



ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΑΤΡΩΝ
UNIVERSITY OF PATRAS

ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ
ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΤΕΧΝΟΛΟΓΙΑΣ ΥΠΟΛΟΓΙΣΤΩΝ

3Δ ΥΠΟΛΟΓΙΣΤΙΚΗ ΓΕΩΜΕΤΡΙΑ ΚΑΙ ΟΡΑΣΗ
ΑΠΑΛΛΑΚΤΙΚΗ ΕΡΓΑΣΙΑ 1

ΑΝΑΓΝΩΡΙΣΗ ΧΑΡΑΚΤΗΡΙΣΤΙΚΩΝ ΓΡΑΜΜΩΝ ΣΕ 3Δ ΜΟΝΤΕΛΑ

ΝΑΤΑΛΙΑ ΡΟΥΣΚΑ - ΑΜ 1092581

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ
ΜΟΥΣΤΑΚΑΣ ΚΩΝΣΤΑΝΤΙΝΟΣ

PIO

15 ΙΟΥΛΙΟΥ 2025

Η είσοδος στο πρόγραμμα είναι 3D meshes με γνωστό connectivity, δηλαδή .obj, .ply αρχεία με γνωστές τις δομές vertices και triangles. Τα πρώτα 4 ερωτήματα υλοποιούνται στο αρχείο feature_extractor.py και το 5^ο στο mesh_comparator.py

Github Repository : <https://github.com/nrouska/Recognition-of-feature-lines-3d-models.git>

ΕΡΩΤΗΜΑ 1

Σε αυτό το ερώτημα κάθε σημείο του μοντέλου ταξινομείται σε μία από τις λίστες edges, faces, corners ανάλογα με το γεωμετρικό σχήμα της γειτονιάς του.

Συγκεκριμένα, δημιουργώ ένα patch γειτόνων σε n hops, δηλαδή για το κάθε σημείο αποθηκεύω τους άμεσους γείτονες του (hop 1), τους γείτονες των γειτόνων (hop 2) κ.ο.κ. Εφαρμόζω Principal Component Analysis σε κάθε patch, για να βρω την κατεύθυνση της μέγιστης διασποράς στο patch. Ο 3x3 covariance matrix που προκύπτει έχει 3 ιδιοτιμές $\lambda_3 > \lambda_2 > \lambda_1$, με την μεγαλύτερη να υποδεικνύει το ιδιοδιάνυσμα της μέγιστης διασποράς στο patch. Κανονικοποιώ τις ιδιοτιμές για να μην εξαρτώνται από το scale του μοντέλου και οι λόγοι τους θα μας δώσουν την επιθυμητή ταξινόμηση.

Αρχικά, αν ισχύει $\frac{\lambda_1}{\lambda_3} \geq 0.08$ σημαίνει ότι η μικρότερη ιδιοτιμή δεν είναι αμελητέα σε σχέση με την μεγαλύτερη, άρα η διασπορά του patch είναι προς όλες τις κατευθύνσεις και το σημείο ταξινομείται σε περιοχή κορυφής. Στη συνέχεια αν ισχύει $\frac{\lambda_3}{\lambda_1 + \lambda_2} > 4$, σημαίνει ότι η μεγαλύτερη ιδιοτιμή υπερέχει σημαντικά των άλλων δύο, άρα η διασπορά είναι γραμμική και το σημείο ταξινομείται σε περιοχή ακμής. Τέλος, αν ισχύει $\lambda_3 < 0.6$ η μεγαλύτερη ιδιοτιμή είναι σχετικά μικρή, οπότε υπάρχει χαμηλή διασπορά και το σημείο ταξινομείται σε περιοχή επιφάνειας.

Τα thresholds 0.08, 4, 0.6 είναι ευρετικά και φάνηκε να λειτουργούν αποδοτικά στην ταξινόμηση στα μοντέλα που δοκίμασα. Προφανώς αν αλλάξουν, αλλάζει και η ταξινόμηση. Επιπλέον για το παρακάτω αποτέλεσμα έχουν χρησιμοποιηθεί 5 hops γειτόνων, καθώς αυτό το εύρος φάνηκε να δίνει ικανοποιητικά οπτικά αποτελέσματα για το πως κατανέμονται τα σημεία τοπικά.

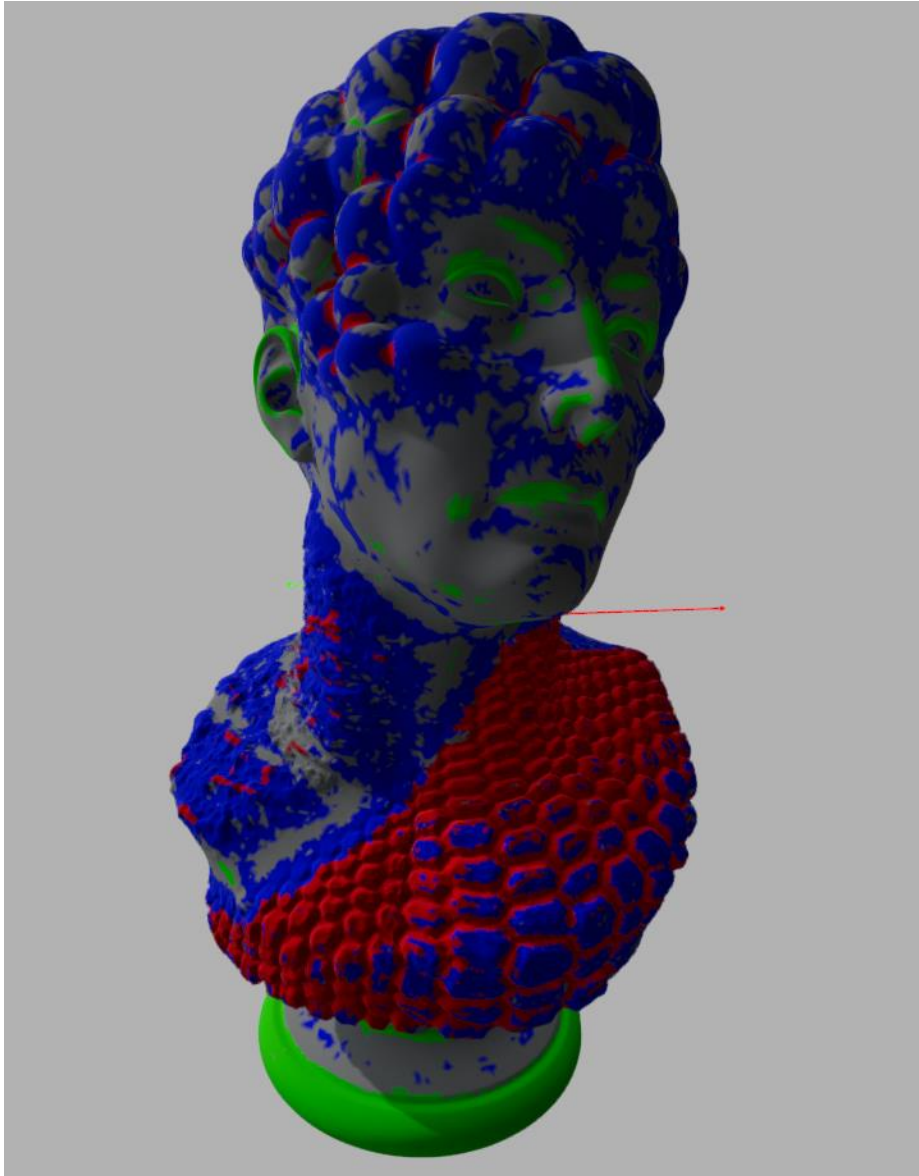


Figure 1: Task1_classify_points() portrait1.ply

Green Points in edges: 25852

Red Points in corners: 34508

Blue Points in faces: 88900

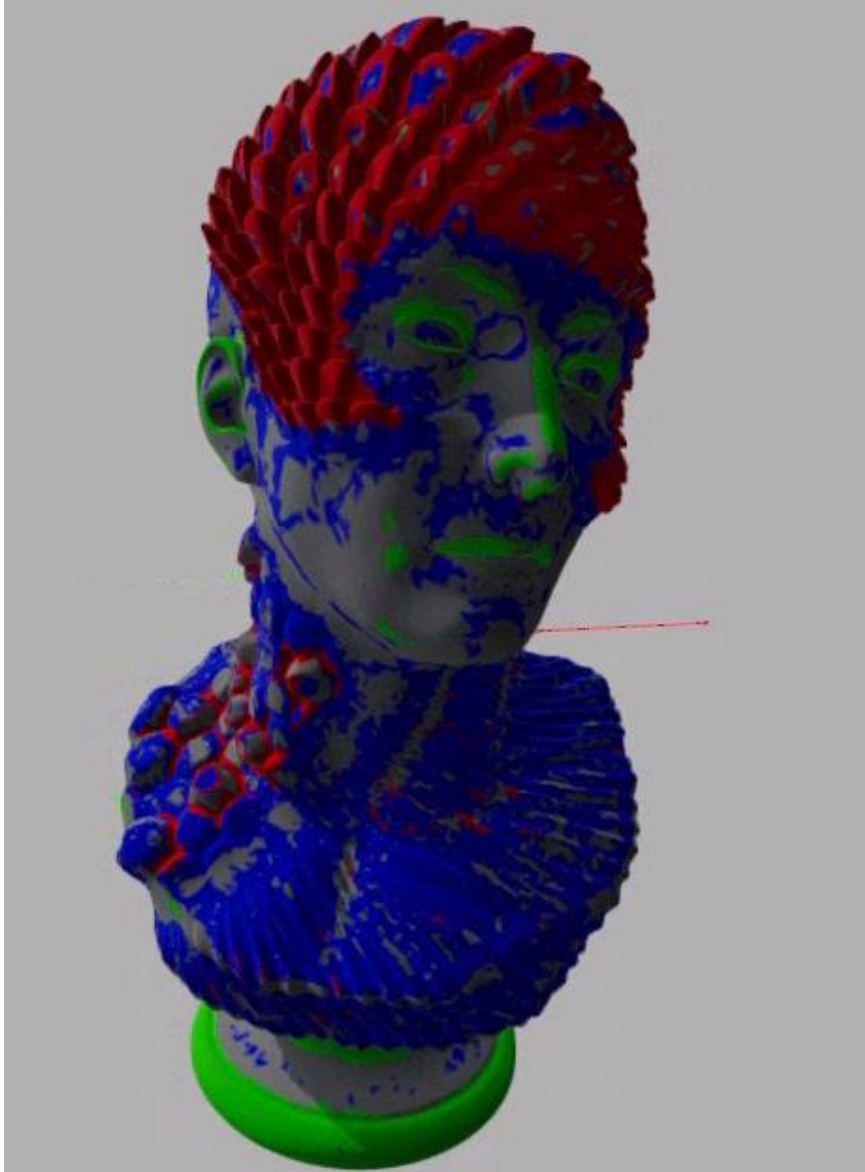


Figure 2: Task1_classify_points() portrait2.ply

Green Points in edges: 28853

Red Points in corners: 42286

Blue Points in faces: 71869

ΕΡΩΤΗΜΑ 2

Στο ερώτημα αυτό επεξεργάζομαι τα σημεία του mesh που έχουν ταξινομηθεί ότι ανήκουν σε κάποιο edge. Για κάθε σημείο, διασχίζω τα υπόλοιπα με Depth First Search ακολουθώντας γειτονικά σημεία και τα τοποθετώ στο ίδιο curve[], μόνο αν δεν παρουσιάζουν απότομη γωνιακή αλλαγή σε σχέση με την πορεία την καμπύλης.

Συγκεκριμένα, υπολογίζω τη γωνία μεταξύ των διανυσμάτων previous point-current point, current point – neighbor point. Αν το συνημίτονο της γωνίας αυτής είναι μικρότερο από το αντίστοιχο των 50° , τότε ο γείτονας απορρίπτεται από την καμπύλη, καθώς προκαλεί απότομη αλλαγή στην κατεύθυνσή της. Έτσι έχω πιο ομαλές κλίσεις των καμπυλών.

Επιπλέον αποθηκεύονται οι καμπύλες που έχουν μέγεθος πάνω από 35 σημεία για να αγνοηθούν θορυβώδη χαρακτηριστικά στο mesh.

Τα σημεία των καμπυλών (λίστες δεικτών σε vertices) που εξάγονται από την `extract_feature_curves()` δεν είναι ταξινομημένα ώστε να εξασφαλιστεί η γραμμική ή κυκλική συνέχεια της καμπύλης αν ενωθούν. Οπότε χρησιμοποιείται η `order_curve_points()` που επιστρέφει τα σημεία των καμπυλών σε σωστή γεωμετρική σειρά, με βάση την λίστα γειτνίασης ξεκινώντας από ένα άκρο.

ΕΡΩΤΗΜΑ 3

Σε αυτό το ερώτημα σχεδιάζω τις καμπύλες, ενώνοντας ευθύγραμμα τμήματα των διαδοχικών σημείων από τα οποία αποτελείται η κάθε καμπύλη, όπως έχουν ταξινομηθεί από τη `order_curve_points()`.

Πριν συνδεθούν δύο διαδοχικά σημεία, γίνεται έλεγχος της μεταξύ τους ευκλείδειας απόστασης, ώστε να αποφευχθούν συνδέσεις μεταξύ σημείων που αν και θεωρούνται γειτονικά στη δομή, απέχουν σημαντικά στο χώρο αυτό θα οδηγούσε σε jumps της καμπύλης. Στην σκηνή, κάθε καμπύλη αναπαρίσταται από ένα LineSet3D με τυχαίο χρώμα, οπότε έχω τόσα χρώματα όσες και οι χαρακτηριστικές γραμμές.

Παρακάτω φαίνεται το αποτέλεσμα των ερωτημάτων 2&3 για τα δύο μοντέλα

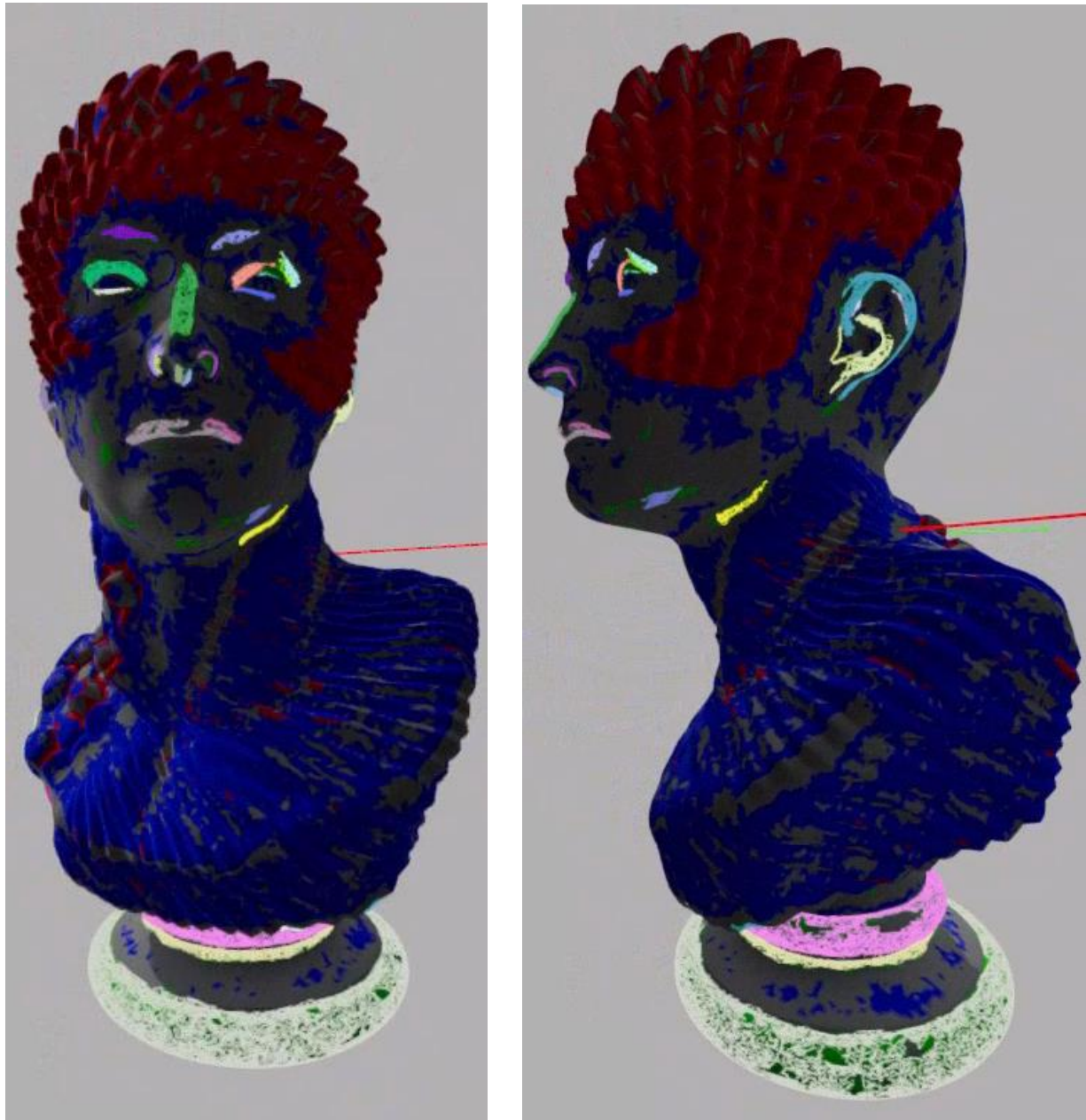


Figure 3: Task2_colored_feature_curves() Curves: 47 portrait2.ply

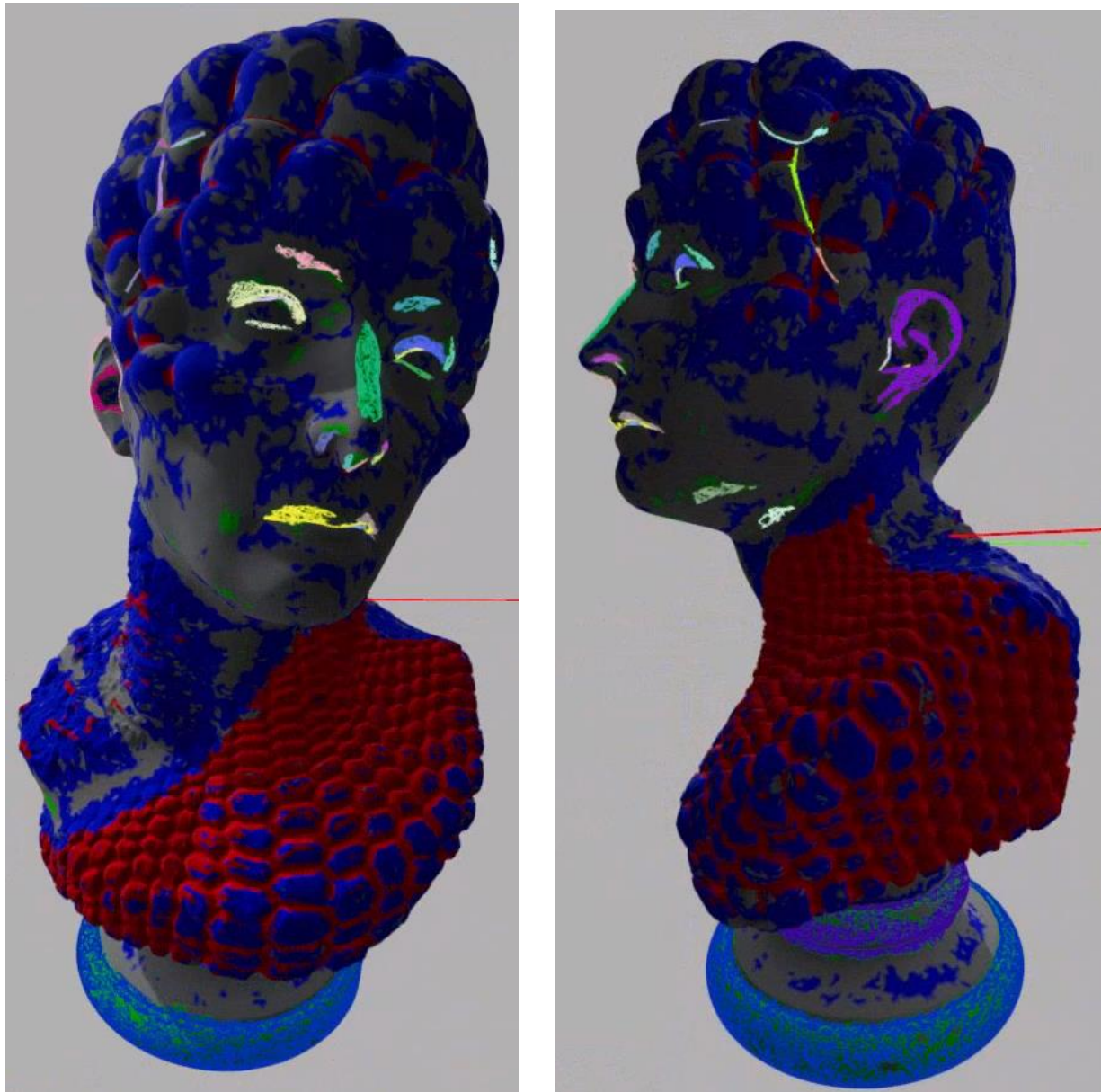


Figure 4: Task2_colored_feature_curves Curves: 43 portrait1.py

ΕΡΩΤΗΜΑ 4

Σε αυτό το ερώτημα κάθε χαρακτηριστική γραμμή αναπαρίσταται από ένα feature vector με τα εξής χαρακτηριστικά : length [1] , curvature [2], direction [3], compactness [4]. Για την ομαδοποίηση των γραμμών χρησιμοποιείται η μέθοδος του cosine similarity μεταξύ των διανυσμάτων χαρακτηριστικών.

Δύο καμπύλες θεωρούνται ότι ανήκουν στην ίδια ομάδα αν η τιμή του cosine similarity μεταξύ των διανυσμάτων τους είναι μεγαλύτερη ή ίση του 0.5. Με αυτόν τον τρόπο, καμπύλες με παρόμοια γεωμετρικά χαρακτηριστικά μπαίνουν στην ίδια ομάδα και αποτελούν το ίδιο αντικείμενο.

Η σύγκριση των διανυσμάτων εξαρτάται από τη σχετική διαφορά των χαρακτηριστικών μεταξύ τους και όχι ποιο έχει μεγαλύτερες τιμές. Άρα τα χαρακτηριστικά κανονικοποιούνται ώστε να έχουν μηδενικό μέσο και μονάδα διασποράς, εξασφαλίζοντας δίκαιη σύγκριση.

Ανάλυση χαρακτηριστικών στο feature vector της χαρακτηριστικής γραμμής

[1] Length : Το μήκος της καμπύλης βρίσκεται από το άθροισμα των αποστάσεων των διαδοχικών σημείων που την αποτελούν

[2] Curvature : Υπολογίζω την καμπυλότητα κάθε εσωτερικού σημείου της γραμμής, δηλαδή πως αλλάζει η γωνία μεταξύ δύο διαδοχικών τμημάτων ανά μονάδα μήκους. Αν η γωνία αλλάζει σε μεγάλο μήκος η καμπυλότητα είναι μικρή. Οπότε ως χαρακτηριστικό θεωρώ τη μέση καμπυλότητα όλων των εσωτερικών σημείων της καμπύλης

[3] Direction : Υπολογίζω το ιδιοδιάνυσμα, δηλαδή την κατεύθυνση της μέγιστης διασποράς των σημείων της καμπύλης (προς τα που εκτείνεται στον χώρο)

[4] Compactness : Δείχνει πόσο στρογγυλό ή μαζεμένο είναι ένα σχήμα π.χ. ο κύκλος έχει μικρή περίμετρο για το εμβαδό που περικλείει, ενώ μία λωρίδα με το ίδιο εμβαδό έχει πολύ μεγαλύτερη περίμετρο άρα μικρό compactness. Προβάλλω τα σημεία της καμπύλης στα ιδιοδιανύσματα των 2 κύριων κατευθύνσεων διασποράς τους. Υπολογίζω το 2D convex hull των σημείων της καμπύλης και το compactness είναι το εμβαδό της επιφάνειας που περικλείει το convex hull προς το τετράγωνο της περιμέτρου του.

Οι χαρακτηριστικές γραμμές που ανήκουν στο ίδιο group – object φαίνονται στη σκηνή με το ίδιο χρώμα.

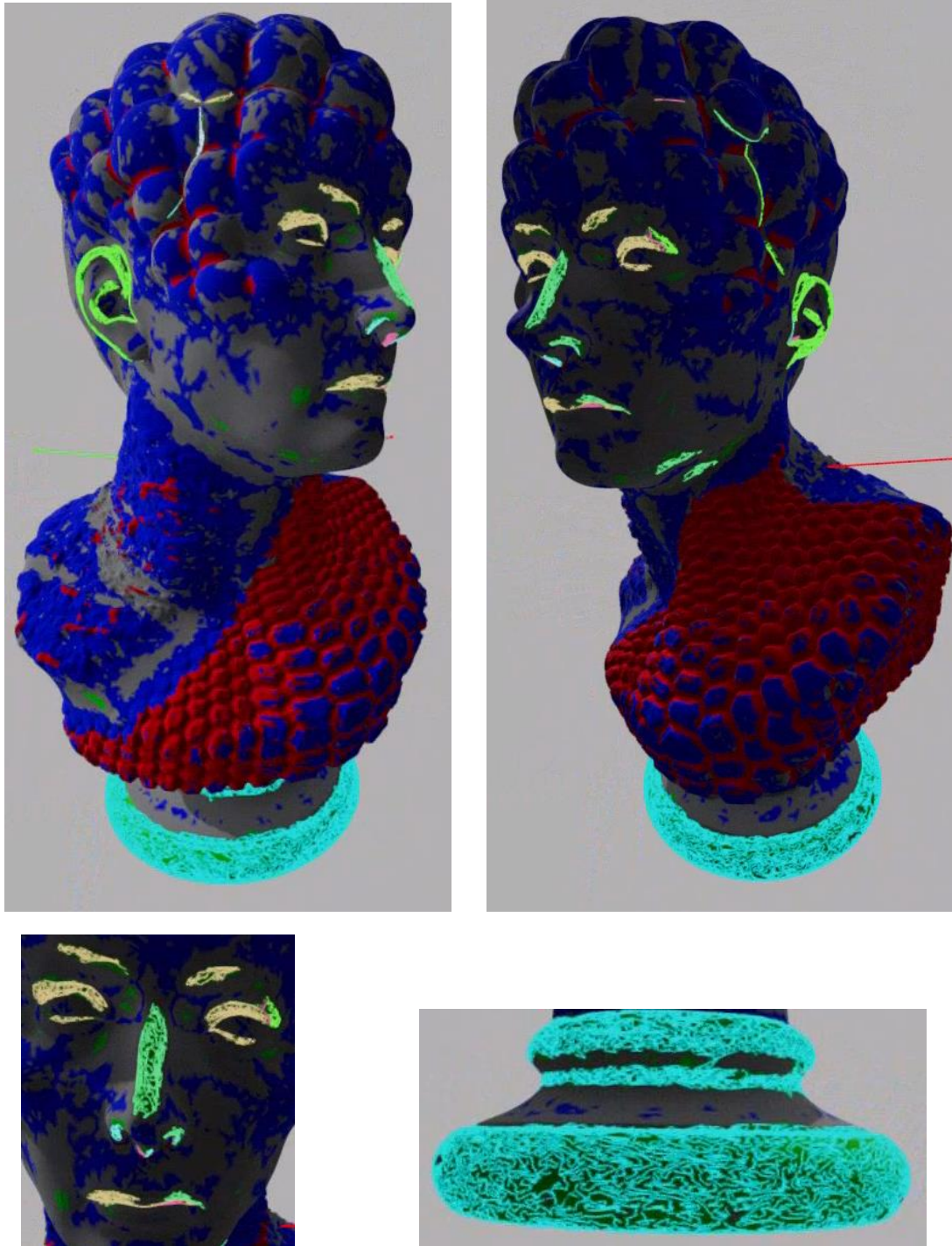


Figure 5: Task4_group_feature_curves portrait1.ply

Στο πρώτο μοντέλο, παρατηρήθηκε ότι οι καμπύλες των ματιών, των φρυδιών και των χειλιών καταλήγουν στο ίδιο γκρουπ, ενώ καμπύλες όπως τα αυτιά, η μύτη, η βάση του μοντέλου και οι λεπτομέρειες στα ρουθούνια κατηγοριοποιούνται σε ξεχωριστά.

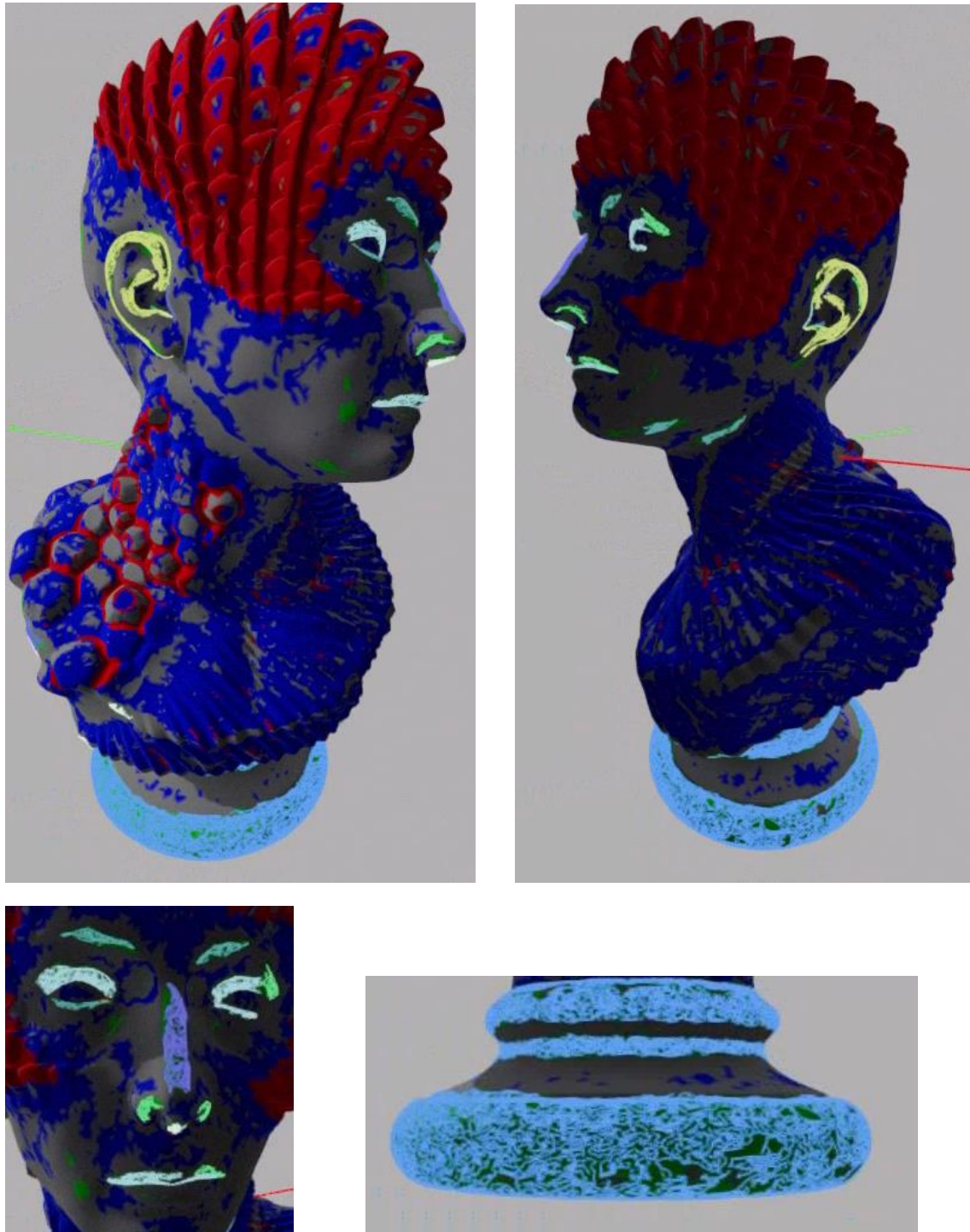


Figure 6: Task4_group_feature_curves portrait2.ply

Τα ίδια αποτελέσματα παρατηρούμε και έδω μόνο που τα φρύδια κατηγοριοποιούνται ξεχωριστά από τα μάτια.

ΕΡΩΤΗΜΑ 5

Για τη σύγκριση δύο 3D μοντέλων προσώπων, σχεδιάστηκε η κλάση MeshComparator, η οποία φορτώνει τα δύο αρχεία .ply, εξάγει τις χαρακτηριστικές καμπύλες από κάθε μοντέλο και συγκρίνει τα αντίστοιχα διανύσματα χαρακτηριστικών.

Η σύγκριση βασίζεται στον συντελεστή συσχέτισης Pearson μεταξύ των διανυσμάτων καμπυλών των δύο μοντέλων.

Για κάθε χαρακτηριστικό διάνυσμα από το πρώτο μοντέλο, υπολογίζεται η μέγιστη συσχέτιση με τα διανύσματα του δεύτερου. Αν αυτή η μέγιστη τιμή είναι μεγαλύτερη από ένα όριο (π.χ. 0.9), θεωρείται πως υπάρχει ταίριασμα της χαρακτηριστικής γραμμής με την αντίστοιχη του B.

Το ποσοστό ομοιότητας μεταξύ των δύο μοντέλων υπολογίζεται ως ο μέσος όρος των ποσοστών επιτυχούς αντιστοίχισης προς κάθε κατεύθυνση (από το πρώτο μοντέλο στο δεύτερο και αντίστροφα).

Στα δύο πρόσωπα έχουμε Similarity score: 88.64%

Παρακάτω βλέπουμε ικανοποιητικά αποτελέσματα και για δύο αντικείμενα που δεν περιλαμβάνουν χαρακτηριστικά προσώπου.

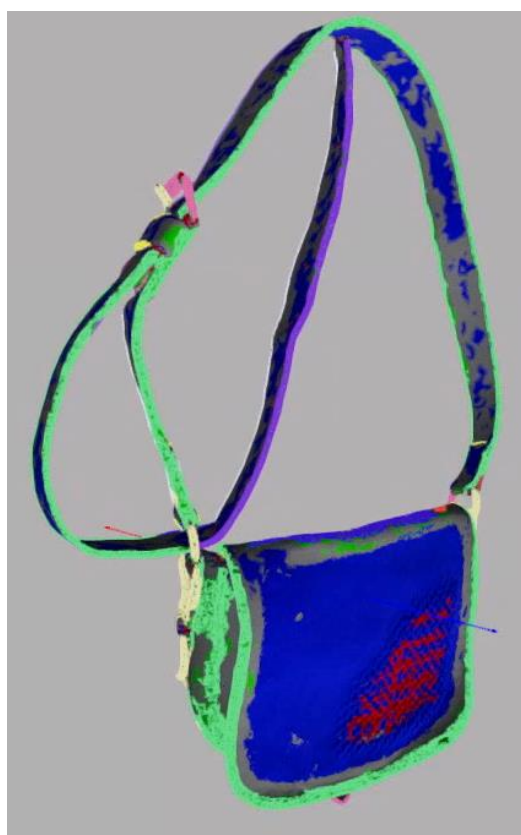


Figure 7: bag.ply

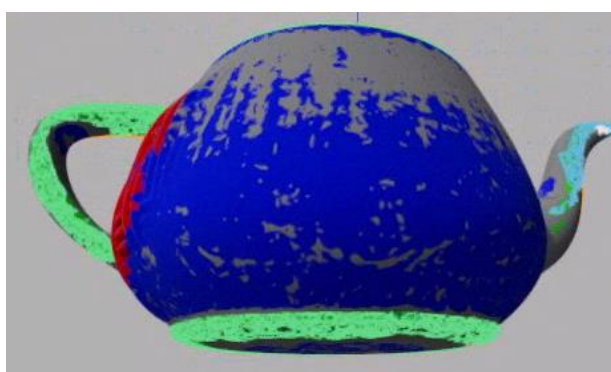
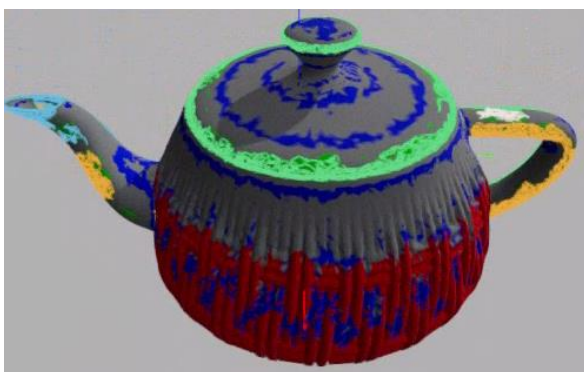


Figure 8: tea_cup.ply

Η σύγκριση του ενός πορτραίτου με τη τσάντα δίνει ποσοστό ομοιότητας 30.72% ενώ η σύγκριση του πορτραίτου με τη τσαγιέρα δίνει 28.47%. Το ότι υπάρχει μία ομοιότητα οφείλεται κυρίως στις κυκλικές καμπύλες τις βάσης του πορτραίτου που ταιριάζουν με τα λουράκια της τσάντας και τις χαρακτηριστικές γραμμές στη βάση και στο καπάκι της τσαγιέρας.

BIBΛΙΟΓΡΑΦΙΑ

[1] SHREC'19 track: Feature Curve Extraction on Triangle Meshes

https://hal.science/hal-02126739/file/sh19_FeatCurve_preprint.pdf

[2] StatQuest: Principal Component Analysis (PCA), Step-by-Step

<https://www.youtube.com/watch?v=FgakZw6K1QQ&t=30s>

[3] 3Δ meshes

https://drive.google.com/drive/folders/1U50Br4Bd-BUFrngUj5RcJa_yYBL8se2u

[4] Understanding Vector Similarity for Machine Learning

<https://medium.com/advanced-deep-learning/understanding-vector-similarity-b9c10f7506de>