

Αναφορά εργασίας εξαμήνου

Ιωάννη Τσάμπρα 1066584 ΣΜΗΝ 2024

3D Υπολογιστική Γεωμετρία και Όραση

Contents

Figures	1
Εισαγωγή.....	2
Dataset	3
Ερώτημα 3.....	4
Ερώτημα 1	4
Ερωτήματα 2 & 4.....	6
Ερώτημα 5.....	11
Ερώτημα 6.....	12
Ερώτημα 7	13
Ερώτημα 8.....	17
Οδηγίες εκτέλεσης	18

Figures

1 Original Scan Simplified to 50k	2 Simplified to 25k after removing some desks (THIS IS USED)...	3
3 Untextured Mesh Used		3
4 Branchless αλγόριθμος ομοιόμορφης τυχαίας δειγματοληψείας τριγώνων		4
5 Παραγόμενο pointcloud 156k σημείων από αρχικό mesh.....		4
6 Vertical Planes (after Z alignment through dominant plane)	7 Dominant Plane on non-aligned PCD	
8 Aligned PCD	9 Closed holes mesh	10 Closed Holes PCD
		5
11 Μέθοδος εξωτερικότητας σε aligned mesh	12 Μέθοδος εξωτερικότητας σε non aligned mesh .	6
13 Mesh clump sizes before welding	14 Mesh clump sizes after welding	7
15 Ερώτημα 2 classification output.....		7
16 Ερώτημα 2 filtered, all clumps		8
17 Ερώτημα 2 filtered, welded, clumps <10 colored green.....		9
18 Output για το ερώτημα 4		9
19 Classification για το ερώτημα 4 [A].....		10
20 Classification για το ερώτημα 4 [B].....		10
21 Regions του ερωτήματος 5		11

22 Vertices των 37 original οπών, κόκκινο στην Μεγαλύτερη οπή	12
23 Vertices όλων των οπών μετά από welding (με φθίνουσα σειρά μεγέθους)	12
24 Αντικείμενα εντός δωματίου	13
25 Τμηματοποιημένοι τοίχοι δωματίου	14
26 Rasterized convex hulls εμποδίων.....	15
27 Pathfinding Scenarios	16
28 Pathfinding scenarios in complex mesh	17
29 Χάρτης που προκύπτει για το complex mesh	18

Εισαγωγή

Η εργασία αποτελείται από 8 ερωτήματα, 1) φόρτωση δωματίου σε σκηνή, 2) εύρεση και αφαίρεση τοίχων δωματίου, 3) δημιουργία point cloud από mesh δωματίου, 4) εύρεση και αφαίρεση με χρήση point cloud δωματίου, 5) συσταδοποίηση φιλτραρισμένου point cloud, 6) εύρεση πόρτας, 7) path finding από τυχαίο σημείο δωματίου προς την πόρτα και 8) γενίκευση σε πολλαπλά δωμάτια. Το προτεινόμενο data set ήταν το 3DIS.

Λόγω των εκτεταμένων οπών στο προτεινόμενο data set καθώς και άλλων προβλημάτων, έγινε η επιλογή της δημιουργίας νέου data set για τους σκοπούς της εργασίας. Το νέο data set απεικονίζει μια αίθουσα στην οποία έχουν αφαιρεθεί με τεχνητό τρόπο πλαίσια τα οποία αντιστοιχούν στις 2 πόρτες και το 1 παράθυρο της αίθουσας. Το data set επιπλέον περιέχει μικρο-ατέλειες παραγόμενες από το φωτογραμμετρικό λογισμικό που αξιοποιήθηκε.

Το πρώτο ερώτημα που υλοποιήθηκε ήταν το ερώτημα (3) καθώς για τη ευθυγράμμιση του data set έγινε χρήση του point cloud.

Το ερώτημα (1) ανέλαβε την φόρτωση και ευθυγράμμιση του point cloud και mesh και επιπλέον την παραγωγή δεύτερων εκδοχών τόσο για το point cloud όσο και για το mesh στα οποία οι οπές παραθύρου και πορτών έχουν καλυφθεί με σκοπό την διευκόλυνση της εκτέλεσης των αλγορίθμων εύρεσης τοίχων. Όλες οι ενέργειες έχουν πραγματοποιηθεί αλγοριθμικά και αυτοματοποιημένα με είσοδο το αρχικό mesh.

Το δεύτερο και τέταρτο ερώτημα υλοποιήθηκαν με παρόμοιο αλγόριθμο είτε με την χρήση σημείων είτε με την χρήση κέντρων τριγώνων, βασισμένα στην «τοπική εξωτερικότητα» των σημείων και στην ευθυγράμμιση του δωματίου. Για αυτά τα ερωτήματα αξιοποιήθηκαν οι εκδόσεις με κλεισμένες τις οπές παραθύρων και θυρών.

Για το πέμπτο ερώτημα αξιοποιήθηκε το φιλτραρισμένο point cloud του ερωτήματος (4). Μια απλή region growing μέθοδος εφαρμόστηκε για την ομαδοποίηση των σημείων.

Στο ερώτημα (6) έγινε χρήση των εκδόσεων με τις οπές, οι οποίες εντοπίστηκαν και ταξινομήθηκαν κατά μέγεθος και οι μικρο-ατέλειες του data set επισκευάστηκαν.

Δημιουργήθηκε διαδραστική σκηνή για τις ανάγκες του ερωτήματος (7). Στη σκηνή έγινε χρήση των μεθόδων από τα ερωτήματα 4,5 και 6 για εύρεση αντικειμένων και θυρών. Επιπλέον αφαιρέθηκε το ταβάνι με μεθόδους παρόμοιες του ερωτήματος 1 για βέλτιστη οπτικοποίηση. Ο χρήστης μπορεί να επιλέξει σημείο της αρεσκείας του και μια από τις θύρες για να πλοηγηθεί.

Τέλος για το ερώτημα (8) το project κλωνοποιήθηκε και ως σύνθετο data set χρησιμοποιήθηκε ο συνδυασμός 2 δωματίων για πλοήγηση από το ένα στο άλλο.

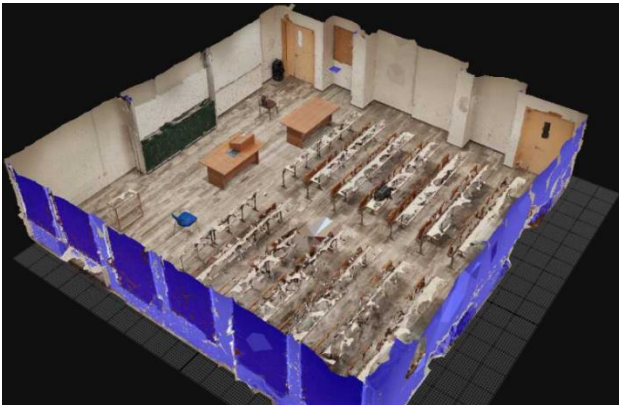
Dataset

Προβλήματα στο προτεινόμενο dataset ήταν τα εξής:

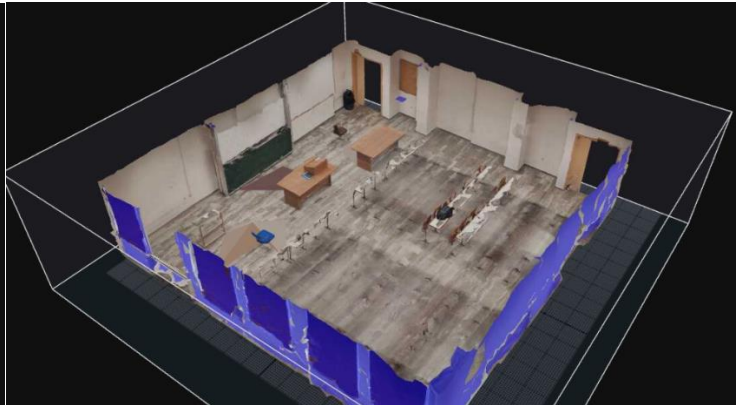
- Έλλειψη isolated room mesh
- **Οπές μεγάλου μεγέθους σε τοίχους πίσω από αντικείμενα**
- Σχετικά χαμηλή ανάλυση (2.5k triangles in a room)
- Εκτεταμένες μικρο-οπές σε αντικείμενα και τοίχους
- Μη συνδεδεμένα γειτονικά τρίγωνα

Εξ αιτίας των παραπάνω έγινε η επιλογή να δημιουργηθεί νέο dataset καλύτερης ποιότητας.

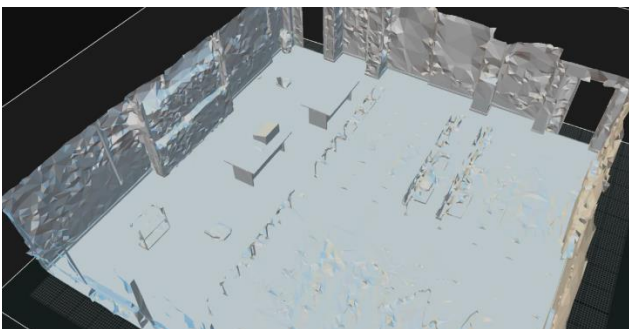
Επιλέχθηκε αίθουσα διδασκαλίας στο κτήριο των Πολιτικών Μηχανικών με 2 πόρτες, παράθυρα, μη επίπεδους τοίχους, πίνακα και πολλαπλά εμπόδια (2 έδρες, 2 καρέκλες, μικρή ραφιάρα στον κεντρικό χώρο, κρεμόμενο προτζέκτορα και πολλά καθίσματα). Τα παράθυρα και οι πόρτες φωτογραφήθηκαν κλειστά και το mesh κόπηκε κατάλληλα για να δημιουργηθούν τα ανοίγματα. Επιπλέον αφαιρέθηκαν μερικά από τα καθίσματα για καλύτερες συνθήκες pathfinding. Ακόμα έγινε η επιλογή παραγωγής mesh των 25k τριγώνων με χρήση του simplification tool του λογισμικού καθώς παρείχε αρκετή ακρίβεια χωρίς το βάρος των 186M τριγώνων που παρήχθησαν αρχικά. Το λογισμικό που αξιοποιήθηκε ήταν το RealityCapture και έγινε χρήση ~500 φωτογραφιών. Μόνο το τελικό mesh της παραπάνω επεξεργασίας «έγινε γνωστό» στο pipeline κώδικα που υλοποιήθηκε για το project.



1 Original Scan Simplified to 50k



2 Simplified to 25k after removing some desks (THIS IS USED)



3 Untextured Mesh Used

Ερώτημα 3

Το ερώτημα 3 αντιμετωπίστηκε πρώτο καθώς δεν εξαρτάται από τα υπόλοιπα και το αποτέλεσμα του αξιοποιήθηκε για το alignment στο ερώτημα 1). Ακολουθώντας την μεθοδολογία από [Weisstein, Eric W.

"Triangle Point Picking." From MathWorld--A Wolfram Web Resource. <https://mathworld.wolfram.com/TrianglePointPicking.html>] αποφεύγουμε την συγκέντρωση των τυχαίων σημείων στο κέντρο ή σε κάποια γωνία του τριγώνου και υλοποιούμε στην python ως εξής:

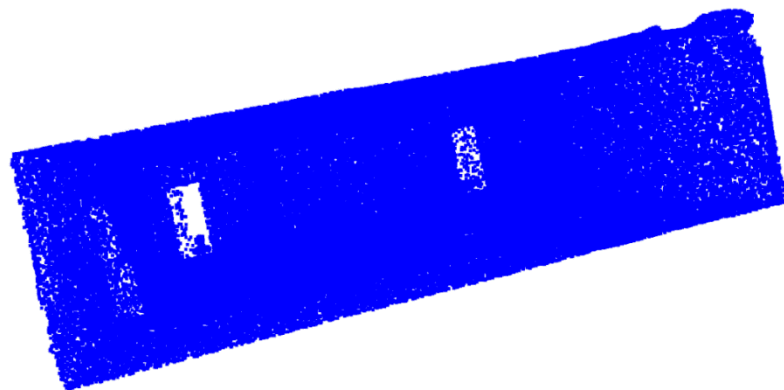
```
vector1 = v1 - v0
vector2 = v2 - v0

for _ in range(points_count):
    r1, r2 = np.random.rand(2)
    is_ok = (r1 + r2) <= 1
    r1 = is_ok * r1 + (not is_ok) * (1 - r1)
    r2 = is_ok * r2 + (not is_ok) * (1 - r2)

    point = v0 + r1 * vector1 + r2 * vector2
    points.append(point)
```

4 Branchless αλγόριθμος ομοιόμορφης τυχαίας δειγματοληψίας τριγώνων

Ουσιαστικά επιλέγουμε τυχαία δυο ακμές του τριγώνου και για καθεμία έναν συντελεστή r μεταξύ $[0,1]$. Πολλαπλασιάζοντας τα διανύσματα των δύο πλευρών με τους συντελεστές τους λαμβάνουμε ένα σημείο εντός του παραλληλογράμου που ορίζουν οι δύο πλευρές, ελέγχουμε όμως αν οι τυχαίοι συντελεστές έχουν άθροισμα μεγαλύτερο του 1 για να ξέρουμε αν το σημείο βρίσκεται εντός του τριγώνου, εάν δεν βρίσκεται η συνθήκη `is_ok` επηρεάζει το αριθμητικό αποτέλεσμα ώστε να «διπλώσει» το σημείο εντός του τριγώνου. Ο κώδικας είναι branchless και δεν απορρίπτει στοιχεία. Επιπλέον το πλήθος των σημείων ανά τρίγωνο καθορίζεται από το εμβαδό του και η πυκνότητα είναι παραμετροποιήσιμη. Το αρχείο σώζεται στον φάκελο results.



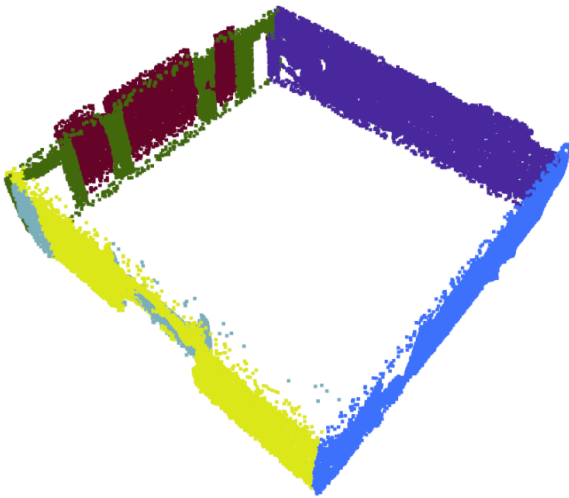
5 Παραγόμενο pointcloud 156k σημείων από αρχικό mesh

Ερώτημα 1

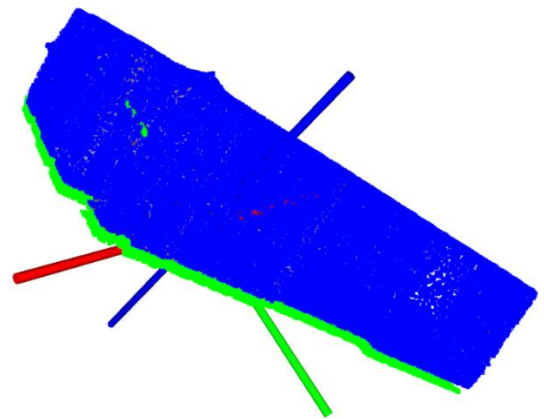
Εντός του ερωτήματος 1) πραγματοποιούνται τα εξής:

1. Φόρτωση αρχικού mesh (με ανοίγματα όπως περιγράφηκε) και random rotations/translations
2. Εξαγωγή αντίστοιχου pointcloud με χρήση ερωτήματος 3
3. Εύρεση dominant plane με χρήση ransac στο pointcloud για τον εντοπισμό του ταβανιού/πατώματος
4. Χρήση dominant plane για alignment pcd και mesh στον άξονα Z
5. Εύρεση όλων των κάθετων επιπέδων στο xy επίπεδο τα οποία αναπαριστούν τμήματα των πλαγιών τοίχων (κατά πλειονότητα).
6. Εξαγωγή γωνίας περιστροφής γύρω από άξονα z για όλα τα παραγόμενα επίπεδα.
7. 1D Weighted Clustering των γωνιών, με βάρη το πλήθος σημείων του αντίστοιχου επιπέδου, για 2 κεντροειδή (οι δύο κατευθύνσεις των τοίχων)
8. Χρήση της γωνίας του μικρότερου κεντροειδούς για περιστροφή του mesh και του pcd γύρω από τον άξονα Z
9. Χρήση μεθόδων όμοιες με αυτές που θα αναπτυχθούν αργότερα στο ερώτημα 6 για την εύρεση οπών και την κάλυψή τους στο aligned mesh.
10. Αποθήκευση τόσο του aligned mesh με κλειστές τρύπες όσο και του pcd που παράγεται από αυτό

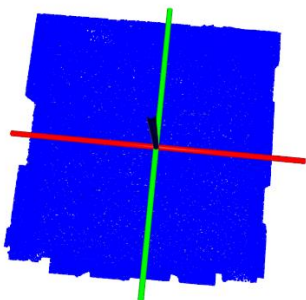
Στο τέλος του ερωτήματος 1 θα έχουμε 2 meshes και τα αντίστοιχα 2 pcd, το ένα set έχει τις τρύπες κλειστές και το άλλο όχι. Και τα δύο σετ θα είναι aligned σε x,y,z και στο ίδιο coordinate space.



6 Vertical Planes (after Z alignment through dominant plane)



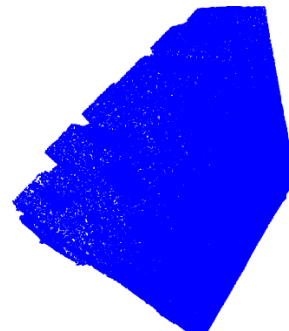
7 Dominant Plane on non-aligned PCD



8 Aligned PCD



9 Closed holes mesh



10 Closed Holes PCD

Ερωτήματα 2 & 4

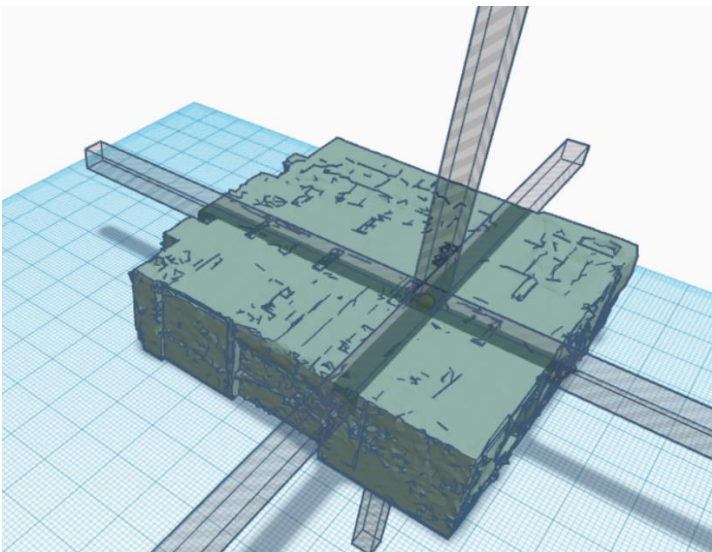
Για τα ερωτήματα 2 και 4 αναπτύχθηκε αλγόριθμος «τοπικής εξωτερικότητας» καθώς ο ransac αδυνατούσε να εξαγάγει όλα τα τμήματα των τοίχων χωρίς να αφαιρέσει και τμήματα εσωτερικών αντικειμένων.

Στην περίπτωση του mesh (ερ. 2) ως είσοδος αξιοποιούνται τα κέντρα των τριγώνων ενώ στο pcd (ερ. 4) όλα τα σημεία που δημιουργήθηκαν από το ερώτημα 3.

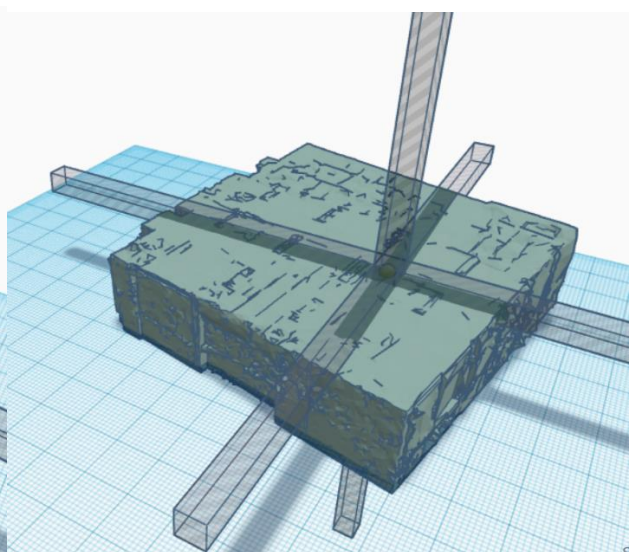
Ο αλγόριθμος βασίζεται στο αν ένα σημείο βρίσκεται στο εξωτερικό του mesh αλλά σε αντίθεση με το απλό thresholding (το οποίο θα ήταν επίσης εφικτό λόγω του πολύ καλού αποτελέσματος του alignment όπως αυτό περιγράφηκε στο ερώτημα 1) διερευνά την τοπική περιοχή του σημείου ως εξής:

- Τα σημεία εισόδου γίνονται sort 3 φορές σε 3 νέα arrays αφού προβληθούν στους άξονες x,y,z.
- Για κάθε array δημιουργείται ένα BST με κλειδί την αντίστοιχη διάσταση των σημείων. Αυτό γίνεται μία φορά.
- Κάθε φορά που θέλουμε να εξετάσουμε την εξωτερικότητα ενός σημείου λαμβανουμε μια «φέτα» της διάστασης 1 των σημείων με παραμετροποιήσιμο πάχος (επιλογή «τοπικότητας») από κάθε array και φιλτράρουμε με βάση της διάσταση 2 (δεύτερη επιλογή «τοπικότητας») δημιουργώντας μια στήλη από την φέτα μας και προβάλλουμε στην διάσταση 3.
- Στην διάσταση 3 βρίσκουμε μέγιστο και ελάχιστο.
- Εξετάζουμε αν το υπό έλεγχο σημείο βρίσκεται εκτός του threshold μας (π.χ. 5%) είτε προς τα πάνω είτε προς τα κάτω όπου 0% το min και 100% το max.
- Επαναλαμβάνουμε 3 φορές, μία για κάθε άξονα

Η μέθοδος αυτή λειτουργεί και σε non-perfect aligned αντικείμενα και οι 3 άξονες που επιλέγονται εδώ (x,y,z) είναι ενδεικτικοί. Η μόνη προϋπόθεση είναι να ρυθμίζονται κατάλληλα τα threshold τοπικότητας της στήλης με βάση την χωρική πυκνότητα της εισόδου. Μια adaptive λύση έχει υλοποιηθεί στην εργασία όπου το πλάτος της στήλης προσαρμόζεται στην πυκνότητα της φέτας σε βήματα διπλασιασμού του πάχους. Για το ερώτημα 4 η διαδικασία έχει παραλληλοποιηθεί για γρηγορότερη εκτέλεση.



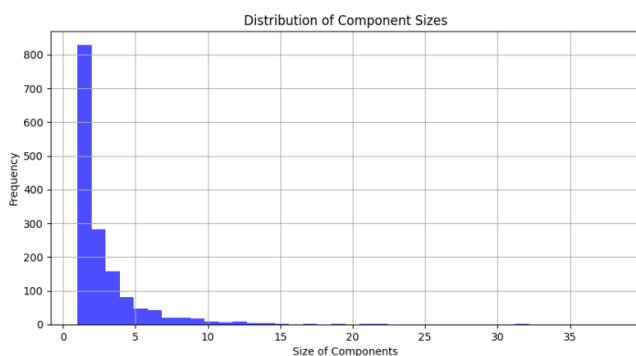
11 Μέθοδος εξωτερικότητας σε aligned mesh



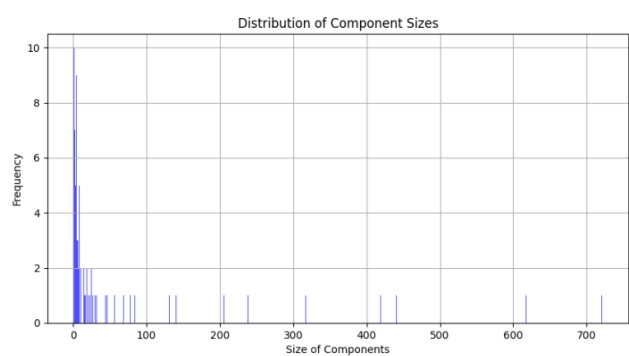
12 Μέθοδος εξωτερικότητας σε non aligned mesh

Η μέθοδος παράγει ακριβή αποτελέσματα αλλά προϋποθέτει την μη ύπαρξη μεγάλων οπών οπότε εδώ γίνεται χρήση της έκδοσης των mesh και pcd χωρίς οπές που παράξαμε στο ερώτημα 1.

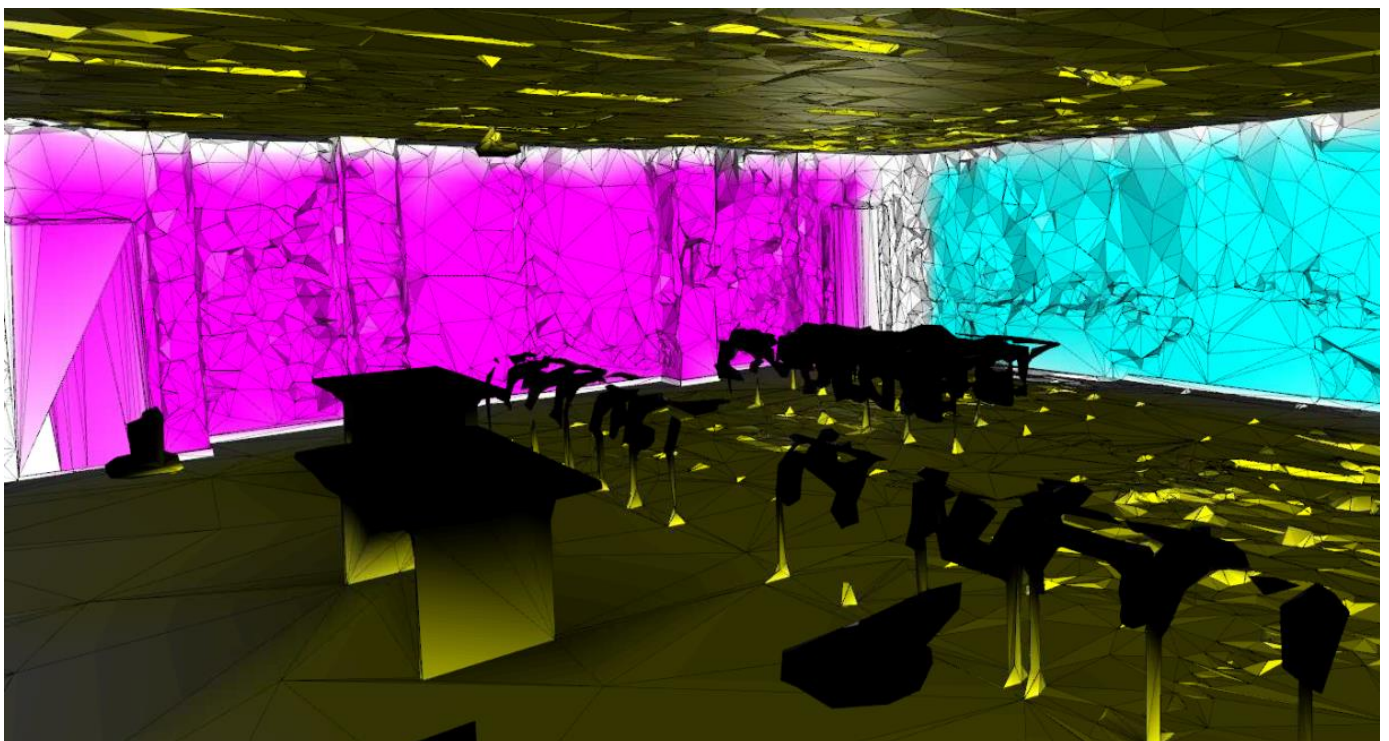
Στην περίπτωση του ερωτήματος 2 όπου ως είσοδο έχουμε τα κέντρα των τριγώνων παράγονται διάσπαρτα μικρά σύνολα ή μεμονομένα τρίγωνα τοίχων τα οποία αναγνωρίζονται ως αντικείμενα. Αυτό γίνεται λόγω της χαμηλής πυκνότητας της εισόδου. Ως αντιμετώπισης συσταδοποιούμε τα τρίγωνα σε «clumps» με βάση τη γειτονικότητα και διαγράφουμε τα μικρά clumps (έγινε χρήση δομής Union Find [<https://www.geeksforgeeks.org/introduction-to-disjoint-set-data-structure-or-union-find-algorithm/>] για την επιτάχυνση της διαδικασίας εύρεσης clumps). Δυστυχώς όμως πολλά vertices έχουν δημιουργηθεί από το φωτογραμμετρικό λογισμικό ως «διπλότυπα» με ελαφρώς διαφορετικές συντεταγμένες οπότε η γειτόνευση είναι ανακριβής. Για να επιλυθεί αυτό εφαρμόζουμε τεχνική «welding» όπου πολύ κοντινά vertices εννώνονται σε ένα (έγινε χρήση kd-tree για τα queries κοντινών σημείων).



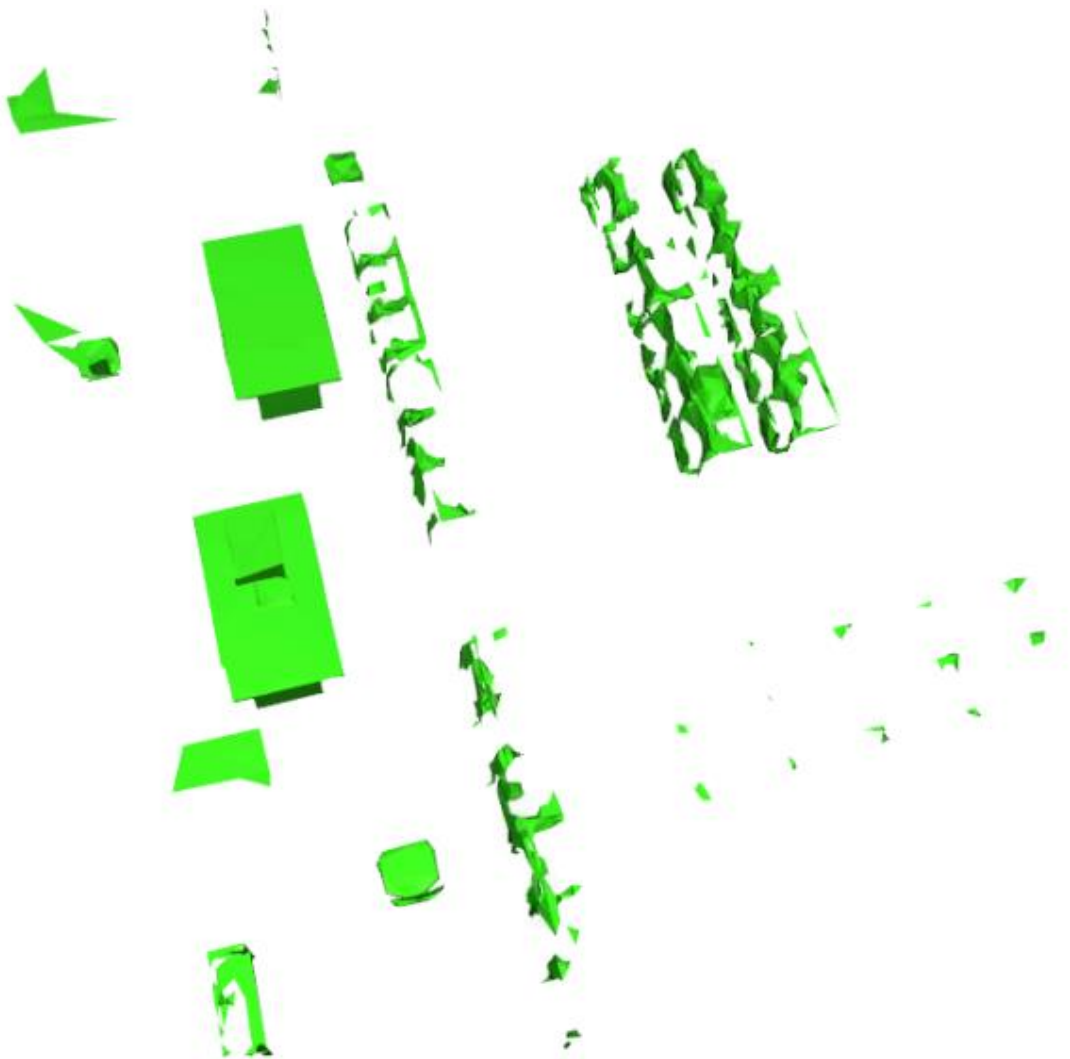
13 Mesh clump sizes before welding



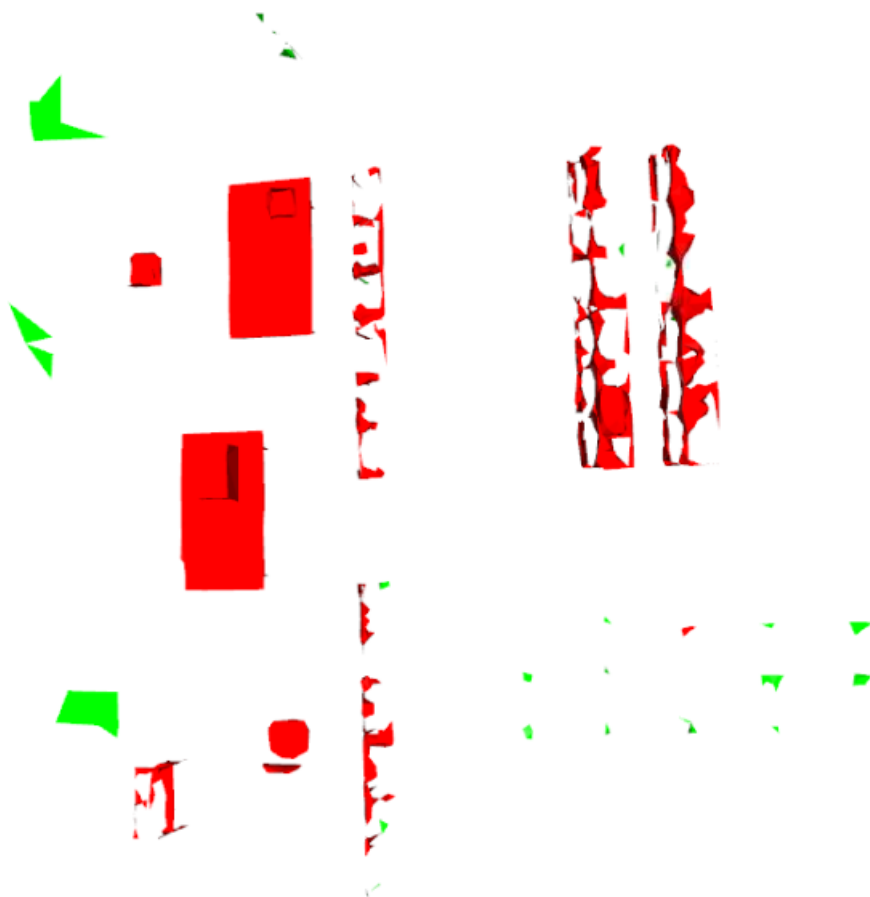
14 Mesh clump sizes after welding



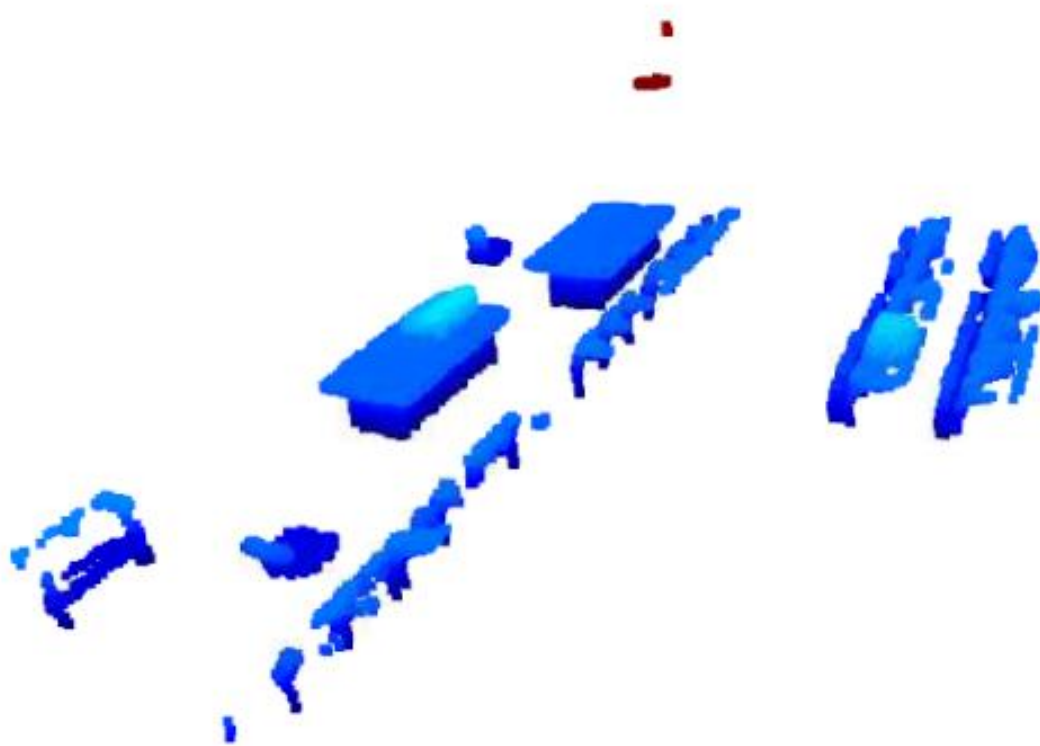
15 Ερώτημα 2 classification output



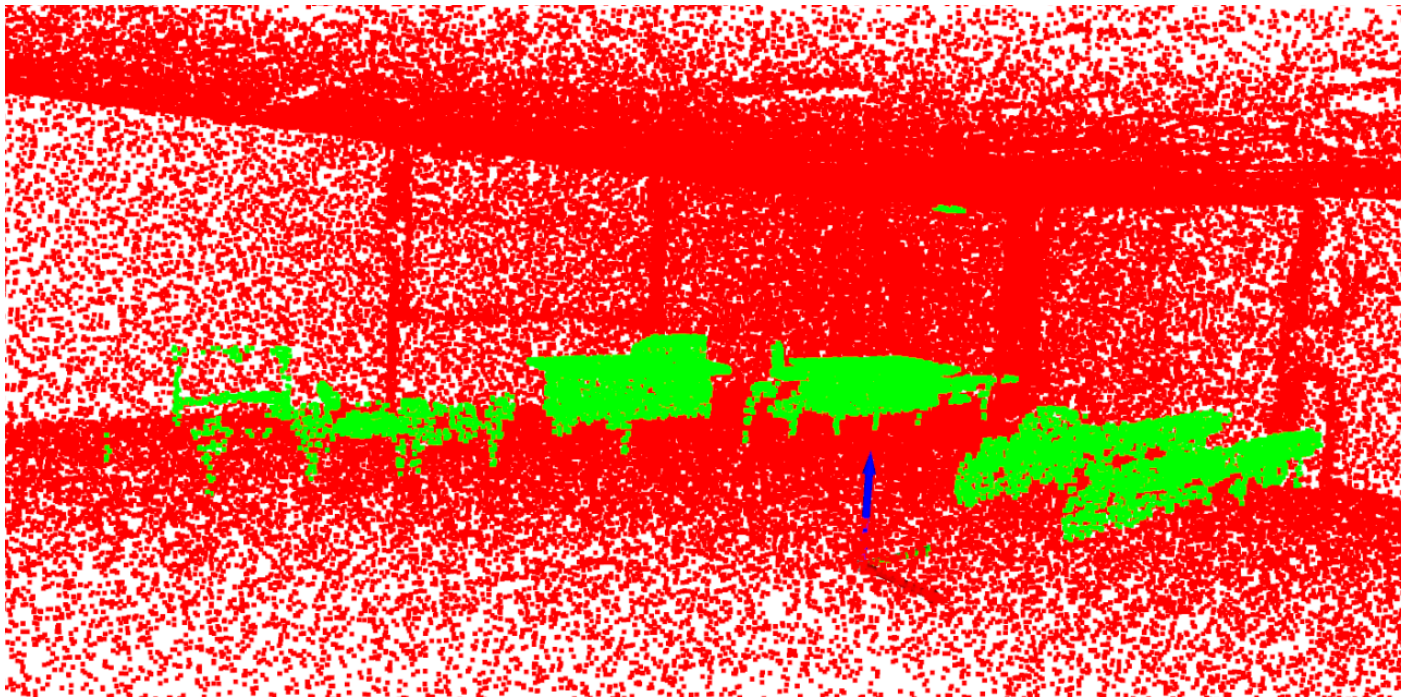
16 Ερώτημα 2 filtered, all clumps



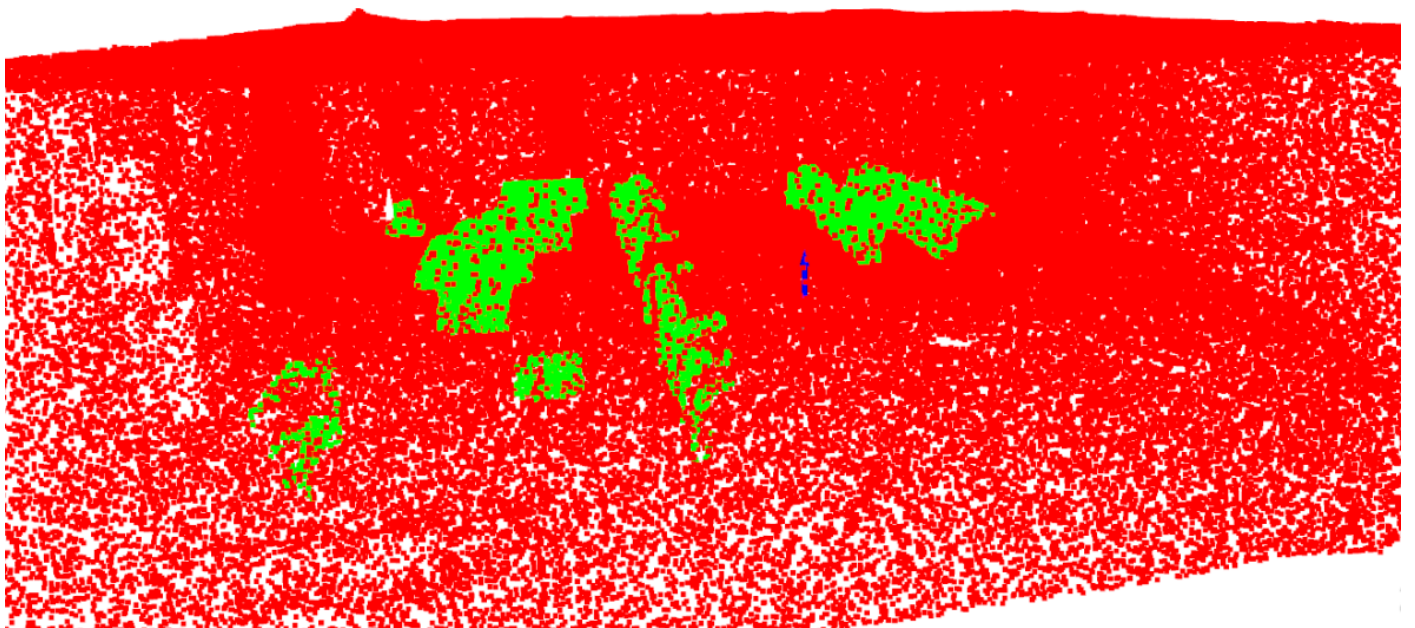
17 Ερώτημα 2 filtered, welded, clumps <10 colored green



18 Output για το ερώτημα 4



19 Classification για το ερώτημα 4 [A]



20 Classification για το ερώτημα 4 [B]

Ερώτημα 5

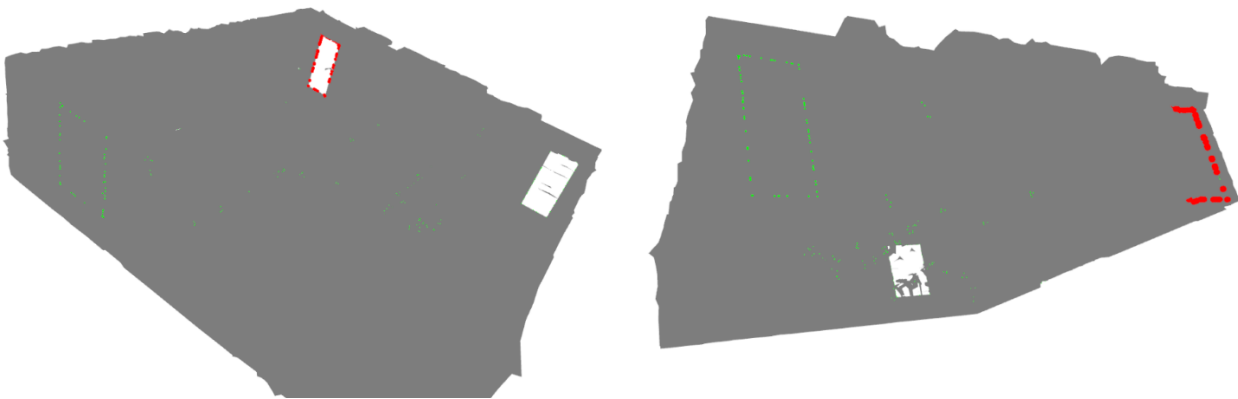
Για το ερώτημα 5 υλοποιήθηκε μια μέθοδος region growing με χρήση kd-tree και union find. Μέθοδοι clustering όπως kmeans και dbscan δεν αποτύπωναν την συνέχεια των αντικειμένων. Για τα regions ορίστηκε ως περιορισμός ελάχιστα σημεία = 4 ώστε να αποφύγουμε μη επιτρεπτές εισόδους σε αλγόριθμος convex hull αργότερα. Χρησιμοποιήθηκε η έξοδος του ερωτήματος 4.



21 Regions του ερωτήματος 5

Ερώτημα 6

Για το ερώτημα 6 ο σκοπός ήταν η εύρεση της πόρτας, σε αυτήν την περίπτωση, των πορτών. Ως κριτήριο αξιοποιήθηκαν οι σπές του mesh και το μέγεθός τους. Επίσης εφαρμόστηκε welding για να αφαιρεθούν μικρο-σπές από ατέλειες του φωτογραμμετρικού λογισμικού. Ως μετρική μεγέθους ορίστηκε το πλήθος των ακμών ενώ οι ακμές βρέθηκαν με κριτήριο το αν ανήκουν σε μόνο ένα τρίγωνο, επιπλέον βάση της γειτόνευσής τους χωρίστηκαν σε επιμέρους ομάδες ακμών, τις τρύπες δηλαδή.



22 Vertices των 37 original σπών, κόκκινο στην Μεγαλύτερη σπή



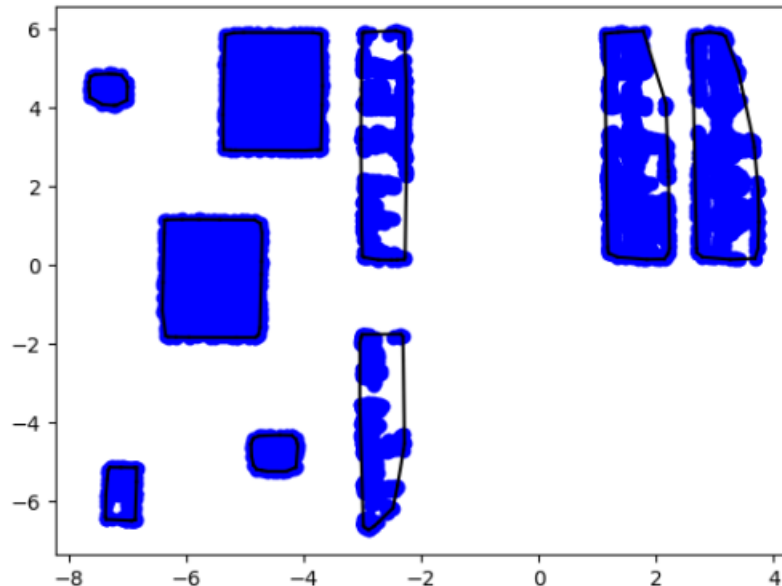
23 Vertices όλων των σπών μετά από welding (με φθίνουσα σειρά μεγέθους)

Ερώτημα 7

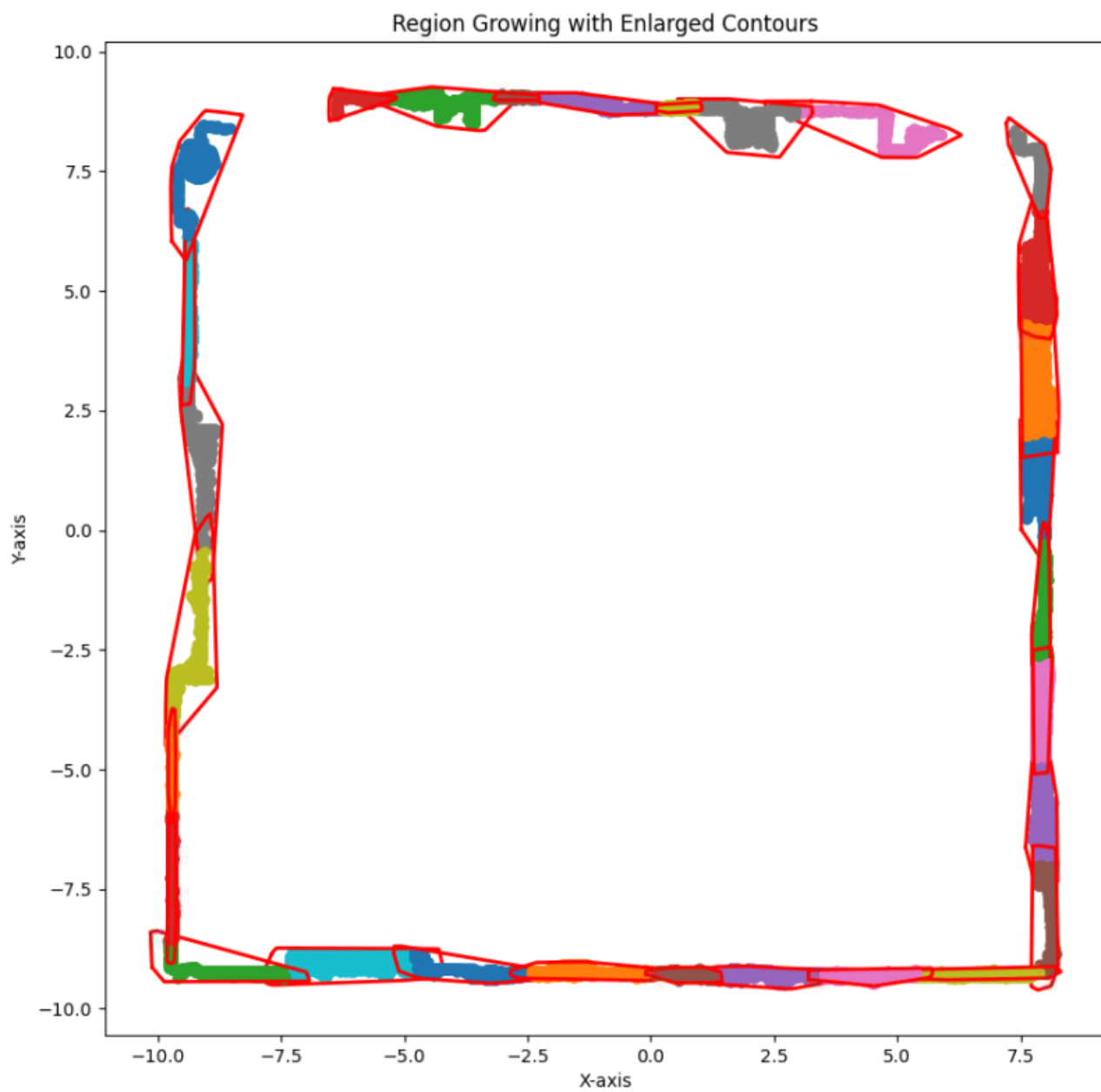
Δημιουργήθηκε διαδραστική σκηνή με συνδιασμό μεθόδων από προηγούμενα ερωτήματα όπου παρουσιάζεται το mesh του δωματίου χωρίς το ταβάνι για λόγους οπτικοποίησης και ο χρήστης δύναται να επιλέξει σημείο του mesh με shift+leftclick και πόρτα με κυκλική περιστροφή μεταξύ των διαθέσιμων θυρών με το πλήκτρο «Τ». Επιπλέον με το πλήκτρο «C» ο χρήστης μπορεί να δει highlighted τα εσωτερικά αντικείμενα του δωματίου με διαφορετικά χρώματα. Τέλος αφού έχει επιλέξει την επιθυμητή πόρτα και σημείο εκκίνησης μπορεί με το πλήκτρο «Ρ» να εκτελέσει pathfinding.

Για το pathfinding δημιουργείται ένα σύνολο από convex hulls για τα εσωτερικά αντικείμενα του δωματίου και για τους τοίχους το οποίο γίνεται rasterized σε ένα grid το οποίο γίνεται και cached στο δίσκο για χρήση σε επόμενες εκτελέσεις. Για τους τοίχους αφού εντοπιστούν όπως και στο πρώτο ερώτημα, τμηματοποιούνται χρησιμοποιώντας τον region growing αλγόριθμο από το ερώτημα 5 αλλά με χαμηλό ανώτατο όριο σημείων ώστε οι τοίχοι να σπάσουν σε πολλά «ευθύγραμμα» σύνολα σημείων επιτρέποντας έτσι την πιστή τους αναπαράσταση από convex hulls και ταυτόχρονα την μη επικάλυψη των θυρών. Τόσο για τα εσωτερικά αντικείμενα όσο και για τους τοίχους φιλτράρονται τα σημεία με ύψος πάνω του 75% του ολικού ύψους για να μείνουν οι πόρτες ανοιχτές και να αφαιρεθούν φωτιστικά σώματα και προτζέκτορες.

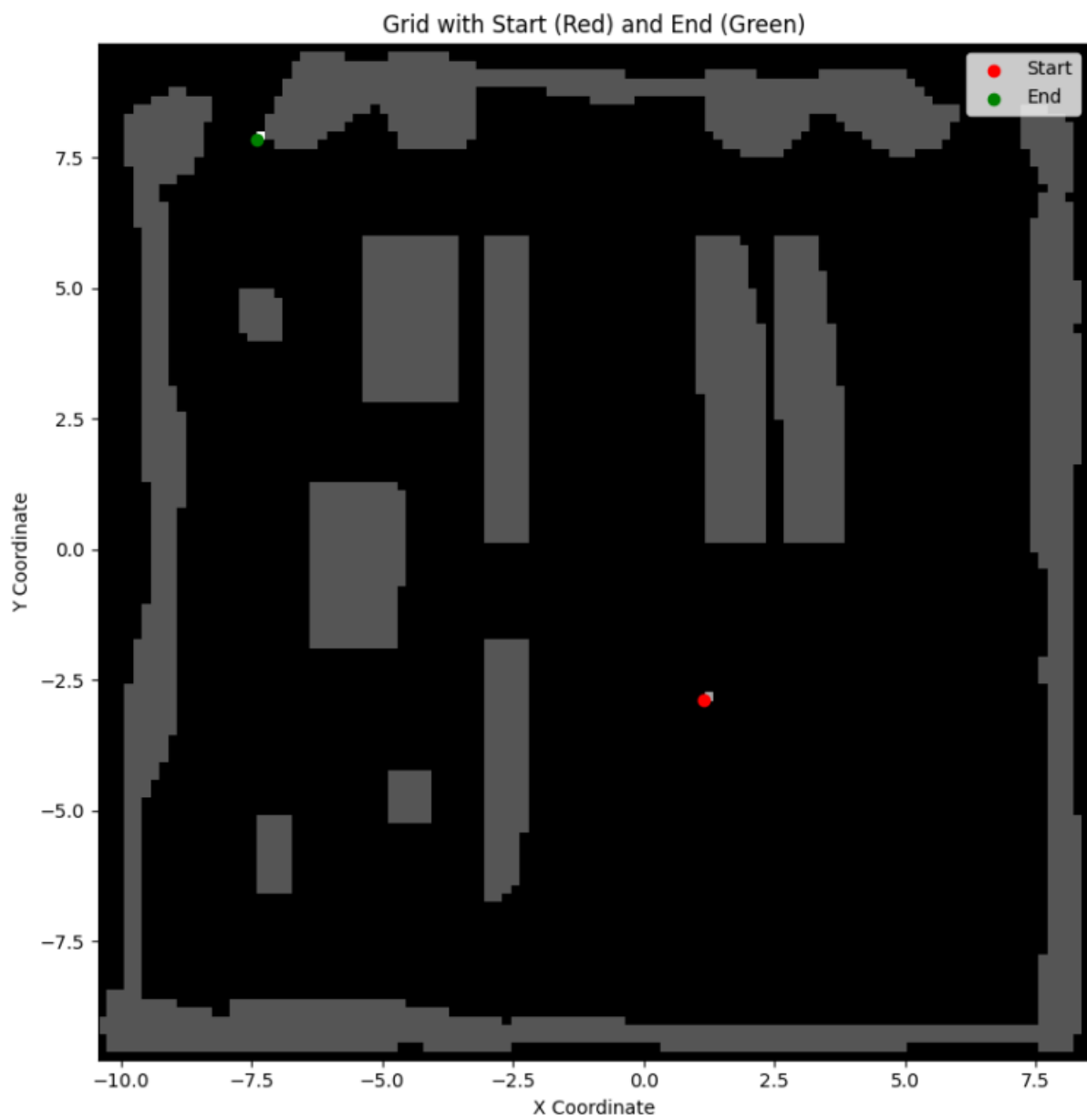
Το grid που προκύπτει αξιοποιείται για pathfinding με τη χρήση a-star και heuristic την ευκλείδεια απόσταση, το αποτέλεσμα γίνεται highlighted με μια σειρά από σφαίρες που αναπαριστούν τα βήματα.



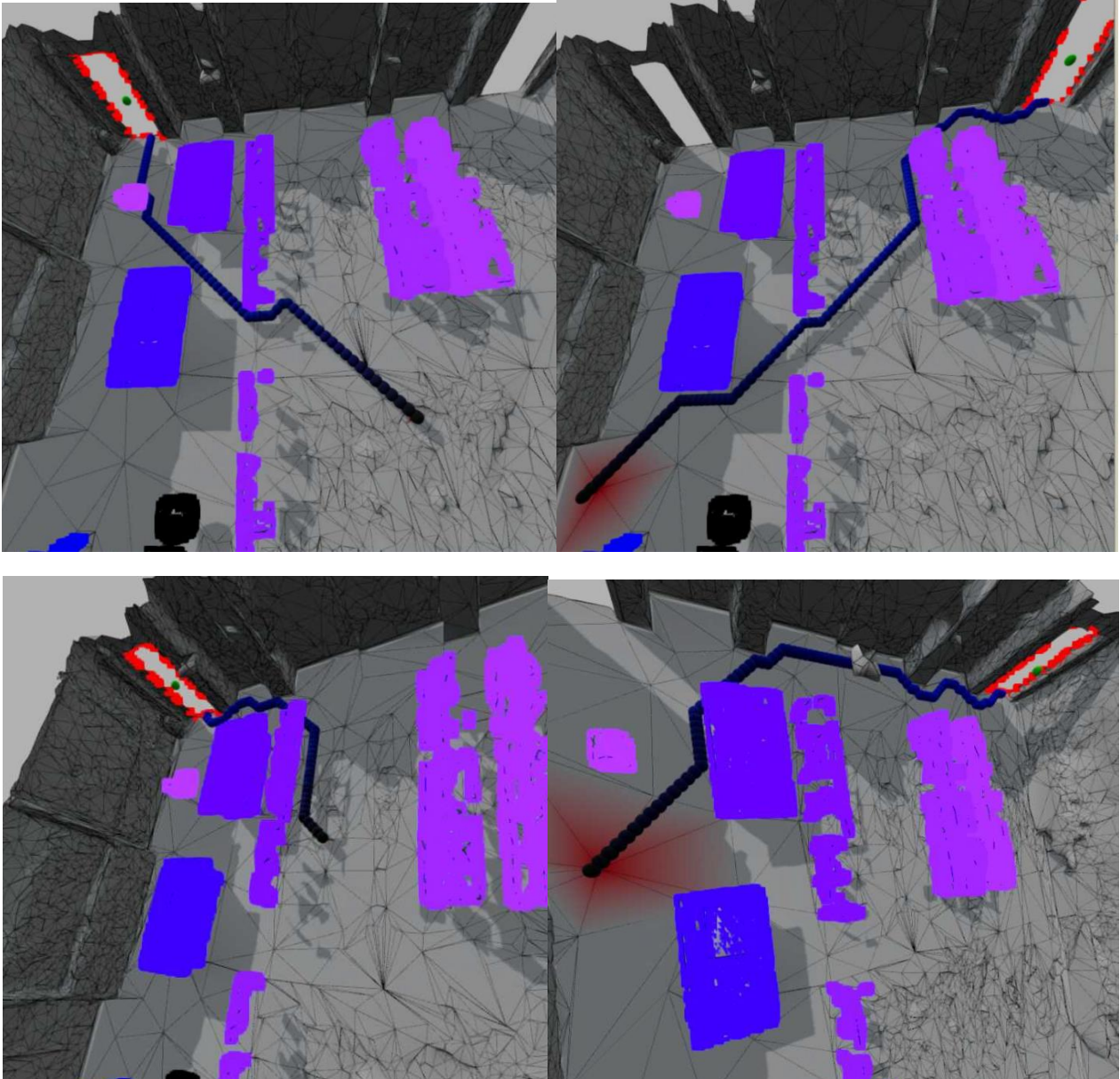
24 Αντικείμενα εντός δωματίου



25 Τμηματοποιημένοι τοίχοι δωματίου



Ακολουθούν διάφορα σενάρια pathfinding:



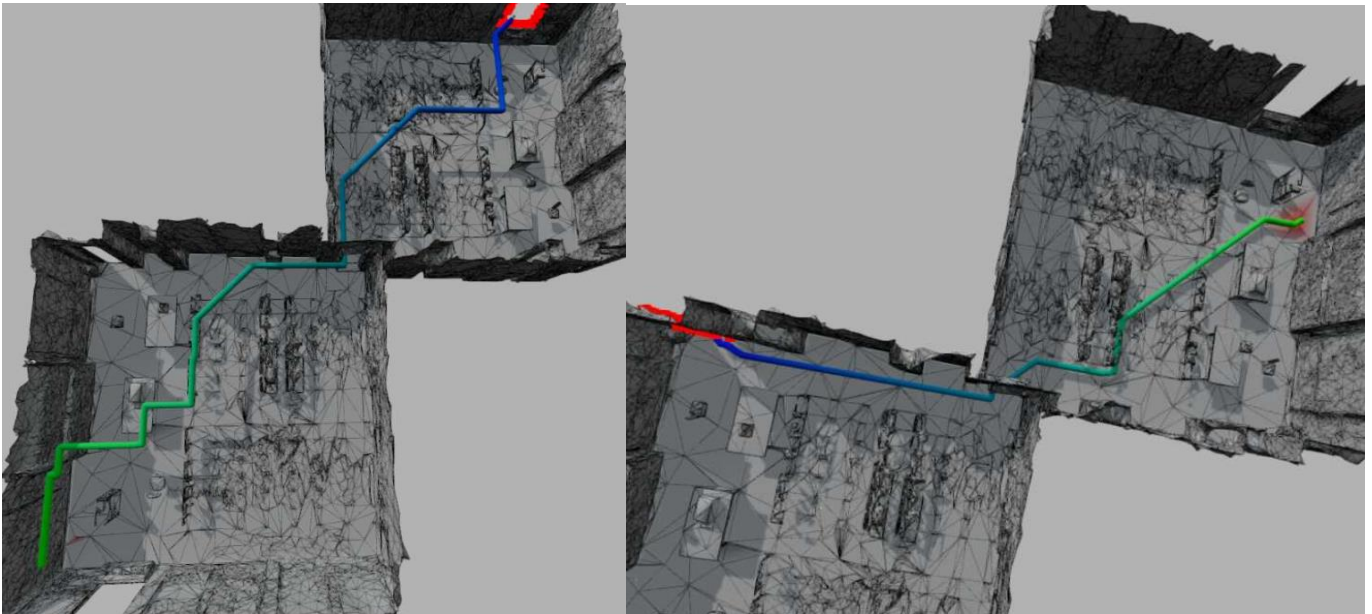
27 Pathfinding Scenarios

Ερώτημα 8

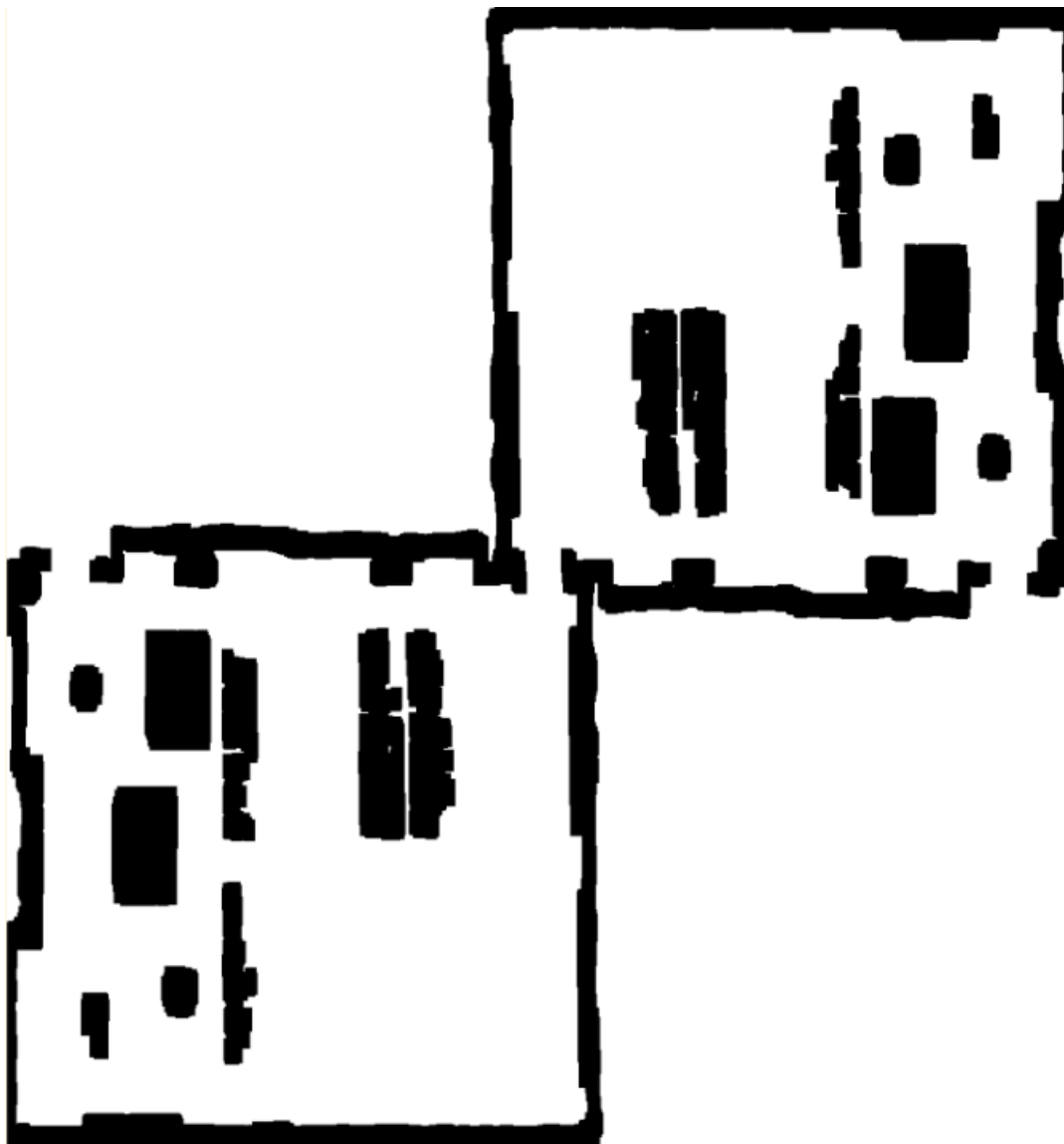
Για το ερώτημα 8 δημιουργήθηκε νέο dataset όπου ενώθηκαν 2 δωμάτια και αναθεωρήθηκε η προσέγγιση του προβλήματος και αντί του διαχωρισμού τοίχων από εσωτερικά αντικείμενα ακολουθήθηκε η παρακάτω διαδικασία:

1. Alignment του complex mesh
2. Εξαγωγή PCD
3. Αφαίρεση όλου του άνω μισού χώρου (φωτιστικά, προτζέκτορες, αετώματα πόρτας)
4. Χρήση ransac για αφαίρεση πατώματος (dominant plane)
5. Χρήση region growing για φιλτράρισμα μικρών clumps σημείων
6. Προβολή εναπομείναντων σημείων στο xy επίπεδο
7. Rasterization σε εικόνα με blobs ανά σημείο ως cached map
8. Εφαρμογή a-star με διαγώνια κίνηση στο «grid» του χάρτη

Η προσέγγιση αυτή απλοποίησε την λύση του ερωτήματος αντιμετωπίζοντας όλα τα εμπόδια ομοίως και επιτάχυνε διαδικασίες όπως το rasterization κάνοντας χρήση μεθόδων βιβλιοθήκης PIL αντί για δημιουργία grid εντός της Python.



28 Pathfinding scenarios in complex mesh



29 Χάρτης που προκύπτει για το *complex mesh*

Οδηγίες εκτέλεσης

Τα task 1-7 βρίσκονται σε ξεχωριστό zip από το 8.

Για να τρέξετε κάποιο από τα tasks κάντε extract το σχετικό zip και cd στον φάκελο του.

Τα task πρέπει να τρεχθούν με αύξουσα σειρά αν επιλέξετε να αφαιρέσετε τα prebuilt αρχεία από τον φάκελο των results.

Θα πρέπει να έχετε στο περιβάλλον σας τα:

- numpy
- scipy
- matplotlib
- scikit-learn
- open3d