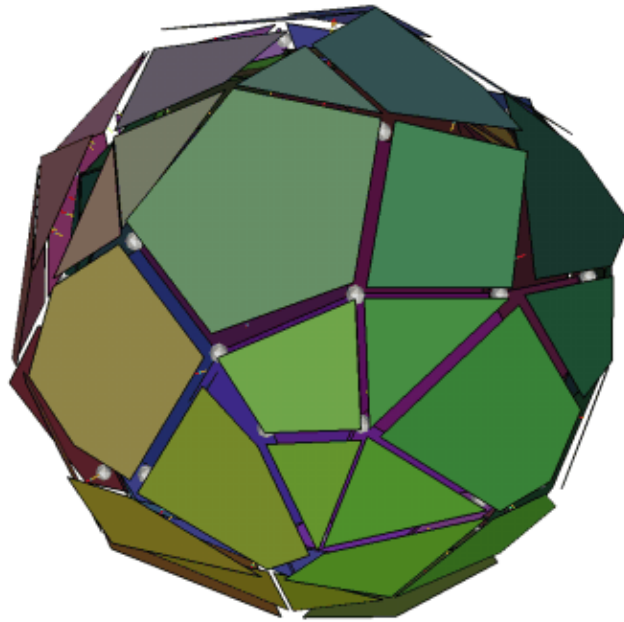


Μέθοδοι Υπολογισμού Κυρτού Περιγράμματος
(Convex Hull)

Συνόλου Σημείων στο n -διάστατο Χώρο

Ιωάννης Π. Δροσίτης

Εργασία στο Μεταπτυχιακό Μάθημα
Αλγόριθμοι και Πολυπλοκότητα II



Εθνικό Μετσόβιο Πολυτεχνείο

Ιούνιος 2001

Copyright © 2001 Δροσίδης Ιωάννης
Εργαστήριο Υπολογιστικών Συστημάτων
Εθνικό Μετσόβιο Πολυτεχνείο.
<http://www.cslab.ntua.gr/>

Η παρούσα εργασία έγινε στα πλαίσια του μεταπτυχιακού μαθήματος “**Αλγόριθμοι και Πολυπλοκότητα II**”, υπό την επίβλεψη του κ. Ευστάθιου Ζάχου.

*μακάριοι οί κλαίοντες νῦν,
ὅτι γελάσσετε.*

κατὰ Λουκᾶν στί 21.

Περιεχόμενα

Ευχαριστίες	v
Πρόλογος	vii
1 Εισαγωγή	1
1.1 Το Μοντέλο του Προβλήματος	2
1.2 Ορισμοί και Συμβολισμός	3
2 Πρώτοι Αλγόριθμοι Κυρτού Περιγράμματος	5
2.1 Ο Αλγόριθμος Graham's Scan	6
2.2 Ο Αλγόριθμος Jarvis's March	6
3 Εξελιξεις των Κλασσικών Αλγορίθμων	9
3.1 Ο Αλγόριθμος MergeHull (<i>divide-and-conquer</i>)	10
3.2 Ο Αλγόριθμος QuickHull (<i>beneath-beyond</i>)	12
3.3 Απλοποιημένος "Ultimate Planar Convex Hull" Αλγόριθμος	15
3.4 Ο Αλγόριθμος "Group-and-Wrap" στο Επίπεδο	17
3.5 Συμπεράσματα για τις Δύο Διαστάσεις	20
4 Αλγόριθμοι Κυρτού Περιγράμματος στις Τρεις Διαστάσεις	23
4.1 Επέκταση του Jarvis's March	24
4.2 Ο Αλγόριθμος QuickHull στις Τρεις Διαστάσεις	25
4.3 Επέκταση του Group-and-Wrap στις Τρεις Διαστάσεις	28
4.4 Συμπεράσματα για τις Τρεις Διαστάσεις	30

5	Κυρτά Περιγράμματα Μεγαλύτερων Διαστάσεων	33
5.1	Δύο Κάτω Όρια	34
5.2	Περιγραφή Αλγορίθμων	35
5.3	Συμπεράσματα	37
6	Ξεφεύγοντας από τη Θεωρία	39
6.1	Πολυεδρική Ιεραρχία (Polyedral Hierarchy)	39
6.2	Διπλά Συνδεδεμένη Λίστα Ακμών (doubly-connected edge list)	40
6.3	Το πρόγραμμα qhull	41
	Βιβλιογραφία	43

Ευχαριστίες

Η παρούσα εργασία δεν θα γινόταν πραγματικότητα χωρίς την βοήθεια του αγαπητού φίλου Χριστόδουλου Φραγκιουδάκη, ο οποίος με βοήθησε να ξεκινήσω και υπήρξε πρόθυμος να μου παρέχει οτιδήποτε σχετικό βιβλιογραφικό υλικό είχε στα χέρια του.

Ευχαριστώ επίσης, τον αγαπητό μου καθηγητή Ευστάθιο Ζάχο, για την αναμονή και την κατανόηση που μου έδειξε, όπως και γι' αυτά που έμαθα κοντά του.

Η εμφάνιση της εργασίας δεν υπήρχε περίπτωση να μοιάζει με την παρούσα (L^AT_EX τρομερό :-), χωρίς τη συμβολή των φίλων και αγαπητών συνεργατών μου Θοδωρή Ανδρόνικου και Γιάννη Παναγόπουλου. Ευχαριστώ πολύ παιδιά!

Πρόλογος

Στην παρούσα εργασία επιχειρούμε να προσεγγίσουμε το πρόβλημα του υπολογισμού του κυρτού περιγράμματος (convex hull) συνόλου σημείων σε διάφορες διαστάσεις. Παρουσιάζονται οι κυριότεροι αλγόριθμοι της βιβλιογραφίας, καθώς και η πολυπλοκότητα που παρουσιάζει ο καθένας.

Πραγματοποιείται αρχικά μια αναδρομή, τόσο στους πρώτους αλγόριθμους που παρουσιάστηκαν για τις δύο διαστάσεις [Gra67, Jar73], όσο και σ' αυτούς που δημοσιεύθηκαν πρόσφατα -την τελευταία πενταετία- και αναφέρονται και σε μεγαλύτερες διαστάσεις. Στα πλαίσια της παρούσας εργασίας, μας ενδιαφέρουν κυρίως οι ντετερμινιστικοί αλγόριθμοι.

Η εργασία χωρίζεται σε 6 κεφάλαια:

Το **Κεφάλαιο 1** αποτελεί μια εισαγωγή στο χώρο. Παρουσιάζεται το μοντέλο του προβλήματος καθώς και κάποιοι βασικοί ορισμοί.

Στα **Κεφάλαια 2 και 3** παρουσιάζονται οι πλέον γνωστοί αλγόριθμοι υπολογισμού κυρτού περιγράμματος για τις δύο διαστάσεις. Στο κεφάλαιο 2 αναφέρονται οι δύο παλαιότεροι και γνωστότεροι στο χώρο Graham's scan [Gra67] και Jarvis's march [Jar73], ενώ στο κεφάλαιο 3 παρουσιάζονται κάποιες εξελίξεις τους αλλά και διάφορες νεότερες προσεγγίσεις.

Στο **Κεφάλαιο 4** αναφέρονται οι αλγόριθμοι κυρτού περιγράμματος τριών διαστάσεων.

Το **Κεφάλαιο 5** περιγράφει εν συντομία τους αλγόριθμους μεγαλύτερων διαστάσεων.

Το **Κεφάλαιο 6**, για να δοθεί μια πρακτική χρεία στην εργασία, παρουσιάζει κάποιες δομές δεδομένων, οι οποίες βιβλιογραφικά κρίθηκαν χρήσιμες κατά την επεξεργασία πολυτόπων.

Στο σημείο αυτό θα ήταν παράλληψη μας να μην αναφέρουμε δύο ακόμη εργασίες της βιβλιογραφίας, οι οποίες επίσης ασχολήθηκαν με αλγόριθμους κυρτού περιγράμματος και τις πολυπλοκότητές τους.

Πρόκειται για την διατριβή του Timothy Chan [Cha95] και για την εργασία της Valentina Bayer [Bay99]. Στο δεύτερο πόνημα, μάλιστα, αναφέρονται όλων των ειδών οι αλγόριθμοι, ντετερμινιστικοί, προσεγγιστικοί και πιθανοτικοί, γεγονός που το καθιστά πολύ χρήσιμο ως αναφορά.

Σημειώνουμε τέλος, μια καταπληκτική υλοποίηση του αλγορίθμου QuickHull που αναπτύχθηκε στο πανεπιστήμιο της Minnesota και επιτρέπει τον υπολογισμό κυρτών περιγραμμάτων, διαγραμμάτων Voronoi, τριγωνοποιήσεων Delaunay κ.α. Το πρόγραμμα ακούει στα ονόματα qhull και qconvex και είναι διαθέσιμο στις ακόλουθες διευθύνσεις:

```
<http://www.geom.umn.edu/locate/qhull>  
<ftp://geom.umn.edu/pub/software/qhull13-0.zip>  
<http://www.geom.umn.edu/~bradb/qhull-news.html>
```

Θα είναι χαρά για μένα οποιαδήποτε υπόδειξη για τυχόν λάθη ή παραλήψεις, που ίσως υπάρχουν. Ελπίζω η εργασία αυτή να σας φανεί χρήσιμη!

Ιωάννης Π. Δροσίτης <jdros@cslab.ntua.gr>

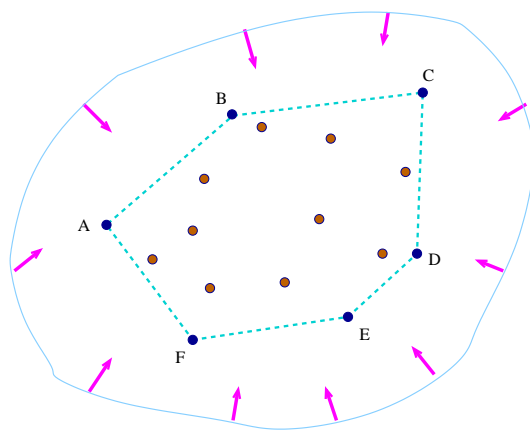
Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών,
Τομέας Πληροφορικής,
Εθνικό Μετσόβιο Πολυτεχνείο

Κεφάλαιο 1

Εισαγωγή

Ένα σημαντικό πρόβλημα, τόσο στο χώρο της υπολογιστικής γεωμετρίας, όσο και στο χώρο της θεωρίας Αλγορίθμων, αποτελεί το πρόβλημα της κατασκευής του κυρτού περιγράμματος (convex hull).

Η σημαντικότητα του προβλήματος έγκειται στο πλήθος των εφαρμογών του (αναγνώριση προτύπων, στατιστική, επεξεργασία εικόνας) αλλά και στο ότι αποτελεί εργαλείο επίλυσης πλήθους προβλημάτων στην υπολογιστική γεωμετρία. Γι' αυτό άλλωστε αποτελεί και (ανοιχτό) αντικείμενο ενασχόλησης σημαντικών θεωρητικών την τελευταία τριακονταετία.



Σχήμα 1: Κυρτό περίγραμμα συνόλου σημείων. Αν θεωρήσουμε τα σημεία ως καρφιά σε μια ξύλινη επιφάνεια, το κυρτό τους περίγραμμα βρίσκεται αν περιβάλλουμε το σύνολο των καρφιών με έναν (κλειστό) ελαστικό ιμάντα. Το σύνολο των καρφιών που ακουμπούν στον ιμάντα, αποτελούν το κυρτό περίγραμμα του αρχικού συνόλου.

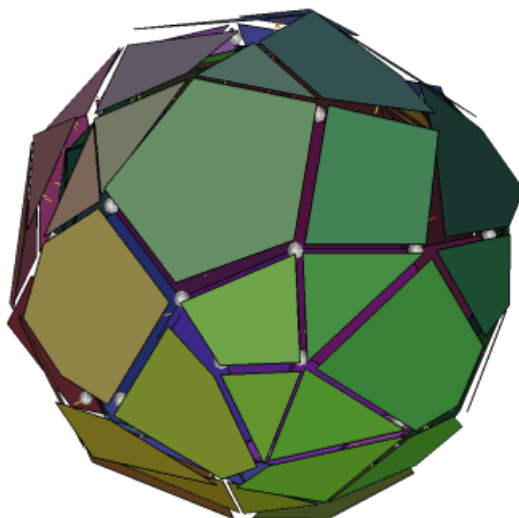
Η έννοια του κυρτού περιγράμματος με δυο λόγια είναι η εξής: για δεδομένο σύνολο σημείων, είναι το μικρότερο υποσύνολο, του οποίου το περίγραμμα, περικλείει το αρχικό σύνολο. Απλούστερα, αν θεωρήσουμε τα σημεία ως καρφιά σε μια ξύλινη επιφάνεια, το κυρτό τους περίγραμμα βρίσκεται αν περιβάλλουμε το σύνολο των καρφιών με έναν κλειστό ελαστικό ιμάντα. Το σύνολο των καρφιών που ακουμπούν στον ιμάντα, αποτελούν το κυρτό περίγραμμα του αρχικού συνόλου (σχήμα 1).

Στην παρούσα ενότητα παρουσιάζεται το μοντέλο του προβλήματος, καθώς και ορισμένοι ορισμοί που χρησιμοποιούνται στη συνέχεια.

1.1 Το Μοντέλο του Προβλήματος

Έστω $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$ ένα σύνολο n -σημείων στον d -διάστατο Ευκλείδειο χώρο E^d , όπου $d \geq 2$. Ως *κυρτό περίγραμμα* του P , $\mathcal{CH}(P) = \text{conv}(P)$, ορίζεται το *μικρότερο* σύνολο σημείων, του οποίου το περίγραμμα περικλείει το P (σχήμα 1).

Στα πλαίσια της συγκεκριμένης εργασίας, θεωρούμε ότι τα σημεία του P βρίσκονται σε *γενική θέση*, δηλαδή, δεν υπάρχουν $d + 1$ σημεία που να ανήκουν στο ίδιο υπερεπίπεδο. Σε διαφορετική περίπτωση, υπάρχουν μέθοδοι που περιστρέφουν ελάχιστα τα σημεία εισόδου, έτσι ώστε να μεταπίπτουν στην παραπάνω γενική περίπτωση [EM90, EC92].



Σχήμα 2: Πολύτοπο τριών διαστάσεων.

Σημειώνουμε επίσης, ότι στην πορεία της εργασίας οι όροι “έπάνω/κάτω” και “ψηλότερο/χαμηλότερο” αναφέρονται ως προς την τελευταία συντεταγμένη.

1.2 Ορισμοί και Συμβολισμός

Ένα σύνολο $S \subseteq E^d$ είναι *κυρτό*, αν για όλα τα σημεία $p, q \in S$, το ευθύγραμμο τμήμα \overline{pq} ανήκει στο S [PS85]. Το κυρτό περίγραμμα του P είναι η τομή όλων των κυρτών συνόλων που περιέχουν το P . Ισοδύναμα, μπορεί να οριστεί και ως η τομή όλων των υποχώρων που περιέχουν το P .

Η τομή ενός πεπερασμένου συνόλου (κλειστών) ημιχώρων λέγεται *πολύτοπο* (polytope) – σχήμα 2. Οι Grünbaum, McMullen και Shephard, απέδειξαν ότι το κυρτό περίγραμμα $\text{conv}(P)$, συνόλου P , αποτελεί πολύτοπο [Grü67, MS71]. Έτσι, όλη η θεωρία των πολυτόπων (μέθοδοι αναπαράστασης, ιδιότητες, κ.λπ.) μπορεί να χρησιμοποιηθεί στα κυρτά περιγράμματα. Για παράδειγμα, για $d = 2$, το κυρτό περίγραμμα είναι κυρτό πολύγωνο (σχήμα 1).

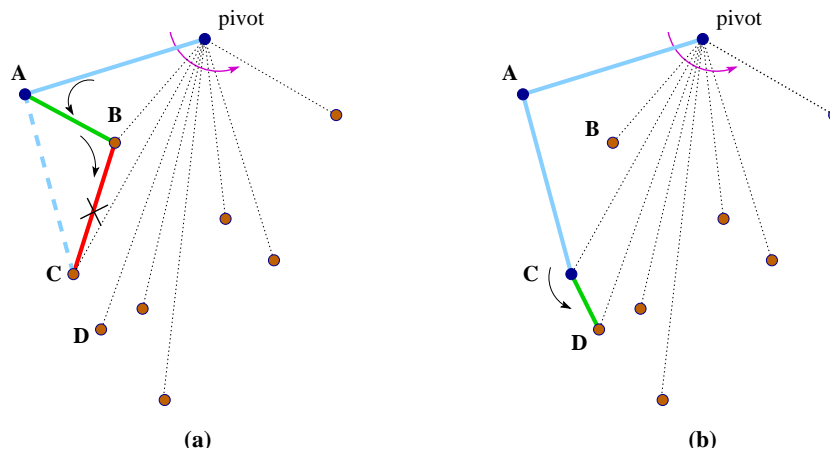
Ένα πολύγωνο μπορεί να αναπαρασταθεί από την ακολουθία των κορυφών του, κατά μια συγκεκριμένη φορά, π.χ. αντίθετα με τη φορά των δεικτών του ρολογιού. Σε μεγαλύτερες διαστάσεις, ένα πολύτοπο απεικονίζεται από ένα σύνολο δομών: ένα σύνολο των κορυφών του, ένα σύνολο των πλευρών του και έναν πίνακα γειτνίασης (βλ. και κεφάλαιο 6) [Ede87].

Έστω $\mathcal{P} \in E^d$ πολύτοπο με μη-κενό εσωτερικό. Αν h ένα υπερεπίπεδο το οποίο έχει κοινά σημεία με το εξωτερικό του \mathcal{P} , αλλά όχι με το εσωτερικό του, τότε η τομή $h \cap \mathcal{P}$ καλείται *πλευρά* του \mathcal{P} . Αν το πολύτοπο είναι διάστασης d , η πλευρά του είναι διάστασης $d - 1$. Σημειώνουμε ότι πλευρά διάστασης 1 καλείται *ακμή*, ενώ διάστασης 0 καλείται *κορυφή*.

Κεφάλαιο 2

Πρώτοι Αλγόριθμοι Κυρτού Περιγράμματος

Το πρόβλημα του υπολογισμού του κυρτού περιγράμματος έχει ιστορία αντίστοιχης ηλικίας με αυτή της υπολογιστικής γεωμετρίας. Οι πρώτοι αλγόριθμοι που παρουσιάστηκαν αναφέρονταν στο επίπεδο ($d = 2$). Η ιστορία ξεκίνησε με τον αλγόριθμο του Graham [Gra67], γνωστός και ως *Graham's scan*.



Σχήμα 3: Δύο βήματα από τον αλγόριθμο *Graham's Scan*. Από το σημείο A στο C πραγματοποιείται μια οπισθοδρόμηση (a), ενώ στο σημείο D ο αλγόριθμος προχωρεί κανονικά (b). Ως *pivot* διαλέγουμε το σημείο με τη μεγαλύτερη συντεταγμένη y .

2.1 Ο Αλγόριθμος Graham's Scan

Ο αλγόριθμος αυτός είναι πολυπλοκότητας $O(n \log n)$ και αποτελεί τον πρώτο αποδοτικό αλγόριθμο που παρουσιάστηκε στις δύο διαστάσεις [Gra67]. Έχει ως εξής:

Graham's Scan

1967

ΕΙΣΟΔΟΣ: ένα σύνολο n σημείων στον διδιάστατο χώρο ($d = 2$)

ΕΞΟΔΟΣ: το υποσύνολο των σημείων αυτών, το οποίο αποτελεί το κυρτό περίγραμμα του αρχικού συνόλου

1. Βρίσκουμε ένα ακρότατο σημείο, το οποίο σίγουρα ανήκει στο περίγραμμα. Το σημείο αυτό το καλούμε *pivot* και το χρησιμοποιούμε ως άξονα. Ένα τέτοιο σημείο μπορεί εύκολα να βρεθεί π.χ. παίρνοντας αυτό με τη μεγαλύτερη y -συντεταγμένη (σχήμα 3).
 2. Βρίσκουμε τη γωνία που σχηματίζει το καθένα από τα υπόλοιπα σημεία γύρω από το σημείο *pivot*. Κατόπιν, ταξινομούμε τα σημεία ανάλογα με τη γωνία που σχηματίζουν με το σημείο *pivot*, κατά αύξουσα σειρά γωνίας (το σημείο με τη μικρότερη γωνία μπαίνει πρώτο, κοκ.). Δημιουργείται έτσι ένα σχήμα με $n - 1$ ακτίνες, οι οποίες ξεκινούν από το σημείο *pivot* και καταλήγουν στα υπόλοιπα.
 3. Το πρώτο σημείο από τα ταξινομημένα (σημείο **A** στο σχήμα) τοποθετείται στο σύνολο του κυρτού περιγράμματος.
 4. Ξεκινώντας από το δεύτερο σημείο, προσθέτουμε στο περίγραμμα τα σημεία για τα οποία κάνουμε αριστερή στροφή, ενώ οπισθοχωρούμε (backtracking) στα σημεία που πραγματοποιείται δεξιά στροφή (σχήμα 3a).
-

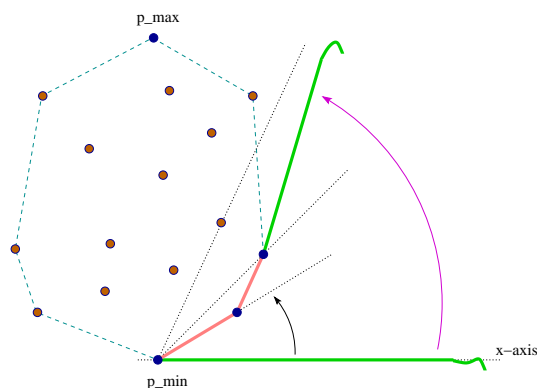
Όσον αφορά το n , ο εν λόγω αλγόριθμος είναι ασυμπτωτικά βέλτιστος, μιας και ένα κατώτατο όριο $\Omega(n \log n)$ τίθεται από την απαίτηση ταξινόμησης.

Τα παραπάνω όρια και αποτελέσματα, προϋποθέτουν ότι ο αριθμός των κόμβων του κυρτού περιγράμματος h αποτελεί έστω και μικρό κομμάτι του αριθμού των αρχικών σημείων n . Στην περίπτωση όπου το h είναι πολύ μικρότερο από το n , όπως αποδείχτηκε από τον Jarvis [Jar73], είναι δυνατόν να υπερπηδηθεί το φράγμα $n \log n$. Ο αλγόριθμος που παρουσίασε περιγράφεται στη συνέχεια.

2.2 Ο Αλγόριθμος Jarvis's March

Όπως και ο αλγόριθμος του Graham, η επιλογή των σημείων εξαρτάται από τις σχετικές γωνίες που σχηματίζονται μεταξύ των σημείων. Η διαφορά από τον παραπάνω αλγόριθμο είναι ότι χρησιμοποιεί την τεχνική που ονομάζεται *gift-wrapping* (ή *package-wrapping*).

Η διαδικασία αναλογικά μοιάζει με το *τύλιγμα* ενός σπάγκου γύρω από τα σημεία. Ο σπάγκος δένεται στο χαμηλότερο σημείο και αρχικά, κρατιέται τεντωμένος οριζόντια, έως ένα σημείο με



Σχήμα 4: Η τεχνική του τυλίγματος, για της εύρεση του κυρτού περιγράμματος συνόλου σημείων στο επίπεδο.

μεγαλύτερη x -συντεταγμένη από το αρχικό. Κατόπιν, αρχίζει να περιστρέφεται, αντίθετα από τη φορά των δεικτών του ρολογιού, γύρω από το σύνολο των σημείων, έως ότου επιστρέψει στο αρχικό σημείο (σχήμα 4). Κάθε σημείο το οποίο αναγκάζει τον σπάγκο να λυγίσει, αποτελεί σημείου του ζητούμενου κυρτού περιγράμματος [Jar73].

Περιγράφουμε τον εν λόγω αλγόριθμο, αναφερόμενοι στο μισό τμήμα του (το δεξί τμήμα του κυρτού περιγράμματος). Αντίστοιχα, πραγματοποιείται και για το υπόλοιπο μισό.

Jarvis's March

1973

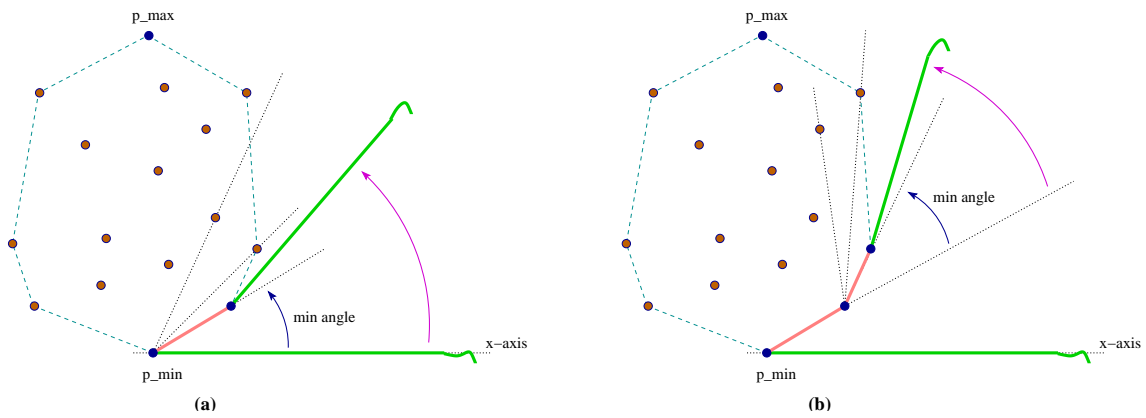
ΕΙΣΟΔΟΣ: ένα σύνολο n σημείων στον διδιάστατο χώρο ($d = 2$)

ΕΞΟΔΟΣ: τα σημεία που αποτελούν το κυρτό περίγραμμα του αρχικού συνόλου

1. Βρίσκουμε το σημείο με τη μικρότερη συντεταγμένη y , έστω \mathbf{p}_{min} . Το σημείο αυτό τίθεται ως σημείο αναφοράς. Βρίσκουμε επίσης, το σημείο με τη μέγιστη συντεταγμένη y \mathbf{p}_{max} .
 2. Υπολογίζουμε τις γωνίες που σχηματίζονται από τα ευθύγραμμα τμήματα τα οποία ενώνουν καθένα από τα υπόλοιπα σημεία με το σημείο αναφοράς, με τον οριζόντιο άξονα (σχήμα 4).
 3. Το σημείο με τη μικρότερη γωνία ανήκει στο κυρτό περίγραμμα και εισάγεται στο σύνολο που το αποτελεί. Αν πρόκειται για το σημείο \mathbf{p}_{max} , ο αλγόριθμος τελειώνει· διαφορετικά, γίνεται σημείο αναφοράς και επαναλαμβάνεται το βήμα 2 (βλέπε σχήμα 5).
-

Ο εν λόγω αλγόριθμος, είναι ο πρώτος που παρουσιάστηκε στη βιβλιογραφία, με πολυπλοκότητα η οποία εξαρτάται από την έξοδο. Η πολυπλοκότητα του είναι $O(nh)$, όπου h , ο αριθμός των κορυφών του κυρτού περιγράμματος.

Σημειώνουμε ότι ο αλγόριθμος είναι ιδιαίτερα αποδοτικός, σε σχέση με τον προηγούμενο του



Σχήμα 5: Για δεδομένο σύνολο σημείων, τα δύο πρώτα βήματα του αλγορίθμου *Jarvis's March* στο επίπεδο.

Graham, όταν $h \ll \log n$. Σε διαφορετική περίπτωση, ο Graham's Scan πραγματοποιεί λιγότερα βήματα.

Κεφάλαιο 3

Εξελίξεις των Κλασσικών Αλγορίθμων

Μετά την παρουσίαση των δύο κλασσικών μεθόδων, από τους Graham και Jarvis, διάφοροι αλγόριθμοι εμφανίστηκαν, όσον αφορά τις δύο διαστάσεις. Σημειώνουμε τους *MergeHull* και *QuickHull* [PS85].

Ο πρώτος οφείλεται στους Preparata και Hong [PH77] και βασίζεται στην τεχνική *divide-and-conquer*. Χρησιμοποιεί την τεχνική χωρισμού και ομαδοποίησης “merge”, όπως μαρτυρεί και το όνομα του. Η πολυπλοκότητα του είναι $O(n \log n)$, αντίστοιχη δηλαδή με αυτή του Graham’s Scan.

Ο αλγόριθμος QuickHull μελετήθηκε και ανακαλύφθηκε ανεξάρτητα από διάφορους ερευνητές κατά το τέλος της δεκαετία του '70. Οι παράλληλες προσεγγίσεις σημειώνονται στην πρόσφατη εργασία [BDH96], όπου παρουσιάζεται ο γενικός αλγόριθμος και για μεγαλύτερες διαστάσεις. Η πολυπλοκότητα του, για τις δύο διαστάσεις, είναι $O(nr)$, όπου r , ο αριθμός των κορυφών που επεξεργάζεται ο αλγόριθμος. Σημειώνουμε ότι είναι ο πρώτος αλγόριθμος που παρουσιάστηκε ο οποίος εξαρτάται από την έξοδο. Στην παρούσα εργασία αναφερόμαστε στην εν λόγω παραλλαγή των Barder, Dobkin και Hundanpaa [BDH96], τόσο επειδή είναι η πιο πρόσφατη, όσο και γιατί περιγράφει τον γενικευμένο αλγόριθμο για διάσταση $d \geq 2$.

Το επόμενο βήμα για τη βελτίωση της πολυπλοκότητας στις δύο διαστάσεις έγινε το 1986 από τους Kirkpatrick και Seidel [KS86] στη δημοσίευση με τίτλο “The Ultimate Convex Hull Algorithm”. Ο αλγόριθμος που παρουσίασαν είχε πολυπλοκότητα $O(n \log h)$, τη μικρότερη δηλαδή, μέχρι τώρα.

Στη συνέχεια, ο Chan παρουσίασε το 1995 μια απλοποιημένη έκδοση του παραπάνω αλγορίθμου, η οποία διατηρεί την ίδια χαμηλή πολυπλοκότητα [Cha95]. Η βιβλιογραφία ολοκληρώνεται για τις δύο διαστάσεις, με μια ακόμα μέθοδο από τον ίδιο, την οποία ονομάζει “group-and-wrap”. Πρόκειται για παραλλαγή της μεθόδου Jarvis’s March με ένα ενδιάμεσο βήμα ομαδοποίησης, το οποίο την κάνει αποδοτικότερη της αρχικής. Η πολυπλοκότητα και αυτής της μεθόδου είναι $O(n \log h)$, το οποίο είναι το βέλτιστο μέχρι στιγμής για τις δύο διαστάσεις.

3.1 Ο Αλγόριθμος MergeHull (*divide-and-conquer*)

Ο αλγόριθμος εφαρμόζει την τακτική *divide-and-conquer* και παρουσιάστηκε το 1977 από τους Preperata και Hong [PH77]. Μελετήθηκε και από άλλους ερευνητές, όπως οι Shamos [PS85] και Edelsbrunner [Ede87].

Το σύνολο των σημείων P χωρίζεται σε δύο υποσύνολα. Αρχικά υπολογίζονται τα κυρτά περιγράμματα για τα δύο αυτά υποσύνολα. Στη συνέχεια αυτά ενώνονται (*merge*), προκειμένου να σχηματιστεί το ενιαίο περίγραμμα.

Ο χωρισμός του συνόλου P στα υποσύνολα P_1 και P_2 , γίνεται μέσω μιας κάθετης γραμμής. Η γραμμή επιλέγεται έτσι ώστε να χωρίζει το σύνολο P περίπου στη μέση, ούτως ώστε τα δύο υποσύνολα να έχουν τον ίδιο αριθμό σημείων, για λόγους απόδοσης. Επίσης, πριν την όλη διαδικασία, το σύνολο P ταξινομείται ως προς τη συντεταγμένη x των σημείων. Χωρίς περιορισμό της γενικότητας, θεωρούμε ότι η ακολουθία $\{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\}$, είναι κατά αύξουσα σειρά της συντεταγμένης αυτής. Ο αλγόριθμος περιγράφεται στη συνέχεια.

MergeHull_2D(P)

1977

ΕΙΣΟΔΟΣ: ένα σύνολο P με n σημεία, στον διδιάστατο χώρο

ΕΞΟΔΟΣ: το ελάχιστο υποσύνολο του P , το οποίο αποτελεί το κυρτό περίγραμμα των σημείων του P

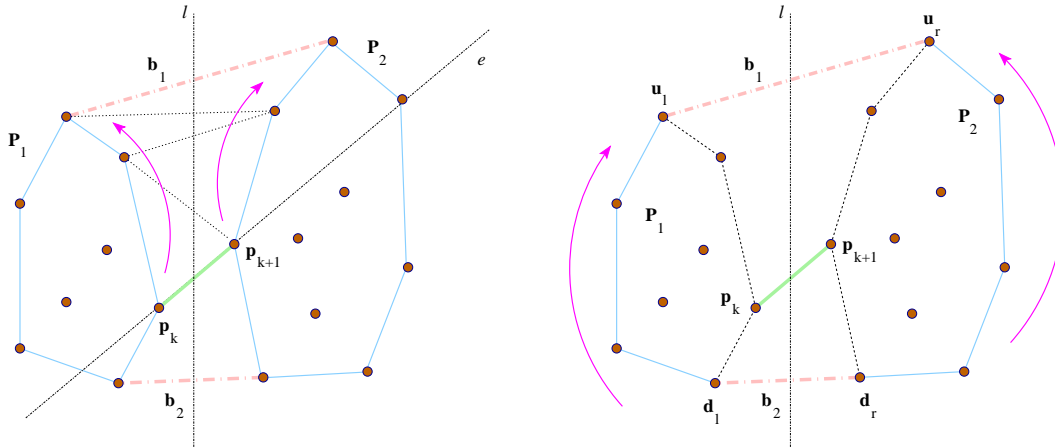
1. Αν $n \leq 3$ τότε: κατασκευάζεται το κυρτό περίγραμμα του P με προφανή τρόπο.
2. Αν $n > 3$ τότε:

DEVIDE: θέτουμε $k := \lfloor n/2 \rfloor$ και ορίζουμε τα σύνολα $P_1 = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_k\}$ και $P_2 = \{\mathbf{p}_{k+1}, \mathbf{p}_{k+2}, \dots, \mathbf{p}_n\}$.

RECUR: υπολογίζονται τα $\text{conv}(P_1)$ και $\text{conv}(P_2)$ αναδρομικά:
 $\text{conv}(P_1) := \text{MergeHull}_2\text{D}(P_1)$ και $\text{conv}(P_2) := \text{MergeHull}_2\text{D}(P_2)$.

MERGE: τα δύο υπολογισμένα κυρτά περιγράμματα συνδυάζονται για τον υπολογισμό του τελικού: $\text{conv}(P) := \text{Merge}(\text{conv}(P_1), \text{conv}(P_2))$.

Το μη προφανές τμήμα του αλγορίθμου είναι το βήμα της συννένωσης (merge). Για να πραγματοποιηθεί, απαιτείται η εύρεση των ακμών του $\mathcal{P} = \text{conv}(P)$ οι οποίες δεν είναι ακμές κανενός από τα $\mathcal{P}_1 = \text{conv}(P_1)$ και $\mathcal{P}_2 = \text{conv}(P_2)$. Δεδομένου ότι το \mathcal{P}_1 περιέχει σημεία με τετμημένη μικρότερη ή ίση του p_{k_x} , ενώ το \mathcal{P}_2 σημεία με τετμημένη μεγαλύτερη του p_{k+1_x} , τα δύο κυρτά περιγράμματα μπορούν να διαχωριστούν από μια κάθετη γραμμή ℓ (σχήμα 6).



Σχήμα 6: Εφαρμογή του αλγορίθμου MergeHull στο επίπεδο. Αφού βρεθούν τα κυρτά περιγράμματα των δύο υποσυνόλων του P , αναζητούνται οι ακμές που τα συνδέουν, ώστε να προκύψει το κυρτό περίγραμμα του P (αριστερά). Αφού βρεθούν οι δύο ακμές διασύνδεσης, δημιουργείται το συνολικό κυρτό περίγραμμα προσθέτοντας τις και αφαιρώντας όλες τις ενδιάμεσες (δεξιά).

Σύμφωνα με τα παραπάνω, υπάρχουν ακριβώς δύο τέτοιες ακμές, οι οποίες τέμνουν τη γραμμή ℓ και βρίσκονται η μία πάνω και η δεύτερη κάτω από το ευθύγραμμο τμήμα $s = \overline{p_k p_{k+1}}$. Ονομάζονται επάνω ακμή διασύνδεσης και κάτω ακμή διασύνδεσης, αντίστοιχα και συμβολίζονται b_1 και b_2 . Για τον εντοπισμό των δύο αυτών ακμών, εργαζόμαστε ως εξής. Θεωρούμε ότι τα \mathcal{P}_1 και \mathcal{P}_2 δίνονται με την μορφή διπλά συνδεδεμένων λιστών, όπου οι συναρτήσεις $\text{succ}(s)$ και $\text{pred}(s)$ δίνουν το επόμενο και το προηγούμενο σημείο του s αντίστοιχα, κατά την φορά των δεικτών του ρολογιού. Για το b_1 , ξεκινώντας από το $s = \overline{p_k p_{k+1}}$, διαδοχικά ανεβάζουμε τα άκρα του, ακολουθώντας τη φορά των δεικτών του ρολογιού στο \mathcal{P}_2 και αντίθετη στο \mathcal{P}_1 (σχήμα 6). Για την κάτω ακμή διασύνδεσης εργαζόμαστε ανάλογα.

Η τυπική περιγραφή του βήματος συννένωσης ακολουθεί με τη μορφή ψευδοκώδικα. Τα κυρτά περιγράμματα αναπαρίστανται ως διπλά συνδεδεμένες κυκλικές λίστες, με υλοποιημένες τις δύο συναρτήσεις που προαναφέραμε.

Merge($\mathcal{P}_1, \mathcal{P}_2$)

1. Υπολογίζονται το δεξιότερο σημείου \mathbf{p}_k του \mathcal{P}_1 και το αριστερότερο σημείο \mathbf{p}_{k+1} του \mathcal{P}_2 .
2. Θέτουμε $\mathbf{u}_l := \mathbf{p}_k$ και $\mathbf{u}_r := \mathbf{p}_{k+1}$: έστω e η ευθεία που περνάει από τα σημεία \mathbf{u}_l και \mathbf{u}_r .
3. */* Αναζητούμε την άνω ακμή διασύνδεσης */*
while [ένα από τα σημεία $\text{succ}(\mathbf{u}_l)$ και $\text{pred}(\mathbf{u}_r)$ βρίσκεται πάνω από τη γραμμή e] **do**
 if [το σημείο $\text{succ}(\mathbf{u}_l)$ βρίσκεται πάνω από την e] **then**
 $\mathbf{u}_l := \text{succ}(\mathbf{u}_l)$
 else
 $\mathbf{u}_r := \text{pred}(\mathbf{u}_r)$
4. */* Αναζητούμε την κάτω ακμή διασύνδεσης */*
 Ανάλογη διαδικασία εφαρμόζεται για την εύρεση της κάτω ακμής διασύνδεσης: υπολογίζονται τα σημεία \mathbf{d}_l και \mathbf{d}_r .
5. */* Δημιουργία της συννέωσης \mathcal{P} */*
 Δημιουργείται μια νέα, διπλά συνδεδεμένη κυκλική λίστα, η οποία αποτελείται από:
 - το τμήμα του \mathcal{P}_1 από το σημείο \mathbf{d}_l έως το σημείο \mathbf{u}_l , αντίθετα από την φορά των δεικτών του ρολογιού,
 - το τμήμα $\overline{\mathbf{u}_l \mathbf{u}_r}$,
 - το τμήμα του \mathcal{P}_2 από το σημείο \mathbf{u}_r έως το σημείο \mathbf{d}_r , κατά την φορά των δεικτών του ρολογιού και
 - το τμήμα $\overline{\mathbf{d}_r \mathbf{d}_l}$

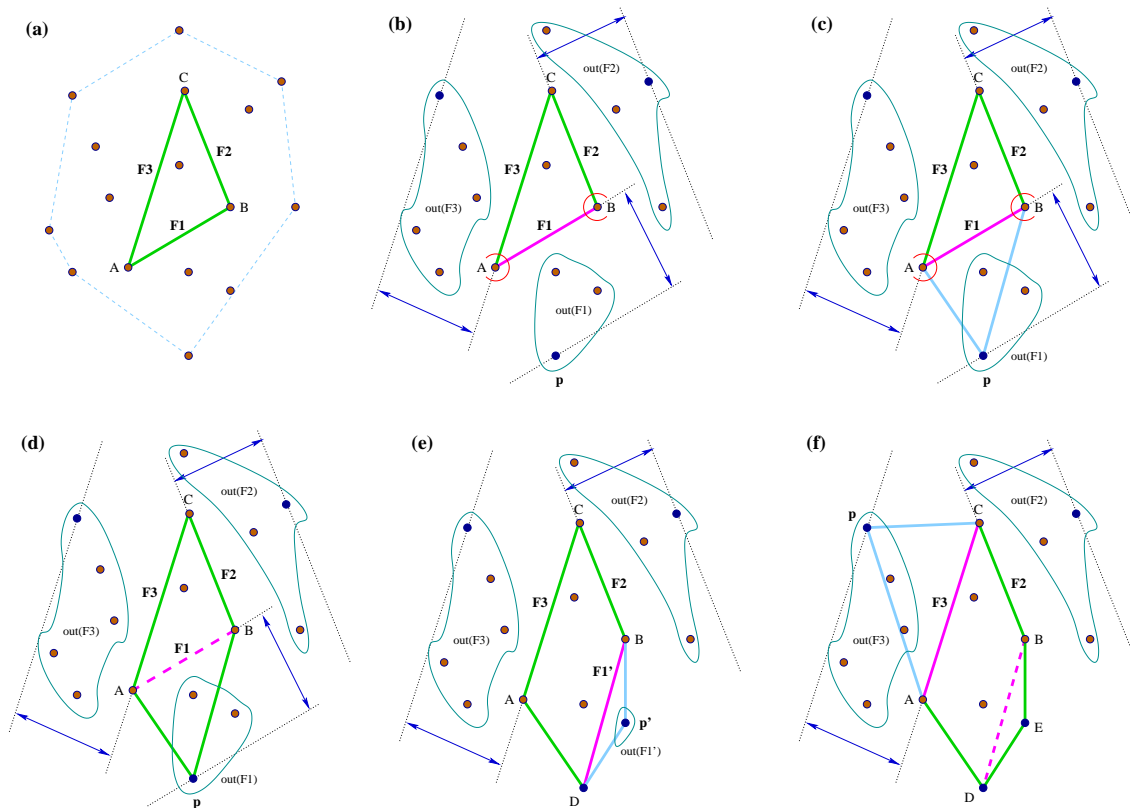
Η λίστα αυτή αποτελεί τη συννέωση των \mathcal{P}_1 και \mathcal{P}_2 και επιστρέφεται.

Η πολυπλοκότητα του αλγορίθμου MergeHull είναι $O(n \log n)$ και αναφέρεται στο επίπεδο. Ο περεταιίρω ενδιαφερόμενος αναγνώστης παραπέμπεται στα συγγράματα [PS85, Ede87], όπου ο εν λόγω αλγόριθμος αναλύεται λεπτομερώς. Στο [PS85] παρουσιάζονται και κάποιες ιδέες για επεκτάσεις του σε μεγαλύτερες διαστάσεις.

3.2 Ο Αλγόριθμος QuickHull (*beneath-beyond*)

Στα τέλη της δεκαετίας του '70 παρουσιάστηκε ο πρώτος αλγόριθμος με πολυπλοκότητα που εξαρτάται από την έξοδο και ακούει στο όνομα QuickHull.

Ο αλγόριθμος αυτός σε κάθε βήμα προσθέτει μια κορυφή στο τμήμα του κυρτού περιγράμματος που έχει ήδη υπολογιστεί: δουλεύει δηλαδή αυξητικά (incremental). Εφαρμόζει την τακτική



Σχήμα 7: Τα έξι πρώτα βήματα του αλγορίθμου QuickHull για δεδομένο σύνολο σημείων στο επίπεδο. **a)** Επιλέγονται τα σημεία A , B και C ως αρχικό κυρτό περίγραμμα. **b)** Για τις σχηματιζόμενες πλευρές F_1 , F_2 και F_3 , δημιουργούνται τα αντίστοιχα σύνολα εξωτερικών σημείων και σημειώνονται τα ακρότατα σημεία: για την F_1 είναι το σημείο p . **c, d)** Για την πλευρά F_1 , από το p και τον οριζοντα του (ακμή \overline{AB}) σχηματίζονται δυο νέες πλευρές, οι οποίες προστίθενται στο περίγραμμα, από το οποίο αφαιρείται η πλευρά F_1 . **e)** Η διαδικασία συνεχίζεται για τη νέα πλευρά F'_1 : ακρότατο σημείο είναι το p' , μέσω του οποίου προστίθενται δυο ακόμη πλευρές \overline{ED} και \overline{EB} , ενώ αφαιρείται η \overline{BD} . **f)** Ο αλγόριθμος συνεχίζεται για την πλευρά F_3 .

beneath-beyond [Grü67, BDH96] για τον εντοπισμό της επόμενης πιθανής κορυφής, αναζητώντας την πιο απομακρυσμένη σύμφωνα με συγκεκριμένα κριτήρια.

Η πολυπλοκότητα του εξαρτάται από την έξοδο και είναι $O(n \log r)$ για $d \leq 3$, όπου r ο αριθμός των σημείων που επεξεργάζεται ο αλγόριθμος. Εν γένει τα σημεία αυτά είναι περισσότερα από h , από τα σημεία δηλαδή, τα οποία αποτελούν το κυρτό περίγραμμα –την έξοδο.

Ο αλγόριθμος εφαρμόζεται και σε μεγαλύτερες διαστάσεις, με αρκετά χειρότερη πολυπλοκότητα, βέβαια: $O(n \cdot f_r/r)$ για $d \geq 4$, όπου f_r ο μέγιστος αριθμός πλευρών για r σημεία. Σημειώνουμε ότι με τον όρο πλευρές αναφερόμαστε σε πολύγωνα $(d - 1)$ -διαστάσεων.

QuickHull

1976-96

ΕΙΣΟΔΟΣ: ένα σύνολο n σημείων στον d -διάστατο χώρο

ΕΞΟΔΟΣ: το υποσύνολο των σημείων αυτών, το οποίο αποτελεί το κυρτό περίγραμμα του αρχικού συνόλου

1. Επιλέγουμε τυχαία $d + 1$ σημεία και κατασκευάζουμε τις (τριγωνικές) πλευρές που σχηματίζονται από αυτά. Ο αριθμός των πλευρών αυτών είναι:

$$\frac{(d+1)!}{3! \cdot (d+1-3)!} = \frac{(d+1)!}{6 \cdot (d-2)!} = \frac{d}{6} \cdot (d^2 - 1)$$

Κατασκευάζεται έτσι, το (αρχικό) κυρτό περίγραμμα των $d + 1$ σημείων.

2. Για κάθε μία από τις σχηματιζόμενες πλευρές (*faces*) F_i :
 - ελέγχουμε τα εναπομείναντα σημεία. Αν βρίσκονται επάνω από την πλευρά, τοποθετούνται στο σύνολο εξωτερικών σημείων της $out(F_i)$ –σημειώνουμε ότι το κάθε σημείο εισάγεται σε ένα τέτοιο σύνολο και μόνον
3. Για κάθε πλευρά F_i με μη κενό σύνολο εξωτερικών σημείων $out(F_i)$:
 - βρίσκουμε το σημείο p του $out(F_i)$ με την μεγαλύτερη απόσταση από την πλευρά F_i
 - αρχικοποιούμε το σύνολο των ορατών πλευρών V_p από το p ως εξής: $V_p = \{F_i\}$
 - όσες από τις γειτονικές πλευρές των πλευρών του V_p βρίσκονται κάτω από το p , τοποθετούνται επίσης, στο V_p (δημιουργείται έτσι η ευρύτερη ορατή περιοχή πλευρών από το σημείο p)
 - το σύνορο του V_p λέγεται *ορίζοντας* του p , και δεν είναι άλλο από το σύνολο των ακμών, οι οποίες περιβάλλουν τις ορατές (από το p) πλευρές· έστω H το σύνολο αυτό
 - για κάθε ακμή R του H :
 - κατασκευάζεται μια νέα πλευρά από τα άκρα του R και το p , η οποία και συνδέεται με τις γειτονικές τις πλευρές
 - για κάθε μία από τις νέες πλευρές $F_j^!$ που δημιουργούνται:
 - ελέγχονται όλα τα εναπομείναντα σημεία των συνόλων εξωτερικών σημείων των πλευρών που περιέχονται στο V_p : αν βρίσκονται επάνω από την πλευρά $F_j^!$, τοποθετούνται στο σύνολο εξωτερικών της σημείων $out(F_j^!)$
 - αφαιρούνται από το κυρτό περίγραμμα οι πλευρές που ανήκουν στο V_p
4. Αν υπάρχει πλευρά με μη κενό σύνολο εξωτερικών σημείων, το βήμα 3 επαναλαμβάνεται· σε διαφορετική περίπτωση, το ζητούμενο κυρτό περίγραμμα έχει υπολογιστεί.

Ο αλγόριθμος επεξηγείται στο σχήμα 7, όπου εφαρμόζεται σε ένα δεδομένο σύνολο σημείων στο επίπεδο ($d = 2$).

3.3 Απλοποιημένος “Ultimate Planar Convex Hull” Αλγόριθμος

Τελειώνοντας με τις δύο διαστάσεις, παρουσιάζουμε δύο αλγορίθμους οι οποίοι επιτυγχάνουν την χαμηλότερη πολυπλοκότητα $O(n \log h)$.

Ο πρώτος, παρουσιάστηκε το 1986 από τους Kirkpatrick και Seidel [KS86], με τίτλο “The Ultimate Planar Convex Hull Algorithm”, ενώ ο δεύτερος αποτελεί εξέλιξη του Jarvis’s March. Όσον αφορά την πρώτη μέθοδο, θα παρουσιάσουμε μια παραλλαγή της, η οποία αποτελεί απλοποίηση της μεθόδου και οφείλεται στον Chan [Cha95]. Η πολυπλοκότητα της είναι επίσης $O(n \log h)$. Στην βιβλιογραφία αναφέρεται ως *Simplified “Ultimate Planar Convex Hull Algorithm”*.

Όπως θα φανεί στη συνέχεια, κατά την εν λόγω μέθοδο, το κυρτό περίγραμμα υπολογίζεται σε δύο φάσεις. Στην πρώτη, φάση υπολογίζεται το άνω κυρτό περίγραμμα και στη δεύτερη φάση το κάτω. Οι φάσεις είναι εντελώς αντίστοιχες, γι’ αυτό και περιοριζόμαστε στην παρουσίαση της μιας φάσης μόνο.

Ως P θεωρούμε το σύνολο εισόδου· βοηθητικά ορίζονται ως \mathbf{p}_l και \mathbf{p}_r τα δύο ακραία σημεία (δεξί και αριστερό αντίστοιχα) του συνόλου εισόδου P . Ορίζεται επίσης, το σύνολο P^\bullet των σημείων του P εκτός των \mathbf{p}_l και \mathbf{p}_r : $P^\bullet = P - \{\mathbf{p}_l, \mathbf{p}_r\}$. Για δυο σημεία \mathbf{p} και \mathbf{q} , ως $\overline{\mathbf{p}\mathbf{q}}$ ορίζεται το ευθύγραμμο τμήμα που ενώνει τα δύο σημεία και ως p_x, p_y οι συντεταγμένες x και y αντίστοιχα του σημείου \mathbf{p} . Σημειώνουμε ότι ο αλγόριθμος λειτουργεί αναδρομικά.

SimplUltConvHull_2D($P^\bullet, \mathbf{p}_l, \mathbf{p}_r$)

1986-95

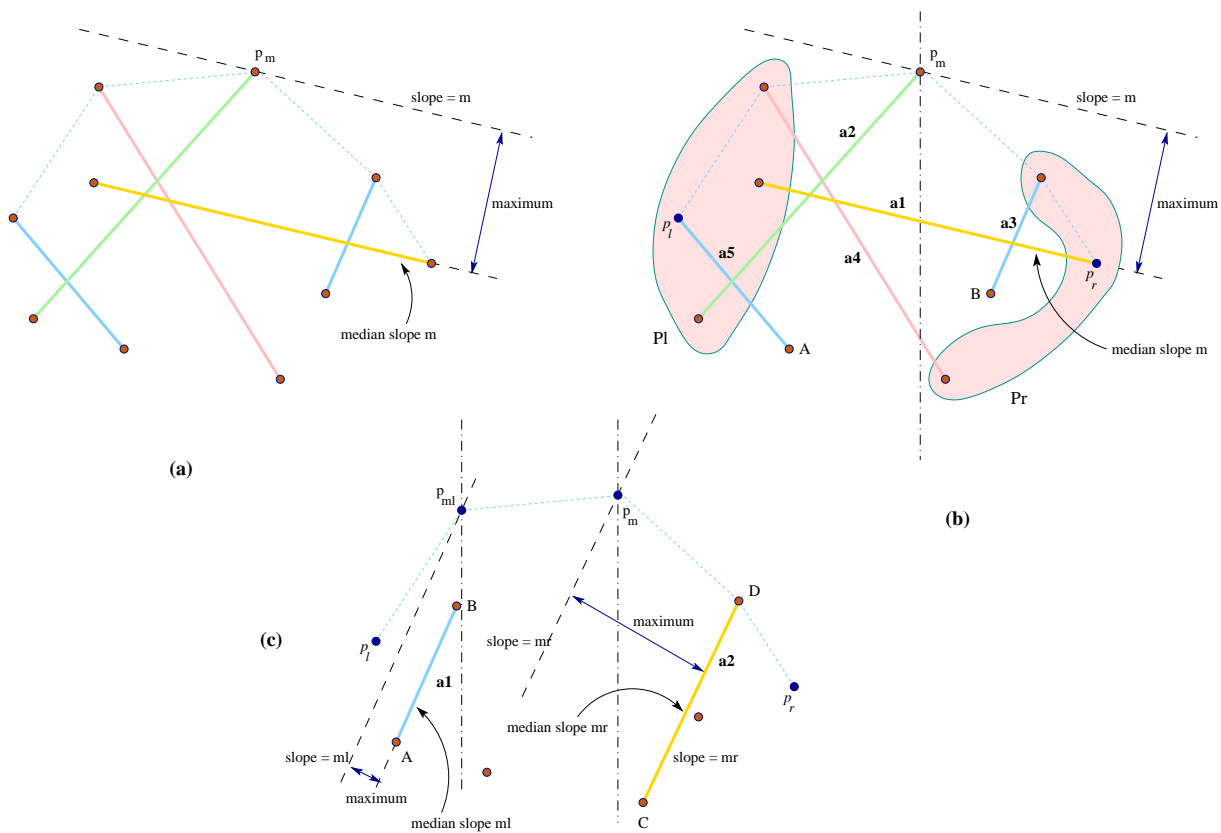
ΕΙΣΟΔΟΣ: δεδομένου συνόλου σημείων $P \subseteq E^d$, με $|P| = n$ και $P^\bullet = P - \{\mathbf{p}_l, \mathbf{p}_r\}$

ΕΞΟΔΟΣ: μια ακολουθία ακμών, η οποία αποτελεί το άνω κυρτό περίγραμμα του συνόλου P

1. Αφαιρούνται από το P^\bullet όλα τα σημεία τα οποία βρίσκονται κάτω από το $\overline{\mathbf{p}_l \mathbf{p}_r}$
2. Ελέγχεται το σύνολο P^\bullet :
 - αν $P = \emptyset$, τότε επιστρέφεται το $\langle \overline{\mathbf{p}_l \mathbf{p}_r} \rangle$
 - αν $P = \{\mathbf{p}\}$, τότε επιστρέφεται το $\langle \overline{\mathbf{p}_l \mathbf{p}}, \overline{\mathbf{p}_r \mathbf{p}} \rangle$
3. Επιλέγονται με τυχαίο τρόπο $\lfloor n/2 \rfloor$ ξεχωριστά ζευγάρια σημείων $\{\{s_1, t_1\}, \dots, \{s_{\lfloor n/2 \rfloor}, t_{\lfloor n/2 \rfloor}\}\}$ από το P^\bullet και ταξινομούνται ούτως ώστε: $s_{ix} < t_{ix}$
4. Έστω $m_i = (t_{iy} - s_{iy}) / (t_{ix} - s_{ix})$, $i = 1, 2, \dots, \lfloor n/2 \rfloor$ και m ο μέσος των $\langle m_1, m_2, \dots, m_{\lfloor n/2 \rfloor} \rangle$
5. Έστω \mathbf{p}_m το σημείο του P το οποίο μεγιστοποιεί τη διαφορά $p_{my} - m \cdot p_{mx}$
6. Έστω $P_l^\bullet = \{\mathbf{p} \in P^\bullet : p_x < p_{mx}\} - \{t_i : m_i \leq m\}$ και $P_r^\bullet = \{\mathbf{p} \in P^\bullet : p_x > p_{mx}\} - \{s_i : m_i \geq m\}$

7. Ελέγχεται το p_m :

- αν $p_m = p_r$, τότε επιστρέφεται το $\text{SimplUltConvHull}_{2D}(P_l^*, p_l, p_r)$
- αν $p_m = p_l$, τότε επιστρέφεται το $\text{SimplUltConvHull}_{2D}(P_r^*, p_l, p_r)$
- διαφορετικά, επιστρέφεται η ένωση των $\text{SimplUltConvHull}_{2D}(P_l^*, p_l, p_m)$ και $\text{SimplUltConvHull}_{2D}(P_r^*, p_m, p_r)$



Σχήμα 8: Τα τρία πρώτα βήματα του Απλοποιημένου “Ultimate Convex Hull Algorithm” για δεδομένο σύνολο σημείων στο επίπεδο. **α)** Το σύνολο των σημείων χωρίζεται σε ζεύγη, τα οποία σχηματίζουν ευθύγραμμα τμήματα με αντίστοιχες κλίσεις. Υπολογίζεται το ευθύγραμμο τμήμα με τη μέση κλίση m καθώς και το σημείο p_m με την μεγαλύτερη απόσταση από το τμήμα αυτό. **β)** Με βάση το σημείο p_m χωρίζεται το αρχικό σύνολο P στα P_l και P_r , ανάλογα με τις κλίσεις των τμημάτων που ανήκουν και τη θέση τους ως προς το p_m . **γ)** Η διαδικασία επαναλαμβάνεται για τα δύο υποσύνολα (divide-and-conquer).

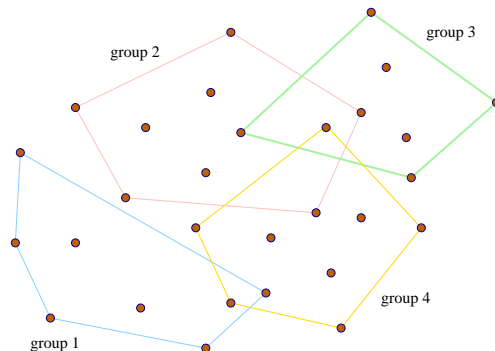
Στο σχήμα 8 εξηγείται ο παραπάνω αλγόριθμος για ένα σύνολο 10 σημείων στις δύο διαστάσεις.

Ο αλγόριθμος εφαρμόζεται στα επιμέρους υποσύνολα του αρχικού. Παρουσιάζονται οι δύο πρώτες επαναλήψεις.

3.4 Ο Αλγόριθμος “Group-and-Wrap” στο Επίπεδο

Ολοκληρώνουμε την αναφορά μας με έναν διαφορετικό $O(n \log h)$ αλγόριθμο. Αποτελεί παραλλαγή του Jarvis’s March και της μεθόδου τυλίγματος, προσθέτοντας ένα ενδιάμεσο βήμα ομαδοποίησης, το οποίο και βελτιώνει την πολυπλοκότητα. Ο αλγόριθμος παρουσιάστηκε από τον Chan [Cha95] και είναι γρηγορότερος από τον προηγούμενο γιατί δεν κάνει πράξεις εύρεσης μέσων (εύρεση μέσης κλίσης, κ.λπ.).

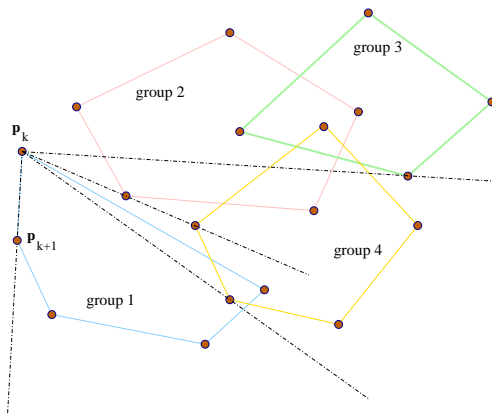
Έστω $P \subseteq N^2$ ένα σύνολο $n \geq 3$ σημείων. Ο αλγόριθμος Jarvis’s March [Jar73, O’R94, PS85], υπολογίζει τις h κορυφές του κυρτού περιγράμματος, με φορά αντίθετη των δεικτών του ρολογιού, με μια σειρά από βήματα τυλίγματος, ως εξής: αν \mathbf{p}_{k-1} και \mathbf{p}_k είναι οι δύο τελευταίες κορυφές που προστέθηκαν, η επόμενη κορυφή \mathbf{p}_{k+1} θα είναι η κορυφή $\mathbf{p} \in P$, η οποία μεγιστοποιεί τη γωνία $\angle \mathbf{p}_{k-1} \mathbf{p}_k \mathbf{p}$, με $\mathbf{p} \neq \mathbf{p}_k$. Το κάθε επιμέρους βήμα τυλίγματος μπορεί να πραγματοποιηθεί, προφανώς, σε $O(n)$ χρόνο, διατρέχοντας όλα τα n σημεία μια φορά· με κατάλληλη αρχικοποίηση, ολόκληρο το κυρτό περίγραμμα κατασκευάζεται σε $O(nh)$ χρόνο.



Σχήμα 9: Το σύνολο P χωρίζεται σε $\lceil n/m \rceil$ ομάδες-υποσύνολα και για κάθε μία υπολογίζεται το κυρτό της περίγραμμα.

Η βασική ιδέα αυτής της παραλλαγής είναι: να γίνει κάποια προεπεξεργασία στα σημεία του P , ώστε το βήμα τυλίγματος να πραγματοποιηθεί γρηγορότερα. Συγκεκριμένα, επιλέγεται μια παράμετρος m , από 1 έως n και χωρίζεται το σύνολο P σε $\lceil n/m \rceil$ ομάδες m το πολύ σημείων (σχήμα 9). Κατόπιν, το κυρτό περίγραμμα της κάθε ομάδας μπορεί να υπολογιστεί σε χρόνο $O(m \log m)$, με οποιαδήποτε μέθοδο, π.χ. την Graham’s scan. Παίρνουμε έτσι $\lceil n/m \rceil$ πιθανώς

επικαλυπτόμενα πολύγωνα, με m κορυφές το πολύ το καθένα, μετά από χρόνο προεπεξεργασίας $O(\frac{n}{m} \cdot (m \log m)) = O(n \log m)$.



Σχήμα 10: Το επόμενο σημείο, επιλέγεται ανάμεσα σ' αυτά που σχηματίζουν τη μεγαλύτερη γωνία από το κάθε πολύγωνο.

Έτσι, το κάθε βήμα τυλίγματος μπορεί να γίνει σαρώνοντας όλα τα $\lceil n/m \rceil$ πολύγωνα και υπολογίζοντας εφαπτομένες, όπως φαίνεται στο σχήμα 10. Δεδομένου ότι ο υπολογισμός εφαπτομένων για κυρτό πολύγωνο μπορεί να γίνει σε λογαριθμικό χρόνο [PS85], ο απαιτούμενος χρόνος για το κάθε βήμα είναι $O(\frac{n}{m} \log m)$. Ο αριθμός των βημάτων είναι h , οπότε συνολικά απαιτείται χρόνος $O(n \log m + h(\frac{n}{m} \log m)) = O(n(1 + h/m) \log m)$.

Ο ψευδοκώδικας του αλγορίθμου παρουσιάζεται στη συνέχεια, με αριθμό βημάτων $H \in \mathbb{N}$. Επιστρέφει επιτυχημένα το κυρτό περίγραμμα του P –με φορά αντίθετη των δεικτών του ρολογιού– όταν $H \geq h$.

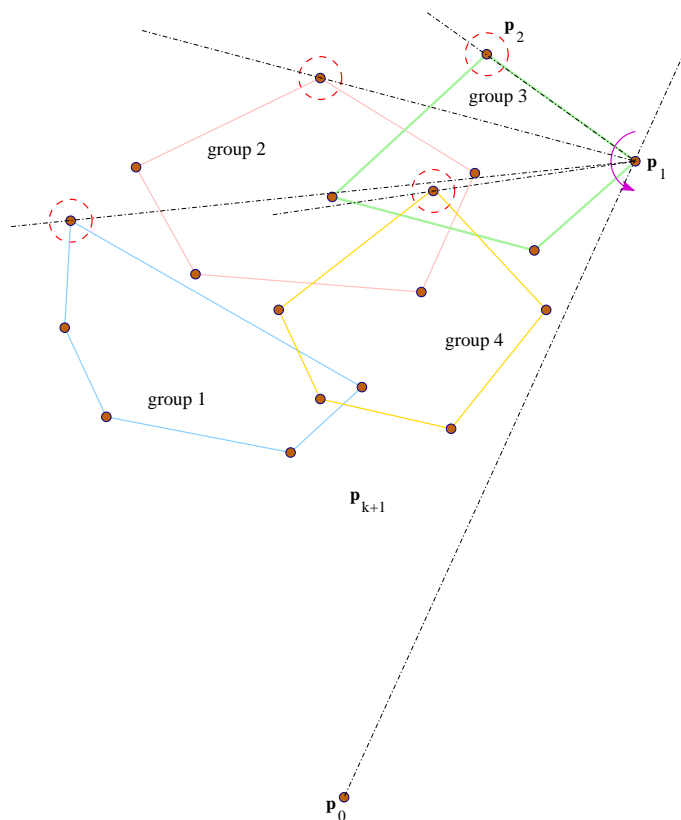
GroupHull2D(P, m, H)

ΕΙΣΟΔΟΣ: σύνολο $P \in \mathbb{N}^2$ με n σημεία, ακέραιοι m και H , όπου $3 \leq m \leq n$ και $H \geq 1$

ΕΞΟΔΟΣ: λίστα σημείων που αποτελούν το κυρτό περίγραμμα του P , με φορά αντίθετη των δεικτών του ρολογιού

1. το σύνολο P χωρίζεται σε υποσύνολα $P_1, \dots, P_{\lceil n/m \rceil}$, κάθε ένα από τα οποία έχει μέγεθος το πολύ m
2. για $i = 1, \dots, \lceil n/m \rceil$
υπολογίζεται το κυρτό περίγραμμα $conv(P_i)$ με Graham's scan και οι κορυφές του αποθηκεύονται σε πίνακα με φορά αντίθετη των δεικτών του ρολογιού
3. $p_0 \leftarrow (0, -\infty)$
4. $p_1 \leftarrow$ το δεξιότερο σημείο του P

5. για $k = 1, \dots, H$
 - για $i = 1, \dots, \lfloor n/m \rfloor$
 - υπολογίζεται το σημείο $\mathbf{q}_i \in P_i$ το οποίο μεγιστοποιεί τη $\angle \mathbf{p}_{k-1} \mathbf{p}_k \mathbf{q}_i$ ($\mathbf{q}_i \neq \mathbf{p}_k$), πραγματοποιώντας δυαδική αναζήτηση στις κορυφές του $\text{conv}(P_i)$
 - $\mathbf{p}_{k+1} \leftarrow$ το σημείο q από το $\{\mathbf{q}_1, \dots, \mathbf{q}_{\lfloor n/m \rfloor}\}$ που μεγιστοποιεί την $\angle \mathbf{p}_{k-1} \mathbf{p}_k \mathbf{q}$
 - αν $\mathbf{p}_{k+1} = \mathbf{p}_1$ επιστρέφεται $\langle \overline{\mathbf{p}_1 \mathbf{p}_2}, \dots, \overline{\mathbf{p}_{k-1} \mathbf{p}_k}, \overline{\mathbf{p}_k \mathbf{p}_1} \rangle$
6. επιστρέφεται *incomplete*



Σχήμα 11: Μετά τον υπολογισμό των επιμέρους κυρτών περιγραμμάτων των ομάδων του P , ξεκινώντας από το αριστερότερο σημείο του \mathbf{p}_1 , για κάθε σημείο, υπολογίζεται αυτό με το οποίο σχηματίζει τη μεγαλύτερη γωνία. Η αναζήτηση γίνεται πρώτα στο επίπεδο της κάθε ομάδας και έπειτα μεταξύ των μεγίστων των ομάδων.

Στο σχήμα 11 παρουσιάζονται τα αποτελέσματα των πρώτων επαναλήψεων του αλγορίθμου για δεδομένο σύνολο σημείων στο επίπεδο. Παρατηρούμε, στο σημείο αυτό, ότι ο αλγόριθμος δεν

ολοκληρώνεται όταν $H < h$. Επίσης, διαλέγοντας $m = H$, η πολυπλοκότητα του γίνεται $O(n(1 + H/m) \log m) = O(n \log H)$.

Λόγω του ότι η τιμή του h δεν είναι γνωστή εκ των προτέρων, χρησιμοποιούμε μια ακολουθία τιμών για να την προσεγγίσουμε, όπως φαίνεται στη συνέχεια (η ίδια τεχνική χρησιμοποιείται και στον αλγόριθμο των Chazelle και Matoušek [CM95]).

Αλγόριθμος Group-and-Wrap στο Επίπεδο

1995

ΕΙΣΟΔΟΣ: σύνολο $P \subseteq N^2$ με n σημεία

ΕΞΟΔΟΣ: λίστα σημείων που αποτελούν το κυρτό περίγραμμα του P , με φορά αντίθετη των δεικτών του ρολογιού

1. για $t = 1, 2, \dots$

$L \leftarrow \text{GroupHull2D}(P, m, H)$, όπου $m = H = \min\{2^{2^t}, n\}$
αν $L \neq \text{incomplete}$ επιστρέφεται L

Η διαδικασία σταματάει όταν η τιμή του H φτάσει το h . Ο αριθμός των επαναλήψεων είναι $\lceil \log(\log h) \rceil$ και η t επανάληψη χρειάζεται χρόνο $O(n \log H) = O(n \cdot 2^t)$. Έτσι, ο συνολικά απαιτούμενος χρόνος είναι:

$$O\left(\sum_{t=1}^{\lceil \log(\log h) \rceil} n \cdot 2^t\right) = O(n \cdot 2^{\lceil \log(\log h) \rceil + 1}) = O(n \log h),$$

όπως αποδεικνύεται στο [Cha95]. Αποδεικνύεται επίσης, ότι το κόστος όσον αφορά το χώρο είναι γραμμικό.

3.5 Συμπεράσματα για τις Δύο Διαστάσεις

Στο κεφάλαιο αυτό, αναφερθήκαμε στις κυριότερες μεθόδους υπολογισμού του κυρτού περιγράμματος συνόλου σημείων στο επίπεδο. Στο πίνακα που ακολουθεί, συνοψίζουμε τις μεθόδους αυτές, σημειώνοντας και τις αντίστοιχες πολυπλοκότητες που παρουσιάζουν.

Υπενθυμίζουμε στον αναγνώστη ότι το πλήθος των σημείων είναι n , h είναι το πλήθος των κορυφών του κυρτού περιγράμματος (της εξόδου) και r είναι ο αριθμός των πλευρών του.

Αλγόριθμος	Χρόνος Εκτέλεσης	Χρονολογία	Αναφορές
Graham's Scan	$O(n \log n)$	1967	[Gra67]
Jarvis's March	$O(nh)$	1973	[Jar73]
MergeHull	$O(n \log n)$	1977	[PH77]
QuickHull	$O(n \log r)$	1976-1996	[BDH96]
Simplified UCHA	$O(n \log h)$	1986, 1995	[KS86, Cha95]
Group-and-Wrap	$O(n \log h)$	1995	[Cha95]

Πίνακας 3.1: Κυριότεροι αλγόριθμοι για τις δύο διαστάσεις ($d = 2$).

Κεφάλαιο 4

Αλγόριθμοι Κυρτού Περιγράμματος στις Τρεις Διαστάσεις

Το πρόβλημα του κυρτού περιγράμματος συνόλου σημείων στις τρεις διαστάσεις ($d = 3$) παρουσιάζει εξίσου μεγάλο ενδιαφέρον και μελετήθηκε κι αυτό εξίσου εντατικά τα τελευταία χρόνια.

Οι Preparata και Hong παρουσίασαν τον πρώτο αλγόριθμο με πολυπλοκότητα $O(n \log n)$, για τις τρεις διαστάσεις, το 1977 [PH77]. Ο αλγόριθμος αυτός βασίζεται στην τεχνική *divide-and-conquer*. Ο πρώτος αλγόριθμος με εξάρτηση πολυπλοκότητας από την έξοδο, παρουσιάστηκε από τους Chand και Karur [CK70]. Βασίζεται στην μέθοδο *τυλίγματος* και αναφέρεται σε οποιαδήποτε διάσταση. Παρότι εμφανίστηκε πριν από τη μέθοδο Jarvis's March, αποτελεί γενίκευση της, για οποιαδήποτε διάσταση. Η μέθοδος αυτή αναλύθηκε από τον Swart, δεκαπέντε χρόνια αργότερα [Swa85], ο οποίος απέδειξε ότι απαιτεί χρόνο $O(nh)$.

Για αρκετό καιρό η μέθοδος τυλίγματος ήταν ο μόνος γνωστός αλγόριθμος με εξάρτηση εξόδου. Η συνέχεια δόθηκε από τους Clarkson και Shor [CS89], ως εφαρμογή κάποιων πιθανοτικών τεχνικών στην υπολογιστική γεωμετρία. Παρουσίασαν έναν αλγόριθμο με αναμενόμενη πολυπλοκότητα τη βέλτιστη $O(n \log h)$. Το αποτέλεσμα αυτό οφείλεται κυρίως, στις τυχαίες επιλογές του αλγορίθμου [Mul93].

Στη συνέχεια, οι Edelsbrunner και Shi [ES91], πρότειναν έναν ντετερμινιστικό αλγόριθμο με πολυπλοκότητα $O(n \log^2 h)$, ακολουθώντας το παράδειγμα των Kirkpatrick και Seidel. Τελικά, ένας βέλτιστος $O(n \log h)$ ντετερμινιστικός αλγόριθμος επετεύχθη όταν οι Chazelle και Matoušek εφάρμοσαν ορισμένες πρόσφατες τεχνικές *derandomization* στον αλγόριθμο των Clarkson και Shor [CM95].

Σημειώνουμε ακόμα τον πρόσφατο πιθανοτικό $O(n \log^2 h)$ αλγόριθμο που προτάθηκε από τον Clarkson [Cla94], ο οποίος μπορεί επίσης να μετατραπεί σε ντετερμινιστικό, σύμφωνα με τον Chan [Cha95]. Οι δύο αυτοί αλγόριθμοι, όπως και οι περισσότεροι που δημιουργούνται με αυτόν τον τρόπο (derandomization), δεν είναι ιδιαίτερα πρακτικοί, λόγω της πολυπλοκότητας της διαδικασίας.

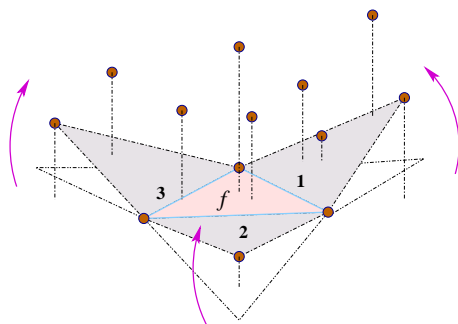
Τέλος, παρουσιάστηκε από τον Chan μια επέκταση του αλγορίθμου group-and-wrap, για τις τρεις διαστάσεις, με βέλτιστη πολυπλοκότητα $O(n \log h)$ [Cha95].

Στο κεφάλαιο αυτό θα παρουσιάσουμε τις αντιπροσωπευτικότερες από τις παραπάνω μεθόδους, όπως επίσης και την επέκταση του αλγορίθμου QuickHull στις τρεις διαστάσεις.

4.1 Επέκταση του Jarvis's March

Το ανάλογο του Jarvis's March σε παραπάνω διαστάσεις, είναι ο αλγόριθμος των Chand και Karim, ο οποίος ονομάζεται μέθοδος *gift-wrapping* [CK70, PS85, Swa85]. Ο αλγόριθμος αυτός, παρόλο που παρουσιάστηκε το 1970, αναλύθηκε δεκαπέντε χρόνια αργότερα από τον Swart, ο οποίος και απέδειξε ότι απαιτεί χρόνο $O(nh)$ [Swa85].

Έστω $P \subseteq E^3$ ένα σύνολο $n \geq 4$ σημείων. Θεωρώντας ότι τα σημεία είναι σε γενική θέση, ξέρουμε από τον τύπο του Euler, ότι το κυρτό περίγραμμα αποτελείται από ακριβώς $2h - 4$ πλευρές (τριγωνικές), αν h οι κορυφές του.



Σχήμα 12: Η μέθοδος τυλίγματος για τις τρεις διαστάσεις. Δεδομένης της πλευράς f , οι παράπλευρες τρεις πλευρές δημιουργούνται από τις ακμές της f . Τα διακεκομμένα κατακόρυφα τμήματα έχουν τοποθετηθεί για τη διευκόλυνση της τρισδιάστατης απεικόνισης.

Οι πλευρές του κυρτού περιγράμματος υπολογίζονται μία-μία ως εξής: από μια δεδομένη (τριγωνική) πλευρά f , υπολογίζονται οι τρεις γειτονικές f_j , εφαρμόζοντας ένα βήμα τυλίγματος,

ως προς κάθε ακμή e_j της f ($j = 1, 2, 3$). Το κάθε βήμα τυλίγματος γύρω από την ακμή e_j , είναι να εντοπιστεί το σημείο $\mathbf{p}_j \in P$ τέτοιο ώστε, η γωνία μεταξύ της f και του $\text{conv}(e_j \cup \{\mathbf{p}_j\})$, $\mathbf{p}_j \notin e_j$, να μεγιστοποιείται (σχήμα 12).

Δεδομένου του ότι ένα τέτοιο βήμα μπορεί να γίνει σε χρόνο $O(n)$, όλες οι γειτονικές πλευρές της f μπορούν να υπολογιστούν σε χρόνο $O(n)$. Αν λοιπόν, μια αρχική πλευρά f_0 δίνεται (μπορεί να κα βρεθεί σε δύο βήματα τυλίγματος), κάνοντας μια αναζήτηση breadth-first ή depth-first, είναι δυνατόν να παράγουμε όλες τις πλευρές του κυρτού περιγράμματος. Τυχόν διπλές προσπελάσεις πλευρών, μπορούν να αποφευχθούν χρησιμοποιώντας ένα λεξικό. Συμπεραίνουμε, λοιπόν, ότι ο αλγόριθμος πραγματοποιεί $3(2h - 4)$ βήματα τυλίγματος και άρα απαιτεί χρόνο $O(nh)$.

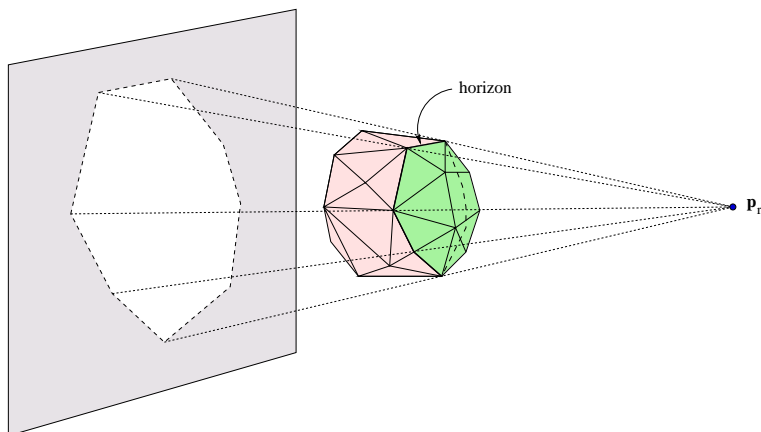
4.2 Ο Αλγόριθμος QuickHull στις Τρεις Διαστάσεις

Ο αλγόριθμος είναι ο ίδιος, ο οποίος παρουσιάστηκε για τις δύο διαστάσεις. Η πολυπλοκότητα του και για τις τρεις διαστάσεις είναι $O(n \log r)$, για σύνολο σημείων P μεγέθους n , όπου r τα σημεία που επεξεργάζεται ο αλγόριθμος. Θα περιγράψουμε τις διαφορές της διαδικασίας, από την αντίστοιχη στις δύο διαστάσεις, περισσότερο σχηματικά. Τόσο ο αλγόριθμος στις τρεις διαστάσεις, όσο και κάποιες δομές δεδομένων για την πρακτική υλοποίησή του, περιγράφονται αναλυτικά στα [BDH96, dBvKOS97].

Η κύρια διαφορά με τις δύο διαστάσεις έγκειται στον καθορισμό της ορατής περιοχής και του οριζόντα (βλέπε σχήμα 13).

Θεωρούμε σύνολο $P = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_n\} \subset E^3$. Ο αλγόριθμος αρχίζει επιλέγοντας τέσσερα σημεία του P , μη συνεπίπεδα. Αυτό μπορεί να γίνει πολύ εύκολα, ες εξής. Επιλέγονται τα δύο πρώτα σημεία του P , \mathbf{p}_1 και \mathbf{p}_2 . Έπειτα, διατρέχονται τα υπόλοιπα σημεία έως ότου να βρεθεί ένα σημείο \mathbf{p}_3 μη συνευθειακό με τα δύο πρώτα. Κατόπιν, αναζητείται το τέταρτο σημείο \mathbf{p}_4 , για το οποίο απαιτείται να μην είναι συνεπίπεδο με τα τρία πρώτα. Χωρίς βλάβη της γενικότητας, θεωρούμε ότι τα πρώτα τέσσερα σημεία του P πληρούν τα παραπάνω –είναι μη συνεπίπεδα. Θεωρούμε, επίσης, ότι τα εναπομείναντα σημεία $\mathbf{p}_5, \dots, \mathbf{p}_n$ είναι σε τυχαία σειρά.

Ο αλγόριθμος λειτουργεί αυξητικά (incrementally), δουλεύοντας με ένα σημείο από τα εναπομείναντα του P , κάθε φορά. Για δεδομένο ακέραιο $r \geq 1$, έστω $P_r = \{\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_r\}$. Σε κάποιο βήμα του αλγορίθμου, έστω r , πρέπει να προστεθεί το σημείο \mathbf{p}_r στο κυρτό περίγραμμα του P_{r-1} , με άλλα λόγια, να μετασχηματιστεί το $\text{conv}(P_{r-1}) = \mathcal{CH}(P_{r-1})$ σε $\text{conv}(P_r) = \mathcal{CH}(P_r)$. Προφανώς, προκύπτουν δύο περιπτώσεις:



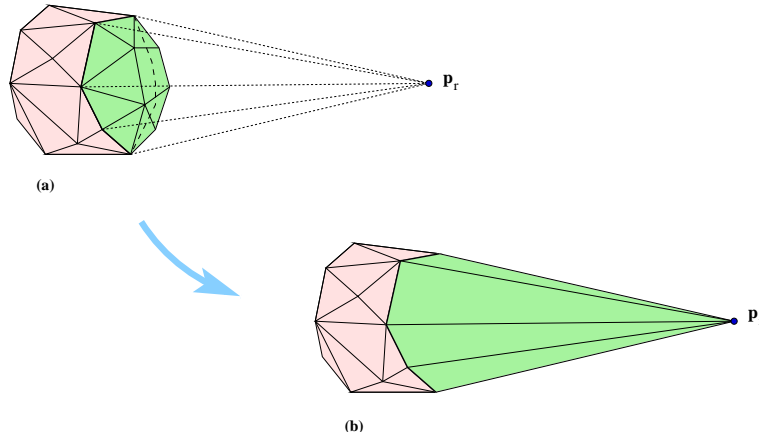
Σχήμα 13: Ο ορίζοντας και η ορατή περιοχή από ένα σημείο σε ένα πολύτοπο τριών διαστάσεων.

- Αν το \mathbf{p}_r βρίσκεται εντός του $\mathcal{CH}(P_{r-1})$ ή επάνω στο σύνορο του, τότε $\mathcal{CH}(P_{r-1}) = \mathcal{CH}(P_r)$, οπότε δεν απαιτείται να γίνει κάτι επιπλέον.
- Υποθέτουμε, τώρα, ότι το \mathbf{p}_r βρίσκεται εκτός του $\mathcal{CH}(P_{r-1})$. Στην περίπτωση αυτή, βλέποντας από το \mathbf{p}_r το κυρτό περίγραμμα $\mathcal{CH}(P_{r-1})$, κάποιες πλευρές του θα είναι ορατές και κάποιες όχι. Οι ορατές πλευρές σχηματίζουν μια επιφάνεια, η οποία καλείται ορατή περιοχή του \mathbf{p}_r στο κυρτό περίγραμμα $\mathcal{CH}(P_{r-1})$. Το περίγραμμα αυτής της περιοχής στο $\mathcal{CH}(P_{r-1})$ καλείται ορίζοντας του \mathbf{p}_r (βλέπε σχήμα 13).

Ο ορίζοντας του \mathbf{p}_r αποτελεί καθοριστικό παράγοντα για τη μετατροπή του $\mathcal{CH}(P_{r-1})$ σε $\mathcal{CH}(P_r)$: αποτελεί το όριο μεταξύ του τμήματος του $\mathcal{CH}(P_{r-1})$ το οποίο παραμένει –οι μη ορατές πλευρές– και του τμήματος το οποίο πρέπει να αντικατασταθεί –οι ορατές περιοχές. Οι ορατές πλευρές πρέπει να αντικατασταθούν από νέες, οι οποίες θα συνδέουν το σημείο \mathbf{p}_r με τις ακμές του ορίζοντα (σχήμα 14).

Το πρόβλημα το οποίο ανακύπτει είναι ο αποδοτικός προσδιορισμός των ορατών περιοχών, για δεδομένο σημείο του P . Η προφανής λύση του προβλήματος, είναι να ελέγξουμε το σημείο το οποίο θέλουμε να προσθέσουμε, ως προς κάθε πλευρά του ήδη δημιουργημένου κυρτού περιγράμματος. Για το σημείο \mathbf{p}_r απαιτείται κόστος $O(r)$, δηλαδή συνολικά έχουμε πολυπλοκότητα $O(n^2)$.

Το εν λόγω πρόβλημα μπορεί να λυθεί αποδοτικά κρατώντας επιπλέον πληροφορίες, τόσο για τα σημεία του P , όσο και για τις πλευρές του $\mathcal{CH}(P_{r-1})$. Οι πληροφορίες αυτές θα πρέπει να ενημερώνονται σε κάθε βήμα του αλγορίθμου. Πολλοί ερευνητές για το σκοπό αυτό, χρησιμοποιούν



Σχήμα 14: Δημιουργία του $\mathcal{CH}(P_r)$ από το $\mathcal{CH}(P_{r-1})$, προσθέτοντας το σημείο \mathbf{p}_r . Αφού βρεθεί ο ορίζοντας του \mathbf{p}_r στο $\mathcal{CH}(P_{r-1})$, αφαιρείται η ορατή περιοχή από το κυρτό περίγραμμα (a). Κατόπιν, προστίθενται νέες πλευρές, οι οποίες σχηματίζονται από το σημείο \mathbf{p}_r και τις ακμές του ορίζοντα (b).

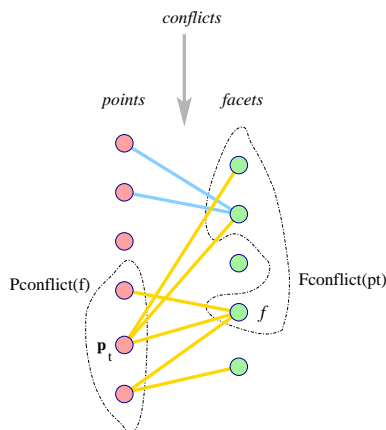
μια επιπλέον δομή, η οποία καλείται *γράφος συγκρούσεων* (*conflict-graph*).¹

Για κάθε πλευρά f του τρέχοντος κυρτού περιγράμματος $\mathcal{CH}(P_r)$, αποθηκεύεται το σύνολο $P_{\text{conflict}}(f) \subseteq \{\mathbf{p}_{r+1}, \mathbf{p}_{r+2}, \dots, \mathbf{p}_n\}$, το οποίο περιέχει τα σημεία τα οποία βλέπουν την f (και δεν έχουν προστεθεί ακόμη στο κυρτό περίγραμμα). Αντίστοιχα, για κάθε σημείο \mathbf{p}_t , με $t > r$ (σημείο δηλαδή που δεν έχει ελεγχθεί ακόμη για το κυρτό περίγραμμα), αποθηκεύεται το σύνολο $F_{\text{conflict}}(\mathbf{p}_t)$ των πλευρών του $\mathcal{CH}(P_r)$, οι οποίες είναι ορατές από αυτό. Λέμε ότι ένα σημείο $\mathbf{p} \in P_{\text{conflict}}(f)$ είναι σε σύγκρουση με την πλευρά f , γιατί δεν μπορούν να συνυπάρξουν στο κυρτό περίγραμμα – με την πρόσθεση οποιοδήποτε σημείου $\mathbf{p} \in P_{\text{conflict}}(f)$, η πλευρά f πρέπει να αφαιρεθεί.

Τα σύνολα συγκρούσεων διατηρούνται στον γράφο συγκρούσεων \mathcal{G} , ως εξής. Πρόκειται για διμερή (bipartite) γράφο, ο οποίος περιέχει ένα σύνολο κόμβων για τα σημεία του P , τα οποία δεν έχουν εισαχθεί ακόμη στο κυρτό περίγραμμα και ένα δεύτερο σύνολο κόμβων για τις πλευρές του μέχρι τώρα κυρτού περιγράμματος. Οι συγκρούσεις μεταξύ σημείων και πλευρών σημειώνονται ως ακμές του γράφου (σχήμα 15).

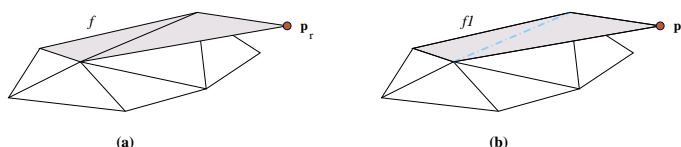
Στην παραπάνω ανάλυση υπάρχει ένα σημείο στο οποίο δεν έγινε αναφορά: τί γίνεται στην περίπτωση κατά την οποία προστίθεται μια πλευρά συνεπίπεδη με μια ήδη υπάρχουσα. Στην περίπτωση αυτή θα πρέπει η νέα τριγωνική πλευρά να ενωθεί με την ήδη υπάρχουσα. Στο σχήμα 16 παρουσιάζεται μια τέτοια περίπτωση. Η νέα τριγωνική πλευρά θα πρέπει να ενωθεί με

¹Η δομή αυτή περιγράφεται αναλυτικά στο [dBvKOS97].



Σχήμα 15: Γράφος συγκρούσεων κατά τη διάρκεια κατασκευής κυρτού περιγράμματος συνόλου σημείων. Για το σημείο p_t , σημειώνεται το σύνολο $F_{conflict}(p_t)$ και αντίστοιχα για την πλευρά f , το σύνολο $P_{conflict}(f)$.

την υπάρχουσα συνεπίπεδη της f και να αφαιρεθούν οι επιπλέον ακμές.



Σχήμα 16: Συνεπίπεδες πλευρές κατά την δημιουργία κυρτού περιγράμματος. Η νέα τριγωνική πλευρά ενώνεται με την υπάρχουσα f και δημιουργείται η νέα (μη τριγωνική πλέον) πλευρά f_1 .

Η τυπική περιγραφή του αλγορίθμου QuickHull παραμένει αυτή που παρουσιάστηκε για τις δύο διαστάσεις, για $d = 2$. Η πολυπλοκότητα παραμένει $O(n \log r)$, όπου r τα σημεία που επεξεργάζεται ο αλγόριθμος.

4.3 Επέκταση του Group-and-Wrap στις Τρεις Διαστάσεις

Αντίστοιχα με τη μέθοδο group-and-wrap, ο Chan παρουσίασε μια επέκταση της στις τρεις διαστάσεις [Cha95]. Επεκτείνοντας την ιδέα της ομαδοποίησης και χρησιμοποιώντας κάποιες κατάλληλες δομές δεδομένων, κατάφερε να διατηρήσει το βέλτιστο χρόνο $O(n \log h)$ και στις τρεις διαστάσεις, διατηρώντας βέβαια, την απαίτηση για γραμμικό χώρο.

Αντί της χρησιμοποίησης του Graham's scan για την εύρεση των επιμέρους κυρτών περιγραμ-

μάτων των ομάδων (βήμα 2 του $\text{GroupHull12D}(P, m, H)$), χρησιμοποίησε τον αλγόριθμο των Preparata και Hong για κυρτά περιγράμματα τριών διαστάσεων [PH77], ο οποίος παρουσιάζει την ίδια πολυπλοκότητα $O(m \log m)$, για σύνολο m σημείων.

Το κυρίως πρόβλημα έγκειται στην επέκταση, στις τρεις διαστάσεις, του εντοπισμού του επόμενου σημείου (αυτού που σχηματίζει τη μεγαλύτερη γωνία με το τρέχον και το προηγούμενο –βήμα 5 του $\text{GroupHull12D}(P, m, H)$). Στις δύο διαστάσεις αυτό γίνεται με δυαδική αναζήτηση.

Για τις τρεις διαστάσεις, απαιτείται ο υπολογισμός εφαπτομένων γωνιών, μεταξύ τρισδιάστατων πολυτόπων και ευθειών. Η μέθοδος που θα χρησιμοποιήσουμε απαιτείται να έχει λογαριθμικό κόστος, ώστε να διατηρηθεί η ίδια συνολική πολυπλοκότητα των δύο διαστάσεων. Μια δομή δεδομένων, η οποία επιτρέπει ακριβώς αυτό, είναι η *πολυεδρική ιεραρχία (polyedral hierarchy)* των Dobkin και Kirkpatrick [DK83, DK90]. Στην παρούσα εργασία περιγράφεται εν συντομία στο τελευταίο κεφάλαιο.

Ο ψευδοκώδικας που περιγράφει τον αλγόριθμο δίνεται στη συνέχεια. Χρησιμοποιώντας την ιεραρχική αναπαράσταση των Dobkin και Kirkpatrick για την αποθήκευση των πολυτόπων στις τρεις διαστάσεις, επιτυγχάνεται η εκτέλεση του βήματος 4 σε λογαριθμικό χρόνο, για κάθε πολύτοπο $\mathcal{P}_i = \text{conv}(P_i)$. Τα υπόλοιπα βήματα του αλγορίθμου είναι ίδια με τα αντίστοιχα των δύο διαστάσεων.

GroupHull3D(P, m, H)

ΕΙΣΟΔΟΣ: σύνολο $P \subseteq E^3$ με n σημεία, ακέραιοι m και H , όπου $4 \leq m \leq n$ και $H \geq 1$

ΕΞΟΔΟΣ: λίστα σημείων που αποτελούν το κυρτό περίγραμμα του P , με φορά αντίθετη των δεικτών του ρολογιού

1. το P χωρίζεται σε υποσύνολα $P_1, \dots, P_{\lceil n/m \rceil}$, μεγέθους το πολύ m το κάθε ένα
2. για $i = 1, \dots, \lceil n/m \rceil$
υπολογίζεται το κυρτό περίγραμμα $\mathcal{P}_i = \text{conv}(P_i)$ με τον αλγόριθμο των Preparata και Hong και αποθηκεύεται σε μια ιεραρχία Dobkin-Kirkpatrick
3. $F, Q \leftarrow \{f_0\}$, όπου f_0 μια αρχική πλευρά του $\mathcal{P} = \text{conv}(P)$
4. για $k = 1, \dots, 2H - 4$
αν $Q = \emptyset$ επιστρέφεται F
επιλέγεται κάποια πλευρά $f \in Q$ και θέτουμε $Q \leftarrow Q - \{f\}$
έστω e_j οι ακμές της f ($j = 1, 2, 3$)²
για $j = 1, 2, 3$
για $i = 1, \dots, \lceil n/m \rceil$
υπολογίζεται το σημείο $\mathbf{q}_i \in P_i$ το οποίο μεγιστοποιεί τη γωνία μεταξύ των f και $\text{conv}(e_j \cup \{\mathbf{q}_i\})$, ψάχνοντας στην ιεραρχία των $\mathcal{P}_i = \text{conv}(P_i)$
 $\mathbf{p}_j \leftarrow$ το σημείο q από το $\{\mathbf{q}_1, \dots, \mathbf{q}_{\lceil n/m \rceil}\}$ που μεγιστοποιεί την γωνία μεταξύ των f και $\text{conv}(e_j \cup \{\mathbf{q}\})$ ($\mathbf{q} \notin e_j$)

$$f_j \leftarrow \text{conv}(e_j \cup \{\mathbf{p}_j\})$$

$$\text{αν } f_j \notin F \text{ τότε } F \leftarrow F \cup \{f_j\}, Q \leftarrow Q \cup \{f_j\}$$

5. επιστρέφεται *incomplete*

Για την υλοποίηση του Q μπορούμε να χρησιμοποιήσουμε ουρά ή στοίβα και για το σύνολο F μια δομή λεξικού. Αφού οι λειτουργίες του λεξικού απαιτούν χρόνο $O(h)$ η καθεμιά, απαιτείται συνολικά $O(h \log h)$ χρόνος. Στην πραγματικότητα, εξυπνότερες υλοποιήσεις της μεθόδου τυλίγματος μέσω μιας *shelling order* [Sei86], αντικαθιστούν την απαίτηση λεξικού, χρησιμοποιώντας απλώς μια ουρά προτεραιότητας.

Όπως και στις δύο διαστάσεις, επιλέγουμε μέγεθος ομαδοποίησης $m = H$ και προσεγγίζουμε την τιμή του h με μια ακολουθία τιμών του H :

Αλγόριθμος Group-and-Wrap στις Τρεις Διαστάσεις

1995

ΕΙΣΟΔΟΣ: σύνολο $P \subseteq E^3$ με n σημεία

ΕΞΟΔΟΣ: λίστα σημείων που αποτελούν το κυρτό περίγραμμα του P , με φορά αντίθετη των δεικτών του ρολογιού

1. για $t = 1, 2, \dots$

$$L \leftarrow \text{GroupHull3D}(P, m, H), \text{ όπου } m = H = \min\{2^{2^t}, n\}$$

αν $L \neq \text{incomplete}$ επιστρέφεται L

Αποδεικνύεται από τον Chan [Cha95] ότι ο παραπάνω αλγόριθμος υπολογίζει το κυρτό περίγραμμα h -κορυφών, συνόλου $P \subseteq E^3$ με n σημεία, σε χρόνο $O(n \log h)$, χρησιμοποιώντας $O(n)$ χώρο.

4.4 Συμπεράσματα για τις Τρεις Διαστάσεις

Κλείνοντας την αναφορά μας στους αλγορίθμους υπολογισμού κυρτού περιγράμματος στις τρεις διαστάσεις, συνοψίζουμε τις αντίστοιχες πολυπλοκότητες που παρουσιάζουν στον πίνακα που ακολουθεί.

Οι παραπάνω μέθοδοι αποτελούν τους περισσότερο γνωστούς αλγορίθμους για τις τρεις διαστάσεις. Αυτός είναι και ο λόγος που παρουσιάστηκαν. Τον βέλτιστο αλγόριθμο αποτελεί ο τελευταίος από τον Chan.

Ιδεές για περεταίρω βελτιστοποιήσεις, μπορεί να βρει ο ενδιαφερόμενος αναγνώστης στη σχετική διατριβή [Cha95].

²Όπως φαίνεται και στο τέλος του βήματος, δεν λαμβάνεται πρόνοια για ομοεπίπεδες πλευρές, όπως στην προηγούμενη μέθοδο. Όλες οι πλευρές θεωρούνται τριγωνικές (ή ότι αποτελούνται από τριγωνικά τμήματα, τα οποία αντιμετωπίζονται σαν να ήταν μη συνεπίπεδες πλευρές).

Αλγόριθμος	Χρόνος Εκτέλεσης	Χρονολογία	Αναφορές
Preparata και Hong	$O(n \log n)$	1977	[PH77]
Γενίκευση του Jarvis's March	$O(nh)$	1970, 1985	[CK70, Swa85]
Edelsbrunner και Shi	$O(n \log^2 h)$	1991	[ES91]
Chazelle και Matoušek	$O(n \log h)$	1995	[CM95]
QuickHull	$O(n \log r)$	1976-1996	[BDH96]
Group-and-Wrap	$O(n \log h)$	1995	[Cha95]

Πίνακας 4.1: Κυριότεροι ντετερμινιστικοί αλγόριθμοι για τις τρεις διαστάσεις ($d = 3$).

Κεφάλαιο 5

Κυρτά Περιγράμματα Μεγαλυτέρων Διαστάσεων

Για τα κυρτά περιγράμματα μεγαλυτέρων διαστάσεων, η έρευνα κατευθύνθηκε προς την εύρεση αποδοτικών worst-case αλγορίθμων. Η πρώτη μέθοδος που παρουσιάστηκε ήταν η μέθοδος τυλίγματος των Chand και Karur το 1970 [CK70]: η μέθοδος αυτή απαιτεί χρόνο $O(n^{\lfloor d/2 \rfloor + 1})$, στη χειρότερη περίπτωση, όπως αποδείχτηκε από τον Swart, δεκαπέντε χρόνια αργότερα [Swa85]. Απέδειξε επίσης, ότι η πολυπλοκότητα ως προς την έξοδο της μεθόδου είναι $O(nf)$, όπου f ο αριθμός των πλευρών του κυρτού περιγράμματος που προκύπτει.³

Το 1981 ο Seidel βελτίωσε το όριο αυτό σε $O(n^{\lfloor d/2 \rfloor})$, ακολουθώντας μια διαφορετική προσέγγιση, την οποία ονόμασε μέθοδο *beneath-beyond* [Sei81]. Ο ίδιος, μερικά χρόνια αργότερα [Sei86], παρουσίασε μια ακολουθία που ονομάστηκε *shelling order*, μέσω της οποίας πέτυχε ένα δεύτερο ντετερμινιστικό αλγόριθμο με πολυπλοκότητα $O(n^{\lfloor d/2 \rfloor} \log n)$, το πολύ. Απέδειξε επίσης, ότι η πολυπλοκότητα ως προς την έξοδο αυτού του αλγορίθμου είναι $O(n^2 + f \log n)$.

Το 1989 οι Clarkson και Shor, παρουσίασαν έναν πιθανοτικό incremental αλγόριθμο [CS89], με πολυπλοκότητα $O(n^{\lfloor d/2 \rfloor})$, η οποία αποτελεί τη βέλτιστη. Ένας δεύτερος πιθανοτικός αλγόριθμος, με την ίδια πολυπλοκότητα, παρουσιάστηκε το 1991 από τον Seidel [Sei91].

Τελικά, ο πρώτος ντετερμινιστικός αλγόριθμος, με πολυπλοκότητα τη βέλτιστη $O(n^{\lfloor d/2 \rfloor})$, παρουσιάστηκε από τον Chazelle [Cha93]: ο αλγόριθμος αυτός βασίζεται στην πιθανοτική λύση των Clarkson και Shor, συνδυάζοντας κάποιες νέες ιδέες για derandomization. Σημειώνουμε

³Ο αριθμός των πλευρών f , πολυτόπου διάστασης d και αριθμού κορυφών n , αποδεικνύεται ότι είναι $\Omega(1)$ και $\Theta(n^{\lfloor d/2 \rfloor})$ –για την απόδειξη βλ. [Ede87].

στο σημείο αυτό, το μεγάλο προβάδισμα των αλγορίθμων με εξάρτηση εξόδου, έναντι αυτής της μεθόδου, στην περίπτωση όπου $f \ll n$.

Όσον αφορά την εξάρτηση από τα n και f , το μοναδικό κάτω όριο είναι το $\Omega(n \log f + f)$, αν και μερικοί ερευνητές υποψιάζονται την ύπαρξη αλγορίθμου με πολυπλοκότητα $O(n \log f + f)$. η εύρεση ενός τέτοιου αλγορίθμου, παραμένει προκλητικό πρόβλημα, ακόμη και για τη διάσταση $d = 4$.

Στο παραπάνω όριο του Seidel υπήρξαν δύο βελτιώσεις. Ο Matoušek απέδειξε ότι η μέθοδος του Seidel είναι δυνατόν να επιτύχει πολυπλοκότητα $O(n^{2-2/(\lfloor d/2 \rfloor + 1)+\varepsilon} \log f + f)$, χρησιμοποιώντας κάποιες ειδικές δομές δεδομένων [Mat93].⁴ Εξαλείφοντας το ε , ο Chan έδειξε [Cha95] ότι η πολυπλοκότητα καταλήγει $O(n^{2-2/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n + f \log n)$.

Η δεύτερη βελτίωση αποτελεί και την τελευταία πρόοδο στην εξέλιξη των αλγορίθμων κυρτού περιγράμματος και δόθηκε από τον Chan [Cha95]. Συγκεκριμένα, για $d = 4$ παρουσίασε έναν αλγόριθμο με πολυπλοκότητα, $O((n + f) \log^2 f)$. Για $d > 4$ παρουσίασε μια εξέλιξη της μεθόδου τυλίγματος, η οποία παρουσιάζει πολυπλοκότητα $O(n \log f + (nf)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n)$.

Δεδομένου ότι οι παραπάνω αλγόριθμοι αποτελούν εξειδικεύσεις και παραλλαγές αλγορίθμων οι οποίοι ήδη έχουν αναλυθεί στην παρούσα εργασία, θα αναφερθούμε επιγραμματικά μόνο σ' αυτούς. Το ενδιαφέρον μας επικεντρώνεται στον τρόπο λειτουργίας τους και στις αρχές που βασίζονται, πέραν της πολυπλοκότητας την οποία επιτυγχάνουν.

5.1 Δύο Κάτω Όρια

Όπως είδαμε παραπάνω, ένα πολύτοπο διάστασης d με n κορυφές, μπορεί να έχει έως $\Theta(n^{\lfloor d/2 \rfloor})$ πλευρές [Ede87]. Στην πραγματικότητα όμως, τα κυρτά περιγράμματα πρακτικών προβλημάτων, ή συνόλων σημείων τα οποία παράγονται τυχαία, έχουν αρκετά λιγότερες πλευρές από το παραπάνω όριο.

Ως αποτέλεσμα αυτού, υπήρξε μια στροφή των μελετητών προς αλγορίθμους με εξάρτηση πολυπλοκότητας από την έξοδο. Σημειώνουμε δύο κατηγορίες αυτών των αλγορίθμων:

- αυτούς οι οποίοι *απαριθμούν* απλώς, τις πλευρές του κυρτού περιγράμματος
- αυτούς οι οποίοι παράγουν το πλήρες *πλευρικό πλέγμα*, το οποίο δεν είναι άλλο από την περιγραφή όλων των πλευρών αλλά και των σχέσεων γειτνίασης τους στο κυρτό πολύτοπο.

⁴Ο αριθμός $\varepsilon > 0$ δηλώνει έναν πολύ μικρό, αλλά σταθερό αριθμό.

Ανάλογα με το μέγεθος της εξόδου f , καθορίζεται και η αναφερόμενη πολυπλοκότητα. Ο αλγόριθμος Jarvis's march, π.χ. έχει πολυπλοκότητα $O(nf)$. Για $f \ll n$, ο αλγόριθμος εκτελείται σε γραμμικό χρόνο, ενώ στη χειρότερη περίπτωση η πολυπλοκότητα του είναι $O(n^2)$ (όλα τα σημεία της εισόδου ανήκουν στο κυρτό περίγραμμα).

Έστω f ο αριθμός των πλευρών (στο επίπεδο, ο αριθμός των πλευρών ή των ακμών, είναι ίδιος με τον αριθμό των κορυφών n). Αν το f είναι πολύ μεγάλο (εκθετικό ως προς τον αριθμό των κόμβων n), ένα προφανές κάτω όριο του χρόνου εκτέλεσης είναι το $\Omega(f)$. Αν το f είναι μικρό, το κάτω όριο πέφτει σε $\Omega(n \log f)$.

Λαμβάνοντας, λοιπόν, υπ' όψιν μας το μέγεθος της εξόδου, το κάτω όριο για το πρόβλημα του κυρτού περιγράμματος είναι $\Omega(n \log f + f)$.

Σημειώνουμε επίσης, ένα κάτω όριο για τον καθορισμό των πλευρών του κυρτού περιγράμματος n σημείων, διάστασης d . Ο Erickson απέδειξε ότι απαιτείται χρόνος $\Omega(n^{\lceil d/2 \rceil - 1} + n \log n)$ το πολύ, για την απαρίθμηση τους ([Eri96]).

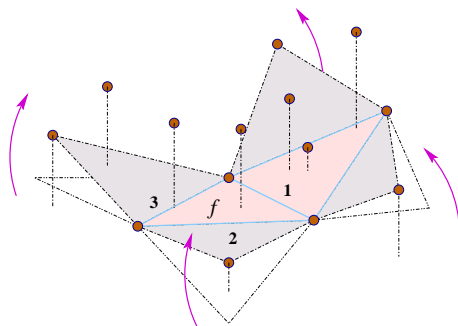
5.2 Περιγραφή Αλγορίθμων

Υπάρχουν δύο γενικές μέθοδοι οι οποίες λύνουν το πρόβλημα του κυρτού περιγράμματος σε μεγάλες διαστάσεις: ο αλγόριθμος τυλίγματος (gift-wrapping) και η μέθοδος *beneath-beyond*.

Σύμφωνα με τη μέθοδο τυλίγματος, όπως είδαμε και στις μικρότερες διαστάσεις, προχωράμε από μία πλευρά στις γειτονικές της, όπως κατά το τυλίγμα ενός αντικειμένου πεπερασμένων διαστάσεων από ένα φύλλο χαρτιού (βλ. σχήμα 17). Αυτό γίνεται αποθηκεύοντας για κάθε πλευρά τις ακμές που τη σχηματίζουν· στη συνέχεια, προστίθεται στο κυρτό περίγραμμα η πλευρά που σχηματίζεται από την κάθε ακμή έτσι ώστε να σχηματίζεται η μεγαλύτερη γωνία, όπως στο σχήμα. Ειδική περίπτωση της μεθόδου αποτελεί ο αλγόριθμος Jarvis's march.

Η πολυπλοκότητα για την απαρίθμηση των πλευρών είναι $O(nf)$, ενώ για την εύρεση του όλου πλέγματος, μεγέθους L , είναι $O(\min(nL^2, n^{\lceil d/2 \rceil + 1}))$. Η μέθοδος αυτή παρουσιάστηκε από τους Chand και Kapur το 1970.

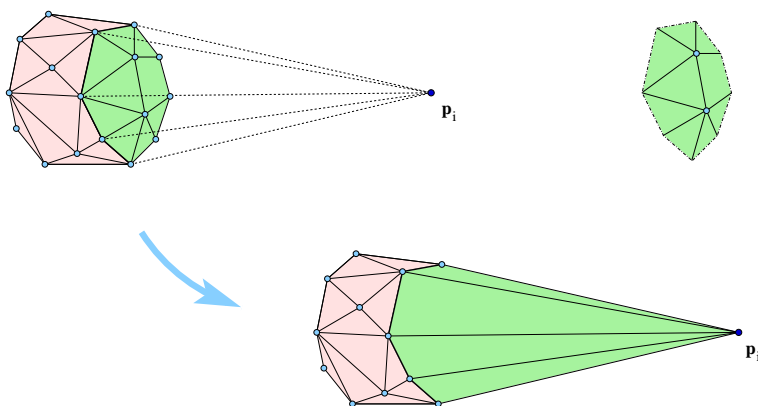
Η μέθοδος *beneath-beyond* αποτελεί μια αυξητική (incremental) προσέγγιση. Το κυρτό περίγραμμα κατασκευάζεται προσθέτοντας ένα σημείο σε κάθε βήμα του αλγορίθμου. Αν $\mathcal{CH}(P_{i-1}) = \text{conv}(P_{i-1})$ το τρέχον κυρτό περίγραμμα (για τα $i - 1$ σημεία) και θέλουμε να προσθέσουμε ένα σημείο \mathbf{p}_r , εξωτερικό του περιγράμματος, το τρισδιάστατο ανάλογο είναι να κατασκευάσουμε ένα κώνο από το \mathbf{p}_i στο $\mathcal{CH}(P_{i-1})$ και να αφαιρέσουμε από το περίγραμμα τα σημεία που βρίσκονται στη σκιά του κώνου αυτού (βλ. σχήμα 18). Προφανώς αφαιρούνται και



Σχήμα 17: Η μέθοδος τυλίγματος για μεγάλες διαστάσεις. Δεδομένης της πλευράς f , οι γειτονικές πλευρές της βρίσκονται μία-μία, ακολουθώντας τις ακμές της f . Τα διακεκομμένα κατακόρυφα τμήματα έχουν τοποθετηθεί για τη διευκόλυνση της τρισδιάστατης απεικόνισης.

οι πλευρές που σχηματίζονταν από τα σημεία αυτά.

Η μέθοδος αυτή παρουσιάστηκε το 1981 από τον Seidel [Sei81]. Η πολυπλοκότητα της είναι $O(n^{\lceil d/2 \rceil})$ και είναι βέλτιστη για περιττές διαστάσεις, όπως αποδεικνύεται στο [Ede87]. Ο ενδιαφερόμενος αναγνώστης μπορεί να βρει περισσότερες πληροφορίες, όπως και έναν ακριβή αλγόριθμο υλοποίησης που βασίζεται σε μια παραλλαγή της μεθόδου στο βιβλίο [dBvKOS97].



Σχήμα 18: Δημιουργία του $CH(P_i)$ από το $CH(P_{i-1})$, προσθέτοντας το σημείο p_i . Προστίθεται ο κώνος που σχηματίζει το σημείο με το περίγραμμα· αφαιρούνται από αυτό οι κορυφές και οι πλευρές που βρίσκονται στη σκιά του κώνου (επάνω δεξιά).

Οι υπόλοιποι αλγόριθμοι έχουν σαν βάση τις δύο αυτές μεθόδους. Χρησιμοποιούν επιπλέον κάποιες ειδικές δομές δεδομένων, οι οποίες ευνοούν τις αναζητήσεις γειτονικών πλευρών και κορυφών και καταφέρνουν έτσι να μειώσουν την πολυπλοκότητα. Περεταίρω λεπτομέρειες ξε-

φεύγουν από τα όρια της παρούσας εργασίας.

5.3 Συμπεράσματα

Συνοψίζουμε τις πολυπλοκότητες για τους αλγόριθμους κυρτού περιγράμματος μεγαλύτερων διαστάσεων στον πίνακα που ακολουθεί. Πέραν της πολυπλοκότητας που εμφανίζουν, η φύση του εκάστοτε προβλήματος καθορίζει την καταλληλότητα τους σε κάθε περίπτωση.

Αλγόριθμος	Χρόνος Εκτέλεσης	Αναφορές
Μέθοδος τυλίγματος	$O(n^{\lfloor d/2 \rfloor + 1}), O(nf)$	[CK70, Swa85]
Μέθοδος beneath-beyond	$O(n^{\lfloor d/2 \rfloor})$	[Sei81]
Αλγόριθμος “shelling”	$O(n^{\lfloor d/2 \rfloor} \log n), O(n^2 + f \log n)$	[Sei86]
Chazelle	$O(n^{\lfloor d/2 \rfloor})$	[Cha93]
Matoušek	$O(n^{2-2/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n + f \log n)$	[Mat93]
Chan, Snoeyink και Yap ($d = 4$)	$O((n + f) \log^2 f)$	[CSY95]
Chan ($d > 4$)	$O(n \log f + (nf)^{1-1/(\lfloor d/2 \rfloor + 1)} \log^{O(1)} n)$	[Cha95]
Amato και Ramos ($d = 5$)	$O((n + f) \log^3 f)$	[AR96]

Πίνακας 5.1: Ντετερμινιστικοί αλγόριθμοι κυρτών περιγραμμάτων μεγάλων διαστάσεων ($d \geq 4$).

Κεφάλαιο 6

Ξεφεύγοντας από τη Θεωρία

Στο κεφάλαιο αυτό, πραγματοποιείται μια στοιχειώδης αναφορά σε δύο δομές δεδομένων, οι οποίες χρησιμοποιούνται για την επεξεργασία πολυτόπων, καθώς και σε μια εφαρμογή που υλοποιεί τα όσα αναφέραμε ως εδώ. Ο λόγος που γίνεται αυτό είναι για να αποκτήσει η παρούσα εργασία μια πιο πρακτική χρεία.

6.1 Πολυεδρική Ιεραρχία (Polyedral Hierarchy)

Η δομή αυτή παρουσιάστηκε από τους Dobkin και Kirkpatrick [DK83, DK90] και αναφερόταν σε πολυέδρα τριών διαστάσεων. Υιοθετήθηκε όμως και από άλλους ερευνητές, όπως ο Chan [Cha95] και ο Eppstein [Epp91], ως εργαλείο για την επίλυση διαφόρων προβλημάτων.

Ως *πολυεδρική ιεραρχία* ορίζεται μια μονότονη ακολουθία τρισδιάστατων πολυτόπων $\mathcal{P}_1 \subset \mathcal{P}_2 \subset \dots \subset \mathcal{P}_\ell$, με την ιδιότητα ότι κάθε συνδεδεμένο τμήμα (connected component) των $\mathcal{P}_{k+1} - \mathcal{P}_k$ είναι σταθερής πολυπλοκότητας. Κάθε \mathcal{P}_k ονομάζεται *επίπεδο* της ιεραρχίας. Αν το \mathcal{P}_1 είναι σταθερού μεγέθους και $\mathcal{P}_\ell = \mathcal{P}$, τότε η ακολουθία καλείται *εσωτερική ιεραρχική αναπαράσταση* του \mathcal{P} . Αντίστοιχα, αν $\mathcal{P}_1 = \mathcal{P}$ και το \mathcal{P}_ℓ είναι σταθερού μήκους, τότε η καλείται *εξωτερική ιεραρχική αναπαράσταση* του \mathcal{P} .

Οι Dobkin και Kirkpatrick έδειξαν ότι, για οποιοδήποτε τριδιάστατο πολύτοπο με n κορυφές, υπάρχει μια εσωτερική/εξωτερική ιεραρχική αναπαράσταση $O(\log n)$ επιπέδων και μπορεί να υπολογιστεί σε χρόνο $O(m)$.

Το πολύτοπο κάθε επιπέδου δεν αποθηκεύεται αποκλειστικά στην αναπαράσταση: παρέχονται δείκτες μεταξύ διαδοχικών επιπέδων, οι οποίοι επιτρέπουν τη διάσχιση της ιεραρχίας, προς τα πάνω ή προς τα κάτω.

Η ιεραρχική αναπαράσταση αποτελεί μια πολύ χρήσιμη δομή δεδομένων, για την επεξεργασία πολυτόπων, στις τρεις διαστάσεις. Για παράδειγμα, διασχίζοντας την εσωτερική ιεραρχία προς τα πάνω (*ανεβαίνοντας*), μπορούμε να βρούμε την εφαπτομένη δεδομένου πολυτόπου με δεδομένη γραμμή που το διαπερνά, σε χρόνο $O(\log n)$. Διασχίζοντας την εξωτερική ιεραρχία προς τα κάτω (*κατεβαίνοντας*), μπορούμε να λύσουμε το δυαδικό πρόβλημα ακτίνων που τέμνουν πολύτοπο, σε χρόνο $O(\log n)$.

Σημειώνουμε μια ακόμη εφαρμογή της δομής αυτής από τον Eppstein [Epp91]. Παρουσίασε ένα πολυλογαριθμικό αλγόριθμο για τον έλεγχο της τομής τριών πολυτόπων στον E^3 .

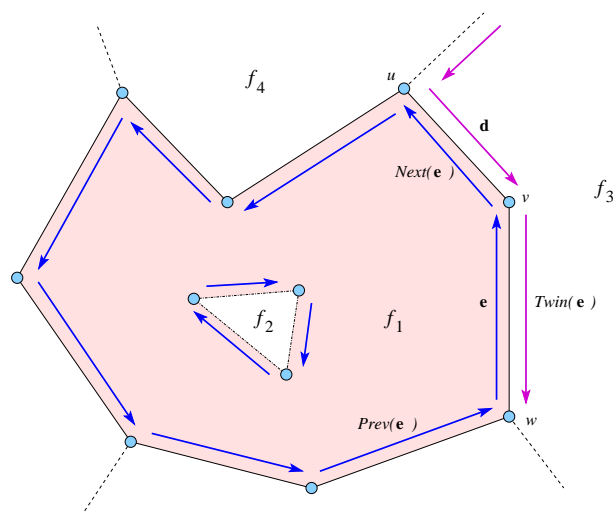
6.2 Διπλά Συνδεδεμένη Λίστα Ακμών (doubly-connected edge list)

Η διπλά συνδεδεμένη λίστα ακμών αποτελεί μια δομή δεδομένων, η οποία επίσης χρησιμοποιείται για τον χειρισμό πολυτόπων. Τόσο η χρησιμότητα όσο και η αποδοτικότητα της έγκειται στο ότι αποθηκεύει συσχετίσεις μεταξύ των στοιχείων των πολυτόπων.

Συγκεκριμένα, αποτελείται από τρεις λίστες εγγραφών (records): μία για τις κορυφές (*vertices*), μία για τις πλευρές (*faces*) και μία για τις ημι-ακμές (*half-edges*). Για δική μας ευκολία αντιμετωπίζουμε τις ακμές ως αποτελούμενες από δύο (ξεχωριστές) προσανατολισμένες δίδυμες ημιακμές· μπορούμε έτσι να διατρέξουμε την περιφέρεια μιας πλευράς διαμέσου των ημιακμών, με φορά αντίθετη των δεικτών του ρολογιού (σχήμα 19).

Οι τρεις αυτές δομές περιέχουν τις τοπολογικές και γεωμετρικές πληροφορίες του σχηματισμού μας:

- Στις *εγγραφές κορυφών* αποθηκεύονται για κάθε κορυφή v οι συντεταγμένες του, σε ένα πεδίο που ονομάζεται $Coordinates(v)$, καθώς επίσης και ένας δείκτης $Incident(v)$ σε μια (οποιαδήποτε) πλευρά, η οποία ξεκινάει από το v .
- Οι *εγγραφές πλευρών* αποθηκεύουν για κάθε πλευρά f , έναν δείκτη $OuterComponent(f)$ σε μια από τις ημιακμές που αποτελούν το περίγραμμα τους. Για τις μη-κλειστές πλευρές ο δείκτης αυτός είναι $null$. Αποθηκεύουν επίσης, και μία λίστα $InnerComponents(f)$, η οποία περιέχει, για κάθε “τρύπα” της πλευράς, έναν δείκτη σε μια από τις ημιακμές του περιγράμματος της.
- Οι *εγγραφές ημιακμών* αποθηκεύουν, για κάθε ακμή e , πέντε δείκτες οι οποίοι αναφέρονται, αντίστοιχα: στην κορυφή εκκίνησης της ακμής $Origin(e)$, στην δίδυμη ημιακμή της



Σχήμα 19: Διάσχυση πλευράς πολυτόπου κατά μήκος των ημιακμών της περιφέρειας του. Η πλευρά βρίσκεται πάντα στα αριστερά των ημιπλευρών της περιφέρειας της.

$Twin(e)$, στην πλευρά την οποία περιβάλλει $IncidentFace(e)$, στην προηγούμενη και την επόμενη ημιακμή $Prev(e)$ και $Next(e)$ (βλ. σχήμα 19).

Οι παραπάνω λίστες εγγραφών επιτρέπουν άμεση εύρεση στοιχείων, όπως τη διάσχυση της περιμέτρου μιας πλευράς, ή την εύρεση όλων των γειτονικών πλευρών μιας πλευράς.

Τα στοιχεία αυτά, αν δεν χρησιμοποιηθούν ειδικές δομές δεδομένων σαν αυτές, αποτελούν ξεχωριστά προβλήματα στους αλγόριθμους που αναφέρονται σε πολύτοπα. Αυτός είναι και ο λόγος που καθιερώθηκαν αυτές οι δομές. Σημειώνουμε ότι αναλυτική περιγραφή της παραπάνω δομής, γίνεται στο βιβλίο [dBvKOS97].

6.3 Το πρόγραμμα qhull

Για τον ενδιαφερόμενο αναγνώστη που δεν θέλει να μείνει μόνο στο θεωρητικό μέρος των κυρτών περιγραμμάτων και των πολυτόπων, σημειώνουμε το πρόγραμμα `qhull`. Δεν είναι παρά μια υλοποίηση του αλγόριθμου `QuickHull`, η οποία συνάμα δημιουργεί και όλες τις δομές δεδομένων που συνήθως απαιτούνται (λίστα κορυφών, πλευρών, λίστα γειτονικών πλευρών για κάθε πλευρά, κ.α.).

Το εν λόγω πρόγραμμα αναπτύχθηκε στο πανεπιστήμιο της Minnesota και επιτρέπει τον υπολογισμό κυρτών περιγραμμάτων, διαγραμμάτων Voronoi και τριγωνοποιήσεων Delaunay. Ακούει

στα ονόματα `qhull` και `qconvex` και είναι προσπελάσιμο μέσω των ακόλουθων διευθύνσεων:

```
<http://www.geom.umn.edu/locate/qhull>  
<ftp://geom.umn.edu/pub/software/qhull13-0.zip>  
<http://www.geom.umn.edu/~bradb/qhull-news.html>
```

Αξίζει να πειραματιστείτε! Το πρόγραμμα είναι ιδιαίτερα εύχρηστο λόγω των χαρακτηριστικών του: δέχεται είσοδο από αρχείο, δουλεύει για οποιαδήποτε διάσταση και η έξοδος του καθορίζεται από τον χρήστη (αν θέλει μόνο σημεία, ή μόνο πλευρές ή/και ακμές, κ.α.). Η λεπτομέρεια της καθορίζεται επίσης, από τον χρήστη (λεπτομέρειες γειτνίασης πλευρών-κορυφών, πλευρών-ακμών, κ.α.).

Βιβλιογραφία

- [AR96] N. M. Amato and E. A. Ramos. On computing voronoi diagrams by divide-prune-and-conquer. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pages 166–175, 1996.
- [Bay99] V. Bayer. Survey of algorithms for the convex hull problem, March 1999. Department of Computer Science, Oregon State University.
- [BDH96] C. B. Barder, D. P. Dobkin, and H. Hundanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, December 1996.
- [Cha93] B. Chazelle. An optimal convex hull algorithm for point sets in any fixed dimension. *Discrete & Computational Geometry*, 10:377–409, 1993.
- [Cha95] T. M. Chan. *Output-Sensitive Construction of Convex Hulls*. PhD thesis, The University of British Columbia, October 1995.
- [CK70] D. R. Chand and S. S. Kapur. An algorithm for convex polytopes. *Journal of the Association of Computing Machinery*, 17:78–86, 1970.
- [Cla94] K. L. Clarkson. More output-sensitive geometric algorithms. In *Proceedings of the 35th Annual IEEE Symposium on Foundations of Computer Science*, pages 695–702, 1994.
- [CM95] B. Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. *Computational Geometry: Theory and Applications*, 5:27–32, 1995.

- [CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, ii. *Discrete & Computational Geometry*, 4:387–421, 1989.
- [CSY95] T. M. Chan, J. Snoeyink, and C.-K. Yap. Output-sensitive construction of polytopes in four dimensions and voronoi diagrams in three. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 282–291, 1995.
- [dBvKOS97] M. de Berg, M. van Kreveld, M. Overmans, and O. Schwarzkopf. *Computational Geometry: Algorithms and Applications*. Springer, 1997.
- [DK83] D.P. Dobkin and D. G. Kirkpatrick. Fast detection of polyedral intersection. *Theoretical Computer Science*, 27:241–253, 1983.
- [DK90] D.P. Dobkin and D. G. Kirkpatrick. Determining the separation of preprocessed polyedra: a unified approach. In *Proceedings of the 17th International Colloquium on Automata, Languages and Programming*, volume 443, pages 440–443, Lecture Notes in Computer Science, 1990. Springer-Verlag.
- [EC92] I. Emiris and J. Canny. An efficient approach to removing geometric degeneracies. In *Proceedings of the Eighth Annual ACM Symposium on Computational Geometry*, pages 74–82, 1992.
- [Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry*. Springer-Verlag, 1987.
- [EM90] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics*, 9(1):66–104, 1990.
- [Epp91] D. Eppstein. Dynamic three-dimensional linear programming. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pages 488–494, 1991.
- [Eri96] J. Erickson. New lower bounds for convex hull problems in odd dimensions. In *Proceedings of the 12th Annual ACM Symposium on Computational Geometry*, pages 1–9, 1996.

- [ES91] H. Edelsbrunner and W. Shi. An $O(n\log^2h)$ time algorithm for the three-dimensional convex hull problem. *SIAM Journal on Computing*, 20:259–277, 1991.
- [Gra67] R. L. Graham. An efficient algorithm for determining the convex hull of a finite planar set. *Information Processing Letters*, 1:132–133, 1967.
- [Grü67] B. Grünbaum. *Convex Polytopes*. John Wiley & Sons, London, 1967.
- [Jar73] R. A. Jarvis. On the identification of the convex hull of a finite set of points in the plane. *Information Processing Letters*, 2:18–21, 1973.
- [KS86] D. Kirkpatrick and R. Seidel. The ultimate planar convex hull algorithm? *SIAM Journal on Computing*, 15:287–299, 1986.
- [Mat93] J. Matoušek. Linear optimization queries. *Journal of Algorithms*, 14:432–448, 1993.
- [MS71] P. McMullen and G. C. Shephard. *Convex Polytopes and the Upper Bound Conjecture*. Cambridge University Press, 1971.
- [Mul93] M. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice-Hall, Englewood Cliffs, N.J., 1993.
- [O'R94] J. O'Rourke. *Computational Geometry in C*. Cambridge University Press, 1994.
- [PH77] F. P. Preparata and S. J. Hong. Convex hulls of finite sets of points in two and three dimensions. *Communications of the Association of Computing Machinery*, 20:87–93, 1977.
- [PS85] F. P. Preparata and M. I. Shamos. *Computational Geometry: An Introduction*. Springer-Verlag, New York, 1985.
- [Sei81] R. Seidel. A convex hull algorithm optimal for point sets in even dimensions. Technical Report 81-14, Department of Computer Science, University of British Columbia, Vancouver, B.C., 1981.
- [Sei86] R. Seidel. Constructing higher-dimensional convex hulls at logarithmic cost per face. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 404–413, 1986.

- [Sei91] R. Seidel. Small-dimensional linear programming and convex hull made easy. *Discrete & Computational Geometry*, 6:423–434, 1991.
- [Swa85] G. F. Swart. Finding the convex hull facet by facet. *Journal of Algorithms*, 6:17–48, 1985.