

Εργαστήριο 3

Αλκίνοος Αλυσσανδράκης 1072752

1 Principal Components

Για να βρούμε τα Principal Components του μοντέλου αρκεί απλά να υπολογίσουμε τον πίνακα συσχετιστικότητας που προκύπτει από όλες τις κορυφές και στη συνέχεια να βρούμε τις ιδιοτιμές και τα ιδιοδιανύσματα αυτού του πίνακα. Οι ιδιοτιμές είναι τα Principal Components του μοντέλου και η ιδιοτιμή που αντιστοιχεί σε καθένα από αυτά δείχνει πόσο πολύ επιρροή έχει στο μοντέλο.

Έχοντας βρεί τα Principal Components μπορούμε να τα αναπαραστήσουμε στη σκηνή με βέλη. Αρχικά δημιουργούμε τρία βέλη, με τη συνάρτηση `create_arrow` της βιβλιοθήκης `Open3D`, τα οποία θα έχουν μήκος `arrow_length` ανάλογο της ιδιοτιμής που αναπαριστούν και τοποθετούνται με τη μια άκρη στην αρχή του συστήματος συντεταγμένων και την άλλη άκρη να δείχνει στο σημείο $(0, 0, \text{arrow_length})$. Πρέπει τώρα να περιστρέψουμε τα βέλη γύρω από την αρχή των αξόνων προκειμένου να συμπίπτουν με τα διανύσματα των Principal Components.

Η περιστροφή αυτή θα γίνει με την εξής διαδικασία. Γνωρίζουμε την αρχική θέση του κάθε βέλους και την θέση στην οποία θέλουμε να καταλήξει. Θεωρούμε αυτά τα δύο σημεία ως τα διανύσματα `startVector` και `endVector` αντίστοιχα και υπολογίζουμε το εξωτερικό γινόμενο τους, δηλαδή ένα διάνυσμα κάθετο στα δύο προηγούμενα το `normalVector`. Αυτό το διάνυσμα αποτελεί τον άξονα γύρω από τον οποίο θα γίνει η περιστροφή του βέλους. Στη συνέχεια πρέπει να βρούμε τη γωνία ανάμεσα στο `startVector` και το `endVector` με τον τύπο

$$\text{angle} = \arccos \left(\frac{\text{startVector} * \text{endVector}}{||\text{startVector}|| * ||\text{endVector}||} \right)$$

Εν τέλει έχουμε τον άξονα περιστροφής και την γωνία, άρα μπορούμε να εκτελέσουμε την περιστροφή με τις κατάλληλες διαδικασίες της `Open3D`, οπότε και τα βέλη καταλήγουν στη σωστή θέση.

2 Mesh-plane Intersection

2.1 Πρώτος αλγόριθμος

Για να βρούμε ποια τρίγωνα του μοντέλου τέμνονται με ένα επίπεδο εκτελούμε την εξής διαδικασία:

- Απλώνουμε τον πίνακα των τριγώνων από `3 * numberOfTriangles` σε μονοδιάστατο πίνακα
- Αντικαθιστούμε τα στον πίνακα αυτό τα κορυφή που αντιστοιχούν σε κάθε τρίγωνο από τον πίνακα των κορυφών
- Για κάθε κορυφή σε αυτόν τον πίνακα υπολογίζουμε την απόσταση του από το επίπεδο

- Αναθέτουμε σε κάθε κορυφή την τιμή True ή False ανάλογα αν βρίσκεται πάνω ή κάτω από το επίπεδο αντίστοιχα
- Μετατρέπουμε ξανά τον μονοδιάστατο πίνακα σε σχήμα $3 * numberOfTriangles$
- Κάθε τρίγωνο που έχει True σε όλες τις κορυφές βρίσκεται πάνω από το επίπεδο
- Κάθε τρίγωνο που έχει False σε όλες τις κορυφές βρίσκεται κατώ από το επίπεδο
- Αλλιώς τα τρίγωνα που έχουν True και False στις κορυφές, τέμνονται με το επίπεδο

Η διαδικασία αυτή είναι αποτελεσματική αλλά μπορεί να γίνει πιο αποδοτική καθώς ο υπολογισμός της απόστασης του κάθε κορυφή από το επίπεδο δεν γίνεται μια φορά αλλά για κάθε τρίγωνο στο οποίο ανήκει.

2.2 Δεύτερος αλγόριθμος

Αν πρώτα υπολογίσουμε την απόσταση των κορυφών από το επίπεδο και αναθέσουμε σε αυτά τιμές True ή False ανάλογα άμα είναι πάνω ή κάτω από το επίπεδο γλιτώνουμε αρκετό χρόνο στον υπολογισμό του αποτελέσματος. Έτσι όταν έρθει η ώρα να δούμε που βρίσκεται ένα τρίγωνο σε σχέση με το επίπεδο δεν χρειάζεται να ξαναγίνουν οι υπολογισμοί από την αρχή. Αντ' αυτού για κάθε κορυφή του τριγώνου ανατρέχουμε στον πίνακα με τα προϋπολογισμένα αποτελέσματα για κάθε κορυφή και όπως στην προηγούμενη περίπτωση συμπεραίνουμε αν το τρίγωνο βρίσκεται πάνω ή κάτω από το επίπεδο ή αν τέμνεται με αυτό.

2.3 Τρίτος αλγόριθμος

2.4 Τέταρτος αλγόριθμος

2.5 Σύγκριση χρόνων εκτέλεσης

Οι μέσοι χρόνοι εκτέλεσης για τους τέσσερεις διαφορετικούς αλγόριθμους είναι ως εξής:

1. 3.568 ms
2. 0.655 ms
3. 53.409 ms
4. 20.772 ms

Πολύ εύκολα μπορούμε να συμπεράνουμε ότι η χρήση επαναλήψεων στο κώδικα καθιστά τη διαδικασία αρκετά πιο αργή από το να αφήσουμε τη numpy να κάνει κατευθείαν τους υπολογισμούς πάνω στους πίνακες (το οποίο γίνεται με πολύ πιο αποδοτικό τρόπο). Πέρα από αυτό και στις δύο περιπτώσεις (με και χωρίς επαναλήψεις) είναι εμφανές το γεγονός ότι η προεπεξεργασία των κορυφών είναι προτιμότερη επιλογή από το να κάνουμε τους ίδιους υπολογισμούς για κάθε τρίγωνο στο οποίο εμφανίζεται μια κορυφή.

3 Divide Mesh

Η διαδικασία του Mesh-plane Intersection μας επιστρέφει τρία σύνολα τριγώνων για το μοντέλο, αυτά που τέμνονται με το επίπεδο, αυτά που βρίσκονται πάνω από το επίπεδο και αυτά που βρίσκονται κάτω από το επίπεδο. Έχοντας αυτή την πληροφορία είναι αρκετά εύκολο να δημιουργήσουμε τρία καινούρια mesh, ένα για κάθε σύνολο τριγώνων, να αναθέσουμε σε αυτά ξεχωριστά χρώματα και να μετακινήσουμε τα mesh πάνω και κάτω από την τομή. Αν στη συνέχεια διαγράψουμε το αρχικό mesh και εμφανίσουμε τα τρία καινούρια meshes, στην οθόνη θα εμφανιστεί το διαχωρισμένο μοντέλο με κενό ανάμεσα στα τρία τμήματα του. Μάλιστα επειδή αυτή η διαδικασία επαναλαμβάνεται όταν μετακινούμε το επίπεδο, η αλλαγή φαίνεται σε πραγματικό χρόνο.