ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΑΤΡΩΝ  
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ  
ΟΜΑΔΑ ΟΠΤΙΚΟΠΟΙΗΣΗΣ ΚΑΙ ΕΙΚΟΝΙΚΗΣ ΠΡΑΓΜΑΤΙΚΟΤΗΤΑΣ

**Υπολογιστική Γεωμετρία &   
Εφαρμογές 3Δ Μοντελοποίησης**

|  |  |
| --- | --- |
| **Αριθμός Απαλλακτικής εργασίας** | 6 |
| **Επώνυμο** | Λαγουτάρης |
| **Όνομα** | Βασίλειος |
| **Έτος** | 4ο |
| **ΑΜ** | 1053539 |

**ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ 6 – LAPLACIAN SHAPES**

**ΟΔΗΓΙΕΣ ΧΡΗΣΗΣ**

Οι οδηγίες βρίσκονται στο τέλος της αναφοράς.

**ΕΡΩΤΗΜΑΤΑ**

**1)**

Ξεκινώντας πρέπει να υπολογίσουμε τις διαφορικές συντεταγμένες του μοντέλου εισόδου και να τις απεικονίσουμε. Οι διαφορικές συντεταγμένες ενός αντικειμένου εκφράζουν την διαφορά κάθε κορυφής του από τον μέσο όρο των γειτονικών κορυφών. Για αυτόν τον σκοπό αρχικά ξεκινάμε με την κατασκευή του πίνακα γειτνίασης Α ο οποίος ορίζεται ως εξής:

Αυτός ο πίνακας μας δείχνει για κάθε vertex του αντικειμένου με ποια από τα υπόλοιπα vertices έχει connection.

Επίσης θα χρειαστεί να κατασκευάσουμε τον διαγώνιο πίνακα D, του οποίου όλα τα μη διαγώνια στοιχεία είναι μηδενικά, ενώ τα διαγώνια στοιχεία παίρνουν ως τιμή το πλήθος των γειτονικών κορυφών του αντίστοιχου vertex. Δηλαδή , όπου είναι το πλήθος των γειτονικών κορυφών της κορυφής .

Για την κατασκευή των πίνακα Α και D θα βασιστούμε στο γεγονός ότι τα τρίγωνα που αποθηκεύονται στα obj αρχεία περιλαμβάνουν τα indices των κορυφών τους. Έτσι αν τα indices από δύο κορυφές υπάρχουν σε ένα τρίγωνο τότε θα υπάρχει στο αντικείμενο μια ακμή που τα ενώνει. Έτσι αφού αρχικοποιήσουμε τους πίνακες με μηδενικές τιμές σε όλα τους τα στοιχεία, εξετάζουμε σε μία λούπα όλα τα τρίγωνα του Mesh. Για κάθε τρίγωνο και κάθε ζευγάρι κορυφών του ελέγχουμε πρώτα αν έχει ήδη συμπεριληφθεί η ακμή εξετάζοντας την τιμή του στοιχείου . Αν η τιμή είναι μηδέν τότε η συγκεκριμένη ακμή δεν έχει συμπεριληφθεί ακόμα και έτσι θέτουμε και . Ταυτόχρονα αυξάνουμε κατά 1 τα στοιχεία και .

Ενώ θεωρητικά η παραπάνω διαδικασία αρκεί για τον υπολογισμό των πινάκων A και D, στην πράξη πολλά αντικείμενα obj εκτός από faces με 3 κορυφές περιέχουν ή ακόμα και αποτελούνται εξ ολοκλήρου από faces με 4 κορυφές. Ο obj loader του εργαστηρίου σε αυτήν την περίπτωση δημιουργεί 2 faces των 3 κορυφών από κάθε ένα των 4. Ταυτόχρονα όμως δημιουργεί αντίγραφα των vertices με αποτέλεσμα τα meshes που φορτώνονται στο πρόγραμμα να αποτελούνται από πολλαπλάσια vertices σε σχέση με το αρχικό αντικείμενο. Το πρόβλημα είναι ότι το καθένα από αυτά τα vertices φαίνεται να μην συνδέονται με κανένα άλλο, δημιουργώντας πρόβλημα στους πίνακες A και D το οποίο παρουσιάζεται κυρίως στα επόμενα ερωτήματα.

Για την επίλυση του παραπάνω προβλήματος κάνουμε μια λούπα για κάθε vertex του Mesh και ελέγχουμε αν υπάρχει και άλλο vertex με τις ίδιες ακριβώς συντεταγμένες. Αν βρούμε κάποιο τέτοιο vertex, το οποίο φυσικά προέκυψε από την διαδικασία που εξηγήθηκε παραπάνω, τότε κάνουμε update τα αντίστοιχα στοιχεία των πινάκων A και D σαν τα vertices αυτά να ενώνονταν με μια ακμή. Επίσης σημειώνεται ότι οι πίνακες A και D αναπαρίστανται ως Sparse Matrices στο πρόγραμμα καθώς η πλειοψηφία των στοιχείων τους είναι μηδενικά.

Με βάση τους πίνακες A και D μπορούμε πλέον να υπολογίσουμε τον πίνακα , όπου ο αντίστροφος του πίνακα D προκύπτει αντιστρέφοντας το κάθε στοιχείου του καθώς είναι διαγώνιος. Έχοντας τώρα τον πίνακα μπορούμε να υπολογίσουμε τις διαφορικές συντεταγμένες του Mesh με τον εξής τύπο:

Στην συνέχεια πρέπει να γίνει η απεικόνιση των διαφορικών συντεταγμένων στις κορυφές και τα τρίγωνα του αντικειμένου. Καθώς όμως οι κορυφές στα αντικείμενα δεν είναι κατανεμημένες ομοιόμορφα στον χώρο αλλά η πυκνότητα τους αλλάζει ανάλογα με το ενδιαφέρον που παρουσιάζει η κάθε περιοχή του αντικειμένου, με την απλή απεικόνιση των διαφορικών συντεταγμένων θα ξεχωρίζουν στο αντικείμενο τα όρια μεταξύ των περιοχών υψηλής και χαμηλής δειγματοληψίας στα οποία οι κορυφές απέχουν πολύ από τον μέσο όρο των γειτονικών κορυφών. Για τον λόγο αυτό περισσότερο νόημα έχει η απεικόνιση του μέτρου των διαφορικών συντεταγμένων κάθε κορυφής στην κατεύθυνση του normal της.

Για τον υπολογισμό του normal μιας κορυφής ξεκινώντας από ένα μηδενικό διάνυσμα προσθέτουμε σε αυτό για κάθε τρίγωνο του οποίου η κορυφή αποτελεί μέλος, το normal του τριγώνου επί το εμβαδόν του. Επίσης φροντίζουμε να κανονικοποιήσουμε το normal μετά τον υπολογισμό του. Στην συνέχεια βρίσκουμε το μήκος της προβολής των διαφορικών συντεταγμένων κάθε κορυφής πάνω στο normal της. Ταυτόχρονα κρατάμε σε μια μεταβλητή το μέγιστο μέτρο αυτών των «νέων» διαφορικών συντεταγμένων που έχουμε συναντήσει καθώς θα το χρειαστούμε αργότερα.

Τέλος για την απεικόνιση του μέτρου των διαφορικών συντεταγμένων πάνω στις κορυφές του αντικειμένου χρωματίζουμε κάθε κορυφή με το κατάλληλο χρώμα. Συγκεκριμένα δίνουμε στην στο κόκκινο κανάλι την τιμή: , στο πράσινο κανάλι την τιμή: και στο μπλε κανάλι την τιμή 0.

Έτσι οι κορυφές με το μεγαλύτερο μέτρο διαφορικών συντεταγμένων στην κατεύθυνση του normal παίρνουν κόκκινο χρώμα, οι κορυφές με μικρό μέτρο διαφορικών συντεταγμένων παίρνουν πράσινο χρώμα, ενώ οι κορυφές με μέσο μέτρο διαφορικών συντεταγμένων παίρνουν κίτρινο και πορτοκαλί χρώμα. Η ίδια ακριβώς διαδικασία πραγματοποιείται και για τα τρίγωνα του αντικειμένου, τα οποία κάνουν interpolation μεταξύ των χρωμάτων των κορυφών τους. Επίσης δίνεται από το πρόγραμμα η δυνατότητα της απενεργοποίησης του χρωματισμού είτε των κορυφών, είτε των τριγώνων, είτε και των δύο. By default είναι ενεργοποιημένος μόνο ο χρωματισμός των τριγώνων.

Τα αποτελέσματα της παραπάνω διαδικασίας για κάποια από τα αντικείμενα που μας είχαν δοθεί παρουσιάζεται παρακάτω.

A close up of an animal

Description automatically generated A close up of a logo

Description automatically generated

A picture containing tree, flower

Description automatically generated A picture containing drawing

Description automatically generated

Παρατηρούμε ότι όπως περιμέναμε, κορυφές οι οποίες «εξέχουν» από την επιφάνεια του αντικειμένου, δηλαδή έχουν μεγάλη απόσταση από τον μέσο όρο των γειτόνων τους στην κατεύθυνση του normal τους, είναι χρωματισμένες με κόκκινο χρώμα.

**2)**

Στο 2ο ερώτημα αντί για τον πίνακα , χρησιμοποιούμε τον πίνακα ο οποίος αποτελεί την συμμετρική έκδοση του πίνακα . Για τον πίνακα αυτόν ισχύει:

Ο πίνακας είναι θετικά ημιορισμένος και συνεπώς εξ ορισμού έχει μη αρνητικές ιδιοτιμές. Πιο συγκεκριμένα ο πίνακας έχει πάντα μια μηδενική ιδιοτιμή, ενώ οι υπόλοιπες ιδιοτιμές του είναι θετικές.

Για να υπολογίσουμε τις ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα εκμεταλλευόμαστε το γεγονός ότι είναι συμμετρικός και χρησιμοποιούμε την κλάση SelfAdjointEigenSolver της Eigen. Εκμεταλλευόμενοι αυτήν την κλάση πετυχαίνουμε τεράστια επιτάχυνση στον υπολογισμό των ιδιοτιμών και ιδιοδιανυσμάτων, όμως ακόμα και έτσι καθώς ο υπολογισμός έχει χρονική πολυπλοκότητα τα μεγαλύτερα Meshes χρειάζονται μέχρι και 2 ώρες για το πλήρες Eigen decomposition. Για αυτόν τον λόγο αφού υπολογιστούν οι ιδιοτιμές και τα ιδιοδιανύσματα του μοντέλου εισόδου, σώζονται σε ένα αρχείο κειμένου προκειμένου να μπορούν να φορτωθούν απευθείας την επόμενη φορά χωρίς να χρειαστεί εκ νέου ο υπολογισμός τους. Μάλιστα το πρόγραμμα δίνει την δυνατότητα επιλογής ανάμεσα στον υπολογισμό των ιδιοτιμών και ιδιοδιανυσμάτων ή την φόρτωσή τους από αρχείο, δίνοντας τιμή 1 ή 0 αντίστοιχα στο define COMPUTE\_EIGENDECOMP.

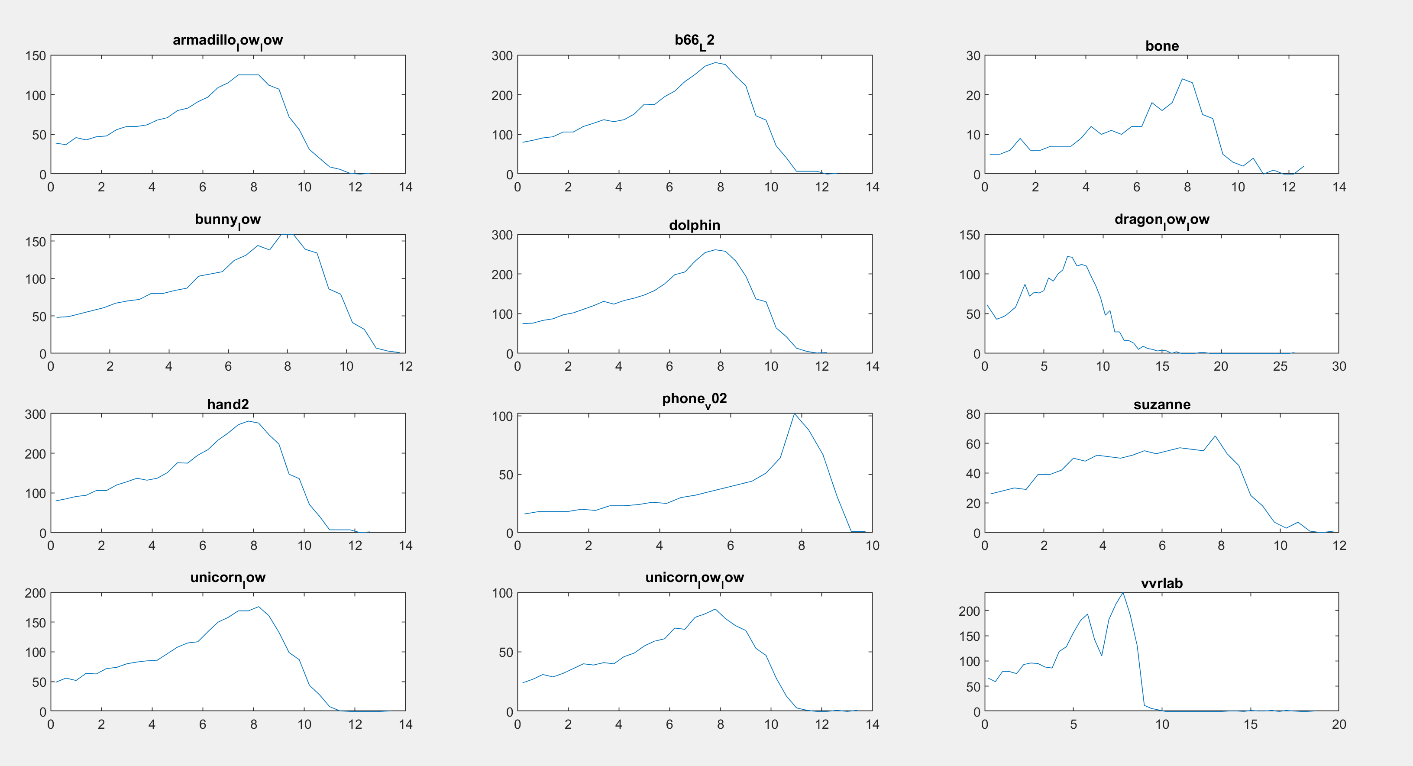
Μετά τον υπολογισμό των ιδιοτιμών και ιδιοδιανυσμάτων του πίνακα μπορούμε να τον ανακατασκευάσουμε ως εξής: , όπου είναι ο διαγώνιος πίνακας που έχει ως στοιχεία της διαγωνίου του τις ιδιοτιμές σε αύξουσα σειρά και ο είναι ο πίνακας που έχει στις στήλες του τα αντίστοιχα ιδιοδιανύσματα κάθε ιδιοτιμής.

Οι ιδιοτιμές και τα ιδιοδιανύσματα του πίνακα έχουν άμεση σχέση με τις συχνότητες που εμφανίζονται στο Mesh. Συγκεκριμένα τα πρώτα ιδιοδιανύσματα που αντιστοιχούν σε μικρές ιδιοτιμές αφορούν ομαλά μεταβαλλόμενες συναρτήσεις στο Mesh, ενώ τα τελευταία ιδιοδιανύσματα που αντιστοιχούν σε μεγάλες ιδιοτιμές αφορούν μεγάλες συχνότητες, ταχείες ταλαντώσεις στο Mesh. Επίσης η μηδενική τιμή που αναφέρθηκε πιο πάνω ότι υπάρχει πάντα στον πίνακα αφορά στην DC συνιστώσα.

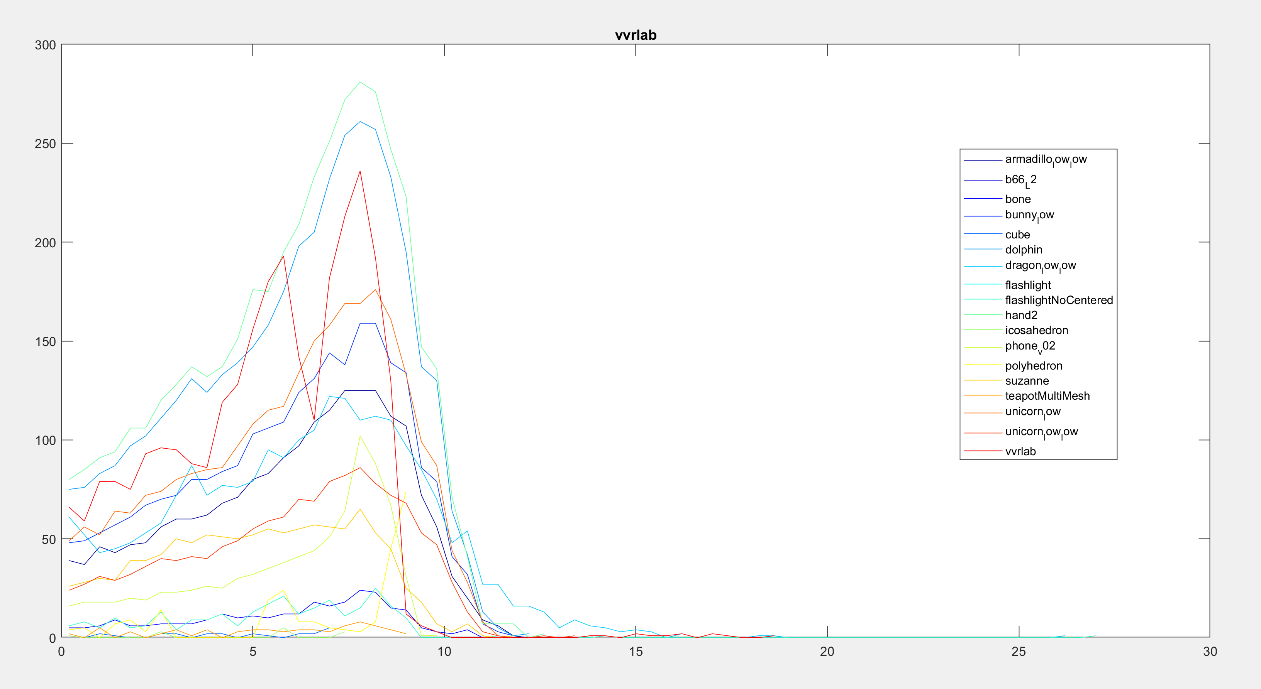
Για την απεικόνιση των ιδιοτιμών όλων των μοντέλων χρησιμοποιήθηκε η MATLAB. Αρχικά όπως περιεγράφηκε παραπάνω οι ιδιοτιμές κάθε μοντέλου αποθηκεύτηκαν σε ένα αρχείο κειμένου από το οποίο διαβάστηκαν στην συνέχεια από ένα πρόγραμμα της MATLAB. Για να απεικονίσουμε τις τιμές των ιδιοτιμών, οι οποίες είναι μέχρι και 4000 για κάποια μοντέλα, με έναν τρόπο ο οποίος θα διευκολύνει την σύγκριση μεταξύ των ιδιοτιμών διαφορετικών μοντέλων ακολουθήσαμε την παρακάτω διαδικασία.

Χωρίζουμε τον άξονα x του γραφήματος σε διαστήματα μήκους 0.4 (αυθαίρετη επιλογή) ξεκινώντας από το μηδέν μέχρι την τιμή της μεγαλύτερης ιδιοτιμής. Για κάθε ένα από αυτά τα διαστήματα μετράμε πόσες από τις ιδιοτιμές κάθε μοντέλου πέφτουν σε αυτό. Έτσι δημιουργείται ένα γράφημα το οποίο έχει στον άξονα x τις τιμές των ιδιοτιμών και στον άξονα y το πλήθος των ιδιοτιμών που έχουν τιμή μέσα στο αντίστοιχο διάστημα του άξονα x.

Παρακάτω παρουσιάζονται τα γραφήματα των ιδιοτιμών για κάποια από τα μοντέλα που μας δόθηκαν, ενώ στο τέλος παρουσιάζονται οι ιδιοτιμές όλων των μοντέλων σε κοινό γράφημα.



Εδώ παρατηρούμε ότι παρόμοια μοντέλα που μοιάζουν μεταξύ τους, έχουν πολύ παρόμοια μορφή στα γραφήματα ιδιοτιμών τους. Για παράδειγμα τα μοντέλα “unicorn\_low” και “unicorn\_low\_low” ή “hand2” και “b66\_L2”. Αυτή η παρατήρηση μάλιστα οδήγησε στην χρήση μίας παρόμοιας μεθόδου με αυτή που δημιούργησε τα γραφήματα για το πρόβλημα του shape indexing του ερωτήματος 5.



Εδώ δεν παρατηρείται μεγάλη ομοιότητα μεταξύ των γραφημάτων διαφορετικών μοντέλων καθώς ακόμα και σε μοντέλα που μοιάζουν μεταξύ τους, ο αριθμός των ιδιοτιμών μπορεί να είναι εντελώς διαφορετικός και έτσι να έχουν παραμορφωθεί στον άξονα y.

**3)**

Για την ανακατασκευή των σχημάτων από τις διαφορικές συντεταγμένες απαιτείται φυσικά η αντιστροφή του πίνακα . Όμως η τάξη του πίνακα δεν ισούται με τις διαστάσεις του και συνεπώς η αντιστροφή του είναι αδύνατη. Συγκεκριμένα ισχύει ότι , όπου ο αριθμός των vertices του Mesh και ο αριθμός των connected components του Mesh, δηλαδή για πλήρως συνδεδεμένα Meshes. Έτσι για την ανακατασκευή των σχημάτων θα πρέπει να χρησιμοποιήσουμε anchor points.

Επιπλέον για την ανακατασκευή δεν θα χρησιμοποιήσουμε όλα τα ιδιοδιανύσματα παρά μόνο ένα ποσοστό τους. Έτσι θα πρέπει να κάνουμε decomposition των συντεταγμένων των κορυφών στην βάση που ορίζεται από τον πίνακα των ιδιοδιανυσμάτων. Δηλαδή , και . Στην συνέχεια κάνουμε approximate τις συντεταγμένες του μοντέλου χρησιμοποιώντας μόνο ένα ποσοστό των συντελεστών και των ιδιοδιανυσμάτων. Καθώς το μεγαλύτερο ποσοστό της ενέργειας είναι αποθηκευμένο στους συντελεστές χαμηλής συχνότητας, δηλαδή στα πρώτα μπορούμε να πετύχουμε υψηλό compression.

Έχοντας υπολογίσει μια προσέγγιση των συντεταγμένων των vertices με χρήση ενός ποσοστού των ιδιοδιανυσμάτων, υπολογίζουμε τις νέες διαφορικές συντεταγμένες αυτού του νέου Mesh ως και ξεκινάμε την ανακατασκευή. Για να έχει λύση το σύστημα πρέπει να αυξήσουμε την τάξη του πίνακα προσθέτοντας νέες στήλες σε αυτόν και κατάλληλα anchor points στον πίνακα των διαφορικών συντεταγμένων.

Όπου είναι τα βάρη που ορίζουν την σημαντικότητα των χωρικών περιορισμών και τα anchor points που χρησιμοποιούνται για την ανακατασκευή. Εμείς χρησιμοποιήσαμε βάρη ίσα με την μονάδα, ενώ τα anchor points επιλέγονται τυχαία μεταξύ του συνόλου των vertices. Με μια επαναληπτική μέθοδο αυξάνουμε τον αριθμό γραμμών του πίνακα , προσθέτοντας μια γραμμή με όλα τα στοιχεία μηδενικά εκτός από το index του vertex που θα χρησιμοποιηθεί ως anchor point το οποίο επιλέγεται τυχαία χωρίς επαναλήψεις. Ταυτόχρονα προσθέτουμε και μια νέα γραμμή στον πίνακα με τις συντεταγμένες του νέου anchor point. Αν έχουμε δώσει την τιμή 1 στο DYNAMIC\_RECONSTRUCTION, αυτή η διαδικασία επαναλαμβάνεται για όσα anchor points είναι απαραίτητα για την επίλυση του συστήματος τα οποία για κάποια Meshes είναι πάνω από 1 καθώς είτε τα ίδια τα Meshes δεν είναι πλήρως συνδεδεμένα είτε το σύστημα δεν έχει λύση λόγω λαθών ακρίβειας από τον υπολογιστή. Διαφορετικά δίνοντας κατάλληλη τιμή στο define CONTROL\_POINTS θέτουμε εμείς τον αριθμό anchor points που θέλουμε να χρησιμοποιηθούν κατά την ανακατασκευή του μοντέλου. Φυσικά όπως θα δούμε και παρακάτω με περισσότερα anchor points έχουμε πιο ακριβή ανακατασκευή του μοντέλου.

Αφού σχηματίσουμε τους πίνακες και είμαστε σε θέση να λύσουμε το least-squares σύστημα. Η λύση του ανάγεται στην λύση του προβλήματος βελτιστοποίησης:

,

ενώ η αναλυτική λύση του είναι:

, όπου είναι ο extended πίνακας

Στην περίπτωση μας καθώς ο πίνακας είναι πολύ αραιός χρησιμοποιήσαμε την κλάση SimplicialLLT της Eigen η οποία λύνει το πρόβλημα κατασκευάζοντας το Cholesky decomposition του πίνακα .

Όλα τα παραπάνω εφαρμόστηκαν 4 φορές συνολικά χρησιμοποιώντας το 30%, το 10%, το 5% και το 1% των ιδιοδιανυσμάτων αντίστοιχα. Ενώ η χρήση της κλάσης SimplicialLLT σε Sparse matrices μας δίνει τεράστια επιτάχυνση σε σχέση με τις μεθόδους οι οποίες δρουν σε Dense matrices, λόγω του μεγάλου αριθμού anchor points που χρειάζονται για την ανακατασκευή κάποιων μοντέλων ο υπολογισμός της ανακατασκευής αυτών των μοντέλων είναι σχετικά μεγάλος (~5 λεπτά). Για αυτόν τον λόγο, όπως και στο eigen decomposition δίνεται η δυνατότητα φόρτωσης των συντεταγμένων των ανακατασκευασμένων μοντέλων από αρχείο θέτοντας την τιμή 0 στο define COMPUTE\_COORDS.

Παρακάτω παρουσιάζονται τα ανακατασκευασμένα Meshes κάποιων μοντέλων και συγκρίνονται με τα original μοντέλα. Επίσης παρουσιάζεται ο ρόλος που έχει στην ποιότητα ανακατασκευής η χρήση μεγαλύτερου αριθμού anchor points.

Original model unicorn\_low\_low

A picture containing drawing

Description automatically generated

30% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

10% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

5% των ιδιοδιανυσμάτων

A close up of an animal

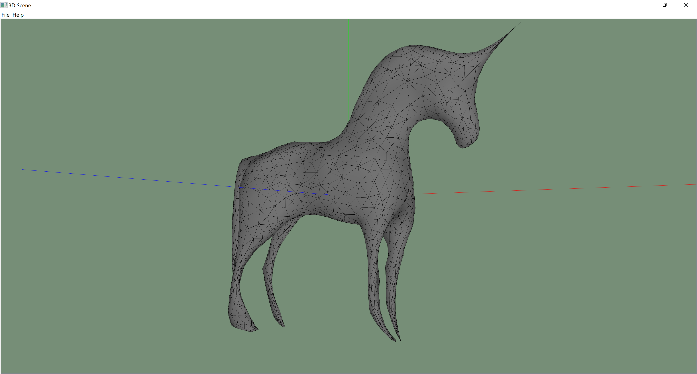
Description automatically generated

1% των ιδιοδιανυσμάτων

A picture containing ax, flying

Description automatically generated

Μέχρι και την ανακατασκευή με το 10% των ιδιοδιανυσμάτων είναι σχεδόν αδύνατο να παρατηρηθεί διαφορά σε σχέση με το αρχικό μοντέλο, όταν βέβαια φτάνουμε στην χρήση του 1% των ιδιοδιανυσμάτων το μοντέλο έχει υποστεί τεράστια αλλοίωση. Αξίζει να σημειωθεί πάντως ότι στην τελευταία εικόνα χρησιμοποιούνται μόλις 13 ιδιοδιανύσματα από τα 1360 συνολικά και η γενική μορφή του μονόκερου παραμένει. Στην παραπάνω ανακατασκευή έγινε χρήση μόλις ενός anchor point κάθε φορά, δηλαδή του ελάχιστου απαραίτητου αριθμού για την λύση του συστήματος. Παρακάτω παρουσιάζεται η ανακατασκευή του ίδιου μοντέλου με χρήση του 1% των ιδιοδιανυσμάτων αλλά αυτήν την φορά 100 τυχαία anchor points από τα 1360 vertices συνολικά.



Όπως φαίνεται υπάρχει μεγάλη βελτίωση σε σχέση με την χρήση 1 μόνο anchor point καθώς το ανακατασκευασμένο αντικείμενο είναι σχεδόν ίδιο με αυτό που έχει ανακατασκευαστεί με την χρήση του 5% των ιδιοδιανυσμάτων. Η ίδια ακριβώς διαδικασία εφαρμόστηκε και στο μοντέλο του χεριού με παρόμοια αποτελέσματα.

Original model hand2

A close up of an animal

Description automatically generated

30% των ιδιοδιανυσμάτων

A close up of a logo

Description automatically generated

10% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

5% των ιδιοδιανυσμάτων

A close up of a logo

Description automatically generated

1% των ιδιοδιανυσμάτων

A close up of a logo

Description automatically generated

Εδώ καθώς ο αριθμός των vertices είναι μεγαλύτερος (4323) ακόμα και η ανακατασκευή με το 5% των ιδιοδιανυσμάτων και 1 anchor point δεν φαίνεται καθόλου διαφορετική στο μάτι σε σχέση με το αρχικό μοντέλο.

**4)**

Μετά την ανακατασκευή του μοντέλου με χρήση ενός ποσοστού από τις ιδιοτιμές μπορούμε εύκολα να υπολογίσουμε το σφάλμα που προκύπτει κατά την ανακατασκευή σε κάθε περίπτωση. Αρχικά ασχολούμαστε με το σφάλμα κάθε vertex ως προς την θέση του στον χώρο και στην συνέχεια με το σφάλμα στις διαφορικές συντεταγμένες κάθε vertex.

Για τον υπολογισμό του σφάλματος στις συντεταγμένες κάθε σημείου απλώς υπολογίζουμε μέσα σε μία λούπα για κάθε vertex την απόστασή του από το αντίστοιχο vertex του αρχικού μοντέλου. Ταυτόχρονα κρατάμε την μέγιστη απόσταση που έχουμε συναντήσει μέχρι εκείνη την στιγμή καθώς θα την χρησιμοποιήσουμε στο τέλος για να κανονικοποιήσουμε τα αποτελέσματα. Όσον αφορά την απεικόνιση του σφάλματος στο μοντέλο, ακολουθούμε ακριβώς την ίδια διαδικασία με αυτήν του ερωτήματος 1 σύμφωνα με την οποία κάθε vertex ή τρίγωνο λαμβάνει χρώμα κοντά στο κόκκινο για μεγάλα σφάλματα και χρώματα κοντά στο πράσινο για μικρά.

Original model dolphin

A close up of an animal

Description automatically generated

30% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

10% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

5% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

1% των ιδιοδιανυσμάτων

A picture containing green, colorful

Description automatically generated

Παρατηρούμε ότι το μεγαλύτερο σφάλμα στις καρτεσιανές συντεταγμένες εμφανίζεται σε περιοχές μικρής δειγματοληψίας, δηλαδή σε περιοχές του Mesh όπου τα vertices είναι αραιά.

Original model armadillo\_low\_low

A picture containing drawing

Description automatically generated

30% των ιδιοδιανυσμάτων

A picture containing map, drawing

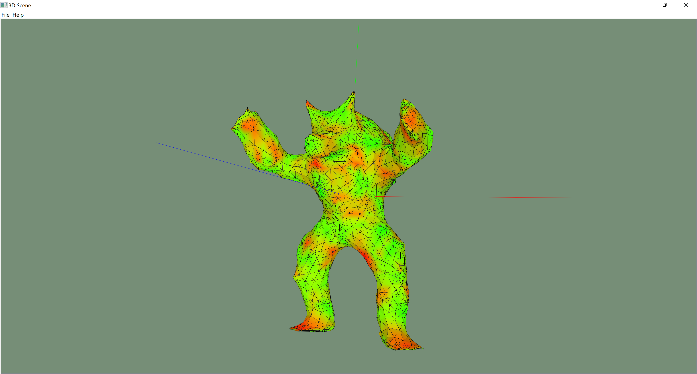
Description automatically generated

10% των ιδιοδιανυσμάτων

A picture containing drawing

Description automatically generated

5% των ιδιοδιανυσμάτων



1% των ιδιοδιανυσμάτων

A picture containing map, flying

Description automatically generated

Εδώ τα vertices είναι ομοιόμορφα κατανεμημένα στην επιφάνεια του Mesh και έτσι και το σφάλμα στις καρτεσιανές συντεταγμένες κατά την ανακατασκευή δεν είναι συγκεντρωμένο σε ένα του σημείο του Mesh όπως στο δελφίνι, αλλά εμφανίζεται σε αρκετά σημεία του αντικειμένου.

Στην συνέχεια θέλουμε να απεικονίσουμε το σφάλμα στις διαφορικές συντεταγμένες κάθε κορυφής, οπότε ξεκινάμε υπολογίζοντας τις διαφορικές συντεταγμένες του ανακατασκευασμένου αντικειμένου πολλαπλασιάζοντας τις συντεταγμένες του με τον πίνακα *.* Αντίστοιχα με το σφάλμα στις καρτεσιανές συντεταγμένες μπορούμε να υπολογίσουμε το μέτρο της διαφοράς στις διαφορικές συντεταγμένες κάθε vertex σε σχέση με το αρχικό μοντέλο πριν την ανακατασκευή. Επίσης κανονικοποιούμε πάλι το μέτρο της διαφοράς έτσι ώστε να έχει τιμή από το 0 μέχρι το 1. Έτσι μπορούμε πολύ εύκολα να ζωγραφίσουμε τις κορυφές και τα τρίγωνα του μοντέλου ώστε να έχουν κόκκινο χρώμα σε περιοχές με μεγάλο σφάλμα και μπλε χρώμα σε περιοχές με μικρό σφάλμα ανακατασκευής. Τα αποτελέσματα της παραπάνω διαδικασίας για τα ίδια μοντέλα που χρησιμοποιήθηκαν και παραπάνω παρουσιάζονται στις παρακάτω φωτογραφίες.

30% των ιδιοδιανυσμάτων

A close up of a fish

Description automatically generated

10% των ιδιοδιανυσμάτων

A close up of a fish

Description automatically generated

5% των ιδιοδιανυσμάτων

A close up of a fish

Description automatically generated

1% των ιδιοδιανυσμάτων

A close up of a fish

Description automatically generated

30% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

10% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

5% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

1% των ιδιοδιανυσμάτων

A close up of an animal

Description automatically generated

Το σφάλμα ανακατασκευής στις διαφορικές συντεταγμένες φαίνεται να εμφανίζεται κατά κύριο λόγο στα ίδια σημεία που εμφανίζεται και το σφάλμα στις καρτεσιανές συντεταγμένες, χωρίς όμως αυτό να είναι απόλυτο.

*Σημείωση*: Σε κάποια από τα μοντέλα δημιουργείται πρόβλημα κατά την απεικόνιση των χρωματισμένων τριγώνων στο ερώτημα 4. Αυτό λογικά συμβαίνει καθώς δημιουργούνται κατά την ανακατασκευή κάποιων αντικειμένων τρίγωνα με μηδενικό εμβαδόν (π.χ. με παράλληλες πλευρές) τα οποία το VVRFRAMEWORK δεν μπορεί να απεικονίσει. Σε αυτήν την περίπτωση με νέα εκτέλεση του προγράμματος μπορεί να απενεργοποιηθεί ο χρωματισμός των τριγώνων με το πλήκτρο T και να ενεργοποιηθεί ο χρωματισμός κορυφών με το πλήκτρο P.

**5)**

Η πρώτη σκέψη για εύρεση μετρικής η οποία να περιγράφει το σχήμα κάθε αντικειμένου στο πεδίο της συχνότητας ξεκίνησε από την μέθοδο που χρησιμοποιήθηκε στο ερώτημα 2 για την απεικόνιση των ιδιοτιμών του πίνακα . Ξέρουμε ότι οι ιδιοτιμές αυτού του πίνακα αντιστοιχούν στις συχνότητες του Mesh και έτσι μοιάζει λογικό να μπορούν να χρησιμοποιηθούν για την περιγραφή των αντικειμένων στο πεδίο της συχνότητας. Το πρόβλημα όμως είναι ότι το πλήθος των ιδιοτιμών του πίνακα ισούται με τον αριθμό των vertices του κάθε μοντέλου και έτσι δεν υπάρχει τρόπος να συγκρίνουμε τις ιδιοτιμές 2 μοντέλων μια προς μια.

Για την επίλυση του παραπάνω προβλήματος ξεκινάμε με την ίδια μεθοδολογία που είχαμε χρησιμοποιήσει στο ερώτημα 2. Χωρίζουμε δηλαδή τις τιμές των ιδιοτιμών σε διαστήματα σταθερού μήκους και υπολογίζουμε πόσες από τις ιδιοτιμές κάθε μοντέλου ανήκουν σε κάθε διάστημα. Στην συνέχεια για να κανονικοποιήσουμε αυτήν την πληροφορία και να την κάνουμε ανεξάρτητη από το πλήθος των ιδιοτιμών διαιρούμε το πλήθος των ιδιοτιμών του κάθε διαστήματος με το πλήθος των ιδιοτιμών του πολυπληθέστερου διαστήματος. Έτσι σε κάθε διάστημα αντιστοιχεί για κάθε μοντέλο μια τιμή από 0 έως 1 και μπορούμε εύκολα να βρούμε την σχετική απόσταση μεταξύ δύο μοντέλων όπως θα αναφερθεί στο επόμενο ερώτημα.

Στην συνέχεια και μετά από αναζήτηση για το πρόβλημα του shape indexing στην βιβλιογραφία υλοποιήθηκε και μία επιπλέον μέθοδος με δύο διαφορετικές παραλλαγές η οποία παρουσιάζεται παρακάτω.

Η δεύτερη μέθοδος όπως και η πρώτη βασίζεται στην εύρεση των ιδιοτιμών κάποιου πίνακα ο οποίος στο στοιχείο περιέχει πληροφορία που αφορά τις κορυφές και . Ο πίνακας γειτνίασης στο στοιχείο κωδικοποιεί με 1 ή 0 την ύπαρξη ή απουσία ακμής η οποία συνδέει τα αντίστοιχα vertices. Αντίστοιχα ο νέος πίνακας που θα χρησιμοποιήσουμε αποθηκεύει ως τιμή του στοιχείου την απόσταση μεταξύ των κορυφών και . Η απόσταση αυτή μπορεί να είναι η ευκλείδεια απόσταση ή η γεωδεσιακή απόσταση μεταξύ των δύο κορυφών. Φυσικά το πρόβλημα της ασυμφωνίας μεταξύ του πλήθους των vertices διαφορετικών μοντέλων παραμένει και επιπλέον η εύρεση των ιδιοτιμών για πίνακες εκατομμυρίων στοιχείων που απαιτείται είναι χρονοβόρα. Για τον λόγο αυτό εκμεταλλευόμενοι την Nyström approximation μπορούμε να εκτιμήσουμε τις ιδιοτιμές του πίνακα με ένα πολύ μικρότερο υποπίνακα και έτσι δεν περιλαμβάνουμε σε αυτόν τον νέο πίνακα όλα τα vertices του κάθε μοντέλου παρά μόνο χρησιμοποιούμε έναν σταθερό αριθμό vertices για κάθε μοντέλο. Ο αριθμός αυτός μπορεί να επιλεγεί δίνοντας την κατάλληλη τιμή στο define SAMPLE\_POINTS και όπως είναι λογικό όσο μεγαλύτερος είναι αυτός ο αριθμός τόσο καλύτερη αναπαράσταση του Mesh στο πεδίο της συχνότητας θα πετύχουμε. Η μέθοδος αρχίζει να δίνει πολύ καλά αποτελέσματα για μόλις 15-20 vertices από τα χιλιάδες συνολικά που περιέχουν τα περισσότερα μοντέλα, ενώ ως default τιμή επιλέχθηκε το 50.

Επιπλέον μεγάλη σημασία έχει η επιλογή των vertices αυτών από τα vertices κάθε αντικειμένου. Εδώ επιλέγουμε το πρώτο vertex τυχαία από το Mesh και στην συνέχεια κάνοντας μια λούπα κάθε φορά που θέλουμε να επιλέξουμε ένα νέο vertex επιλέγουμε εκείνο που έχει την μεγαλύτερη ελάχιστη απόσταση από τα ήδη επιλεγμένα vertices. Δηλαδή ψάχνουμε σε κάθε βήμα το

,όπου τα ήδη επιλεγμένα vertices.

Με αυτόν τον τρόπο εξασφαλίζουμε ότι τα vertices που επιλέξαμε βρίσκονται κατανεμημένα ομοιόμορφα σε όλες τις περιοχές του Mesh και δεν είναι συγκεντρωμένα γύρω από ένα σημείο. Αφού επιλεγούν όλα τα vertices που θα χρησιμοποιήσουμε γεμίζουμε τον πίνακα αποστάσεων με τις αποστάσεις μεταξύ τους. Επιπλέον κανονικοποιούμε τα στοιχεία του διαιρώντας τα με την μέγιστη απόσταση μεταξύ των vertices του Mesh. Τέλος υπολογίζουμε τις ιδιοτιμές του πίνακα των αποστάσεων και από αυτές έχουμε την περιγραφή του σχήματος των αντικειμένων στο πεδίο της συχνότητας.

Την παραπάνω διαδικασία την επαναλαμβάνουμε δύο φορές συνολικά, μία με χρήση της ευκλείδειας απόστασης και μια με την χρήση της γεωδεσιακής απόστασης μεταξύ των vertices. Για την εύρεση της γεωδεσιακής απόστασης μεταξύ δύο vertices πρέπει να υπολογίσουμε το shortest path μεταξύ τους πάνω στον γράφο που σχηματίζεται από το Mesh. Και καθώς σε κάθε βήμα χρειαζόμαστε τις αποστάσεις από κάποια συγκεκριμένα vertices προς όλα τα άλλα για την επιλογή του επόμενου vertex, υλοποιούμε τον αλγόριθμο του Dijkstra και βρίσκουμε το vertex με την μεγαλύτερη ελάχιστη απόσταση προς τα ήδη επιλεγμένα vertices.

**6)**

Στο τελευταίο ερώτημα θέλουμε να χρησιμοποιήσουμε τις μεθόδους που αναπτύξαμε στο ερώτημα 5 για να λύσουμε το πρόβλημα του shape indexing. Δηλαδή θέλουμε δοσμένου ενός νέου αντικειμένου εισόδου να κατατάξουμε τα αντικείμενα μιας βάσης δεδομένων ως προς φθίνουσα ομοιότητα με τον αντικείμενο εισόδου. Στην περίπτωση μας τα αντικείμενα της βάσης δεδομένων είναι τα παρακάτω obj αρχεία.

1. armadillo\_low\_low
2. b66\_L2
3. bone
4. bunny\_low
5. cube
6. dolphin
7. dolphin
8. dragon\_low\_low
9. flashlight
10. flashlightNoCentered
11. hand2
12. icosahedron
13. phone\_v02
14. polyhedron
15. Suzanne
16. unicorn\_low
17. unicorn\_low\_low
18. vvrlab

Ξεκινώντας με την μέθοδο των ιδιοτιμών του πίνακα διαβάζουμε από τα αρχεία τις ιδιοτιμές των αντικειμένων που αναφέρθηκαν παραπάνω καθώς και του αντικειμένου εισόδου. Άρα λοιπόν για να εφαρμοστεί αυτή η μέθοδος σε ένα νέο αντικείμενο πρέπει πρώτα να έχουμε υπολογίσει το eigen decomposition του πίνακα του αντικειμένου πράγμα που είναι ιδιαίτερα χρονοβόρο. Στην συνέχεια υλοποιούμε την διαδικασία που αναφέρθηκε στο ερώτημα 5 ώστε να υπολογίσουμε το πλήθος των ιδιοτιμών του κάθε αντικειμένου που ανήκουν σε κάθε διάστημα και να κανονικοποιήσουμε τα αποτελέσματα. Τέλος για να βρούμε την σχετική απόσταση καθενός αντικειμένου με το αντικείμενο εισόδου, απλώς υπολογίζομε το άθροισμα των γινομένων των διαφορών μεταξύ των αντίστοιχων τιμών των διαστημάτων τους. Δηλαδή

,

όπου είναι το διάνυσμα που περιέχει το κανονικοποιήμενο πλήθος των ιδιοτιμών του αρχικού αντικειμένου για κάθε διάστημα και το διάνυσμα που περιέχει την ίδια πληροφορία για το αντικείμενο .

Αφού υπολογίσουμε την σχετική απόσταση κάθε αντικειμένου από το αντικείμενο εισόδου, κατατάσσουμε τα αντικείμενα με αύξουσα σειρά με βάση την σχετική τους απόσταση από το αντικείμενο εισόδου με την χρήση του αλγορίθμου bubble sort. Τα αποτελέσματα παρουσιάζονται στο τέλος μαζί με σύγκριση των διαφορετικών μεθόδων μεταξύ τους.

Στην δεύτερη μέθοδο όπως εξηγήθηκε δειγματοληπτούμε τα vertices του αντικειμένου και στην συνέχεια υπολογίζουμε τις ιδιοτιμές ενός πίνακα που περιέχει τις αποστάσεις μεταξύ των δειγματοληπτημένων vertices. Εδώ λόγω του μικρού και σταθερού μεγέθους του πίνακα των αποστάσεων, του οποίου θέλουμε να υπολογίσουμε τις ιδιοτιμές, η διαδικασία του eigen decomposition δεν είναι ιδιαίτερα χρονοβόρα. Όμως ο υπολογισμός των γεωδεσιακών αποστάσεων μεταξύ των κορυφών του αντικειμένου απαιτεί την εύρεση της συντομότερης διαδρομής πάνω σε γράφο και έτσι η χρονική πολυπλοκότητα του αλγορίθμου είναι πολύ μεγάλη. Για αυτόν τον λόγο γίνεται μόνο στην περίπτωση των γεωδεσιακών αποστάσεων προϋπολογισμός των ιδιοτιμών του πίνακα των αποστάσεων για όλα τα αντικείμενα σύγκρισης και αποθήκευσης τους σε αρχείο κειμένου. Έτσι αφήνεται για το run time μόνο ο υπολογισμός των ιδιοτιμών του αντικειμένου εισόδου.

Αφού έχουμε λοιπόν όλες τις ιδιοτιμές για όλα τα αντικείμενα της βάσης δεδομένων καθώς και για το αντικείμενο εισόδου μπορούμε να υπολογίσουμε την σχετική απόσταση μεταξύ δύο αντικειμένων με την χρήση του εξής τύπου που εφαρμόζεται και για τις ευκλείδειες αλλά και για τις γεωδεσιακές αποστάσεις.

Στην συνέχεια απλώς κατατάσσουμε τα αντικείμενα με αύξουσα σειρά ως προς την απόστασή τους με το αντικείμενο εισόδου, δηλαδή τα κατατάσσουμε με φθίνουσα ομοιότητα.

Για να ελέγξουμε την αποτελεσματικότητα των παραπάνω μεθόδων τι εξετάσαμε σε κάποια από τα obj αρχεία που μας είχαν δοθεί. Συγκεκριμένα τα καταλληλότερα αρχεία για αυτόν τον σκοπό είναι το unicorn\_low\_low που μοιάζει πολύ με το unicorn\_low αλλά έχει μικρότερο αριθμό vertices, το hand2 που είναι το ίδιο με το b66\_L2 και το flashlight που είναι ίδιο με το flashlightNoCentered απλώς μετατοπισμένο στον χώρο. Παρακάτω παρουσιάζονται τα αποτελέσματα για καθεμία από τις μεθόδους.

unicorn\_low\_low

A close up of text on a black background

Description automatically generated A close up of text on a black background

Description automatically generated A close up of text on a black background

Description automatically generated

Παρατηρούμε ότι όλες οι μέθοδοι κατηγοριοποιούν το αντικείμενο unicorn\_low\_low ως αυτό με την μικρότερη απόσταση με τον εαυτό του. Βέβαια στην πρώτη περίπτωση έχει μηδενικό error ενώ στις υπόλοιπες όχι αφού η επιλογή των sample vertices είναι μη ντετερμινιστική. Το αντικείμενο unicorn\_low\_low κατηγοροποιείται σωστά ως 2ο σε ομοιότητα στην δεύτερη και τρίτη περίπτωση και 3ο στην πρώτη περίπτωση. Επιπλέον η σχετική απόσταση του αντικειμένου unicorn\_low\_low από τα αντικειμένα hand2 και b66\_L2 που έχουν τα ίδια vertices είναι σχεδόν η ίδια και στις τρεις περιπτώσεις.

flashlight

A close up of text on a black background

Description automatically generated A screenshot of text

Description automatically generated A close up of text on a black background

Description automatically generated

Εδώ παρατηρούμε ότι και οι 3 μέθοδοι κατέταξαν τα αντικείμενα flashlight και flashlightNoCentered πολύ κοντά μεταξύ τους με σχεδόν μηδενική σχετική απόσταση. Επιπλέον τα unicorn\_low και unicorn\_low\_low όπως και τα hand2 και b66\_L2 είναι και πάλι πολύ κοντά μεταξύ τους.

hand2

A screenshot of a cell phone screen with text

Description automatically generated A close up of text on a black background

Description automatically generated A close up of text on a black background

Description automatically generated

Και στην τελευταία δοκιμή οι 3 μέθοδοι φαίνεται να βρίσκουν σωστά πολύ χαμηλή σχετική απόσταση μεταξύ των αντικειμένων hand2 και b66\_L2, ενώ όσον αφορά την απόσταση του αντικειμένου εισόδου από τα αντικείμενα unicorn\_low και unicorn\_low\_low η δεύτερη και η τρίτη μέθοδος σωστά δίνουν παρόμοια τιμή με την πρώτη μέθοδο να μην το πετυχαίνει.

Συνολικά το πρόβλημα του shape indexing λύθηκε με 2 διαφορετικούς τρόπους και μια επιπλέον παραλλαγή στον δεύτερο, ενώ τα αποτελέσματα που προκύπτουν από τον περιορισμένο έλεγχο που μπορούμε να κάνουμε σε αυτό το πρόβλημα είναι καλύτερα για την δεύτερη μέθοδο και την παραλλαγή της σε σχέση με την πρώτη.

Οδηγίες χρήσης:

* Παράμετροι προγράμματος (DEFINE):

FILENAME -> το όνομα του obj αρχείου που θα φορτωθεί χωρίς την κατάληξη “.obj”. Το αρχείο πρέπει να βρίσκεται στον φάκελο «/resources/obj».

CONTROL\_POINTS -> ο αριθμός των anchor points που θα χρησιμοποιηθούν κατά την ανακατασκευή του μοντέλου. Default τιμή είναι το 1

DYNAMIC\_RECONSTRUCTION -> αν πάρει την τιμή 1 τότε ο αριθμός των anchor points που θα χρησιμοποιηθούν για την ανακατασκευή του μοντέλου δεν είναι σταθερός αλλά επιλέγεται με βάση την ορίζουσα του πίνακα . Η χρήση του προορίζεται για την σωστή ανακατασκευή κάποιων μοντέλων (π.χ. flashlight και vvrlab) τα οποία δεν είναι πλήρως συνδεδεμένα και έτσι χρειάζονται μεγαλύτερο αριθμό από anchor points. Δεν είναι enabled by default γιατί υπερεκτιμά τον αριθμό των anchor points που απαιτούνται με αποτέλεσμα να καθυστερεί τους υπολογισμούς.

SAMPLE\_POINTS -> ο αριθμός των δειγματοληπτημένων vertices που θα χρησιμοποιηθούν για την περιγραφή των σχημάτων των μοντέλων στο ερώτημα 5. Default τιμή το 50.

SECTION\_SIZE -> το πλάτος των διαστημάτων στα οποία μετράμε το πλήθος των ιδιοτιμών κάθε μοντέλου στο ερώτημα 5. Default τιμή 0.5.

COMPUTE\_EIGENDECOMP -> δίνουμε τιμή 1 όταν θέλουμε να υπολογιστεί το eigen decomposition του πίνακα Ls του μοντέλου εισόδου και 0 αν θέλουμε να διαβαστεί από αρχείο (εφόσον φυσικά υπάρχει ήδη το κατάλληλο αρχείο).

COMPUTE\_COORDS -> δίνουμε τιμή 1 όταν θέλουμε να υπολογιστούν οι συντεταγμένες των vertices του ανακατασκευασμένου μοντέλου και 0 αν θέλουμε να διαβαστούν από αρχείο (εφόσον φυσικά υπάρχει ήδη το κατάλληλο αρχείο).

COMPUTE\_GEODESIC -> δίνουμε τιμή 1 όταν θέλουμε να υπολογιστούν οι ιδιοτιμές του πίνακα των γεωδεσιακών αποστάσεων μεταξύ των δειγματοληπτημένων vertices στο ερώτημα 5 και 0 αν θέλουμε να διαβαστούν από αρχείο (εφόσον φυσικά υπάρχει ήδη το κατάλληλο αρχείο). Αυτό αφορά μόνο την περίπτωση κατά την οποία το SAMPLE\_POINTS έχει την τιμή 50 καθώς μόνο σε αυτήν την περίπτωση έχουν προϋπολογιστεί οι ιδιοτιμές. Συνεπώς αν αλλάξει η τιμή του SAMPLE\_POINTS πρέπει το COMPUTE\_GEODESIC να λάβει την τιμή 1.

* Λειτουργίες πληκτρολογίου:

Με το πλήκτρο ‘T’ ενεργοποιούμε ή απενεργοποιούμε το ζωγράφισμα των τριγώνων του μοντέλου.

Με το πλήκτρο ‘P’ ενεργοποιούμε ή απενεργοποιούμε το ζωγράφισμα των κορυφών του μοντέλου.

Με το πλήκτρο ‘O’ αλλάζουμε ανάμεσα στο original μοντέλο και το ανακατασκευασμένο μοντέλο.

Με το πλήκτρο ‘V’ αλλάζουμε ανάμεσα στα ανακατασκευασμένα μοντέλα που έχουν δημιουργηθεί χρησιμοποιώντας διαφορετικό ποσοστό ιδιοδιανυσμάτων κάθε φορά (πρέπει πρώτα να έχουμε πατήσει το ‘O’ για να βλέπουμε τα ανακατασκευασμένα μοντέλα έναντι του αρχικού).

Με το πλήκτρο ‘D’ αλλάζουμε ανάμεσα στην απεικόνιση του λάθους στις καρτεσιανές ή στις διαφορικές συντεταγμένες (ερώτημα 4) στα vertices ή και τα τρίγωνα του μοντέλου.

Τέλος μετά από κάθε πάτημα κάθε πλήκτρου εμφανίζεται στην κονσόλα κατάλληλο μήνυμα για το τι απεικονίζεται.