INTERNSHIP REPORT

Studying and implementing 3D geometric routing algorithm and spanners

UNIVERSITY BORDEAUX 1

BY

LY KIM QUYEN

SUPERVISED BY

NICOLAS BONICHON, CYRIL GAVOILLE

FROM

15th JUNE, 2009

TO

11th NOVEMBER, 2009

PREFACE

The purpose of this report is to explain what I had done and learned during my internship at the France National Institute for Research in Computer Science and Control (INRIA). The report focuses primarily on what I had done in the subject “Studying and implementing 3D geometric routing algorithm and spanners”, handled and succeeded assignments, working environment, and working experience during my internship.

ACKNOWLEDMENTS

I would like to thank my supervisor **Prof. Nicolas Bonichon** for his support the passed five months. He gave me the freedom to discover and helped me to choose the direction of my research and to my evolvement. Also, I would like to thank **Prof. Cyril Gavoille,** who gave me this opportunity to have such marvelous internship. Specially, I would like to thank *Rafaële Bacqué-Sassot, Chau Thu Tran, Lam Yihua, Nguyen Minh Hai* and *Nguyen Vu Ngoc Tung* for their helpful comments.

TABLE OF CONTENTS

TABLE OF FIGURES

ABSTRACT

In this report we present about the Geometric spanner in 3D. t – Spanners are Euclidean graphs in which distances between vertices in G are at most t times the Euclidean distances between them. We focus on an important family of geometric spanners: θ-graphs. The basic idea of the θ-graph is to cut the space around each node into sectors (cones) of equal angle and to connect each node to the nearest neighbor in each of its sectors. We modified the θ-graph to ½ θ-graph; the ½ θ-graph is to choose the half sectors of θ-graph, each sector chosen is know as a positive sector.

And then, we want to know if the generalization of ½ θ-graph is a spanner or not in 3D. We proven that the ½ θ-graph with eight cones is not a spanner. Moreover, we continue to study it with the θ-graph. We prove that the θ-Walk algorithm in the θ-graph with eight cones has a stretch factor greater than *t,* for any *t*. The property of θ-graph with Meeting algorithm is a *t*-spanner then $t \geq 7.67$ stretch factors. The other property of the θ-graph is a *t*-spanner then $t \geq 3$ stretch factors.

1. Introduction

As an intern with INIRA, I was working in the Geometric spanner with spanner based on the θ-graph in 3D under the supervision of Prof. Nicolas Bonichon; The θ-graph, which insist on adding an edge in each of k different directions for each of the n input points. Thus, to find a short path from one vertex in the graph to another, that is, pick an edge in the general direction of the destination.

I began my internship program on 15th June, 2009 with INRIA. The internship was to last for five (5) months and as result my internship came to an end on 11th November, 2009.

In somewhat more concrete terms, the θ-graph start with some starting points such as: six cones, eight cones in the plane and in 3D, and then modified the θ-graph to the ½ θ-graph with these starting points, use routing algorithm to find properties for this graph. It is while I was with eight cones in 3D and with compass routing. I will present the major areas identified as research about the θ-graph in 2D and 3D, write the report for the ½ θ-graph with six cones in the plane, ½ θ-graph with eight cones in 3D, and implemented 2 compass routing algorithm for θ-graph with eight cones in 3D.

About the INRIA, it is the French national institute for research in computer science and control, operating under the dual authority of the Ministry of Research and the Ministry of Industry, is dedicated to fundamental and applied research in information and communication science and technology (ICST). The Institute also plays a major role in technology transfer by fostering training through research, diffusion of scientific and technical information, development, as well as providing expert advice and participating in international programs.

It has eight research centers in:

- INRIA Bordeaux – Sud – Ouest
- INRIA Grenoble – Rhône – Alpes
- INRIA Lille – Nord Europe
- INRIA Nancy – Grand – Est
- INRIA Paris – Rocquencourt
- INRIA Rennes – Bretagne Atlantique
- INRIA Saclay – Ile – de – France
- INRIA Sophia Antipolis – Méditerranée

The organization chart:

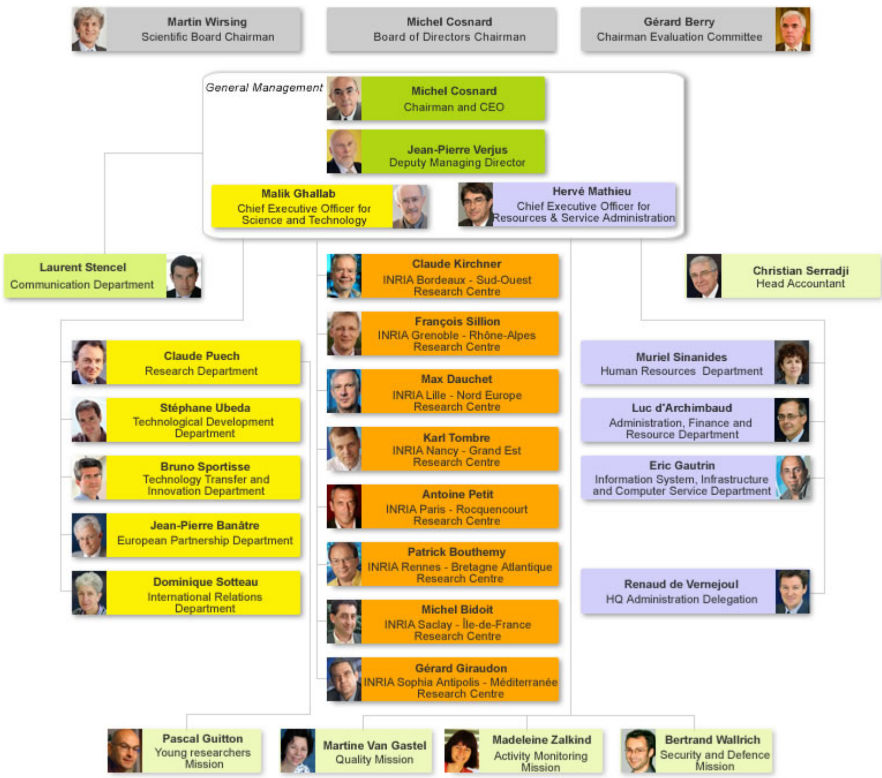

Figure 1: Organization chart of INRIA

The INRIA's major goal for 2008-2012 is to achieve scientific and technological breakthroughs in seven priority domains:

- Modeling, simulation and optimization of complex dynamic systems
- Programming: security and reliability of computing systems
- Communication, information, and ubiquitous computing
- Interaction with real and virtual worlds
- Computational engineering
- Computational sciences
- Computational medicine

Work experiences:

Under supervision of Prof. Nicolas Bonichon, I was able to understand the assigned problems and to what extend I was capable to fulfill it. The following were some of the motivations that made my work better and enjoyable:

- Working on projects and publications that I believed would eventually provide a clear understanding on the state of the art geometric spanner. This gave me the morale to work even harder in order to achieve more challenging objectives.
- Attending the seminar and workshop EuroComp 2009 about Combinatorics, Graph Theory and Applications was excellent!
- Meeting with professionals, experiencing the way they trouble shoot and solve problems.

Successes: There were many successes. Personally the followings are what I succeeded on:

- First, to me it was a success having been given a chance to handle work on various informative publications that I believe will go a long way in geometric spanner.

- Through my work in here, my knowledge about geometric spanner is improved such as how to use the algorithm to look for properties for the graph, how to prove a mathematic for geometric spanner.
- I was not familiar with geometric spanner, mathematic proving and java3d before but now I can confidently understand it and program it.
- Besides, I improved the skill of writing the report.
- I can saw my weakness, and I learnt from it. It helps me improve myself about working and studying for the future.

Weakness: In this period I recognized my deficiency.

- I was not a familiar of mathematic so it made me hard to give Theorems, lemmas, and proving it.
- I was hardly to communicate with my supervisor; it made my work slowly and very hard to understand what the purpose of my work.
- Internet has a lot of information, sometime I lost myself in this.

2. Problem definition

Notation and Definition: Let S be a finite set of points in the plane and let G be a graph with vertex set S, in which edge (u, v) has a weight equal to the Euclidean distance $|uv|$ between u and v. For a real number $t \geq 1$, we say that G is a t-spanner for S, if for any two point's u and v of S, there exists a path in G between u and v whose Euclidean length is at most $t|uv|$. The smallest such t is called the stretch factor of G.

Let $|uv|$ to denote the Euclidean distance between u and v; then we use the notation $|uv|_G$ to denote the Euclidean length of a shortest path between u and v in a geometric network G, we have $|uv|_G \leq t.|uv|$. To a comprehensive overview of geometric spanners, see the book by Narasimhan and Smid [20]. The geometric spanner we use is the θ-graphs.

The θ-graph: Let S be a set of points, for each $u \in S$, among all "*nearly – parallel*" edges incident on u in the complete graph, the θ-graph retains the "*shortest*" one. These graphs are t-spanner for an appropriate value of t. [20]

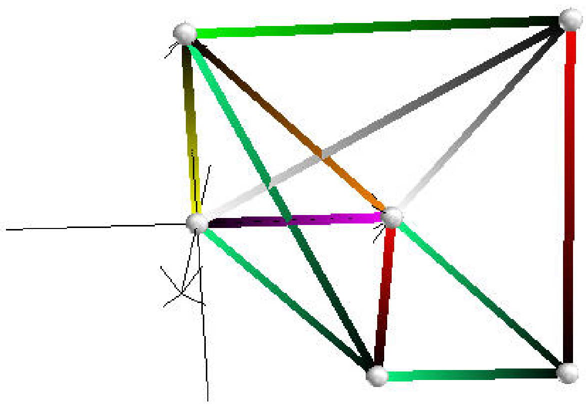

Figure 2: The graph of $\theta(S,k)$ in 3D

In the geometric spanner, let *S* be a set of points in the plane and let *k* be an integer, we want to know the ½ $\theta(S,k)$ [12] in the plane what are properties for this graph and properties of the $\theta(S,k)$ in 3D. Which ½ $\theta(S,k)$ in the plane we give one of starting points with six cones and we show that it works in the plane. After that we give starting points with eight cones in 3D, we tested with the ½ $\theta(S,k)$ and with the $\theta(S,k)$. We also present two algorithms: Meeting walk and $\theta-Walk$.

Problems are formalized below.

Problem 1 (½ θ-graphs in 3D with eight cones): *Let $k=8$ be an integer, let $\theta=2\pi/k$, and let S be a set of points in $\mathbb{R}^3$. Assume that we have an undirected graph ½ $\theta(S,k)$ does there exist a t-spanner for S?*

Problem 2 (θ-graphs in 3D with eight cones): *Let $k=8$ be an integer, let $\theta=2\pi/k$, and let S be a set of points in $\mathbb{R}^3$. Assume that we have an undirected graph $\theta(S,k)$ has a t-spanner, and what are properties of $\theta(S,k)$ in 3D?*

3. Key results

We have some results with the $\theta-graph$, ½ $\theta-graph$ in 3D. In the ½ $\theta-graph$ with eight cones in 3D, we used the Meeting walk algorithm and we show that it is not a *t*- spanner. In the $\theta-graph$ with eight cones we used two algorithms: Meeting - Walk and $\theta-Walk$.

Let us define the path for routing algorithm, let *S* be a set of points in the plane. Assume that we have an undirected graph G with the property that for a two distinct points *u* and *v* in *S*. Then, we can (attempt to) construct a path in *G* between *u* and *v*. We have a path *P* $(s = u_0, u_1, \ldots, u_i = t)$.

We have a stretch of this path is: $\frac{\sum |u_i u_{i+1}|}{|st|}$, and the stretch factor of this routing algorithm is:

$$\max_{s,t}(\min_{Ps-t} stretch(P)) \leq stretch(P_{routing}(s,t))$$

We also give properties for this graph: For any *t* there exists a set *S* such that the $\theta - Walk$ algorithm have a stretch factor greater than *t*. For the Meeting walk algorithm we have shown that this algorithm is a *t*-spanner then $t \geq 7.67$ stretch factors. The property of $\theta - graph$ is a *t*-spanner then $t \geq 3$ stretch factors.

To show the geometric spanner graph on the ½ $\theta - graph$ and the $\theta - graph$ with eight cones and simulate our results we were programming the software in 3D. Also we simulate the $\theta - Walk$'s and Meeting walk's algorithms. We give some design to show properties for this graph.

4. Introduction to Spanner

Chew [5] discovered in 1986 that even planar networks (i.e. without crossings) could ensure small transmission distances. It was the beginning of a long standing study about the so-called *spanner networks*, which corresponds to the ratio between the actual path and the straight line distance – when moving in the networks from one site to another is at most a constant.

The first spanners Chew introduced were based on L_1 metric Delaunay triangulations: the *spanning ratio*- is bounded by $\sqrt{10}$. Chew [6], Dobkin *et al.* [9], and Keil and Gutwin [16] all studied the spanning ratio of Euclidean Delaunay Triangulations, and concluded that it is upper bounded by $\frac{2\pi}{2\cos\pi/6} \approx 2.42$.

Spanners with small size and/or weight are referred to as ***sparse spanner***.

- *Size*: is defined as the number of edges in the network. In general, it is preferable to have networks with as few edges as possible, perhaps linear in the number of points.
- *Weight:* is defined as the sum of the weights of the edges. Since any network must connect all the points, its weight is bounded from below by the weight of a minimum spanning tree. The weight is a good measure of the cost of building the network; thus, it is often desirable to have networks with small weight.

Let us give some definition of Spanner:

Definition 1.1 (Spanner): Let *S* be a set of *n* points in $\mathbb{R}^d$ and let $t \geq 1$ be a real number. A *t-spanner* for *S* is an undirected graph *G* with vertex set *S*, such that for any two points *u* and *v* of *S*, there is a path in *G* between *u* and *v*, whose length is less than or equal to $t|uv|$. Any path satisfying condition is called a *t-spanner path* between *u* and *v*.

If G is a t-spanner for the point set S, then obviously, G is also a t'-spanner for any real number t' with $t'>t$. This leads to the following definition:

Definition 1.2 (Stretch factor): Let S is a set of n points in $\mathbb{R}^d$ and let G be a Euclidean graph with vertex set S. The *stretch factor* of G is the smallest real number t such that G is a t-spanner of S.

We will denote the stretch factor by t. Let $|uv|_G$ is the Euclidean length of a shortest path between u and v in a geometric network G, $|uv|$ is the Euclidean distance between u and v. Note that

$$t = \max\left\{\frac{|uv|_G}{|uv|} : u, v \in S, u \neq v\right\}$$

5. The θ-graph

In this section we define θ**-graph** in two – dimension and higher – dimension. After, we present some important properties of it. We use the ½ $\theta-graph$ with six cones in the plane and eight cones in 3D to find some properties, after we want to find properties for $\theta-graph$ in 3D with eight cones.

The θ**-graph** has a rich history in the computational geometry. Yao [4] used this construction to compute the minimum spanning tree connecting n points in the d-dimensional space under the L_1, L_2 and L_∞ metric[1]. Using this graph Clarkson [13] solved the following motion planning problem in two and three – dimensions.

The notion of θ**-graph** were introduced by Keil [16]. Keil [16] and Keil and Gutwin [17] studied graphs approximating the complete Euclidean graph in the two dimensional space and they proved a trade off between

[1] For any real $1 \leq p \leq \infty$ the L_p distance of two points in $\mathbb{R}^d$ is defined as *(s, t) : =* $\left(\sum_{1\leq i\leq d} |s_i - t_i|^p\right)^{1/p}$, where s_i and t_i denote the i^{th} coordinate of s and t, respectively

the number of cones and the stretch factor of the θ**-graph.** Ruppert and Seidel [22] improved this trade off using some stronger definition of the θ-graph and they generalized it for higher dimensions.

Let θ and *k* be constants, with $k = 2\pi / \theta$. We restrict *k* to be an integer, such that θ divides 2π. A $\theta - graph\, G$ is a directed graph defined by a point set *V*. Each vertex has up to *k* outgoing edges connected to the closet vertex in *k* different cones. The i^{th} **cone** associated to a vertex *u* in a $\theta - graph$ is the subspace containing all the points with absolute angle for *u* between $i\theta$ included and $(i+1)\theta$ excluded. As there are $k = 2\pi / \theta$ cones for each vertex, the *k* cones cover the plane.

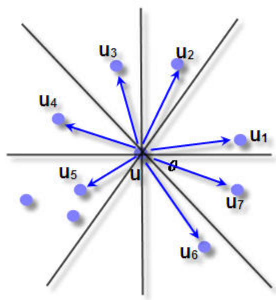

Figure 3: Plane subdivision using 8 cones, as used in $\theta - graphs$

Figure 3 show cones associated with one vertex. In each cone *i* of the vertex *u*, an outgoing edge is connected to the closet vertex, denoted by u_i, if such a vertex exists.

$\theta - graph$ have at most *kn* edges, at most *k* outgoing edges per vertex and at most *n – 1* ingoing edges. The diameter of the graph is $\theta(n)$ for instance when the vertices of the graph are placed on a line (here θ indicates an asymptotically tight bound, and not the parameter of the graph).

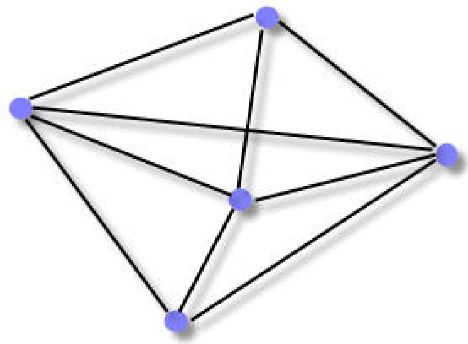

Figure 4: θ - Graph in the plane

5.1The two – dimension θ-graph

In this subsection we present the definition of the θ**-graph** in 2D done by G. Narasimhan and M. Smid [20] and give some basic properties of this spanner. Then we introduce the *θWalk* algorithm to find the compass routing from source to the destination.

A *cone* is the region in the plane between two rays that emanate from the same point, called the *apex* of the cone.

Let S be a set of n points in $\mathbb{R}^2$, $k \geq 2$ be an integer and $\theta = 2\pi / k$ an angle. If we rotate the positive $x-$ axis by angles $i\theta, 0 \leq i < k$, then we get k rays. Each pair of successive rays defines a cone whose apex is at the origin. We denote the collection of these k cones by C_k. It is clear that the cones of C_k partition the plane. Also, the two bounding rays of any cone of C_k make an angle θ.

For each cone $C \in C_k$, let l_C be a fixed ray that emanates from the origin and that is contained in C. The ray l_C can be chosen arbitrarily; as a concrete example, we can choose it to be the bisector of C. In other words, for the set of direction in cone C, l_C is a representative direction.

Let C be any cone of C_k and let p be any point in the plane. We defined $C_p := C + p := \{x + p : x \in C\}$; that is, C_p is the cone obtained by translating C such that its apex is at p. Similarly, we define, $l_{C,p} := l_C + p$.

Hence, $l_{C,p}$ is the ray that emanates from *p*, that is contained in the translated cone C_p and that is parallel to l_C.

Definition 2.1 (θ -graph) [20]: Let $k \geq 2$ be an integer, let $\theta = 2\pi / k$, and let *S* be a set of points in the plane. The undirected graph $\theta(S,k)$ is defined as follows:

a. The vertices of $\theta(S,k)$ are the points of *S*.
b. For each point *p* of *S* and for each cone *C* of C_k, such that the translated cone C_p contains one or more points of $S \setminus \{p\}$, the graph $\theta(S,k)$ contains one edge $\{p, r\}$, where *r* is a point in $C_p \cap S \setminus \{p\}$, whose orthogonal projection onto $l_{C,p}$ is closet to *p*.

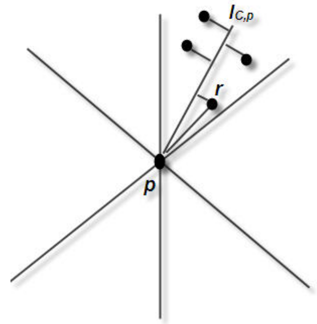

Figure 5: The graph $\theta(S,k)$ contains an edge between *p* and *r* for k = 6

Remark: If we take for *r* a point in $C_p \cap S \setminus \{p\}$ that is closet to *p*, in the Euclidean metric, then we obtain a graph that is called the *geometric neighborhood graph.* It can be shown that this graph has properties similar to that of a θ**-graph** and is a sparse *t*-spanner, for some real number *t* that depends on the angle θ.

5.1.1 Spanner property

Keil [11] proved that the θ**-graph** for a given set of points is a spanner if $0 < \theta < \frac{\pi}{4}$ and he established an upper bound on the stretch factor of the obtained graph in dependence on the value of θ. G. Narasimhan and M.

Smid [20] proved the spanner property of the θ**-graph** for $k \geq 9$ an upper bound on the length of the path is at most a constant factor times the Euclidean distance, where the constant depends on *k*. In this subsection we present an obvious algorithm that constructs a path in $\theta(S,k)$ between two points' *u* and *v* of *S*.

θ-Walk Algorithm

Goal: Give source *u* and destination *v*; find a path from *u* to *v*.

Idea: Let *S* be a set of points in the plane. Assume that we have an undirected graph G with the property that for an two distinct points *u* and *v* in *S*, *G* contains an edge *(u, r)* such that

a. the vector $\overrightarrow{ur}$ points "in the general direction" of *v*, and
b. Following this edge from *u* to *r* does not take us "too far" beyond *v*.

Then, we can (attempt to) construct a path in *G* between *u* and *v* as follows: Start at $u_0 := u$. Let $i \geq 0$, and assume we have already constructed a path $u_0, u_1, \ldots, u_i$. If $u_i = v$, then we have reached our destination. Otherwise, if $u_i \neq v$, but (u_i, v) is an edge of *G*, then we follow this edge, and arrive at our destination. Assume that $u_i \neq v$, and (u_i, v) is not an edge of *G*. Let u_{i+1} be a point of *S* such that (u_i, u_{i+1}) is an edge of *G* that satisfies a. and b. above. That is, (u_i, u_{i+1}) takes us in the general direction of *v*, but not too far beyond *v*. Then u_{i+1} are the next point on our path.

A formal description of this algorithm is given below.

Algorithm θ-WALK (u, v)

Comment: This algorithm take as input two points *u* and *v* in *S*, and returns a path in $\theta(S, k)$ between *u* and *v*.

```
u0 := u;
i := 0
while ui ≠ v
do
```

```
    C := cone of C_k such that v ∈ C_{u_i};
    u_{i+1} := point of C_{u_i} ∩ S\{u_i} such that {u_i, u_{i+1}}
    is an edge of θ(S, k);
    i := i + 1;
endwhile;
return the path u_0, u_1,…,u_i
```

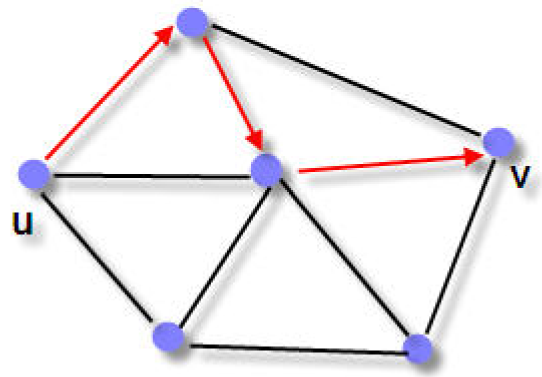

Figure 6: Path in 2D

Below, we prove an upper bound on the length of the path constructed by this algorithm.

Theorem 2.2: [20] *Let $k \geq 9$ be an integer; let $\theta = 2\pi / k$, and let S be a set of points in the plane. The graph $\theta(S,k)$ is a t − spanner for S, for $t = 1/(\cos\theta - \sin\theta)$. It contains at most kn edges.*

The proof of this Theorem is based on the following lemma 2.3.

Lemma 2.3: [20] *Let $k \geq 8$ be an integer, let $\theta = 2\pi / k$, let u and v be two distinct points in the plane, and let C be the cone of C_k such that $v \in C_u$. Let r be a point in C_u such that the orthogonal projection of r onto the ray $l_{C,u}$ is at least as close to u as the orthogonal projection of v onto $l_{C,u}$. Then,*

a. $|ur| \leq |uv| / \cos\theta$ *,and*

b. $|rv| \leq |uv| - (\cos\theta - \sin\theta)|ur|$

These proves are presented in [20].

5.1.2 The spanner property ½ θ-graph with six cones in the plane

In this subsection we give a formal definition of ½ θ- **graph** [12] and present the "*Meeting - walk*" algorithm. After that, we prove which ½ θ-**graph** with six cones has a property spanner and its stretch factor is 2 – spanners and it is a spanner of TD – Delaunay graph [6].

Let $k \geq 2$, and define $\theta = 2\pi / k$, let C_k be the collection of these k cones. It is clear that C_k partition the plane into k cones. Recall that (i) each cone in C_k has its apex at the origin, (ii) the angular diameter of each cone in C_k is at most θ, (iii) between two adjacent cones (sectors), choose 1 cone, this cone is know as a positive cone; For each positive cone define each color.

As in definition of θ **-graph**, for each cone $C \in C_k$, a ray l_C that emanates from the origin and that is contained in C. For any point p in the plane, we define $C_p := C + p = \{x + p : x \in C\}$ and $l_{C,p} := l_C + p$.

Definition 2.4 (½θ-graph): Let $k \geq 6$ be an integer, let $\theta = 2\pi / k$, and let S be a set of points in the plane. The ½ θ**-graph** $\theta(S, k)$ is defined as follows:

a. The vertices of ½ $\theta(S, k)$ are the points of S in positive cones.
b. For each positive cone give each color.
c. For each point p of S and for each cone C of C_k, such that the translated cone C_p contains one or more points of $S \setminus \{p\}$, the graph ½ θ (S, k) contains one edge $\{p, r\}$, where r is a point in $C_p \cap S \setminus \{p\}$ whose orthogonal projection onto $l_{C,p}$ is closet to p.

For example: For $k = 6$ in $\theta - graph$, we have 6 cones with 6 directions: North_East, North, North_West, South_West, South_East, South. In ½ $\theta - graph$ we have 3 cones with 3 directions: North (color1), South_West (color2), South_East (color3).

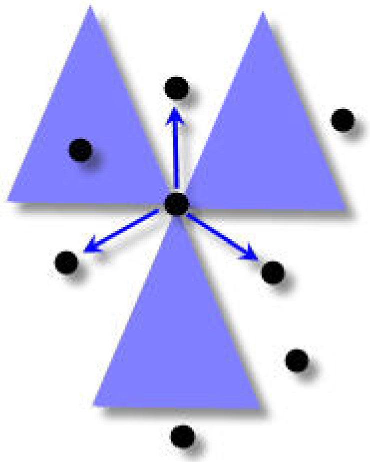

Figure 7: ½ $\theta - graph$ **with six cones**

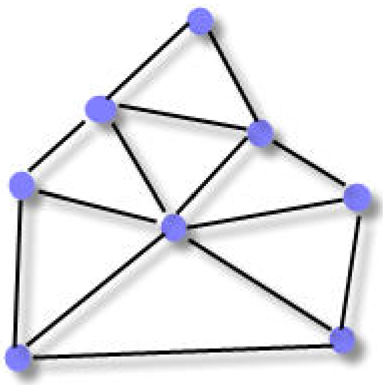

Figure 8: ½ $\theta - graph$ **with six cones in the plane**

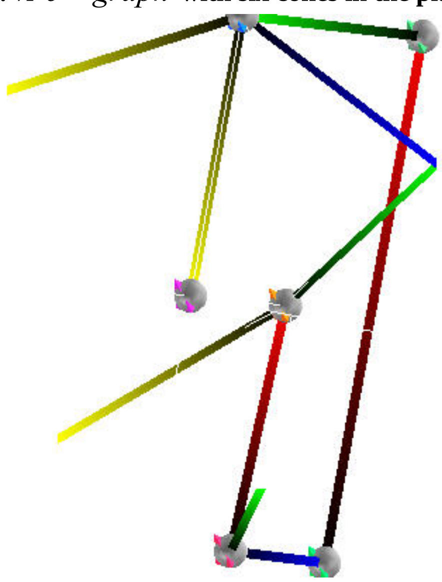

Figure 9: ½ $\theta - graph$ **with eight cones in 3D**

"Meeting Walk" algorithm

Goal: Give source *u* and destination *v*; find a path from *u* to *v* and a path from *v* to *u*.

Idea: Let *S* be a set of points in the plane. Assume that we have an undirected graph *G* with the property that for a two distinct points *u* and *v* in *S*, *G* contains an edge *(u, v)* such that

a. The vector $\overrightarrow{ur}$ points "in the positive cone" of *v*, and
b. Following this edge from *u* to *r* does not take us "too far" beyond *v*.

Then, we can (attempt to) construct a path in *G* between *u* and *v* as follows: Start at $u_0 = u$, $v_0 = v$. Let $i, j \geq 0$, and assume we have already constructed paths $u_0, u_1 \ldots u_i$; $v_0, v_1 \ldots v_j$. If $u_i = v_j$, then we have reached our destination. Otherwise, if $u_i \neq v_j$ but (u_i, v_j) is an edge in positive cone of *G*, then we follow this edge, and arrive at our destination. Assume that $u_i \neq v_j$ and (u_i, v_j) is not an edge in positive cone of *G*.

If u_i is in a "positive" cone of v_j, let v_{j+1} is a point of *S* such that (v_j, v_{j+1}) is the edge in the positive cone of v_j containing u_i. Then v_{j+1} is the next point on our path.

Else if v_j is in a "positive" cone of u_i, let u_{i+1} be a point of *S* such that (u_i, u_{i+1}) takes us go to the direction of *v,* but not too far beyond *v*. Then u_{i+1} is the next point on our path.

A formal description of this algorithm is given below.

Algorithm Meeting Walk (u, v)

Comment: This algorithm take as input two points *u* and *v* in *S*, and returns a path in ½ $\theta(S,k)$ between *u* and *v*.

```
u0 = u;
v0 = v;
i := 0;
j := 0;
```

```
while u_i ≠ v_j
do
if (u_i is in a "positive" cone of v_j)
     C := cone of C_k such that this is a positive
     cone and u_i ∈ C_{v_j}
     V_{j+1} := point of C_{v_j} ∩ S \ {v_j} such that (v_j, v_{j+1})
     is an edge in the positive cone of ½ θ(S,k);
     j := j + 1;
else
     C := cone of C_k such that this is a positive
     cone and v_j ∈ C_{u_i}
     u_{i+1} := point of C_{u_i} ∩ S \ {u_i} such that (u_i, u_{i+1})
     is an edge in the positive cone of ½ θ(S,k);
     i := i + 1;
endwhile;
return the path u = u_0, u_1, …, u_i; v = v_0, v_1,…,v_j
```

Theorem 2.5: *Let let* $\theta = 2\pi / 6$. *For any two points u and v of S, algorithm Meeting Walk (u, v) constructs a t-spanner path in Meeting Walk (S, 6) between u and v, for t = 2.*

The proof this Theorem is based on the following lemma 2.6.

Lemma 2.6: *The meeting path is the union of 2 mono-colored oriented path that stays in the triangle u, v*

Proof (Lemma 2.6):

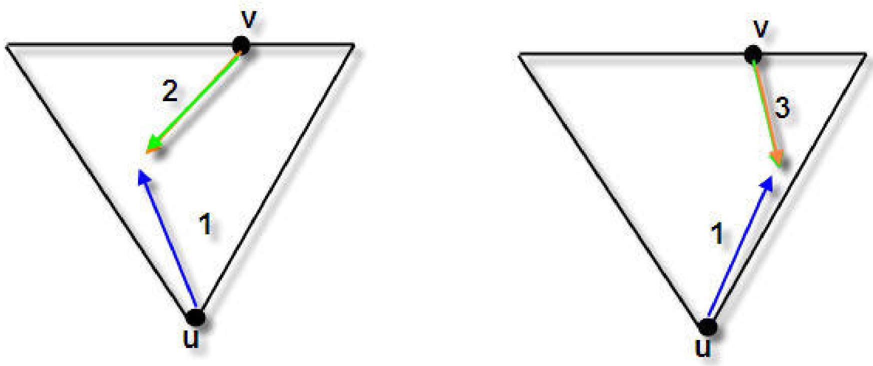

Figure 10: Two oriented path stays in the triangle u, v

Let $\theta = 2\pi / 6$, let S be a set of points in the plane, let $u, v \in S$, let G *(S, 6)* is the graph of $½\theta - graph$. By the definition of $½\theta - graph$, we have 3 positive cones such as: North, South_West, South_East, for each cone we have different colors: color1-blue (the oriented path go from *u* to *v*), color2-green (the oriented path go from *v* to the left of *u*), color3-orange (the oriented path go from *v* to the right of *u*). See figure 10.

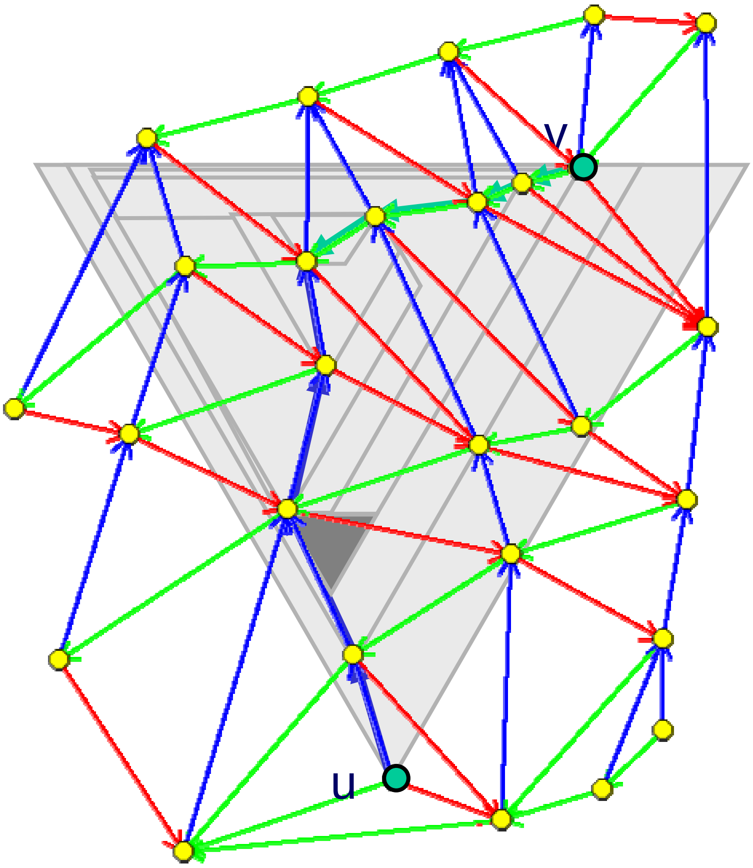

Figure 11: ½ θ-graph has stretch factors 2-spanners

In this proof, we use the notation of algorithm Meeting-Walk. Consider the two partial paths $u = u_0, u_1, u_2...$ and $v = v_0, v_1, v_2...$ that are constructed by the algorithm. The proof of the lemma is by induction on the number of levels. To prove the base case, assume that Meeting – Walk consists of only one level. In the first level, a path $u = u_0, u_1, u_2...$ is constructed. This inner terminal if and only if the last point on this path is equal to v. Let $i \geq 0$ and consider the point's u_i and u_{i+1}. Then $u_i \neq v$. Let C be the positive cone of u such that $v \in C_{ui}$. We have the triangle (u, v) has u is the apex, v is above u. It follows from the algorithm and the definition of the graph $\theta(S, 6)$ that: (i) (u_i, u_{i+1}) is an edge of this graph, (ii) $u_{i+1} \in C_{ui}$. As a result, this inner while – loop terminates. Let j

be the number of iterations made. Then the algorithm has constructed a path $u = u_0, u_1, u_2 \ldots u_j = v$.

The second case is when u_{i+1} remains at the right of v, the triangle (u_{i+1}, v) is empty because by the definition of θ-graph we do not have two paths stays in one cone.

Consider again the first iteration of the inner while-loop a path $v = v_0, v_1, v_2 \ldots$ is constructed. This inner terminal if and only if the last point on this path is equal to u_{i+1}. Let $j \geq 0$ and consider the points v_j and v_{j+1}. Then $v_j \neq u_{i+1}$. Let C be the positive cone of v such that $u_{i+1} \in C_{vj}$. It follows from the algorithm and the definition of the graph θ $(S, 6)$ that: (i) (v_j, v_{j+1}) is an edge of this graph, (ii) $v_{j+1} \in C_{vj}$. As a result, this inner while – loop terminates when v_{j+1} and u_{i+1} are meeting. See figure 11.

We have shown that in the triangle the meeting path is union 2 mono-colored oriented direction path.

■

Proof (Theorem 2.5):

Now we prove the stretch factor of this graph is *2* – spanners. See figure 12.

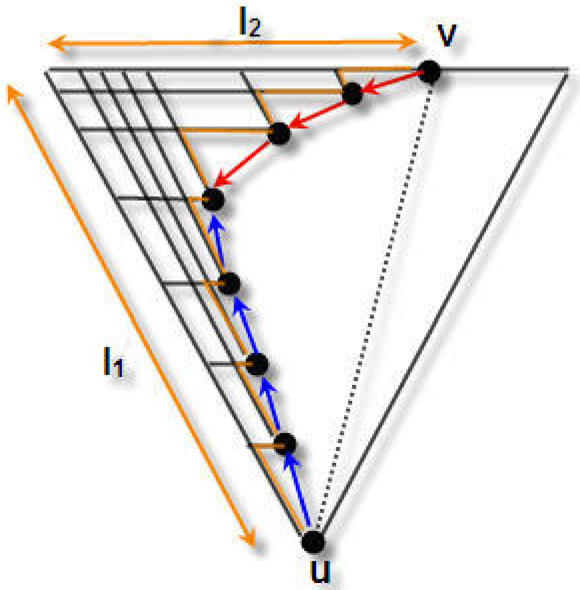

Figure 12: Path from *u* to *v*

Let *P (u, v)* is the path between *u* and *v*. Let *C* be any cone of C_k, let l_C be a fixed ray that emanates from the each vertex and that is contained in *C*. Consider the two partial paths $u=u_0, u_1, u_2$... and $v=v_0, v_1, v_2$... that are constructed by the algorithm *Meeting walk.* From the Lemma 2.4, the Meeting algorithm is the path union of 2 mono-colored oriented path stays in the triangle *u, v*.

Let $l_1 = |u_0 u_0^{'}|$, $l_2 = |u_0^{'} v|$

$\forall i \geq 0$, let $u_{i+1}^{'}$ be the projection of u_{i+1} onto l_1, let $u_{i+1}^{"}$ be the projection of u_{i+1} onto l_2

Applying the triangle inequality, we get:

$$|u_i u_{i+1}| \leq |u_i^{'} u_{i+1}^{'}| + |u_{i+1}^{"} u^{"}_{i+1}|$$

The path between *u* and *v* has length at most:

$$P(u,v) \leq \sum_{i=0}^{i-1} |u_i^{'} u_{i+1}^{'}| + \sum_{i=0}^{i-1} |u_{i+1}^{"} u^{"}_{i+1}|$$

We have:

$$P(u,v) \leq l_1 + l_2$$

We have:

$$|uv| = \sqrt{l_1^{\,2} + l_2^{\,2} - l_1 l_2}$$

$$|uv| \le \frac{l_1 + l_2}{2}$$

$$P(u,v) \le 2\,|\,uv\,|$$

We have shown that the graph ½ θ*(S, k)* is a *t*-spanner of *S* for *t = 2*.

■

5.2 The higher – dimension θ- graph [20]

Now we examine the *d*-dimensional θ**-graph**, where $d \ge 3$ an integer is constant. We introduce the notion of cones. A (simplicial) *cone* is the intersection of *d* half-spaces in $\mathbb{R}^d$. The hyper-planes that bound these half-spaces are assumed to be in general position, in the sense that their intersection is a point, called the *apex* of the cone. In the plane, a cone having its apex at the point *p* is wedge bounded by two rays emanating from *p* that make an angle at most equal to π.

Let *C* be any cone in $\mathbb{R}^d$ having its apex at the point *p*. The *angular diameter* of *C* is defined as the maximum angle between any two vectors $\overrightarrow{pq}$ and $\overrightarrow{pr}$, where *q* and *r* range over all points of $C \cap R^d$.

For a point $p \in \mathbb{R}^d$ and a set $V = \{v^1, v^2, \ldots, v^k\}$ of points in $\mathbb{R}^d$, we define the *cone with apex p* that is *generated by V* to be the set

$$cone(p,V) := \left\{ p + \sum_{j=1}^{k} \lambda_j v^j : \lambda_j \ge 0, \forall j = 1,2,\ldots,k \right\}$$

If *p = 0*, that is, *p* is the origin, then we write *cone (V)* instead of *cone (0, V).* Thus *cone (V)* is the set of all points obtained by linear combinations with nonnegative coefficients of points in *V*, and *cone (p, V)* is *cone (V)* translated to point *p.*

Let θ be a real number, such that $0 < \theta < \pi$. A θ*-frame* is a collection *C* of cones, having the following properties:

a. Each cone in *C* has its apex at the origin.

b. The cones in C cover $\mathbb{R}^d$; that is, $\bigcup_{C\in C} C = \mathbb{R}^d$

c. The angular diameter of each cone in C is at most θ

d. Each cone in C is a simplicial cone.

We call such a collection C of simplicial cones a θ-frame. We use the θ-frame definition suggested by G. Narasimhan and M. Smid [20].

Let θ be a real number such that $0 < \theta < \pi$, let $k = k_{d\theta} = k(d-1)! = 2d!\left\lceil \sqrt{2(d-1)/(1-\cos\theta)} \right\rceil^{d-1}$, and let C_k be the θ-frame[20]. Recall that (i) each cone in C_k has its apex at the origin, (ii) the angular diameter of each cone in C_k is at most θ, (iii) each cone in C_k is a simplicial cone; that is, it is the intersection of d half-spaces, (iv) the cones in C_k cover $\mathbb{R}^d$, and (v) the number of cones in C_k is $k = O(1/\theta^{d-1})$.

For each cone $C \in C_k$, let l_C be fixed rays that emanates from the origin and that is contained in C. The ray l_C can be chosen arbitrarily.

Let C be any cone of C_k and let p be any point in the plane. We define $C_p := C + p := \{x + p : x \in C\}$; that is, C_p is the cone obtained by translating C such that its apex is at p. Similarly, we define $l_{C,p} := l_C + p$.

Definition 2.6 (θ-graph in $\mathbb{R}^3$): Let S be a set of n point in $\mathbb{R}^3$. The 3-dimensional θ**-graph** $\theta(S,k)$ is defined as follows:

a. The vertices of $\theta(S, k)$ are the points of S.

b. For each point p of S and for each cone C of C_k, such that the translated cone C_p contains one or more points of $S \setminus \{p\}$, the graph $\theta(S, k)$ contains one edge $\{p, r\}$, where r is a point in $C_p \bigcap S \setminus \{p\}$ whose orthogonal projection onto $l_{C,p}$ is closet to p.

This section is organized as follows. In subsection 5.2.1 we take a closer look at proof of the spanner property of the d-dimensional θ**-graph**, done by G. Narasimhan and M. Smid [20] and we show why it actually works also in d-dimensional. After this, in Subsection 5.2.2 we present the property of ½ θ**-graph** with eight cones in 3D, we show that the graph is not a *t*-spanner. We give some spanner properties for the θ-**graph** with eight cones.

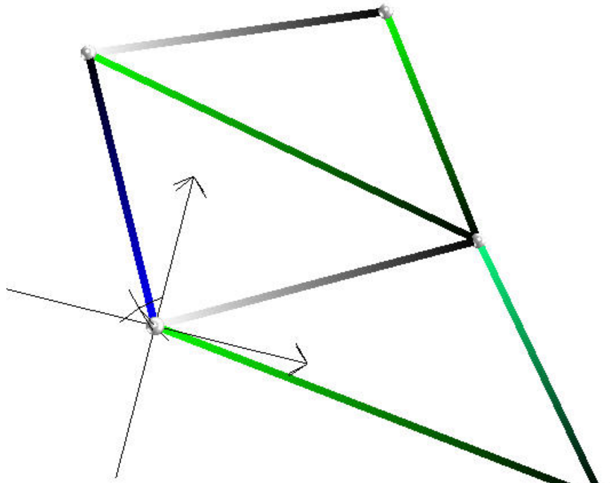

Figure 13: The graph of θ-graph with eight cones in 3D

5.2.1 Spanner property

We give our proof for the *d*-dimensional version of Lemma 2.7 which is base of the proof of the spanner property.

Theorem 2.7[20]: *Let S be a set of n points in* $\mathbb{R}^d$*, let* θ *be a real number such that* $0 < \theta < \pi / 4$*, and let* $k = k_{d_\theta}$.

a. *The graph* $\theta(S,k)$ *is a* $t-$*spanner for S, for* $t = 1/(\cos\theta - \sin\theta)$

b. *The graph* $\theta(S,k)$ *contains at most* $kn = O(n/\theta^{d-1})$ *edges.*

c. *The graph* $\theta(S,k)$ *can be constructed in* $O((n/\theta^{d-1})\log^{d-1} n)$ *time, using* $O(n/\theta^{d-1} + n\log^{d-2} n)$ *space.*

This proof is presented in [20].

Lemma 2.8[20]: *Let* $0 < \theta \le \pi$ *and C be a* $\theta-$*frame. Let p and q be any two distinct points in* $\mathbb{R}^d$*, and let C be the cone of C such that* $q \in C_p$*. Let r be any point in* $\mathbb{R}^d \cap C_p$ *such that the projection of r onto the ray* $l_{C,p}$ *is* *at least as close to p as the projection of q onto* $l_{C,p}$. *Then* $|rq| \le |pq| - (\cos\theta - \sin\theta)|pr|$

Proof

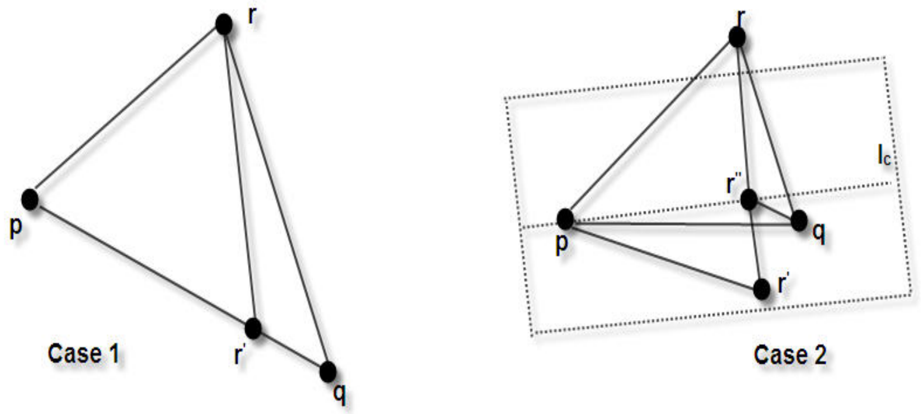

Figure 14: The two cases in the proof of Lemma 2.7

We distinguish two cases depending on whether $|pr| \le |pq|$ or $|pr| > |pq|$

Case 1: $|pr| \le |pq|$. For this case we can simply repeat the proof of case 1 of Lemma 2.3 , i.e., we immediately obtain the claim by applying the triangle inequality to the triangle $rr'q$, where r' is the point on the line segment pq such that $|pr| = |pr'|$

Case 2: $|pr| > |pq|$. Let r'' be the perpendicular projection of r to the cone axis l_C and let r' be the point on the 2 – dimension plane containing the triangle pqr'' such that $|pr'|=|pr|$ and $|r''r'|=|r''r|$. Then we obtain the inequalities:

$$|rq| \le |rr''| + |r''q|, \qquad (1)$$

$$|r''q| + |pr'| \le |r''r'| + |pq| \qquad (2)$$

Inequalities (1) and (2) together with $|rr''| = |r''r'|$; $|rr''| \le \sin\theta|pr|$ $|pr'| \le \cos\theta|pr|$. Imply that:

$$|rq| \le |rr''| + |r''r'| + |pq| - |pr'|$$

$$\le |pq| - (\cos\theta - \sin\theta)|pr|$$

■

5.2.2 The spanner property for ½ θ-graph with eight cones

In this subsection we give properties of ½ θ**-graph** and θ**-graph** in 3D with the starting point: eight points. First we examine the ½ θ**-graph** for *S*. We prove the spanner property of this graph is not a *t*-spanner, the ½θ**-graph** in 3D is not work. After this we examine with θ**-graph**, we used the "$\theta - Walk$"algorithm and "Meeting Walk" algorithm to find properties for the θ**-graph** in 3D.

Let us define positions and colors for 8 points in the space.

Yellow: (+, +, +) is the cone defined by $v_1 = (v_x, v_y, v_z)$;

Red: (-, -, +) is the cone defined by $v_2 = (-v_x, -v_y, v_z)$;

Green: (+, -, -) is the cone defined by $v_3 = (v_x, -v_y, -v_z)$;

Blue: (-, +, -) is the cone defined by $v_4 = (-v_x, v_y, -v_z)$;

Pink: (-, -, -) is the cone defined by $v_5 = (-v_x, -v_y, -v_z)$;

Black – white: (+, +, -) is the cone defined by $v_6 = (v_x, v_y, -v_z)$;

Orange: (-, +, +) is the cone defined by $v_7 = (-v_x, v_y, v_z)$;

Blue – green: (+, -, +) is the cone defined by $v_8 = (v_x, -v_y, v_z)$;

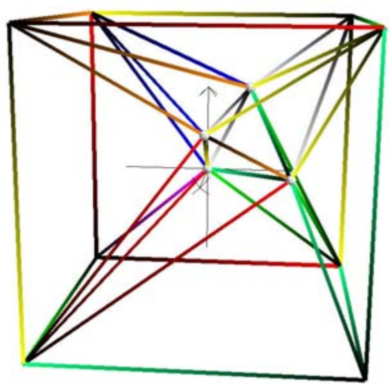

Figure 15: θ-graph with 8 cones in the space

For the ½ θ**-graph** we have four cones with four positions and colors:

Yellow: (+, +, +) is the cone defined by $v_1 = (v_x, v_y, v_z)$;

Red: (-, -, +) is the cone defined by $v_2 = (-v_x, -v_y, v_z)$;

Green: (+, -, -) is the cone defined by $v_3 = (v_x, -v_y, -v_z)$;

Blue: (-, +, -) is the cone defined by $v_4 = (-v_x, v_y, -v_z)$;

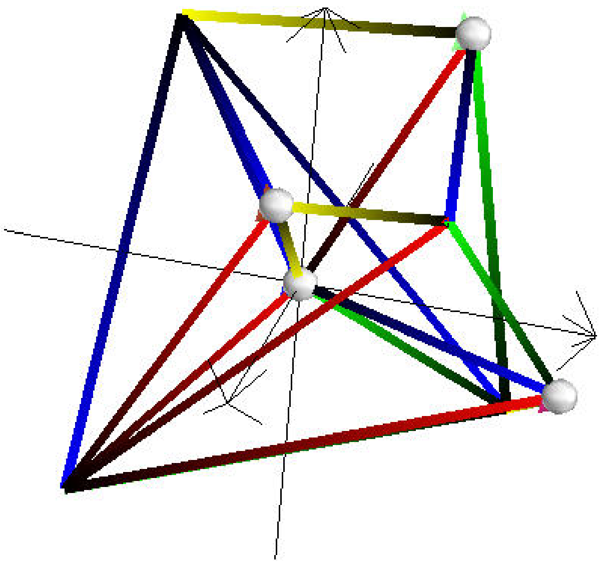

Figure 16: ½ θ-graph with 8 cones in the space

Theorem 2.9: *Let $k = 8$, let $\theta = 2\pi / k$, for any t, there exists a set S such that the ½ θ-graph of S is not a $t-$spanner.*

Proof:

We present a construction (see Figure 17) for *n* points in the plane, such that the ½ θ**-graph** of these points is not a *t*-spanner for any constant *t*.

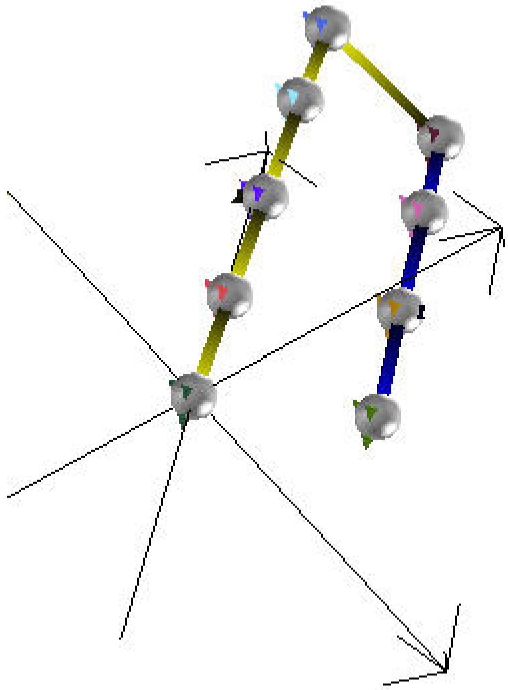

Figure 17: Counter example of ½ θ-graph with eight cones in 3D

Consider the vertex sets $U = \{u_0, u_1 u_2 \ldots u_m\}$ and $V = \{v_1, v_2 \ldots v_m\}$,

Let $u_0 = (0, 0, 0)$ we construct u_1 stay in positive cone of u_0 and v_0 stay in negative cone of u_0. By the definition of ½ θ**-graph** we have the path go from u_0 to u_1 and do not have path go to v_0. We continue to construct the path from u_1, we have an edge (u_1, u_2). At vertex u_m we have v_m stay in positive cone of u_m, we have an edge (u_m, v_m). From v_m we construct the vertex v_{m-1} stay in positive cone of v_m, we continue until we reached v_0.

We have to shown that in the ½ θ**-graph** the only edges are only between vertices (u_i, u_{i+1}) and (v_i, v_{i+1}), (u_m, v_m). There is only one path in the θ**-graph**. In this construction we can continue to add vertex u and v to infinite. The length stretch factor of this θ**-graph** cannot be bounded by a constant. We have to shown that for ½ $\theta(S, k)$ with eight cones is not a t-spanner.

■

5.2.3 The spanner property for θ-graph with eight cones

In this subsection we present properties of $\theta - graph$ with eight cones in $\mathbb{R}^3$. We used two algorithms: $\theta - Walk$ and Meeting Walk. To found properties we got some questions:

- What is the worse case for this graph?
- Does the $\theta - Walk$ algorithm work in 3D?
- What is the stretch t_1 of the θ-graph?
- What is the stretch t_2 of $\theta - Walk$ path?
- What is the stretch t_3 of Meeting Walk path?

Theorem 3.0: *Let k = 8, let* $\theta = 2\pi / k$ *, for any t, there exists a set S such that the* θ*-walk algorithm have a stretch greater than t.*

Proof:

As for ½ θ**-graph** in 3D, let *S* be a set of points in $\mathbb{R}^3$, let $u, v, w \in S$. Let *u (0, 0, 0)* and $v\ (x_v, y_v, z_v)$. For a sector i^{th}, define the $\theta - neighbor$ *w* in the same cone of *u* and *v*, such that the distance of |*uv*| is *same length* with |*uw*| or $|uw| \le |uv|$. By the definition of $\theta - graph$ we always have path from *u* to *w*.

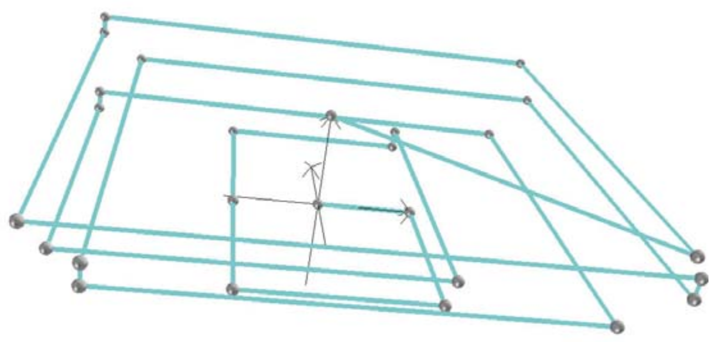

Figure 18: The worst case of θ-graph with eight cones

We construct this model like the spiral such that all vertices turn around the target.

For the first round, from the neighbor *w*, in a sector i^{th} add the vertex has a coordinate stay in the same space of *w*, *u*; make it as long as possible. By the definition of θ - walk we have a path go from *w* to this neighbor and do not have the path go to *u*. We continue to construct vertices make these vertices turn around our source.

For a second round, we create an upper vertex is closet to the end of vertex of the first round. In a second round make the path is going through there vertices. We continue this method until it is the same space of the target and we reached our target. See figure 18.

We have to shown that in this construction the stretch of this path is greater than *t* for any *t*.

■

Property 1: *Let k = 8, let* $\theta = 2\pi / k$ *, let S be a set of points in* $\mathbb{R}^3$*. If meeting walk algorithm of* $\theta - graph$ *with eight cones in* $\mathbb{R}^3$ *is a t – spanner then t ≥7.67 stretch factors.*

Proof:

Let *S* be a set of points in the plane, let *u, v, w,* $t \in S$. Let *u (0, 0, 0)* and *v* (x_v, y_v, z_v). For a sector i^{th}, define the $\theta - neighbor$ *w* in the same cone of *u* and *v*, such that the distance of |*uv*| is *same length* with |*uw*| or $|uw| \le |uv|$. By the definition of $\theta - graph$ we always have path from *u* to *w*.

Continue to construct the path has general direction goes to *v*. From neighbor *w*, in sector i^{th}, i'^{th} of *w*, add two vertices w_1, w_9 stay in same space of *w* and *u*. Make two vertices as far as possible, from the definition of $\theta - graph$ we have path from *w* to w_1, and w_2 and do not have path from *u*.

From the sector i^{th} and i'^{th} of w_1, w_9 add four vertices w_{11}, w_2, w_8, w_{10}, such that w_2, w_8 is same space as t, and w_{11}, w_{10} are same space with w_1, w_9.

Continue to construct these vertices w_7, w_3, w_4, w_5 and w_6 is same space as t. See figure 19.

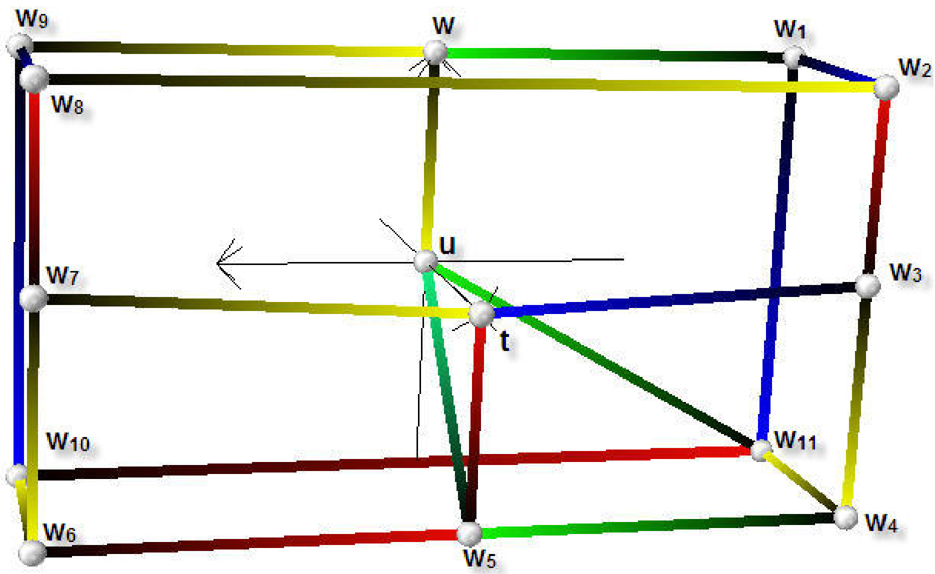

Figure 19: The graph of θ –graph

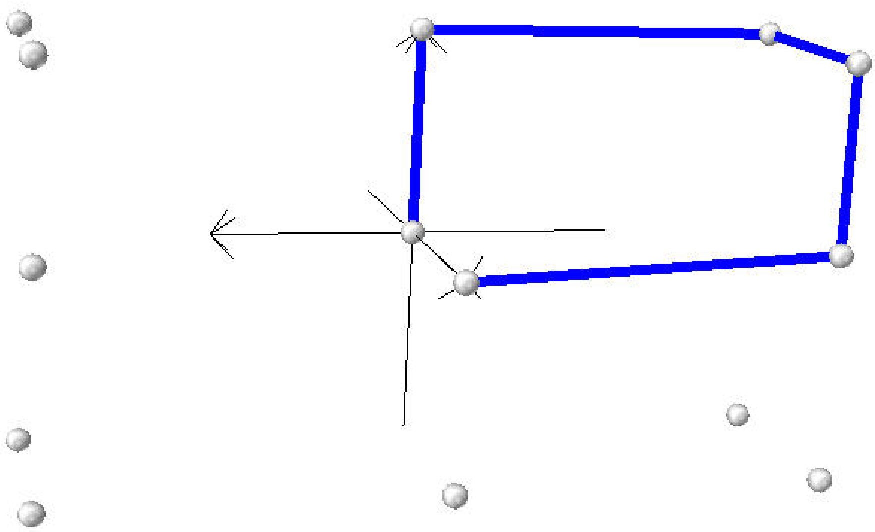

Figure 20: The graph of Meeting path with 7.67 stretch factors

■

Property 2: *Let k = 8, let* $\theta = 2\pi / k$ *, let S be a set of points in* $\mathbb{R}^3$*. If the* $\theta - graph$ *with eight cones in* $\mathbb{R}^3$ *is a t – spanner then t ≥ 3 stretch factors.*

Proof: Let *S* be a set of points in the plane, let *u, v, w,* $t \in S$. Let *u (0, 0, 0)* and *v* (x_v, y_v, z_v). For a sector i^{th}, define the $\theta - neighbor$ *w* in the same cone of *u* and *v*, such that the distance of $|uv|$ is *same length* with $|uw|$ or $|uw| \leq |uv|$. By the definition of $\theta - graph$ we always have path from *u* to *w*. Construct vertices like the figure 21.

For the bounded of θ –graph in this time we found the stretch $t \geq 3$. We hope it will have more stretch.

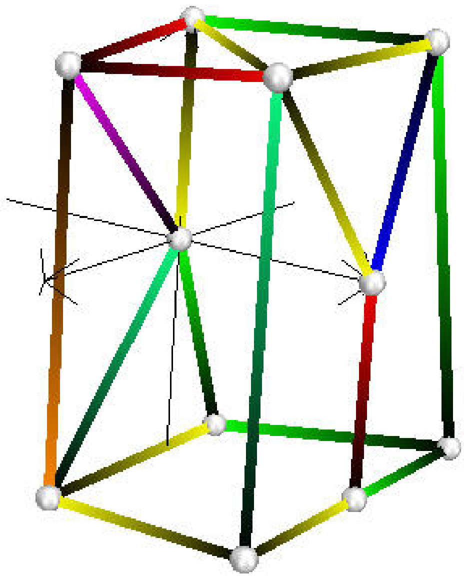

Figure 21: The graph of θ –graph with 3 stretch factors

Conclusion

We have studied the two- and higher dimension $\theta - graph$ for a given point set. We have introduced the spanner property of the $\theta - graph$ and two routing algorithms: θ-*walk* and Meeting walk.

Then we have considered higher dimension $\theta - graph$. We have shown that in the 3-dimension case, the ½ $\theta - graph$ with eight cones there exists a set *S* such that the ½ $\theta - graph$ of *S* is not a *t*-spanner. After that, we shown that with the θ-*walk* algorithm for any *t*, there exists a set *S* have a stretch greater than *t*. For the meeting walk algorithm of $\theta - graph$ with eight cones we found the property of this algorithm is a *t*-spanner then $t \geq 7.67$ stretch factors. And another property for the $\theta - graph$ with eight cones is a $t -$ spanner then $t \geq 3$ stretch factors.

In this time we just found the property of the Meeting algorithm and the $\theta - graph$ with eighth cones. We want to know more properties of the Meeting algorithm, it has an infinite stretch factor, and what is the stretch of the $\theta - graph$.

Bibliography

[1] I. Althofer, G. Das, D. Dobkin, D. Joseph, and J. Soares, *On sparse spanners of weighted graphs,* Discrete & Computational Geometry **9** (1993), no. 1, 81-100.

[2] S.Arya, D. M. Mount, and M.Smid. Randomized and deterministic algorithms for geometric spanners of small diameter. In *35^{th} IEEE Symposium on Foundations of Computer Science (FOCS'94), pages 703-712, 1994.*

[3] J. Burge. *Geometric Optimization Project*. Comparison of Geometric Spanner Construction Methods for Network Applications.

[4] A. C. Yao. On constructing minimum spanning tree in k – dimensional spaces and related problems. *SIAM Journal on Computing*, 11:721 – 736, 1982.

[5] L.P. Chew, *There is a planar graph almost as good as the complete graph,* Proceedings of the 2^{nd} Annual ACM Symposium on Computational Geometry (SoCG), 1986, pp. 169-177.

[6] L.P. Chew, *There are planar graphs almost as good as the complete graph,* Journal of Computer and System Sciences **39** (1989), no. 2, 205-219.

[7] S. Collette, *Regions, Distances and Graphs*, 2: 18 – 21, Annee academique 2006-2007

[8] G. Das and D. Joseph, *Which triangulations approximate the complete graph?*, Proceedings of the International Symposium on Optimal algorithms, 1989, pp. 168-192.

[9] D. Dobkin, S. Firedman, and K. Supowit, *Delaunay graphs are almost as good as complete graphs,* Discrete & Computational Geometry **5** (1990), 399-407.

[10] D. Epstein. *Spanning Trees and Spanners*. Dept. Information and Computer Science, University of California, Irvine, CA 92717. Tech. Report 96-16, May 8, 1996

[11] M. Fischer, F. Meyer auf der Heide, and W. – B. Strothmann. Dynamic data structures for real-time management of large geometric scenes. In *5th Annual European Symposium on Algorithms (ESA '97),* pages 157-170, 1997.

[12] N. Bonichon, C. Gavoille, N.Hanusse and D. Ilcinkas Equivalence of half - θ -graph, TD – Delaunay triangulations and geometric embedding of co – planar orthogonal surfaces, 2009

[13] K. L. Clarkson. Approximation algorithms for shortest path motion planning. In *19th ACM Symposium on Theory of Computing (STOC'87), pages 56-65, 1987.*

[14] T. Lukovszki. *New Results on Geometric Spanners and Their Applications*. Fachbereich Mathematik/ Informatik und Heinz Nixdorf Institut Universitat – Gesamthochshule Paderborn, 2:9 – 49, 1999

[15] Linial, London, and Rabinovich. *The geometry of graphs and some of its algorithmic applications.* Combinatorica, 15, 1995.

[16] J. M. Keil. Approximating the complete Euclidean graph. In *1st Scandinavian Workshop on Algorithm Theory (SWAT'88),* pages 203-213, 1988.

[17] J. M. Keil and C. A. Gutwin. Classes of graphs which approximate the complete Euclidean graph. *Discrete & Computational Geometry*, 7: 13-28, 1992.

[18] K. Mulmuley. *Computational Geometry, an Introduction through Randomized Algorithms.* Prentice Hall, Englewood Cliffs, 1994.

[19] G. Narasimhan and Michiel Smid. *Approximating the stretch factor of Euclidean Graphs*. 2000

[20] G. Narasimhan and Michiel Smid. *Geometry Spanner Networks*. Cambridge University Press, 2007

[21] D. Peleg and A. Schaffer. *Graph spanners.* J. Graph Theory, 13:99 -116, 1989

[22] J. Ruppert and R. Seidel. Approximating the d – dimension complete Euclidean graph. In *3rd Canadian Conference on Computational Geometry (CCCG'91),* pages 207-210, 1991.

[23] K. Volbert. *Geometric Spanners for Topology Control in Wireless Networks*. Fakultat fur Elektrotechnik, Informatik und Mathematik. Institut fur Informatik und Heinz Nixdorf Institu. Universitat Paderborn, 3:17 – 34, February 2005.