Rapport du projet de CPA

- *Le calcul de Toussaint pour le rectangle minimum*
- *L'approximation de Ritter pour le cercle minimum*

Mohamed Tarek KASSAR

mohamed.kassar@etu.upmc.fr

3674435 - M1 STL

Mars 2017

Plan du rapport :

I. Introduction [1]:

Dans la géométrie algorithmique il existe beaucoup de problèmes tels que le cercle minimum, le polygone convexe, la sphère minimum, le diamètre d'un ensemble de points, le rectangle minimum, etc…. Parmi les motivations importantes qui ont poussé la géométrie algorithmique à se dévoiler on cite le développement des outils de conception assistée par ordinateur, les jeux vidéos (des calculs géométriques en temps réel), les systèmes d'informations géographiques etc...

Dans notre recherche on va s'intéresser à deux solutions de deux problèmes, l'algorithme de toussaint pour résoudre le problème du rectangle minimum, et l'approximation de Ritter pour résoudre le problème du cercle minimum. On va montrer l'efficacité de ces deux solutions en terme de rapidité et de qualité du résultat.

II. Le calcul de Toussaint pour le rectangle minimum [2][3][4]:

1. Description :

Le problème du rectangle minimum consiste à trouver la plus petite boite orientée qui circonscrit un ensemble de points dans R^2.

Il existe une variété de méthodes pour calculer le rectangle ayant l'aire minimum, parmi ces méthodes on a choisi d'implanter, d'expérimenter et de discuter l'algorithme proposé par *Godfried Toussaint*, qui est basé sur la méthode de *rotating calipers* utilisé par *Michael Shamos* pour calculer le diamètre d'un polygone convexe. *Toussaint* a démontré qu'avec cette méthode on peut résoudre plein de problèmes géométriques en

temps linéaire $O(n)$ tels que la distance maximum entre deux polygones, fusion de deux polygones etc…

2. Algorithme de Toussaint:

En considérant le théorème prouvé par *Freeman* et *Shapira* [4] qui dit que: "*Le rectangle ayant l'aire minimale entourant un polygone convexe a un côté colinéaire avec l'un des côtés du polygone*", et qui aide à limiter le nombre de rectangles à vérifier au nombre de côtés du polygone convexe (la solution de ces deux chercheurs était de calculer le rectangle pour chaque côté en $O(n)$, alors pour un total de $O(n^2)$ pour tous les rectangles). Cet algorithme prend en entrée un polygone convexe dont les points sont triés en ordre négatif[1], et retourne le rectangle minimal. L'idée est de simuler deux pieds à coulisses orthogonaux l'un à l'autre, les tourner sur tous les côtés du polygone, et à chaque fois calculer l'aire du rectangle obtenu et prendre le plus petit rectangle à la fin. Pour ce faire, au début on considère les points p_i, p_j, p_k et p_l, tels que chacun de ces points appartient à un des quatre côtés du rectangle minimum, on initialise ces points avec les points ayant les coordonnées x et y minimales et maximales (Figure 01), on trace quatre vecteurs $\vec{v_i}$, $\vec{v_j}$, $\vec{v_k}$ et $\vec{v_l}$ tel que $\vec{v_i}$ passe par le point p_i (resp. j, k et l), et formant un rectangle régulier[2], et orientés d'une manière tel que tous les points seront à droite de chacun parmi ces vecteurs, puis on calcule l'angle minimum θ entre chaque vecteur et le côté suivant du polygone i.e. : $\theta_i = (P_i\vec{P}_{i+1}, \vec{v_i})$ (resp. j, k et l), on tourne les vecteurs avec cet angle (soit θ_j l'angle minimal), on avance le point p_j vers le point p_{j+1} (Figure 02), on calcule l'aire du rectangle obtenu puis on répète la rotation jusqu'à la fin de tous les côtés du polygone.

[1] Sens horaire
[2] Un rectangle non orienté

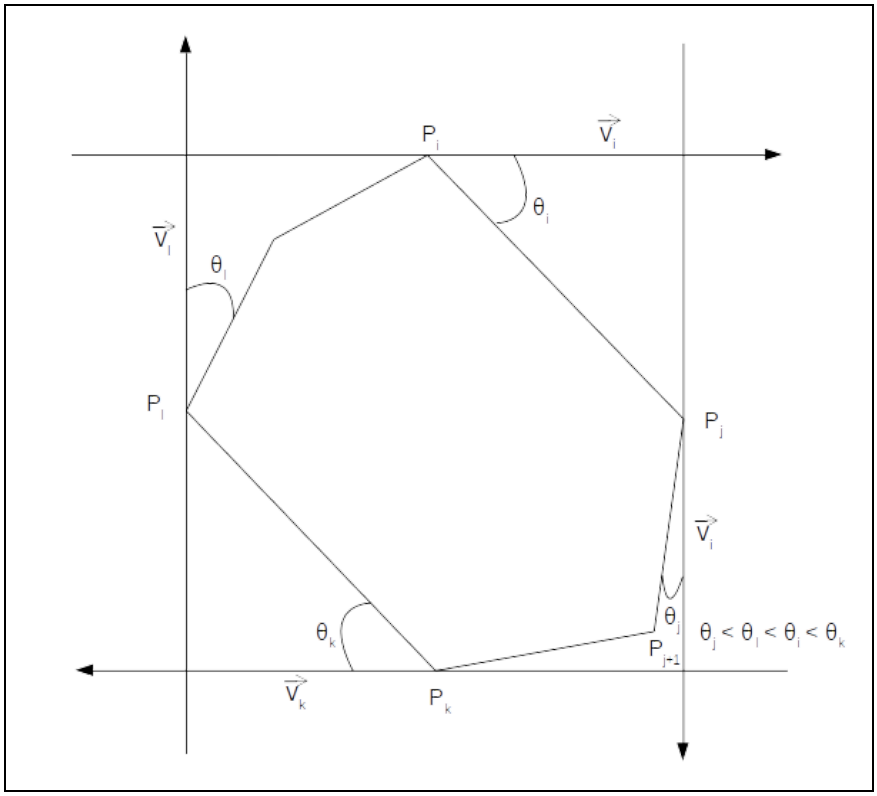

Figure 01 : étape d'initialisation

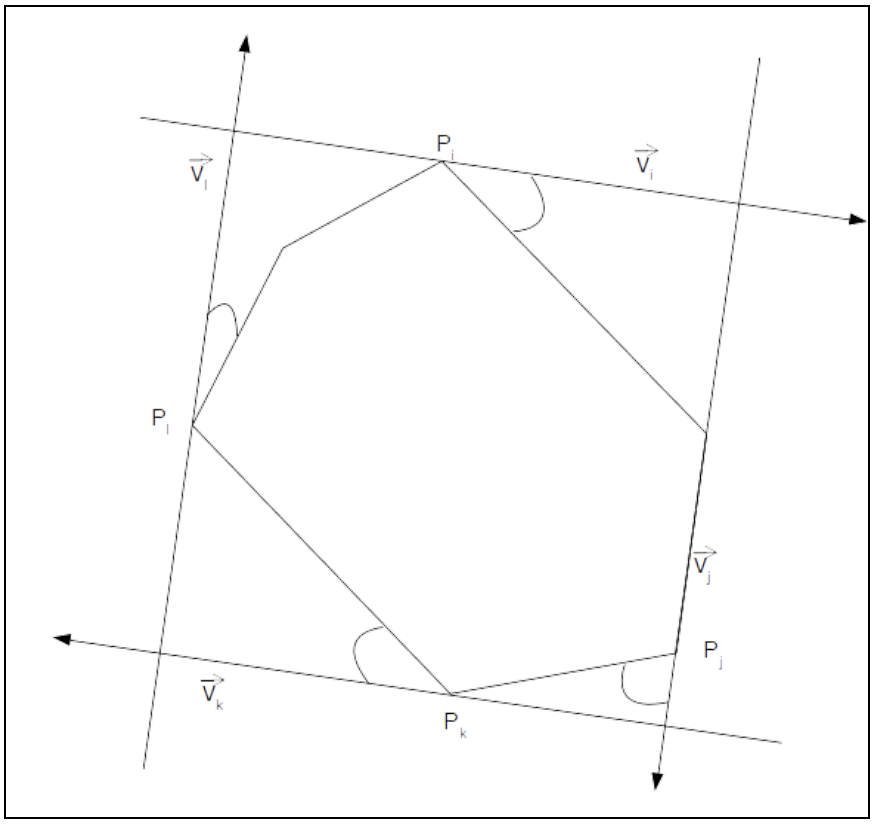

Figure 02 : rotation des vecteurs

3. Etude expérimentale (résultat et discussion) :

Afin d'expérimenter cet algorithme, j'ai utilisé la base de test VAROUMAS qui contient 1663 instances de test, chacune contient 256 points aléatoires, et en se basant sur le rapport de qualité $\frac{aire\ du\ rectangle}{aire\ du\ polygone} - 100\%$, qui sert à comparer les résultats par rapport à un critère fixe et optimal pour chaque instance. la moyenne de la qualité des instances était ≈ 0.252, avec un écart-type de ≈ 0.021 qui signifie que la distribution des qualités est resserrée, en moyenne comprises entre 0.231 et 0.273 (Figure 03), signifiant que l'aire d'un rectangle minimum bat l'aire du polygone convexe de $\approx 25.2\%$. Le temps d'exécution[3] moyen pour une instance était de $\approx 61.2\ \mu s$ sans compter le temps de calcul du polygone convexe et de $\approx 749.1\ \mu s$ en comptant le temps de calcul du polygone convexe.

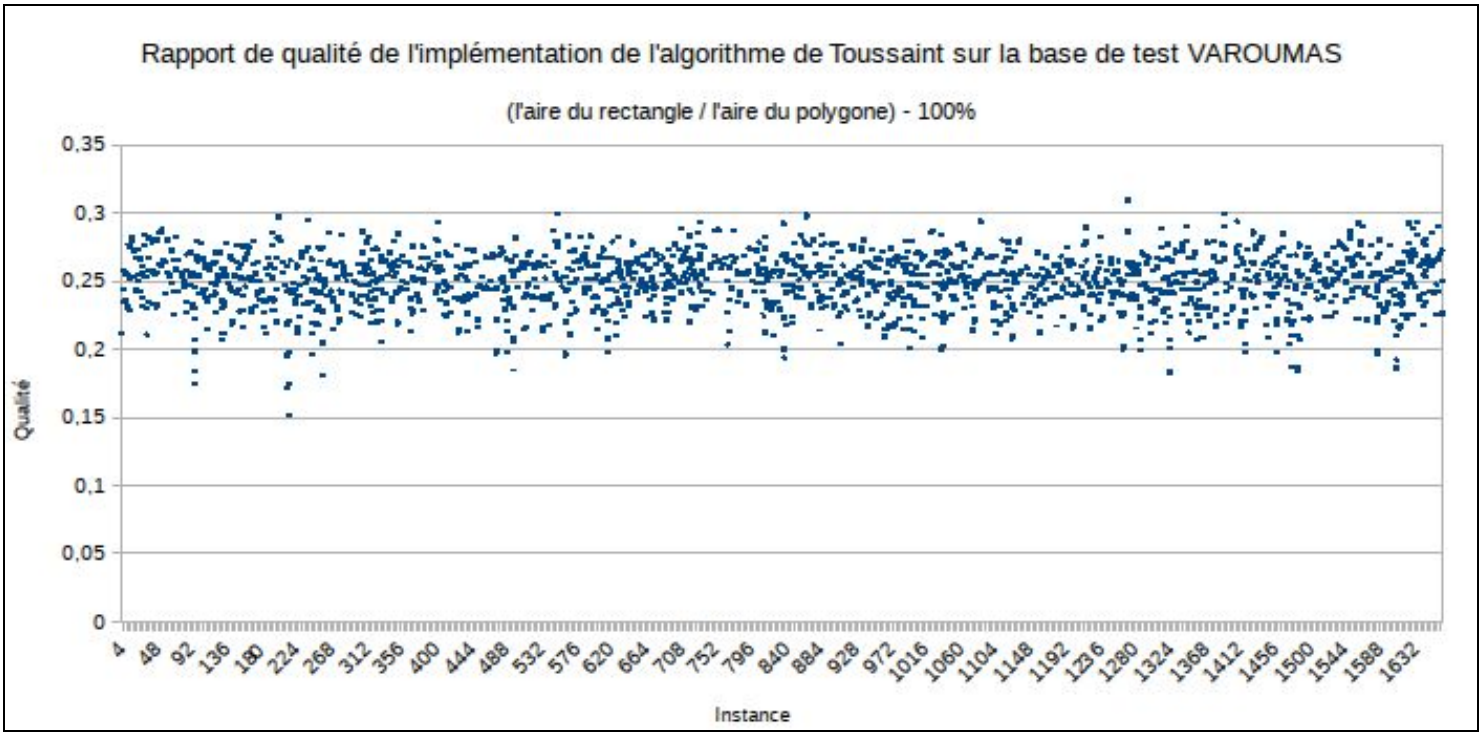

Figure 03 : distribution des qualités pour l'algorithme de Toussaint sur la base de test VAROUMAS

[3] Tests effectués sur une machine : i5 2.3 Ghz, Ram 4 Go

Pour calculer le polygone convexe j'ai utilisé l'algorithme de Graham qui est de complexité de $O(n \cdot Log(n))$, et qui donne en sortie les points du polygone trié.

J'ai rajouté à l'algorithme de Toussaint avant de commencer le traitement une phase d'initialisation qui consiste à associer à chaque point un vecteur partant du point courant vers le point suivant, pour éviter la redondance des calcules des vecteurs.

L'avantage de cet algorithme est le fait que le nombre d'itérations est limité au nombre des côtés du polygone convexe, et avec peu d'opérations pour chacune de ces itérations : la détermination de l'angle minimal parmis quatre angles, le calcule des côtés du rectangle à partir de cinq points déjà calculés, le calcul de l'aire de ce rectangle, et puis la rotation des vecteurs.

Pour implémenter cet algorithme j'ai utilisé le langage Java 8, et pour la visualisation des calculs j'ai utilisé JavaFx 8.

4. Le rectangle minimum en trois dimensions [6][7]:

Il est plus compliqué de calculer le parallélépipède rectangle minimal contenant un nuage de points dans R^3. Selon [6] l'algorithme le plus rapide qui résout ce problème et donne un résultat exact est proposé par Joseph O'Rourke en 1985 qui tourne en temps cubique $O(n^3)$ et difficile à implémenter, il prend en entrée un polyèdre convexe et retourne le pavé minimal circancivant ce polyèdre, cet algorithme est aussi basé sur la méthode du *rotating calipers* mais en terme trois dimensionnel, et est basé sur l'observation de son auteur qui dit : "*Une boîte de volume minimal circonscrivant un polyèdre convexe doit avoir au moins deux faces adjacentes qui contiennent des bords du polyèdre*". Il existe plein d'autres méthodes plus rapide pour résoudre ce problème, mais approximatives, elles ne donnent pas un résultat exact, voir dans [6] pour plus d'informations.

III. L'approximation de Ritter pour le cercle minimum [8][9]:

1. Description:

Le problème du cercle minimum consiste à trouver le plus petit cercle qui entoure un ensemble de points dans R^2. La solution naïve qui donne un résultat exact à ce problème tourne en $O(n^4)$ qui n'est pas pratique pour les systèmes en temps réels ou bien pour un grand nombre de points, qui a poussé les chercheurs à trouver des solutions et des approximations plus efficaces, on trouve parmis ces solutions par exemple des solutions qui proposent des pré-calculs, un filtrage des points internes de l'ensemble pour éliminer les points inutiles.

Comme on a déjà cité précédemment, on va s'intéresser à l'algorithme d'approximation de Ritter, qui est un algorithme glouton (incremental) tourne en $O(n)$, et donne un résultat proche de l'optimum, selon Ritter cet algorithme ne perd que 5% en qualité par rapport au résultat optimal. Pour un calcul aussi rapide que ça on peut négliger cette perte de qualité.

2. Algorithme de Ritter:

L'idée de cet algorithme est très simple, il consiste à chercher approximativement deux points très éloignés dans l'ensemble de points, pour ce faire on prend un point au hasard, on cherche le point le plus éloigné de ce point soit x puis on cherche le point le plus éloigné de x soit y, on initialise par un cercle (δ) qui a le segment $[xy]$ comme diamètre (cette phase d'initialisation joue un rôle important sur la qualité du résultat), on supprime x et y de l'ensemble de points, puis pour chaque point p de l'ensemble on

teste si p appartient à (δ) on le supprime de l'ensemble, sinon on trace un nouveau cercle inclut le cercle (δ) et passe par le point p (pour trouver ce cercle on trace une droite (d) passant par le centre de (δ) et par le point p, soit z le point d'intersection de (d) avec (δ) et le plus loin du p, le nouveau cercle a le segment $[pz]$ comme diamètre (Figure 04)), on supprime p et on met à jour le cercle (δ) par le nouveau cercle, et on répète cette itération jusqu'à la consommation de tous les points.

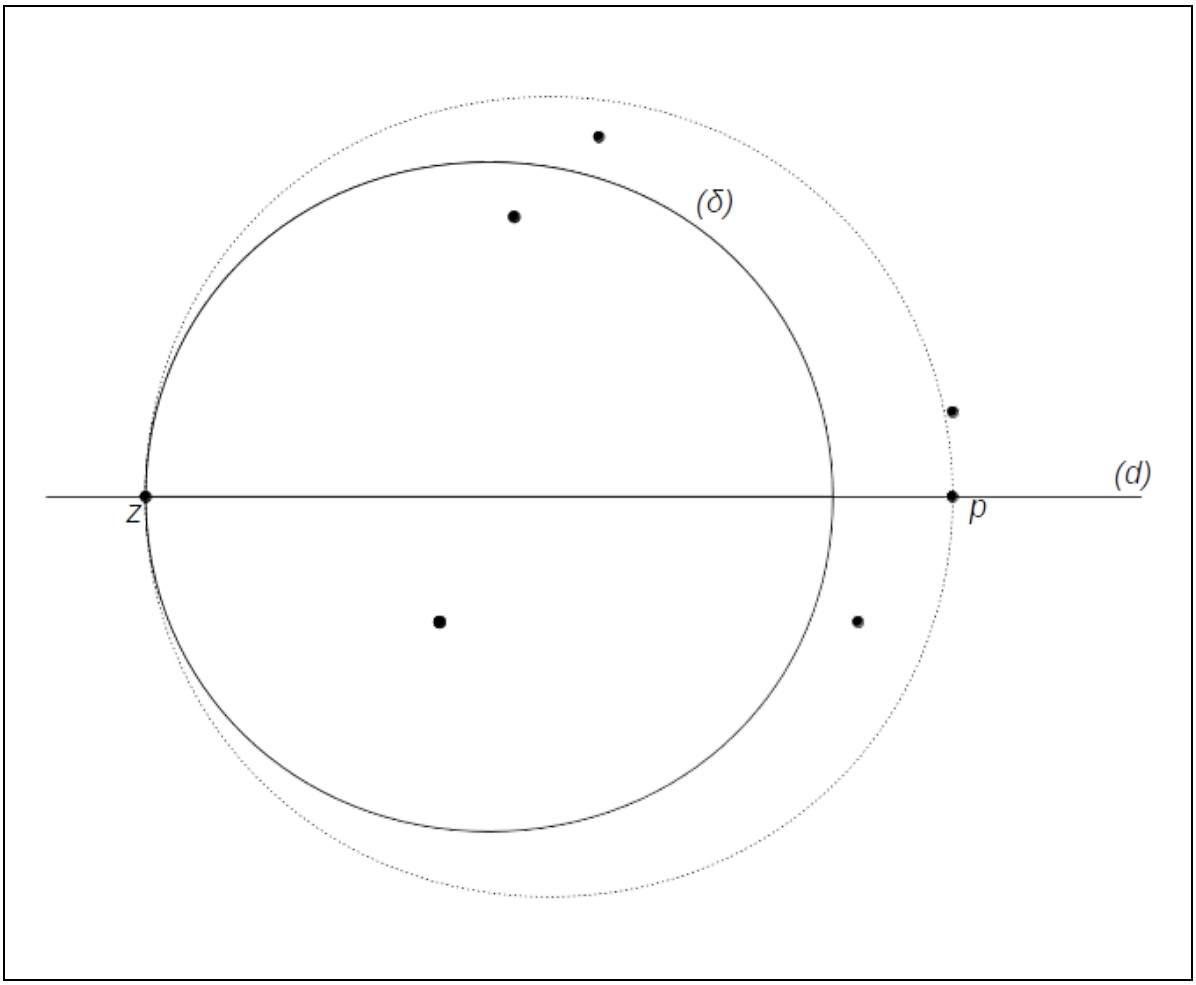

Figure 04 : Incrementation du cercle

3. Etude expérimentale (résultat et discussion) :

Identique à l'algorithme précédent, pour tester la qualité de cette approche j'ai utilisé aussi la base de test VAROUMAS, avec un rapport de qualité défini par $\frac{aire\ du\ cercle}{aire\ du\ polygone} - 100\%$, la moyenne de la qualité des instances était de ≈ 0.198, avec un écart-type de ≈ 0.073, qui est légèrement mauvais, et signifie que le résultat varie selon le test, forcément avec des résultats meilleurs aussi que mauvais, mais avec une petite variation. En moyenne la qualité est comprises entre 0.125 et 0.271 (Figure 05), signifiant que l'aire d'un cercle obtenu par cette approximation bat l'aire du polygone convexe en moyenne de $\approx 19.8\%$. Le temps d'exécution[4] moyen de cet algorithme sur une instance était de $\approx 72\ \mu s$

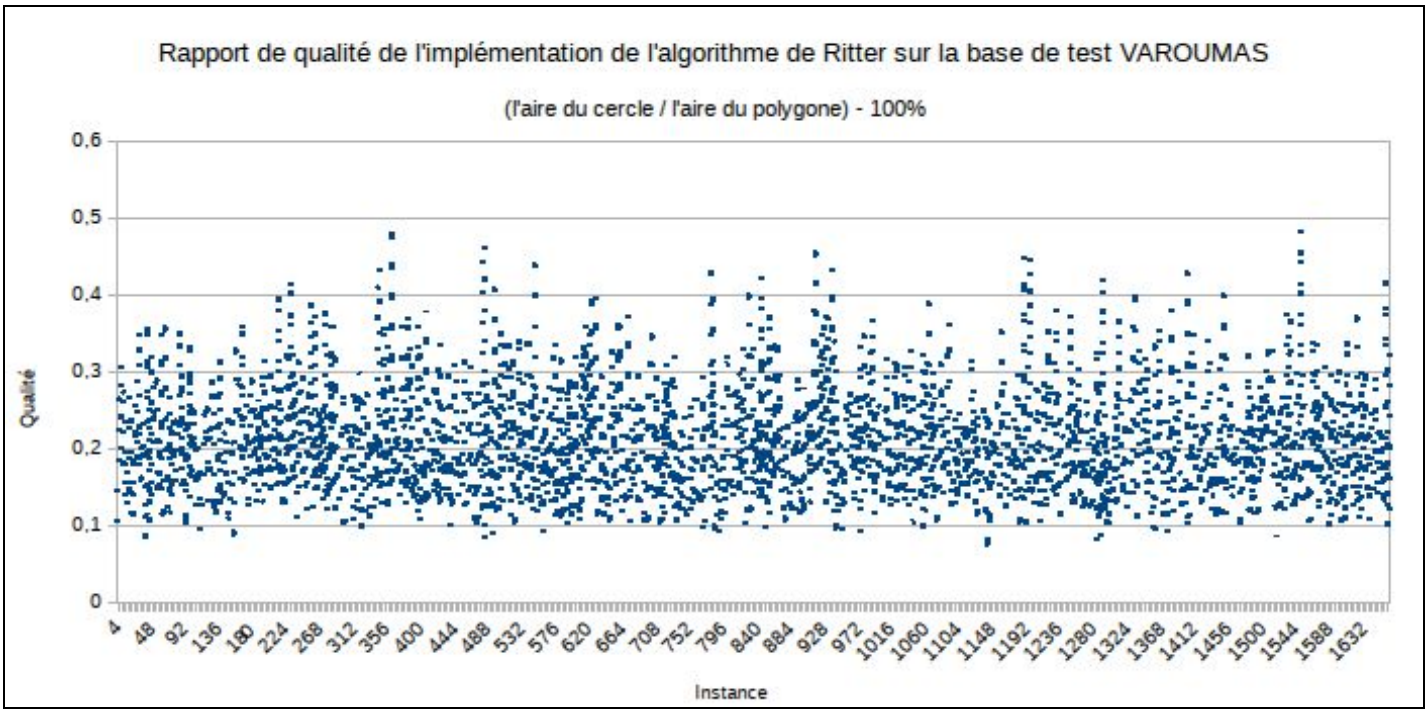

Figure 05 : distribution des qualités pour l'algorithme de Ritter sur la base de test VAROUMAS

[4] Tests effectués sur une machine : i5 2.3 Ghz, Ram 4 Go

4. Le problème de la sphère minimale:

le problème de la sphère minimale contenant un nuage de points dans l'espace est quasiment similaire au problème du cercle minimal, on garde presque les mêmes propriétés, une sphère est comme un cercle définie par un rayon et un point, le vrai changement est seulement qu'on passe à une autre dimension, nos points sont désormais définis dans R^3, par conséquent la formule de calcul de la distance entre les points change, alors les algorithmes en 2D sont toujours valables en 3D mais juste avec une petite conversion, alors on garde la complexité des anciens algorithmes.

IV. Conclusion:

Ce projet a permis de décrire le problème du cercle minimum et le problème du rectangle minimum, et de présenter et d'étudier deux solutions efficaces de ces deux problèmes, l'algorithme de Toussaint pour resoudre le probleme du rectangle minimum et l'approximation de Ritter pour resoudre le probleme du cercle minimum, On a bien remarqué le bon rapport rapidité / qualité de ces deux solutions.

L'exploit de telles solutions rapides et efficaces, est très important vu les besoins nécessités par divers domaines.

V. References:

[1]. “Computational geometry”, Wikipedia. 16-Mars-2017.

[2]. Toussaint, Godfried T. (1983). "Solving geometric problems with the rotating calipers". MELECON '83, Athens.
Lien : https://www.cs.swarthmore.edu/~adanner/cs97/s08/pdf/calipers.pdf

[3]. “Rotating calipers”, Wikipedia. 17-Mars-2017.

[4]. The infinite loop, “ Computing oriented minimum bounding boxes in 2D”, 23-jan-2014.
Lien: https://geidav.wordpress.com/2014/01/23/computing-oriented-minimum-bounding-boxes-in-2d/.
[visité: 19-Mars-2017]

[5]. H. Freeman and R. Shapira, “Determining the minimum-area encasing rectangle for an arbitrary closed curve”, Comm. A.C.M. , Vol. 18, July 1975, pp. 409-413.

[6]. Geometry Algorithms, “Bounding Containers for Polygons, Polyhedra and Point Sets (2D & 3D)”. Lien : http://geomalgorithms.com/a08-_containers.html
[visité: 20-Mars-2017]

[7]. “Minimum bounding box algorithms”, Wikipedia. 20-Oct-2015.

[8]. BUI-XUAN Binh-Minh, “Cours 01 CPA, Introduction à la géométrie algorithmique”, UPMC, jan-2017.

[9]. Marble Mice, “Bounding Circles – Algorithm Comparision”, 04-sep-2006.
Lien : https://marblemice.wordpress.com/2006/09/04/bounding-circles-algorithm-comparision/
[visité: 21-Mars-2017]