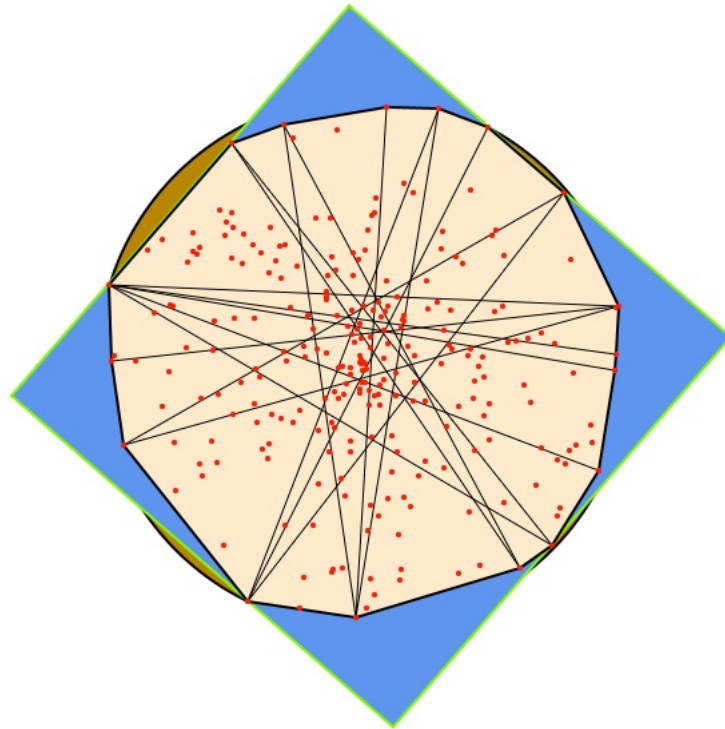


Algorithme Toussaint résolvant le problème du rectangle minimum

Couassi-blé Yannick-Alain



Rapport de recherche
Master Informatique



Science et Technologie du Logiciel
Université Pierre & Marie Curie
Paris

Mars 2017

L'algorithme Toussaint résolvant le problème du rectangle minimum

Couassi-Blé Yannick-Alain

Conception et Pratique de l'Algorithmique – Science et Technologie
du Logiciel(STL)

Rapport de recherche – 26 Mars 2017 – 18 pages

Résumé : Étant donné un ensemble fini Δ de points de coordonnées dans \mathbb{R}^2 , nous proposons dans ce rapport l'algorithme Toussaint déterminant le rectangle minimum couvrant tous ces points. Cet algorithme n'est pas dans le pire des cas le plus optimal mais nous partons d'approche simple et il s'adapte bien sur des distributions pas énormes.

Mots-clés : Toussaint, Ritter, enveloppe convexe, géométrie algorithmique

Table des matières

Résumé	iii
1 Introduction	2
2 Définitions, notations, préliminaires géométriques	3
3 Présentation et implémentations des Algorithmes	5
3.1 Algorithme Toussaint	5
3.2 Algorithme Ritter	8
4 Expérimentations et résultats	10
4.1 Temps d'exécution	10
4.2 Couverture d'Aire et qualité	12
5 Discussion	15
5.1 Extension en 3D	16
6 Conclusion	17
Bibliography	18

Table des figures

3.1	Enveloppe convexe et paires antipodales	6
3.2	Rectangle minimum obtenu	7
3.3	Δ d'entrée	8
3.4	Cercle minimum obtenu	9
4.1	Temps d'exécution Cercle(violet) vs Rectangle (vert)	11
4.2	Couverture en aire rectangle(bleu) vs polygone(vert)	13
4.3	Couverture en aire cercle(violet) vs polygone(vert)	13
4.4	Couverture en aire cercle(violet) vs polygone(vert) vs rectangle(bleu)	13
4.5	Rapport qualité rectangle(vert) vs cercle(violet)	14

Chapitre 1

Introduction

Ce rapport de recherche intitulé **l’algorithme Toussaint résolvant le problème du rectangle minimum** s’effectue dans le cadre de l’UE Conception et Pratique de l’Algorithmique. Le thème de cette recherche est un des classiques des problèmes de la géométrie algorithmique. Cependant notre objectif est de proposer une implémentation de l’algorithme Toussaint et l’algorithme de Ritter qui quant à lui donne une approximation du problème du cercle minimum. À la suite des présentations et implémentations de ces algorithmes , une expérience de benchmarking à partir de la base VARAMOUS sera effectuée. L’objectif de cette expérience est d’évaluer la qualité en tant que conteneur du rectangle et du cercle, mais on introduira également un polygone convexe qui sera comme un témoin de cette expérience. Concernant le polygone convexe nous avons choisit volontairement d’implémenter l’algorithme de Graham qui résout le problème de l’enveloppe convexe étant donné une distribution de points. Enfin, nous ferons une discussion et nous nous interrogerons sur de possibles extensions ou adaptation de ces algorithmes vers le merveilleux monde de la 3D .

Remarque : Sauf mention contraire tout notre raisonnement se fera dans le plan 2D donc des points de coordonnées dans R^2 .

Chapitre 2

Définitions, notations, préliminaires géométriques

Nous notons Δ un ensemble fini de points de coordonnées dans R^2 , n le cardinal de Δ .

a et b respectivement le coefficient directeur et l'ordonnée à l'origine d'une droite d'équation de la forme $y = ax + b$.

NB : Dans cette section on admettra les propriétés .

Rappel : *Aire cercle* $= \pi.R^2$, R le rayon du cercle

Soit le polygone convexe formé par les points de coordonnées $(X_1, Y_1), \dots, (X_n, Y_n)$

$$\text{Aire polygone} = \frac{1}{2} \begin{vmatrix} X_1 & Y_1 \\ \cdots & \cdots \\ X_n & Y_n \\ X_1 & Y_1 \end{vmatrix} = \frac{1}{2} [(X_1.Y_2 + \dots + X_n.Y_1) - (Y_1.X_2 + \dots + Y_n.X_1)]$$

Définitions :

Polygone convexe : polygone tel que tout segment joignant deux points du polygone appartient au polygone.

Côté d'un polygone convexe : segment maximal du polygone tel que tout autre point du polygone appartient au même demi-plan défini par le segment

Une paire A,B d'un ensemble Δ de points est dite **antipodale**, s'il existe 2

droites parallèles passant par A et B telles que la bande du plan entre ces deux droites contient tous les points de Δ .

AABB : Axis Aligned Bounding Box

OBB : Oriented Bounding Box

Propriétés :

Propriété 1 : il y a au plus $2n$ de paires antipodales d'un ensemble de n points.

Propriété 2 : Dans R^2 , il existe un et un seul cercle passant par 3 points non-colinéaires.

Propriété 3 : Soient D et D' deux droites d'équations respectives $y = ax + b$ et $y = a'x + b'$ alors :

$$D \parallel D' \Leftrightarrow a = a'$$

$$D \perp D' \Leftrightarrow a \cdot a' = -1$$

Chapitre 3

Présentation et implémentations des Algorithmes

3.1 Algorithme Toussaint

Le but premier de cet algorithme est comme déjà mentionné dans l'introduction, de trouver le rectangle minimum qui encapsule tous les points de Δ . Plusieurs techniques d'approche à ce problème existent. Cependant un algorithme naïf consisterait par exemple à rechercher directement les quatre points les plus extrêmes c'est à dire situés à l'EST, l'OUEST, NORD, SUD et ensuite tracer les droites deux à deux parallèles passant par ces points. Cet algorithme n'est pas du tout optimal et un gros inconvénient est qu'il fournit uniquement un rectangle du type AABB donc avec une aire pas minimal. Alors qu'un OBB est encore meilleur.

Etant donné ce constat on cherchera une autre approche qui nous donnera un OBB mais aussi le minimum possible. Au lieu de travailler directement sur l'ensemble Δ , on partira d'une enveloppe convexe. Cette idée se justifie par le théorème suivant prouvé en 1975 par Freeman et Shapira [1] :

Théorème : le rectangle minimum contenant un ensemble de points à un côté parallèle avec l'un des côtés de l'enveloppe convexe de ces points.

Grâce à ce théorème le nombre de candidat de OBB est considérablement réduit dans notre algorithme. On se munira enfin d'un dernier outil qui est la notion de

paire antipodale definit plus haut dans le chapitre 2. Ci dessous le pseudo-code et différentes lignes de notre algorithme :

1. On réalise l'enveloppe convexe E de Δ , $\text{Card}(E)=k$ et on calcul ses paires antipodales

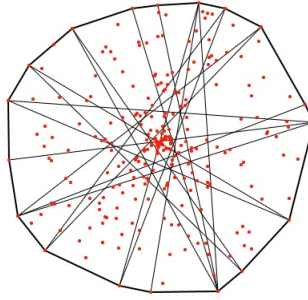


FIGURE 3.1 – Enveloppe convexe et paires antipodales

2. Soit $AireMin \leftarrow +\infty$, 2 tableaux de 4 points : *rectangleTemporaire* et *rectangleMinimum*
3. On parcourt l'ensemble des points des paires antipodales en prenant un P_i quelconque, $1 < i < k$
4. On prend P_{i+1} le successeur de P_i dans E
5. On calcul le a et b de la droite $(P_i P_{i+1})$
6. On considère un b_0 intermédiaire, $b' \leftarrow +\infty$ et un point T
7. On parcourt l'ensemble des points avec qui P_i forme une paire antipodales en prenant T_i quelconque
8. On calcule le b_0 de la droite passant par T_i
9. On compare b_0 à b pour déterminer le côté du demi-plan delimité par $(P_i P_{i+1})$ de la droite passant par T_i , on actualise b' et T en faisant $b' \leftarrow b_0$, $T \leftarrow T_i$
10. On considère un $b_g \leftarrow +\infty$, $b_d \leftarrow -\infty$, b_1 servant d'intermédiaire comme b_0 et 2 points G et D
11. On parcourt E en prenant S_i

12. On calcul le b_1 de la droite perpendiculaire passant par S_i
13. On compare b_1 à b_g et b_d pour déterminer le coté des droites passant respectivement par G et D , Cependant soit $G \leftarrow S_i$ soit $D \leftarrow S_j$, $i \neq j$
14. On calcul à présent les 4 points A, B, C, D d'intersections formé par les droites ci dessus
15. On affecte à $rectangleTemporaire \leftarrow A, B, C, D$
16. Si l'aire de $rectangleTemporaire$ est inférieur à $AireMin$ alors $rectangleMin \leftarrow rectangleTemporaire$
17. On reitère l'étape 3 jusqu'à parcourir tous les points
18. On obtient alors $rectangleMin$ qui est le rectangle minimum de Δ

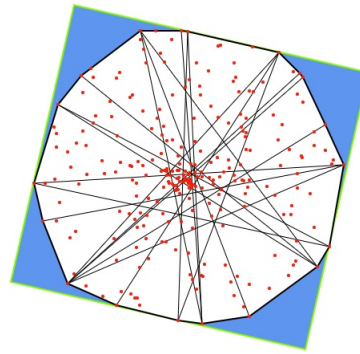


FIGURE 3.2 – Rectangle minimum obtenu

Complexité : Pour évaluer la complexité de l'algorithme Toussaint on prend en compte la complexité du précalcul effectué, c'est à dire celle de l'enveloppe convexe et ensuite on ajoute la complexité des autres traitements. Dans notre recherche nous avons choisi d'utiliser le parcours de Graham, cependant sa complexité peut à première vue sembler être du $O(n^2)$ mais il est dominé par un tri par pixel donc sa complexité globale est de l'ordre de $O(n \log(n))$. Ensuite l'étape 4 qui constitue la boucle globale est de l'ordre de $O(k)$, l'étape 7 et 11 de même, mais comme ils sont une imbrication de la 4, la complexité totale de cette partie est $O(k^2)$. **Conclusion** la complexité de l'algorithme Toussaint est dans le pire des cas $O(n \log(n) + k^2)$.

3.2 Algorithme Ritter

L'algorithme de Ritter est une technique d'approximation pour le problème du cercle minimum contenant Δ . Cet algorithme est qualifié d'incrémental car il repose sur le principe suivant : si un cercle ne couvre pas tous les points, on l'agrandit pour couvrir l'ancien cercle, plus au moins un nouveau point. Dans la section suivante nous donnerons un pseudo-code.

Soit $\Delta = p_1, \dots, p_n$ un ensemble de départ d'une distribution de n points de coordonnées dans \mathbb{R}^2 dont on recherche à déterminer le cercle minimum contenant ces points.

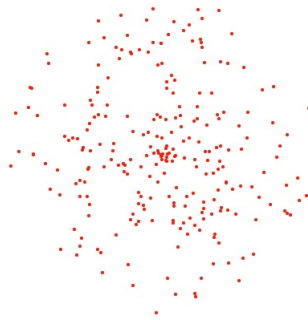


FIGURE 3.3 – Δ d'entrée

1. On prend un point $P_{init} = p_i$ aléatoire de Δ , $i = 1 \dots n$
2. $\forall p_j$ de Δ , $j = 1 \dots n$ et $i \neq j$ On détermine $P_{max} = \text{Max}(\text{distance}(P_{init}, p_j))$
3. $\forall p_j$ de Δ , $j = 1 \dots n$ et $i \neq j$ On détermine $Q_{max} = \text{Max}(\text{distance}(P_{max}, p_j))$
4. Soit le cercle (Γ) de centre C de rayon $[P_{max}Q_{max}]$
5. On retire ensuite P_{max} et Q_{max} de Δ
6. Tant qu'il y'a des p_i dans Δ on choisit un point quelconque $S = p_i$
7. Si $S \in$ au disque fermé formé par (Γ) on retire S de Δ et on reitère l'étape précédente
8. Sinon on considère la droite (SC) , elle coupe (Γ) en 2 points. On considère T de distance maximal à S .

9. On considère un nouveau point C' milieu de $[ST]$ définie par les coordonnées barycentriques : $C' = \alpha.C + \beta.S$

Détails des expressions :

$$\alpha = \frac{|C'S|}{|CS|} \text{ et } \beta = \frac{|C'C|}{|CS|}$$

$$|C'S| = \frac{|CS|+r}{2}, r \text{ est le rayon de } (\Gamma)$$

$$|C'C| = \frac{|CS|-|CS+r|}{2}$$

10. On remplace (Γ) par (Γ') le cercle de centre C' de rayon $[C'T]$
11. On repète l'étape 6 tant qu'il reste des p_i dans Δ

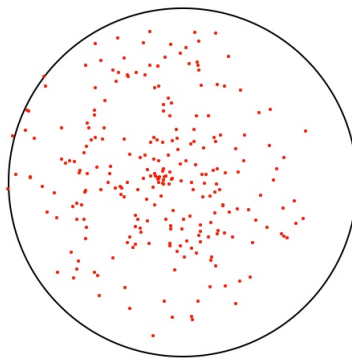


FIGURE 3.4 – Cercle minimum obtenu

12. On obtient alors le cercle final (Γ') suivant couvrant tous les points.

Complexité : L'algorithme de Ritter s'exécute de manière quasi linéaire. Sa complexité en temps dans le cas le pire est donc $O(n)$. Elle est dominée par les parcours indépendants des boucles sur l'ensemble des points (pas d'imbrication).

Chapitre 4

Expérimentations et résultats

Nous réaliserons une étude qualitative du temps d'exécution et de la qualité de l'aire couvert par le rectangle minimum, le cercle minimum vis à vis de l'enveloppe convexe. On fera un croisement de ces données pour plus de crédibilité de sens de jugement.

Test : Les tests sont effectués sur une machine à Processeur Intel Core i5 une Ram de 4 Go cadencée à 1333 MHz DDR3. On utilise la base VAROUMAS qui contient 1664 instances et chaque instance contient 256 points. La qualité en tant que conteneur sera établie par la formule suivante :

$$qualité_1 = \frac{aire\ rectangle}{aire\ polygone} - 100\%$$

$$qualité_2 = \frac{aire\ cercle}{aire\ polygone} - 100\%$$

Remarque : Les différents graphes sont tracés de la manière suivante : en abscisse le numéro des instances et en ordonnée soit le temps d'exécution soit l'aire en fonction de la section

4.1 Temps d'exécution

Nous avons évalué les temps d'exécution en nanosecondes pour avoir des données plus significatives. Sur le graphe 4.1 ci dessous, on remarque bien que le temps d'exécution de l'algorithme Toussaint pour le rectangle minimum est plus important que celui de Ritter. En moyenne l'algorithme Toussaint s'exécute sur une instance en

182934.9055354994 ns, tandis que Ritter en 63277.99879663056 soit environ 3 fois plus vite. Cet résultat est en adéquation avec les estimation théorique établie dans le chapitre précédent on a bien Ritter qui est quasi linéaire et Toussaint de l'ordre de $O(n \log(n) + k^2)$ ou k est le cardinal de l'enveloppe convexe.

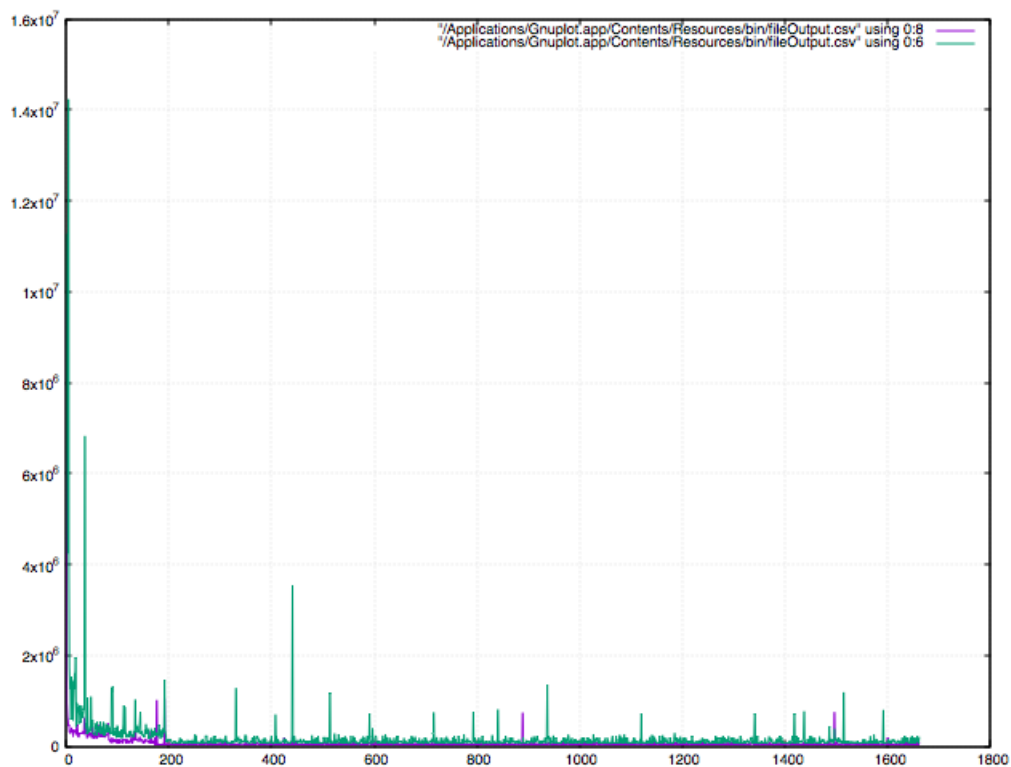


FIGURE 4.1 – Temps d'exécution Cercle(violet) vs Rectangle (vert)
[fig-benchTemps]

4.2 Couverture d'Aire et qualité

Le graphe 4.4 superpose les graphes 4.2 et 4.3 pour en faire une analyse croisée. Il montre bien que l'aire couverte par le rectangle est largement inférieure par rapport à celle du cercle sur l'ensemble des instances. $AirePolygone \leq AireRectangle \leq AireCercle$.

Cependant à présent intéressons nous à leur qualité.

À première vue du graphe 4.5 on fait la même observation que sur le graphe 4.4 ce qui est tout à fait logique. Néanmoins quand on évalue la moyenne de ces différents taux on obtient pour la qualité du rectangle $0.25314300189699185 \approx 25\%$ et la qualité du cercle $0.2578832139277609 \approx 25\%$.

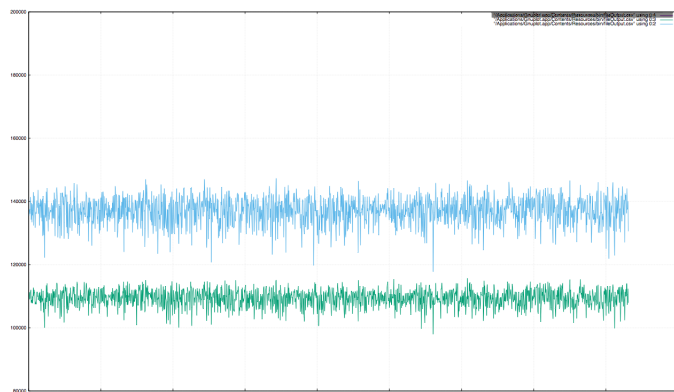


FIGURE 4.2 – Couverture en aire rectangle(bleu) vs polygone(vert)

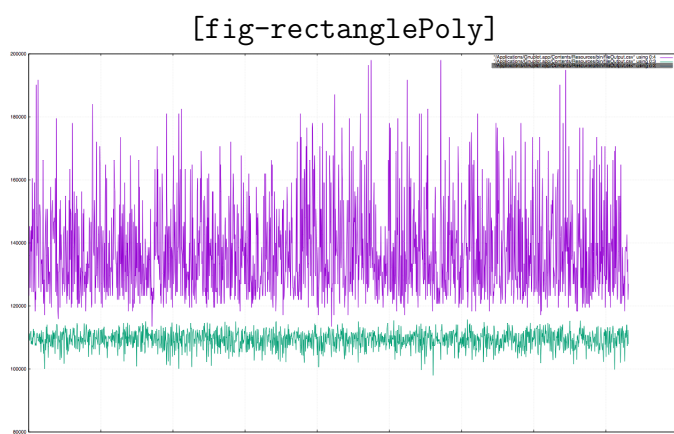


FIGURE 4.3 – Couverture en aire cercle(violet) vs polygone(vert)

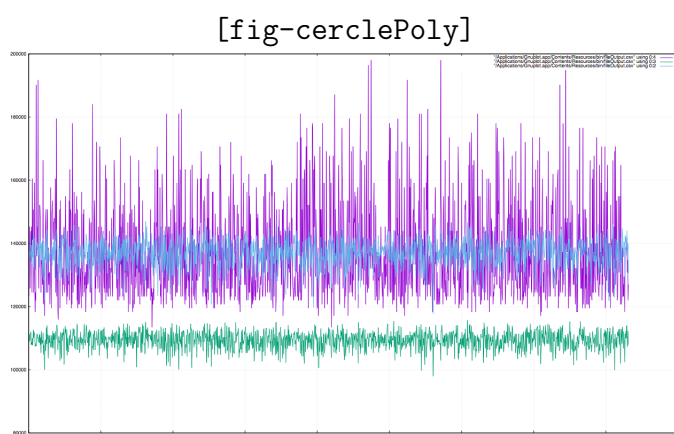


FIGURE 4.4 – Couverture en aire cercle(violet) vs polygone(vert) vs rectangle(bleu)

[fig-analyseCroisee]

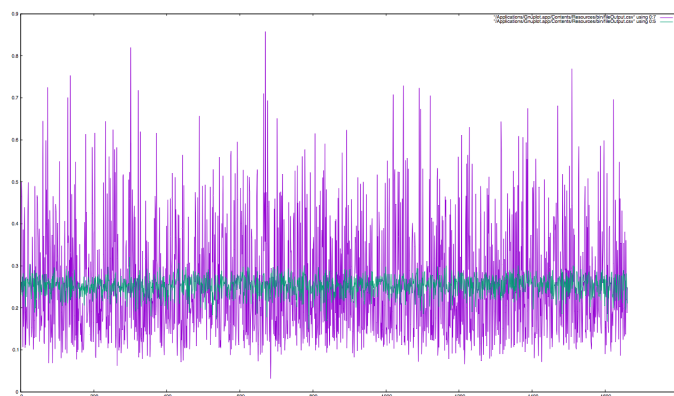


FIGURE 4.5 – Rapport qualité rectangle(vert) vs cercle(violet)
[fig-qualite]

Chapitre 5

Discussion

Durant cette recherche notre méthode pour la couverture du rectangle minimum n'a pas été immédiate à implementer. Il a fallu faire quelques choix pour rendre l'algorithme optimal mais aussi avec un temps d'exécution convenable. Dans ce algorithme nous effectuons des rotations implicitement en manipulant les équations de droites. C'est à dire au lieu de rechercher un angle de deviation minimal (pente) pour se positionner sur le bon coté de l'enveloppe, rotating calipers [3], on bascule d'un point de l'enveloppe a son successeur. Le coût de cet algorithme est majoritairement influencé par les précalculs sur Δ pour obtenir l'enveloppe convexe, car l'ensemble réduit à l'enveloppe est de tailles négligeables devant Δ . Dans nos expériences ont a travaillé sur des instances avec $n=256$ points et on obtient en moyenne une enveloppe de 15 points.

Quant à l'algorithme de Ritter il n'est pas optimal, mais il est un bon compromis entre l'implémentation naïve qui se fait en $O(n^4)$. Il à l'avantage d'être simple à implémenter. De ce fait le choix du conteneur depend de la plateforme d'exécution et des objectifs de l'utilisation d'un conteneur. Par exemple sur des peripheriques portables ou le temps de calcul n'est pas important l'idéal serait d'implementer l'agorithme de Ritter. Dans les années 1970-1980 presque tous les objets dans les jeux vidéos etaient modelisés par des cercles car simples à manipuler. Mais sur d'autres machines ayant de puissants processeurs et ou la précision du resultat est de taille, on devrait privilegier Toussaint vis à vis de Ritter.

5.1 Extension en 3D

L'extension de ces différentes problématiques dans R^3 seraient, pour Ritter [2] de trouver la sphère minimum contenant tous les points. Cet algorithme se comporterait sur le même principe que celui que nous avons proposé dans le chapitre 3. Il n'y vraisemblablement aucune grande modification, on considère juste pour les différents points une troisième composante Z pour la profondeur, et au lieu de retourner un cercle on retourne une sphère (coordonnée du centre (x,y,z) et le rayon R).

Pour Toussaint cet algorithme ne s'adapte pas, une généralisation à été faite par O'Rourke et s'exécute en $O(n^3)$.

Chapitre 6

Conclusion

Les études théoriques et expérimentales que nous avons réalisées n'ont pas été en contradiction. Des trois conteneurs que nous avons eu à manipuler le meilleur est clairement l'enveloppe convexe , mais il n'a pas été au coeur de notre recherche. On pourra conclure donc que l'algorithme Toussaint resolvant le probleme du rectangle minimum est meilleur que le cercle de Ritter.

Bibliographie

- [1] H. Freeman, R. Shapira (1975), *Determining the minimum-area encasing rectangle for an arbitrary closed curve*, Communications of the ACM, **18 Issue 7**, pp 409–413.
- [2] J. Ritter (1990), **An efficient bounding sphere** .In **Andrew S. Glassner, editor, Graphics Gems. Academic Press**, Boston, MA
- [3] Godfried Toussaint (1983), *Solving Geometric Problems with the Rotating Calipers* , Proceedings of IEEE MELECON'83,Athens, Greece .