Construction de l'enveloppe convexe dans le plan par balayage selon un rectangle

B. Adam, P. Kauffmann, D. Schmitt, and J.-C. Spehner *

RÉSUMÉ : *Nous introduisons une nouvelle méthode de balayage qui consiste à balayer le plan par un rectangle et nous utilisons cette technique pour construire l'enveloppe convexe d'un ensemble de points du plan. Nous obtenons ainsi un algorithme dont les performances pratiques sont comparables à celles des algorithmes les plus efficaces.*
MOTS-CLÉS : *Enveloppe convexe plane, balayage par rectangle, géométrie algorithmique.*

1 Introduction

La construction de l'enveloppe convexe d'un ensemble S de n sites du plan est l'un des premiers problèmes étudié en géométrie algorithmique et de nombreux algorithmes optimaux en $O(n \log n)$ ont été proposés [Gra72], [PH77], [Pre79], ... Kirkpatrick et Seidel [KS86] ont donné un algorithme en $O(n \log h)$ où h est le nombre de sommets de l'enveloppe convexe. Akl et Toussaint [AT78] ont montré comment accélérer l'algorithme de Graham en supprimant lors d'un premier parcours de S, tous les sites qui appartiennent à l'intérieur du polygone formé par les points de S d'abscisses et d'ordonnées maximales et minimales. Une implémentation efficace de cette méthode a été décrite par Bhattacharya et Toussaint [BT83]. Borgwardt, Gaffke, Jünger et Reinelt [BGJR91] ont montré comment tout algorithme optimal de construction d'enveloppe convexe dans le plan peut être rendu linéaire en moyenne lorsque les sites sont répartis quasi-uniformément dans le carré ou le cercle unité.

L'algorithme que nous proposons ici utilise l'idée intuitive suivante : les sites qui sont proches de la frontière de la boîte englobante de S ont une forte probabilité d'appartenir à la frontière de l'enveloppe convexe. Nous proposons donc de construire cette enveloppe convexe en balayant le plan par un rectangle décroissant.

2 Le balayage du plan par un rectangle

Soient s_E et s_O (resp. s_N et s_S) les sites de S d'abscisses (resp. d'ordonnées) maximale et minimale et soient x_E et x_O (resp. y_N et y_S) ces abscisses (resp. ces ordonnées). Deux au moins parmi les sites s_E, s_O, s_N et s_S sont distincts.

Le rectangle R_0 qui passe par ces sites et dont les côtés sont horizontaux ou verticaux est appelé *la boîte englobante* de S.

Pour tout site s de S de coordonnées (x, y), soit $\Delta_s = \min\{x_E - x, x - x_O, y_N - y, y - y_S\}$ la distance de s à R_0.

Les sites de S sont alors triés par distances Δ croissantes.

*Laboratoire MAGE, Université de Haute-Alsace, 68093 Mulhouse, France
[B.Adam|P.Kauffmann|D.Schmitt|JC.Spehner]@univ-mulhouse.fr

Le traitement des sites selon cet ordre revient à balayer le plan par un rectangle R_t de position initiale R_0 et tel que, à chaque instant t, tous les points de R_t soient à la même distance t de R_0.

Lorsque t croît de 0 à une valeur limite qui sera précisée plus tard, le rectangle R_t balaye l'intérieur du rectangle R_0.

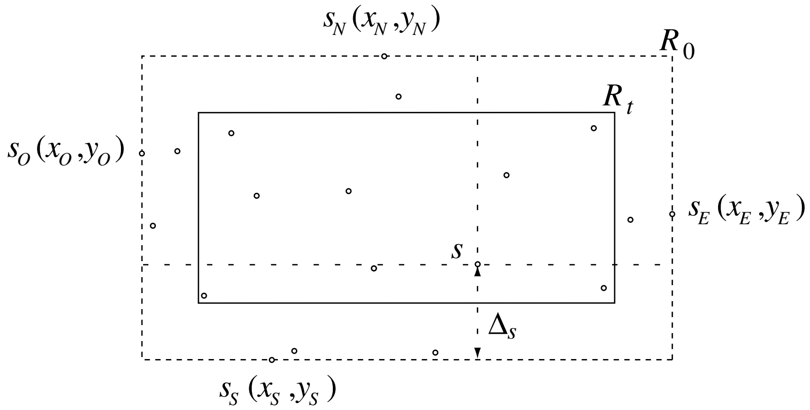

Figure 1: La distance Δ_s du site s à R_0 est égale à $y - yS$. s sera donc balayé par le « côté sud »du rectangle R_t.

3 Les côtés coupants

A un instant t du balayage, soit T l'ensemble des sites qui ont été traités et soit $conv(T)$ leur enveloppe convexe. T est composé des sites de S qui sont à l'extérieur de R_t et éventuellement d'un ou plusieurs sites qui sont sur R_t.

Un *côté* c de la frontière $\Delta(conv(T))$ de $conv(T)$ est dit *coupant* si l'intersection de c avec l'intérieur de R_t n'est pas vide. Alors $\Delta(conv(T))$ admet au plus 4 côtés coupants et tout côté coupant fermé coupe deux côtés consécutifs de R_t.

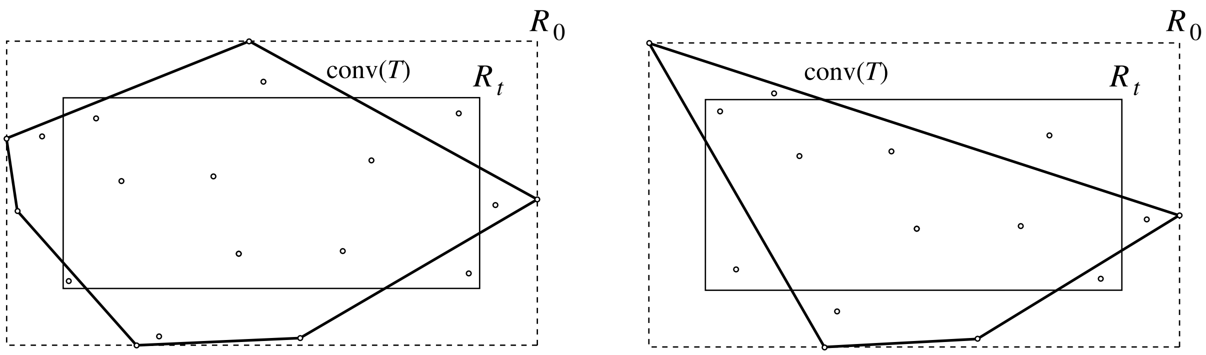

Figure 2: Chaque côté coupant « découpe un coin » du rectangle de balayage R_t.

Pour tout site s de $S \setminus T$, un *côté* c de $\Delta(conv(T))$ est dit *visible* de s si s et $conv(T)$ sont de part et d'autre de la droite supportante de c et c est dit *semi-visible* de s si s appartient à cette droite.

Théorème 1 *Proposition.- Un site s de $S \setminus T$ est extérieur à $conv(T)$ si, et seulement si, il existe un côté coupant visible de s et, dans ce cas, ce côté coupant est unique.*

Preuve.- Si $conv(R_t)$ est l'ensemble des points du plan qui sont sur R_t et à l'intérieurs de R_t, $conv(R_t) \setminus conv(T)$ est formé d'au plus 4 « coins » car les extrémités d'un côté coupant sont extérieurs à R_t ou sur R_t.

Tout site s de $S \setminus T$ qui n'appartient pas à $conv(T)$ appartient alors à un des « coins » de $conv(R_t) \setminus conv(T)$ et à un seul, ce qui prouve à la fois l'existence et l'unicité des côtés coupants visibles de s. □

Il résulte de cette proposition que, pour savoir si un site s non encore balayé est à l'extérieur de l'enveloppe convexe courante $conv(T)$, il suffit de le localiser par rapport aux côtés coupants. Il faut donc connaître ces côtés à chaque instant du balayage et pouvoir les supprimer lorsqu'il ne sont plus coupants. De ce fait on associe à chaque côté coupant c un point limite l qui est le dernier point de c qui sera balayé par le rectangle. l est le point de c qui est équidistant des deux côtés de R_t coupés par c, ce qui permet de déterminer ses coordonnées.

Pour tout point limite l on note Δ_l la distance de l à R_0.

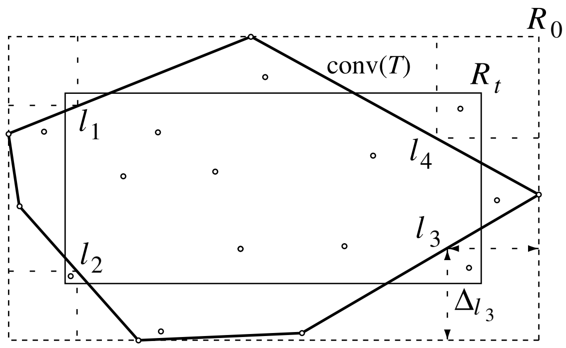

Figure 3: Les points limites des quatre côtés coupants de $\Delta(conv(T))$.

4 Les événements du balayage et l'algorithme de construction

L'enveloppe convexe courante et l'ensemble des côtés coupant ne peuvent être modifiés que lorsque le rectangle de balayage rencontre soit un site soit un point limite. Un tel instant t est appelé un *événement du balayage*.

4.1 L'événement site

Soit s le site balayé à l'instant t et T l'ensemble des sites déjà traités.

S'il existe un côté coupant c visible de s, il faut

- supprimer la ligne polygonale P formée par tous les côtés de $conv(T)$ visibles ou semi-visibles de s ;
- créer deux côtés c_1 et c_2 pour relier s aux deux extrémités de P ;
- déterminer le nouveau côté coupant (avec son point limite) pour remplacer c : si c_1 n'est pas coupant, c_2 est considéré comme coupant.

Remarques.-

- Le test de visibilité se réduit à un test de signe sur un produit vectoriel et celui de semi-visibilité correspond au produit vectoriel nul.

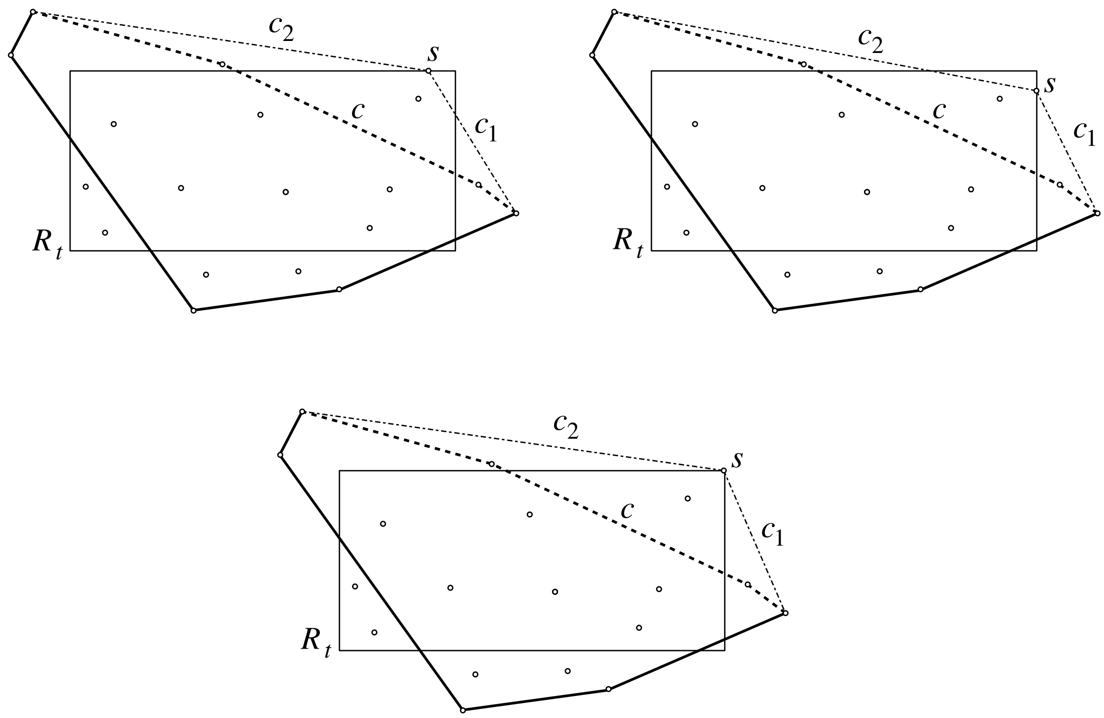

Figure 4: Les 3 cas de l'événement site. La ligne polygonale P des côtés visibles de s est en traits pointillés épais et les nouveaux côtés c_1 et c_2 sont en traits pointillés fins.

- Pour déterminer le nouveau côté coupant il suffit de savoir à quel côté de R_t appartient s. Cette information peut être conservée lors du calcul de Δ_s.
- Lorsque s appartient à deux côtés de R_t, c'est à dire que s est un sommet de R_t, aucun des côtés c_1 ou c_2 n'est coupant, d'après la définition des côtés coupants. Il n'est cependant pas nécessaire de traiter ce cas à part dans l'algorithme. En effet, le point limite de c_1 ou de c_2 est, dans ce cas, confondu avec s et sera traité par l'événement point limite suivant.

4.2 L'événement point limite

Lorsque le point limite d'un côté coupant est atteint, ce côté n'est plus coupant et il suffit de le supprimer de l'ensemble des côtés coupants. Le « coin » du rectangle de balayage qui était découpé par ce côté est maintenant vide. Lorsqu'il n'y a plus aucun côté coupant, le rectangle de balayage est complètement contenu dans l'enveloppe convexe courante. Il en est de même pour tous les sites non encore balayés et la construction de l'enveloppe convexe est alors terminée.

4.3 L'algorithme de balayage

L'algorithme de construction de l'enveloppe convexe par balayage selon un rectangle comporte une phase d'initialisation et une phase de traitement des événements.

Prétraitement :

– rechercher les sites extrémaux initiaux s_E, s_O, s_N, s_S ;

- construire $conv(s_E, s_O, s_N, s_S)$ et déterminer ses côtés coupants (i.e. les côtés qui coupent l'intérieur de R_0) avec leurs points limites ;
- pour chaque site s de S déterminer la distance Δ_s ;
- trier les sites de S selon cette distance.

Traitement :

- tant qu'il y a des sites **et** des côtés coupants
  soit s et l respectivement le prochain site et le prochain point limite à traiter
  si $\Delta_s < \Delta_l$ traiter le site s
  sinon traiter le point limite l

Théorème 2 *Théorème.- Cet algorithme construit l'enveloppe convexe d'un ensemble S de n sites du plan avec un prétraitement en $O(n \log n)$ et un traitement en $O(n')$ où n' est le nombre de sites traités à la fin de l'algorithme.*

Preuve.- (i) La recherche des sites extrémaux et le calcul de la distance Δ_s de chaque site s à R_0 est linéaire en n. Comme le tri est en $O(n \log n)$, la phase de prétraitement est en $O(n \log n)$. (ii) Comme le nombre de côtés coupants est au plus 4, la gestion des événements limites est en $O(1)$. D'après la proposition, lors d'un événement au plus un côté coupant est supprimé. Le nombre total de côtés coupants supprimés lors du balayage est donc inférieur à n'.

En outre, toute suppression d'un côté visible ou semi-visible non coupant de $conv(T)$, entraîne également la suppression d'un sommet de $conv(T)$. Comme les sommets de $conv(T)$ sont des sites précédemment balayés, le nombre total de côtés non coupants supprimés est inférieur à n'.

Il en résulte que la phase de traitement est en $O(n')$. □

5 Performances pratiques

L'algorithme a été implanté en langage C sur un station Silicon Graphics *Indigo*^{2} Hight Impact R4400 avec 128 Mo de mémoire.

Nous donnons ci-dessous les performances obtenues pour des ensembles de 200.000 sites répartis quasi-uniformément dans un carré 2x2 et dans un cercle de rayon 1.

	Temps total	Prétraitement	Traitement	Nombre n' de sites traités	Rapport n'/n
Carré	3,21 s	3,19 s	0,02 s	1.896	0,01
Cercle	3,38 s	3,17 s	0,21 s	72.831	0,36

Ces résultats montrent que l'algorithme est pénalisé par une étape de prétraitement particulièrement coûteuse. En effet, tous les sites sont triés alors que le traitement ne concerne qu'une partie des sites. L'algorithme passe trop de temps à trier et trie inutilement un nombre important des sites, notamment lorsque ces sites sont répartis dans un carré.

Afin d'améliorer les performances de l'algorithme, il convient donc de réduire la phase de prétraitement.

6 L'algorithme avec des tris fractionnés

Afin de réduire le temps de tri, on peut se fixer une distance Δ_m maximale et ne trier que les sites s de S qui sont tels que $\Delta_s = \Delta_m$.

Si à la fin du traitement des sites sélectionnés, il ne reste plus de côté coupant, la construction de $conv(S)$ est terminée.

Dans le cas contraire, soit c le côté coupant dont le point limite l maximise la distance Δ_l. Seuls les sites s qui sont tels que $\Delta_m < \Delta_s = \Delta_l$ peuvent encore appartenir à la frontière de l'enveloppe convexe de S. Il suffit donc de trier et de traiter cette partie des sites.

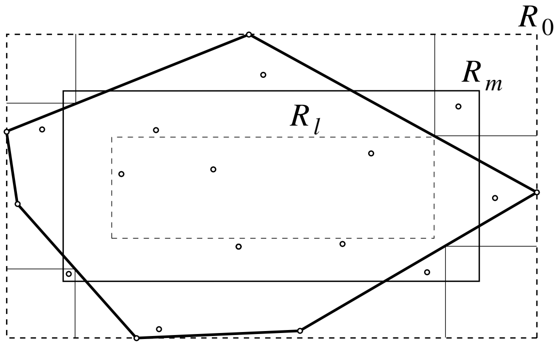

Figure 5: Lorsque l'enveloppe convexe des sites compris entre R_0 et R_m est construite, il ne reste plus qu'à traiter les sites compris entre R_m et R_l.

Le choix initial de Δ_m est déterminant quant aux performances de l'algorithme.

Les tableaux ci-dessous donne le temps d'exécution de l'algorithme en fonction de Δ_m pour des distributions quasi-uniformes dans un carré et dans un cercle.

Δ_m	0	0,0001	0,001	0,002	0,004	**0,005**	0,1	0,2	0,5
Temps (s)	2,55	0,97	0,53	0,44	0,44	**0,32**	0,33	1,21	2,33

Ensemble de 200.000 sites répartis quasi-uniformément dans un carré 2x2.

Δ_m	0	0,05	0,1	0, 2	**0,3**	0,4	0,5
Temps (s)	2,53	1,91	1,68	1,49	**1,4**	1,9	2,41

Ensemble de 200.000 sites répartis quasi-uniformément dans un cercle de rayon unité.

Le temps minimum est obtenu lorsque la valeur de Δ_m est égale à la distance Δ_l du dernier point limite l traité à R_0. La valeur de Δ_m détermine alors exactement la position du rectangle de balayage à la fin de l'algorithme.

Toute valeur de Δ_m supérieure à Δ_l entraîne des tris inutiles alors qu'une valeur inférieure nécessite de reparcourir l'ensemble des sites pour sélectionner ceux qui restent à traiter. Un problème ouvert est de déterminer à priori une bonne valeur pour Δ_m.

7 Comparaison avec l'algorithme de Bhattacharya et Toussaint

Nous comparons nos résultats avec ceux de l'algorithme de Bhattacharya et Toussaint [BT83]. Cet algorithme consiste à déterminer dans un premier temps les 8 sites de S qui maximisent ou minimisent les valeurs x, y, $x+y$ et $x-y$ et à supprimer tous les sites de S qui appartiennent au polygone convexe formé par ces 8 sites extrémaux (certains de ces 8 sites peuvent éventuellement être confondus). L'algorithme de Graham [Gra72] est ensuite utilisé pour construire l'enveloppe convexe des sites restants.

Les résultats ci-dessous on été obtenus sur la station Silicon Graphics où les deux algorithmes ont été implantés.

Nombre de sites	200.000	500.000	1.000.000
Algo. de balayage	0,32 s	0,77 s	1,5 s
Algo. de [BT83]	0,57 s	1,51 s	3,2 s

Ensembles de sites répartis quasi-uniformément dans un carré 2x2.

Nombre de sites	200.000	500.000	1.000.000
Algo. de balayage	1,4 s	4,05 s	9,3 s
Algo. de [BT83]	0,7 s	1,71 s	3,82 s

Ensembles de sites répartis quasi-uniformément dans un cercle de rayon unité.

Les résultats statistiques montrent que notre algorithme est particulièrement efficace lorsque les sites sont répartis quasi-uniformément dans un carré ou plus généralement dans un rectangle. L'algorithme de Bhattacharya et Toussaint est plus adapté lorsque les sites sont répartis dans un cercle.

8 Conclusion

Nous avons proposé un algorithme qui construit l'enveloppe convexe d'un ensemble de sites du plan en balayant ce plan par un rectangle.

Les résultats statistiques montrent que l'algorithme est particulièrement efficace lorsque les sites sont répartis quasi-uniformément dans un rectangle.

Plus généralement, le plan peut être balayé par un polygone convexe quelconque lorsque les sites sont répartis dans un tel polygone. Le nombre maximum de côtés coupants est alors égal au nombre de côtés du polygone mais le calcul de la distance d'un point au polygone initial est plus coûteux.

Lorsque les sites sont répartis dans un cercle, il est plus intéressant de balayer le plan par un cercle. Cependant, tous les côtés de l'enveloppe convexe courante peuvent être coupants dans ce cas. En plaçant les côtés coupants dans un arbre binaire de recherche, on obtient alors un algorithme dont la partie traitement est en $O(n' \log h)$ où h est le nombre de sommets de l'enveloppe convexe finale et n' le nombre de sites traités.

La méthode de balayage introduite ici peut être utilisée pour la résolution d'autres problèmes de géométrie algorithmique tels que la construction du diagramme de Delaunay des points les plus éloignés [AKSS96].

References

[AKSS96] Bruno Adam, Pierre Kauffmann, Dominique Schmitt, and Jean-Claude Spehner. Shrinking sweep algorithms for planar convex hulls. Report 96-3, Labo. MAGE, Univ. Haute-Alsace, Mulhouse, FRANCE, 1996.

[AT78] S. G. Akl and G. T. Toussaint. An improved algorithm to check for polygon similarity. *Inform. Process. Lett.*, 7:127–128, 1978.

[BGJR91] K. H. Borgwardt, N. Gaffke, M. Junger, and G. Reinelt. Computing the convex hull in the Euclidean plane in linear expected time. In P. Gritzman and B. Sturmfels, editors, *Applied Geometry and Discrete Mathematics: The Victor Klee Festschrift*, volume 4 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 91–107. AMS Press, 1991.

[BT83] B. K. Bhattacharya and G. T. Toussaint. Time- and storage-efficient implementation of an optimal convex hull algorithm. *Image Vision Comput.*, 1:140–144, 1983.

[Gra72] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Inform. Process. Lett.*, 1:132–133, 1972.

[KS86] D. G. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM J. Comput.*, 15:287–299, 1986.

[PH77] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Commun. ACM*, 20:87–93, 1977.

[Pre79] F. P. Preparata. An optimal real-time algorithm for planar convex hulls. *Commun. ACM*, 22:402–405, 1979.