ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΛΟΠΟΝΝΗΣΟΥ

Σχολή Θετικών Επιστημών και Τεχνολογίας

Τμήμα Επιστήμης και Τεχνολογίας Υπολογιστών

Μεταπτυχιακή Διπλωματική Εργασία

Προσεγγιστική Καταμέτρηση Περιοχής

Κωνσταντίνα Σ. Τσίρμπα

Α.Μ.: 2009032

Επιβλέπων Καθηγητής: Νικόλαος Πλατής

Τρίπολη, Απρίλιος 2013

Ευχαριστίες

ΣΤΟΝ ΚΑΘΗΓΗΤΗ ΜΟΥ ΓΙΑ ΤΗΝ ΥΠΟΣΤΗΡΙΞΗ ΠΟΥ ΜΟΥ ΠΡΟΣΕΦΕΡΕ ΚΑΙ ΣΤΗΝ ΟΙΚΟΓΕΝΕΙΑ ΜΟΥ

ΠΕΡΙΕΧΟΜΕΝΑ

ΠΕΡΙΛΗΨΗ

Η παρούσα εργασία πραγματεύεται την προσεγγιστική καταμέτρηση και αναζήτηση περιοχής (approximate range counting - searching) η οποία συγκαταλέγεται στα κυριότερα προβλήματα της υπολογιστικής γεωμετρίας.

Στο **1ο** κεφάλαιο γίνεται μια εισαγωγή στις έννοιες της υπολογιστικής γεωμετρίας για την καλύτερη κατανόηση του επιστημονικού χώρου. Δίδονται παραδείγματα εφαρμογής της στο σύγχρονο κόσμο και εξηγούνται οι βασικές έννοιες των δομών για ερωτήσεις περιοχής (range searching).

Στο **2ο** κεφάλαιο παρουσιάζεται το πρόβλημα αναζήτησης περιοχής. Για την καλύτερη κατανόηση του προβλήματος παρουσιάζουμε συγκεκριμένες δεντρικές δομές δεδομένων, Range Trees και k-d Trees.

Στο **3ο** κεφάλαιο αναλύεται το Balanced Box-Decomposition δέντρο (BBD-Tree) και η σημασία του για την επίλυση του παραπάνω προβλήματος. Επιπλέον, αναλύεται η δυναμική δομή δεδομένων «quadtreap», η οποία χρησιμοποιείται για την αποθήκευση πολυδιάστατων συνόλων από σημεία. Παρατίθεται, τέλος, και ο αλγόριθμος για την απάντηση ερωτημάτων προσεγγιστικής καταμέτρησης περιοχής χρησιμοποιώντας τη δομή quadtreap.

Στο **4ο** κεφάλαιο ερευνούμε τα ε-NN ερωτήματα πλησιέστερου γείτονα (nearest neighbor queries) στο πεδίο των παράλληλων τμημάτων. Παρουσιάζουμε μια δομή δεδομένων που απαντά σε χρονικά-εξαρτώμενα ε-NN ερωτήματα σε οποιαδήποτε σταθερή διάσταση. Επιπλέον, παρουσιάζουμε μια μέθοδο που μειώνει τόσο τον χώρο όσο και τον χρόνο στα ερωτήματα πλησιέστερου γείτονα. Τέλος, ερευνούμε τα ερωτήματα αναζήτησης ε-NN με περιορισμένο βάρος και παρουσιάζουμε ένα βασικό θεώρημα.

ABSTRACT

The present thesis deals with approximate range counting and searching, which constitutes one of the main problems of computational geometry.

The first chapter introduces the concepts of computational geometry aiming at a better understanding of the relevant scientific sector. It provides examples of its application in the modern world, as well as further insights into the basic concepts of the structures of range searching queries.

The second chapter goes on to present the range searching problem. We present specific tree data structures (Range Trees and k-d Trees) to better understand the problem.

The third chapter analyzes the Balanced Box-Decomposition Tree (BBD-Tree) and its contribution to solving the above problem. Moreover, we analyze the dynamic “quadtreap” data structure, which is used to store multi-dimensional sets of points. This chapter ends with the presentation of an algorithm for answering approximate range counting queries using the quadtreap structure.

The fourth chapter investigates the nearest neighbor queries in the field of parallel sections. It presents a data structure that responds to time-dependent e-NN queries in any fixed dimension. A method is also introduced that reduces both the time and space of the nearest neighbor queries. Finally, the chapter explores the e-NN search queries with limited weight and ends up with a basic theorem.

Κεφάλαιο 1

ΥΠΟΛΟΓΙΣΤΙΚΗ ΓΕΩΜΕΤΡΙΑ

ΕΙΣΑΓΩΓΗ

1.1 Συνδυαστική Υπολογιστική Γεωμετρία

1.2 Αριθμητική Υπολογιστική Γεωμετρία

1.3 Ψηφιακή ή Διακριτή Γεωμετρία

1.4Δομές για ερωτήσεις περιοχής (range searching)

ΚΕΦΑΛΑΙΟ 1^{ο}: Υπολογιστική Γεωμετρία

Εισαγωγή

Γεωμετρικά αντικείμενα όπως σημεία, γραμμές και πολύγωνα αποτελούν τη βάση σε ευρύ φάσμα σημαντικών εφαρμογών και δίνουν ώθηση προς επίλυση σε διάφορα προβλήματα κατασκευής αλγορίθμων. Το όνομα γεωμετρία ανακαλεί στη μνήμη μας την αρχέγονη χρησιμότητα της στη μέτρηση των εδαφών και των υλικών. Στις μέρες μας η ραγδαία ανάπτυξη των ηλεκτρονικών υπολογιστών καθιστά δυνατή την επίλυση γεωμετρικών προβλημάτων μεγάλης κλίμακας. Στις περασμένες τέσσερις δεκαετίες ένα σύνολο από εφόδια και τεχνικές αναπτύχθηκαν που εκμεταλλεύονται την θεωρία που μας παρέχει η γεωμετρία. Ο επιστημονικός κλάδος αυτός είναι γνωστός σαν υπολογιστική γεωμετρία.

Ο τομέας αυτός ονομάστηκε και κατά κύριο λόγω ξεκίνησε περίπου το 1978 από τον Shamos, του οποίου η διδακτορική διατριβή προκάλεσε σημαντικό ενδιαφέρον. Μετά από μια δεκαετία εξέλιξης ο τομέας έγινε αυτόνομος το 1985. Το βιβλίο των Preparata και Shamos [19] αποτελεί το πρώτο βιβλίο αποκλειστικά του τομέα της υπολογιστικής γεωμετρίας, το οποίο παρουσιάστηκε στη δημιουργία του πρώτου συνεδρίου στο πεδίο αυτό. Ο τομέας πλέον ακμάζει με μεγάλη ταχύτητα.

Σήμερα με τον όρο Υπολογιστική Γεωμετρία εννοούμε τον κλάδο της επιστήμης των υπολογιστών που ασχολείται με αλγόριθμους και προβλήματα που ορίζονται και περιγράφονται με βάση τα θεωρήματα και τις αρχές της γεωμετρίας (για παράδειγμα προβλήματα της κλασικής γεωμετρίας απεικονισμένα στο διακριτό χώρο που δρουν οι υπολογιστικές τεχνικές). Η βασική ανάγκη ανάπτυξης της υπολογιστικής γεωμετρίας ξεκίνησε από τις εφαρμογές των γραφικών για υπολογιστές, αλλά και των

συστημάτων σχεδίασης με βάση τους υπολογιστές (CAD/CAM). Οι πιο σημαντικές εφαρμογές εντοπίζονται κυρίως στο χώρο της ρομποτικής (προγραμματισμός κίνησης και αναγνώριση εικόνων), στα γεωγραφικά συστήματα πληροφοριών (γεωμετρική αναζήτηση, εύρεση βέλτιστων διαδρομών), στη σχεδίαση ολοκληρωμένων κυκλωμάτων και γενικά στα συστήματα σχεδίασης μέσω υπολογιστών και στον προγραμματισμό αριθμητικών μηχανών.

Από τους κύριους κλάδους της υπολογιστικής γεωμετρίας η συνδυαστική υπολογιστική γεωμετρία, περιλαμβάνει και την αλγοριθμική γεωμετρία η οποία ασχολείται με γεωμετρικά αντικείμενα αντιμετωπίζοντάς τα σαν διακριτά αντικείμενα. Από το άλλο μέρος, η αριθμητική υπολογιστική γεωμετρία, η οποία περιλαμβάνει τη γεωμετρία μηχανών, τη γεωμετρική σχεδίαση μέσω υπολογιστών, τη γεωμετρική μοντελοποίηση, ασχολείται με την γεωμετρική αναπαράσταση και επεξεργασία γεωμετρικών αντικειμένων που προκύπτουν σε πραγματικά προβλήματα.

1.1 Συνδυαστική Υπολογιστική Γεωμετρία

Ο βασικός στόχος της συνδυαστικής υπολογιστικής γεωμετρίας είναι η ανάπτυξη αποτελεσματικών αλγορίθμων και δομών δεδομένων επίλυσης προβλημάτων που αφορούν βασικά γεωμετρικά αντικείμενα όπως τα σημεία, τα ευθύγραμμα τμήματα, τα πολύγωνα, τα πολύεδρα κλπ. Μερικά από τα προβλήματα αυτά μπορούν να θεωρηθούν ως τετριμμένα μακριά από το περιβάλλον των υπολογιστών. Έτσι, η εύρεση δύο πλησιέστερων σημείων από ένα σύνολο σημείων ναι μεν δεν αποτελεί ένα νέο γεωμετρικό πρόβλημα, αλλά αν το πλήθος των σημείων είναι τεράστιο ή η εύρεση του ζεύγους αυτού είναι μέρος ενός προγράμματος, τότε η εξαντλητική αναζήτηση θα έχει υπολογιστική πολυπλοκότητα της τάξης του

τετραγώνου του πλήθους των σημείων, αριθμός που σε κάποιες εφαρμογές μπορεί να είναι απαγορευτικός. Η εξεύρεση, λοιπόν, ενός πιο αποτελεσματικού αλγορίθμου ανήκει στην κατηγορία των προβλημάτων της συνδυαστικής υπολογιστικής γεωμετρίας.

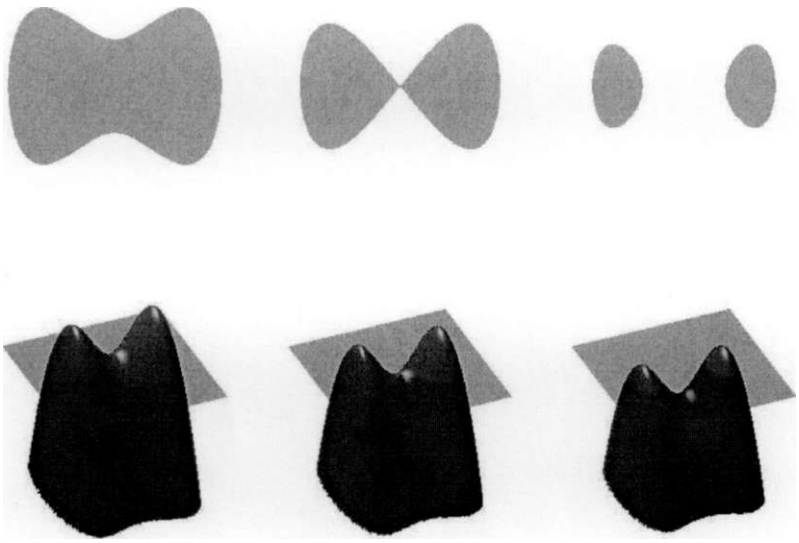
Τα κύρια προβλήματα αυτής της κατηγορίας μπορούν να ταξινομηθούν σε κλάσεις οι οποίες περιλαμβάνουν τα εξής επιμέρους προβλήματα. Τα στατικά προβλήματα, όπως η εύρεση του κυρτού κελύφους (convex hull), η εύρεση της τομής ευθύγραμμων τμημάτων, η τριγωνοποίηση, τα διαγράμματα Voronoi, η εύρεση της ελάχιστης απόσταση σε ένα Ευκλείδειο χώρο με εμπόδια κ.α. Τα προβλήματα γεωμετρικής αναζήτησης στα οποία επιζητείται η βέλτιστη λύση σε ένα γεωμετρικό χώρο όπως η εύρεση σημείων με συγκεκριμένες ιδιότητες, η εύρεση του πλησιέστερου γείτονα, ο προσδιορισμός τροχιάς κ.α. Τα δυναμικά γεωμετρικά προβλήματα στα οποία γίνεται αναζήτηση μιας λύσης σε διαδοχικές χρονικές στιγμές όπως π.χ. η απόδοση κίνησης σ' έναν όγκο και γενικά τα προβλήματα που αφορούν γραφικά υπολογιστών.

1.2 Αριθμητική Υπολογιστική Γεωμετρία

Ο κλάδος της υπολογιστικής γεωμετρίας που αφορά στην αναπαράσταση γεωμετρικών οντοτήτων με απλές γεωμετρικές κατασκευές εύκολα παραμετροποιημένες για χρήση τους και επεξεργασία από τον υπολογιστή αναφέρεται και σαν γεωμετρική μοντελοποίηση. Έτσι για παράδειγμα, η γραφική αναπαράσταση μιας πολύπλοκης επιφάνειας μπορεί να γίνει προσεγγίζοντάς την με ένα σύνολο τριγώνων τα οποία είναι πολύ πιο εύκολο να τα επεξεργαστεί ο υπολογιστής. Έτσι, οι παραμετρικές καμπύλες όπως οι καμπύλες Bezier και splines, οι επιφάνειες Bezier και οι ισοεπιφάνειες αποτελούν αντικείμενα έρευνας σε αυτή την κλάση των

προβλημάτων. Από την άλλη, έντονη είναι η ερευνητική προσπάθεια εδώ για την ανίχνευση interfaces και γεωμετρικών δομών. Μια δημοφιλής μέθοδος είναι αυτή του συνόλου επιπέδων κατά την οποία μια κλειστή καμπύλη Γ αναπαρίσταται σαν το σύνολο των μηδενικών μια βοηθητικής συνάρτησης φ

$$\Gamma = \{(x, y) \mid \varphi(x, y) =)\}$$



Σχήμα 1.1: Η μέθοδος του μηδενικού επιπέδου η οποία με τη χρήση μιας συνάρτησης δύο μεταβλητών μπορεί να αναπαραστήσει τους μετασχηματισμούς μιας καμπύλης, η οποία είναι μια συνάρτηση μίας μεταβλητής.

και ο υπολογισμός της καμπύλης Γ γίνεται έμμεσα. Στο Σχ. 1.1 φαίνεται η αποτελεσματικότητα της μεθόδου για την αναπαράσταση μιας καμπύλης που διασπάται σε δύο διαφορετικές καμπύλες (η επάνω γραμμή του

σχήματος). Ένας τέτοιος μετασχηματισμός θα ήταν δύσκολο να παραμετροποιηθεί. Στην κάτω γραμμή του σχήματος φαίνεται η συνάρτηση φ(χ, y) και ένα επίπεδο που κινείται προς τα επάνω. Έτσι, η τομή του επιπέδου με την επιφάνεια ζ = φ(χ, y) δίνει το ζητούμενο μετασχηματισμό. Έτσι, αν η ταχύτητα του επιπέδου είναι υ, η συνάρτηση φt τη χρονική στιγμή t θα δίνεται φt =u|Δφ|

Τέλος, σε αυτήν την κλάση προβλημάτων ανήκουν και τα προβλήματα της υπολογιστικής τοπολογίας ή αλγοριθμικής τοπολογίας όπου το ζητούμενο είναι η εύρεση αποτελεσματικών αλγορίθμων για την επίλυση τοπολογικών προβλημάτων.

1.3 Ψηφιακή ή Διακριτή Γεωμετρία

Η διαχείριση διακριτών συνόλων (συνήθως από σημεία) τα οποία θεωρούνται ψηφιακές αναπαραστάσεις ή μοντέλα γεωμετρικών Ευκλείδειων δομών στη σύγχρονη υπολογιστική γεωμετρία είναι γνωστή με τον όρο Ψηφιακή Γεωμετρία. Χαρακτηριστικό παράδειγμα αποτελούν τα γραφικά υπολογιστών και η ανάλυση εικόνας.

Μερικά από τα χαρακτηριστικά προβλήματα που αντιμετωπίζονται στα πλαίσια αυτά είναι η κατασκευή διακριτών αναπαραστάσεων και μοντέλων γεωμετρικών δομών, η μελέτη των ιδιοτήτων διακριτών συνόλων, ο μετασχηματισμός των διακριτών αναπαραστάσεων (εικόνα και κίνηση), η ανακατασκευή γεωμετρικών δομών από τα διακριτά μοντέλα, η μελέτη διακριτών καμπυλών και επιφανειών, ο σχεδιασμός αλγορίθμων ανίχνευσης κ.α.. Ιδιαίτερο ενδιαφέρον παρουσιάζει η τομογραφία και ειδικά στο πεδίο της απεικονιστικής ιατρικής.

Σχετικά πρόσφατα ανακαλύφθηκε στο πεδίο της διακριτής γεωμετρίας η διακριτή διαφορική γεωμετρία η οποία εφαρμόζει τις ιδέες της κλασικής διαφορικής γεωμετρίας σε διακριτούς χώρους, στους οποίους το ρόλο των ομαλών καμπυλών και επιφανειών παίζουν τα πολύγωνα, τα πλέγματα και γενικά τα συμπλέγματα hyper-cubes (υπερτετράεδρο σε μία διάσταση είναι ένα ευθύγραμμο τμήμα, σε δύο διαστάσεις είναι ένα τρίγωνο, σε τρεις ένα τετράεδρο κ.ο.κ., βλέπε Σχ. 1.2). Ένα σύμπλεγμα υπερτετραέδρων K ορίζεται ως το σύνολο υπερτετραέδρων τέτοιων ώστε

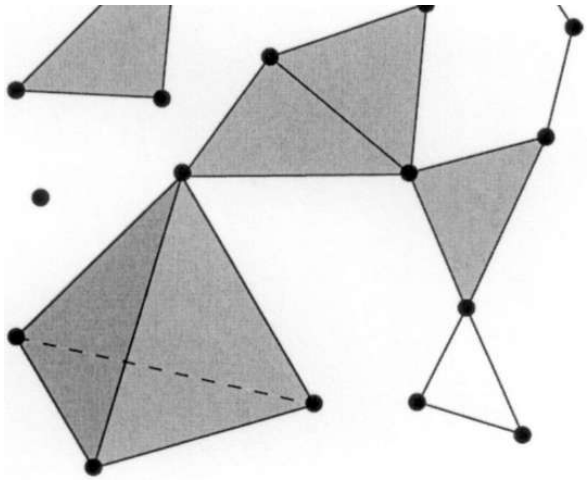
1. Κάθε έδρα ενός υπερτετραέδρου του K ανήκει στο K.

2. Η τομή δύο οποιονδήποτε υπερτετραέδρων σ1, σ2 είναι έδρα και των δύο υπερτετραέδρων.

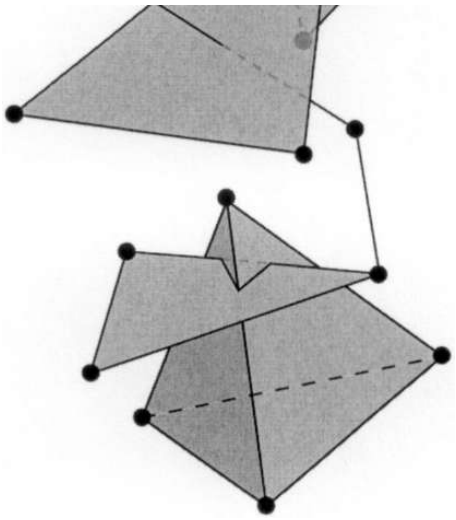
Ένα σύμπλεγμα k-υπερτετραέδρων είναι ένα σύμπλεγμα υπερτετραέδρων στο οποίο η μεγαλύτερη διάσταση οποιουδήποτε υπερτετραέδρου είναι k. Στο Σχ. 1.2 φαίνεται ένα σύμπλεγμα 3-υπερτετραέδρων ενώ στο Σχ. 1.3 φαίνεται ένα μη κανονικό σύμπλεγμα υπερτετραέδρων (η τομή και των δύο τεμνόμενων υπερτετραέδρων δεν αποτελεί έδρα και των δυο). Ένα ομογενές k-σύμπλεγμα είναι ένα σύμπλεγμα K στο οποίο όλα τα τετράεδρα με διάσταση μικρότερη από k είναι έδρες κάποιου τετραέδρου με διάσταση ακριβώς k. Για παράδειγμα, ένα 2-σύμπλεγμα το οποίο αποτελείται από ένα τρίγωνο και από ένα ευθύγραμμο τμήμα του οποίου η μία μόνο άκρη είναι κορυφή του τριγώνου δεν αποτελεί ομογενές 2-σύμπλεγμα. Μια απλή έδρα είναι ένα υπερτετράεδρο το οποίο δεν αποτελεί έδρα κανενός μεγαλύτερου υπερτετράεδρου. Τέλος, αν S είναι ένα σύνολο υπερτετραέδρων που ανήκουν σε ένα σύμπλεγμα, η κλειστότητα Cls του S

είναι το μικρότερο υποσύμπλεγμα που περιέχει το S (στο Σχ. 1.4 φαίνεται ότι η κλειστότητα ενός τριγώνου και μια ακμής είναι το τρίγωνο με τις τρεις κορυφές του και η ακμή με τα δύο άκρα της). Επίσης ορίζεται ο αστέρας του S , Sts ως το σύνολο των υπερτετραέδρων που έχουν κάποια έδρα τους να ανήκει στο S (βλ. Σχ. 1.5) και η ένωση του S είναι Lks = Clsts ~ Sts (βλέπε Σχ. 1.6).

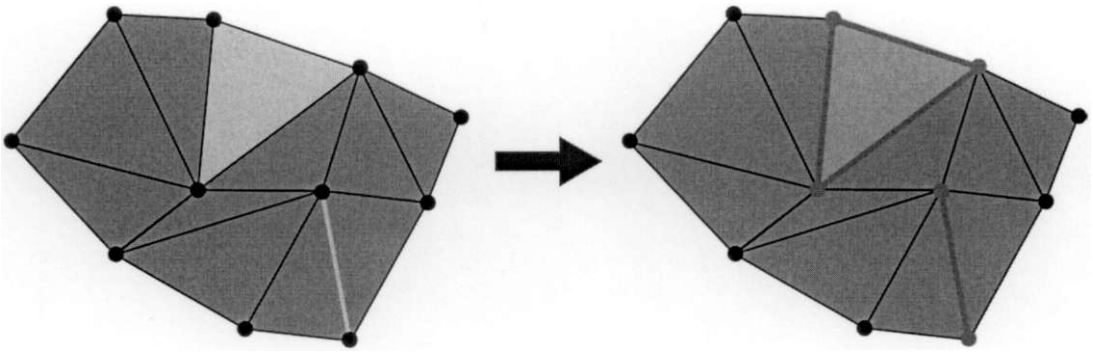
Τέλος, με το όρον εξωτερικός διακριτός λογισμός εννοούμε το λογισμό του εξωτερικού γινομένου διαφορίσιμων μορφών σε διαφορίσιμες διακριτές πολλαπλότητες. Για παράδειγμα, το θεώρημα του Stokes αναφέρει ότι το ολοκλήρωμα μιας (n-1)-μορφής ω στο περίβλημα dM μιας n-διάστατης πολλαπλότητας είναι ίσο με το ολοκλήρωμα της εξωτερικής παραγώγου dω της ω στην πολλαπλότητα M.



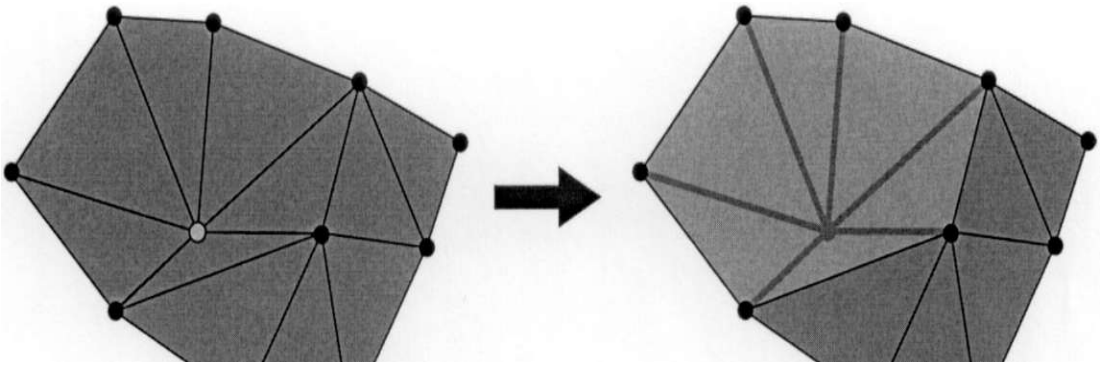
Σχήμα 1.2: Ένα σύμπλεγμα k-υπερτετραέδρων είναι ένα σύνολο από υπερτετράεδρα μέγιστης διάστασης k, στο οποίο, κάθε τομή δύο υπερτετραέδρων είναι ακμή και των δύο υπερτετραέδρων που τέμνονται.



Σχήμα 1.3: Σε αντίθεση με το σύμπλεγμα του Σχ. 1.2, εδώ έχουμε ένα σύνολο 3-υπερτετραέδρων στο οποίο η τομή των μελών δεν αποτελεί ακμή τους.



Σχήμα 1.4: Η κλειστότητα ενός τριγώνου σε ένα 2-σύμπλεγμα υπερτετραέδρων είναι το τρίγωνο με τις κορυφές του, ενώ μιας ακμής είναι η ακμή με τα δύο άκρα.



Σχήμα 1.5: Ο αστέρας μιας κορυφής σε ένα 2-σύμπλεγμα είναι όλα τα τρίγωνα των οποίων μια κορυφή είναι η δοσμένη.

Σχήμα 1.6: Η ένωση μιας κορυφής S είναι το σύνολο που περιέχει όλα τα τρίγωνα και τις ακμές τους που έχουν μια κορυφή το S μείον τον αστέρα του S.

Στην περίπτωση των διακριτών γεωμετριών, θεωρούμε ότι ο χώρος είναι ένα σύμπλεγμα υπερτετραέδρων. Μια k-μορφή σε αυτόν το χώρο είναι ένας γραμμικός τελεστής που δρα σε υπερτετράεδρα διάστασης k. Έτσι, μια 0-μορφή δρα σε σημεία με γραμμικό τρόπο, μια 1-μορφή δρα σε ευθύγραμμα τμήματα, κ.ο.κ.. Επίσης, το περίβλημα θL ενός υπερτετραέδρου L ορίζεται με φυσιολογικό τρόπο (για παράδειγμα το περίβλημα ενός ευθύγραμμου τμήματος με άκρα τα σημεία α, b είναι η τυπική διαφορά α-b. Αν τώρα S είναι ένα (k + 1)-υπό-σύμπλεγμα ενός συμπλέγματος T και ω μια fc-μορφή, τότε η διακριτή εξωτερική παράγωγος θω της ω είναι η μοναδική (k + 1)-μορφή για την οποία ισχύει

<dω|S> = <ω|M>

Μια γενίκευση της διακριτής γεωμετρίας είναι η πεπερασμένη γεωμετρία, δηλαδή ένα γεωμετρικό σύστημα ορισμένο σε ένα σύνολο από πεπερασμένα στοιχεία. Μια πεπερασμένη γεωμετρία μπορεί να έχει οποιαδήποτε πεπερασμένη διάσταση.

1.4 Δομές για ερωτήσεις περιοχής (range searching)

Έστω P ένα σύνολο από σημεία στο R^d και έστω R μία οικογένεια από υποσύνολα του R^d. Τα στοιχεία του R λέγονται περιοχές. Το πρόβλημα της ερώτησης περιοχής ζητά να προεπεξεργαστούμε τα στοιχεία του συνόλου P, σε μία δομή δεδομένων, έτσι ώστε για μία δοσμένη περιοχή $Q \in R$, το σύνολο $P \cap Q$ να μπορεί να αναφερθεί αποτελεσματικά, οπότε μιλάμε για ερώτηση αναφοράς περιοχής (range reporting query), ή το μέγεθος του συνόλου $P \cap Q$ να μπορεί να προσδιοριστεί αποτελεσματικά, οπότε μιλάμε για ερώτηση μέτρησης (αρίθμησης) περιοχής (range counting query).

Τυπικά παραδείγματα του R είναι:

1. το σύνολο όλων των δυνατών ορθογωνίων στο R^d,
2. το σύνολο όλων των δυνατών halfspaces στο R^d και
3. το σύνολο όλων των δυνατών simplexes στο R^d.

Οι ερωτήσεις αναφοράς περιοχής και μέτρησης περιοχής αποτελούν μόνο δύο στιγμιότυπα των δυνατών ερωτήσεων περιοχής. Άλλα παραδείγματα τέτοιων ερωτήσεων είναι οι ερωτήσεις ύπαρξης, όπου για δοσμένο $Q \in R$ θέλουμε να προσδιορίσουμε αν το σύνολο $P \cap Q$ είναι κενό, και οι ερωτήσεις εύρεσης ακραίου σημείου, όπου από το σύνολο $P \cap Q$ θα πρέπει να προσδιορίσουμε ένα σημείο στο οποίο μία ιδιότητα γίνεται μέγιστη ή ελάχιστη (λόγου χάρη το σημείο εκείνο που έχει την μέγιστη τιμή ως προς κάποια συντεταγμένη).

Σε ένα σύνολο από δημοσιεύσεις, μία ενιαία γενίκευση των παραπάνω ερωτήσεων περιοχής έχει προταθεί. Πιο συγκεκριμένα, σε κάθε σημείο $p \in R$ αντιστοιχίζεται ένα βάρος $w(p) \in S$, όπου $(S, +)$ είναι μία αντιμεταθετική ημιομάδα. Τότε κάθε είδους ερώτηση περιοχής μπορεί να αντιστοιχιστεί στο πρόβλημα του υπολογισμού των βαρών όλων των σημείων του P τα οποία βρίσκονται στην περιοχή ερώτησης: $\Sigma_{p \in C} w(p)$. Για παράδειγμα, στην ερώτηση μέτρησης περιοχής, η ημιομάδα $(S, +)$ είναι το σύνολο των φυσικών αριθμών εφοδιασμένο με την πράξη της πρόσθεσης και όλα τα βάρη θα είναι ίσα με 1. Στην περίπτωση των ερωτήσεων εύρεσης ακραίου σημείου η αντίστοιχη ημιομάδα θα είναι το σύνολο των πραγματικών αριθμών, με πράξη την λειτουργία της εύρεσης του μέγιστου μεταξύ δύο αριθμών. Στην περίπτωση των ερωτήσεων ύπαρξης μπορούμε να επιλέξουμε ως S το σύνολο των boolean μεταβλητών false, true,

εφοδιασμένο με την πράξη της λογικής διάζευξης (OR). Στην περίπτωση αυτή τα βάρη όλων των σημείων θα είναι "true". Τέλος, για τις ερωτήσεις αναφοράς περιοχής, μπορούμε να θεωρήσουμε ως S το δυναμοσύνολο του P εφοδιασμένο με την πράξη της ένωσης συνόλων, ενώ ως βάρος ενός οποιουδήποτε σημείου p θα μπορούσαμε να θεωρήσουμε το ίδιο το σημείο.

Ένας βασικός παράγοντας στην ομαδοποίηση δεδομένων που επηρεάζει την ταχύτητα ενός αλγορίθμου, είναι οι δομές που χρησιμοποιούνται για την αποθήκευση των δεδομένων. Αυτό οφείλεται στο γεγονός ότι κατά την εκτέλεση των περισσοτέρων αλγορίθμων, τα δεδομένα πρέπει να προσπελαστούν αρκετές φορές για την απάντηση ερωτημάτων πλησιέστερου γείτονα (nearest - neighbour queries) ή / και ερωτήματα αναζήτησης διαστήματος (range queries). Συνεπώς, είναι αναγκαία η χρήση μιας δομής που προσφέρει τη δυνατότητα να απαντήσει στα παραπάνω ερωτήματα στο μικρότερο δυνατό χρόνο.

Ένα κοινό στοιχείο των δομών αναζήτησης που θα αναφέρουμε στη συνέχεια είναι ότι βασίζονται στην διαμέριση του συνόλου P των αποθηκευμένων σημείων σε μία οικογένεια από κανονικά υποσύνολα του P και για κάθε κανονικό υποσύνολο C αποθηκεύουν το βάρος $w(C) = \Sigma_{p \in C} w(p)$. Δοθείσης λοιπόν μίας ερώτησης περιοχής r η δομή δεδομένων ψάχνει για μία υποοικογένεια $C_1,...,C_k$ των διακριτών κανονικών υποσυνόλων και στην συνέχεια υπολογίζει την τιμή $w(\{C_i\})$. Ο γρήγορος προσδιορισμός της κατάλληλης υποοικογένειας επιτυγχάνεται με την αποθήκευση επιπλέον βοηθητικής πληροφορίας στους κόμβους της δομής. Κατά κανόνα τα κανονικά υποσύνολα οργανώνονται σε μία δενδρική δομή, όπου κάθε κόμβος u της δομής συσχετίζεται με ένα κανονικό υποσύνολο A. το u αποθηκεύει το βάρος w(A) και επιπλέον πληροφορία. Μία ερώτηση περιοχής ψάχνει το δέντρο από πάνω προς τα κάτω, χρησιμοποιώντας την βοηθητική πληροφορία.

Κεφάλαιο 2

ΑΝΑΖΗΤΗΣΗ ΠΕΡΙΟΧΗΣ

2.1 Το πρόβλημα αναζήτησης περιοχής

2.2 Range Trees

2.3 k-d Trees

ΚΕΦΑΛΑΙΟ 2^{ο}: Αναζήτηση περιοχής

2.1 Το πρόβλημα αναζήτησης περιοχής

Το πρόβλημα αναζήτησης περιοχής [3] είναι ένα από τα θεμελιώδη προβλήματα στην υπολογιστική γεωμετρία. Δίνεται ένα σύνολο P από n σημεία δεδομένων στον πραγματικό χώρο d-διαστάσεων, R^d, και θεωρούμε ένα χώρο πιθανών περιοχών (π.χ., d-διαστάσεων ορθογώνια, σφαίρες, ημισφαίρες). Ο στόχος είναι να προεπεξεργαστούμε τα σημεία, έτσι ώστε, αν δίνεται οποιοδήποτε ερώτημα περιοχής Q, τα σημεία $P \cap Q$ να μπορούν να μετρηθούν ή αναφερθούν επαρκώς. Πιο γενικά, μπορεί κανείς να υποθέσει ότι στα σημεία έχουν ανατεθεί βάρη, και το πρόβλημα είναι να υπολογίσουμε το συσσωρευμένο βάρος των σημείων $P \cap Q$.

Υπάρχει μια πλούσια βιβλιογραφία για αυτό το πρόβλημα. Στην εργασία αυτή θεωρούμε την εκδοχή σταθμισμένης καταμέτρησης του προβλήματος. Μας ενδιαφέρουν εφαρμογές στις οποίες ο αριθμός των σημείων δεδομένων είναι επαρκώς μεγάλος ώστε κάποιος να επιλύει το πρόβλημα χρησιμοποιώντας μόνο το γραμμικό ή σχεδόν γραμμικό χώρο.

Για ορθογώνιες περιοχές, είναι γνωστό ότι μπορούν να χρησιμοποιηθούν για την επίλυση του προβλήματος δέντρα περιοχών σε $O(\log^{d-1} n)$ χρόνο με $O(\log^{d-1} n)$ χώρο. Οι Chazelle και Welzl [12] έδειξαν ότι οι ερωτήσεις τριγωνικής περιοχής μπορούν να λυθούν σε $O(\sqrt{n}\ \log n)$ χρόνο χρησιμοποιώντας $O(n)$ χώρο. Ο Matousek [16] έχει δείξει πώς να επιτύχουμε $O(n^{1-1/d})$ χρόνο επερώτησης για αναζήτηση simplex περιοχής με σχεδόν γραμμικό χώρο. Αυτό είναι κοντά στο κάτω όριο του Chazelle $\Omega(n^{1-1/d}/\log n)$ [11]. Για ερωτήσεις περιοχής ημιχώρου οι Bronnimann κ.α. [8] έχουν δώσει κάτω όριο $\Omega(n^{1-2/(d+1)})$ (αγνοώντας τους λογαριθμικούς

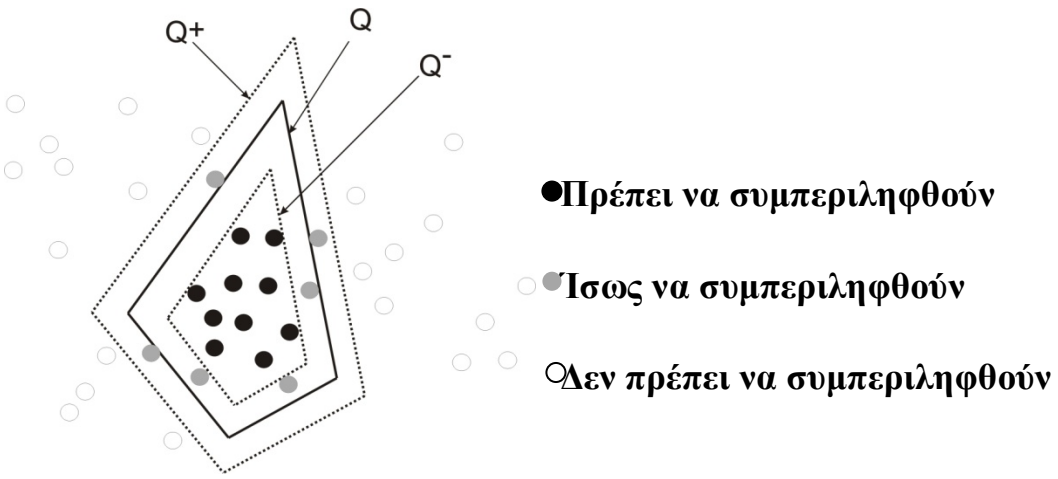
παράγοντες) όσον αφορά τον γραμμικό χώρο. Αυτό το κάτω όριο ισχύει και στις πιο γενικές περιπτώσεις των ερωτήσεων σφαιρικών περιοχών.

Δυστυχώς, τα επιχειρήματα κατώτερων ορίων νικούν κάθε λογική ελπίδα επίτευξης πολυλογαριθμικής απόδοσης για αυθαίρετες (μη ορθογώνιες) περιοχές. Αυτό υποδεικνύει ότι μπορεί να είναι χρήσιμο να θεωρήσουμε παραλλαγές του προβλήματος, οι οποίες μπορεί να επιτύχουν καλύτερους χρόνους λειτουργίας.

Στην εργασία αυτή θεωρούμε μια προσεγγιστική εκδοχή της αναζήτησης περιοχής. Αντί να προσεγγίζουμε το μέτρημα, θεωρούμε ότι η περιοχή είναι μια συγκεχυμένη περιοχή, και υποθέτουμε ότι σημεία που είναι «κοντά» στο όριο της περιοχής (σε σχέση με την διάμετρο της περιοχής) θα μπορούσαν να συμπεριληφθούν στην καταμέτρηση, ή και όχι.

Για να γίνει αυτή η ιδέα ακριβής, υποθέτουμε ότι οι περιοχές είναι οριοθετημένα σύνολα με οριοθετημένη πολυπλοκότητα. (Έτσι, τα αποτελέσματά μας δεν θα ισχύουν για αναζήτηση περιοχής ημιχώρου). Δεδομένης μια περιοχής Q διαμέτρου w, και δεδομένου ενός $\varepsilon>0$, ορίζουμε ότι η εσωτερική περιοχή Q^- είναι ο γεωμετρικός τόπος των σημείων των οποίων η Ευκλείδεια απόσταση από ένα σημείο εξωτερικό του Q είναι τουλάχιστον $w\varepsilon$, και καθορίζουμε την εξωτερική περιοχή Q^+ να είναι ο γεωμετρικός τόπος των σημείων των οποίων η απόσταση από ένα εσωτερικό σημείο του Q είναι το μέγιστο $w\varepsilon$. Καθορίζουμε ότι μια νόμιμη απάντηση σε μια ερώτηση ε-προσεγγιστικής περιοχής είναι το $w(P')$ για κάθε υποσύνολο P' τέτοιο ώστε:

$$P \cap Q^- \subseteq P' \subseteq P \cap Q^+.$$



Σχήμα 2.1: Ερωτήσεις αναζήτησης προσεγγιστικής περιοχής

Αυτός ο ορισμός επιτρέπει σφάλματα δύο όψεων, παραλείποντας να μετρήσει τα σημεία που είναι μόλις στο εσωτερικό της περιοχής, και μετρώντας τα σημεία που είναι ελάχιστα έξω από την περιοχή. Είναι τετριμμένο να τροποποιήσουμε τον αλγόριθμο, έτσι ώστε να παράγει μονόπλευρα σφάλματα.

Τα αποτελέσματα μπορούν να γενικευτούν σε άλλους ορισμούς των Q^- και Q^+ υπό την προϋπόθεση ότι το ελάχιστο όριο διαχωρισμού απόστασης με το Q είναι τουλάχιστον we. Υποθέτουμε ότι παρέχονται τα ακόλουθα αρχέτυπα δοκιμής περιοχών:

Ένταξη σημείου στο Q: αν ένα σημείο p βρίσκεται μέσα στο Q,

Τομή κουτιού με το Q^-: αν ένα ορθογώνιο ευθυγραμμισμένο με τους άξονες έχει μια μη κενή τομή με το Q^-,

Ένταξη κουτιού στο Q^+: αν ένα ορθογώνιο ευθυγραμμισμένο με τους άξονες περιέχεται εντός του Q^+.

Οι χρόνοι λειτουργίας μας δίνονται υποθέτοντας καθένας από αυτούς μπορεί να υπολογιστεί σε σταθερό χρόνο. Ο αλγόριθμος μας μπορεί εύκολα να γενικευθεί ώστε να αναφέρει το σύνολο των σημείων που βρίσκονται εντός της περιοχής, και ο χρόνος λειτουργίας αυξάνεται ώστε να περιλαμβάνει το χρόνο που χρειάζεται για να εξάγει τα σημεία.

Η προσεγγιστική αναζήτηση περιοχής είναι πιθανώς ενδιαφέρουσα μόνο για «φαρδιές» περιοχές. Ο Overmars [18] καθορίζει ένα αντικείμενο Q να είναι k-φαρδύ αν και μόνο αν για κάθε σημείο p στο Q, και για κάθε μπάλα B με το σημείο p ως κέντρο που δεν περιέχει πλήρως το Q στο εσωτερικό της, το ποσοστό της B που καλύπτεται από το Q είναι τουλάχιστον 1/k. Για περιοχές που δεν είναι k-φαρδιές, η διάμετρος της περιοχής μπορεί να είναι αυθαίρετα μεγάλη σε σχέση με το πάχος της περιοχής σε κάθε σημείο της. Παρόλα αυτά, υπάρχουν πολλές εφαρμογές αναζήτησης περιοχών που περιλαμβάνουν φαρδιές περιοχές.

Υπάρχουν αρκετοί λόγοι που αξίζει να εξεταστεί αυτή η διατύπωση του προβλήματος. Υπάρχουν πολλές εφαρμογές όπου τα δεδομένα είναι ανακριβή, και οι περιοχές οι ίδιες είναι ασαφείς. Για παράδειγμα, ο χρήστης ενός γεωγραφικού πληροφοριακού συστήματος που θέλει να μάθει πόσες μονοκατοικίες βρίσκονται σε ακτίνα 60 μιλίων του Μανχάταν, μπορεί να είναι αρκετά ευχαριστημένος με μια απάντηση που είναι ακριβής για απόσταση λίγων μιλίων.

Επίσης, τα ερωτήματα περιοχής χρησιμοποιούνται συχνά ως μέρος μιας αρχικής διαδικασίας φιλτραρίσματος σε πολύ μεγάλα σύνολα δεδομένων, μετά την οποία θα εφαρμοστεί ένα πιο πολύπλοκο τεστ στα σημεία εντός της περιοχής. Σε αυτές τις εφαρμογές, ο χρήστης μπορεί να δεχτεί ένα ευρύ φίλτρο που τρέχει πιο γρήγορα. Ο χρήστης είναι ελεύθερος να ρυθμίσει την τιμή του ε με όποια ακρίβεια επιθυμεί (χωρίς την ανάγκη να εφαρμοστεί

και πάλι προεπεξεργασία), κατανοώντας την αντίστοιχη επίπτωση στους χρόνους λειτουργίας.

Στην εργασία αυτή δείχνουμε ότι αν επιτρέψουμε την προσεγγιστική αναζήτηση περιοχών, είναι πιθανό να επιτευχθούν σημαντικές βελτιώσεις στους χρόνους λειτουργίας, τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο. Δείχνεται ότι για μια συγκεκριμένη διάσταση d, μετά από προεπεξεργασία O(n log n) και με χώρο O(n), οι ερωτήσεις ε-προσεγγιστικών περιοχών μπορούν να απαντηθούν σε χρόνο $O(\log n + (1/\varepsilon)^d)$. Με την προϋπόθεση ότι οι περιοχές είναι κυρτές, δείχνουμε ότι μπορούμε να το βελτιώσουμε σε $O(\log n + (1/\varepsilon)^{d-1})$. Κάποια σημαντικά στοιχεία είναι τα εξής:

- Η δομή δεδομένων και ο χρόνος προεπεξεργασίας είναι ανεξάρτητα από το χώρο των πιθανών περιοχών και από το ε (υποθέτοντας ότι τα ανωτέρω αρχέτυπα δοκιμής περιοχής παρέχονται όταν εφαρμόζεται το ερωτήμα).
- Ο χώρος και ο χρόνος προεπεξεργασίας δεν περιέχουν εκθετικούς παράγοντες ως προς τη διάσταση. Ο χώρος είναι της τάξης O(dn) και ο χρόνος προεπεξεργασίας είναι της τάξης O(dn log n).
- Οι αλγόριθμοι είναι σχετικά απλοί και έχουν τεθεί σε εφαρμογή. Η δομή δεδομένων είναι μια παραλλαγή της γνωστής δομής τετραδικού δένδρου δεδομένων.
- Πειραματικά αποτελέσματα μας δείχνουν ότι ακόμη και για ομοιόμορφα κατανεμημένα σημεία στην διάσταση 2, υπάρχει μια σημαντική βελτίωση στο χρόνο λειτουργίας, αν επιτραπεί ένα μικρό προσεγγιστικό λάθος. Επιπλέον, το μέσο σφάλμα που παράγεται από τον αλγόριθμο είναι τυπικά πολύ μικρότερο από το επιτρεπόμενο σφάλμα ε.

2.2 Range Trees

Τα Range Trees είναι δέντρα που έχουν προταθεί από τον Bentley [23, 24]. Μολονότι εμφανίζουν καλό άνω φράγμα στο χρόνο απάντησης μίας αναζήτησης περιοχής, ωστόσο εμφανίζουν μεγάλο κόστος σε χρόνο προεπεξεργασίας και χώρο. Συνοπτικά, το κόστος των Range Trees είναι:

$$P(n, d) = O(n \log^{d-1} n),$$

$$S(n, d) = O(n \log^{d-1} n),$$

$$Q(n, d) = O(\log^{d} n + s),$$

όπου με P(n,d) συμβολίζεται το υπολογιστικό κόστος για την προεπεξεργασία n σημείων, διάστασης d, με S(n, d) συμβολίζεται ο χώρος που χρειάζεται για την αποθήκευση της δομής, ενώ με Q(n, d) ο χρόνος που χρειάζεται για την απάντηση ενός μίας αναζήτησης περιοχής, το οποίο εντοπίζει s σημεία. Ένα Range Tree αποτελείται από ένα δέντρο διαστημάτων (segment tree), που αποτελεί τη βασική δομή, καθώς και από επιπλέον δέντρα διαστημάτων που αποτελούν τις δευτερεύουσες δομές.

Τα δέντρα διαστημάτων χειρίζονται διαστήματα του R και πρόκειται στην ουσία για δυαδικά δέντρα. Έστω x_1, x_2, ..., x_n τα σημεία που θα εισαχθούν στο δέντρο. Ένα δέντρο διαστημάτων $T(x_1, x_n)$ κατασκευάζεται αναδρομικά ως εξής: Στη ρίζα του δέντρου r ορίζονται οι τιμές $B[r] = x_1$ και $E[r] = x_n$ σαν αρχή και πέρας του διαστήματος αντίστοιχα. Αν $n - 1 > 1$, τότε το δέντρο δεν είναι τετριμμένο. Τα υποδέντρα του κατασκευάζονται αναθέτοντας το διάστημα $(x_1, x_{\lfloor \frac{n}{2} \rfloor})$ στη ρίζα του αριστερού υποδέντρου και το διάστημα $(x_{\lfloor \frac{n}{2} \rfloor + 1}, x_n)$ στο δεξί υποδέντρο. Οι ρίζες των υποδέντρων

εισάγονται σαν αριστερό και δεξί παιδί της ρίζας αντίστοιχα. Ένα δέντρο διαστημάτων είναι ισορροπημένο και έχει βάθος $\lceil \log(n-1) \rceil$. Επιπλέον, πρόκειται για μια δυναμική δομή που μπορεί να υποστηρίξει εισαγωγές και εξαγωγές στοιχείων [19].

Εν συνεχεία περιγράφεται η διαδικασία κατασκευής ενός Range Tree για ένα σύνολο S = (x_1, x_2, ..., x_n) όπου $x_i \in R^d$. Η βασική δομή ενός Range Tree είναι ένα δέντρο διαστημάτων που κατασκευάζεται χρησιμοποιώντας τις τιμές της πρώτης συντεταγμένης των στοιχείων του S. Περιέχει δηλαδή τα στοιχεία $\{x_{11}, x_{21}, x_{31}, ..., x_{n1}\}$, όπου με x_{ij} συμβολίζουμε την j-οστή συντεταγμένη του x_i. Ένας κόμβος u της βασικής δομής, πέραν του διαστήματος το οποίο περιέχει, έχει και ένα δείκτη σε μια δευτερεύουσα δομή. Η δευτερεύουσα αυτή δομή είναι ένα δέντρο διαστημάτων κατασκευασμένο με βάση τη δεύτερη συντεταγμένη των στοιχείων που περιέχονται στο υποδέντρο με ρίζα τον κόμβο u. Εν γένει, σε κάθε κόμβο της βασικής δομής αντιστοιχίζεται και ένα δεύτερο δέντρο διαστημάτων. Όμοια, η δευτερεύουσα αυτή δομή περιέχει στους κόμβους της δείκτες σε δομές με βάση όλες τις τρίτες συντεταγμένες. Η αναδρομική κατασκευή των δευτερευουσών δομών συνεχίζεται μέχρι να εξαντληθούν οι διαστάσεις.

2.2.1 Ερωτήματα αναζήτησης περιοχής

Ο αλγόριθμος για την εύρεση των σημείων του συνόλου S που ανήκουν σε ένα διάστημα R = [a, b] του R^d είναι αναδρομικός. Έστω [α_j, b_j] τα άκρα του R στην j-οστή διάσταση. Ο αλγόριθμος ξεκινάει ελέγχοντας αν το [α_1, b_1] βρίσκεται εντός του διαστήματος [B[r], E[r]), όπου με r συμβολίζεται η ρίζα του δέντρου. Αν το διάστημα της ρίζας βρίσκεται εξ ολοκλήρου εντός του διαστήματος R, τότε το ερώτημα μεταφέρεται στη δευτερεύουσα δομή,

ελέγχοντας πλέον τις δεύτερες συντεταγμένες του διαστήματος R. Διαφορετικά εξετάζεται η τιμή στην οποία το διάστημα της ρίζας υποδιαιρείται στα δύο υποδιαστήματα των παιδιών του. Αν το διάστημα R είναι εξολοκλήρου αριστερά της τιμής αυτής, τότε η διαδικασία συνεχίζεται με το αριστερό παιδί της ρίζας, ενώ αν το διάστημα R είναι εξολοκλήρου δεξιά της τιμής αυτής, τότε η διαδικασία συνεχίζεται με το δεξί παιδί της ρίζας. Αν το διάστημα R περιέχει την τιμή αυτή, η διαδικασία επαναλαμβάνεται αναδρομικά και με τα δύο παιδιά της ρίζας. Η διαδικασία αυτή ακολουθείται για ολόκληρο το δέντρο αλλά με τον ίδιο τρόπο εφαρμόζεται και στις δευτερεύουσες δομές. Έστω u ένας κόμβος του δέντρου για την διάσταση i, LSON[u], RSON[u] το αριστερό και το δεξί παιδί του u αντίστοιχα, M[u] η μεσαία τιμή του κόμβου u (αυτή για στην οποία διαχωρίζεται στο αριστερό και στο δεξί παιδί του) και r_u η ρίζα της δευτερεύουσας δομής που αντιστοιχεί στον κόμβο u (προφανώς πρόκειται για τη ρίζα ενός δέντρου διαστημάτων για την i + 1 διάσταση των σημείων του κόμβου u). Συνοπτικά, ο αλγόριθμος δίνεται από τον παρακάτω ψευδοκώδικα.

Αλγόριθμος 2.1 Αναζήτηση Διαστήματος(R, u, i)

ΑΝ $\alpha_i \leq$ B[u] ΚΑΙ E[u] < b_i **ΤΟΤΕ**

    Αναζήτηση Διαστήματος(R, r_u, i+1);

ΑΛΛΙΩΣ ΑΝ b_i < M[r] **ΤΟΤΕ**

    Αναζήτηση Διαστήματος(R, LSON[u], i);

ΑΛΛΙΩΣ ΑΝ M[r] ≤ α_i **ΤΟΤΕ**

    Αναζήτηση Διαστήματος(R, RSON[u], i);

ΑΛΛΙΩΣ

Αναζήτηση Διαστήματος(R, LSON[u], i);

Αναζήτηση Διαστήματος(R, RSON[u], i);

ΤΕΛΟΣ ΑΝ

2.3 k-d Trees

Τα k-d Trees είναι από τα πλέον διαδεδομένα δέντρα. Τα δέντρα αυτά χωρίζουν το χώρο στον οποίο βρίσκονται τα σημεία ανάλογα με την κατανομή τους σ' αυτόν. Ξεκινώντας από ένα αρχικό χωρίο που περικλείει τα σημεία $x_1, x_2, ..., x_n \in R^d$ του συνόλου δεδομένων S, ο αλγόριθμος κατασκευής ενός k-d δέντρου επιλέγει ένα στοιχείο του S και με βάση αυτό χωρίζει το αρχικό χωρίο σε δύο μικρότερα. Η διαδικασία αυτή επαναλαμβάνεται μέχρι το δέντρο να περιέχει το πολύ ένα προκαθορισμένο πλήθος σημείων σε κάθε κόμβο.

Σύμφωνα με τους Bentley και Friedman, για την επιλογή του σημείου όπως επίσης και για τον τρόπο που θα χωριστεί το χωρίο που θα χρησιμοποιηθεί για την κατασκευή του δέντρου είναι δυνατή η επιλογή οποιουδήποτε ευρετικού [24]. Ωστόσο, η συνηθέστερη επιλογή είναι ο διαχωρισμός να γίνεται από ένα υπερεπίπεδο, κάθετο στον άξονα μιας συντεταγμένης του χωρίου, το οποίο περνάει από το μεσαίο (median) όλων των σημείων του τρέχοντος κόμβου, όσον αφορά την συγκεκριμένη συντεταγμένη του σημείου. Η διαδικασία αυτή επαναλαμβάνεται για όλες τις συντεταγμένες, ξεκινώντας από την πρώτη. Το υπολογιστικό κόστος, ο χώρος αποθήκευσης και ο χρόνος απάντησης μίας αναζήτησης περιοχής για ένα k-d Tree είναι:

$$P(n, d) = O(n \log n),$$

$$S(n, d) = O(nd),$$

$$Q(n, d) \leq O(n^{1-1/d} + s).$$

Στην εργασία των Bentley και Friedman [24], για συγκεκριμένα μεγέθη του S έχουν δοθεί καλύτερες πολυπλοκότητες για το Q(n, d).

Για να κατασκευαστεί ένα k-d Tree σύμφωνα με τα παραπάνω, όλα τα σημεία του συνόλου δεδομένων διατάσσονται κατ' αύξουσα σειρά σύμφωνα με την πρώτη συντεταγμένη. Εν συνεχεία, το χωρίο χωρίζεται σύμφωνα με το μεσαίο στοιχείο (x_p) της λίστας αυτής. Το χωρίο που περιγράφει το χώρο με πρώτη συντεταγμένη μικρότερη του x_p εισάγεται ως αριστερό παιδί της ρίζας, ενώ αντίστοιχα το χωρίο που περιγράφει το χώρο με πρώτη συντεταγμένη μεγαλύτερη του x_p εισάγεται ως δεξί παιδί της ρίζας. Η διαδικασία αυτή συνεχίζεται στο επόμενο επίπεδο του δέντρου για τη δεύτερη συντεταγμένη.

Έστω S(u) η λίστα όλων των σημείων του κόμβου u, P(u) το σημείο x_p σύμφωνα με το οποίο χωρίζεται το χωρίο, t(u) η συντεταγμένη κατά την οποία πραγματοποιείται η τομή του χωρίου του κόμβου u και M(u) η τιμή της συντεταγμένης στην οποία διαχωρίζεται το χωρίο του ιδίου κόμβου. Επίσης, με LSON[u] συμβολίζεται το αριστερό παιδί και με RSON[u] το δεξί παιδί του κόμβου u. Η διαδικασία της κατασκευής ενός k-d Tree για το σύνολο δεδομένων S μπορεί να περιγραφεί από τον παρακάτω ψευδοκώδικα.

Αλγόριθμος 2.2 Κατασκευή Δέντρου(u, i)

t(u) = i

S(u). Ταξινόμηση σε αύξουσα σειρά σύμφωνα με την συντεταγμένη(t(u));

P(u) = S(u). Βρες Το Μεσαίο Στοιχείο();

$M(u) = P(u)_{t(u)}$;

ΓΙΑ ΚΑΘΕ Στοιχείο x_i του S(u) **ΚΑΝΕ**

ΑΝ $x_{i,\, t(u)} < M(u)$ **ΤΟΤΕ**

S (LSON[u]). Εισήγαγε(x_i);

ΑΛΛΙΩΣ ΑΝ $x_{i,\, t(u)} > M(u)$ **ΤΟΤΕ**

S (RSON[u]). Εισήγαγε(x_i);

ΤΕΛΟΣ ΑΝ

ΤΕΛΟΣ ΓΙΑ

ΑΝ S (LSON[u]) όχι κενή **ΤΟΤΕ**

Κατασκευή Δέντρου(LSON[u], i + 1);

ΤΕΛΟΣ ΑΝ

ΑΝ S(RSON[u]) όχι κενή **ΤΟΤΕ**

Κατασκευή Δέντρου(RSON[u], i + 1);

ΤΕΛΟΣ ΑΝ

ΕΠΕΣΤΡΕΨΕ: u

Παράλληλα με τις παραπάνω διαδικασίες πρέπει να ανανεώνεται το χωρίο R(u) το οποίο περιγράφεται από ένα κόμβο u. Για κάποιο u ισχύει R(u) = $[x_{1,\ min}, x_{1,\ max}]$ x $[x_{2,\ min}, x_{2,\ max}]$ x … x $[x_{d,\ min}, x_{d,\ max}]$ με $x_{i,\ min}$ να συμβολίζει τη μικρότερη τιμή της συντεταγμένης i στο S(u) και $x_{i,\ max}$ να συμβολίζει τη μεγαλύτερη τιμή της ίδιας συντεταγμένης στο S(u). Το χωρίο σε ένα κόμβο u χωρίζεται στα χωρία:

$$[x_{1,\ min}, x_{1,\ max}] \text{ x } [x_{2,\ min}, x_{2,\ max}] \text{ x } \dots \text{ x } [x_{t(u),\ min}, M(u)) \text{ x } \dots \text{ x } [x_{d,\ min}, x_{d,\ max}]$$

για το αριστερό υποδέντρο και

$$[x_{1,\ min}, x_{1,\ max}] \text{ x } [x_{2,\ min}, x_{2,\ max}] \text{ x } \dots \text{ x } [M(u), x_{t(u),\ max}] \text{ x } \dots \text{ x } [x_{d,\ min}, x_{d,\ max}]$$

για το δεξί υποδέντρο.

2.3.1 Ερωτήματα αναζήτησης περιοχής

Η διαδικασία της απάντησης σε ερωτήματα αναζήτησης περιοχής αποτελεί μια συνάρτηση η οποία ξεκινάει από τη ρίζα του δέντρου. Σε κάθε κόμβο u ελέγχεται αν το χωρίο το οποίο συσχετίζεται με τον κόμβο αυτόν, R(u), έχει κάποια επικάλυψη με το διάστημα αναζήτησης Q. Εφόσον δεν υπάρχει τομή του R(u) με το Q, ο κόμβος u δεν περιέχει σημεία που κείνται μέσα στο Q. Αν το σημείο P(u) στο οποίο έχει χωριστεί το χωρίο R(u), βρίσκεται στα δεξιά της τομής του διαστήματος αναζήτησης, τότε η αναζήτηση συνεχίζεται μόνο στο αριστερό παιδί του κόμβου. Αντιθέτως, αν το σημείο P(u) βρίσκεται στα αριστερά της παραπάνω τομής, τότε η αναζήτηση συνεχίζεται μόνο στο δεξί παιδί του κόμβου. Στην τελευταία περίπτωση, όπου υπάρχει τομή του διαστήματος αναζήτησης τόσο με το R(LSON[u]), όσο και με το R(RSON[u]), ελέγχεται αν το σημείο P(u) ανήκει στο Q. Αν το P(u) ανήκει στο Q, εισάγεται στη λίστα των αποτελεσμάτων. Η

διαδικασία συνεχίζεται και στα δύο παιδιά του κόμβου u. Ο αλγόριθμος αναζήτησης τερματίζει όταν επισκεφτεί φύλλο του δέντρου. Αν υποτεθεί ότι η τομή του διαστήματος αναζήτησης με το χωρίο R(u) στην i συντεταγμένη συμβολίζεται με [Τομή$_1$, Τομή$_2$], ο αλγόριθμος αναζήτησης συνοψίζεται στον παρακάτω ψευδοκώδικα:

Αλγόριθμος 2.3 Αναζήτηση Διαστήματος(Q, u, Λίστα Αποτελεσμάτων)

Τομή = Υπολόγισε Την Τομή(Q, R(u));

ΑΝ Τομή είναι κενή **ΤΟΤΕ**

Τερμάτισε;

ΤΕΛΟΣ ΑΝ

ΑΝ u είναι φύλλο **ΤΟΤΕ**

ΓΙΑ ΚΑΘΕ u ∈ S(u) **ΚΑΝΕ**

ΑΝ u ∈ Q **ΤΟΤΕ**

Λίστα Αποτελεσμάτων. Εισήγαγε(u);

ΤΕΛΟΣ ΑΝ

ΤΕΛΟΣ ΓΙΑ

ΤΕΛΟΣ ΑΝ

ΑΝ Τομή$_1$ < M(u) **ΤΟΤΕ**

Αναζήτηση Διαστήματος(Q, LSON[u], Λίστα Αποτελεσμάτων);

ΑΛΛΙΩΣ ΑΝ Τομή$_2$ > M(u) **ΤΟΤΕ**

Αναζήτηση Διαστήματος(Q, RSON[u], Λίστα Αποτελεσμάτων);

ΑΛΛΙΩΣ

ΑΝ P(u) ∈ Q **ΤΟΤΕ**

Λίστα Αποτελεσμάτων. Εισήγαγε(P(u));

ΤΕΛΟΣ ΑΝ

Αναζήτηση Διαστήματος(Q, LSON[u], Λίστα Αποτελεσμάτων);

Αναζήτηση Διαστήματος(Q, RSON[u], Λίστα Αποτελεσμάτων);

ΤΕΛΟΣ ΑΝ

Κεφάλαιο 3

ΠΡΟΣΕΓΓΙΣΤΙΚΗ ΑΝΑΖΗΤΗΣΗ ΠΕΡΙΟΧΗΣ

3.1 Balanced Box-Decomposition Tree

3.2 Η δυναμική δομή δεδομένων «quadtreap»

ΚΕΦΑΛΑΙΟ 3^{ο}: Προσεγγιστική αναζήτηση περιοχής

3.1 Balanced Box-Decomposition Tree

Σε αυτή την ενότητα θα εξετάσουμε τη δομή των δεδομένων balanced box-decomposition δέντρο (ή BBD-δέντρο) για το σύνολο S των δεδομένων σημείων. Το BBD-δέντρο [4] είναι μια ισορροπημένη παραλλαγή μιας σειράς από γνωστές δομές δεδομένων που βασίζονται σε ιεραρχική υποδιαίρεση του χώρου σε παραλληλόγραμμες περιοχές.

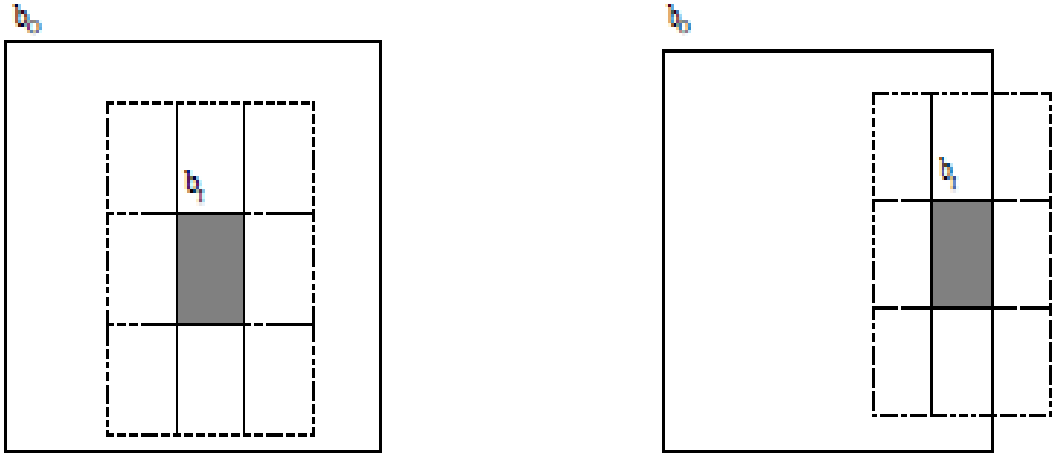
Όπως τα k-d trees, έτσι και τα Balanced Box Decomposition (BBD) δέντρα βασίζονται στην ιεραρχική διαμέριση του χώρου[4]. Χρησιμοποιώντας τα δέντρα αυτά, ο χώρος χωρίζεται αναδρομικά σε ένα σύνολο από κελιά (cells), κάθε ένα από τα οποία είναι είτε ένα d-διάστατο παραλληλεπίπεδο, είτε η συνολοθεωρητική διαφορά δύο παραλληλεπίπεδων του d-διάστατου χώρου που το ένα είναι μέσα στο άλλο. Κάθε κόμβος συσχετίζεται με ένα κελί και κατ' αυτό τον τρόπο συσχετίζεται εμμέσως με τα σημεία του κελιού, ενώ τα φύλλα του δέντρου ορίζουν μια διαμέριση του χώρου. Συνολικά ένα BBD δέντρο έχει O(n) κόμβους και μπορεί να κατασκευαστεί σε χρόνο O(dn log n). Ένα τέτοιο δέντρο έχει ύψος O(log n) διαχωρίζοντας το χώρο σε χωρία παράλληλα στους άξονες του συστήματος συντεταγμένων που χρησιμοποιείται.

Στα χωρία αυτά η αναλογία της μακρύτερης πλευράς προς την μικρότερη είναι φραγμένη, προκειμένου το δέντρο να μην περιγράφει στους κόμβους του πολύ «λεπτές» περιοχές του χώρου. Αντιθέτως, ορίζοντας ένα άνω φράγμα στο λόγο της μεγαλύτερης πλευράς προς τη μικρότερη εξασφαλίζεται ότι ένας κόμβος θα ορίζει μια μη τετριμμένη περιοχή του χώρου. Το μέγεθος s ενός d-διάστατου παραλληλεπιπέδου είναι το μήκος της μεγαλύτερης πλευράς του. Ως κουτί (box) ορίζεται ένα παραλληλεπίπεδο με το λόγο της μεγαλύτερης πλευράς προς τη μικρότερη

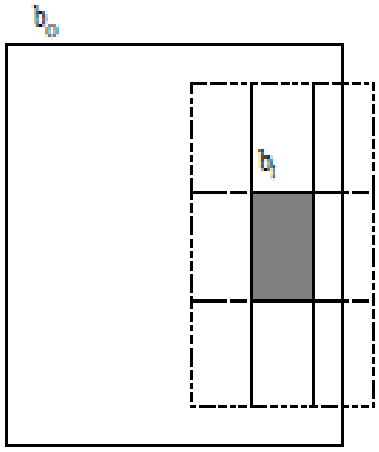
φραγμένο από μια σταθερά. Ο λόγος αυτός καλείται λόγος πλευρών (aspect ratio). Τελικά ένα κελί αποτελείται απ' ένα κουτί ή μια συνολοθεωρητική διαφορά δύο κουτιών, το ένα μέσα στο άλλο, που καλούνται το εξωτερικό κουτί (outer box) και το εσωτερικό κουτί (inner box). Τα κελιά είναι κλειστά χωρία και τα σημεία που βρίσκονται πάνω στο όριο δύο κελιών μπορούν να συσχετιστούν με οποιοδήποτε από τα δύο κελιά. Το μέγεθος ενός κελιού ορίζεται ως το μέγεθος του εξωτερικού του κουτιού.
Μια βασική ιδιότητα, απαραίτητη για τα BBD δέντρα είναι η εγγύτητα (stickiness). Η ιδιότητα αυτή αποτρέπει το εσωτερικό κουτί ενός κελιού να είναι περίπου ίσο με το εξωτερικό κουτί του ίδιου κελιού και συνεπώς να υπάρχουν τετριμμένες διαμερίσεις. Η ιδιότητα αυτή είναι απαραίτητη για την θεωρητική θεμελίωση των BBD δέντρων [4].

Ορισμός (Εγγύτητα) Ένα εσωτερικό κουτί $b_I = [x_{I1}, y_{I1}] \times [x_{I2}, y_{I2}] \times \ldots \times [x_{In}, y_{In}]$ λέγεται ότι διατηρεί την εγγύτητα με το εξωτερικό κουτί $b_O = [x_{O1}, y_{O1}] \times [x_{O2}, y_{O2}] \times \ldots \times [x_{On}, y_{On}]$ που το περιβάλλει, αν και μόνον αν $x_{Ii} - x_{Oi} = 0$ ή $x_{Ii} - x_{Oi} >= w_i$ και $y_{Oi} - y_{Ii} = 0$ ή $y_{Oi} - y_{Ii} >= w_i$ με $w_i = y_{Ii} - x_{Ii}$ για κάθε συντεταγμένη i.

Περιγραφικά, ένα διάστημα διατηρεί την εγγύτητα με το εξωτερικό του διάστημα αν έχουν το ίδιο άκρο ή η απόσταση του άκρου του εσωτερικού διαστήματος από το άκρο του εξωτερικού διαστήματος είναι τουλάχιστον όσο το μήκος του εσωτερικού διαστήματος για τη συγκεκριμένη διάσταση. Αν όλα τα διαστήματα που ορίζουν το εσωτερικό κουτί διατηρούν την εγγύτητα με τα αντίστοιχα διαστήματα που ορίζουν το εξωτερικό κουτί, τότε τα δύο κουτιά λέγεται ότι διατηρούν την εγγύτητα.



Σχήμα 3.1: Οι δύο περιπτώσεις κατά τις οποίες διατηρείται η εγγύτητα. Τα διακεκομμένα παραλληλόγραμμα έχουν ακριβώς τις ίδιες διαστάσεις με το b_I.



Σχήμα 3.2: Περίπτωση κατά την οποία δεν υπάρχει εγγύτητα.

3.1.1 Κατασκευή ενός BBD Tree

Σε αντίθεση με τα range trees και τα k-d trees, τα παιδιά ενός κόμβου ενός BBD δέντρου μπορεί να προκύψουν με δύο τρόπους. Ο πρώτος τρόπος καλείται διαχωρισμός (split) ενώ ο δεύτερος καλείται συρρίκνωση (shrink). Ο πρώτος τρόπος δίνει δύο κόμβους με όνομα «χαμηλό» και «ψηλό» παιδί, ενώ ο δεύτερος δύο κόμβους με όνομα «εσωτερικό» και «εξωτερικό» παιδί. Συνηθίζεται το χαμηλό παιδί να αποκαλείται εσωτερικό παιδί και το ψηλό να αποκαλείται εξωτερικό για λόγους απλούστευσης. Ανεξαρτήτως του τρόπου που χρησιμοποιήθηκε για να δημιουργηθούν τα παιδιά ενός κόμβου, πρέπει να πληρούνται τρεις κανόνες:

1. Πρέπει να ικανοποιείται ο λόγος πλευρών.
2. Αν ο πατέρας έχει ένα εσωτερικό κουτί, τότε αυτό είναι εξ ολοκλήρου εντός του ενός παιδιού.
3. Τα εσωτερικά κουτιά διατηρούν πάντα την εγγύτητα με τα εξωτερικά τους κουτιά.

Μια βασική παράμετρος είναι η επιλογή του κατάλληλου τρόπου για τη δημιουργία των παιδιών ενός κόμβου. Σε σχέση με την συρρίκνωση, ο διαχωρισμός (split) είναι ένας απλούστερος και πολύ πιο οικονομικός, από υπολογιστικής πλευράς, τρόπος τομής του κόμβου. Ωστόσο, μετά από d συνεχείς τομές, το μέγεθος του χωρίου που συσχετίζεται με ένα κόμβο είναι μικρότερο κατά ένα συγκεκριμένο λόγο. Επιπλέον, η τομή δεν εγγυάται τον ομοιόμορφο διαχωρισμό των σημείων. Αντιθέτως, η συρρίκνωση εγγυάται ότι τα σημεία θα κατανεμηθούν ομοιόμορφα στους

κόμβους ενώ ταυτόχρονα μειώνει το μέγεθος των χωρίων που συσχετίζονται με ένα κόμβο. Μάλιστα με την εναλλαγή τομών και συρρικνώσεων οι Arya et al. [4] ισχυρίζονται ότι τόσο το μέγεθος, όσο και το πλήθος των σημείων που συσχετίζονται με έναν κόμβο μειώνονται εκθετικά.

Μια διαφορετική στρατηγική είναι η συνεχής χρήση της τομής όσο η μείωση των σημείων ενός κόμβου είναι σημαντική. Όταν ο ρυθμός με τον οποίο ελαττώνονται τα σημεία μειωθεί σημαντικά, τότε εφαρμόζεται η συρρίκνωση. Οι Arya et al. [4] συμπέραναν ότι η συρρίκνωση χρειάζεται μόνο αποσπασματικά. Επισημαίνουν, όμως, πως στην περίπτωση που τα σημεία σχηματίζουν ομάδες υψηλής πυκνότητας, η χρήση των συρρικνώσεων είναι σημαντική ειδικά για τα ερωτήματα πλησιέστερου γείτονα και αναζήτησης περιοχής.

Ένας τρόπος για να υλοποιηθεί η τομή είναι ο αλγόριθμος Middle-interval. Προκειμένου να πραγματοποιηθεί η τομή σε ένα κουτί (έστω b_O), το πρώτο βήμα πρέπει να είναι ο εντοπισμός της μεγαλύτερης πλευράς του. Σε αυτή την πλευρά, η τομή μπορεί να γίνει σε οποιοδήποτε σημείο ανάμεσα στο 1/3 και στα 2/3 του μήκους της συγκεκριμένης πλευρά. Ωστόσο, αν υπάρχει εσωτερικό κουτί (έστω b_I), τότε η πολυπλοκότητα της τομής αυξάνεται, καθώς θα πρέπει να διατηρείται η εγγύτητα. Έστω η προβολή του εσωτερικού κουτιού b_I πάνω στην μεγαλύτερη πλευρά του b_O. Αν η προβολή του b_I ταυτίζεται με τη μεγαλύτερη πλευρά, τότε η εν λόγω διαδικασία μεταφέρεται στη δεύτερη μεγαλύτερη πλευρά του b_O και ελέγχεται εκ νέου η προβολή του b_I πάνω στη δεύτερη μεγαλύτερη πλευρά του b_O. Εφόσον εντοπιστεί η πλευρά κατά μήκος της οποίας θα πραγματοποιηθεί η τομή, η διαδικασία συνεχίζεται με την επιλογή του σημείου στο οποίο θα πραγματοποιηθεί η τομή. Αν η πλευρά του b_O κατά την οποία πραγματοποιείται η τομή περιγράφεται από το διάστημα $[x_O, y_O]$, η αντίστοιχη πλευρά του b_I

από το διάστημα $[x_I, y_I]$, το μήκος του b_O με w και το σημείο τομής συμβολίζεται με x_c, τότε η εύρεση του σημείου τομής περιγράφεται από τους παρακάτω κανόνες:

(α) Αν $x_I \geq x_O + (2/3)w$, τότε $x_c = x_O + (1/3)w$.

(β) Αν $y_I \leq x_O + (1/3)w$, τότε $x_c = x_O + (2/3)w$.

(γ) Αλλιώς $x_c = x_I$ αν $x_O + (1/3)w \leq x_I \leq x_O + (2/3)w$,

αλλιώς $x_c = y_I$.

Όσον αφορά τη συρρίκνωση, ο αλγόριθμος της «κεντρικής συρρίκνωσης» (central shrink) είναι πολύ πιο περίπλοκος από τον αλγόριθμο τομής. Ο σκοπός του αλγορίθμου συρρίκνωσης είναι να δημιουργηθούν κελιά με το πολύ 2/3 σημεία από τα σημεία του πατέρα. Διατηρώντας τους παραπάνω συμβολισμούς, η συρρίκνωση μπορεί να περιγραφεί ως εξής:

Έστω r το μικρότερο παραλληλόγραμμο το οποίο περικλείει τα σημεία που ανήκουν στο b_O καθώς και το b_I. Με βάση τον τρόπο που ένα BBD δέντρο κρατάει τα σημεία του, ένα κουτί b μπορεί να κατασκευαστεί σε $O(d)$ χρόνο, με τις ακόλουθες προϋποθέσεις (Arya and Mount [3]):

- η μεγαλύτερη πλευρά του b είναι το πολύ κατά ένα συγκεκριμένο λόγο μεγαλύτερη από τη μεγαλύτερη πλευρά του r,
- το εσωτερικό κουτί του b_O διατηρεί την εγγύτητα με το b και
- το b διατηρεί την εγγύτητα με το b_O.

Αν υποτεθεί ότι το μήκος του r είναι σχεδόν όσο το μήκος του b_O, τότε $b = b_O$. Ειδάλλως το b κατασκευάζεται με την εφαρμογή στο r μιας σειράς από μεγεθύνσεις. Επίσης, έστω ότι το b_O έχει ένα εσωτερικό κουτί b_I. Τότε αυτό, λόγω των παραπάνω προϋποθέσεων, περιβάλλεται από το r. Κάθε πλευρά του r μεγεθύνεται έτσι ώστε να περικλείει ένα 3^d πλέγμα αντίγραφων του b_I (να εξασφαλιστεί έτσι η εγγύτητα μεταξύ των r και b_I). Αυτή η διαδικασία μπορεί να μεγεθύνει το μήκος του r μέχρι 3 φόρες του αρχικού του μήκους. Στο επόμενο βήμα το r μεγεθύνεται έτσι ώστε να σχηματίσει έναν υπερκύβο. Έστω ότι l_{max} είναι το μήκος του r. Κάθε πλευρά του r μεγεθύνεται ώστε να γίνει ίση με l_{max}. Εφόσον το l_{max} είναι μικρότερο από κάθε πλευρά του b_O, τότε ο μετασχηματισμός του r σε υπερκύβο, δεν αλλάζει το γεγονός ότι $r \subseteq bo$. Οι Arya et al. απέδειξαν στο [4] ότι η κατασκευή του b σύμφωνα με τα παραπάνω διατηρεί τις προϋποθέσεις που τέθηκαν.

3.1.2 Απαρίθμηση των κελιών εντός μιας ακτίνας

Η απόσταση μεταξύ ενός σημείου και ενός κελιού ορίζεται ως η μικρότερη απόσταση του σημείου από οποιοδήποτε σημείο του κελιού. Για κάθε q, τα m πλησιέστερα κελιά μπορούν να απαριθμηθούν σε $O(md \log n)$ χρόνο. Αυτή η διαδικασία επιτυγχάνεται με τη χρήση μιας ουράς προτεραιότητας που περιέχει κόμβους του BBD δέντρου, με την προτεραιότητα ενός κόμβου να είναι αντιστρόφως ανάλογη της απόστασης του κόμβου από το σημείο ερώτησης q. Η απόσταση ενός κελιού από ένα σημείο μπορεί να υπολογιστεί σε χρόνο $O(d)$. Αρχικά εισάγεται η ρίζα του δέντρου στην ουρά. Στη συνέχεια ακολουθείται η παρακάτω διαδικασία αναδρομικά. Εξάγεται από την ουρά ο κόμβος με τη μεγαλύτερη προτεραιότητα (μικρότερη απόσταση από το σημείο q). Ακολούθως διασχίζεται το δέντρο μέχρι το φύλλο που βρίσκεται

πλησιέστερα στο σημείο q. Καθώς κάθε κελί αποτελεί τη διαφορά δύο d-διάστατων παραλληλόγραμμων, είναι δυνατόν να βρεθεί ποιο παιδί είναι πλησιέστερα στο σημείο q σε O(d) χρόνο. Καθώς διασχίζεται το δέντρο, υπολογίζεται η απόσταση που έχει ο αδερφός του εκάστοτε κόμβου που ο αλγόριθμος επισκέπτεται και εισάγει τον αδερφό αυτό στην ουρά προτεραιότητας. Κάθε κόμβος εισάγεται στην ουρά το πολύ μια φορά. Δεδομένου ότι υπάρχουν το πολύ n κόμβοι στην ουρά, ο κόμβος με τη μέγιστη προτεραιότητα μπορεί να βρεθεί σε O(log n) χρόνο. Αν χρησιμοποιηθεί ένας σωρός Fibonacci ως ουρά προτεραιότητας, ο αποσβησμένος χρόνος που απαιτείται για μια εισαγωγή είναι O(1). Δεδομένου ότι το BBD tree έχει ύψος O(log n) και η επεξεργασία ενός κόμβου απαιτεί O(d) χρόνο, το επόμενο φύλλο μπορεί να υπολογιστεί σε O(d log n) χρόνο. Είναι προφανές ότι τα m πλησιέστερα κελιά μπορούν να υπολογιστούν σε O(md log n) χρόνο.

3.1.3 Ερωτήματα προσεγγιστικής περιοχής

Ο αλγόριθμος για την απάντηση ενός ερωτήματος περιοχής με τη χρήση ενός BBD δέντρου στηρίζεται στη φιλοσοφία απάντησης ενός ερωτήματος περιοχής σε ένα διαμεριστικό δέντρο (partitioning tree). Ο αλγόριθμος ξεκινάει από τη ρίζα και εξετάζει αν ένας κόμβος βρίσκεται εντός ή εκτός του χωρίου. Εφόσον ο κόμβος δεν κείται πλήρως εντός ή εκτός του χωρίου, το ερώτημα μεταφέρεται στα παιδιά του συγκεκριμένου κόμβου. Έστω ένα χωρίο του d-διάστατου χώρου Q. Δεδομένης της προσεγγιστικής φύσης της αναζήτησης, ορίζεται το εσωτερικό διάστημα (inner range) Q^- και το εξωτερικό διάστημα (outer range) Q^+ ως η συρρίκνωση και η αύξηση του Q κατά ε αντίστοιχα [3].

Με βάση τα παραπάνω, τα σημεία που βρίσκονται εντός του Q μπορούν να βρεθούν με τον ακόλουθο τρόπο, με τη βοήθεια ενός BBD δέντρου. Ξεκινώντας από τη ρίζα, ελέγχεται αρχικά αν το κελί του εκάστοτε κόμβου είναι υποσύνολο του Q^+. Εφόσον ο έλεγχος αυτός ισχύει, τότε τα σημεία που ανήκουν στον κόμβο αυτό, ανήκουν στο χωρίο Q και μόνον αυτά. Ο αλγόριθμος τερματίζει επιστρέφοντας τα σημεία αυτά. Αν το κελί δεν τέμνεται με το Q^-, τότε δεν έχει κοινά σημεία με το Q και ο αλγόριθμος τερματίζει χωρίς να επιστρέψει σημεία. Στην περίπτωση που ο τρέχον κόμβος είναι φύλλο, κάθε σημείο του ελέγχεται ξεχωριστά αν ανήκει ή όχι στο χωρίο Q και ο αλγόριθμος επιστρέφει τα σημεία που ανήκουν στο χωρίο. Τέλος, αν ο τρέχον κόμβος δεν είναι φύλλο, τότε ο αλγόριθμος καλείται και για τα δύο παιδιά. Οι Arya και Mount [3] απέδειξαν ότι για ένα BBD δέντρο που περιέχει n σημεία του R^d, τότε τα σημεία που βρίσκονται εντός ενός χωρίου Q εντοπίζονται με ακρίβεια $\varepsilon > 0$ σε χρόνο $O(2^d \log n + (3\sqrt{d}/\varepsilon)^d)$.

Ο αλγόριθμος για την απάντηση ενός ερωτήματος περιοχής με ακρίβεια ε για ένα χωρίο Q και ένα κόμβο u μπορεί να περιγράφει από τον παρακάτω ψευδοκώδικα.

Αλγόριθμος 3.1 [Αναζήτηση Διαστήματος]

Δεδομένα: Ένα χωρίο Q, δοσμένο με ένα εσωτερικό χωρίο Q_I και ένα εξωτερικό χωρίο Q_O, και ένας κόμβος u στο BBD-δέντρο.

Επιστρέφει: Το συνολικό βάρος των σημείων στο προσεγγιστικό χωρίο.

Ερώτημα(Q, u):

***εάν κελί*$(u) \subseteq Q_O$ *τότε επέστρεψε το βάρος*(u);

***εάν κελί*$(u) \cap Q_I = \emptyset$ *τότε επέστρεψε* 0;

εάν u ***είναι φύλλο τότε***

```
w = 0;
για κάθε p ∈ σημεία(u) τότε
        αν p ∈ Q τότε w += το βάρος(p);
επέστρεψε w;
αλλιώς επέστρεψε Ερώτημα(Q, αριστερό παιδί(u))
  + Ερώτημα(Q, δεξί παιδί(u));
```

3.2 Η δυναμική δομή δεδομένων «quadtreap»

3.2.1 Ψευδο-κόμβοι και περιστροφή

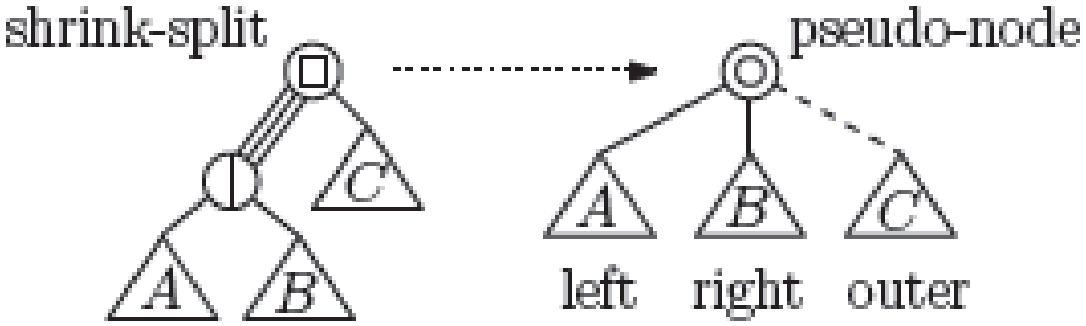
Το BD-δέντρο μπορεί να εξισορροπηθεί μέσω μιας απλής τοπικής λειτουργία, η οποία αναφέρεται πολλές φορές ως περιστροφή ή προαγωγή. Αυτή η λειτουργία είναι παρόμοια με τη γνωστή λειτουργία περιστροφής όπως ορίζεται στα δέντρα δυαδικής αναζήτησης, αλλά κάποιες τροποποιήσεις θα χρειαστούν στο πλαίσιο των BD-δέντρων. Αυτή η λειτουργία θα είναι καθοριστική στη διατήρηση της ισορροπίας στη δομή των δεδομένων μας καθώς σημεία θα εισάγονται και θα διαγράφονται.

Καθ' όλη αυτή την ενότητα υποθέτουμε ότι το BD-δέντρο ικανοποιεί την ιδιότητα ότι οι εσωτερικοί κόμβοι του BD-δέντρου μπορούν να χωριστούν σε ζεύγη, έτσι ώστε κάθε ζεύγος αποτελείται από έναν κόμβο συρρίκνωσης ως πατέρας και από έναν κόμβο διάσπασης ως το εσωτερικό του παιδί. Αυτή την ιδιότητα την ονομάζουμε συρρίκνωσης - διάσπασης. Όπως αναφέρεται στο [17], στο BD-δέντρο υπάρχουν αλγόριθμοι κατασκευής που ικανοποιούν αυτή την ιδιότητα.

Αυτή η υπόθεση μας δίνει τη δυνατότητα να δούμε κάθε συνδυασμό συρρίκνωσης - διάσπασης ως ενιαία πράξη αποσύνθεσης, η οποία

υποδιαιρεί ένα κελί σε τρία υπο-κελιά. Έτσι, μπορούμε εννοιολογικά να συγχωνεύσουμε κάθε ζευγάρι συρρίκνωσης - διάσπασης σε ένα μόνο κόμβο, που ονομάζεται ψευδο-κόμβος (βλ. Σχ. 3.1), ο οποίος έχει τρία παιδιά.

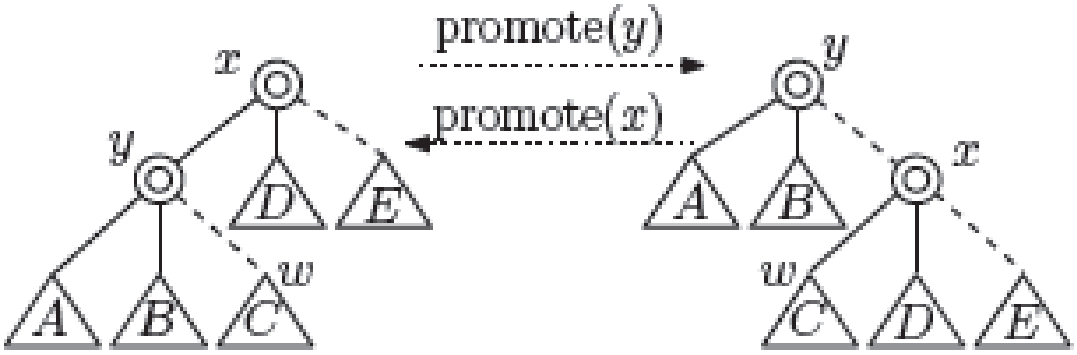
Τα πρώτα δύο παιδιά, ονομάζονται το αριστερό και το δεξί παιδί, αντιστοιχούν στα παιδιά του κόμβου διάσπασης, και το τρίτο παιδί, που ονομάζεται το εξωτερικό παιδί, αντιστοιχεί στο εξωτερικό παιδί του κόμβου συρρίκνωσης. Σημειώστε ότι αυτό είναι μια εννοιολογική συσκευή και δεν συνεπάγεται καμία δομική αλλαγή στο δέντρο.



Σχήμα 3.3: Ο ψευδο-κόμβος που αντιστοιχεί σε ζευγάρι συρρίκνωσης – διάσπασης

Ας ορίσουμε τώρα την περιστροφή των ψευδο-κόμβων. Καλούμε τη λειτουργία αυτή ως προαγωγή και ορίζεται δίνοντας έναν κόμβο y του δέντρου που είναι είτε ένα αριστερό παιδί ή ένα παιδί εξωτερικό. Έστω x είναι ο πατέρας του y. Αν το y είναι ένα αριστερό παιδί, η λειτουργία προαγωγή (y) κάνει το x το εξωτερικό παιδί του y, και το προηγουμένως εξωτερικό παιδί του y γίνεται το νέο αριστερό παιδί του x. Παρατηρούμε ότι η εσωτερική αριστερή σύμβαση διατηρείται, δεδομένου ότι το αριστερό παιδί του y είναι αμετάβλητο, και το αριστερό παιδί του x

(επισημασμένο στο σχήμα ως w) περιέχει το εσωτερικό κουτί (που αποτελείται από την ένωση των κελιών A και B στο σχήμα).



Σχήμα 3.4: Λειτουργία περιστροφής των ψευδο-κόμβων.

Στη συνέχεια, ας εξετάσουμε την προαγωγή ενός εξωτερικού παιδιού x. Έστω y ο πατέρας του. Η λειτουργία προαγωγή (x) είναι το αντίστροφο αυτού που περιγράφηκε προηγουμένως ως προαγωγή αριστερού παιδιού. Το y γίνεται το αριστερό παιδί του x, και το παλιά αριστερό παιδί του x γίνεται το νέο εξωτερικό παιδί του y. Και πάλι, η εσωτερική αριστερή σύμβαση διατηρείται, δεδομένου ότι πριν από την προαγωγή, εάν το κελί y είχε ένα εσωτερικό κουτί, θα πρέπει να βρίσκεται στο αριστερό παιδί του A, και έτσι μετά από την προαγωγή που βρίσκεται στο αριστερό παιδί του x, το y.

Με βάση τις ανωτέρω περιγραφές, είναι εύκολο να δούμε ότι οι βασικές ιδιότητες του BD-δέντρου διατηρούνται μετά την εφαρμογή οποιασδήποτε λειτουργίας προαγωγής. Σημειώστε ότι η προαγωγή δεν μεταβάλλει την υποκείμενη χωρική αποσύνθεση, μόνο τη δομή του δέντρου. Δεδομένου ότι πρόκειται για μια καθαρά τοπική λειτουργία, η προαγωγή μπορεί να πραγματοποιηθεί σε O(1) χρόνο.

3.2.2 Εισαγωγή σημείου

Αλγόριθμος 3.2: Αναδιάρθρωση δέντρου μετά την εισαγωγή σημείου σε έναν νέο κόμβο – φύλλο u

while u $\neq$ root(T) **do**

u $\leftarrow$ parent(u);

if u is a left or right child **then**

update(u);

if $u^{[2]} < u^{[2]}$ **then**

promote(u); update(u);

else u $\leftarrow$ u;

else if $u^{[2]} < u^{[2]}$ **then**

promote(u); update(u);

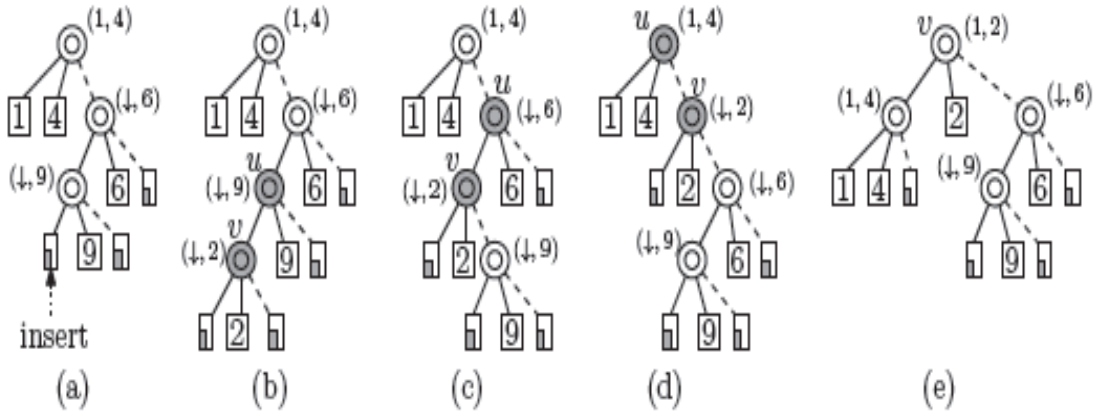
else break

Λήμμα 3.1: Μετά την εισαγωγή ενός σημείου και την εφαρμογή του Αλγορίθμου 3.1, η δομή quadtreap ικανοποιεί τις ακόλουθες ιδιότητες:

i) Εάν τα u και w είναι αριστερά και δεξιά παιδιά ενός κόμβου διάσπασης αντιστοίχως, τότε $u^{[1]} < w^{[1]}$

ii) Δοθέντος οποιουδήποτε ζευγαριού που αποτελείται από τον γονέα u και το παιδί v, ισχύει $u^{[2]} <= u^{[2]}$. Εάν ο u είναι κόμβος διάσπασης, η ανισότητα είναι ορθή.

Λήμμα 3.2: Ο χρόνος που απαιτείται για την εισαγωγή ενός σημείου σε μια δομή quadtreap μεγέθους n και ύψους h είναι O(h). Συνεπώς, ο χρόνος εισαγωγής είναι O(log n) με μεγάλη πιθανότητα.



Σχήμα 3.5: Παράδειγμα από την αναδιάρθρωση μιας δομής quadtreap μετά την εισαγωγή.

3.2.3 Διαγραφή σημείου

Αλγόριθμος 3.3: Αναδιάρθρωση του δέντρου ως μέρος της διαδικασίας διαγραφής ενός σημείου σε ένα κόμβο – φύλλο u.

label(u) ← (↑, ↑);

r ← parent(u);

while r ≠ null **do**

    u ← r; r ← parent(r);

    update(u);

w ← left(u); x ← outer(u);

while $w^{[2]} < u^{[2]}$ or $x^{[2]} < u^{[2]}$ **do**

if $w^{[2]} < x^{[2]}$ **then**

promote(w); update(u);

w ← left(u);

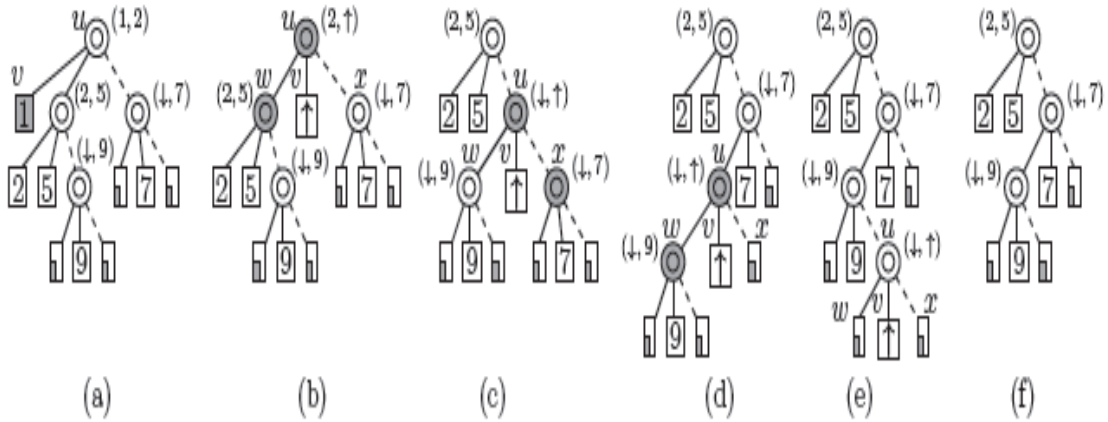
else

promote(x); update(x);

x ← outer(u);

Θεώρημα 3.1: Μετά την διαγραφή ενός σημείου και την εφαρμογή του Αλγόριθμου 3.2, η δομή quadtreap ικανοποιεί τις ιδιότητες του Λήμματος 3.1

Λήμμα 3.3: Ο αναμενόμενος χρόνος για τη διαγραφή ενός τυχαίου σημείου μιας δομής quadtreap του μεγέθους n και ύψους h είναι $O(h^2)$. Συνεπώς, ο αναμενόμενος χρόνος διαγραφής είναι $O(\log^2 n)$ με μεγάλη πιθανότητα.



Σχήμα 3.6: Ένα παράδειγμα διαγραφής στη δομή quadtreap

3.2.4 Ερωτήματα προσεγγιστικής περιοχής

Σε αυτό το σημείο θα παρουσιάσουμε ένα αλγόριθμο για να απαντά σε ερωτήματα προσεγγιστικής καταμέτρησης περιοχής χρησιμοποιώντας τη δομή quadtreap που χρησιμοποίησαν οι Mount και Park [17]

Αλγόριθμος 3.4: Προσεγγιστική καταμέτρηση περιοχής(u, Q)

$C \leftarrow$ cell(u);

if $C \cap Q^- = \emptyset$ **then** return 0;

else if $C \subseteq Q^+$ **then** return wgt(u);

else if u is a leaf **then**

    if u contains a point in Q **then** return wgt(p);

    else return 0;

else if u is a split node **then**

    return range(left(u), Q) + range(right(u), Q);

else "/"u is a shrink node"/"

    return range(inner(u), Q) + range(outer(u), Q);

Λήμμα 3.4: Δοθέντος ενός συνόλου σημείων με ληφθέντα βάρη και ένα επαυξημένο BD-δέντρο που αποθηκεύει αυτά τα σημεία (το οποίο δεν χρειάζεται να ικανοποιεί την ιδιότητα της συρρίκνωσης – διάσπασης), ο τροποποιημένος αλγόριθμος ερωτήματος επιστρέφει σωστά μία μέτρηση που περιλαμβάνει όλα τα σημεία που βρίσκονται εντός της εσωτερικής

περιοχής και αποκλείει όλα τα σημεία που βρίσκονται εκτός της εξωτερικής περιοχής.

Θεώρημα 3.2: Ας θεωρήσουμε ένα σύνολο σημείων στο R^d τα οποία έχουν αποθηκευτεί σε ένα επαυξημένο BD-δέντρο ύψους h. Τότε, για κάθε $\varepsilon > 0$ και κάθε κυρτή περιοχή Q που ικανοποιεί την υπόθεση μοναδιαίου κόστους:

(i) Εάν τα βάρη των σημείων προέρχονται από commutative group, είναι πιθανό να απαντηθούν τα ερωτήματα ε-προσεγγιστικής καταμέτρησης περιοχής σε χρόνο $O(h + (1/\varepsilon)^{d-1})$.

(ii) Εάν τα βάρη των σημείων προέρχονται από commutative semigroup, είναι πιθανό να απαντηθούν τα ερωτήματα ε-προσεγγιστικής καταμέτρησης περιοχής σε χρόνο $O(h \cdot (1/\varepsilon)^{d-1})$.

(iii) Είναι πιθανό να απαντήσουμε ερωτήματα ε-προσεγγιστικής καταμέτρησης περιοχής σε χρόνο $O(h + (1/\varepsilon)^{d-1} + k)$ όπου το k είναι ο αριθμός των σημείων που δηλώθηκαν.

Κεφάλαιο 4

ΕΡΩΤΗΜΑΤΑ ΠΛΗΣΙΕΣΤΕΡΟΥ ΓΕΙΤΟΝΑ ΑΝΑΜΕΣΑ ΣΕ ΠΑΡΑΛΛΗΛΑ ΤΜΗΜΑΤΑ

4.1 Εισαγωγή

4.2 Μέθοδοι για παράλληλα τμήματα

4.3 Βελτιώνοντας τον χρόνο χώρου και ερωτήματος

4.4 Ερωτήματα αναζήτησης ε-NN με περιορισμένο βάρος

ΚΕΦΑΛΑΙΟ 4^{ο}: Ερωτήματα πλησιέστερου γείτονα ανάμεσα σε παράλληλα τμήματα.

4.1 Εισαγωγή

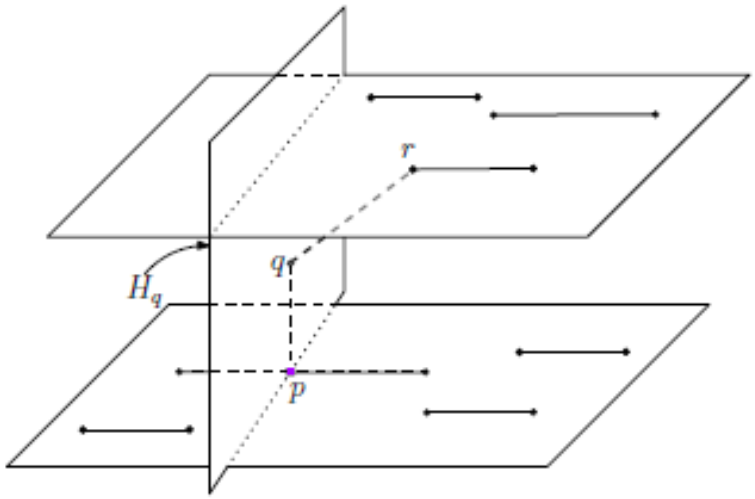
Έστω d(p, q) η ευκλείδεια απόσταση μεταξύ των σημείων p, q. Δοθέντος ενός συνόλου P σημείων στο R^d και έστω παράμετρος ε > 0, λέμε ότι ένα σημείο p του P είναι ένας ε-κατά προσέγγιση πλησιέστερος γείτονας (ε-NN), σε ένα σημείο q, αν d(p, q) ≤ (1 + ε) d(q΄, q), όπου q΄ είναι πιο κοντινό σημείο από το q στο P.

Μια ενδιαφέρουσα γενίκευση του προβλήματος προκύπτει εάν γίνει αντικατάσταση του σημείου του P με ένα σύνολο αντικειμένων O. Για αυτή την έκδοση υπάρχουν ελάχιστα γνωστά αποτελέσματα. Στο κεφάλαιο αυτό θα εξεταστεί η περίπτωση όπου O είναι ένα σύνολο παράλληλων τμημάτων [25]. Δίνουμε δύο λύσεις για το πρόβλημα. Η δεύτερη λύση που ενισχύει την πρώτη χρησιμοποιώντας αποτελέσματα της. Τα αποτελέσματα προέρχονται από το πρόβλημα του φθηνού βενζινάδικου. Δεδομένου n βενζινάδικων (τοποθεσίες) και ένα αυτοκίνητο σε μια θέση q (σημείο ερωτήματος) θέλουμε να βρούμε ένα βενζινάδικο που είναι πιο κοντά ή περίπου πλησιέστερα προς το q (αφού η ακριβής απόσταση δεν είναι τόσο σημαντική), το οποίο πωλεί βενζίνη για το πολύ w ευρώ. Παρακάτω συνδυάζουμε τα αποτελέσματα και τα παρουσιάζουμε με μια δομή δεδομένων που απαντά χρονικά εξαρτώμενα ε-NN ερωτήματα σε οποιαδήποτε σταθερή διάσταση.

4.2 Μέθοδοι για παράλληλα τμήματα

Έστω S είναι ένα σύνολο από n παράλληλα ασυνεχή τμήματα στην R^3. Υποθέτουμε ότι όλα τα τμήματα του S είναι παράλληλα προς το x-άξονα.

Έστω P το σύνολο των 2n άκρων των τμημάτων στο S. Έστω q ένα ερώτημα στο R^3, έστω s είναι ένα τμήμα του S πλησιέστερο προς το q, και p είναι το σημείο του s που είναι πλησιέστερου προς q. Παρατηρούμε ότι το p είναι ένα από τα άκρα του s ή ότι το qp τμήμα είναι κάθετο προς το s. Ας υποθέσουμε ότι το H_q είναι το επίπεδο που διέρχεται από το q που είναι παράλληλο προς το επίπεδο yz. Σημειώστε ότι αν p είναι εσωτερικό στο s τότε το p είναι ένα από τα σημεία $H_q \cap S$.



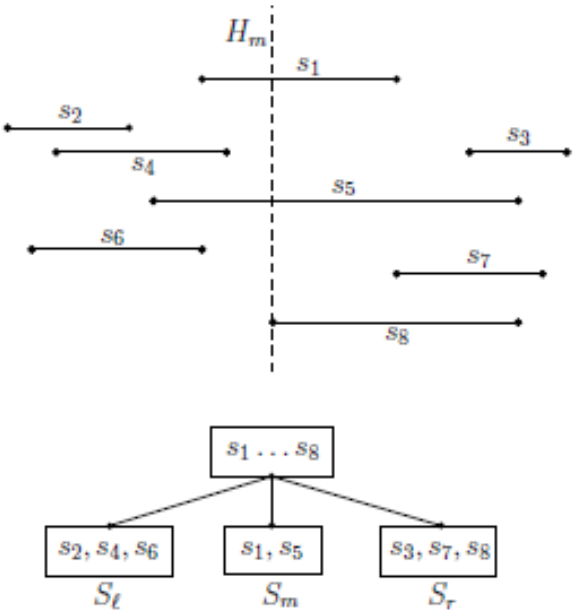
Σχήμα 4.1: Παράλληλα τμήματα που βρίσκονται σε δύο παράλληλα επίπεδα στον R^3 και ένα ερώτημα σημείου q. Το πλησιέστερο άκρο στο q είναι το r, αλλά το πλησιέστερο σημείο στο q είναι το p.

Επομένως, για να βρεθεί ένας ε-NN του q στο S αρκεί **(α)** να βρεθεί ένας ε-NN του q στο P, **(β)** να βρεθεί ένας ε-NN του q στο σύνολο $H_q \cap S$ και στη συνέχεια να αναφερθεί ποιο από τα δύο είναι πλησιέστερο στο q. Για την επίλυση του **(α)** εφαρμόζουμε ένα (t, ε) προσεγγιστικό Voronoi διάγραμμα (AVD) στο σύνολο P με t = O(1 / ε) μαζί με τη συναφή δομή δεδομένων. Αυτή η δομή έχει O(n) χώρο και επιστρέφει έναν ε-NN σε κάθε q σε O(log n + 1 / ε) χρόνο. Για την επίλυση του **(β)** παρουσιάζουμε

δύο μεθόδους οι οποίες και οι δύο βασίζονται σε μια παραλλαγή του γνωστού δέντρου διαστημάτων [6]. Η δεύτερη μέθοδος είναι πιο περίπλοκη, αλλά οδηγεί σε μια σημαντική βελτίωση σε σχέση με την πρώτη.

4.2.1 Πρώτη μέθοδος

Η κατασκευή του δέντρου διαστημάτων T για το S συμβαίνει ως εξής. Έστω x_m είναι το διάμεσο μεταξύ όλων των x-συντεταγμένων του P. Ας υποθέσουμε ότι H_m είναι το επίπεδο που διέρχεται από το σημείο $(x_m,0,0)$ και το οποίο είναι παράλληλο προς το επίπεδο yz. Αποθηκεύουμε το x_m στη ρίζα του T. Διαμερίζουμε το σύνολο των τμημάτων S σε τρία σύνολα S_l, S_m και S_r όπου κάθε ομάδα αποτελείται από τα τμήματα που βρίσκονται στα αριστερά του H_m, που τέμνονται με το H_m, και που βρίσκονται στα δεξιά του H_m, αντίστοιχα. Συνεχίζουμε την κατασκευή του δέντρου T αναδρομικά για τις S_l και S_r. (Αν S_l ή S_r είναι το κενό σύνολο σαφώς παίρνουμε ένα φύλλο). Οι δύο ρίζες των δένδρων που βασίζονται στην S_l και S_r γίνονται το αριστερό και το δεξί παιδί της ρίζας του T, αντίστοιχα. Για το σύνολο S_m δομούμε μια

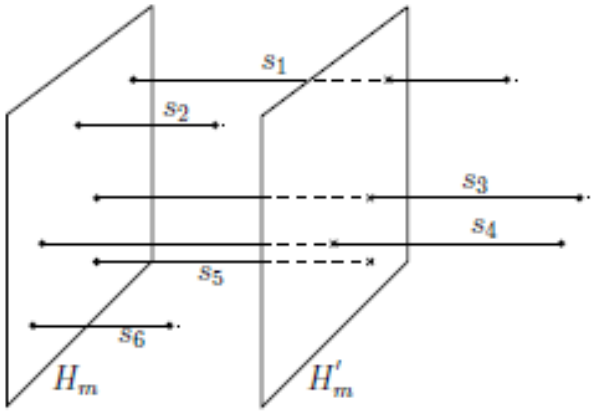


Σχήμα 4.2: Μια προβολή ενός συνόλου τμημάτων στο επίπεδο xz. Το αριστερό τελικό σημείο της s_8 είναι το διάμεσο x_m. Η H_m διαιρεί τα τμήματα σε τρία σύνολα S_l, S_m και S_r. Παρακάτω το δέντρο που αντιστοιχεί σε αυτή την κατάτμηση των τμημάτων.

βοηθητική δομή δεδομένων T_m την οποία συνδέουμε με τη ρίζα του T. (Φτιάχνουμε δομές παρόμοιες με την T_m για όλους τους εσωτερικούς τους κόμβους του T). Η T_m δομείται ως εξής. Η H_m περικόπτει φυσικά κάθε τμήμα στην S_m σε δυο κομμάτια. Έστω C_l είναι τα αριστερά κομμάτια των τμημάτων και C_r τα δεξιά κομμάτια. Χτίζουμε ένα δέντρο για το C_l και ένα για το C_r. Περιγράφουμε την κατασκευή του δέντρου μόνο για το C_r, δεδομένου ότι είναι συμμετρικά για το C_l.

Ας υποθέσουμε ότι x'_m είναι το διάμεσο μεταξύ όλων των x-συντεταγμένων των δεξιών άκρων των τμημάτων στο C_r. (Σημειώστε ότι οι x-συντεταγμένες των αριστερών άκρων είναι όλες ίσες). Ας υποθέσουμε ότι H'_m είναι το επίπεδο που διέρχεται από το σημείο $(x'_m,0,0)$ και το οποίο είναι παράλληλο προς το επίπεδο yz. Αποθηκεύουμε το x'_m στη ρίζα. Τα τμήματα του C_r που δεν είχαν κοπεί από το H'_m αποτελούν το σύνολο S'_l. Τα δεξιά κομμάτια από τα τμήματα κοπής H'_m αποτελούν το S'_r. Τα αριστερά κομμάτια (τα οποία καλύπτουν τα επίπεδα H_m και H'_m) αποτελούν το σύνολο S'_m. Για τα S'_l, S'_r συνεχίζουμε με την κατασκευή αναδρομικά (εκτός εάν είναι κενά), όπως χτίσαμε το δέντρο διαστημάτων T παραπάνω. Χρησιμοποιούμε το σύνολο των τμημάτων S'_m για την κατασκευή ενός 2D Voronoi διαγράμματος για το σύνολο σημείων $H'_m \cap S'_m$. Τότε αυτό συνδυάζεται με ένα πρότυπο αλγόριθμο θέσης σημείου για να μας δώσει μια δομή δεδομένων T_2 που απαντά με τον βέλτιστο τρόπο στα 2D ερωτήματα

πλησιέστερου γείτονα πάνω στο σύνολο $H'_m \cap S'_m$. Το T_2 συνδέεται με τη ρίζα του δέντρου στο C_r. Παρόμοιες δομές με την T_2 κατασκευάζονται για όλους τους εσωτερικούς κόμβους του T_m.



Σχήμα 4.3: Το αριστερό τελικό σημείο του s_5 είναι η μέση x'_m. Τα τμήματα των τμημάτων s_1, s_2 και s_3 που είναι στην αριστερή πλευρά της H'_m αποτελούν το σύνολο S'_r. Τα τμήματα μεταξύ H_m και H'_m, συνοδευόμενα από το s_5 σχηματίζουν το S'_m. Το S'_l σχηματίζεται από τα s_2 και s_6.

Υπολογίζουμε ένα όριο στο μέγεθος του T_m, δηλαδή, για τα επαυξημένα δέντρα στο C_l και στο C_r. Έστω $n' = |S_m|$. Επειδή χωρίζουμε πάντα στη διάμεσο, το ύψος του T_m είναι $O(\log n')$. Αυτό σημαίνει ότι ένα τμήμα του S_m μπορεί να κοπεί το πολύ $0(\log n')$ φορές και έτσι το συνολικό μέγεθος της T_m είναι $O(n' \log n')$. Χρησιμοποιώντας γνωστά αποτελέσματα, είναι εύκολο να δούμε ότι το συνολικό μέγεθος όλων των δομών T_2 που συνδέονται με τους κόμβους του T_m είναι επίσης $O(n' \log n')$.

Υπολογίζουμε ένα όριο στο μέγεθος της κύριας δομής δεδομένων T. Αφού το σύνολο των τμημάτων S_m που χρησιμοποιούνται σε κάθε κόμβο

του T για τη βοηθητική δομή δεδομένων T_m είναι ασύνδετα και χρησιμοποιώντας το όριο χώρου στο T_m προκύπτει εύκολα ότι το συνολικό μέγεθος του T είναι O(n log n). Λόγω των ισορροπημένων διασπάσεων, το T έχει ύψος O(log n).

Θα περιγράψουμε τώρα πώς να λύσουμε το **(β)** το οποίο είναι, δεδομένου ενός ερωτήματος $q = (q_x, q_y, q_z)$, πώς να βρούμε έναν ε-NN στο q στο σύνολο $H_q \cap S$. (Στην πραγματικότητα, αυτή η **πρώτη μέθοδος** βρίσκει τον ακριβή πλησιέστερο γείτονα στο q). Ξεκινάμε από τη ρίζα του T και σε κάθε κόμβο u ακολουθούμε το παιδί ανάλογα με το αποτέλεσμα της σύγκρισης μεταξύ q_x και x_m, και η τιμή αποθηκεύεται στο u. Σε κάθε κόμβο u μπορούμε επίσης να επισκεφθούμε τη βοηθητική δομή δεδομένων T_m. Ακολουθούμε ομοίως το μονοπάτι από τη ρίζα του T_m στο φύλλο που περιέχει το q και σε κάθε κόμβο χρησιμοποιούμε το T_2 για να βρούμε το πλησιέστερο γείτονα στο (q_y, q_z) μεταξύ του. Αναφέρουμε ως απάντηση το πλησιέστερο σημείο στο q πάνω από όλα τα σημεία που επιστράφηκαν από όλα τα ερωτήματα στο T_2.

Αφού το βάθος του T δέντρου είναι O(log n) και για κάθε κόμβο του T επισκεπτόμαστε μια βοηθητική δομή δεδομένων T_m με βάθος το πολύ O(log n) και σε κάθε κόμβο του T_m η δομή δεδομένων T_2 μπορεί να έχει κατασκευαστεί για το πολύ n χώρους, συνάγεται ότι ο χρόνος του ερωτήματος είναι $O(\log^3 n)$. Ο συνολικός χρόνος ερωτήματος είναι το άθροισμα των χρόνων που χρησιμοποιούνται για την επίλυση (α) και (β) και έτσι παίρνουμε το παρακάτω αποτέλεσμα:

Θεώρημα 4.1: Δεδομένης n παράλληλων τμημάτων σε 3D μπορούμε να κατασκευάσουμε μια δομή δεδομένων με O(n log n) χώρο για την εύρεση

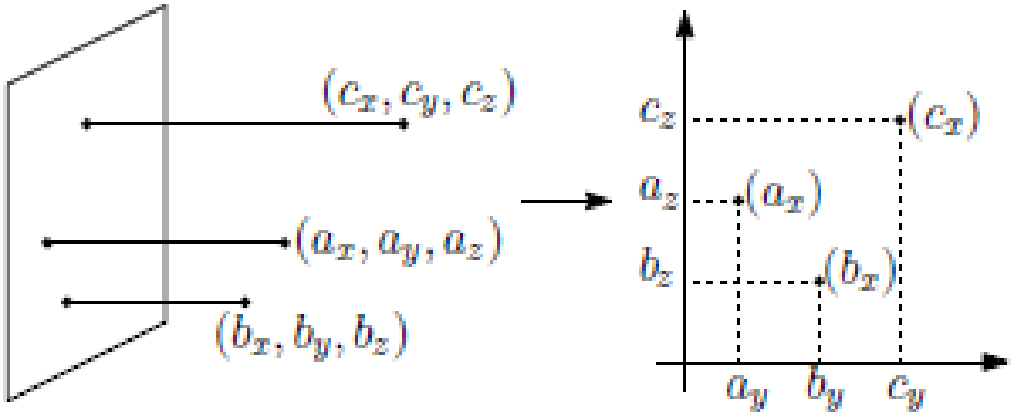
ενός ε-NN σε οποιοδήποτε δοσμένο σημείο ερωτήματος q σε $O(\log^3 n + 1/\varepsilon)$ χρόνο.

Θα συζητήσουμε τους χρόνους κατασκευής των δομών δεδομένων στην πλήρη έκδοση, όμως, όλοι τους είναι σε O(n poly(log n, 1/ε)).

4.3 Βελτιώνοντας τον χρόνο χώρου και ερωτήματος

Παρουσιάζουμε παρακάτω μια **δεύτερη μέθοδο** που μειώνει τον χρόνο τόσο του χώρου όσο και του χρόνου με συντελεστή log n. Το μέρος της πρώτης μεθόδου που αλλάζουμε είναι η βοηθητική δομή δεδομένων T_m για τα σύνολα C_l και C_r. Περιγράφουμε τη δομή δεδομένων T_r για τα τμήματα της C_r και μια ανάλογη δομή δεδομένων μπορεί να κατασκευαστεί για το C_l.

Σύμφωνα με το **(β)**, ο στόχος μας για το C_r είναι να βρεθεί ένας ε-NN σε q στο $H_q \cap C_r$. Λύνουμε το πρόβλημα αυτό, μειώνοντας το πρόβλημα μας στην αναζήτηση του κατά-προσέγγιση με περιορισμένο βάρος σε δύο διαστάσεις. Συγκεκριμένα κάθε δεξιό τελικό σημείο (p_x, p_y, p_z) ενός τμήματος στο C_r αντιστοιχίζεται στο σημείο (p_y, p_z) με βάρος p_x. Συμβολίζουμε με το P΄ το 2D σταθμισμένο σημείο που λαμβάνουμε. Ας υποθέσουμε ότι P'_w είναι το υποσύνολο του P΄ που περιέχει μόνο τα σημεία με βάρος τουλάχιστον w. Δοθέντος ενός ερωτήματος $q = (q_x, q_y, q_z)$, ο στόχος μας είναι να βρεθεί ένας ε-NN στο σημείο $q_2 = (q_y, q_z)$ στο P'_{q_x}. Σημειώστε ότι αυτό αρκεί για να επιτύχουμε τον πρώτο στόχο μας.



Σχήμα 4.4: Προβολή σε 2D. Ο x-συντεταγμένη κάθε σημείου χρησιμοποιείται ως βάρος (μέσα σε παρένθεση).

Εφαρμόζουμε Θεώρημα 4.3 της επόμενης ενότητας (ερωτήματα ε-NN με περιορισμένο βάρος) στο σύνολο σημείων P΄ για d = 2, γ = 2 και q = q_2 και να πάρουμε αυτό το λήμμα:

Λήμμα 4.1 Δεδομένων q και C_r μπορούμε να οικοδομήσουμε μια δομή δεδομένων T_r από O(|C_r| log(1/ε)) μέγεθος για να βρούμε έναν ε-NN σε q στο $H_q \cap C_r$σε O(log |C_r| + $(1/\varepsilon)^2$) χρόνο.

Χρησιμοποιώντας το παραπάνω λήμμα παίρνουμε το ακόλουθο αποτέλεσμα: (Θα παραλείψουμε την ανάλυση η οποία είναι παρόμοια με εκείνη της πρώτης μεθόδου).

Θεώρημα 4.2: Δεδομένων n παράλληλων τμημάτων σε 3D μπορούμε να κατασκευάσουμε μια δομή δεδομένων σε O(n log(1/ε)) χώρο για την εύρεση ενός ε-NN σε οποιοδήποτε σημείο ερωτήματος q σε O($\log^2$ n + log n/ε^2) χρόνο.

4.4 Ερωτήματα αναζήτησης ε-NN με περιορισμένο βάρος.

Δοθέντος ενός συνόλου από σταθμισμένα d-διαστάσεων σημεία, ορίζουμε το πρόβλημα του ε-κατά προσέγγιση πλησιέστερου γείτονα με περιορισμένο βάρος: δίνεται ένα ερώτημα q και το βάρος w, να βρεθεί ένας ε-NN στο q μεταξύ των σημείων στο P, που έχουν βάρος τουλάχιστον w. Εδώ το w είναι ένας αριθμός που καθορίζεται από το χρόνο ερωτήματος. Σημειώστε ότι έχουμε επιτρέψει ένα κατά προσέγγιση λάθος σε μία παράμετρο (την απόσταση), αλλά χρειαζόμαστε ακρίβεια σε μια άλλη (το βάρος). Μπορούμε επίσης, να ορίσουμε το συμμετρικό πρόβλημα όπου αναζητούμε έναν ε-NN μεταξύ των σημείων του P με βάρος το πολύ w.

Για το υπόλοιπο αυτής της ενότητας θεωρούμε μόνο τη μέγιστη έκδοση του προβλήματος. Παραθέτουμε παρακάτω το αποτέλεσμα μας. Αυτό το αποτέλεσμα επίσης παρέχει ένα χωροχρονικό σκαλοπάτι το οποίο ελέγχεται από την παράμετρο γ.

Θεώρημα 4.3: Έστω P είναι ένα σύνολο n σταθμισμένων σημείων στο R^d, και εάν $0 < \varepsilon < 1/2$ και $2 \leq \gamma \leq 1/\varepsilon$ είναι δύο πραγματικές παράμετροι. Μπορούμε να κατασκευάσουμε μια δομή δεδομένων με $O(n\gamma^d \log(1/\varepsilon))$ διάστημα το οποίο μας επιτρέπει να απαντήσουμε ένα ερώτημα ε-NN με περιορισμένο βάρος σε χρόνο $O(\log(\gamma n) + 1/(\varepsilon\gamma)^d)$.

Για λόγους χώρου δίνουμε εδώ μόνο μερικές από τις βασικές ιδέες και μεθόδους τις οποίες έχουμε χρησιμοποιήσει για να αποδείξουμε το θεώρημα. Αρχίζουμε με μερικούς ορισμούς. Έστω b(q, r) είναι μια σφαίρα με ακτίνα r με κέντρο το σημείο q. Ας ορίσουμε $b^+(q, r)$ ότι είναι μια σφαίρα με ακτίνα $(1 + \varepsilon)r$ με κέντρο το σημείο q. Ας ορίσουμε ότι αποτελεί το σύνολο $\bar{b}(q, r)$ των σημείων που δεν περιλαμβάνονται στον b(q, r).

Δεδομένων P, q και r, ένα ερώτημα ε-κατά προσέγγιση σφαιρικής μέγιστης περιοχής επιστρέφει ένα σημείο στο P μαζί με το βάρος του, που βρίσκεται στο $b^{+}(q, r)$ και έχει βάρος τουλάχιστον τόσο μεγάλο όσο το μέγιστο βάρος μεταξύ όλων των σημείων b(q, r). Υπάρχει μια σειρά δομών δεδομένων για την αποτελεσματική απάντηση ερωτημάτων μέγιστης ε-περιοχής. Εδώ θα χρησιμοποιήσουμε τη δομή δεδομένων που περιγράφεται στο [1]. Για $2 \leq \gamma \leq 1/\varepsilon$ χρησιμοποιεί $O(n\gamma^{d} \log(1/\varepsilon))$ και έχει χρόνο ερωτήματος $O(\log(\gamma n) + 1/(\varepsilon\gamma)^{d-1})$. Η εξάρτηση από το ε του χρόνου του ερωτήματος μπορεί να βελτιωθεί περαιτέρω με τη βοήθεια γνωστών αποτελεσμάτων για τα ερωτήματα αναζήτησης προσεγγιστικής idempotent περιοχής. Αυτό συνεπάγεται μια παρόμοια βελτίωση σε σχέση με το ερώτημα του χρόνου του θεωρήματος, ωστόσο δεν θα το αναλύσουμε περαιτέρω στην παρούσα εργασία.

Η βασική ιδέα είναι να παρατηρήσουμε ότι ένα ερώτημα ε-NN με περιορισμένο βάρος μπορεί να απαντηθεί με τη βοήθεια μιας προσεκτικά επιλεγμένης σειρά από ερωτήματα μέγιστης ε-περιοχής. Ας υποθέσουμε ότι υπάρχει μια μέθοδος, π.χ. η δομή δεδομένων στο [1], που μπορεί να απαντήσει σε μέγιστης ένα ερώτημα μέγιστης ε-περιοχής σε t(n, ε) χρόνο. Δοθέντος ενός σημείου ερωτήματος q και ενός βάρους w, ψάχνουμε για έναν ε-NN με περιορισμένο βάρος στο q. Ας υποθέσουμε ότι εκτελούμε ένα ερώτημα μέγιστης ε-περιοχής στην περιοχή b(q, r) για κάποιο r της επιλογής μας και ότι αυτή επιστρέφει ένα σημείο που βρίσκεται σε απόσταση r' από q με το βάρος w΄. Σημειώνουμε ότι $r' \leq (1 + \varepsilon)r$. Εάν $w' \geq w$ αυτό συνεπάγεται, ότι μπορούμε να περιορίσουμε την αναζήτησή μας για έναν ε-NN με περιορισμένο βάρος στο q μεταξύ των σημείων στο b(q, r΄). Διαφορετικά, εάν $w' < w$ είμαστε σίγουροι ότι η απάντηση μπορεί να βρεθεί μόνο ανάμεσα στα σημεία $\bar{b}(q, r)$ (αφού γνωρίζουμε από την απάντηση στο ερώτημα μέγιστης απόστασης ε-NN ότι όλα τα

σημεία στο b(q, r) έχουν βάρος το πολύ w΄). Όταν όλα τα σημεία P, έχουν βάρος λιγότερο από το w δεν υπάρχει ε-NN με περιορισμένο βάρος. Για να αποφύγουμε αυτήν την περίπτωση μπορούμε να υποθέσουμε ότι υπάρχει ένα βοηθητικό στοιχείο αρκετά μακριά από την P με άπειρο βάρος.

Καταλαβαίνουμε ότι αφού επαναλάβαμε αρκετές φορές ερωτήματα μέγιστης απόστασης ε-NN με το ίδιο κέντρο q αλλά για διαφορετικές τιμές της ακτίνας r θα έχουμε περιορίσει την έρευνα μας σε μια σπείρα $A = \bar{b}(q, r_1) \cap b(q, r_2)$ για ορισμένες τιμές r_1, r_2 με $r_1 < r_2$ και θα έχουμε βρει ένα σημείο p στο A που ικανοποιεί τον περιορισμό του βάρους. Παρατηρήστε ότι αν $r_2 \leq (1 + \varepsilon)r_1$ τότε σαφώς το p είναι μια έγκυρη απάντηση στο ζεύγος ερωτήματος q και w. Θα δείξουμε στη συνέχεια ότι με μια προσεκτική επιλογή των ακτινών φάσματος μπορούμε εύκολα να αποκτούμε μια αποτελεσματική λύση για το πρόβλημά μας όταν το P έχει μικρή εξάπλωση. Ας υποθέσουμε ότι η εξάπλωση Δ ενός συνόλου σημείων P είναι η αναλογία μεταξύ της διαμέτρου και της απόστασης ενός πλησιέστερου ζεύγους P. Είναι σαφές ότι μπορούμε να εκτελέσουμε μια δυαδική αναζήτηση για την δεξιά ακτίνα χρησιμοποιώντας το πολύ O(log Δ) ερωτήματα μέγιστης ε-απόστασης και, συνεπώς, να βρούμε έναν ε-NN με περιορισμένο βάρος σε O(t(n, ε) log Δ) χρόνο.

Το μειονέκτημα της προσέγγισης που περιγράφεται παραπάνω είναι ότι εξαρτάται από την εξάπλωση και κοστίζει τουλάχιστον O(log n) συντελεστή, για κάθε ερώτημα μέγιστης ε-απόστασης ακόμη κι αν όλα αυτά τα ερωτήματα μοιράζονται το ίδιο κέντρο και μπορεί να έχουν επίσης παρόμοιες ακτίνες. Είναι ενδιαφέρον ότι αν και χρησιμοποιώντας μαζί με τις μεθόδους της προσέγγισης αυτής και τις μεθόδους που παρουσιάζονται στο [1] για την απάντηση σε ερωτήματα προσέγγισης ε-

περιοχής και ιδιαίτερα για την απάντηση σε ερωτήματα κ-στου πλησιέστερου γείτονα με ε-κατά προσέγγιση μπορούμε να καταργήσουμε αυτό το μειονέκτημα και να φθάσουμε στο θεώρημα.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Το πρόβλημα καταμέτρησης περιοχής είναι ένα βασικό πρόβλημα στην υπολογιστική γεωμετρία με πολλές σημαντικές εφαρμογές. Έχουν γίνει πολλές προσπάθειες στην παγκόσμια έρευνα για να λυθεί το πρόβλημα, αλλά η θεωρία των κάτω ορίων δείχνει ότι αν υποθέσουμε ότι ο χώρος είναι γραμμικός, το πρόβλημα δεν μπορεί να λυθεί σε πολυλογαριθμικό χρόνο, εκτός της περίπτωσης των ορθογώνιων περιοχών. Σε αυτή την εργασία παρατηρείται ότι αν επιτραπούν οι προσεγγιστικές περιοχές, τότε έχουμε πολύ καλύτερα αποτελέσματα.

Για την καλύτερη παρατήρηση του προβλήματος της εργασίας παρουσιάστηκαν συγκεκριμένες δεντρικές δομές δεδομένων. Τα Range Trees μολονότι εμφανίζουν καλό άνω φράγμα στο χρόνο απάντησης μίας αναζήτησης περιοχής, ωστόσο εμφανίζουν μεγάλο κόστος σε χρόνο προεπεξεργασίας και χώρο.

Τα k-d Trees είναι από τα πλέον διαδεδομένα δέντρα. Το υπολογιστικό κόστος, ο χώρος αποθήκευσης και ο χρόνος απάντησης μίας αναζήτησης περιοχής για ένα k-d Tree είναι: $P(n, d) = O(n \log n)$, $S(n, d) = O(nd)$, $Q(n, d) \leq O(n^{1-1/d} + s)$.

Ακόμη, παρουσιάστηκε μια δομή δεδομένων BBD που απαντά σε ερωτήματα προσεγγιστικής περιοχής, όπου το σφάλμα που επιτρέπεται από τον αλγόριθμο είναι συνάρτηση της διαμέτρου της αντίστοιχης περιοχής. Παρατηρήθηκε ότι σε οποιαδήποτε σταθερή διάσταση d, ένα σύνολο n σημείων στο R^d μπορεί να προεπεξεργαστεί σε $O(\log n)$ χρόνο και $O(n)$ χώρο, έτσι ώστε τα ερωτήματα προσεγγιστικής περιοχής να μπορούν να απαντηθούν σε $O(\log n + 1/\varepsilon^d)$ χρόνο, και για κυρτές περιοχές ο χρόνος λειτουργίας είναι $O(\log n + 1/\varepsilon^{d-1})$. Παρουσιάστηκε ακόμη, ένα

κάτω όριο $\Omega(\log n + (1/\varepsilon)^{d-1})$ για αναζήτηση προσεγγιστικής περιοχής. Αυτό σημαίνει ότι ο αλγόριθμος είναι ασυμπτωτικά βέλτιστος για κυρτές περιοχές (υποθέτοντας ότι έχουμε σταθερές διαστάσεις). Ο αλγόριθμος είναι αρκετά πρακτικός και έχει εφαρμογή.

Η quadtreap, δυναμική δομή δεδομένων για αποθήκευση πολυδιάστατων συνόλων σημείων, παρέχει σημαντικές ιδιότητες: Για κάθε διάσταση d μπορεί να αποθηκευτεί ένα σύνολο n σημείων σε O(n) χώρο. Το ύψος h του δέντρου είναι O(log n) με μεγάλη πιθανότητα. Υποστηρίζει εισαγωγή σημείων σε χρόνο O(h) και διαγραφή σημείων στην χειρότερη περίπτωση σε χρόνο $O(h^2)$. Μπορεί να απαντήσει σε ε-προσεγγιστικά ερωτήματα σφαιρικής περιοχής και ερωτήματα προσεγγιστικού πλησιέστερου γείτονα σε χρόνο $O(h+(1/\varepsilon)^{d-1})$.

Εξετάστηκε ακόμη, μια δομή δεδομένων για την αποτελεσματική απάντηση σε ερωτήματα πλησιέστερου γείτονα σε ένα σύνολο παράλληλων τμημάτων στις τρεις διαστάσεις. Συνδέθηκε το πρόβλημα αυτό με το πρόβλημα αναζήτησης του κατά προσέγγιση πλησιέστερου γείτονα με περιορισμούς βάρους και συνάχθηκαν τα παρακάτω συμπεράσματα: **Α)** Δεδομένων n παράλληλων τμημάτων σε 3D μπορούμε να κατασκευάσουμε μια δομή δεδομένων με O(n log n) χώρο για την εύρεση ενός ε-NN σε οποιοδήποτε δοσμένο σημείο ερωτήματος q σε $O(\log^3 n + 1/\varepsilon)$ χρόνο. **Β)** Δεδομένων n παράλληλων τμημάτων σε 3D μπορούμε να κατασκευάσουμε μια δομή δεδομένων σε O(n log(1/ε)) χώρο για την εύρεση ενός ε-NN σε οποιοδήποτε σημείο ερωτήματος q σε $O(\log^2 n + \log n/\varepsilon^2)$ χρόνο. **Γ)** Μπορούμε να κατασκευάσουμε μια δομή δεδομένων με $O(n\gamma^d \log(1/\varepsilon))$ διάστημα το οποίο μας επιτρέπει να απαντήσουμε ένα ερώτημα ε-NN με περιορισμένο βάρος σε χρόνο $O(\log(\gamma n) + 1/(\varepsilon\gamma)^d)$.

Προκύπτει ένας αριθμός ενδιαφερόντων προβλημάτων για συζήτηση. Ένα σημαντικό πρόβλημα είναι η σχετικά μεγάλη εκθετική εξάρτηση από την διάσταση. Μπορούν αυτοί οι εκθετικοί παράγοντες να ελαχιστοποιηθούν; Μια ακόμη ενδιαφέρουσα παρατήρηση είναι το γεγονός ότι το παρατηρούμενο μέσο σφάλμα του αλγορίθμου τείνει να είναι πολύ μικρότερο από τον επιτρεπόμενο παράγοντα σφάλματος ε.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] S. Arya, T. Malamatos, and D. M. Mount. Space-time tradeoffs for approximate spherical range counting. In Proc. 16th Annu. ACM-SIAM Sympos. Discrete Algorithms, pages 535-544, 2005.

[2] S. Arya, D. M. Mount. Algorithms for fast vector quantization. In Proc. of DCC '93: Data Compression Conference, eds. J. A. Storer and M. Cohn, IEEE Press, pages 381-390, 1993.

[3] S. Arya and D. M. Mount. “Approximate range searching”, Comput. Geom.: Theory and Appl., vol. 17, 135–163, 2000.

[4] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Wu. An optimal algorithm for approximate nearest neighbor searching. Journal of the ACM, 45:891-923, 1998.

[5] J. L. Bentley. K-d trees for semidynamic point sets. In Proc. 6th Ann. ACM Sympos. Comput. Geom., pages 187-197, 1990.

[6] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars. Computational Geometry: Algorithms and Applications. Springer-Verlag, Berlin, Germany, 3rd edition, 2008.

[7] M. Bern. Approximate closest-point queries in high dimensions. Inform. Process. Lett., 45:95-99, 1993.

[8] H. Bronnimann, B. Chazelle, and J. Pach. How hard is halfspace range searching. Discrete Comput. Geom., 10:143-155, 1993.

[9] P. B. Callahan and S. R. Kosaraju. A decomposition of multidimensional point-sets with applications to k-nearest-neighbors and n- body potential fields. In Proc. 24th Annu. ACM Sympos. Theory Comput., pages 546-556, 1992.

[10] P. B. Callahan and S. R. Kosaraju. Algorithms for dynamic closest pair and n-body potential fields. In Proc. 6th ACM-SIAM Sympos. Discrete Algorithms, 1995.

[11] B. Chazelle. Lower bounds on the complexity of polytope range searching. J. Amer. Math. Soc., 2:637-666, 1989.

[12] B. Chazelle and E. Welzl. Quasi-optimal range searching in spaces of infinite. VC-dimension. Discrete Comput. Geom., 4:467-489, 1989.

[13] K. L. Clarkson. Fast algorithms for the all nearest neighbors problem. In Proc. 24th Ann. IEEE Sympos. on the Found. Comput. Sci., pages 226-232, 1983.

[14] C. A. Duncan, M. T. Goodrich, and S. G. Kobourov. Balanced aspect ratio trees: Combining the advantages of k-d trees and octrees. In Proc. 10th ACM-SIAM Sympos. Discrete Algorithms, 1999.

[15] N. Farvardin and J. W. Modestino. Rate-distortion performance of DPCM schemes for autoregressive sources. IEEE Transactions on Information Theory, 31:402-418, 1985.

[16] J. Matousek. Range searching with efficient hierarchical cuttings. Discrete Comput. Geom., 10(2):157-182, 1993.

[17] D.M. Mount and E. Park. A Dynamic Data Structure for Approximate Range Searching.

[18] M. H. Overmars. Point location in fat subdivisions. Inform. Process. Lett., 44:261-265, 1992.

[19] F. P. Preparata and M. I. Shamos. Computational Geometry: an Introduction. Springer-Verlag, New York, NY, USA, 1985.

[20] H. Samet. The Design and Analysis of Spatial Data Structures. Addison Wesley, Reading, MA, 1990.

[21] D. E. Willard. Polygon retrieval. SIAM J. Comput., 11:149-165, 1982.

[22] P. M. Vaidya. An O(n log n) algorithm for the all-nearest-neighbors problem. Discrete Comput. Geom., 4:101-115, 1989.

[23] J.L. Bentley, ‘‘Decomposable searching problems’’, Information Processing Letters, vol. 8, No. 5, 244–151, 1979.

[24] J.L. Bentley and J.H. Friedman, ‘‘Data structures for range searching’’, ACM Comput. Suru., vol. 11, No. 4, 397–409, 1979.

[25] I. Emiris, T. Malamatos, E. Tsigaridas Approximate Nearest Neighbor Queries among Parallel Segments.