ΥΠΟΛΟΓΙΣΤΙΚΗ ΓΕΩΜΕΤΡΙΑ ΚΑΙ 3Δ ΜΟΝΤΕΛΟΠΟΙΗΣΗ

PROJECT

Pointclouds

Γιώργος Μπολάτογλου

228424

Μέρος Α:

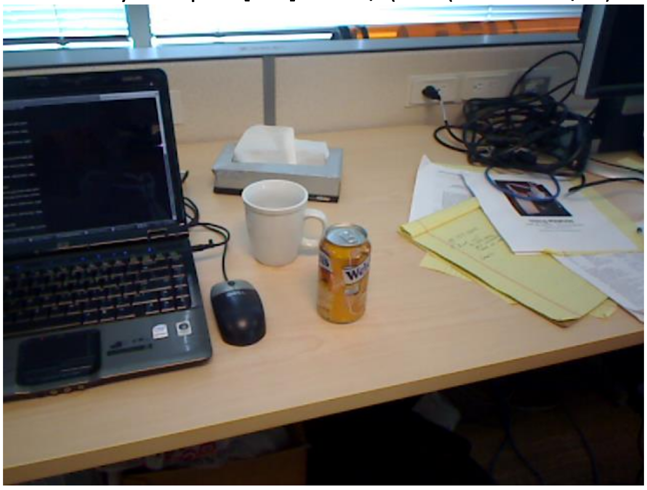
i) Φορτώστε την αλληλουχία εικόνων βάθους σε ένα διάνυσμα $Fi = \{F1, F2, \dots FN\}$ και δημιουργήστε τα αντίστοιχα 3Δ νέφη σημείων $Ci = \{C1, C2, \dots CN\}$ (Θεωρήστε κατακόρυφο/οριζόντιο οπτικό πεδίο (fov) της κάμερας 48.6ο & 62ο αντίστοιχα.). Δώστε δυνατότητα στον χρήση να απεικονίσει ένα συγκεκριμένο νέφος σημείων

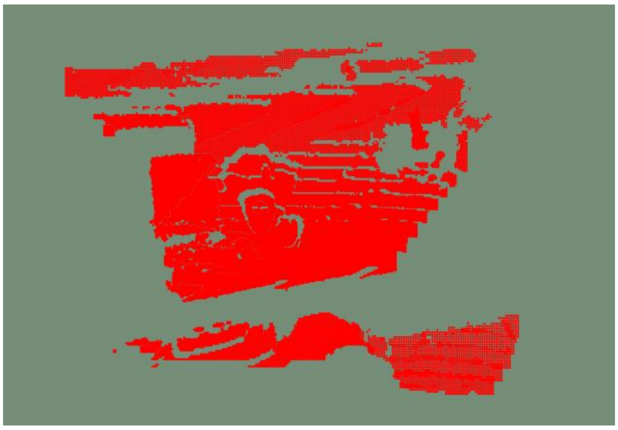
Για κάθε εικόνα χρησιμοποιούμε τους εξής μετασχηματισμούς,
όπου Vo και Uo το κεντρο της εικόνας,.

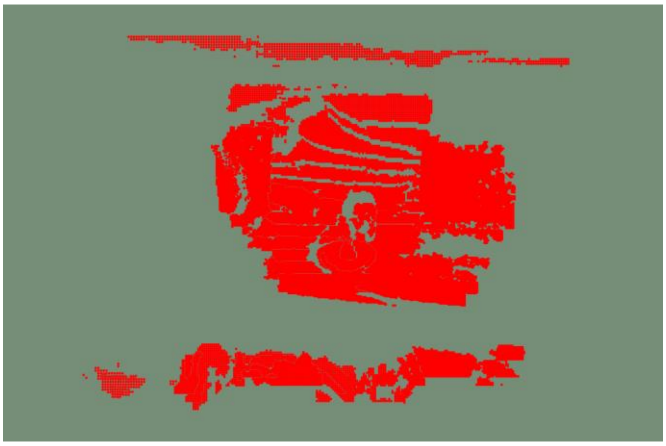
Για την τιμή του βάθους χρησιμοποιούμε τις τιμές των αντίστοιχων pixel από τις εικόνες βάθους

Έπειτα, οι αντίστοιχες τιμές των τεταγμένων στον τρισδιάστατο χώρο είναι οι εξής:

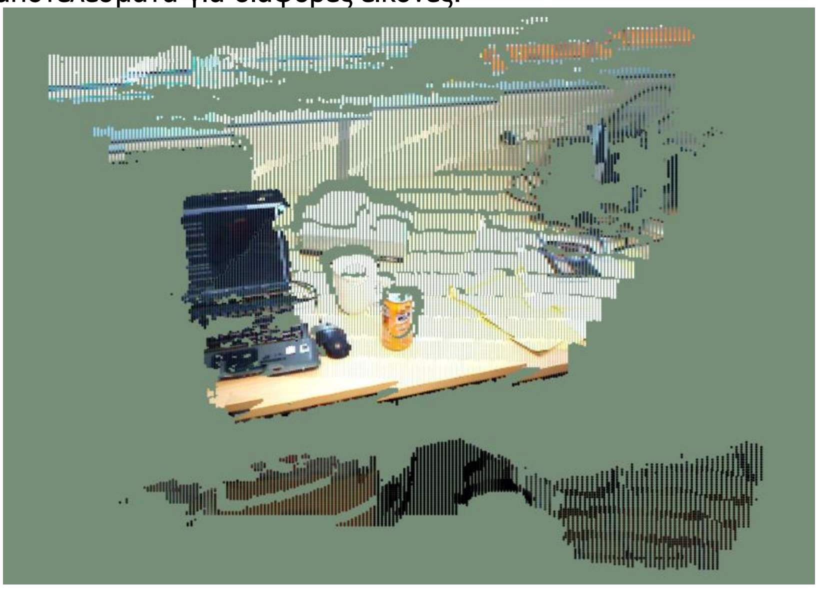
```
X = -Z * (i - Vo) / fx;
Y = Z  * (j - Uo) / fy;
Όπου,   fx = depths[cnt].cols / (tanf(fovHeight / 2) * 2);
        fy = depths[cnt].rows / (tanf(fovWidth / 2) * 2);
```

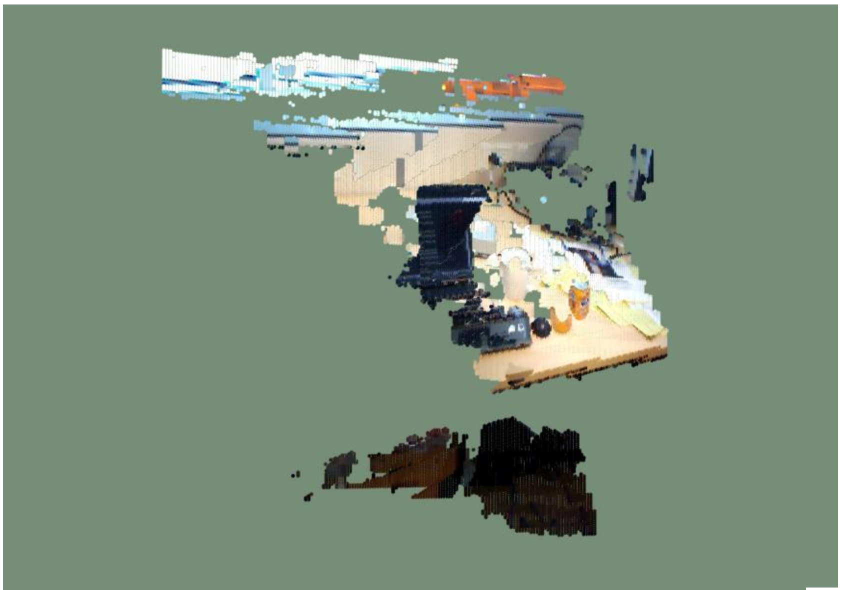


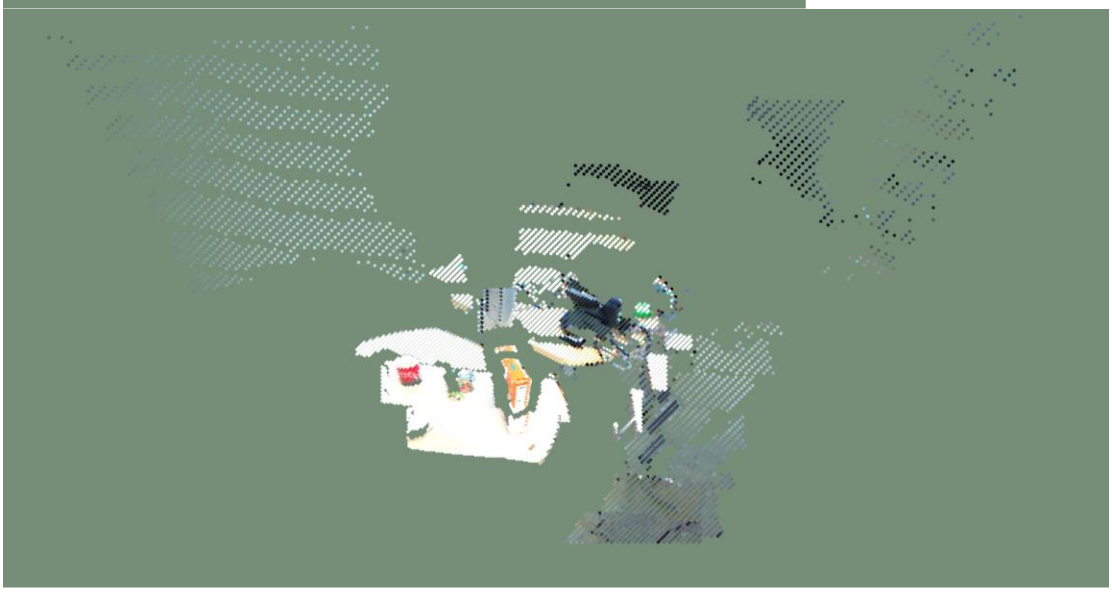




Χρησιμοποιώντας και την πληροφορία χρώματος από τις εικόνες RGB έχουμε τα εξής αποτελέσματα για διάφορες εικόνες:



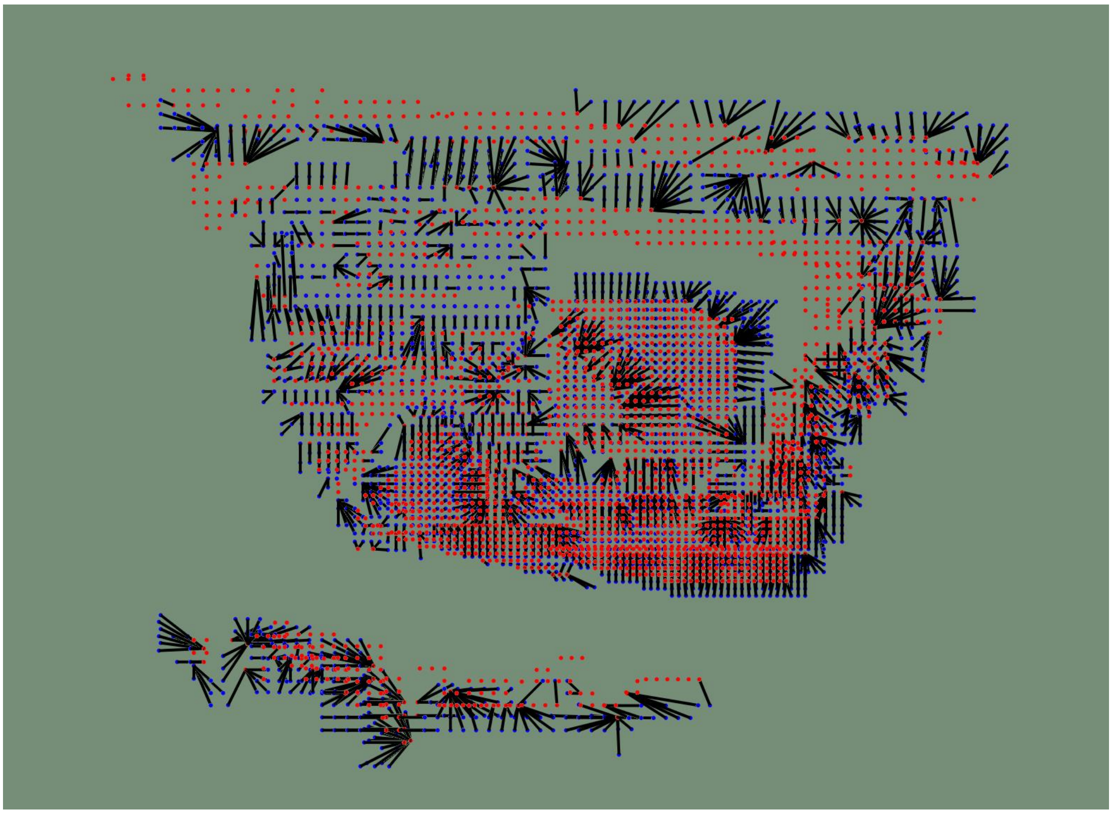


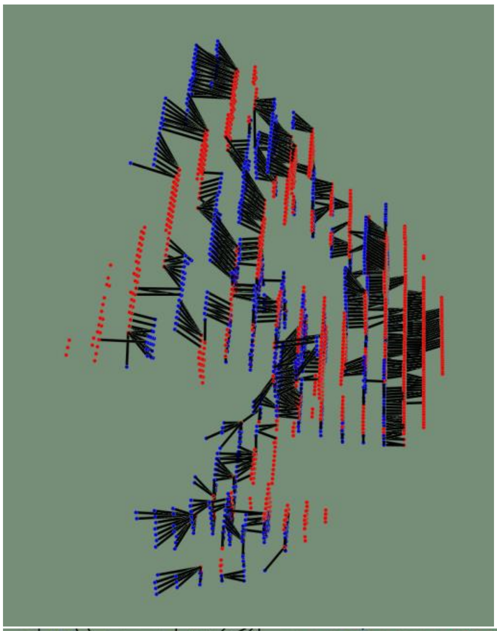


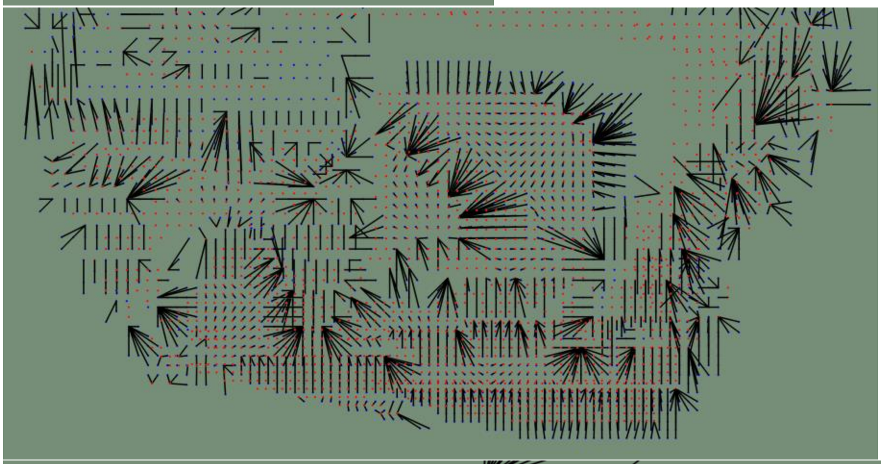
ii) Δεδομένου νέφους σημείων Ci = {$p1$, $p2$, … pn}, υπολογίστε την αντιστοίχιση σημείων με το νέφος Ci−1 = {$q1$, $q2$, … qn}, βρίσκοντας τον πλησιέστερο γείτονα για κάθε σημείο $pi \in Ci$. Απεικονίστε τα δύο (2) νέφη σημείων με διαφορετικά χρώματα καθώς και μια γραμμή για κάθε σημείο που δείχνει την αντιστοίχιση. Τυπώστε την συνολική απόσταση $e = \sum \|pi - qi\|\ n\ i$=1 2 , την μέση απόσταση, καθώς και τον χρόνο υπολογισμού της αντιστοίχισης σημείων.

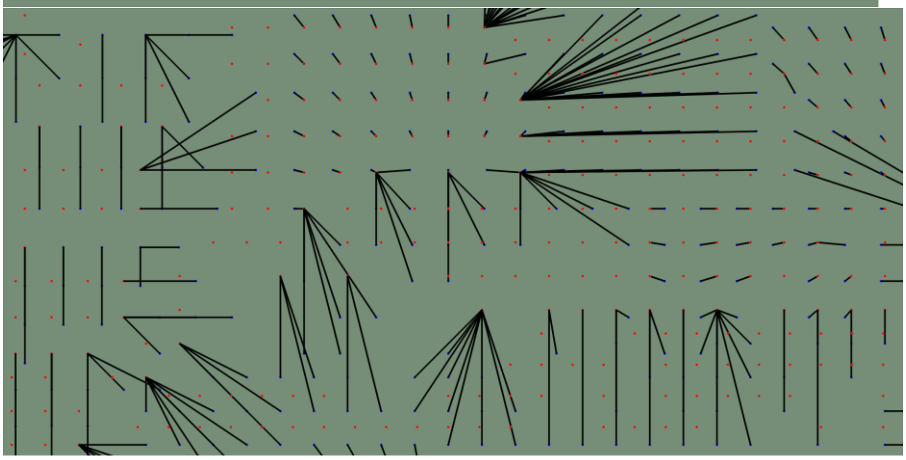
Για κάθε σημείο του νέφους Ci κάνουμε αναζήτηση στο νέφος Ci-1, βρίσκοντας κάθε φορα την απόσταση των αντίστοιχων σημείων και αν είναι μικρότερη της προηγούμενης μικρότερης απόστασης την ορίζουμε σαν την μικρότερη απόσταση εως ότου ελεγθούν όλα τα σημεία. Σε κάθε επανάληψη σαν αρχική μικρότερη απόσταση ορίζουμε την μεγαλύτερη δυνατή. Τελικά, το κάθε σημείο του Ci και το αντίστοιχο σημείο με την μικρότερη απόσταση τα αποθηκεύουμε διαδοχικά σε έναν πίνακα (θέση n το σημείο του Ci, θέση n+1 το αντίστοιχο σημείο με την μικρότερη απόσταση του Ci-1) , από τον οποίο και θα δημιουργηθούν τα αντίστοιχα ευθύγραμμα τμήματα.

Desk_1
Image = 30
Step = 10
total_time = 17.637
total_dist = 2441.14
ave_dist = 0.875274



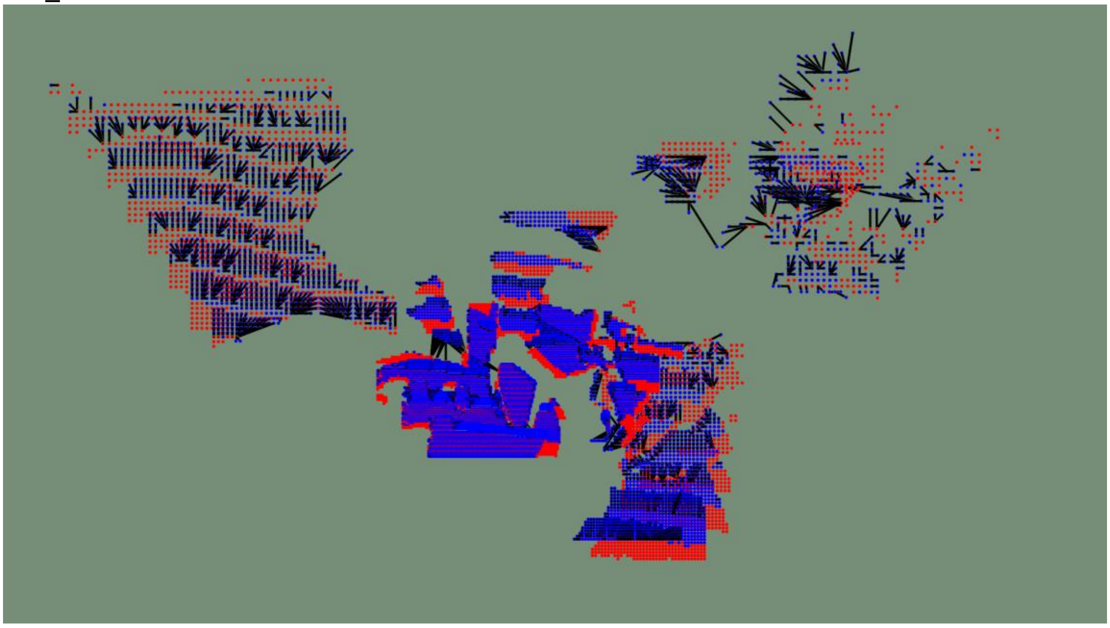






table_small_1
Image = 30
Step = 5
total_time = 327.702
total_dist = 1018.66

ave_dist = 0.0914335



Μέρος Β:

iii) Δημιουργήστε μια συνάρτηση η οποία υπολογίζει την περιστροφή $\boldsymbol{R} \in \mathbb{R}3x3$ και την μετατόπιση $\boldsymbol{t}$ ώστε να ευθυγραμμίζει τα νέφη σημείων του προηγούμενου ερωτήματος, δηλαδή ελαχιστοποιεί την συνάρτηση $e = \sum \|(\boldsymbol{R}pi + \mathbf{t}) - qi\| \; n \; i$=1 2 .

Τώρα, διασπάμε τον πίνακα με τα αντίστοιχα διαδοχικά αποθηκευμένα σημεία ελάχιστης απόστασης του προηγούμενου ερωτήματος σε δύο αντίστοιχους πίνακες A(νέφος Ci) και B(αντίστοιχα σημεία του Ci-1). Θέλουμε να περιστρέψουμε των Ci-1 ώστε να συμπέσει όσο γίνεται καλύτερα με τον Ci.
Αρχικά υπολογίζουμε τα αντίστοιχα κέντρα p και q των αντίστοιχων νεφών, όπως φαίνεται παρακάτω:

Έπειτα υπολογίζουμε τους πίνακες X και Y όπου κάθε σημείο τους είναι το αντίστοιχο των A και B αφαιρώντας το αντίστοιχο κέντρο.
Χρησιμοποιώντας την βιβλιοθήκη της Eigen υπολογίζουμε τον SVD του S = XWYT , όπου W ο πίνακας με τα βάρη τα οποία όλα έχουν την τιμή 1, και έχουμε σαν έξοδο δύο πίνακες τον U και τον V.
Τέλος,

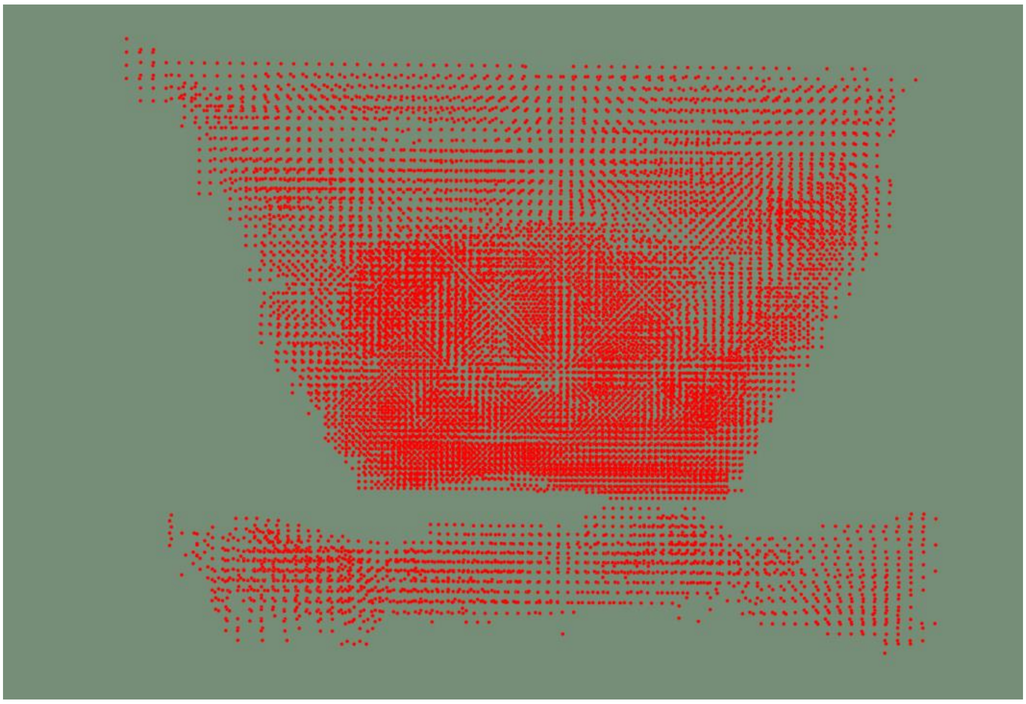
- η ιδανικότερη περιστροφή είναι R = V * d * UT , όπου
  d μοναδιαίος διαγώνιος πίνακας με τιμή στο τελευταίο του κελί 1 ή -1 (ανάλογα την τιμή της ορίζουσας του V*Ut).
- Η ιδανικότερη μετατόπιση είναι t = q – R* p .

Έχοντας ,λοιπόν, υπολογίσει την ιδανικότερη περιστροφή και μετατόπιση του Ci-1 την εφαρμόζουμε σε κάθε του σημείο A[i] ως εξής A_new[i] = R * A[i] + t .

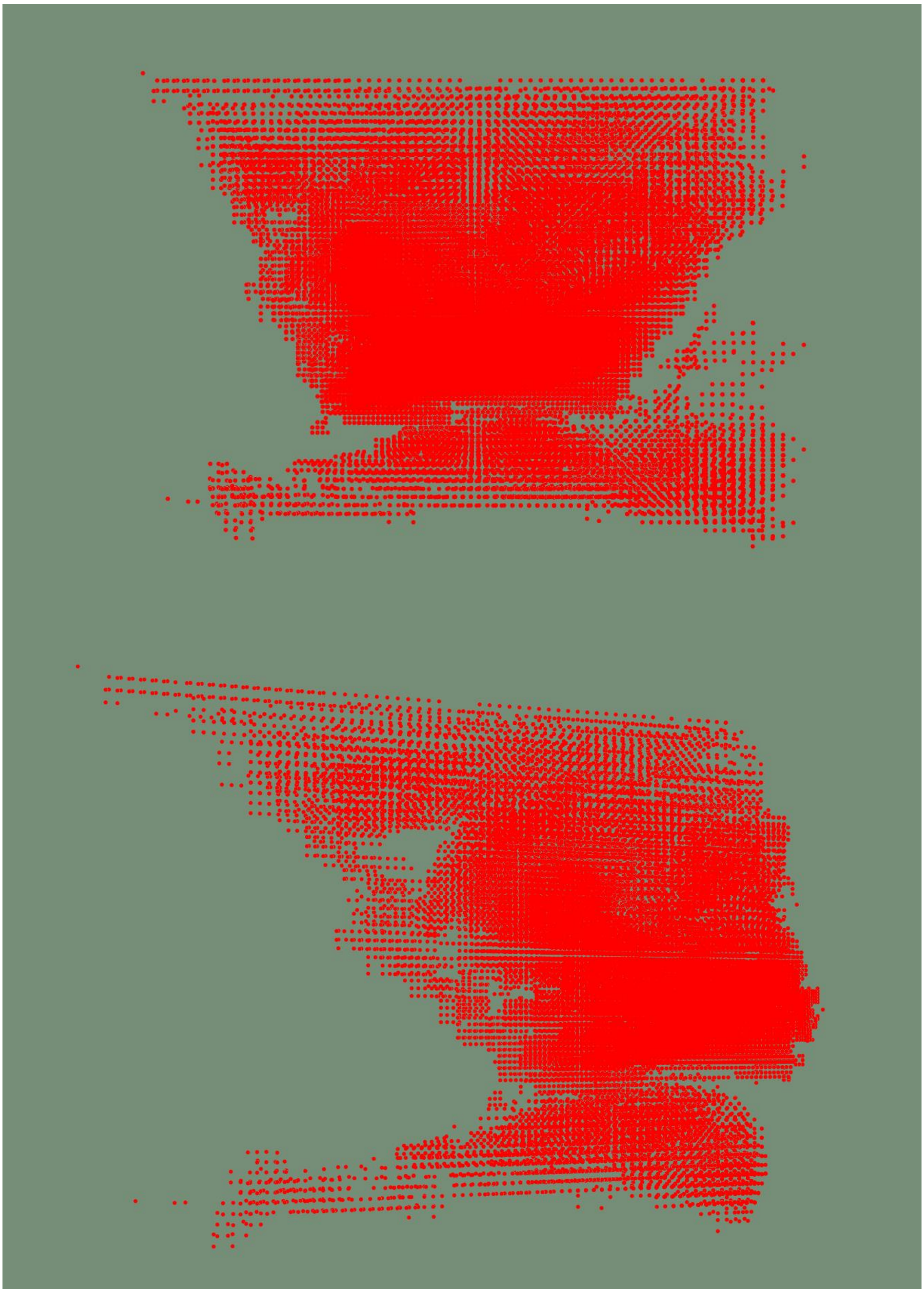
iv) Χρησιμοποιήστε την συνάρτηση του προηγούμενου ερωτήματος για να ευθυγραμμίσετε τα νέφη Ci έως Cj για i,j είσοδο από τον χρήστη (π.χ. $C2$ έως $C24$). Δημιουργήστε το ολικό νέφος M ως σύνολο των ευθυγραμμισμένων νεφών αφαιρώντας τα διπλά σημεία μετά την ευθυγράμμιση, δηλαδή τα σημεία που έχουν απόσταση από το αντίστοιχο τους $d < \varepsilon$, όπου ε κάποια αυθαίρετη ανοχή. Απεικονίστε το ολικό νέφος M και τυπώστε τον χρόνο υπολογισμού του για διάφορες

Χρησιμοποιούμε την συνάρτηση του προηγούμενου ερωτήματος για να βρούμε όλες τις επιμέρους περιστροφές και μετατοπίσεις των αντίστοιχων νεφών. Έπειτα, η περιστροφή του κάθε νέφους πολλαπλασιάζεται με όλες τις προηγούμενες περιστροφές των προηγούμενων νεφών. Για παράδειγμα, αν θέλουμε να ευθυγραμμίσουμε μία σκήνη με νέφη Ci όπου i = 1,2,3,4,5 , τότε αφού υπολογίσουμε όλες τις περιστροφές Ri, η περιστροφή του C4 θα είναι tota_R = R4, του C3 θα είναι tota_R = R3*R4, του C2 tota_R =R2*R3*R4, του C1 tota_R =R1*R2*R3*R4. Έτσι, εφαρμόζοντας αυτην την περιστροφή και την αντίστοιχη μετατόπιση σε κάθε σημείο του αντίστοιχου νέφους έχουμε το ολικό ευθυγραμμισμένο νέφος.

Αποθηκεύουμε κάθε σημείο σε έναν άλλο πίνακα προσπερνώντας μόνο τα σημεία που έχουν απόσταση από τα σημεία που έχουμε ήδη αποθηκεύσει μικρότερη του ε.
Desk_1
2 -> 25
Step = 5
kd_total = 309.07
whole = 368.604

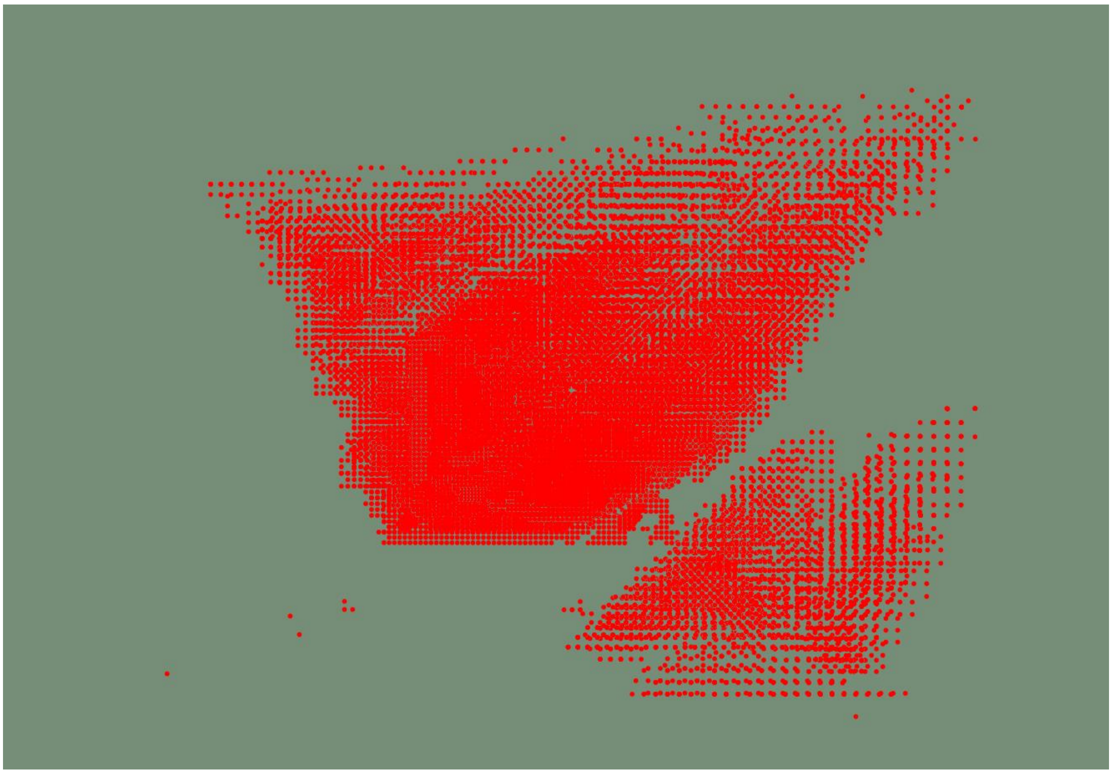


20 -> 30
kd_total = 212.905
whole = 246.102
no_duplicates.size() = 14638
delet_duplicates_time = 501.063
m_POINTCLOUD.size() = 30383



2 -->25
kd_total = 398.998
whole = 460.431

60->80 desk_2
kd_total = 288.327
whole = 332.82



v) Χρησιμοποιήστε μια δομή δεδομένων διάτμησης χώρου (octree, kd-tree κτλ.) για να επιταχύνετε την αναζήτηση του πλησιέστερου γείτονα και επαναλάβετε το προηγούμενο ερώτημα. Δείξτε τον βαθμό επιτάχυνσης του αλγορίθμου. Τι περιορισμούς μας δημιουργεί?

Χρησιμοποιώντας το KD-Tree οι αντίστοιχα χρόνοι υπολογισμού για διάφορα πλήθη σημείων(ανάλογα το step που επιλέγουμε) των πλησιέστερων γειτόνων αναφέρονται παρακάτω:
m_pointclouds[image - 1].size() = 689
total_time = 0.149

m_pointclouds[image - 1].size() = 1223
total_time = 0.356

m_pointclouds[image - 1].size() = 2768
total_time = 1.936

m_pointclouds[image - 1].size() = 5757
total_time = 10.811

m_pointclouds[image - 1].size() = 11258
total_time = 35.903

m_pointclouds[image - 1].size() = 31345
total_time = 312.993
Riza n

KD-tree στον υπολογισμό των ευθυγραμμισμένων νεφων:

20 to 30 desk_1
kd_total = 26.544
whole = 60.055

60 το 80 desk_2
whole = 1350.54

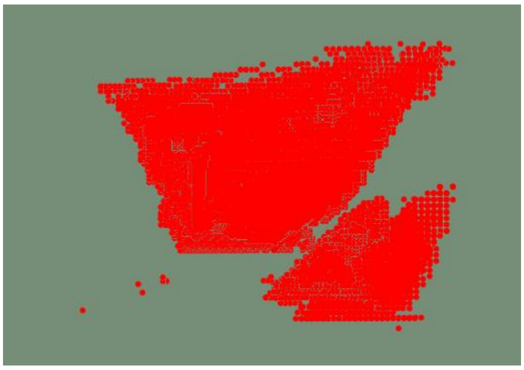
vi) Εφαρμόστε φίλτρο Sobel στις εικόνες χρώματος, και αφού απορρίψετε χαμηλές τιμές με χρήση κατωφλιού (thresholding), χρησιμοποιήστε μόνο τα αντίστοιχα σημεία για την βελτιστοποίηση της συνάρτησης του ερωτήματος (iii). Δείξτε το αποτέλεσμα και τον βαθμό επιτάχυνσης του αλγορίθμου.
Εργαζόμαστε ως εξής:

1. Μετατρέπουμε την εικόνα σε grayscale.
2. Κάνουμε gradient τοσο οριζόντια όσο και κατακόρυφα .
3. Τα προσθέτουμε και τα δύο gradient ισόβαρα παίρνοντας την τελική φωτογραφία.

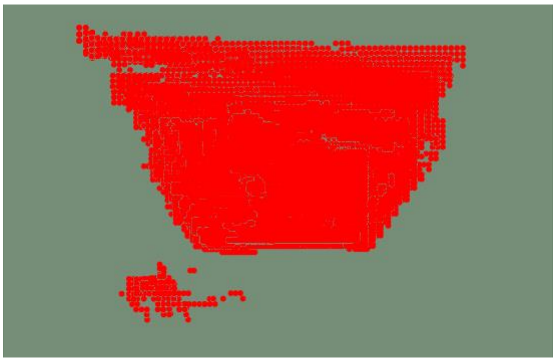
4. Κάνοντας χρήση κατωφλίου με τιμή 20 παίρνουμε μόνο τα pixel με τιμές πάνω από 20, δημιουργώντας έτσι το καινούριο νέφος σημείων με μόνο αυτά τα σημεία.

5. Επαναλαμβάνουμε τον αλγόριθμο των παραπάνω ερωτημάτων χρησιμοποιώντας τώρα τα νέφη σημείων που έχουν προκύψει από την εφαρμογή του sobel και βρίσκουμε τις αντίστοιχες μετατοπίσεις και περιστροφές, τις οποίες και εφαρμόζουμε στα αρχικά νέφη σημείων.

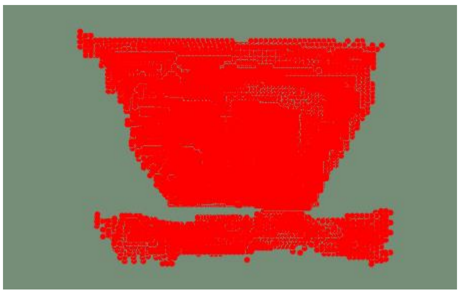
Με αυτόν τον τρόπο πετυχαίνουμε εξίσου καλά αποτελέσματα χρησιμοποιώντας πολύ λιγότερα σημεία και άρα εξοικονομώντας υπολογισμούς και χρόνο.
Desk_1
20->30
kd_total = 11.961
whole = 15.541



Desk_2
20->30
kd_total = 13.861
whole = 17.814



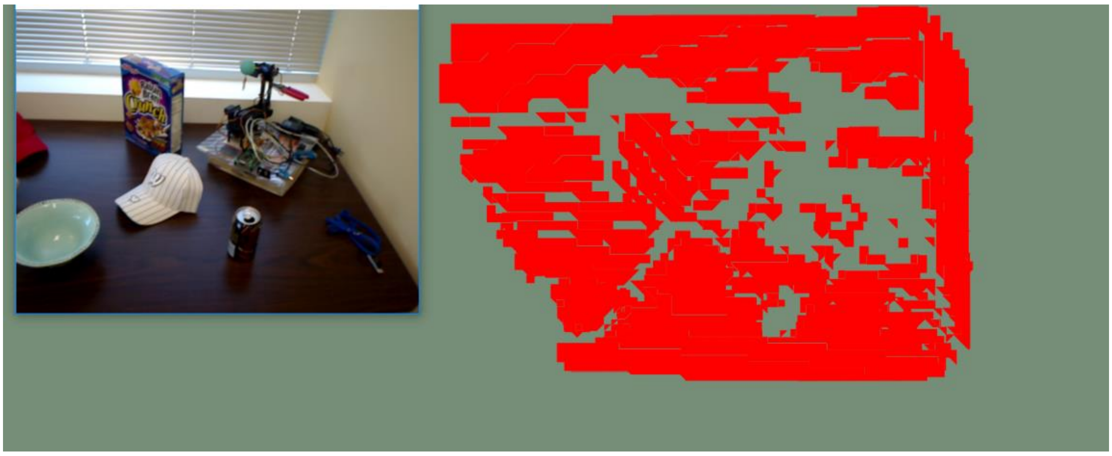
60->80
kd_total = 16.656
whole = 23.331



vii) Δημιουργήστε την τριγωνοποιημένη επιφάνεια του ολικού νέφους M, λαμβάνοντας υπόψη το γεγονός ότι τα σημεία είναι στοιχισμένα (δηλαδή δομημένα από πίνακα pixel) και δείξτε το αποτέλεσμα για διάφορες αλληλουχίες. Μπορούμε να χρησιμοποιήσουμε την πληροφορία χρώματος?

Βρίσκω ξεχωριστά την τριγωνοποιημένη επιφάνεια του κάθε νέφους και έπειτα του εφαρμόζω την αντίστοιχη περιστροφή και την μετατόπιση που έχω υπολογίσει σε αυτό καταλήγοντας στην τελική τριγωνοποιημένη επιφάνεια.
Για το κάθε νέφος βρίσκω όλα τα διαφορετικά επίπεδά του και τα τριγωνοποιώ ξεχωριστά λαμβάνοντας υπόψη την δυαδική του στοίχιση.

Desk_1
20 to 30



Την πληροφορία χρώματος μπορούμε να την χρησιμοποιήσουμε για να δώσουμε χρώμα στα τρίγωνα, όπως και στο ερώτημα ένα όπου χρωματίστηκαν τα σημεία.