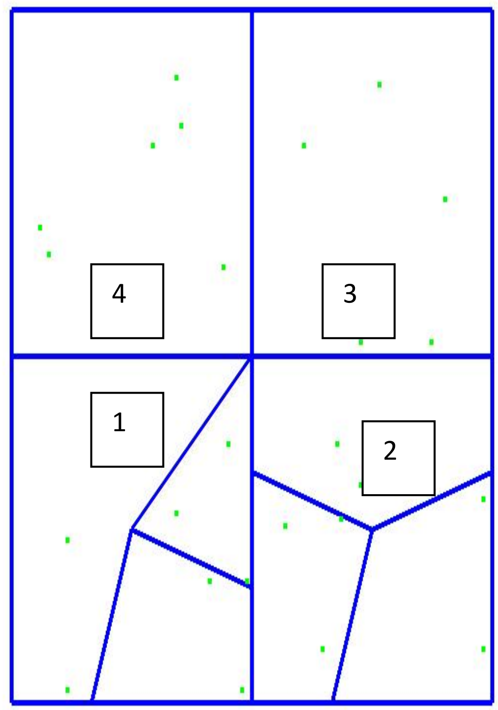
If square has less than 4*K points and more than K points, than pie cutting is applied. Worst possible pie cut gives an upper bound of 4*K-1 points in sectors .This worst case usually does not occur on a random point set.

Pie cut extends its three rays from the midpoint of the square to the defined set of points on the sides of the square. The angle of three rays can be discretely controlled because rays are supposed to connect to only set of points chosen on the sides of the squares.

Aspect ratio can also be discretely controlled by using far subset of finite chosen point set on the sides of the square. For above example we could restricted only every alternate points our subset of selection for pie cutting would be { 1 ,3 ,5,7} , {1, 4,6,8} , {1,5,7} etc . Every other point is ignored. Algorithm selects the best balancing pie cut partition out of the aspect ratio restricted finite chosen point set on the square side.

If farther points on the square sides selected for pie cut, bigger the angle created at the midpoint of the square, hence aspect ratio can be controlled by choosing larger angle points on the sides of the square.

Results of Quad and Pie Cutting Heuristics:



In above Figure,   N = 25 points,   K = 3, N/K = 8 sectors. A square with pie cut i.e. in square 1 and square 2 has less than 4*K points, pie cut optimally balances the sectors with 3 points in each of them. Squares with less than some constant (1.5 in above figure) multiple of K will never proceed to pie cut or Quad subdivisions. Hence this sort of squares will have worst balance of constant multiple of K points.

It can be broadly classified into 3 categories.

1) Squares which undergo pie cut will have worst maximum sector imbalance of 4*K -1 points and minimum imbalance of zero points.

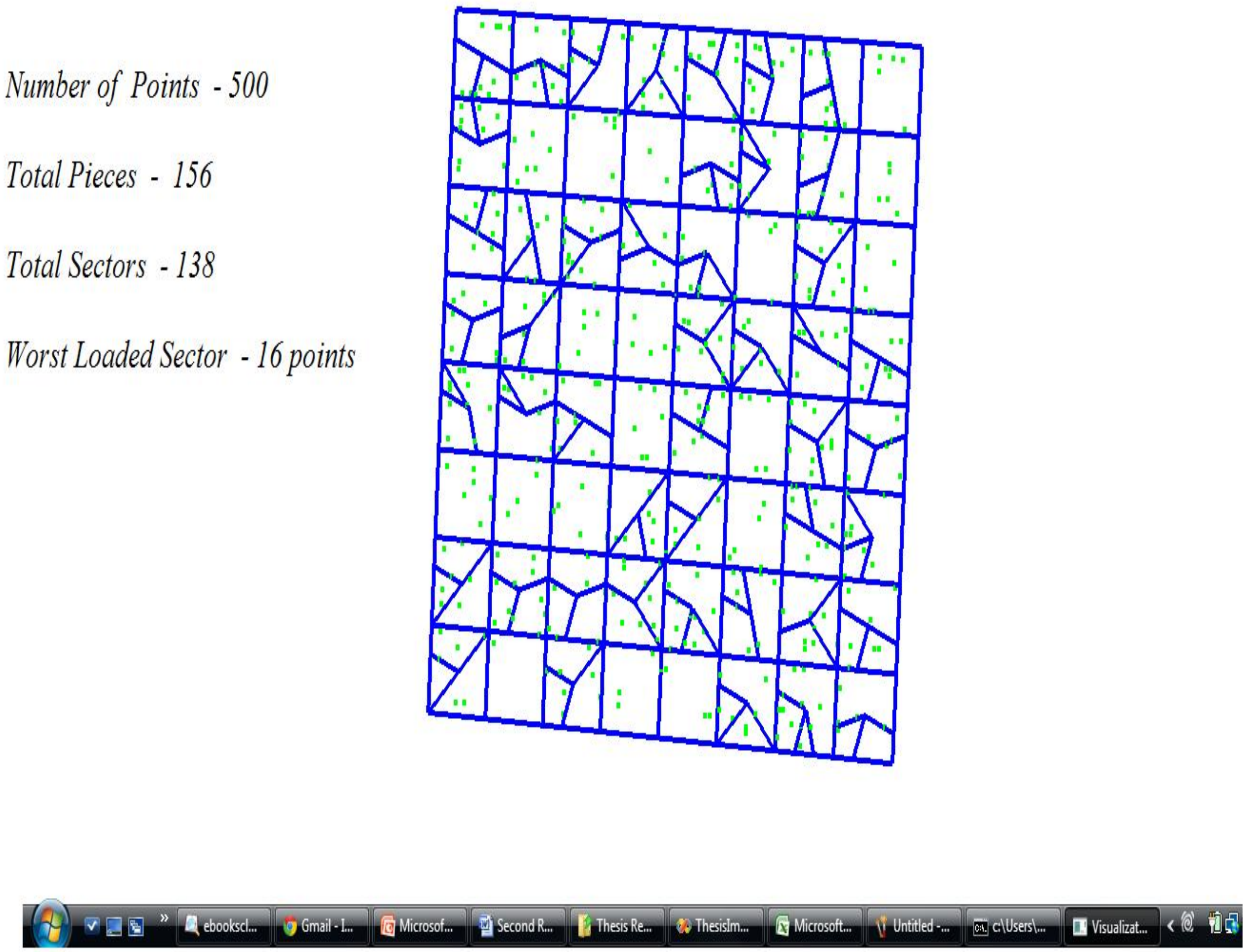
2) Squares which undergo further Quad Subdivisions has more than 4*K points.

3) Squares which have less than constant multiple of K points will neither undergo pie cut or Quad subdivisions and has worst maximum sector imbalance of αK where $\alpha < 4$.

Chapter 5

More Visualization Snapshots

Snapshot 1:



Partition Parameters:

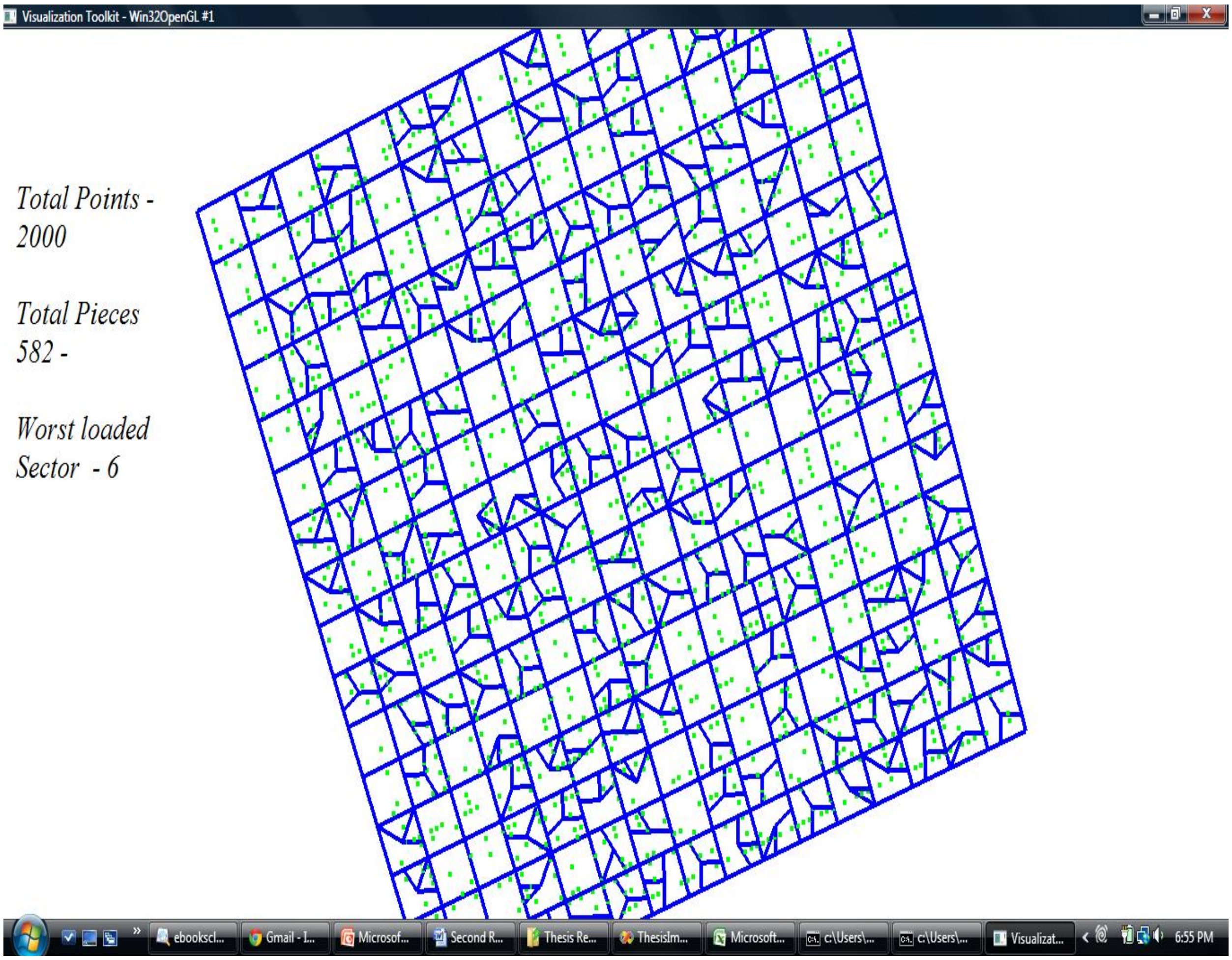
N = 500 Points.

K = 4 Points Sectors

Worst Imbalanced Piece =16 Points, Which is exactly 4*K points.

There are about 156 sectors in this partition.

Snapshot 2:



Partition Parameters:

N = 2000

K = 4

Total number of sectors - 582

From above figure, we can see the worst loaded sector is 6. This shows that as the point density increases, the worst loaded sector is minimized i.e. the sectors tends be more balanced.

Chapter 6

Experimental Setup:

Implementation of the system is with following system setup:

Programming Language: C++

IDE Frame Work: Microsoft Visual Studio 2008 Version 9.0

Visualization Framework: Visualization Toolkit 5.6

On Machines: Intel Core 2 Duo P8600, 2.4 GHz, 4GB RAM

Conclusion

In this thesis we have studied, defined, surveyed, compared different set of geometric load balance / partition problem. We have defined many variation of the problem having several varying optimization criteria and varying subset of geometric constraints such as equal area, aspect ratio, convexity etc. Finally we implemented simple Quad Tree / Pie Cutting decomposition heuristics for point load balancing / aspect ratio. It is easy to see that we could have practical random point set well balanced through this simple heuristics. Aspect ratio of the partition can be discretely controlled well with some lower bound.

References:

[1] Atsushi Kaneko and M.Kano. Perfect n-Partitions of Convex Sets in the Plane. Proceedings of the Tenth Canadian Conference on Computational Geometry. (1998) 114-115.

[2] Sergei Bespamyatnikh, David Kirkpatrick, Jack Snoeyink . Generalizing Ham Sandwich Cuts to Equitable Subdivisions.

[3] John Gunnar Carlsson, YinYu Ye. On Equitable Partitioning a Convex Polygon and a Point Set. *ACM Transactions on Algorithms*, August 2009.

[4] John Gunnar Carlsson, YinYu Ye, Sharon Arroyo, Roman Freshnedo, Raju Mattikalli. A geometric framework for resource allocation problems. Proceedings of the 2009 NSF CMMI Research and Innovation Conference.

[5] Amitabh Basu, Joseph S.B. Mitchell and, Girishkumar Sabhnani, Geometric Algorithms for Optimal Airspace Design and Air Traffic Controller Workload Balancing. Proc. of the Workshop on Algorithm Engineering and Experiments (ALENEX'08), 75-89, San Francisco, California, USA, January 19, 2008.

[6]Joseph S.B Mitchell, Girishkumar Sabhnani, Jimmy Krozel, Robert Hoffman, and Arash Yousefi, Dynamic Airspace Configuration Management Based on Computational Geometry Techniques. Proc. of AIAA Guidance, Navigation, and Control Conference and Exhibit, Honolulu, Hawaii, August 18-20, 2008.

[7]P. Carmi and M. J. Katz. Minimum-cost load-balancing partitions. In Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG’05), pages 63–65, 2005.

[8] S. Muthukrishnan, V. Poosala, and T. Suel. On rectangular partitionings in two dimensions: Algorithms, complexity, and applications. Lecture Notes in Computer Science, 1540:236–256, 1999.

[9] I. Wyndemere. Dynamic resectorization: Accommodating increased flight flexibility. Technical report, http://www.wynde.com/papers/atca97-2.pdf, Boulder, CO, 1997.

[10]Sariel Har-Peled. Geometric Approximation Algorithms. 1991 Mathematics Subject Classification. Primary 68U05, 68W25; Secondary 68P05, 52Cxx.

[11] John Gunnar Carlsson, Benjamin Armbruster, YinYu Ye. Finding equitable convex partitions of points in a polygon efficiently. *ACM Transactions on Algorithms*.