



Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Ηλεκτρονικής και Υπολογιστών

Υπολογισμός της Ευκλείδειας Απόστασης δύο Τριγωνικών Πλεγμάτων

Διπλωματική Εργασία
Καρελής Παναγιώτης

Επιβλέπων: Πιτσιάνης Νικόλαος
Αναπληρωτής Καθηγητής Α.Π.Θ.

23 Σεπτεμβρίου 2022

Περίληψη

Σε μια πληθώρα εφαρμογών της Υπολογιστικής Γεωμετρίας (Μηχανική με τη Βοήθεια Υπολογιστών (CAE), Προσομοιώσεις με Υπολογιστές, Ρομποτική, Γραφική με Υπολογιστές κ.α.) τα αντικείμενα του χώρου αναπαρίστανται συνήθως από πολυγωνικά πλέγματα. Κοινό πρόβλημα για όλους τους παραπάνω τομείς αποτελεί η εύρεση της απόστασης που διαχωρίζει δύο αντικείμενα και η ανίχνευση σύγκρουσης μεταξύ τους. Στην παρούσα εργασία, προτείνουμε αποδοτικούς αλγορίθμους που υπολογίζουν την Ευκλείδεια απόσταση δύο αντικειμένων του τρισδιάστατου χώρου και τους υλοποιούμε για την περίπτωση των τριγωνικών πλεγμάτων. Για τους αλγορίθμους αυτούς σχεδιάζουμε μια δενδρική δομή δεδομένων που ανήκει στην κατηγορία των Ιεραρχιών Οριοθετικών Όγκων (BVH). Η διαδικασία κατασκευής της παραπάνω δομής είναι παρόμοια με αυτή του KD-Tree, με τη διαφορά ότι η δομή μας διαχειρίζεται χωρικά δεδομένα και κάνει χρήση Οριοθετικών Πλαισίων Ευθυγραμμισμένων με τους Άξονες (AABB). Επιπλέον, περιγράφουμε έναν τρόπο διάσχισης της δομής ώστε να υποστηρίξει ερωτήματα κοντινότερου γείτονα για χωρικά δεδομένα. Η διάσχιση της δενδρικής δομής σχεδιάζεται ως μια κατευθυνόμενη αναζήτηση κατά βάθος (DFS), που στοχεύει στη σμίκρυνση του χώρου αναζήτησης μέσω κλαδέματος του δένδρου κατά την οπισθοδρόμηση. Ακόμη, στην υλοποίηση μας παραλληλοποιούμε τη διαδικασία κατασκευής του δένδρου όπως και τη διαδικασία αναζήτησης της ελάχιστης απόστασης κάνοντας χρήση πολλαπλών νημάτων επεξεργασίας. Τέλος, μετράμε και αναλύουμε την επίδοση των αλγορίθμων μας σε μια σειρά από περιπτώσεις ελέγχου που κατασκευάσαμε.

Λέξεις-κλειδιά:

Υπολογιστική Γεωμετρία, Πολυγωνικά Πλέγματα, Ευκλείδεια Απόσταση, Ιεραρχίες Οριοθετικών Όγκων

Abstract

In a plethora of fields in Computational Geometry (Computer Aided Engineering, Computer Simulations, Robotics, Computer Graphics etc.) objects in space are represented as polygonal meshes. A common problem, for all the above, is the computation of separation distance and collision detection of two objects. In this thesis, we propose efficient algorithms that compute the Euclidean distance of two objects in 3D space and we implement them for the case of triangle meshes. For these algorithms we design a tree data structure that belongs to the family of Bounding Volume Hierarchies (BVH). The construction procedure of this data structure is similar to the one used by the KD-Tree, but it differs, as our data structure manages spatial data and also uses Axis-Aligned Bounding Boxes (AABB). In addition, we describe a traversal scheme of the data structure in order to answer nearest neighbor queries for spatial data. The traversal of the tree structure is implemented as a directed depth first search (DFS), aiming to reduce the searching space by pruning the tree during backtracking. Furthermore, we parallelize the construction of the data structure as well as the procedure of finding the Euclidean distance, using multithreading. Finally, we measure and analyze the efficiency of our algorithms on a series of test cases we created.

Keywords:

Computational Geometry, Polygonal Meshes, Euclidean Distance, Bounding Volumes Hierarchies

Ευχαριστίες

Άδειο

Υπολογισμός της Ευκλείδειας Απόστασης δύο Τριγωνικών Πλεγμάτων

Παναγιώτης Καρελής
karelisp@ece.auth.gr

23 Σεπτεμβρίου 2022

Περιεχόμενα

1 Εισαγωγή	3
1.1 Κίνητρο	3
1.2 Περιγραφή του Προβλήματος	4
1.3 Στόχοι της Διπλωματικής Εργασίας	4
1.4 Διάρθρωση της Διπλωματικής Εργασίας	5
2 Θεωρητικό Υπόβαθρο	6
2.1 Πλέγματα	6
2.2 Οριοθετικοί Όγκοι	8
2.3 Ιεραρχίες Οριοθετικών Όγκων	10
3 Σχετική Βιβλιογραφία	13
3.1 Εύρεση Κοντινότερου Κόμβου από Σημείο	13
3.2 Ανίχνευση Σύγκρουσης και Υπολογισμός Απόστασης Κυρτών Πολυγώνων	15
3.3 Ιεραρχικές Δομές Δεδομένων	15
4 Μεθοδολογία	16
4.1 Υπολογισμοί Απόστασης Στοιχειωδών Γεωμετρικών Αντικειμένων	16
4.1.1 Ευκλείδεια Απόσταση δύο Τριγώνων	16
4.1.2 Ευκλείδεια Απόσταση δύο AABB	16
4.2 Αλγόριθμοι Εξαντλητικής Αναζήτησης	16
4.3 Ορισμός Μετρικής Κόστους Αναζήτησης	16
4.4 Σχεδιασμός μιας BVH Δομής Δεδομένων, το spatial KD-Tree	16
4.4.1 Κατασκευή του sKD-Tree	16
4.4.2 Ερωτήματα Κοντινότερου Γείτονα στο sKD-Tree	16
4.5 Αλγόριθμοι που χρησιμοποιούν τη δομή sKD-Tree	16
4.6 Βελτιστοποίηση των Αλγορίθμων για Πραγματικά Συστήματα Υπολογιστών	16
4.6.1 Παραλληλοποίηση με χρήση Πολλαπλών Νημάτων (Multi-threading)	16
4.6.2 Χρήση Κουβάδων στα Φύλλα του sKD-Tree (Buckets)	16
5 Πειράματα και Αποτελέσματα	17
5.1 Σύγκριση Σχετικού Κόστους των Πράξεων μεταξύ Στοιχειωδών Γεωμετρικών Οντοτήτων	17
5.2 Κατασκευή Δεδομένων Ελέγχου	17
5.3 Εκτίμηση της Μετρικής Κόστους Αναζήτησης	17
5.4 Χρόνοι Κατασκευής του sKD-Tree	17
5.4.1 Σειριακή Κατασκευή	17
5.4.2 Παράλληλη Κατασκευή	17

5.5	Συνολικός Χρόνος Εκτέλεσης - Σύγκριση Αλγορίθμων	17
5.5.1	Δύο Αεροπλάνα	17
5.5.2	Scooby με Stanford Bunny	17
5.5.3	Δύο Ομοαξονικοί Κύλινδροι	17
5.5.4	Δύο Αεροπλάνα με Ανομοιόμορφο Πλέγμα	17
6	Συμπεράσματα και Μελλοντική Εργασία	18
6.1	Σχολιασμός των Αποτελεσμάτων από τα Πειράματα	18
6.2	Μελλοντική Εργασία	18
Α΄	Ακρωνύμια και συντομογραφίες	19

Κεφάλαιο 1

Εισαγωγή

1.1 Κίνητρο

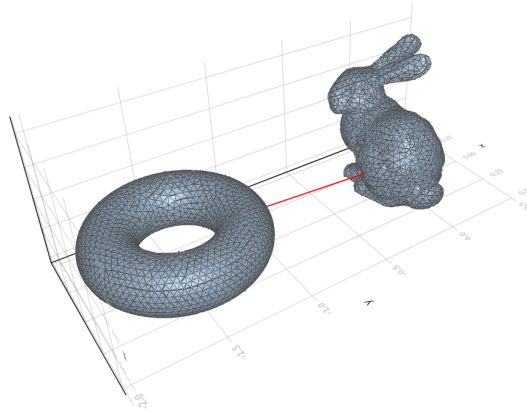
Ο υπολογισμός της απόστασης που διαχωρίζει δύο αντικείμενα στο χώρο αποτελεί θεμελιώδες πρόβλημα στον τομέα της Υπολογιστικής Γεωμετρίας. Στη ρομποτική, στη σχεδίαση και μηχανική με τη βοήθεια υπολογιστών (CAD και CAE), στις προσομοιώσεις με υπολογιστές και στη γραφική υπολογιστών είναι σημαντικό να γνωρίζουμε εάν δύο αντικείμενα, που περιγράφονται από μαθηματικά μοντέλα στον τρισδιάστατο χώρο, τέμνονται/συγκρούονται ή βρίσκονται σε κοντινή απόσταση. Για την παρούσα διπλωματική εργασία, εξετάζουμε την περίπτωση όπου τα αντικείμενα του χώρου περιγράφονται από τριγωνικά πλέγματα (βλ. 2.1).

Στη ρομποτική, για παράδειγμα, η επίλυση του παραπάνω προβλήματος είναι απαιτητή για τον σχεδιασμό διαδρομής με την παρουσία εμποδίων (path-planning problem) [4], [5], [6], [7]. Επιπλέον, σε εφαρμογές CAD-CAE στις οποίες σχεδιάζονται περίπλοκες κατασκευές που αποτελούνται από μεγάλο αριθμό εξαρτημάτων, ο εντοπισμός σύγκρουσης μεταξύ των εξαρτημάτων είναι απαραίτητος τόσο για τις αναλύσεις και δοκιμές των προϊόντων όσο και για την παραγωγή τους [3]. Στη γραφική με υπολογιστές και συγκεκριμένα κατά την κίνηση των αντικειμένων σε μια εικονική σκηνή, όπως στα βιντεοπαιχνίδια και στα κινούμενα σχέδια, είναι πιθανό τα αντικείμενα να διεισδύσουν το ένα στο άλλο. Αυτή η κατάσταση δεν είναι επιθυμητή όταν με τα γραφικά επιδιώκεται η αναπαράσταση ενός ρεαλιστικού κόσμου [12]. Τέλος, το πρόβλημα υπολογισμού της απόστασης που διαχωρίζει δύο αντικείμενα στο χώρο συναντάται και στον τομέα της υπολογιστικής φυσικής για εφαρμογές ανάλυσης πεπερασμένων στοιχείων (FEA) και προσομοιώσεων [11].

Η ραγδαία ανάπτυξη όλων των παραπάνω τομέων τα τελευταία χρόνια και η απαίτηση λεπτομερέστερης περιγραφής των αντικειμένων του τρισδιάστατου χώρου, κάνουν επιτακτική την ανάγκη για μελέτη και σχεδιασμό αλγορίθμων ικανών να διαχειριστούν μεγάλους όγκους δεδομένων εισόδου. Σε αυτή τη διπλωματική εργασία προτείνουμε αποδοτικούς αλγορίθμους για τον υπολογισμό της απόστασης δύο πλεγμάτων στον τρισδιάστατο χώρο.

1.2 Περιγραφή του Προβλήματος

Δοθέντων δύο αντικείμενων στον χώρο, τα οποία περιγράφονται από τριγωνικά πλέγματα, το πιο φυσικό μέγεθος για την περιγραφή της εγγύτητας μεταξύ τους είναι η Ευκλείδεια απόσταση. Δηλαδή, το μήκος του μικρότερου ευθυγράμμου τμήματος που ενώνει τα δύο αντικείμενα. Σε περίπτωση που τα αντικείμενα συγκρούονται η απόσταση τους είναι μηδέν.



Σχήμα 1.1: Ευκλείδεια Απόσταση δύο Τριγωνικών Πλεγμάτων - Το κόκκινο ευθύγραμμο τμήμα αναπαριστά την Ευκλείδεια απόσταση ενός Τόρου και του Stanford Bunny.

Με τον ίδιο ακριβώς τρόπο ορίζεται και η Ευκλείδεια απόσταση μεταξύ δύο τριγώνων στον τρισδιάστατο χώρο. Σχεδιάζοντας μια τέτοια ρουτίνα (βλ. 4.1.1) τότε μπορούμε να δώσουμε έναν εναλλακτικό ορισμό του προβλήματος και ταυτόχρονα έναν αλγόριθμο που το επιλύει:

Ορισμός 1 Έστω δύο αντικείμενα του τρισδιάστατου χώρου που περιγράφονται από τριγωνικά πλέγματα. Επιπλέον, έστω τα σύνολα X, Y που αποτελούνται από τα τρίγωνα των δύο πλεγμάτων, αντίστοιχα, και $tria_dist(x, y)$ η ρουτίνα που υπολογίζει την Ευκλείδεια απόσταση δύο τριγώνων x, y . Η Ευκλείδεια απόσταση d των δύο τριγωνικών πλεγμάτων είναι

$$d = \min_{x \in X, y \in Y} tria_dist(x, y)$$

Ο τετριμμένος αλγόριθμος, που βασίζεται στον ορισμό, υπολογίζει την απόσταση κάθε πιθανού ζεύγους τριγώνων και επιλέγει την ελάχιστη. Τέτοιοι αλγόριθμοι εξαντλητικής αναζήτησης περιγράφονται στο 4.2, όμως είναι αδύνατο να χρησιμοποιηθούν σε πραγματικές εφαρμογές. Ο λόγος είναι η μεγάλη υπολογιστική τους πολυπλοκότητα της τάξης $O(N \cdot M)$ (όπου N, M το πλήθος των τριγώνων των συνόλων X, Y αντίστοιχα).

1.3 Στόχοι της Διπλωματικής Εργασίας

Η συνεισφορά της διπλωματικής εργασίας είναι η ακόλουθη:

- Σχεδιάζουμε μια δενδρική δομή δεδομένων που ανήκει στην οικογένεια των Ιεραρχικών Οριοθετικών Όγκων (BVH). Η δομή αυτή χρησιμοποιείται ως χωρικό ευρετήριο (spatial indexing).
- Προτείνουμε έναν τρόπο διάσχισης της παραπάνω δομής ώστε να υποστηρίζονται ερωτήματα κοντινότερου γείτονα για χωρικά δεδομένα. Στόχος είναι να ελεγχθεί μόνο ένα μικρό υποσύνολο του χώρου αναζήτησης. Αυτό επιτυγχάνεται με το κλάδεμα (pruning) του δένδρου κατά την αναζήτηση.
- Προτείνουμε δύο αλγορίθμους που επιλύουν το πρόβλημα υπολογισμού της Ευκλείδειας απόστασης δύο πλεγμάτων. Οι αλγόριθμοι είναι γενικοί και μπορούν να εφαρμοστούν σε πλέγματα που αποτελούνται από διαφόρων ειδών πολύτοπα (πολύγωνα και πολύεδρα) υπό προϋποθέσεις. Στην εργασία αυτή μελετώνται τα τριγωνικά πλέγματα.
- Υλοποιούμε τους αλγορίθμους για συστήματα υψηλής απόδοσης που υποστηρίζουν πολλαπλά νήματα (multithreading). Παραλληλοποιούμε τις διαδικασίες κατασκευής του δένδρου και αναζήτησης της ελάχιστης απόστασης.
- Μετράμε και αναλύουμε την επίδοση των αλγορίθμων μας σε μια σειρά από περιπτώσεις ελέγχου που κατασκευάσαμε.

1.4 Διάρθρωση της Διπλωματικής Εργασίας

Στο **Κεφάλαιο 1** έγινε μια εισαγωγή στο πρόβλημα υπολογισμού της απόστασης δύο πλεγμάτων και παρουσιάστηκαν τα κίνητρα που οδήγησαν στην υλοποίηση των αλγορίθμων που θα παρουσιαστούν.

Στο **Κεφάλαιο 2** παρουσιάζεται το θεωρητικό υπόβαθρο που απαιτείται από τον αναγνώστη ώστε να κατανοήσει πλήρως το πρόβλημα και την προτεινόμενη λύση.

Στο **Κεφάλαιο 3** αναφέρεται η σχετική βιβλιογραφία, δηλαδή πώς αντιμετώπισαν άλλοι ερευνητές το ίδιο ή παρόμοια προβλήματα. Παρουσιάζονται επίσης ομοιότητες και διαφορές των υπολοίπων προσεγγίσεων σε σχέση με τη δική μας.

Στο **Κεφάλαιο 4** αναλύεται η μεθοδολογία που προτείνουμε και παρουσιάζεται η υλοποίηση των αλγορίθμων μας.

Στο **Κεφάλαιο 5** παρατίθενται τα αποτελέσματα από τα πειράματα που εκτελέσαμε. Οι μετρήσεις των πειραμάτων περιλαμβάνουν την εκτίμηση της μετρικής κόστους που ορίζεται στην ενότητα 4.3 καθώς και τους χρόνους εκτέλεσης των αλγορίθμων.

Στο **Κεφάλαιο 6** σχολιάζονται τα αποτελέσματα και παρουσιάζονται σκέψεις για μελλοντική επέκταση και βελτίωση των ιδεών της παρούσας διπλωματικής εργασίας.

Κεφάλαιο 2

Θεωρητικό Υπόβαθρο

2.1 Πλέγματα

Στην Υπολογιστική Γεωμετρία τα *πλέγματα* αποτελούν την αναπαράσταση μιας μεγαλύτερης γεωμετρικής περιοχής από μικρότερα διακριτά στοιχεία. Τα πλέγματα χρησιμοποιούνται συνήθως για τον υπολογισμό λύσεων μερικών διαφορικών εξισώσεων, για την απόδοση γραφικών υπολογιστών, και για ανάλυση γεωγραφικών και χαρτογραφικών δεδομένων. Ένα πλέγμα χωρίζει τον χώρο σε μικρότερα στοιχεία (πολύγωνα ή πολύεδρα) όπου μπορούν να λυθούν οι εξισώσεις, το οποίο στη συνέχεια προσεγγίζει τη λύση στο ευρύτερο πεδίο. Τα πλέγματα που αποτελούνται από πολύεδρα αντιπροσωπεύουν ρητά τόσο την επιφάνεια όσο και τον όγκο ενός αντικειμένου, ενώ τα πολυγωνικά πλέγματα αντιπροσωπεύουν μόνο την επιφάνεια (ο όγκος υπονοείται). Για το πρόβλημα υπολογισμού της Ευκλείδειας απόστασης, ενδιαφερόμαστε μόνο για την εξωτερική επιφάνεια των αντικειμένων του τρισδιάστατου χώρου.



Σχήμα 2.1: Παράδειγμα τριγωνικού πλέγματος που αναπαριστά ένα δελφίνι.

Ένας τύπος πολυγωνικών πλεγμάτων είναι τα τριγωνικά πλέγματα (σχήμα 2.1). Αποτελούνται από ένα σύνολο τριγώνων στις τρεις διαστάσεις, τα οποία συνδέονται με τις κοινές τους ακμές ή κορυφές. Γεωμετρικά, ένα πλέγμα είναι μια τμηματικά επίπεδη επιφάνεια. Η τελευταία ιδιότητα ισχύει πάντοτε για τα τριγωνικά πλέγματα.

Υπάρχουν διάφοροι τρόποι για την αποθήκευση ενός τριγωνικού πλέγματος στη μνήμη του υπολογιστή. Υπάρχουν επίσης μέθοδοι μετατροπής του ενός τρόπου αποθήκευσης σε έναν άλλο. Ενδεικτικά αναφέρουμε την αποθήκευση με:

- **Σετ τριγώνων:** Το πλέγμα αναπαρίσταται απλά από τα τρίγωνα του. Δηλαδή αποθηκεύονται οι συντεταγμένες των κορυφών κάθε τριγώνου.
- **Σετ Τριγώνων με δείκτες:** Το πλέγμα αναπαρίσταται από ένα σετ κόμβων και ένα σετ από τριπλέτες με δείκτες στους κόμβους. Η κάθε τριπλέτα αναπαριστά ένα τρίγωνο.
- **Λωρίδες Τριγώνων:** Η αποθήκευση αυτή βασίζεται στο γεγονός ότι δύο γειτονικά τρίγωνα μοιράζονται τη μία τους πλευρά. Αυτός ο τρόπος χρησιμοποιείται για συμπίεση των πλεγμάτων.

- **Δομή Τρίγωνου-Γείτονα:** Υποστηρίζει ερωτήματα γειτνίασης τριγώνων.
- **Δομή Winged-Edge:** Αποθηκεύει δεδομένα κόμβων, ακμών και όψεων. Επιτρέπει εύκολη διάβαση στο πλέγμα μεταξύ όψεων, ακμών και κορυφών.

Για τους αλγορίθμους που περιγράφουμε παρακάτω αρκεί ο πρώτος τρόπος αναπαράστασης.

Τέλος, για τις διάφορες εφαρμογές που χρησιμοποιούνται, τα πλέγματα χαρακτηρίζονται και από την ποιότητα τους. Οι πιο συνηθισμένες μετρικές ποιότητας είναι:

- **Λοξότητα (Skewness):**
Η λοξότητα είναι ο λόγος της απόκλισης μεταξύ του βέλτιστου μεγέθους στοιχείου προς στο υπάρχον μέγεθος στοιχείου. Το εύρος της λοξότητας είναι μεταξύ 0 (ιδανικό) έως 1 (χειρότερο). Τα πολύ λοξά στοιχεία δεν προτιμώνται λόγω της κακής ακρίβειας που προκαλούν στις παρεμβαλλόμενες περιοχές. Ανάλογα με το στοιχείο (τρίγωνο, τετράπλευρο, τετράεδρο, εξάεδρο κλπ) διαφοροποιείται μαθηματικός τύπος για τον υπολογισμό της λοξότητας.
- **Ομαλότητα (Smoothness):**
Η αλλαγή στο μέγεθος των στοιχείων πρέπει να είναι ομαλή. Συνήθως αποφεύγονται ξαφνικά άλματα στο μέγεθος των στοιχείων γιατί αυτό μπορεί να προκαλέσει λανθασμένα αποτελέσματα σε κοντινούς κόμβους.
- **Αναλογία Διαστάσεων (Aspect Ratio):**
Εν συντομία, ο λόγος διαστάσεων είναι ο λόγος του μεγαλύτερου μήκους ενός στοιχείου προς το μικρότερο μήκος. Ο ιδανικός λόγος διαστάσεων είναι 1. Όσο μικρότερος είναι, τόσο υψηλότερη είναι η ποιότητα ενός στοιχείου. Η μέθοδος υπολογισμού ποικίλλει ανάλογα με τον τύπο κελιού.

Σε πραγματικές εφαρμογές, τα πλέγματα συνήθως σχεδιάζονται έτσι ώστε να μην παραβιάζουν σε μεγάλο βαθμό τις παραπάνω μετρικές ποιότητας. Αυτή η παρατήρηση είναι χρήσιμη για τον σχεδιασμό της δομής δεδομένων που προτείνουμε.



Σχήμα 2.2: Παράδειγμα Ποιότητας Πλεγμάτων - Και τα δύο πλέγματα αναπαριστούν το ίδιο αεροπλάνο. Το δεξί πλέγμα είναι καλύτερης ποιότητας από το αριστερό που φαίνεται να παραβιάζει όλα τα κριτήρια ποιότητας που αναφέρθηκαν (skewness, smoothness, aspect ratio). Και τα δύο πλέγματα αποτελούνται από τον ίδιο αριθμό τριγώνων περίπου (γύρω στα 15000 τρίγωνα).

2.2 Οριοθετικοί Όγκοι

Οριοθετικός όγκος (bounding volume) ενός συνόλου από αντικείμενα του τρισδιάστατου χώρου ονομάζεται οποιοσδήποτε κλειστός όγκος που εξ' ολοκλήρου περικλείει τα αντικείμενα του συνόλου. Οι οριοθετικοί όγκοι χρησιμοποιούνται για να επιταχύνουν αλγορίθμους που εκτελούν γεωμετρικούς ελέγχους χρησιμοποιώντας απλούς όγκους που περικλείουν πολύπλοκα αντικείμενα. Οι έλεγχοι σε οριοθετικούς όγκους είναι τυπικά πολύ ταχύτεροι από ελέγχους στο ίδιο το στοιχείο ή αντικείμενο που περικλείουν. Σε πολλές περιπτώσεις, ένας έλεγχος αρκεί για να απορριφθούν ή να επιβεβαιωθούν πολλαπλοί έλεγχοι που θα απαιτούνταν για κάθε ένα από τα στοιχεία ξεχωριστά.

Οι οριοθετικοί όγκοι βρίσκουν ευρεία εφαρμογή στους παρακάτω τομείς:

- **Ανίχνευση Ακτίνων (Ray Tracing):**

Οι οριοθετικοί όγκοι χρησιμοποιούνται σε ελέγχους τομής ακτίνων με αντικείμενα και σε πολλούς αλγόριθμους απόδοσης γραφικών. Για παράδειγμα, εάν η ακτίνα ή το οπτικό πεδίο της κάμερας δεν τέμνει τον οριοθετικό όγκο, τότε δεν μπορεί να τέμνει ούτε το αντικείμενο που περιέχεται μέσα. Έτσι αποφεύγονται οι αντίστοιχοι έλεγχοι, οι οποίοι κοστίζουν υπολογιστικά. Όμοια, εάν το οπτικό πεδίο της κάμερας περιέχει εξ' ολοκλήρου τον οριοθετικό όγκο, το αντικείμενο, δηλαδή τα στοιχεία από τα οποία αποτελείται θα απεικονιστούν στην οθόνη χωρίς περισσότερους ελέγχους.

- **Ανίχνευση Σύγκρουσης (Collision Detection):**

Όμοια με πριν, όταν δύο οριοθετικοί όγκοι δε συγκρούονται/τέμνονται, τότε ούτε και τα αντικείμενα που περικλείουν δεν μπορούν να συγκρούονται.

Για τη δημιουργία οριοθετικών όγκων σύνθετων αντικειμένων, συνήθως χρησιμοποιούνται *Ιεραρχίες Οριοθετικών Όγκων* (βλ. 2.3). Δηλαδή, δένδρικές δομές δεδομένων όπου η βασική ιδέα κατασκευής τους είναι η ρίζα να περικλείει ολόκληρο το αντικείμενο ενώ τα φύλλα ένα μικρό υποσύνολο του.

Η επιλογή του τύπου οριοθετικού όγκου για μια δεδομένη εφαρμογή καθορίζεται από διάφορους παράγοντες. Τέτοιοι είναι το κόστος υπολογισμού ενός οριοθετικού όγκου για ένα αντικείμενο, το κόστος της ενημέρωσης του σε εφαρμογές στις οποίες τα αντικείμενα μπορούν να μετακινηθούν ή να αλλάξουν σχήμα, το κόστος ανίχνευσης σύγκρουσης ή υπολογισμού απόστασης και η επιθυμητή ακρίβεια για ελέγχους σύγκρουσης ή απόστασης. Η ακρίβεια αυτή σχετίζεται με τον όγκο του *κενού χώρου* που περικλείεται από τον οριοθετικό όγκο όμως δε σχετίζεται με το οριοθετημένο αντικείμενο. Τυπικά, ισχύει ο εξής συμβιβασμός: οι πιο εκλεπτυσμένοι οριοθετικοί όγκοι περικλείουν γενικά λιγότερο κενό χώρο, αλλά είναι πιο ακριβοί υπολογιστικά.

Οι πιο συνηθισμένοι τύποι οριοθετικών όγκων είναι:

- Η **Οριοθετική Σφαίρα (Bounding Sphere)**, η οποία είναι μια σφαίρα που περικλείει το αντικείμενο. Αναπαρίσταται από το κέντρο και την ακτίνα της και επιτρέπει πολύ γρήγορους ελέγχους σύγκρουσης και υπολογισμού απόστασης. Δύο σφαίρες τέμνονται όταν η απόσταση μεταξύ των κέντρων τους δεν υπερβαίνει το άθροισμα των ακτίνων τους.
- Ο **Οριοθετικός Κύλινδρος (Bounding Cylinder)**, είναι ένας κύλινδρος που περικλείει το αντικείμενο. Στις περισσότερες εφαρμογές ο άξονας του κυλίνδρου

είναι ευθυγραμμισμένος με την κατακόρυφη διεύθυνση της σκηνής. Οι κύλινδροι είναι κατάλληλοι για τρισδιάστατα αντικείμενα που μπορούν να περιστρέφονται μόνο γύρω από έναν κατακόρυφο άξονα αλλά όχι γύρω από άλλους άξονες, και κατά τα άλλα η κίνηση τους να είναι μόνο μεταφορική. Δύο κύλινδροι ευθυγραμμισμένοι με κατακόρυφο άξονα τέμνονται όταν, ταυτόχρονα, τέμνονται οι προβολές τους στον κατακόρυφο άξονα (που είναι ευθύγραμμο τμήματα), καθώς και οι προβολές τους στο οριζόντιο επίπεδο (που είναι κύκλοι). Οι δύο συνθήκες είναι εύκολο να ελεγχθούν. Στα βιντεοπαιχνίδια, οι οριοθετικοί κύλινδροι χρησιμοποιούνται συχνά ως οριοθετικοί όγκοι για χαρακτήρες που στέκονται όρθια.

- Το **Οριοθετικό Πλαίσιο (Bounding Box)**, είναι ένα ορθογώνιο παραλληλεπίπεδο που περικλείει το αντικείμενο. Στις προσομοιώσεις όπου η σκηνή αλλάζει δυναμικά, τα οριοθετικά πλαίσια προτιμώνται από άλλα σχήματα οριοθετικών όγκων (σφαίρες ή κυλίνδρους) για αντικείμενα που έχουν χονδρικά κυβοειδές σχήμα όταν ο έλεγχος σύγκρουσης πρέπει να είναι αρκετά ακριβής. Το όφελος είναι προφανές, για παράδειγμα, για αντικείμενα που ακουμπούν πάνω σε άλλα, όπως ένα αυτοκίνητο που ακουμπά στο έδαφος: μια οριοθετική σφαίρα θα έδειχνε πως το αυτοκίνητο πιθανώς να τέμνεται με το έδαφος, το οποίο στη συνέχεια θα έπρεπε να απορριφθεί από μια πιο ακριβή υπολογιστικά δοκιμή του πραγματικού μοντέλου του αυτοκινήτου. Ένα οριοθετικό πλαίσιο δείχνει αμέσως ότι το αυτοκίνητο δεν τέμνεται με το έδαφος, εξοικονομώντας έτσι την κοστοβόρα δοκιμή.

Στη γενική περίπτωση, ένα αυθαίρετο οριοθετικό πλαίσιο ονομάζεται και **Προσανατολισμένο Οριοθετικό Πλαίσιο (Oriented Bounding Box) OBB** ή **OBBB** όταν χρησιμοποιείται το τοπικό σύστημα συντεταγμένων ενός αντικείμενου. Σε πολλές εφαρμογές το οριοθετικό πλαίσιο είναι ευθυγραμμισμένο με τους άξονες του συστήματος συντεταγμένων και ονομάζεται **Οριοθετικό Πλαίσιο Ευθυγραμμισμένο με τους Άξονες (Axis-Aligned Bounding Box)** ή **AABB**. Τα AABB είναι πιο απλά και αποδοτικά στην ανίχνευση σύγκρουσης μεταξύ τους από τα OBB, αλλά έχουν το μειονέκτημα ότι όταν το μοντέλο περιστρέφεται δεν μπορούν απλώς να περιστραφούν με αυτό, αλλά πρέπει να υπολογιστούν εκ νέου.

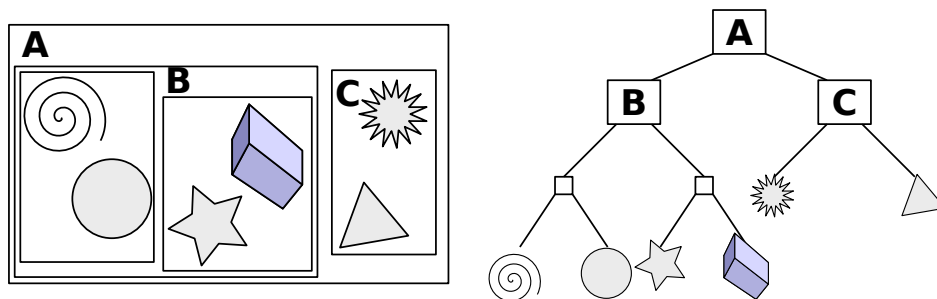
Στην περίπτωση των δύο διαστάσεων, χρησιμοποιείται το **Ελάχιστο Οριοθετικό Παραλληλόγραμμα (Minimum Bounding Rectangle)** ή **MBR**. Το MBR είναι ειδική περίπτωση του AABB στο επίπεδο και χρησιμοποιείται συχνά για να περικλείει γεωγραφικά (ή γεωχωρικά) δεδομένα.

- Η **Οριοθετική Κάψουλα (Bounding Capsule)**, η οποία προκύπτει από τον συνολικό όγκο που περικλείει μια σφαίρα καθώς κινείται πάνω σε ένα ευθύγραμμο τμήμα (swept sphere). Ο όγκος που προκύπτει αποτελείται από έναν κύλινδρο και δύο ημισφαίρια στα άκρα του. Μπορούν να αναπαρασταθούν από το ευθύγραμμο τμήμα και το μήκος της ακτίνας της σφαίρας. Έχει χαρακτηριστικά παρόμοια με έναν κύλινδρο, αλλά είναι πιο εύκολη στη χρήση, επειδή ο έλεγχος σύγκρουσης είναι απλούστερος. Για παράδειγμα, δύο κάψουλες τέμνονται εάν η απόσταση μεταξύ των τμημάτων που τις ορίζουν είναι μικρότερη από το άθροισμα των ακτίνων τους. Αυτό ισχύει για τις αυθαίρετα προσανατολισμένες κάψουλες, γι' αυτό είναι πιο ελκυστικές από τους κυλίνδρους στην πράξη.

Στα πλαίσια αυτής της εργασίας, θα χρησιμοποιηθούν τα AABB.

2.3 Ιεραρχίες Οριοθετικών Όγκων

Η *Ιεραρχία Οριοθετικών Όγκων* (**Bounding Volume Hierarchy** ή **BVH**) είναι μια δενδρική δομή που αποθηκεύει και οργανώνει γεωμετρικά (ή χωρικά) στοιχεία (elements). Τέτοια είναι τα πολύτοπα, τα πολύεδρα, παραμετρικές καμπύλες ή επιφάνειες και άλλα. Κάθε κόμβος του δένδρου αναπαρίσταται από τον οριοθετικό όγκο του υποσυνόλου των στοιχείων που αποθηκεύει. Ο όγκος αυτός προκύπτει από τη συγχώνευση των οριοθετικών όγκων των παιδιών του κόμβου. Με αυτό τον αναδρομικό τρόπο, οργανώνεται το σύνολο των στοιχείων του δένδρου από τα φύλλα έως τη ρίζα. Συνεπώς, η ρίζα του δένδρου αναπαρίσταται από τον οριοθετικό όγκο που περικλείει όλα τα στοιχεία του δένδρου, ενώ τα φύλλα αναπαρίστανται από τους οριοθετικούς όγκους του κάθε στοιχείου ξεχωριστά (βλ. Σχήμα 2.3).



Σχήμα 2.3: Παράδειγμα μιας Ιεραρχίας Οριοθετικών Όγκων στις δύο διαστάσεις που χρησιμοποιεί ορθογώνια παραλληλόγραμμα για οριοθετικούς όγκους. Αριστερά απεικονίζονται τα αντικείμενα και τα οριοθετικά πλαίσια που περικλείουν τα αντικείμενα. Δεξιά απεικονίζεται η ιεραρχία σε μορφή δέντρου.

Γενικά, η ενθυλάκωση αντικειμένων μέσα σε οριοθετικούς όγκους επιταχύνει πολλές γεωμετρικές πράξεις πχ. collision detection, ray tracing, κλπ όπως έγινε φανερό στο 2.2, παρ' όλα αυτά δε μειώνει τον αριθμό των ελέγχων που απαιτούνται. Δηλαδή η υπολογιστική πολυπλοκότητα των αλγορίθμων παραμένει ίδια (βλ. 4.2 για ένα παράδειγμα τέτοιου αλγορίθμου). Με την οργάνωση των αντικειμένων σε ιεραρχίες οριοθετικών όγκων, ωστόσο, η υπολογιστική πολυπλοκότητα (δηλαδή το πλήθος των ελέγχων που εκτελούνται) μπορεί να μειωθεί σε λογαριθμική σε σχέση με το πλήθος των αντικειμένων. Σε μια τέτοια ιεραρχία τα αντικείμενα ενός κόμβου δε χρειάζεται να εξετασθούν αν το αποτέλεσμα του ελέγχου μπορεί να εξακριβωθεί από κάποιον πρόγονό του. Αυτή η τεχνική ονομάζεται κλάδεμα του δέντρου (tree pruning).

Για τη σχεδίαση ιεραρχιών οριοθετικών όγκων υπάρχουν πολλές επιλογές.

- Η *επιλογή του οριοθετικού όγκου* που θα χρησιμοποιείται από την ιεραρχία παίζει σημαντικό λόγο. Η επιλογή καθορίζεται από μια σύμβαση μεταξύ δύο στόχων: Από τη μία πλευρά, θα θέλαμε να χρησιμοποιήσουμε οριοθετικούς όγκους που έχουν πολύ απλό σχήμα γιατί η αποθήκευσή τους στη μνήμη είναι μικρή και οι διάφοροι γεωμετρικοί έλεγχοι είναι γρήγοροι. Από την άλλη πλευρά, θα θέλαμε να έχουμε οριοθετικούς όγκους που να περικλείουν πολύ στενά τα αντικείμενα. Ο πιο κοινός τύπος οριοθετικού όγκου που χρησιμοποιείται είναι τα AABB.
- Η *τεχνική κατασκευής του δένδρου* αποτελεί επίσης σχεδιαστική επιλογή. Υπάρχουν τρεις κύριες κατηγορίες αλγορίθμων κατασκευής δένδρων:

- Οι *από πάνω προς τα κάτω (top-down)*, οι οποίοι λειτουργούν με διαμερισμό του συνόλου εισόδου σε δύο (ή περισσότερα) υποσύνολα, οριοθετώντας τα στον επιλεγμένο οριοθετικό όγκο και, στη συνέχεια, επαναλαμβάνουν την ίδια διαδικασία αναδρομικά έως ότου κάθε υποσύνολο να αποτελείται από ένα μόνο στοιχείο (δηλαδή η διαδικασία τερματίζει όταν φτάσει στα φύλλα του δέντρου).
- Οι *από κάτω προς τα πάνω (bottom-up)*, οι οποίοι ξεκινούν με το σύνολο των στοιχείων που αποθηκεύονται στα φύλλα του δέντρου και στη συνέχεια ομαδοποιούν δύο (ή περισσότερα) από αυτά για να σχηματίσουν έναν νέο εσωτερικό κόμβο. Η διαδικασία συνεχίζεται με τον ίδιο τρόπο έως ότου όλα τα στοιχεία ομαδοποιηθούν σε έναν μόνο κόμβο (τη ρίζα του δέντρου). Οι μέθοδοι από κάτω προς τα πάνω είναι πιο δύσκολοι στην υλοποίηση, αλλά είναι πιθανό να παράγουν καλύτερης ποιότητας δέντρα γενικά. Ένας ευρέως διαδεδομένος τρόπος για ομαδοποίηση των δεδομένων από κάτω προς τα πάνω είναι η χρήση *καμπυλών πλήρωσης χώρου (space filling curves)* [9] [10]. Τα στοιχεία ταξινομούνται αρχικά με βάση την καμπύλη και ομαδοποιούνται σύμφωνα με την ακολουθία που προκύπτει.
- Οι *μέθοδοι με εισαγωγή (insertion methods)*, οι οποίοι κατασκευάζουν το δέντρο εισάγοντας ένα στοιχείο τη φορά, ξεκινώντας από ένα κενό δέντρο. Η θέση που τοποθετείται ένα νέο στοιχείο επιλέγεται έτσι ώστε το δέντρο να μεγαλώσει όσο το δυνατόν λιγότερο, σύμφωνα με κάποια μετρική κόστους. Οι μέθοδοι με εισαγωγή θεωρούνται on-line μέθοδοι δεδομένου ότι δεν απαιτούν να είναι διαθέσιμα όλα τα στοιχεία πριν από την έναρξη της κατασκευής και επομένως επιτρέπουν την εκτέλεση ενημερώσεων (updates) κατά το χρόνο εκτέλεσης. Οι άλλες δύο κατηγορίες που αναφέρθηκαν θεωρούνται off-line μέθοδοι καθώς και οι δύο απαιτούν να είναι διαθέσιμα όλα τα στοιχεία πριν από την έναρξη της κατασκευής.

Η εισαγωγή στοιχείων στο BVH συνήθως καταλήγει σε δέντρα με χειρότερη επίδοση στα ερωτήματα σε σχέση με μια πλήρη ανακατασκευή από την αρχή. Οι λύσεις που προτείνονται είναι είτε η ασύγχρονη ανακατασκευή του δέντρου, είτε η ανακατασκευή όταν ανιχνευθεί σημαντική αλλαγή (πχ υπάρχει μεγάλη επικάλυψη κοντά στα φύλλα, ή ο αριθμός των προσθαφαιρέσεων στοιχείων ξεπεράσει κάποιο κατώφλι, ή άλλες πιο εκλεπτυσμένες ευρετικές μεθόδους).

- Ο *βαθμός* (αριθμός των παιδιών των κόμβων) του δένδρου. Ένα δέντρο χαμηλού βαθμού θα έχει μεγαλύτερο ύψος. Αυτό αυξάνει τον χρόνο διάσχισης από τη ρίζα έως τα φύλλα. Από την άλλη πλευρά, λιγότεροι υπολογισμοί πρέπει να δαπανηθούν σε κάθε κόμβο που επισκέπτεται ο αλγόριθμος για να ελέγξει ποια παιδιά θα επισκεφθεί και με ποια σειρά. Το αντίθετο ισχύει για ένα δέντρο υψηλού βαθμού, δηλαδή αν και το δέντρο θα είναι μικρότερου ύψους, δαπανάται περισσότερη δουλειά σε κάθε κόμβο. Στην πράξη, τα δυαδικά δέντρα (βαθμός = 2) είναι μακράν τα πιο κοινά. Ένας από τους κύριους λόγους είναι ότι τα δυαδικά δέντρα είναι πιο εύκολο να κατασκευαστούν.
- Η *μέθοδος διάσχισης* του δένδρου για την απάντηση γεωμετρικών ερωτημάτων. Όταν θα πρέπει να γίνει επίσκεψη σε περισσότερα από ένα από τα παιδιά ενός κόμβου, η σειρά με την οποία αυτά θα ελεγχθούν δεν επηρεάζει την ορθότητα του

αλγορίθμου, όμως έχει αντίκτυπο στην επίδοση του.

Τέλος, αναφέρουμε κάποιες επιθυμητές ιδιότητες [8] μιας Ιεραρχίας Οριοθετικών Όγκων οι οποίες πρέπει να ληφθούν υπόψη κατά τον σχεδιασμό της για τις διάφορες εφαρμογές:

- Τα στοιχεία που περιλαμβάνει οποιοδήποτε υπο-δέντρο θα πρέπει να είναι κοντά το ένα στο άλλο (χωρικά). Όσο πιο χαμηλά στο δέντρο, τόσο πιο κοντά θα πρέπει να βρίσκονται τα αντικείμενα.
- Ο όγκος της επικάλυψης των αδελφικών κόμβων πρέπει να είναι ελάχιστος
- Κάθε κόμβος του δέντρου πρέπει να έχει τον ελάχιστο δυνατό όγκο.
- Το άθροισμα όλων των οριοθετικών όγκων πρέπει να είναι ελάχιστο.
- Θα πρέπει να δοθεί μεγαλύτερη προσοχή στους κόμβους κοντά στη ρίζα του BVH. Το κλάδεμα ενός κόμβου κοντά στη ρίζα του δέντρου αφαιρεί περισσότερα αντικείμενα από περαιτέρω εξέταση.
- Το BVH θα πρέπει να είναι όσο πιο ισορροπημένο γίνεται. Η εξισορρόπηση επιτρέπει να κλαδευτεί όσο το δυνατόν μεγαλύτερο μέρος του BVH όποτε δε διασχίζεται ένα κλαδί.

Κεφάλαιο 3

Σχετική Βιβλιογραφία

Τα ερωτήματα εγγύτητας και η ανίχνευση σύγκρουσης διερευνώνται εκτενώς για δεκαετίες από ερευνητές στα γραφικά υπολογιστών, στη ρομποτική, στις προσομοιώσεις, την υπολογιστική γεωμετρία και τα κινούμενα σχέδια υπολογιστών. Εκτός από την ανίχνευση σύγκρουσης και τον υπολογισμό απόστασης, υπάρχει σημαντική βιβλιογραφία σχετική με τις Ιεραρχίες Οριοθετικών Όγκων (BVH) και τις διάφορες εφαρμογές τους. Σε αυτό το κεφάλαιο κάνουμε αναφορά σε άρθρα της βιβλιογραφίας που μελετούν το ίδιο ή παρόμοιο πρόβλημα με το δικό μας και επισημαίνουμε τις τεχνικές που χρησιμοποιούνται και στη δική μας μεθοδολογία.

3.1 Εύρεση Κοντινότερου Κόμβου από Σημείο

Αρχικά μελετάμε το παρακάτω πρόβλημα: “Δοθέντος ενός πλέγματος και ενός σημείου (*query-point*), ζητείται να βρεθεί ο κοντινότερος προς το σημείο **κόμβος** του πλέγματος”. Το πρόβλημα αυτό ανάγεται στο κλασικό πρόβλημα εύρεσης των k κοντινότερων γειτόνων από ένα σύνολο σημείων (k NN, όπου $k = 1$).

Για το πρόβλημα αυτό έχουν προταθεί διάφορες λύσεις. Για παράδειγμα η χρήση διαγραμμάτων Voronoi στην περίπτωση δεδομένων με μικρό αριθμό διαστάσεων [1]. Οι σημαντικότερες όμως συνεισφορές για το συγκεκριμένο πρόβλημα είναι αυτές του [2] με την εισαγωγή του kD -Tree, και του [14] με την εισαγωγή του VP -Tree (βλ. για παράδειγμα το [13] όπου γίνεται χρήση του kD -Tree για τον υπολογισμό του κοντινότερου κόμβου ενός τριγωνικού πλέγματος από σημείο).

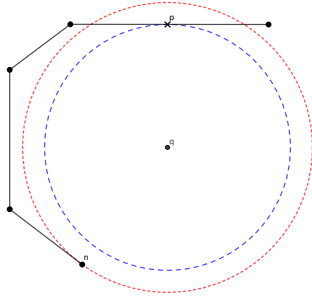
Τα δύο παραπάνω δυαδικά δέντρα αναζήτησης απαιτούν $O(n * \log n)$ χρόνο προεπεξεργασίας των δεδομένων (για την κατασκευή τους από το σύνολο των n σημείων του τρισδιάστατου χώρου) και έπειτα $O(\log n)$ για την απάντηση ερωτημάτων κοντινότερου γείτονα, στη μέση περίπτωση. Στο ίδιο πλαίσιο εργασίας αναπτύσσεται ο πρώτος εκ των δύο αλγορίθμων που προτείνουμε, για το δικό μας πρόβλημα, όπου κατασκευάζουμε μια δομή δεδομένων και στη συνέχεια κάνουμε ερωτήματα κοντινότερου γείτονα για να υπολογίσουμε την απόσταση.

Πιο συγκεκριμένα, για την κατασκευή του kD -Tree το σύνολο των σημείων ενός κόμβου διχοτομείται σε δύο υποσύνολα που αποτελούν τα δύο παιδιά του κόμβου. Η διαδικασία αυτή επαναλαμβάνεται αναδρομικά:

- Η διχοτόμηση γίνεται με βάση κάποιον άξονα ο οποίος εναλλάσσεται κυκλικά (round-robin) για τα διαδοχικά επίπεδα του δέντρου. Για παράδειγμα, στις τρεις διαστάσεις, ο διαχωρισμός γίνεται πρώτα με βάση τον άξονα x , έπειτα με βάση τον άξονα y , έπειτα με τον z , ξανά με βάση τον x και ούτω καθεξής.
- Το κριτήριο για τον διαχωρισμό του συνόλου των σημείων V , ενός κόμβου, σε δύο υποσύνολα L, R είναι το εξής: Χωρίς βλάβη της γενικότητας θεωρούμε πως ο άξονας διαχωρισμού είναι ο x . Αν m είναι η διάμεσος των x -συντεταγμένων των σημείων, τότε $L = \{p \in V \mid p_x \leq m\}$ και $R = \{p \in V \mid p_x \geq m\}$. Τα σύνολα L, R επιλέγονται ώστε να έχουν το ίδιο πλήθος στοιχείων ή να διαφέρουν κατά ένα. Όμοια προκύπτει το κριτήριο και για τους άλλους άξονες. Η διάμεσος m είναι πληροφορία που αποθηκεύεται για κάθε κόμβο του δέντρου και χρησιμοποιείται στα ερωτήματα εύρεσης του κοντινότερου γείτονα.

Έτσι προκύπτει ένα σχεδόν πλήρες, ισορροπημένο, δυαδικό δέντρο. Οι ιδέες 1) της διχοτόμησης του συνόλου των δεδομένων με κυκλική εναλλαγή του άξονα διαχωρισμού και 2) η χρήση της διαμέσου των αντίστοιχων συντεταγμένων για τον διαχωρισμό χρησιμοποιούνται *αυτούσιες* και στη δική μας μεθοδολογία για την κατασκευή μιας Ιεραρχίας Οριοθετικών Όγκων. Η διαφορά είναι πως χειριζόμαστε χωρικά δεδομένα, και όχι σημεία, επομένως επιλέγεται ένα *αντιπροσωπευτικό σημείο* για κάθε χωρικό αντικείμενο και εφαρμόζεται η ίδια μέθοδος κατασκευής της ιεραρχίας.

Για την κατασκευή του *VP-Tree*, ξανά, το σύνολο των σημείων ενός κόμβου διχοτομείται σε δύο υποσύνολα αναδρομικά:



Σχήμα 3.1: Παράδειγμα όπου ο κοντινότερος κόμβος n ενός πλέγματος διαφέρει από το κοντινότερο σημείο p , δοθέντος του query-point q .

- Αρχικά, επιλέγεται ένα τυχαίο σημείο από το σύνολο V των σημείων του κόμβου. Το σημείο αυτό ονομάζεται *vantage-point* (vp) και αποτελεί πληροφορία που αποθηκεύεται στον κόμβο.
- Έπειτα υπολογίζονται οι αποστάσεις όλων των υπολοίπων σημείων προς το vp και η διάμεσος m , που επίσης αποθηκεύεται στον κόμβο.
- Αν m είναι η διάμεσος των αποστάσεων που υπολογίστηκαν τότε το σύνολο του αριστερού παιδιού είναι το $L = \{p \in V \mid distance(p, vp) \leq m\}$ και του δεξιού παιδιού $R = \{p \in V \mid distance(p, vp) \geq m\}$

Ο τυπικός τρόπος για την απάντηση ερωτημάτων κοντινότερου κόμβου είναι, αρχικά, ο υπολογισμός μιας εκτίμησης αυτού διασχίζοντας το δέντρο από τη ρίζα έως κάποιο φύλλο. Το φύλλο αποτελεί την πρώτη εκτίμηση κοντινότερου

κόμβου, ενώ η επιλογή του μονοπατιού από τη ρίζα μέχρι το φύλλο αποτελεί σχεδιαστική επιλογή και χρησιμοποιούνται ευρετικοί μέθοδοι (heuristics). Έπειτα μια σφαίρα εκτείνεται από το query-point έως το φύλλο και, με τη βοήθεια του δέντρου, από τα υπόλοιπα σημεία ελέγχονται μόνο αυτά που βρίσκονται εντός της σφαίρας. Η ακτίνα της σφαίρας μικραίνει όσο ανακαλύπτονται όλο και κοντινότεροι κόμβοι. Αυτή η διαδικασία λέγεται *κλάδεμα* (*pruning*) του δέντρου, και μειώνει τον χώρο αναζήτησης με το να απορρίπτει περιοχές που δεν μπορούν να περιέχουν τον κοντινότερο γείτονα. Με αυτόν τον τρόπο, ο αριθμός των αναμενόμενων συγκρίσεων είναι της τάξης $O(\log n)$ για

τη μέση περίπτωση. Με ίδιο μοτίβο λογικής σχεδιάζεται και η διαδικασία αναζήτησης κοντινότερου γείτονα για χωρικά δεδομένα που προτείνουμε.

Το παραπάνω πρόβλημα δεν πρέπει να συγχέεται με το πρόβλημα εύρεσης του κοντινότερου σημείου ενός πολυγωνικού πλέγματος από κάποιο query-point. Γενικά ο κοντινότερος κόμβος και το κοντινότερο σημείο ενός πλέγματος μπορούν να απέχουν πολύ μεταξύ τους όπως φαίνεται στο Σχήμα 3.1: Η πολυγωνική γραμμή αντιπροσωπεύει το σύνορο ενός υποθετικού πλέγματος στις δύο διαστάσεις και η ελάχιστη απόσταση του πλέγματος από το σημείο q (ακτίνα του κόκκινου κύκλου) διαφέρει από την απόσταση του κοντινότερου κόμβου (ακτίνα του μπλε κύκλου).

3.2 Ανίχνευση Σύγκρουσης και Υπολογισμός Απόστασης Κυρτών Πολυγώνων

3.3 Ιεραρχικές Δομές Δεδομένων

Κεφάλαιο 4

Μεθοδολογία

4.1 Υπολογισμοί Απόστασης Στοιχειωδών Γεωμετρικών Αντικειμένων

4.1.1 Ευκλείδεια Απόσταση δύο Τριγώνων

Ευκλείδεια Απόσταση δύο Ευθυγράμμων Τμημάτων

Ευκλείδεια Απόσταση Σημείου και Τριγώνου

4.1.2 Ευκλείδεια Απόσταση δύο AABB

4.2 Αλγόριθμοι Εξαντλητικής Αναζήτησης

4.3 Ορισμός Μετρικής Κόστους Αναζήτησης

4.4 Σχεδιασμός μιας BVH Δομής Δεδομένων, το spatial KD-Tree

4.4.1 Κατασκευή του sKD-Tree

4.4.2 Ερωτήματα Κοντινότερου Γείτονα στο sKD-Tree

4.5 Αλγόριθμοι που χρησιμοποιούν τη δομή sKD-Tree

4.6 Βελτιστοποίηση των Αλγορίθμων για Πραγματικά Συστήματα Υπολογιστών

4.6.1 Παραλληλοποίηση με χρήση Πολλαπλών Νημάτων (Multithreading)

4.6.2 Χρήση Κουβάδων στα Φύλλα του sKD-Tree (Buckets)

Κεφάλαιο 5

Πειράματα και Αποτελέσματα

- 5.1 Σύγκριση Σχετικού Κόστους των Πράξεων μεταξύ Στοιχειωδών Γεωμετρικών Οντοτήτων**
- 5.2 Κατασκευή Δεδομένων Ελέγχου**
- 5.3 Εκτίμηση της Μετρικής Κόστους Αναζήτησης**
- 5.4 Χρόνοι Κατασκευής του sKD-Tree**
 - 5.4.1 Σειριακή Κατασκευή**
 - 5.4.2 Παράλληλη Κατασκευή**
- 5.5 Συνολικός Χρόνος Εκτέλεσης - Σύγκριση Αλγορίθμων**
 - 5.5.1 Δύο Αεροπλάνα**
 - 5.5.2 Scooby με Stanford Bunny**
 - 5.5.3 Δύο Ομοαξονικοί Κύλινδροι**
 - 5.5.4 Δύο Αεροπλάνα με Ανομοιόμορφο Πλέγμα**

Κεφάλαιο 6

Συμπεράσματα και Μελλοντική Εργασία

6.1 Σχολιασμός των Αποτελεσμάτων από τα Πειράματα

6.2 Μελλοντική Εργασία

Παράρτημα Α΄

Ακρωνύμια και συντομογραφίες

AABB Axis-Aligned Bounding Box

BVH Bounding Volume Hierarchy

CAE Computer Aided Engineering

DFS Depth First Search

FEA Finite Element Analysis

MBR Minimum Bounding Rectangle

OBB Oriented Bounding Box

Bibliography

- [1] F. Aurenhammer. Voronoi diagrams—a survey of a fundamental geometric data structure. *ACM Computing Surveys (CSUR)*, 23(3):345–405, 1991.
- [2] J. L. Bentley. Multidimensional binary search trees used for associative searching. *Communications of the ACM*, 18(9):509–517, 1975.
- [3] J. W. Boyse. Interference detection among solids and surfaces. *Communications of the ACM*, 22(1):3–9, 1979.
- [4] R. A. Brooks and T. Lozano-Perez. A subdivision algorithm in configuration space for findpath with rotation. *IEEE Transactions on Systems, Man, and Cybernetics*, (2):224–233, 1985.
- [5] S. Cameron. A study of the clash detection problem in robotics. In *Proceedings. 1985 IEEE International Conference on Robotics and Automation*, volume 2, pages 488–493. IEEE, 1985.
- [6] J. Canny. Collision detection for moving polyhedra. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2):200–209, 1986.
- [7] R. Culley and K. Kempf. A collision detection algorithm based on velocity and distance bounds. In *Proceedings. 1986 IEEE International Conference on Robotics and Automation*, volume 3, pages 1064–1069. IEEE, 1986.
- [8] C. Ericson. *Real-Time Collision Detection*. Morgan Kaufmann series in interactive 3D technology. Taylor & Francis, 2004.
- [9] Y. Gu, Y. He, K. Fatahalian, and G. Bluelloch. Efficient bvh construction via approximate agglomerative clustering. In *Proceedings of the 5th High-Performance Graphics Conference*, pages 81–88, 2013.
- [10] T. Karras. Thinking parallel, part iii: Tree construction on the gpu. <https://developer.nvidia.com/blog/thinking-parallel-part-iii-tree-construction-gpu/>, December 2012.
- [11] A. Khamayseh and G. Hansen. Use of the spatial kd-tree in computational physics applications. *Commun Comput Phys*, 2(3):545–576, 2007.
- [12] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298, 1988.

- [13] D. A. Simon. *Fast and accurate shape-based registration*. Carnegie Mellon University, 1996.
- [14] P. N. Yianilos. Data structures and algorithms for nearest neighbor. In *Proceedings of the fourth annual ACM-SIAM Symposium on Discrete algorithms*, volume 66, page 311. SIAM, 1993.