



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ

Αναγνώριση αντικειμένων με
προσδιορισμό θέσης και περιστροφής με
χρήση RGB-D δεδομένων έπειτα από
μάθηση με μοντέλα 3D-CAD

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΒΑΣΙΛΕΙΟΥ ΜΑΝΣΟΛΑ

Επιβλέπων: Κόλλιας Στέφανος
Καθηγητής Ε.Μ.Π.

ΕΡΓΑΣΤΗΡΙΟ ΨΗΦΙΑΚΗΣ ΕΠΕΞΕΡΓΑΣΙΑΣ ΕΙΚΟΝΑΣ, ΒΙΝΤΕΟ ΚΑΙ ΠΟΛΥΜΕΣΩΝ
Αθήνα, Ιανουάριος 2020



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Ψηφιακής Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων

Αναγνώριση αντικειμένων με
προσδιορισμό θέσης και περιστροφής με
χρήση RGB-D δεδομένων έπειτα από
μάθηση με μοντέλα 3D-CAD

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

ΒΑΣΙΛΕΙΟΥ ΜΑΝΣΟΛΑ

Επιβλέπων: Κόλλιας Στέφανος
Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 05/02/2020

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....
Κόλλιας Στέφανος
Καθηγητής Ε.Μ.Π.

.....
Ανδρέας Σταφυλοπάτης
Καθηγητής Ε.Μ.Π.

.....
Γιώργος Στάμου
Καθηγητής Ε.Μ.Π.

Αθήνα, Ιανουάριος 2020

(Υπογραφή)

.....

ΜΑΝΣΟΛΑΣ ΒΑΣΙΛΕΙΟΣ

Διπλωματούχος Ηλεκτρολόγος Μηχανικός και Μηχανικός Υπολογιστών
Ε.Μ.Π.

© 2020 – All rights reserved



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Τομέας Τεχνολογίας Πληροφορικής και Υπολογιστών
Εργαστήριο Ψηφιακής Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων

Copyright ©–All rights reserved Μανσόλας Βασίλειος, Μάρτιος 2020.

Με επιφύλαξη παντός δικαιώματος.

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα. Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

Περίληψη

Αναγνώριση αντικειμένων, στη επιστήμη της όρασης υπολογιστών, είναι η διαδικασία της εύρεσης και προσδιορισμού του είδους των αντικειμένων, μέσα σε μία εικόνα ή αλληλουχία εικόνων (βίντεο). Ο ανθρώπινος εγκέφαλος σε συνδυασμό με τους αισθητήρες που διαθέτει ο άνθρωπος για να αντιλαμβάνεται τον περιβάλλοντα χώρο, μπορούν με ευκολία να αναγνωρίσουν και να προσδιορίσουν το είδος τεράστιας ποικιλίας και αριθμού αντικειμένων, παρά το γεγονός πως αυτά ποικίλουν στην οπτική γωνία παρατήρησης, σε μέγεθος και κλίμακα, και μπορεί να είναι μετατοπισμένα ή περιστραμμένα. Το αντικείμενο μελέτης αυτό αποτελεί μεγάλη πρόκληση για την επιστήμη των υπολογιστών, δηλαδή αυτή η ευκολία του ανθρώπου στην αναγνώρισή, να γίνει εφικτή μέσω ηλεκτρονικών υπολογιστών.

Η 3D αναγνώρισή περιλαμβάνει τόσο την αναγνώριση, όσο και τον προσδιορισμό της θέσης και στάσης στο χώρο από φωτογραφίες ή από 3D σάρωση του χώρου. Τυπικά, ένα παράδειγμα αντικειμένου παρουσιάζεται στο σύστημα σε ένα ελεγχόμενο περιβάλλον, και στη συνέχεια με είσοδο δεδομένων, εικόνα ή βίντεο, το σύστημα προσδιορίζει την τοποθεσία του προς αναγνώρισή αντικειμένου. Η Διαδικασία αυτή μπορεί να λάβει χώρα μετά από την απόκτηση των πληροφοριών ή σε πραγματικό χρόνο. Εξαιτίας του χαμηλού κόστους και της ευκολίας απόκτησης 3D σαρώσεων, μεγάλο κομμάτι έρευνας έχει αποδοθεί τα τελευταία χρόνια στην αναγνώρισή με 3D μέσα.

Στόχος αυτής της διπλωματικής εργασίας είναι η μελέτη και υλοποίησή ενός συστήματος αναγνώρισης αντικειμένων και προσδιορισμού της στάσης τους στο χώρο, εκμεταλλευόμενοι πληροφορία για το βάθος που προέρχεται από 3D αισθητήρα - κάμερα. Κάθε αντικείμενο περνά από διαδικασία μάθησης μέσω μιας 3D αναπαράστασης του. Το σύστημα με είσοδο μία σάρωση του χώρου και απομονώνοντάς το κάθε αντικείμενο από το περιβάλλον του, υπολογίζει το είδος του αντικειμένου και τη θέση του.

Λέξεις Κλειδιά

Αναγνώριση αντικειμένων, Εκτίμηση στάσης, 3D αναπαράσταση, RGB-D αισθητήρας, 3D-CAD

Abstract

Object recognition - in computer vision is the task of finding and identifying objects in an image or video sequence. Humans recognize a multitude of objects in images with little effort, despite the fact that the image of the objects may vary somewhat in different view points, in many different sizes / scale or even when they are translated or rotated. Objects can even be recognized when they are partially obstructed from view. This task is still a challenge for computer vision systems trying to develop methods and systems that mimic this ease of human recognition using computers.

3D object recognition involves recognizing and determining the pose of user-chosen 3D object in a photograph or a range 3D scan. Typically, an example of the object to be recognized is presented to a vision system in a controlled environment, and then for an arbitrary input such as a video stream, the system locates the previously presented object. This can be done either off-line, or in real-time. Due to the low cost and ease of acquiring 3D scans, a significant amount of research has been devoted to 3D object recognition during the last years.

In this diploma thesis, we present an object recognition and pose estimation algorithmic pipeline, exploiting depth information. The target region is cleanly segmented from background, and then it can estimate pose of a target object.

Keywords

Object Recognition, Pose Estimation, 3D representation, RGB-D camera scanner, 3D-CAD

Ευχαριστίες

Θα ήθελα καταρχήν να ευχαριστήσω τον καθηγητή κ.Κόλλια για την επίβλεψη αυτής της διπλωματικής εργασίας και για την ευκαιρία που μου έδωσε να την εκπονήσω στο εργαστήριο Ψηφιακής Επεξεργασίας Εικόνας, Βίντεο και Πολυμέσων. Επίσης ευχαριστώ ιδιαίτερα τον Δρ. κ.Καρπούζη για την καθοδήγησή του και την εξαιρετική συνεργασία που είχαμε. Τέλος θα ήθελα να ευχαριστήσω τους γονείς μου για την καθοδήγηση και την ηθική συμπαράσταση που μου προσέφεραν όλα αυτά τα χρόνια.

Περιεχόμενα

Περίληψη	1
Abstract	3
Ευχαριστίες	5
Περιεχόμενα	9
Κατάλογος Σχημάτων	12
Κατάλογος Πινάκων	13
1 Εισαγωγή	15
1.1 Αντικείμενο της διπλωματικής	15
1.2 Οργάνωση του τόμου	15
2 Θεωρητικό υπόβαθρο	17
2.1 Ψηφιακή Εικόνα	17
2.2 Τρισδιάστατη Αναπαράσταση	18
2.2.1 Αισθητήρας - κάμερα 3D	19
2.2.2 Λήψη δεδομένων βάθους	24
2.2.3 Χάρτης Βάθους	26
2.2.4 Point Clouds	26
2.3 Γεωμετρικά χαρακτηριστικά - Features	27
2.3.1 Local Features	28
2.3.2 Global Features	28
2.3.3 PFH	29
2.3.4 FPFH	30
2.3.5 VFH	30
2.3.6 CVFH	31
2.3.7 OUR-CVFH	33

2.4	Εκπαίδευση - Μάθηση Συστήματος	33
2.5	Διάτμιση - Segmentation	35
2.5.1	Κατοφλιοποίηση	35
2.5.2	Τμηματοποίηση σε ομάδες	35
2.5.3	Τμηματοποίησή σε τρεις διαστάσεις	38
2.6	Αναγνώριση -Recognition	39
3	Ανάλυση και σχεδίαση	41
3.1	Περιγραφή Αρχιτεκτονικής	41
3.2	Εκπαίδευση	42
3.3	Global Pipeline	44
3.3.1	Συλλογή Δεδομένων	44
3.3.2	Μετατροπή σε νέφος σημείων	44
3.3.3	Προεπεξεργασία	46
3.3.4	Φιλτράρισμα	46
3.3.5	Αναγνώριση επιφανειών	47
3.3.6	Διάτμιση	47
3.3.7	Εξαγωγή χαρακτηριστικών	47
3.3.8	Αναγνώριση	47
4	Υλοποίηση	49
4.1	Λεπτομέρειες υλοποίησης	49
4.1.1	Περιβάλλον Υλοποίησης	49
4.1.2	Γλώσσες Προγραμματισμού - Βιβλιοθήκες	49
4.2	Περιγραφή αλγορίθμων	52
4.2.1	Υπολογισμός indices	52
4.2.2	Αφαίρεση σημείων χωρίς δεδομένα (NaNs)	53
4.2.3	Υπολογισμούς του κεντρικού σημείου ενός νέφους (centroid)	53
4.2.4	Υπολογισμός features	54
4.2.5	Υπολογισμός Normals	54
4.2.6	Decomposition	55
4.2.7	Δέντρο k διαστάσεων (k-d tree)	55
4.2.8	Δέντρο Octree	57
4.2.9	Φιλτράρισμα	59
4.2.10	Φίλτρο Pass-through	59
4.2.11	Αφαίρεση οριακών σημείων	62
4.2.12	Επαναδειγματοληψία (resampling)	63
4.2.13	Υποδειγματοληψία (Downsampling)	63
4.2.14	Υπερδειγματοληψία (Upsampling)	65

4.2.15 Plane Segmentation	67
5 Αποτελέσματα - Σύγκριση Αποτελεσμάτων	71
5.1 Εκπαίδευση συστήματος	71
5.1.1 Βιβλιοθήκη αντικειμένων	71
5.1.2 Εξαγωγή σχημάτων από διάφορες οπτικές γωνίες	71
5.1.3 Χώρος σάρωσης	75
5.1.4 Αποτελέσματα αναγνώρισης	77
5.1.5 Διάγραμμα Χαρακτηριστικών	86
6 Επίλογος	87
6.1 Επεκτάσεις	87
6.2 Περιορισμοί	87
6.3 Συμπεράσματα	88
Βιβλιογραφία	90

Κατάλογος Σχημάτων

2.1	Εικονοστοιχεία - pixels	18
2.2	3D αναπαράσταση	18
2.3	Τρισδιάστατες Κάμερες	19
2.4	Αισθητήρας με δύο κάμερες	20
2.5	Αισθητήρας LIDAR	21
2.6	Αισθητήρες Time-of-flight	22
2.7	Χάρτης Βάθους	23
2.8	αισθητήρας Kinect	23
2.9	Υπολογισμός βάθους	25
2.10	Σφάλμα μέτρησης	26
2.11	Point clouds	27
2.12	Περιγραφέας PFH	29
2.13	Περιγραφέας FPFH	30
2.14	Περιγραφέας VFH	31
2.15	Περιγραφέας VFH	31
2.16	Περιγραφέας CVFH	32
2.17	Περιγραφέας OUR-CVFH	33
3.1	41
3.2	42
3.3	43
3.4	44
3.5	Σκηνή και νέφος Σημείων	45
4.1	Υπολογισμός Normals	55
4.2	56
4.3	58
4.4	Δέντρο Octree	59
4.5	Φίλτρο Pass-through	61
4.6	Downsampling	65

4.7	Upsampling	67
4.8	Plane Segmentation	69
5.1	Βιβλιοθήκη αντικειμένων	72
5.2	Εξαγωγή σκηνών	73
5.3	Δημιουργία εικονικών σκηνών από τα 3Δ αντικείμενα	74
5.4	Χώρος σάρωσης	75
5.5	Χώρος σάρωσης (2)	76
5.6	Το φανάρι αναγνωρίζεται από 3 διαφορετικές στάσεις	77
5.7	Το φανάρι αναγνωρίζεται από διαφορετικές στάσεις ακόμη και σε επι- κάλυψη στον ζ άξονα	78
5.8	Τρία αντικείμενα στην επιφάνεια	79
5.9	Αναγνώριση μικρού αντικειμένου σε διαφορετικές στάσεις	80
5.10	Σφάλμα αναγνώρισης	81
5.11	Τα δύο όμοια αντικείμενα αναγνωρίζονται ξεχωριστά ως φανάρια	82
5.12	Δύο αντικείμενα στην επιφάνεια	82
5.13	Δύο αντικείμενα στην επιφάνεια με συμμετρία	83
5.14	Αντικείμενα που δεν υπάρχουν στη βάση	84
5.16	Διάγραμμα Χαρακτηριστικών	86

Κατάλογος Πινάκων

Κεφάλαιο 1

Εισαγωγή

1.1 Αντικείμενο της διπλωματικής

Αντικείμενο αυτής της διπλωματικής εργασίας είναι η μελέτη και υλοποίησή ενός συστήματος αναγνώρισης αντικειμένων και προσδιορισμού της στάσης και θέσης τους, χρησιμοποιώντας πληροφορία προερχόμενη από εικόνες βάθους που αποτυπώθηκε ψηφιακά με χρήση 3D αισθητήρα - κάμερα.

Η αναπαράσταση κάθε αντικειμένου και του περιβάλλοντα χώρου γύρω από τα αντικείμενα γίνεται με ένα νέφος από σημεία(Point clouds). Το σύστημα εκπαιδεύεται με την εισαγωγή 3D μοντέλων αντικειμένων που έχουν σχεδιαστεί σε υπολογιστή, και 'μαθαίνει' να τα αναγνωρίζει, χωρίς όμως να έχει προηγηθεί μάθηση με εικόνες βάθους από το πραγματικό περιβάλλον ή με τα πραγματικά αντικείμενα. Στη συνέχεια, με είσοδο δεδομένων, τις εικόνες που παρέχονται από τον 3D αισθητήρα, το σύστημα εξετάζει το περιβάλλον, αφαιρεί το παρασκήνιο και εστιάζοντας στα αντικείμενα του χώρου, αναγνωρίζει το είδος τους και εκτιμά τη θέση τους. Το σύστημα υποστηρίζει πολλαπλά αντικείμενα τοποθετημένα στο χώρο και μπορεί και λειτουργεί σε πραγματικό χρόνο (real-time) ανανεώνοντας συνεχώς τα αποτελέσματα της αναγνώρισης όταν τα αντικείμενα αλλάζουν ή η αισθητήρια κάμερα μετακινείται στο χώρο. Τα αποτελέσματα παρουσιάζονται και οπτικά καθώς το σύστημα τοποθετεί ταμπέλες στην εικόνα του χώρου που παρέχει ο αισθητήρας, ονομάζοντας το αντικείμενο και παρέχοντας στοιχεία για τη θέση του.

1.2 Οργάνωση του τόμου

Η εργασία αυτή είναι οργανωμένη σε έξι κεφάλαια:

Στο κεφάλαιο 1 αναφέρεται επιγραμματικά το αντικείμενο της διπλωματικής εργασίας και γίνεται μια εισαγωγή για τα αναμενόμενα τελικά αποτελέσματα.

Στο Κεφάλαιο 2 δίνεται το θεωρητικό υπόβαθρο των βασικών μεθόδων που μελετήθηκαν και χρησιμοποιήθηκαν στα πλαίσια αυτής της εργασίας, καθώς επίσης αναλύονται οι έννοιες και τεχνολογίες που σχετίζονται με την εργασία και την υλοποίηση του συστήματος.

Στο Κεφάλαιο 3 παρουσιάζεται η ανάλυση και η σχεδίαση του συστήματος, περιγράφονται τα βασικά δομικά μέρη που συνθέτουν το σύστημα αναγνώρισης αντικειμένων, και τα υποσυστήματα που το απαρτίζουν, καθώς και πως αυτά συνεργάζονται και αλληλεπιδρούν.

Στο Κεφάλαιο 4, περιγράφονται λεπτομερώς τα στοιχεία της υλοποίησης του συστήματος, με ανάλυση των βασικών αλγορίθμων καθώς και λεπτομέρειες σχετικά με τις πλατφόρμες και τα προγραμματιστικά εργαλεία που χρησιμοποιήθηκαν.

Στο Κεφάλαιο 5 παρουσιάζεται τα αποτελέσματα από τη λειτουργία του συστήματος με παραδείγματα χρήσης καθώς επίσης παρατίθενται και ποσοτικές συγκρίσεις.

Τέλος, στο Κεφάλαιο 6 γίνεται αναφορά στη συνεισφορά αυτής της διπλωματικής εργασίας, ενώ σχολιάζονται μελλοντικές επεκτάσεις, βελτιώσεις και πιθανές χρήσεις αυτής της τεχνολογίας.

Κεφάλαιο 2

Θεωρητικό υπόβαθρο

2.1 Ψηφιακή Εικόνα

Η εικόνα αποτελεί μία αναπαράσταση της πραγματικότητας σε δύο διαστάσεις με τέτοιο τρόπο που να είναι κατανοητή από τον άνθρωπο. Μια εικόνα μπορεί να οριστεί αλγεβρικά ως μια δισδιάστατη συνάρτηση $f(x,y)$, όπου οι μεταβλητές x και y είναι οι επίπεδες χωρικές συντεταγμένες. Το πλάτος της συνάρτησης f για κάθε ζεύγος συντεταγμένων (x,y) ονομάζεται ένταση και αντιπροσωπεύει για τις ασπρόμαυρες εικόνες το επίπεδο της κλίμακας του γκρι. Όσο μικρότερη η τιμή της έντασης, τόσο πιο κοντά στο λευκό είναι η απόχρωση του συγκεκριμένου χωρικού σημείου, ενώ όσο η ένταση αυξάνεται, η απόχρωση προσεγγίζει το μαύρο.

Αν οι διαθέσιμες τιμές των συντεταγμένων αλλά και της έντασης αποτυπώνονται σε διακριτές βαθμίδες/στάθμες, μία τέτοια εικόνα μπορεί να χαρακτηριστεί ψηφιακή και είναι δυνατόν να αποθηκευτεί ως τα αποτελέσματα της συνάρτησης f για συγκεκριμένες τιμές εισόδου. Έτσι μια πλήρης ψηφιακή εικόνα αποτελείται από ένα πεπερασμένο πλήθος στοιχείων, με συγκεκριμένη θέση και τιμή τα οποία ονομάζονται εικονοστοιχεία (pixels) και συνήθως αποτυπώνουν ορθογώνιο σχήμα (πχ. εικόνα 600x800).

Η ψηφιακή εικόνα αναλύεται με βάση κάποιο ορθογώνιο πλέγμα που λέγεται bitmap. Με βάση αυτό το πλέγμα, η εικόνα μοιράζεται σε μια κάθετη ακολουθία από οριζόντιες σειρές με μικρές υποδιαιρέσεις, που ονομάζονται εικονοστοιχεία. Κάθε εικονοστοιχείο του πλέγματος καθορίζεται από τη θέση του στο πλέγμα (x και y). Συνήθως τα pixels χαρακτηρίζονται ξεκινώντας από την πάνω αριστερή γωνία $(0,0)$, χωρίς αυτό να ισχύει πάντα.

Οι έγχρωμες ψηφιακές εικόνες αποτελούν μία προέκταση των ασπρόμαυρων. Ακολουθούν την ίδια αρχή, δίνοντας όμως στην ένταση του κάθε εικονοστοιχείου αντί για μία τιμή φωτεινότητας, περισσότερες ανάλογα με την απόχρωση των βασικών χρωμάτων που επιλέγονται να αποτυπωθούν. Η πιο συνήθης αναπαράσταση για έγχρωμες εικόνες είναι η κλίμακα RGB, που αποτελείται από τα RED (κόκκινο), GREEN (πράσινο) και



(α') Παράδειγμα ψηφιακής εικόνας

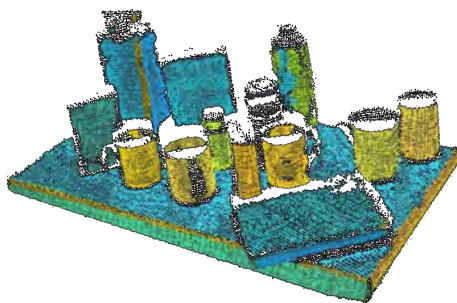
Σχήμα 2.1: Εικονοστοιχεία - pixels

BLUE (μπλε), με την ένταση κάθε εικονοστοιχείου να αποτελεί ένα διάνυσμα τριών στοιχείων. Χρησιμοποιώντας έτσι αυτά τα 3 βασικά χρώματα, όπου από τον κάθε πιθανό συνδυασμό τους μπορούν να προκύψουν και όλα τα ενδιάμεσα αποτελέσματα, μία έγχρωμη ψηφιακή εικόνα μπορεί να αποτυπωθεί με την αναπαράσταση μιας απόχρωσης για κάθε σημείο (x, y) .

Μεταβάλλοντας τον αριθμό των εικονοστοιχείων που αποτελούν μια εικόνα αυξάνεται ή μειώνεται και η ανάλυση της ως προς το μέγεθος, ενώ μεταβάλλοντας το επίπεδο του κβαντισμού ως προς τις τιμές των εντάσεων, αλλάζει η ανάλυση της ως προς το χρώμα.

2.2 Τρισδιάστατη Αναπαράσταση

Η αναπαράσταση εικόνων τριών διαστάσεων, ακολουθεί παρόμοια λογική με αυτή των δύο διαστάσεων και των έγχρωμων εικόνων, με ειδοποιό διαφορά την ένταση που αποτυπώνεται στο χώρο (σε 3 διαστάσεις) και όχι ως προβολή σε μια επίπεδη επιφάνεια.



(α') Παράδειγμα Τρισδιάστατης αναπαράστασης

Σχήμα 2.2: 3D αναπαράσταση

Ο χώρος μπορεί να χωριστεί σε διακριτά σημεία, κάθε ένα από το οποίο έχει τιμή έντασης και μοναδική θέση ως συνάρτηση των μεταβλητών x , y και z . Η ένταση του κάθε σημείου $f(x,y,z)$ μπορεί να περιέχει τιμές για την ύπαρξη ή όχι πληροφορίας στο

σημείο του χώρου δηλαδή να είναι δυαδική ή λογική μεταβλητή, ή και πληροφορία για το χρώμα και τη φωτεινότητα όπως και στις δισδιάστατες εικόνες. Έτσι η ένταση μπορεί να περιέχει περισσότερες της μίας συνιστώσας παίρνοντας τιμές για την ύπαρξη ή μη πληροφορίας στο συγκεκριμένο σημείο του χώρου και παράλληλα, παίρνοντάς τιμές σε κάποιο σύστημα βασικών χρωμάτων.

Σήμερα και ανάλογα με την εφαρμογή, υπάρχει πληθώρα διαφορετικών αναπαραστάσεων τρισδιάστατων μοντέλων και των σημείων αυτών στο χώρο. Για την αναπαράσταση μπορεί να χρησιμοποιηθεί η σχετική γεωμετρία μεταξύ των σημείων, ο χάρτης βάθους, η αναπαράσταση με τρίγωνα ή άλλα πολύγωνα, η αναπαράσταση επιφανειών ή και πιο υψηλού επιπέδου περιγραφή όπως η σκελετική.

2.2.1 Αισθητήρας - κάμερα 3D

Ένας αισθητήρας 3D είναι μία συσκευή που αναλύει και σαρώνει χώρο ή αντικείμενα του πραγματικού κόσμου, προκειμένου να τα ψηφιοποιήσει και να γίνει δυνατή η αναπαράσταση και η αποθήκευση πληροφορίας για αυτά, με ψηφιακά μέσα.



(α') Διαδεδομένοι τρισδιάστατοι αισθητήρες

Σχήμα 2.3: Τρισδιάστατες Κάμερες

Υπάρχουν πολλές τεχνολογίες που προσφέρουν τέτοιου είδους σάρωση, η κάθε μία με τα πλεονεκτήματά και τους περιορισμούς της.

Ξεκινώντας από τις αρχικές μεθόδους επαφής που χρησιμοποιούνταν τα προηγούμενα χρόνια, έχουμε περάσει σε πιο εξελιγμένες τεχνολογίες που χρησιμοποιούν ανάκλαση φωτός ή ακτίνων laser προσφέροντας μεγαλύτερη ανάλυση και ταχύτητα σάρωσης.

Σε γενικές γραμμές όλοι οι τρισδιάστατοι αισθητήρες μετράνε την απόσταση των σημείων μίας σκηνής σε σχέση με ένα προκαθορισμένο σημείο αναφοράς. Έχοντας διαθέσιμες πληθώρα τεχνολογιών και τύπου αισθητήρων, ο κάθε αισθητήρας χρησιμοποιεί και διαφορετικές τεχνικές για την εύρεση των σημείων. Κάποιοι αισθητήρες είναι αργοί και άλλοι αρκετά γρήγοροι, ενώ άλλοι έχουν πολύ υψηλή ανάλυση δεδομένων ή

υψηλό θόρυβο μέτρησης. Ανάλογα κάθε φορά με την εφαρμογή πρέπει να επιλέγεται και ο κατάλληλος αισθητήρας.

Τρισδιάστατοι Αισθητήρες με δύο κάμερες

Ο αισθητήρας αυτός είναι απλά ένας παθητικός δέκτης δισδιάστατων εικόνων. Αποτελείται από δύο κάμερες ίδιων προδιαγραφών, που είναι τοποθετημένες σε μία προκαθορισμένη απόσταση μεταξύ τους, συνήθως στον οριζόντιο άξονα, οι οποίες καταγράφουν η κάθε μία τη σκηνή από μία διαφορετική οπτική, που διαφέρουν λίγο. Υπολογίζοντας την διαφορά μεταξύ των δύο εικόνων είναι δυνατόν να εξάγουμε πληροφορίες για το βάθος της σκηνής στο κάθε σημείο. Όσο μεγαλύτερη η διαφορά μεταξύ δύο σημείων τόσο μικρότερη και η απόσταση από τον αισθητήρα.

Μία τέτοια συσκευή καταγραφής είναι οικονομική αλλά και η ακρίβεια καταγραφής της είναι αρκετά μικρή. Κακές συνθήκες φωτισμού, αντικείμενα τοποθετημένα πολύ κοντά ή μακριά από τον καταγραφέα, καθιστούν αυτή τη μέθοδο απαγορευτική. Επίσης, λόγω του τρόπου με τον οποίο τέτοιοι αλγόριθμοι λειτουργούν, το αποτέλεσμα είναι πολύ φτωχό για σκηνές που έχουν πολύ μικρή πληροφορία ή μεγάλα αντικείμενα με ομοιογενείς επιφάνειες, με λίγα σημεία ενδιαφέροντος.



(α') Αισθητήρας με δύο κάμερες

Σχήμα 2.4: Αισθητήρας με δύο κάμερες

Τρισδιάστατοι Αισθητήρες Time-of-flight

Οι αισθητήρες Time-of-flight (ToF) λειτουργούν μετρώντας το χρόνο που χρειάζεται για μία ακτίνα ή ένα κύμα φωτός να ταξιδέψει μία απόσταση στο χώρο. Καθώς η ταχύτητα του φωτός είναι μία γνωστή σταθερά, μία σχετικά απλή και υπολογιστικά μικρή εξίσωση μπορεί να χρησιμοποιηθεί για να υπολογιστεί η απόσταση σημείων σε

μία σκηνή. Οι αισθητήρες αυτοί δεν επηρεάζονται από τις συνθήκες φωτός που επικρατούν στο περιβάλλον της μέτρησης και έχουν τη δυνατότητα για μετρήσεις μεγάλης ακρίβειας.

LIDAR

Ένας αισθητήρας LIDAR (light+radar) αποτελεί έναν απλό αισθητήρα laser τοποθετημένο σε πλατφόρμα, που έχει τη δυνατότητα να περιστρέφεται πολύ γρήγορα, σκανάροντας τη σκηνή σημείο σημείο. Τέτοιοι αισθητήρες έχουν πολύ μεγάλη ακρίβεια αλλά και πολύ μεγάλο κόστος κατασκευής και δεν μπορούν να εξάγουν πληροφορίες για την υφή των επιφανειών. Μπορούν να τοποθετηθούν σε πληθώρα κτισμάτων και οχημάτων, όπως δορυφόροι, αεροπλάνα, ρομποτικά οχήματα. Λόγω της υψηλής ανάλυσης των μετρήσεων απαιτείται και πολύ μεγάλη υπολογιστική ισχύ για την επεξεργασία τους.



(α') Αισθητήρας LIDAR

Σχήμα 2.5: Αισθητήρας LIDAR

ToF Cameras

Ένας αισθητήρας τέτοιου τύπου δεν εκτελεί σκανάρισμα σημείο προς σημείο όπως ο αισθητήρας LIDAR αλλά χρησιμοποιεί ένα κύμα φωτός για να σαρώσει μία σκηνή μία φορά ανά καρέ. Εξαιτίας αυτής της λειτουργίας μπορούν να λειτουργήσουν πολύ γρήγορα, ακόμα και με δειγματοληψίες μεγαλύτερες των 100 Hz. Η ακρίβεια μέτρηση βάθους με αυτή τη μέθοδο φτάνει τα 0,5cm.

Ο νέος αισθητήρας kinect v2 είναι ένας χαμηλής ανάλυσης ToF αισθητήρας που λειτουργεί σε συνεργασία με μίας υψηλής ανάλυσης RGB κάμερα.



(α) Αισθητήρας
Time-of-flight

(β) Αισθητήρας Kinect v2

Σχήμα 2.6: Αισθητήρες Time-of-flight

Τρισδιάστατοι Αισθητήρες Υπέρυθρου φωτός

Οι αισθητήρες-κάμερες υπερέυθρων προκειμένου να δώσουν μια τρισδιάστατη απόδοση του χώρου, αποτελούνται από τα εξής επιμέρους συστατικά στοιχεία:

Προβολέας υπερέυθρων ο οποίος προβάλλει στο χώρο σημεία, υπερέυθρου φωτός, που ακολουθούν γνωστές αναπαραστάσεις (patterns).

Κάμερα υπερέυθρων η οποία εντοπίζει την αντανάκλαση από αυτά τα προβαλλόμενα σημεία.

Με αυτό τον τρόπο καθίσταται δυνατός ο προσδιορισμός της απόστασης κάθε τέτοιου σημειακού φωτός, που ανακλάται στο χώρο, από την αισθητήρια κάμερα. Με απλό μετασχηματισμό και γνωρίζοντας κάποια σταθερά σημεία του χώρου μπορούμε να τον αναπαραστήσουμε ψηφιακά. Αυτός ο τρόπος αναπαράστασης λέγεται χάρτης βάθους και αναλύεται θεωρητικά στη συνέχεια.

Τέτοιες συσκευές διαθέτουν επιπλέον συμβατική ψηφιακή έγχρωμη κάμερα, προκειμένου με το συνδυασμό των δύο εισόδων να μπορούμε να παράγουμε έγχρωμη και σε τρεις διαστάσεις τις αναπαραστάσεις του χώρου και των αντικειμένων.



(α') RGB εικόνα

(β') Χάρτης βάθους

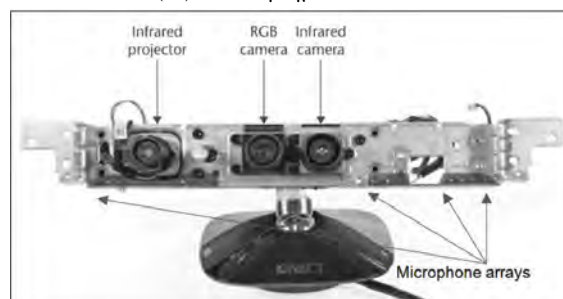
Σχήμα 2.7: Χάρτης Βάθους

Ο αισθητήρας Kinect

Ένας από τους πιο διαδεδομένους αισθητήρες τρισδιάστατης σάρωσης είναι ο αισθητήρας Kinect. Ο αισθητήρας αυτός εισήχθη στην καταναλωτική αγορά, αρχικά ως επιπρόσθετο εξάρτημα ηλεκτρονικής παιχνιδιομηχανής, αλλά λόγω της υψηλής του ακρίβειας σε συνδυασμό με το χαμηλό κόστος απόκτησης του, έγινε η βάση για καινοτόμες εφαρμογές και σε άλλους τομείς έκτος της ψυχαγωγίας, όπως στη βιομηχανία, στην αποθήκευση και ακόμα και σε ιατρικές εφαρμογές.



(α') αισθητήρας Kinect



(β') αισθητήρας Kinect (Hardware)

Σχήμα 2.8: αισθητήρας Kinect

Η συσκευή Kinect απαρτίζεται από πληθώρα αισθητήρων και συγκεκριμένα τρισδιάστατο αισθητήρα υπέρυθρου φωτός, έγχρωμη κάμερα, μικρόφωνα και κινητήρα που ελέγχει την οριζόντια γωνία θέασης των καμερών.

2.2.2 Λήψη δεδομένων βάθους

Όπως προαναφέρθηκε, σε μία απεικόνιση βάθους, η τιμή της έντασης του κάθε εικονοστοιχείου εκφράζει την απόσταση του σημείου του αντικειμένου που αντιπροσωπεύεται από το συγκεκριμένο εικονοστοιχείο σε σχέση με ένα σημείο αναφοράς το οποίο συνήθως ταυτίζεται με το σύστημα λήψης της απεικόνισης (την κάμερα).

Ένας αισθητήρας τρισδιάστατης σάρωσης χρησιμοποιεί τον υπέρυθρο πομπό ακτίνων laser, μια υπέρυθρη κάμερα και μία έγχρωμη κάμερα για να συλλέξει την απαιτούμενη πληροφορία. Ο πομπός εκπέμπει μία δέσμη laser η οποία ανακλάται σε ένα κάτοπτρο με αποτέλεσμα να διαχωρίζεται σε πολλές δέσμες, κάτι που προκαλεί τη δημιουργία ενός σταθερού μοτίβου από κουκκίδες οι οποίες προβάλλονται στον χώρο. Οι δέσμες αυτές, ανακλώνται στα αντικείμενα του χώρου και η υπέρυθρη κάμερα μπορεί και ανακτά ψηφιακά τη θέση τους ως μια δισδιάστατη απεικόνιση.

Το μοτίβο από κουκκίδες που ανακτάται από τον αισθητήρα, συσχετίζεται με ένα μοτίβο αναφοράς που έχει συλλεχθεί κατά την βαθμονόμηση του εξοπλισμού του αισθητήρα. Ένα σημείο μπορεί να συγκριθεί με το μοτίβο αναφοράς, και η θέση του ανάλογα με το ποσό έχει μετακινηθεί στον οριζόντιο ή κατακόρυφο άξονα μπορεί να δείξει και την απόσταση του αντικειμένου πάνω στο οποίο εμφανίζεται.

Μαθηματικό Μοντέλο Εύρεσης Βάθους

Η σχέση μεταξύ της απόστασης ενός αντικειμένου k από τον αισθητήρα και της επιφάνειας με τις κουκκίδες αναφοράς παρουσιάζονται γραφικά παρακάτω. Η απόσταση (d) του επιπέδου στο οποίο βρίσκεται το αντικείμενο σε σχέση με το επίπεδο αναφοράς, είναι η διαφορά η οποία θέλουμε να υπολογίσουμε προκειμένου να μπορεί να καθοριστεί η απόσταση του σημείου.

Στον τρισδιάστατο χώρο και με αρχή των αξόνων το κέντρο της κάμερας το οποίο θεωρούμε ότι συμπίπτει και με το προβολικό κέντρο του συστήματος, ο άξονας Z καθορίζεται ως ο άξονας που είναι κάθετος στο επίπεδο της εικόνας και εκτείνεται προς το αντικείμενο και ο άξονας X , ο άξονας που είναι παράλληλος με την νοητή γραμμή η οποία ενώνει την κάμερα με τον πομπό υπέρυθρων.

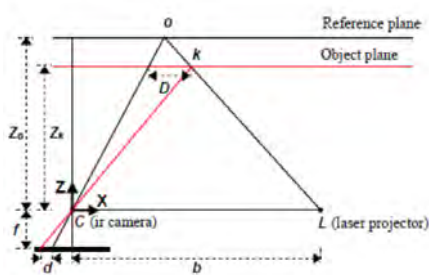
Υποθέτουμε ότι η κουκκίδα βρίσκεται στο επίπεδο αναφοράς με απόσταση από τον αισθητήρα και η απεικόνισή της απεικονίζεται στο επίπεδο της εικόνας. Αν η κουκκίδα μετακινηθεί ως προς τον αισθητήρα, τότε η απεικόνισή της στο επίπεδο της εικόνας θα παρουσιάσει μια μετακίνηση στον άξονα X . Αυτή την μετακίνηση υπολογίζεται με την

βοήθεια της μεθόδου ομοίων τριγώνων:

$$\frac{D}{d} = \frac{Z_0 - Z_k}{Z_0}$$

$$\frac{d}{f} = \frac{D}{Z_k}$$

Z_k : η νέα απόσταση της κουκκίδας και κατά συνέπεια το επίπεδο του αντικειμένου,
 b : η απόσταση μεταξύ της κάμερας από τον πομπό υπερύθρων, f : η εστιακή απόσταση της κάμερας, D : η μετακίνηση της κουκκίδας στο επίπεδο του αντικειμένου, d : η μετακίνησης της κουκκίδας.



(α') γεωμετρική αναπαράσταση

Σχήμα 2.9: Υπολογισμός βάθους

Λύνοντας ως προς Z_k έχουμε μαθηματικά τον υπολογισμό του βάθους:

$$Z_k = \frac{Z_0}{1 + \frac{Z_0 d}{f b}}$$

Τέλος, οι συντεταγμένες του κάθε σημείου του αντικειμένου μπορούν να υπολογιστούν:

$$X_k = \frac{Z_k}{f}(x_k - x_0 + \delta_x)$$

και

$$Y_k = \frac{Z_k}{f}(y_k - y_0 + \delta_y)$$

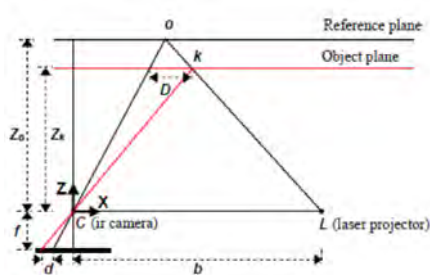
Σφάλμα μέτρησης

Η διαδικασία σάρωσης παρουσιάζει πολλές συνιστώσες οι οποίες μπορούν να προκαλέσουν την εμφάνιση σφαλμάτων στη μέτρηση.

Αρχικά, ο ίδιος ο αισθητήρας μπορεί να εισαγάγει λάθος πληροφορίες στην ψηφιακή αποτύπωση που οφείλονται σε κατασκευαστικούς περιορισμούς του, όπως η μέγιστη

ανάλυση που μπορεί να υποστηρίξει ο αισθητήρας αλλά και σε λάθος ή έλλειψη βαθμονόμηση των ηλεκτρονικών αλλά και μηχανικών συστατικών του στοιχείων. Σύμφωνα με τις προδιαγραφές του, ο κάθε αισθητήρας είναι σχεδιασμένος και ικανός να κάνει μετρήσεις σε συγκεκριμένη απόσταση από τα αντικείμενά της μετρήσεις που είναι συνήθως 0.5 έως 5 μ, με τα σφάλματα να αυξάνονται όσο αυξάνεται η απόσταση του αντικειμένου.

Συνήθης πηγή σφαλμάτων αποτελεί και το περιβάλλον κάτω από το οποίο γίνονται μετρήσεις με τον αισθητήρα. Ο φωτισμός, οι ανακλάσεις στα αντικείμενα του περιβάλλοντος και η γεωμετρία της διάταξης μπορούν να εμφανίσουν λάθη στη γεωμετρία και το νέφος που σαρώνει ο αισθητήρας ή να αποκρύψουν σημεία της γεωμετρίας των αντικειμένων.



(α') γεωμετρική αναπαράσταση

Σχήμα 2.10: Σφάλμα μέτρησης

2.2.3 Χάρτης Βάθους

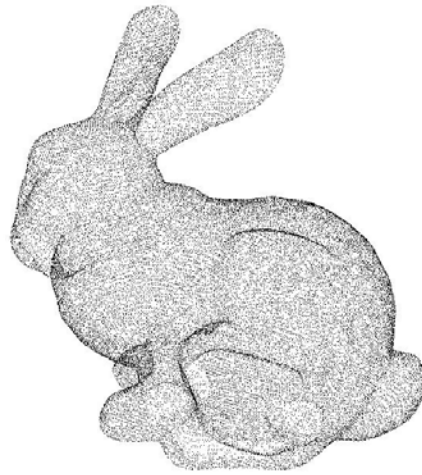
Ο χάρτης βάθους αποτελεί έναν απλό τρόπο αποθήκευσης και οπτικής αναπαράστασης τρισδιάστατων αντικειμένων.

Η αναπαράσταση αυτή αποτελεί ψηφιακή εικόνα, διαφοροποιούμενη από τη συμβατική στο γεγονός πως η ένταση του κάθε εικονοστοιχείου δεν αποτελεί δείκτη φωτεινότητας αλλά της απόστασης που ανιχνεύθηκε από τον αισθητήρα.

Αλλάζοντας το χώρο της απόστασης και προβάλλοντας τον στη κλίμακα του γκρι μπορούμε οπτικά να αντιληφθούμε την απόσταση, πιο φωτεινό σημείο θα βρίσκεται κοντά στην κάμερα και πιο σκοτεινό μακριά.

2.2.4 Point Clouds

Τα νέφη σημείων αποτελούν αναπαράσταση δεδομένων σε κάποιο σύστημα συντεταγμένων. Για αναπαράσταση αντικειμένων στις τρεις διαστάσεις ένα Point Cloud περιέχει πληροφορίες για τη θέση και τη φωτεινότητα, έγχρωμη ή άχρωμη.



(α') σάρωση point-cloud για μοντέλο λαγού

Σχήμα 2.11: Point clouds

Ένας Χάρτης βάρους μπορεί να μετατραπεί σε νέφος σημείων γνωρίζοντας όμως γεωμετρικά χαρακτηριστικά των φακών του αισθητήρα που δημιούργησε το χάρτη καθώς και την κλίμακα στην οποία δίνεται το βάθος. Για την εξαγωγή αυτών των μοναδικών χαρακτηριστικών της κάμερας που χρησιμοποιείται, απαιτείται βαθμονόμηση του αισθητήρα, μία διαδικασία που θα μειώσει την εμφάνιση σφαλμάτων μέτρησης.

Δουλεύοντας σε χαμηλό επίπεδο με point-clouds προκύπτουν ενδιαφέρουσες και ποικίλες δυνατότητες. Καθώς τα σύννεφα αυτά αποτελούν μία "σάρωση", μπορούν να αποτυπώσουν ολόκληρο το χώρο εργασίας ή δωμάτιο και αυτή η αποτύπωση να χρησιμοποιηθεί για διαδικασίες τμηματοποίησης ή αναγνώρισης.

Μπορούν να χρησιμοποιηθούν για τη μέτρηση αποστάσεων, αναγνώριση της γεωμετρίας αντικειμένων ή εμποδίων, ως βοήθημα για ρομποτικές πλατφόρμες. Η πιο εξελιγμένη αλλά και χρήσιμη χρήση τους, αποτελεί η 3D αναγνώριση και προσδιορισμός της θέσης αντικειμένων, ακόμα και σε πραγματικό χρόνο.

2.3 Γεωμετρικά χαρακτηριστικά - Features

Υπάρχουν πολλά 3D χαρακτηριστικά που μπορούν να εφαρμοστούν σε ένα νέφος σημείων. Κάθε ένα έχει τη δική του μέθοδο για τον υπολογισμό μοναδικών τιμών για αυτό. Ορισμένα χαρακτηριστικά χρησιμοποιούν τη διαφορά μεταξύ των γωνιών των normal του σημείου και των γειτόνων του. Άλλα χρησιμοποιούν τις αποστάσεις μεταξύ των σημείων. Εξαιτίας αυτού, μερικά χαρακτηριστικά είναι εγγενώς καλύτερα ή χειρότερα για συγκεκριμένους σκοπούς. Ένας δεδομένος περιγραφέας μπορεί να είναι ανεξάρτητος από τη κλίμακα, και ένας άλλος μπορεί να είναι καλύτερος για

παραμορφωμένη ή και μερική θέαση των αντικειμένων.

Η εξαγωγή των χαρακτηριστικών γίνεται εισάγοντας στο σύστημα τα σημεία και τις μεταβλητές που τα περιγράφουν. Ο κάθε περιγραφέας κάνει τους ανάλογους υπολογισμούς για να προκύψει η τιμή ή οι τιμές που περιγράφουν το προς χαρακτηρισμό αντικείμενο ή εικόνα.

Μετά τον υπολογισμό των απαραίτητων τιμών, εκτελείται ένα πρόσθετο βήμα για να μειώσει το μέγεθος του περιγραφέα και το αποτέλεσμα ομαδοποιείται και αποθηκεύεται σε ένα ιστόγραμμα. Για να γίνει αυτό, το εύρος της κάθε μεταβλητής που συνθέτει τον περιγραφέα χωρίζεται σε N υποδιαίρεσεις, όπου μετράται ο αριθμός των παρατηρήσεων σε κάθε μία.

Ας φανταστούμε έναν περιγραφέα που υπολογίζεται για μια μόνο μεταβλητή, η οποία κυμαίνεται από 1 έως 100 και επιλέξουμε να δημιουργήσουμε 10 υποδιαίρεσεις. Έτσι στη πρώτη μπορούμε να συγκεντρώσουμε όλα τα περιστατικά μεταξύ 1 και 10, στη δεύτερη 11-20, και ούτω καθεξής. Εξετάζουμε τώρα, την τιμή της μεταβλητής για το πρώτο γειτονικό ζεύγος, υποθέτοντας τιμή ίση με 27. Έτσι αυξάνουμε την αξία της τρίτης υποδιαίρεσης-ομάδας κατά 1. Συνεχίζοντας αυτή τη διαδικασία για όλες τις τιμές προκύπτει και το τελικό ιστόγραμμα. Το μέγεθος και το πλήθος των ομάδων πρέπει να επιλέγεται με προσοχή ανάλογα πάντα με την εφαρμογή.

Οι περιγραφείς μπορούν να χωριστούν σε δύο κατηγορίες: Τοπικοί και Γενικοί. Η εξαγωγή και η διαδικασία αναγνώρισης που ακολουθείται ανάλογα με τη διαδικασία είναι διαφορετική και εξηγείται στη συνέχεια.

2.3.1 Local Features

Οι τοπικές περιγραφείς υπολογίζονται για τα επιμέρους σημεία που δίνουμε ως είσοδο στο σύστημα. Ο περιγραφέας εδώ δεν 'γνωρίζει' πως περιγράφει ένα αντικείμενο, το μόνο που περιγράφει είναι πώς εκτείνεται η τοπική γεωμετρία γύρω από το σημείο αυτό.

Ανάλογα με την εφαρμογή μπορούμε να επιλέξουμε για ποια σημεία του αντικειμένου θα γίνει ο υπολογισμός. Επίσης γίνεται μια μείωση δειγματοληψίας, μειώνοντας το πλήθος των συνολικών σημείων του αντικειμένου πριν τον υπολογισμό.

2.3.2 Global Features

Global περιγραφείς κωδικοποιούν και περιγράφουν το αντικείμενο όσον αφορά τη γεωμετρία του. Δεν υπολογίζονται για μεμονωμένα σημεία, αλλά για ένα ολόκληρο σύμπλεγμα που αντιπροσωπεύει ένα αντικείμενο. Εξαιτίας αυτού, ένα στάδιο προεπεξεργασίας (τμηματοποίηση) είναι πάντα απαραίτητη, προκειμένου να ανακτηθούν πιθανά υποψηφία αντικείμενα μέσα σε μία σάρωση που περιέχει περισσότερα του ενός

αντικείμενα, θόρυβο αλλά και τον περιβάλλοντα χώρο των αντικειμένων.

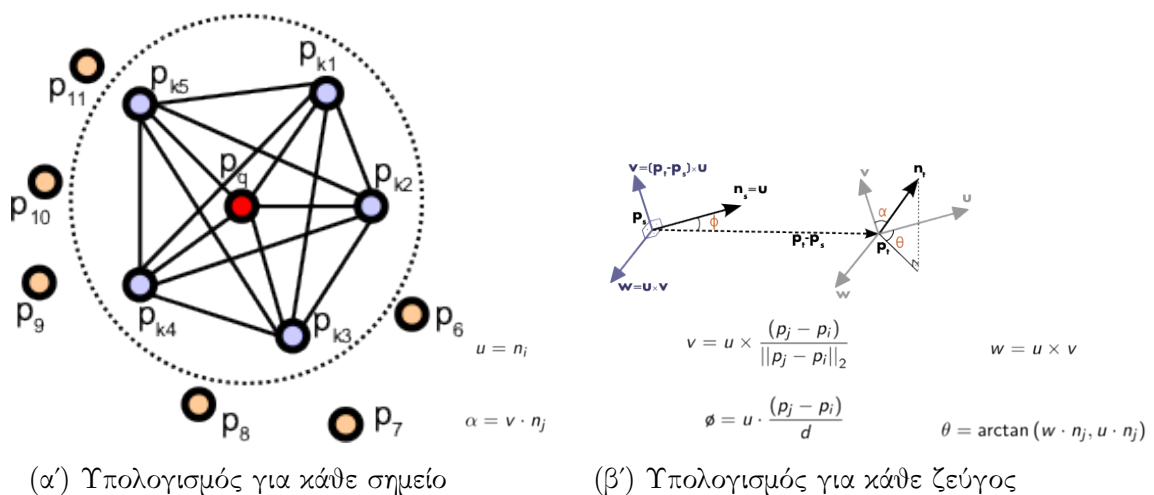
Πολλοί τοπικοί περιγραφείς μπορούν επίσης να χρησιμοποιηθούν και ως Global. Αυτό μπορεί να γίνει με περιγραφείς που χρησιμοποιούν ακτίνα κύκλου για να ψάξουν γειτονικά σημεία. Υπολογίζοντας το χαρακτηριστικό για ένα σημείο στο νέφος και θέτοντας την ακτίνα στη μέγιστη πιθανή απόσταση μεταξύ δύο σημείων, όλα τα σημεία μέσα στο νέφος θεωρούνται γειτονικά.

Στα ακόλουθα υπο-κεφάλαια θα αναφερθούν συγκεκριμένοι περιγραφείς βάση των οποίων καταλήγουμε σε αυτούς που χρησιμοποιήθηκαν στην υλοποίηση της Εργασίας.

2.3.3 PFH

Ο τοπικός περιγραφέας PFH(Point Feature Histogram) όπως δηλώνει το όνομα του, παράγει ένα ιστόγραμμα, προσπαθώντας να περιγράψει το αντικείμενο από τα σημεία που το αποτελούν. Είναι από τους πιο σημαντικούς περιγραφείς στη 3D αναγνώριση καθώς αυτός αποτελεί τη βάση για άλλους πιο περίπλοκους και μοντέρνους περιγραφείς. Για να περιγράψει τη γεωμετρία του αντικειμένου αναλύει τη διαφορά μεταξύ των κατευθύνσεων των μέσων που υπολογίστηκαν για το αντικείμενο. Για το λόγο αυτό κάποιο μέσοι που προέκυψαν από σφάλμα μπορεί να προκαλέσουν την χαμηλή ποιότητα στη περιγραφή.

Αρχικά, ο αλγόριθμος ταιριάζει τα σημεία μεταξύ τους και με τα γειτονικά τους. Έπειτα, για κάθε ζευγάρι σημείων υπολογίζεται μία τιμή με βάση των μέσων. Η τιμή αυτή αποτελείται από τρεις πολικές μεταβλητές οι οποίες μαζί με την ευκλείδεια απόσταση μεταξύ των σημείων αποθηκεύονται και προστίθενται στο τελικό ιστόγραμμα. Η Διαδικασία επαναλαμβάνεται και προκύπτει ο τελικός περιγραφέας με τέσσερις συνολικά μεταβλητές.

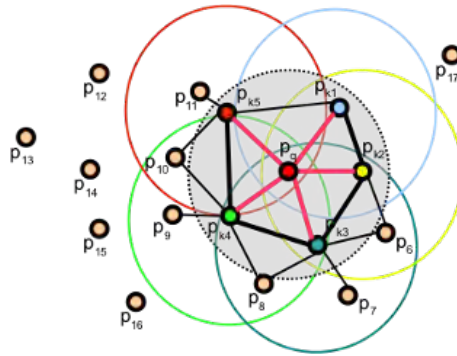


Σχήμα 2.12: Περιγραφέας PFH

2.3.4 FPFH

Ο περιγραφέας PFH δίνει τις περισσότερες φορές ακριβή αποτελέσματα, παρ' όλα αυτά έχει ένα μειονέκτημα. Είναι αρκετά χρονοβόρος σε ένα πραγματικό σύστημα, καθώς έχει μεγάλη υπολογιστική πολυπλοκότητα. Έτσι, η χρήση του σε συστήματα πραγματικού χρόνου είναι απαγορευτική. Για ένα νέφος N σημείων με K γείτονες η πολυπλοκότητα του υπολογίζεται ως $O(NK^2)$. Λόγω αυτής της αδυναμίας δημιουργήθηκε ο περιγραφέας FPFH (Fast Point Feature Histogram).

Ο νέος λοιπόν αυτός, επίσης τοπικός περιγραφέας λαμβάνει υπ' όψιν του μόνο την άμεση σύνδεση του σημείου με τα γειτονικά του, αφαιρώντας επιπλέον συνδέσεις και επιταχύνοντας έτσι τη διαδικασία υπολογισμού. Έτσι, η πολυπλοκότητα μειώνεται σε $O(nK)$. Το ιστόγραμμα που δημιουργείται είναι απλούστερο, με την έννοια ότι περιέχει λιγότερη πληροφορία αλλά αποτελεί αποδεκτή παραδοχή του αρχικού.



(α') Υπολογισμός για κάθε ζεύγος

Σχήμα 2.13: Περιγραφέας FPFH

2.3.5 VFH

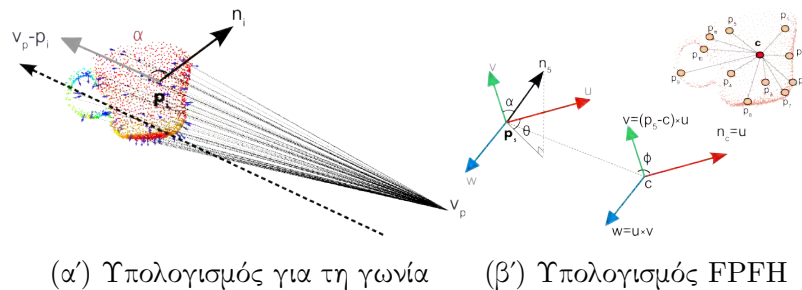
Ο περιγραφέας VFH Viewpoint Feature Histogram βασίζεται στον FPFH αλλά είναι Global περιγραφέας. Βελτιώνοντας και πάλι τις δυνατότητες του προηγούμενου περιγραφέα, εδώ προστέθηκε η έννοια της οπτικής γωνίας. Ο περιγραφέας FPFH προκειμένου να χρησιμοποιηθεί για αναγνώριση, η οπτική γωνία έπρεπε να παραμένει σταθερή. Ο νέος αυτός περιγραφέας υπολογίζει τον FPFH για όλο το σύννεφο των σημείων.

Ο περιγραφέας αποτελείται από δύο επιμέρους στοιχεία. Ένα στοιχείο που προσδιορίζει τη γωνία και ένα που υπολογίζει τον FPFH.

Για τη γωνία υπολογίζεται το κέντρο του αντικειμένου, ως η μέση τιμή των X , Y , Z μεταβλητών. Έτσι στη συνέχεια υπολογίζεται η γωνία που δημιουργείται ανάμεσα στον αισθητήρα και το σημείο αυτό και τελικά κανονικοποιείται. Στη συνέχεια υπολογίζεται η

γωνία ανάμεσα στο κάθε σημείο και της μέσης τιμής του και το αποτέλεσμα προστίθεται στο τελικό ιστόγραμμα.

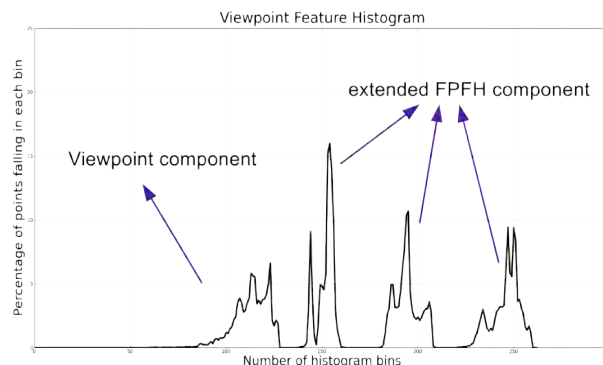
Το δεύτερο στοιχείο αποτελείται από τον υπολογισμό του FPFH, δηλαδή προκύπτουν 3 ιστογράμματα και 3 πολικά χαρακτηριστικά.



(α') Υπολογισμός για τη γωνία (β') Υπολογισμός FPFH

Σχήμα 2.14: Περιγραφέας VFH

Ο τελικός περιγραφέας αποτελείται από 4 ιστογράμματα τα οποία συνδυάζονται για τον τελικό περιγραφέα, όμως μπορεί να συνυπολογιστεί και ένα πέμπτο ιστόγραμμα με τις αποστάσεις των σημείων από το κέντρο του αντικειμένου.



(α') Ιστόγραμμα VFH

Σχήμα 2.15: Περιγραφέας VFH

2.3.6 CVFH

Ο αρχικός VFH περιγραφές δεν είναι εύρωστος όσον αφορά τη παραμόρφωση, αν το αντικείμενο βρίσκεται πίσω από άλλο έχοντας μερική θέαση και άλλους παράγοντες ή σφάλματα μέτρησης που μπορεί να εισαγάγει η κάμερα. Αν από την αναπαράσταση του αντικειμένου σε νέφος σημείων πολλά από αυτά απουσιάζουν ή έχουν παραμορφωθεί και αλλάξει σχετική θέση το υπολογισθέν κεντρικό σημείο του αντικειμένου θα διαφέρει κατά πολύ από το αναμενόμενο, αλλάζοντας τον τελικό περιγραφέα που θα εξαχθεί και

αποτρέποντας της εύρεση μιας θετικής ταύτισης. Για να εξαλειφθεί αυτός ο παράγοντας δημιουργήθηκε ο περιγραφέας Clustered Viewpoint Feature Histogram (CVFH).

Η ιδέα που εισαγάγει ο καινούριος αυτός περιγραφέας έχει ως εξής: αντί να υπολογιστεί ένας μοναδικός VFH περιγραφέας για όλο το νέφος που αναπαριστά ένα αντικείμενο, το νέφος υφίσταται τμηματοποίηση και χωρίζεται σε περιοχές, κάτι που εξαλείφει πολλά σφάλματα κατά τον υπολογισμό του κέντρου του αντικειμένου. Στη συνέχεια υπολογίζεται ένας περιγραφέας VFH για κάθε τμήμα του αρχικού νέφους. Έτσι, ένα αντικείμενο μπορεί να βρεθεί ανάμεσα στο τοπίο ακόμα και αν ένα μόνο κομμάτι του είναι καθαρά εμφανές.



(α') Παράδειγμα: σημεία του(β') Τμηματοποίηση αντικει-
νέφους απουσιάζουν μένου

Σχήμα 2.16: Περιγραφέας CVFH

Επιπλέον υπολογίζεται και περιλαμβάνεται στον τελικό περιγραφέα ένα αντικείμενο κατανομής σχήματος Shape Distribution Component (SDC) . Μετρώντας τις αποστάσεις των σημείων ως προς το κέντρο του κάθε τμήματος το SDC βοηθάει ώστε να διαφοροποιηθούν αντικείμενα με κοινά χαρακτηριστικά όπως το σχήμα , παραδείγματος χάριν, δύο επίπεδες επιφάνειες.

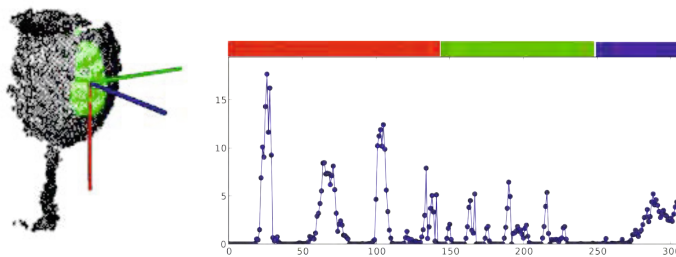
Παρόλο που στον περιγραφέα VFH γίνεται κανονικοποίηση του τελικού ιστογράμματος η διαδικασία αυτή απουσιάζει ώστε να είναι ο περιγραφέας ανεξάρτητος κλίμακας και να μπορούν δύο αντικείμενα με κοινά χαρακτηριστικά αλλά διαφορετικό μέγεθος να διακριθούν.

Ο περιγραφέας CVFH είναι ανεξάρτητος της περιστροφής όπως και οι περισσότεροι global περιγραφείς. Αυτό συμβαίνει καθώς περιστροφές γύρω από τον άξονα της κάμερας δεν αλλάζουν της παρατηρούμενη γεωμετρία σύμφωνα με την οποία γίνεται ο υπολογισμός του περιγραφέα, δυσχεραίνοντας έτσι όμως τον εντοπισμό της πλήρης στάσης του αντικειμένου και δίνοντας πληροφορία για πέντε βαθμούς ελευθερίας. Ο έκτος και τελευταίος βαθμός ελευθερίας που απαιτείται για τον πλήρη εντοπισμό της στάσης μπορεί να αποκτηθεί με την επιπλέον χρήση περιγραφέα της περιστροφής της κάμερας Camera Roll Histogram (CRH).

2.3.7 OUR-CVFH

Τέλος, εξέλιξη των προηγούμενων περιγραφών αποτελεί ο Oriented, Unique and Repeatable CVFH . Ο περιγραφέας αυτός προσθέτει τον υπολογισμό ενός μοναδικού αρχικού καρέ που χρησιμοποιείται ως αναφορά για τους υπόλοιπους υπολογισμούς κάνοντας έτσι τον περιγραφέα πιο εύρωστο.

Ο περιγραφέας βασίζεται στη χρήση ενός semi-global καρέ το οποίο αποτελείται από επαναλαμβανόμενα συστήματα αναφοράς υπολογισμένα ξεχωριστά για την κάθε περιοχή του νέφους. Έτσι αφαιρείται η παράμετρος της περιστροφής της κάμερας κάνοντας τον περιγραφέα ικανό να δώσει μια εκτίμηση για τη στάση του αντικειμένου σε έξι βαθμούς ελευθερίας χωρίς την προσθήκη παραπάνω βημάτων.



(α') Υπολογισμός καρέ

Σχήμα 2.17: Περιγραφέας OUR-CVFH

Το πρώτο κομμάτι του υπολογισμού είναι το ίδιο με αυτό του CVFH αλλά αφού επέλθει η τμηματοποίηση τα σημεία κάθε τμήματος φιλτράρονται ακόμη μία φορά, σύμφωνα με τη διαφορά ανάμεσα στα normals (διανύσματα κατεύθυνσης) και τον μέσο όρο των normal. Έτσι έχουμε ως αποτέλεσμα, καλύτερα χωρισμένα αντικείμενα βοηθώντας έτσι στην εγκυρότητα της τελικής αναγνώρισης.

Ως συνέχεια αυτού του βήματος, το μοναδικό καρέ υπολογίζεται για το κάθε τμήμα. Έτσι γίνεται μια διαδικασία αποσαφήνισης για να διαπιστωθεί σύμφωνα με την κατανομή των σημείων η φορά των αξόνων. Έτσι προκύπτει ο τελικός περιγραφέας, ο οποίος και χρησιμοποιείται στα πλαίσια αυτής της εργασίας για την μοντελοποίηση και υλοποίηση συστήματος αυτόματης αναγνώρισης αντικειμένων.

2.4 Εκπαίδευση - Μάθηση Συστήματος

Η Μάθηση ή Μηχανική μάθηση είναι ένας υποτομέας της επιστήμης των υπολογιστών (CS) και τεχνητής νοημοσύνης (AI) που ασχολείται με την κατασκευή και τη μελέτη των συστημάτων που μπορούν να μάθουν από τα δεδομένα, αντί να ακολουθούν μόνο τις ρητά προγραμματισμένες οδηγίες όπως ένα πρόγραμμα υπολογιστή. Έχει ισχυρούς δεσμούς με τη στατιστική και τις τεχνικές βελτιστοποίησης, και χρησιμοποιείται

σε μια σειρά υπολογιστικών εργασιών, όπως για το σχεδιασμό και τον προγραμματισμό εφαρμογών όπως το φιλτράρισμα ανεπιθύμητων μηνυμάτων σε ηλεκτρονικά μηνύματα, οπτικής αναγνώρισης χαρακτήρων, μηχανική μάθηση, εξόρυξη δεδομένων, και την αναγνώριση προτύπων.

Οι διαδικασίες μηχανικής μάθησης μπορεί να πάρουν διάφορες μορφές. Στην επιβλεπόμενη μάθηση, ο υπολογιστής παρουσιάζεται με παραδείγματα εισόδων και τα επιθυμητά αποτελέσματα τους, τα οποία δίνονται από έναν "δάσκαλο", και ο στόχος του συστήματος είναι να μάθει ένα γενικό κανόνα που να απεικονίζει εισόδους σε εξόδους. Μετά το πέρας της μάθησης αν ο "κανόνας" αυτός είναι φτιαγμένος από ένα καλό δείγμα των πληροφοριών που τελικά θα δοθούν ως είσοδος, το σύστημα θα καταφέρει να απεικονίσει την έξοδο για κάθε είσοδο χωρίς αυτή να είναι ίδια απαραίτητα με κάποιο από τα δεδομένα εισόδου που δόθηκαν κατά τη μάθηση.

Στην εκμάθηση χωρίς επίβλεψη, δεν υπάρχουν ετικέτες που να έχουν δοθεί στον αλγόριθμο μάθησης σχετικά με τα δεδομένα εισόδου, δίνοντας στο σύστημα ως στόχο να κατηγοριοποιήσει μόνο του τα δεδομένα σε ομάδες παρόμοιων εισροών (clustering). Η επιβλεπόμενη μάθηση μπορεί να είναι αυτοσκοπός (ανακαλύπτοντας κρυμμένα μοτίβα στα δεδομένα) ή ένα μέσο που θα βοηθήσει σε μία αναγνώριση.

Στην ενισχυτική μάθηση, ένα πρόγραμμα υπολογιστή αλληλεπιδρά με ένα δυναμικό περιβάλλον στο οποίο θα πρέπει να εκτελέσετε ένα συγκεκριμένο στόχο (όπως η οδήγηση ενός οχήματος), χωρίς ο χρήστης ρητά να δηλώνει αν έχει έρθει κοντά στο στόχο της ή όχι. Κάθε σύστημα αναγνώρισης για να επιτελέσει τη λειτουργία του απαιτείται να έχει αποκτήσει πληροφορίες για το είδος και τα χαρακτηριστικά των αντικειμένων που θα λάβει ως είσοδο. Αυτή η πληροφορία υπόκειται επεξεργασία, προσθέτοντας ή αφαιρώντας στοιχεία που θα κάνουν την αναγνώριση ευκολότερη, ταχύτερη και πιο αξιόπιστη.

Οι τεχνικές αυτές δίνουν έτσι στο σύστημα μια τεχνητή νοημοσύνη και του επιτρέπουν να προσαρμόζεται διαφορετικά κάθε φορά ανάλογα με τα στοιχεία βάση στα οποία εκπαιδεύτηκε. Για την εκπαίδευση πρέπει ανάλογα με την περίπτωση να επιλεγούν τέτοια χαρακτηριστικά που να μπορούν να αναδείξουν όσο το δυνατόν ευκολότερα τις διαφορές στις εισόδους.

Η καλή και επιτυχής μάθηση ενός συστήματος αποτελεί λεπτή έννοια καθώς το σύστημα μπορεί να είναι τέλεια εκπαιδευμένο αλλά τη στιγμή της αναγνώρισης να είναι εξαιρετικά επιρρεπές σε μικρές μεταβολές και σφάλματα. Αντίθετα ένα σύστημα μπορεί να μην είναι σε μεγάλη λεπτομέρεια εκπαιδευμένο με αποτέλεσμα ένα είδος η μία κατηγορία να καταλήγει πάντα να είναι το αποτέλεσμα της αναγνώρισης.

2.5 Διάτμιση - Segmentation

Η πληροφορία που δέχεται το σύστημα ως είσοδο τις περισσότερες φορές προέρχεται από το φυσικό κόσμο και ως επακόλουθο δεν περιλαμβάνει αποκλειστικά το στοιχείο προς αναγνώριση. Έτσι δημιουργείται η ανάγκη κατάτμησης της εισόδου σε τμήματα για να αξιοποιηθούν τα στοιχεία που απαιτείται και να κατηγοριοποιηθούν τα υπόλοιπα.

Στην όραση υπολογιστών, κατάτμιση είναι η διαδικασία τεμαχισμού μιας ψηφιακής εικόνας σε πολλαπλά τμήματα βασιζόμενοι σε διαφορετικά κριτήρια ανάλογα με την περίπτωση. Ο στόχος της κατάτμησης είναι η απλοποίηση ή η αλλαγή μιας εικόνας σε κάτι πιο ουσιαστικό και πιο εύκολο να κατανοηθεί ή να αναλυθεί. Η τμηματοποίηση εικόνων συχνά χρησιμοποιείται για τον εντοπισμό αντικειμένων και των ορίων τους σε γραμμές, καμπύλες, κ.λπ.. Πιο συγκεκριμένα, κατάτμιση εικόνας είναι η διαδικασία ανάθεσης μιας ετικέτας σε κάθε εικονοστοιχείο ή ομάδα εικονοστοιχείων μιας εικόνας έτσι ώστε ομάδες με τα ίδια ετικέτα να έχουν ορισμένα κοινά χαρακτηριστικά.

Οπότε το αποτέλεσμα της κατάτμησης μιας ψηφιακής εικόνας είναι ένα σύνολο τμημάτων που καλύπτουν όλο το σύνολο μιας εικόνας, ή μια σειρά από καμπύλες γραμμές που εξάγονται από την εικόνα (Αναγνώριση ακμών). Σε κάθε μία περιοχή το σύνολο από εικονοστοιχεία που περιλαμβάνει είναι παρόμοια σε σχέση με κάποιο χαρακτηριστικό ή ιδιότητα, όπως το χρώμα, η ένταση, η υφή.

2.5.1 Κατοφλιοποίηση

Η κατοφλιοποίηση αποτελεί την πιο απλή μέθοδο τμηματισμού. Βάση κάποιου χαρακτηριστικού όπως η φωτεινότητα, η θέση δημιουργείται μια δυαδική αναπαράσταση της εισόδου ανάλογα με τον αν βρίσκεται πάνω ή κάτω από μια καθορισμένη τιμή για το χαρακτηριστικό ή συνδυασμό χαρακτηριστικών που επιλέχθηκαν.

Κύριο στοιχείο της μεθόδου είναι η επιλογή της οριακής τιμής στην οποία θα βασιστεί η κατοφλιοποίηση. Σήμερα έχουν αναπτυχθεί διάφορες μέθοδοι για την επιλογή αυτής της τιμής όπως η μέθοδος μέγιστης εντροπίας η μέθοδος του Otsu, και η τμηματοποίηση K-means.

2.5.2 Τμηματοποίηση σε ομάδες

Ο αλγόριθμος K-means αποτελεί τεχνική η οποία χρησιμοποιείται για το διαχωρισμό και την ομαδοποίηση τμημάτων μιας ψηφιακής εικόνας. Ο βασικός αλγόριθμος αποτελείται από τα εξής βήματα:

- Επιλέγει K κεντρικά σημεία, ένα για κάθε ομάδα, είτε τυχαία ή βασιζόμενο σε κάποια ευρετική τεχνική

- Κάθε εικονοστοιχείο ανατίθεται σε μία από τις ομάδες και συγκεκριμένα σε αυτή που ελαχιστοποιεί την απόσταση ανάμεσα στο εικονοστοιχείο και το κέντρο της ομάδας
- Γίνεται απανυπολογισμός των κέντρων βασιζόμενος στο μέσο όρο όλων των εικονοστοιχείων που ανατέθηκαν σε μία ομάδα
- Τα βήματα 2 και 3 επαναλαμβάνονται μέχρι η τμηματοποίηση να βρεθεί σε ισορροπία, δηλαδή κανένα εικονοστοιχείο να μην αλλάξει ομάδα κατά μία επανάληψη του αλγορίθμου.

Όταν αναφερόμαστε σε απόσταση μεταξύ εικονοστοιχείου και κέντρων, εννοούμε ένα συνδυασμό με ανάλογα βάρη της διαφοράς στη θέση, την ένταση, την υφή. ο Αριθμός K επιλέγεται τυχαία ή όχι, αλλά ανεξάρτητα αυτού ο αλγόριθμος θα συγκλίνει μετά από ορισμένο πλήθος επαναλήψεων. Η ποιότητα της λύσης που θα προσφέρει εξαρτάται από την αρχική εκτίμηση της θέσης των κέντρων και τον επιλεγμένο αριθμό K .

Μέθοδοι που βασίζονται στη συμπύεση

Οι μέθοδοι αυτές βασίζονται στην υπόθεση πως η ιδανική τμηματοποίηση είναι αυτή που ελαχιστοποιεί το κωδικό μήκος των δεδομένων.

Η σύνδεση μεταξύ της κωδικοποίησης των δεδομένων και της τμηματοποίησης έγκειται στο ότι η τμηματοποίηση αναζητά patterns σε μία εικόνα και κάθε κοινό χαρακτηριστικό μεταξύ τμημάτων της εικόνας μπορεί να χρησιμοποιηθεί και για τη συμπύεση της.

Μέθοδος Ανίχνευσης Ακμών

Χρησιμοποιώντας την τεχνική της ανίχνευσης ακμών που προσφέρεται μεγάλη ποιότητα για των υπολογισμό τους μπορούμε να εξάγουμε περιοχές και να θεωρήσουμε πως έχουν κοινά χαρακτηριστικά και πρέπει να ταξινομηθούν στην ίδια ομάδα.

Μέθοδοι που βασίζονται στη διαστολή χωρίων

Η μέθοδος βασίζεται στην υπόθεση ότι γειτονικά εικονοστοιχεία θα έχουν και κοινές τιμές αν αυτά ανήκουν στην ίδια ομάδα. Έτσι αν ένα κριτήριο συγγένειας ικανοποιείται μεταξύ γειτονικών σημείων θεωρείται πως ανήκουν στην ίδια ομάδα και η διαδικασία επαναλαμβάνεται και για το κάθε νεοεισαχθέν μέλος στην ομάδα διαστέλλοντας έτσι τον χώρο της τμηματοποιημένης ομάδας.

Μέθοδοι μερικών διαφορικών εξισώσεων

Με τη χρήση μιας μεθόδου βασισμένη στη λύση μερικών διαφορικών εξισώσεων, μπορεί να επιτευχθεί τμηματοποίηση εικόνας. Μια δημοφιλής τεχνική είναι το Curve

propagation, μέσω του οποίου μπορεί να επιτευχθεί στερεοσκοπική αναδόμησή, παρακολούθηση αντικειμένων κ.α. Η κεντρική ιδέα είναι να δημιουργηθεί μια αρχική καμπύλη ως προς τη χαμηλότερη δυνατή τιμή μιας συνάρτησης κόστους.

Μέθοδοι μερικών διαφορικών εξισώσεων

Οι Λανγκρασιανές τεχνικές βασίζονται στην παραμετροποίηση του περιγράμματος της εικόνας βασιζόμενες σε μία στρατηγική δειγματοληψίας και στη συνέχεια αναπτύσσεται η κάθε τμηματοποίηση εφαρμόζοντας μεθόδους ελαχιστοποίησης ενέργειας steepest-gradient descent.

Μέθοδοι που βασίζονται σε Γεωμετρικά Μοντέλα

Η βασική υπόθεση που γίνεται από αυτή τη προσέγγιση είναι πως οι δομές ενδιαφέροντος μπορούν να αναπαρασταθούν από γεωμετρικές μορφές. Έτσι σκοπός της μεθόδου αποτελεί η εύρεση ενός probabilistic μοντέλου που θα περιγράφει τις μεταβολές στο σχήμα και την γεωμετρία και έτσι κατά τη τμηματοποίηση να χρησιμοποιηθεί ένα τέτοιο μοντέλο θέτοντας περιορισμός και διαχωρίζοντας την εικόνα σε τμήματα.

Μέθοδοι που βασίζονται σε ιστόγραμμα

Οι μέθοδοι αυτή είναι πολύ αποδοτικές συγκρινόμενες με άλλες καθώς τυπικά απαιτούν μόνο ένα πέρασμα των δεδομένων των εικονοστοιχείων. Με αυτή την τεχνική υπολογίζεται ιστόγραμμα από όλα τα εικονοστοιχεία μιας εικόνας και οι κορυφές και οι κοιλίες χρησιμοποιούνται για να αναγνωριστούν οι ομαδοποιήσεις στην εικόνα. Το χρώμα όπως και η ένταση χρησιμοποιούνται σαν δείκτες για την ομαδοποίηση.

Μία παραλλαγή της διαδικασίας αυτής, είναι να εφαρμόζεται η διαδικασία του ιστογράμματος αναδρομικά επαναλαμβάνοντας τη διαδικασία σε όλο και μικρότερες ομάδες στοιχείων μέχρι να μην είναι δυνατός εντοπισμός μικρότερων ομάδων.

Ημιαυτόματοι μέθοδοι

Σε τέτοιο είδος διάτμισης ο χρήστης εισάγει πληροφορία για το χωρίο που απαιτείται διάτμιση και με βάση αυτά τα δεδομένα εφαρμόζονται αλγόριθμοι για να ταιριάξει με καλύτερο τρόπο και να προσαρμοστεί το χωρίο στην εικόνα η περιγραφή που θέλουμε να διατμηθεί.

Μέθοδοι διάτμισης με εκμάθηση

Πολλές από τις τεχνικές διάτμισης βασίζονται στην αναγνώριση χαρακτηριστικών στο αντικείμενο και με βάση κοινά χαρακτηριστικά που θα εντοπιστούν προσφέρεται μία εκτίμηση για τις διάφορες ομάδες που περιέχονται. Με μία εικόνα ως παράδειγμα το χρώμα μπορεί να χρησιμοποιηθεί για να γίνει ένας διαχωρισμός σε περιοχές. Ένας εναλλακτικός τρόπος διάτμισης είναι μέσω εκπαίδευσης του συστήματος με ήδη τμηματοποιημένα παραδείγματα βάση των οποίων το σύστημα μπορεί να προσαρμοστεί και να είναι σε θέση να δώσει εκτιμήσεις διάτμισης για παρόμοια δεδομένα εισόδου με αυτά βάση των οποίων εκπαιδεύτηκε.

Για το σκοπό αυτό μπορούν να χρησιμοποιηθούν συστήματα νευρωνικών δικτύων ή μηχανικής μάθησης που χρησιμοποιώντας μικρά κομμάτια της πληροφορίας εισόδου μπορούν να παράγουν εκτίμηση για όρια κάθε τμήματος της εικόνας, σημειώνοντας το κάθε κομμάτι που επεξεργάστηκε με μία περιγραφή και τελικά συνθέτοντας όλες αυτές τις ετικέτες μαζί.

2.5.3 Τμηματοποίησή σε τρεις διαστάσεις

Plane Model Segmentation

Ο αλγόριθμος βασίζεται στη μέθοδο RANSAC που αποτελεί μία επαναληπτική μέθοδος υπολογισμού παραμέτρων ενός μοντέλου από ένα σύνολο τυχαία επιλεγμένων δεδομένων. Η μέθοδος αποτελεί μια μη-ντετερμινιστική μέθοδος, καθώς παράγει ένα λογικό αποτέλεσμα μόνο με συγκεκριμένη πιθανότητα, με την πιθανότητα να αυξάνεται σε κάθε επανάληψη. Η βασική υπόθεση ότι το σύνολο των δεδομένων αποτελείται από "inliers", σημεία που η κατανομή τους ανήκει σε ένα μοντέλο, και "outliers" δηλαδή σημεία που δεν ανήκουν σε αυτό το μοντέλο, τα οποία μπορεί να προέρχονται από ακραίες τιμές θορύβου ή λάθος μετρήσεις κατά τη σάρωση ή συλλογή δεδομένων.

Συνοπτικά η μέθοδος αποτελείται από τα εξής βήματα:

- Ένα μέρος από τα αρχικά δεδομένα επιλέγονται τυχαία τα οποία ονομάζονται υποθετικά
- Δημιουργείται ένα μοντέλο που προσαρμόζεται σε αυτά τα υποθετικά inliers
- Το σύνολο των δεδομένων συγκρίνονται με το μοντέλο και τα σημεία που ταιριάζουν με το μοντέλο, σύμφωνα με συγκεκριμένη συνάρτηση απώλειας, θεωρείται ότι ανήκουν στο σύνολο παραδοχής
- Το μοντέλο θεωρείται αρκετά καλό αν επαρκής αριθμός σημείων έχουν συμπεριληφθεί στο σύνολο παραδοχής

Με την εκτέλεση της διαδικασίας προκαθορισμένο αριθμό επαναλήψεων, παράγεται ένα μοντέλο που απορρίπτεται λόγω λίγων σημείων στο σύνολο παραδοχής, ή ένα αποδεκτό μοντέλο μαζί με το αντίστοιχο μέγεθος συνόλου παραδοχής. Στη περίπτωση ανίχνευσης ενός αποδεκτού μοντέλου αν το σύνολο των σημείων είναι μεγαλύτερο από αυτό του προηγούμενου αποδεκτού μοντέλου τότε αυτό σημαίνεται ως το τρέχον επικρατές μοντέλο, αλλιώς η διαδικασία επαναλαμβάνεται στην επόμενη επανάληψη. [27][28]

Cluster Extraction Segmentation

Ο αλγόριθμος αυτός χρησιμοποιείται με στόχο την ομαδοποίηση των σημείων του νέφους σε συστάδες σημείων, κάθε μια εκ των οποίων θα αποτελεί ιδανικά ένα αντικείμενο. Για ελαχιστοποίηση του χρόνου εκτέλεσης, ο αλγόριθμος χρησιμοποιεί ένα

μοντέλο RANSAC για την εύρεση των planar inliers. Έπειτα από αφαίρεση τους από το αρχικό νέφος σημείων, ακολουθούνται τα εξής βήματα:

1. Δημιουργείται μια αναπαράσταση σε δένδρο του νέφους P , των σημείων που απομένουν έπειτα από αφαίρεση των planar inliers.
2. Δημιουργείται μια λίστα συστάδων C
3. Δημιουργείται μια ουρά σημείων προς έλεγχο Q
4. Για κάθε σημείο P_i που ανήκει στο σύνολο P :
 - το p_i που ανήκει στην ουρά Q :
 - Για κάθε σημείο p_i που ανήκει στην ουρά Q :
 - (α') Το σύνολο P_k^i των γειτονικών σημείων του P_i που απέχουν από αυτό μέχρι προκαθορισμένη απόσταση, που αντιστοιχεί σε ακτίνα σφαίρας με κέντρο το P_i
 - (β') Για κάθε γειτονικό σημείο όπου $p_k^i \in P_k^i$ προστίθεται στη λίστα Q
 - (γ') Η λίστα Q ελέγχεται για διπλότυπα και αυτά αφαιρούνται ώστε να αποτελείται από διακριτά σημεία.
 - (δ') Η λίστα Q προστίθεται στις συστάδες C και το περιεχόμενο της σβήνεται για να χρησιμοποιηθεί στην επόμενη επανάληψη.

[34]

2.6 Αναγνώριση -Recognition

Η αναγνώριση προτύπων ορίζεται φορμαλιστικά ως η διαδικασία δια της οποίας ένα προσλαμβανόμενο σήμα/πρότυπο αντιστοιχίζεται σε μία κλάση που ανήκει σε ένα προκαθορισμένο σύνολο κλάσεων.

Η αναγνώριση patterns είναι κατά μεγάλο βαθμό συνώνυμη με την μάθηση μηχανής. Το παρακλάδι αυτό της τεχνητής νοημοσύνης εστιάζει στην αναζήτηση και αναγνώριση ομοιοτήτων σε δεδομένα. Σε πολλές περιπτώσεις προηγείται μάθηση του συστήματος αναγνώρισης με προσημασμένα δεδομένα(επιβλεπόμενη μάθηση) αλλά αναγνώριση μπορεί να επιτευχθεί χωρίς μάθηση με τη χρήση αλγορίθμων και τεχνικών(μη επιβλεπόμενη μάθηση).

Στόχος της αναγνώρισης είναι να αποδώσει στα στοιχεία εισόδου ένα χαρακτηρισμό για την ομάδα που ανήκουν, το είδος τους ή την αναγνώριση του ίδιου του στοιχείου εισόδου ονομαστικά. Η αναγνώριση προτύπων αποτελεί ένα γενικό πρόβλημα που βρίσκει

εφαρμογές σε μεγάλο εύρος επιστημονικών πεδίων, στην Ιατρική, τη βιομηχανία και άλλα.

Οι αλγόριθμοι και οι τεχνικές που χρησιμοποιούνται για την αναγνώριση προτύπων στοχεύουν στον να μπορέσουν να παράγουν μία πρόβλεψη ή απόφαση παίρνοντας υπόψη όλες τις εισόδους που δέχονται ως σύστημα. Ένα τυπικό παράδειγμα τέτοιας συμπεριφοράς είναι η αναζήτηση σε αρχεία κειμένου χρησιμοποιώντας (regular expressions). Αντιθέτως όμως με την αναγνώριση προτύπων, οι μέθοδοι ταύτισης προτύπων δεν θεωρείται πως αποτελούν είδος μηχανικής μάθησης αν και μπορούν πολλές φορές να επιτύχουν παρόμοια ή ίδια αποτελέσματα ανάλογα με την εφαρμογή.

Η αναγνώριση προτύπων μελετάται και χρησιμοποιείται σε πληθώρα τομέων όπως η ψυχολογία, η ψυχιατρική, η ηθολογία, σε συγκοινωνιακά συστήματα και την επιστήμη των υπολογιστών.

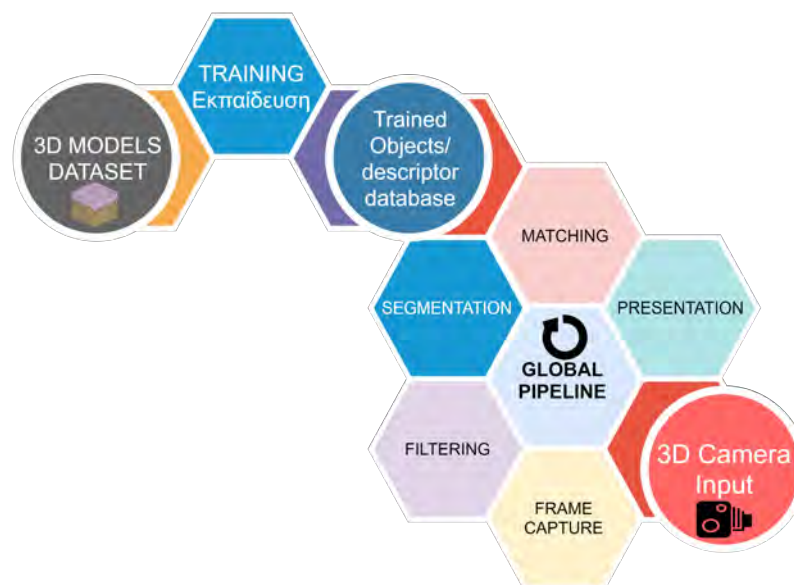
Κεφάλαιο 3

Ανάλυση και σχεδίαση

Στο κεφάλαιο αυτό παρουσιάζονται τα συστατικά στοιχεία του τελικού συστήματος, ο τρόπος που επικοινωνούν και η αλληλουχία εκτέλεσης και σύνδεσής τους.

3.1 Περιγραφή Αρχιτεκτονικής

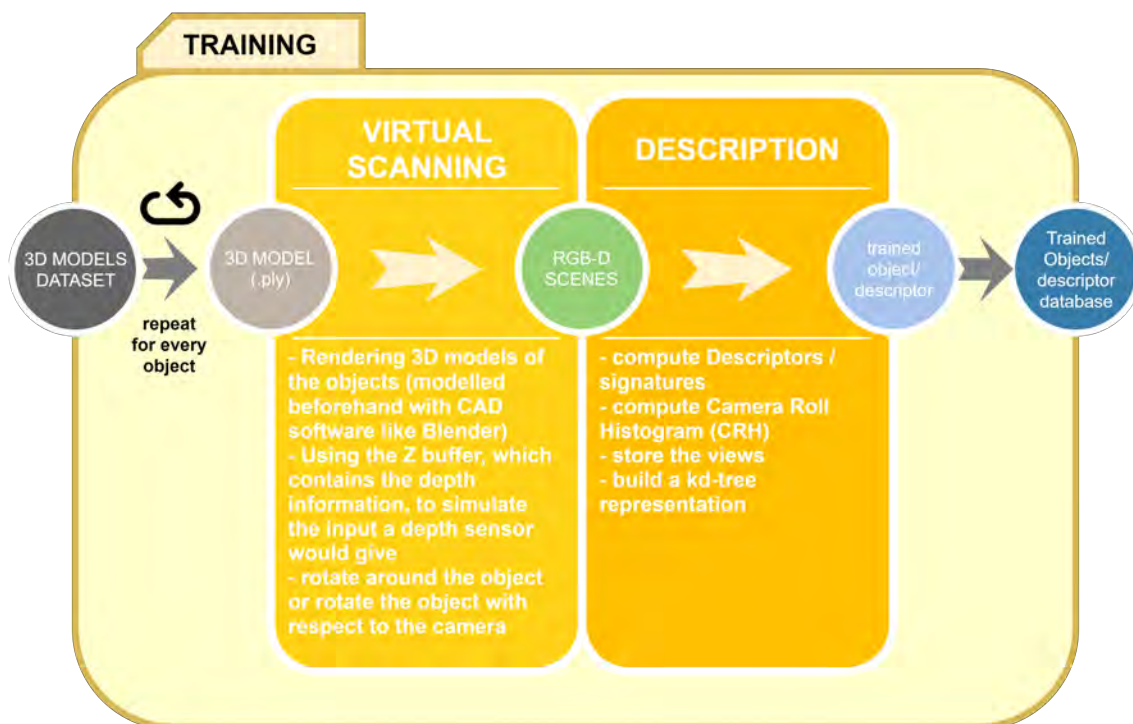
Το σύστημα χωρίζεται σε δύο βασικά υποσυστήματα, αυτό που αναλαμβάνει τη μάθηση των προς αναγνώριση αντικειμένων και την αποθήκευση των χαρακτηριστικών τους και το σύστημα που εκτελείται εν συνεχεία και αναλαμβάνει τη σε πραγματικό χρόνο αναγνώριση των αντικειμένων που δέχεται το σύστημα ως είσοδο.



(α') Pipeline Αναγνώρισης

Σχήμα 3.1

3.2 Εκπαίδευση



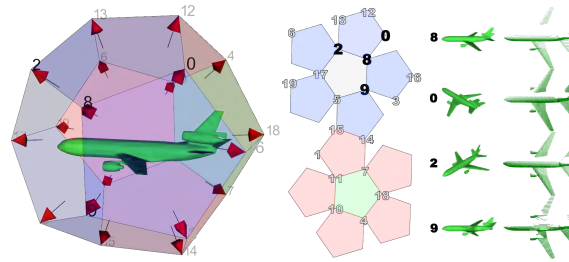
(α') Εκπαίδευση

Σχήμα 3.2

Για να γίνει εφικτό να αναγνωρίζει αντικείμενα, το σύστημα χρειάζεται να γνωρίζει κάποια από τα χαρακτηριστικά των αντικειμένων εκ των προτέρων. Έτσι, με είσοδο 3D αντικείμενα, δημιουργείται μία βάση δεδομένων με πληροφορίες και χαρακτηριστικά του κάθε αντικειμένου, προκειμένου βάση αυτών το σύστημα να συγκρίνει και να αναζητήσει την πιο πιθανή ταύτιση της εισόδου με τα γνωστά του δεδομένα. Για τη διαδικασία της εκπαίδευσης λοιπόν του συστήματος, δεν χρειάζεται η κάμερα να έχει σαρώσει το ίδιο το αντικείμενο, αλλά η εκμάθηση και η ενσωμάτωση του αντικειμένου στο σύστημα γίνεται μέσω προσχεδιασμένου τρισδιάστατου μοντέλου.

Για κάθε αντικείμενο που θέλουμε να προστεθεί στη βιβλιοθήκη του συστήματος, και με βάση το τρισδιάστατο μοντέλο που το περιγράφει, το σύστημα παράγει πόζες, από όλες τις πιθανές οπτικές γωνίες με βήμα και ακρίβεια που καθορίζεται έπειτα από πειραματισμό. Από αυτές τις πόζες γίνεται εξαγωγή χαρακτηριστικών και το σύστημα μαθαίνει και αποθηκεύει χαρακτηριστικά που περιγράφουν τη γεωμετρία του αντικειμένου ανεξαρτήτως του πως αυτό παρουσιάζεται στην κάμερα.

Έτσι για κάθε αντικείμενο, το σύστημα εκπαιδεύεται με βάση το τρισδιάστατο του μοντέλο, αποθηκεύονται ποικίλες πληροφορίες για κάθε πιθανή οπτική γωνία που μπορεί να εμφανιστεί. Το βήμα που θα καθοριστεί ανάμεσα στις πιθανές γωνίες εμφάνισης



(α') Εικονική σάρωση μοντέλου αεροσκάφους

Σχήμα 3.3

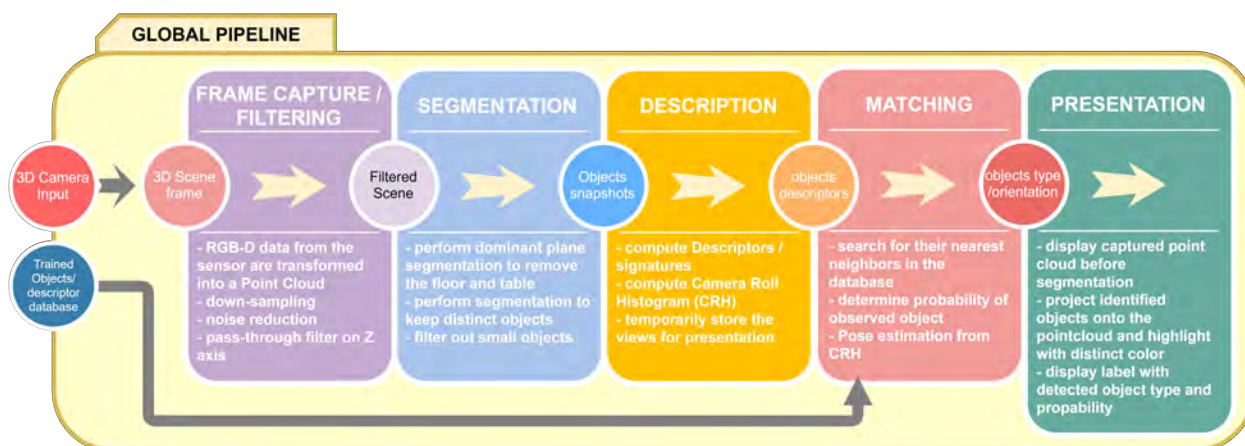
του αντικειμένου κατά την εκπαίδευση του στο σύστημα, παίζει σημαντικό ρόλο και στην ακρίβεια που θα δώσει το σύστημα ως αποτέλεσμα μετά την αναγνώριση του αντικειμένου.

Μετά από την παραγωγή των διάφορων οπτικών σκηνών του αντικειμένου, που αντιστοιχούν σε διαφορετική οπτική γωνία, για κάθε μία από αυτές γίνεται εξαγωγή χαρακτηριστικών και αποθήκευση. Επιπρόσθετα, απαιτείται ο υπολογισμός του Camera Roll Histogram (CRH) προκειμένου να είναι γνωστή η πόζα του αντικειμένου με 6 βαθμούς ελευθερίας.

Η εξαγωγή χαρακτηριστικών γίνεται σύμφωνα με τον περιγραφέα OUR CVFH που εκμεταλλεύεται πληροφορίες για τον προσανατολισμό και τη γεωμετρία.

Τέλος, υπολογίζονται και αποθηκεύονται πληροφορίες για την εικονική κάμερα της κάθε σκηνής. Η πληροφορία αυτή αφορά την θέση σε σχέση με τα σημεία του αντικειμένου και τον προσανατολισμό της στο τρισδιάστατο επίπεδο. Η πληροφορία αποθηκεύεται στη μορφή ενός πίνακα περιστροφής 3×3 (rotation matrix) και με έναν τριών διαστάσεων πίνακα μεταφοράς (3D translation vector).

3.3 Global Pipeline



(α') Pipeline Αναγνώρισης

Σχήμα 3.4

3.3.1 Συλλογή Δεδομένων

Η συλλογή των δεδομένων λαμβάνει χώρα πριν την αναγνώριση και κατά τη διάρκεια της αναγνώρισης.

Η πρώτη γίνεται ανεξαρτήτως χρόνου πριν το σύστημα μπει στο στάδιο της κύριας λειτουργίας του. Η συλλογή δεδομένων αυτή αφορά την εκπαίδευση, όπως περιγράφηκε, με βάση τα τρισδιάστατα μοντέλα των αντικειμένων. Οι πληροφορίες αυτές αποθηκεύονται μόνιμα και δεν χρειάζονται να παραχθούν ξανά, εκτός αν υπάρξει επικαιροποίηση ή αλλαγή στο 3D μοντέλο που περιγράφει κάποιο αντικείμενο.

Η συλλογή δεδομένων κατά την κύρια λειτουργία συμβαίνει σε πραγματικό χρόνο, πληροφορίες και χαρακτηριστικά που εξάγονται από την τρισδιάστατη είσοδο δεδομένων στο σύστημα αποθηκεύονται προσωρινά για ένα κύκλο ολοκλήρωσης του αλγορίθμου αλλά και για επόμενους προκειμένου το σύστημα με βάση την ιστορία των δεδομένων εισόδου να μπορεί να απαλλαγθεί και να ελαχιστοποιήσει σφάλματα εισόδου.

Κατά τη λειτουργία της εφαρμογής και σε τακτά διαστήματα, συλλέγονται δεδομένα από την 3D αισθητήρια κάμερα, τα οποία αμέσως μετά τη συλλογή τους μετατρέπονται σε νέφος σημείων για να φιλτραριστούν και να αξιοποιηθούν από την εφαρμογή.

3.3.2 Μετατροπή σε νέφος σημείων

Η εικόνα στο κάτω μέρος του Σχήματος 3.4 αναπαριστά με χρώμα τα δεδομένα που συλλέχθηκαν από την κάμερα. Πάνω φαίνεται το νέφος σημείων όπως δημιουργήθηκε από τα δεδομένα της κάμερας. Λόγω της διάταξης του χώρου και των αντικειμένων

πολλά σημεία του δωματίου δεν απεικονίζονται στο νέφος. Τα σημεία αυτά του χώρου φαίνονται λευκά στην πάνω εικόνα καθώς κανένα σημείο του νέφους δεν υπάρχει στις συγκεκριμένες συντεταγμένες. Ο λόγος που συμβαίνει αυτό είναι πως συγκεκριμένα αντικείμενα του χώρου και στο προκείμενο οι καρέκλες του δωματίου κρύβουν από το οπτικό πεδίο της κάμερας μεγάλα κομμάτια του δωματίου και έτσι αυτά δεν απεικονίζονται.

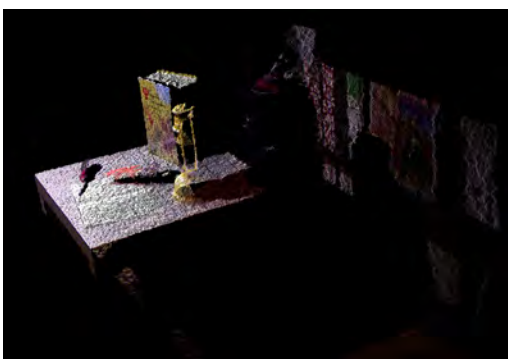
Το πρόβλημα αυτό μπορεί να αποφευχθεί με τη χρήση περισσότερων του ενός αισθητήρων και συνθέτοντας τα δεδομένα για να καταλήξουμε σε μία ολοκληρωμένη αναπαράσταση του χώρου. Στην εφαρμογή μας θα ασχοληθούμε με αντικείμενα που είναι στο οπτικό πεδίο της κάμερας. Όπου υπάρχουν επικαλύψεις το σφάλμα αυτό που εισαγάγεται θα εξαλειφθεί με τη χρήση αλγορίθμων που μας επιτρέπουν την αναγνώριση ακόμα και με τμηματική απεικόνιση των αντικειμένων.



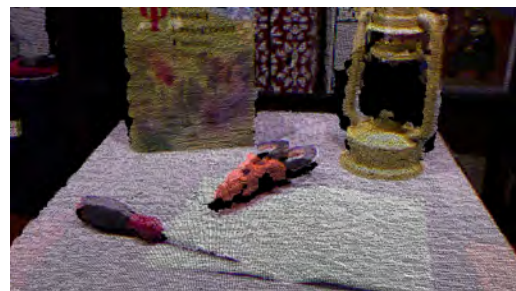
(α') Εικόνα RGB



(β') Pointcloud



(γ') Pointcloud 2



(δ') Pointcloud 3

Σχήμα 3.5: Σκηνή και νέφος Σημείων

3.3.3 Προεπεξεργασία

Το νέφος σημείων που περιγράφει το χώρο μαζί με τα αντικείμενα, όπως σαρώνεται και δίνεται ως είσοδος στο σύστημα από τη συσκευή, έχει υψηλή ανάλυση, δίνοντας μεγάλος πλήθος από δεδομένα και κάνοντας έτσι την επεξεργασία τους πιο δαπανηρή σε χρόνο αλλά και σε υπολογιστική ισχύ. Παράλληλα υψηλή ανάλυση νέφους σημαίνει πως μικρές διαφορές που δημιουργούνται σε γειτονικά εικονοστοιχεία λόγω θορύβου και άλλων μικροαλλαγών από σάρωση σε σάρωση, θα δυσχεραίνουν την εξαγωγή χαρακτηριστικών και την επιτυχή αναγνώριση. Έτσι πριν την εισαγωγή του, το νέφος υφίσταται υποδειγματολειτουργία και στη συνέχεια γκαουσιανό φιλτράρισμα απαλείφοντας έτσι σφάλματα.

Ο βαθμός της υποδειγματολειτουργίας αυτής προκύπτει έπειτα από δοκιμές, καθώς αν γίνει σε μεγάλο βαθμό το σύστημα δεν θα μπορεί να αναγνωρίσει όλα τα αντικείμενα επιτυχώς, ενώ αν γίνει σε μικρό βαθμό το σύστημα θα είναι πολύ αυστηρό, αναγνωρίζοντας όλα τα αντικείμενα στην ίδια ομάδα ή και αδυνατώντας να τα κατατάξει.

Επειδή τα τρισδιάστατα μοντέλα έχουν δημιουργηθεί υπό διαφορετικές συνθήκες απαιτείται κανονικοποίηση των χαρακτηριστικών του συνόλου των δειγμάτων έτσι ώστε το σύστημα να μην ευνοεί κάποια από τις κατηγορίες και να της δίνει μεγαλύτερη πιθανότητα να είναι αυτή που θα επιλεγεί για το αντικείμενο.

3.3.4 Φιλτράρισμα

Καθώς η σκηνή που θα χρησιμοποιηθεί για την τελική αναγνώριση των αντικειμένων περιέχει και άλλα στοιχεία όπως έπιπλα και δομικά στοιχεία, αυτά εύκολα μπορούν να αφαιρεθούν μετά τη σάρωση του χώρου.

Το αρχικό μοντέλο που προκύπτει από τη σάρωση υπόκειται σε κατωφλιοποίηση ως προς τη διάσταση του χώρου που αντιπροσωπεύει το βάθος έτσι ώστε να περιέχει τα πιο κοντινά στην συσκευή καταγραφής αντικείμενα, σύμφωνα με ένα ποσοστό που καθορίζεται.

Στη συνέχεια το μοντέλο υπόκειται σε γκαουσιανό φιλτράρισμα και κανονικοποίηση εξαλείφοντας έτσι τις όποιες ανωμαλίες και σφάλματα που προέρχονται από τη συσκευή σάρωσης το χώρο ή της συνθήκες του χώρου που γίνεται η καταγραφή.

Με το φιλτράρισμα η επεξεργασία γίνεται πιο γρήγορη καθώς αφαιρούνται τα δεδομένα που δεν χρειάζονται για την αναγνώριση, αλλά και το αποτέλεσμα της αναγνώρισης αποκτά περισσότερη ακρίβεια καθώς αφαιρούνται στοιχεία που θα εισαγάγουν πιθανό θόρυβο αλλοιώνοντας το αποτέλεσμα ή προκαλώντας λανθασμένη αναγνώριση.

3.3.5 Αναγνώριση επιφανειών

Μετά το φιλτράρισμα και τη μετατροπή σε νέφος, εκτελείται μία διαδικασία υπολογισμού και αναγνώρισης οριζόντιων επιφανειών όπως πάτωμα, τραπέζι. Οι επιφάνειες αυτές θέλουμε να αφαιρεθούν από το βασικό νέφος έτσι ώστε να απομονωθούν τα νέφη των αντικείμενων που θέλουμε να αναγνωριστούν.

Μετά την αναγνώριση των επιφανειών στη σκηνή, η μεγαλύτερη σε εμβαδόν επιλέγεται και αφαιρείται από το νέφος. Αυτή η διαδικασία εκτελείται ανάλογα με τις ανάγκες και τις συνθήκες δύο φορές. Η πρώτη φορά αφαιρεί το πάτωμα του χώρου που σαρώθηκε και η δεύτερη το τραπέζι στο οποίο τοποθετούνται τα αντικείμενα.

3.3.6 Διάτμηση

Γειτνιάζοντα σημεία που παρουσιάζουν κοινά χαρακτηριστικά μπορούν να ομαδοποιηθούν και να χωριστούν σε κατηγορίες. Ακόμα σημεία που παρουσιάζουν σημαντικές διαφορές στα χαρακτηριστικά που συλλέχθηκαν μπορούν να αποτελέσουν τη διαχωριστική τομή μεταξύ δύο κατηγοριών.

Έτσι η εικόνα χωρίζεται σε κατηγορίες που μπορεί ήδη να αποτελούν ένα ξεχωριστό αντικείμενο ή δομικό κομμάτι ενός αντικείμενου του χώρου. Με τον τρόπο αυτό το νέφος σημείων αποτελείται πλέον από μικρότερα νέφη

3.3.7 Εξαγωγή χαρακτηριστικών

Κάθε ομάδα σημείων που προέκυψε από τη διάτμηση και το φιλτράρισμα αποτελεί ένα πιθανό αντικείμενο προς αναγνώριση. Από κάθε αντικείμενο εξάγονται τα ίδια χαρακτηριστικά με αυτά που είχαν παραχθεί κατά τη διαδικασία της μάθησης. Στο συγκεκριμένο σύστημα γίνεται εξαγωγή του CRH (Camera Roll Histogram) και του OUR - CVFH.

3.3.8 Αναγνώριση

Για κάθε σετ από χαρακτηριστικά που συλλέχθηκαν για κάθε αντικείμενο, προκειμένου να αναγνωριστεί, συγκρίνεται η ομοιότητα του με κάποια από τα χαρακτηριστικά που το σύστημα γνωρίζει ήδη από τη βάση του που δημιουργήθηκε κατά το στάδιο της εκπαίδευσης. Έτσι, το δέντρο που δημιουργήθηκε κατά την εκπαίδευση χρησιμοποιείται για να εντοπιστεί το αντικείμενο αυτό που καλύτερα αντιπροσωπεύει τα χαρακτηριστικά που συλλέχθηκαν από την πραγματική σάρωση. Εφόσον κατά την εκπαίδευση έγινε προβολή του τρισδιάστατου μοντέλου από πολλές διαφορετικές οπτικές γωνίες λήφει το σύστημα να βρει μεγαλύτερη ομοιότητα με το αντικείμενο που αντιστοιχεί αλλά και με την όψη που είναι πιο κοντά στην γωνία λήψης της σκηνής. Χρησιμοποιώντας την

πληροφορία από το CRH της κάθε λήψης μπορεί να εξαχθεί και πληροφορία για την δεδομένη γωνία λήψης που τελικά επιλέχθηκε ως αυτή με την μεγαλύτερη πιθανότητα ταύτισης.

Το υποσύστημα της αναγνώρισης, μαζί με το όνομα του αντικειμένου που αναγνωρίστηκε, δίνει ως αποτέλεσμα και την πιθανότητα ταύτισης ως αριθμό.

Κεφάλαιο 4

Υλοποίηση

4.1 Λεπτομέρειες υλοποίησης

Για την χρήση της εφαρμογής που δημιουργήθηκε απαιτείται ηλεκτρονικός υπολογιστής με τα εκτελέσιμα αρχεία και τις απαιτούμενες βιβλιοθήκες προφορτωμένες. Σε αυτόν συνδέεται ο 3D αισθητήρας ο οποίος σκανάρει σε πραγματικό χρόνο το περιβάλλον στο οποίο θα λάβει χώρα η αναγνώριση. Στο οπτικό πεδίο της η κάμερα, πρέπει να περιλαμβάνεται δάπεδο ή άλλη επίπεδη επιφάνεια όπως ένα τραπέζι στην επιφάνεια του οποίου θα τοποθετούνται τα αντικείμενα προς αναγνώριση.

4.1.1 Περιβάλλον Υλοποίησης

Η υλοποίηση έγινε σε λειτουργικό σύστημα UNIX(Ubuntu 18.04) και σε αρχιτεκτονική επεξεργαστή x86. Η κάμερα που χρησιμοποιήθηκε είναι Microsoft Kinect for Xbox360. Κάμερες που υποστηρίζονται από την εφαρμογή:

- Microsoft Kinect for Xbox360
- Asus Xtion Pro
- Primesence Sensors

Το Ολοκληρωμένο περιβάλλον ανάπτυξης (Integrated Development Environment) και λογισμικό που χρησιμοποιήθηκε για την ανάπτυξη των προγραμμάτων είναι το KDevelop IDE (kdevplatform).

4.1.2 Γλώσσες Προγραμματισμού - Βιβλιοθήκες

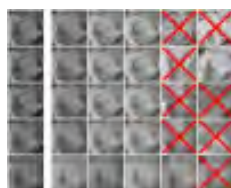
Ο κώδικας της υλοποίησης αυτής της εργασίας γράφτηκε σε γλώσσα προγραμματισμού C++ με χρήση και ενσωμάτωση των εξής βιβλιοθηκών:



Boost: Προέκταση των βιβλιοθηκών της C++ με την προσθήκη λειτουργιών και βελτιστοποιήσεις.



Eigen (template library for linear algebra): Βιβλιοθήκη για αλγεβρικές πράξεις που περιλαμβάνει συναρτήσεις για πίνακες, vectors, τελεστές και μαθηματικούς αλγόριθμους.



FLANN: Βιβλιοθήκη που αναλαμβάνει την αναζήτηση του κοντινότερου γειτονικού σημείου σε χώρους πολλών διαστάσεων. Περιλαμβάνει μία συλλογή αλγορίθμων, ενώ επιλέγει κατάλληλα, ανάλογα με τα δεδομένα εισόδου ποιος είναι ο βέλτιστος για την κάθε περίπτωση και με τι περιορισμούς και κριτήρια θα εκτελεστεί ο κάθε αλγόριθμος.



VTK: Βιβλιοθήκη επεξεργασίας εικόνας και απεικόνισης γραφικών.



OpenNI: Βιβλιοθήκη διασύνδεσης με συσκευές 3D απεικόνισης και καταγραφής.



CUDA: Βιβλιοθήκη που κάνει χρήση πλατφόρμας παράλληλης επεξεργασίας για κάρτες γραφικών. Οι πολλαπλοί πυρήνες της GPU χρησιμοποιούνται παράλληλα για υπολογισμούς με υψηλή ταχύτητα.



PCL(Point Cloud Library): Βιβλιοθήκη που χρησιμοποιεί αντικείμενα για εύκολη χειραγώγηση, επεξεργασία, μετατροπή και αποθήκευση νέφους σημείων.

4.2 Περιγραφή αλγορίθμων

Παρακάτω αναφέρονται και παρουσιάζονται τα σημαντικότερα σημεία, αποσπάσματα του ενιαίου αλγορίθμου ο οποίος και βρίσκεται πλήρης στο Παράρτημα του τόμου, με αναλυτική περιγραφή.

4.2.1 Υπολογισμός indices

Πολλοί από τους αλγορίθμους που αναπτύχθηκαν και που είναι διαθέσιμοι σε βιβλιοθήκες επιστρέφουν δείκτες σημείων μέσα σε ένα νέφος και όχι όλα τα δεδομένα του. Ως παράδειγμα, ένας αλγόριθμος κυλινδρικής τμηματοποίησης επιστρέφει όλα τα σημεία που θεωρούνται κομμάτι ενός κυλινδρικού μοντέλου. Οι δείκτες που επιστρέφονται μπορούν να χρησιμοποιηθούν για να εξαχθούν σημεία του νέφους σε ένα δεύτερο νέφος ή το αρνητικό τους πλήθος από σημεία.

Κώδικας 4.1: Υπολογισμός indices

```
{  
// Objects for storing the point-clouds  
pcl::PointCloud<pcl::PointXYZ>::Ptr cloudAll(new pcl::PointCloud<pcl::PointXYZ  
    >);  
pcl::PointCloud<pcl::PointXYZ>::Ptr cloudExtracted(new pcl::PointCloud<pcl::  
    PointXYZ>);  
  
...  
//compute coefficients  
...  
  
// Object for storing the indices  
pcl::PointIndices::Ptr pointIndices(new pcl::PointIndices);  
extract.setInputCloud(cloudAll);  
extract.setIndices(pointIndices);  
  
//extracting the points not indexed  
extract.setNegative(true);  
extract.filter(*cloudExtracted);  
}
```

4.2.2 Αφαίρεση σημείων χωρίς δεδομένα (NaNs)

Το νέφος που σαρώνεται από τον αισθητήρα, μπορεί να περιλαμβάνει λάθη μετρήσεων και σημεία που δεν μπόρεσε να μετρηθεί η τιμή τους. Μία τέτοια περίπτωση είναι τα σημεία με τιμή NaN (Not a Number).

Ένα νέφος μπορεί να χαρακτηριστεί πυκνό νέφος όταν όλα τα σημεία που το αποτελούν έχουν μία τιμή που μπορεί να μετρηθεί. Η ύπαρξη Nans λόγω σφάλματος, επειδή ο αισθητήρας μπορεί να ήταν πολύ κοντά στο αντικείμενο ή λόγω ανάκλασης, καθιστά πιο δύσκολη την επεξεργασία του νέφους και έτσι είναι ορθό σαν βήμα προ-επεξεργασίας να αφαιρεθούν.

Κώδικας 4.2: Αφαίρεση NaNs

```
// Object for storing the point cloud.  
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);  
...  
std::vector<int> mapping;  
...  
pcl::removeNaNFromPointCloud(*cloud, *cloud, mapping);
```

Ένα πρόβλημα όμως που προκύπτει με την εφαρμογή του παραπάνω αλγορίθμου είναι πως το νέφος δεν είναι πια οργανωμένο. Οργανωμένο θεωρείται ένα νέφος όπου το πλήθος των σημείων που το αποτελούν ισούται με το πλάτος και το ύψος είναι 1. Επειδή η αφαίρεση των Nans αλλάζει το πλήθος των σημείων το νέφος δεν μπορεί να παραμείνει οργανωμένο με την αρχική αναλογία πλάτους/ύψους, οπότε και η συνάρτηση `removeNaNFromPointCloud` θέτει το ύψος σε 1.

Το παραπάνω πρέπει να λαμβάνεται υπ όψιν όταν αναπτύσσονται αλγόριθμοι, έτσι ώστε να μην βασίζονται στη γεωμετρία του νέφους με τη υπόθεση του οργανωμένου νέφους.

4.2.3 Υπολογισμούς του κεντρικού σημείου ενός νέφους (centroid)

Το centroid ενός νέφους είναι το σημείο με συντεταγμένες που προκύπτουν μετά από υπολογισμό του μέσου των τιμών όλων των σημείων του νέφους. Μπορεί να γίνει κατανοητό αν παρομοιαστεί με το "κέντρο βάρους" του νέφους. Για νέφη που σαρώνονται από αισθητήρα βάρους, επειδή το νέφος περιλαμβάνει μόνο το θεατό από τον αισθητήρα κομμάτι μίας σκηνης ή ενός αντικειμένου, είναι σημαντικό να λυθεί υπ όψιν ότι ένας αλγόριθμος μπορεί να υπολογίσει το centroid του τμήματος του αντικειμένου.

Κώδικας 4.3: Αφαίρεση NaNs

```
// Object for storing the point cloud.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);

// Object to store the centroid coordinates.
Eigen::Vector4f centroid;

pcl::compute3DCentroid(*cloud, centroid);

std::cout << "XYZ coordinates of the centroid: ("
            << centroid[0] << ", "
            << centroid[1] << ", "
            << centroid[2] << ")." << std::endl;
```

4.2.4 Υπολογισμός features

Όπως αναλύονται στο κεφάλαιο 2, τα features είναι τα χαρακτηριστικά ενός σημείου που μπορούν να βοηθήσουν να το διαχωρίσουμε από τα υπόλοιπα, πέρα από τις τιμές των συντεταγμένων του. Τα χαρακτηριστικά υπολογίζονται πάντα σε σχέση με τη θέση του σημείου ανάλογα με τα γειτονικά του ή ως προς όλο το πλήθος των σημείων. Έτσι ένα χαρακτηριστικό πρέπει να εμφανίζει παρόμοια αποτελέσματα όταν υπολογίζεται για το ίδιο σημείο μίας επιφάνειας σε διαφορετικά set μετρήσεων.

4.2.5 Υπολογισμός Normals

Το διάνυσμα κατεύθυνσης (normal) είναι ένα διάνυσμα κάθετο ως προς μία επιφάνεια. Τέτοια διανύσματα πάνω σε μία επιφάνεια μπορούν να υπολογιστούν και για ένα νέφος σημείων και παρόλο που δεν μπορεί μόνο του να αποτελέσει σημαντική ένδειξη, αποτελεί και αυτό ένα feature. Ο αλγόριθμος για τον υπολογισμό του βασίζεται στη μέθοδο των κοντινότερων γειτόνων, η οποία μέθοδος μπορεί να παραμετροποιηθεί και να βασίζεται σε κωνικό, κυκλικό ή σφαιρικό μοντέλο το οποίο χρησιμοποιείται συνήθως για υπολογισμό σε τρεις διαστάσεις.

Εκτός από την κατεύθυνση, τα normals μας δίνουν ενδείξεις και για την καμπυλότητα μίας επιφάνειας σε κάθε σημείο. Οι περισσότεροι αλγόριθμοι υπολογισμού χαρακτηριστικών προϋποθέτουν τον υπολογισμό των normals.

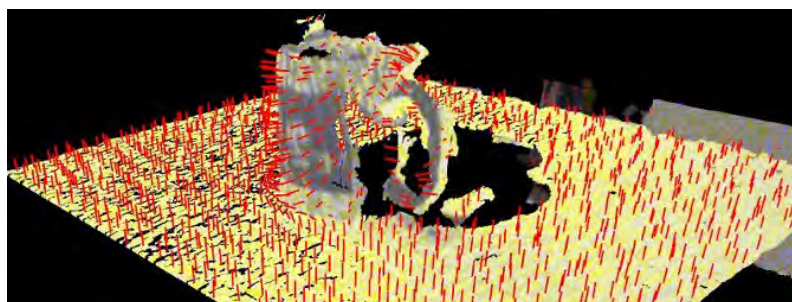
Κώδικας 4.4: Υπολογισμός Normals

```
// Object for storing the point cloud.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
// Object for storing the normals.
pcl::PointCloud<pcl::Normal>::Ptr normals(new pcl::PointCloud<pcl::Normal>);
```

```
// Object for normal estimation.
pcl::NormalEstimation<pcl::PointXYZ, pcl::Normal> normalEstimation;

normalEstimation.setInputCloud(cloud);
normalEstimation.setRadiusSearch(0.03);
// The normal estimation object will use it to find nearest neighbors. pcl::search::
    KdTree<pcl::PointXYZ>::Ptr kdtree(new pcl::search::KdTree<pcl::PointXYZ>);
normalEstimation setSearchMethod(kdtree);

normalEstimation.compute(*normals);
```



(α')

Σχήμα 4.1: Υπολογισμός Normals

4.2.6 Decomposition

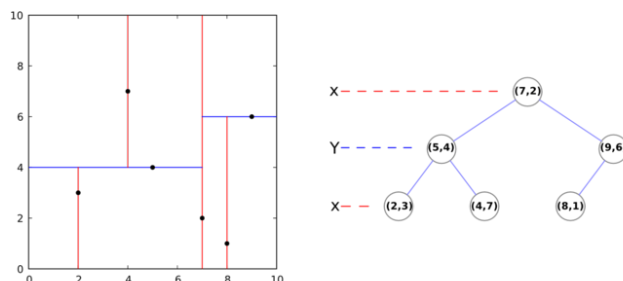
Με τον όρο Decomposition, αναφερόμαστε στις διαδικασίες εξαγωγής και δημιουργίας μίας οργανωμένης δομής από δεδομένα, που προέρχονται από ένα πλήθος σημείων, με σκοπό η ανάλυση και επεξεργασία να γίνει ευκολότερη.

4.2.7 Δέντρο k διαστάσεων (k-d tree)

Ένα δέντρο k διαστάσεων είναι μία δομή δεδομένων που οργανώνει μία ομάδα σημείων σε ένα χώρο k-διαστάσεων. Η οργάνωση αυτή γίνεται με τέτοιο τρόπο που καθιστά ενέργειες όπως η αναζήτηση πολύ γρήγορες και αποδοτικές. Παραδείγματος χάρη η εύρεση του κοντινότερου σημείου στο χώρο γίνεται πολύ πιο γρήγορα αν έχει προηγηθεί τακτοποίηση των δεδομένων σε δέντρο.

Το k-d δέντρο είναι ένα διαδικό δέντρο, με κάθε σημείο όπου δεν είναι φύλλο (τερματικό), να έχει δύο παιδιά, συνήθως απεικονιζόμενα, το ένα αριστερά και το άλλο δεξιά. Οι κόμβοι ενός δέντρου μπορούν να χωριστούν σε επίπεδα, με την έννοια ότι κόμβοι που απέχουν το ίδιο από τη ρίζα βρίσκονται στο ίδιο επίπεδο. Πιο αυστηρά, όλοι οι κόμβοι των οποίων τα μονοπάτια από τη ρίζα προς αυτά έχουν το ίδιο μήκος, ανήκουν

στο ίδιο επίπεδο. Τα δέντρα είναι εξορισμού γράφοι συνεκτικοί και δεν περιέχουν κύκλους. Συνέπεια των ιδιοτήτων αυτών είναι ότι αν ένα δέντρο έχει n κορυφές, τότε έχει $n-1$ ακμές. Επιπλέον, κάθε δέντρο έχει τουλάχιστον έναν κόμβο βαθμού 1. Βαθμός ενός κόμβου είναι το πλήθος των προσκείμενων σ' αυτόν ακμών.



(α') Δέντρο k διαστάσεων
Σχήμα 4.2

Κώδικας 4.5: Υπολογισμός γειτόνων με k -d tree

```
// Object for storing the point cloud.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
// kd-tree object.
pcl::search::KdTree<pcl::PointXYZ> kdtree;

kdtree.setInputCloud(cloud);

pcl::PointXYZ point;
point.x=1.0;
point.y = -0.5;
point.z = 1.9;

//Vector to store the output 5 neighbors.
std::vector<int> pointIndices(5);
//Vector to store the squared distances to the initial point.
std::vector<float> squaredDistances(5);

if (kdtree.nearestKSearch(point, 5, pointIndices, squaredDistances) > 0)
{
    std::cout << "The 5 nearest neighbors to the point:" << std::endl;
    for (size_t i = 0; i < pointIndices.size(); ++i)
        std::cout << "\t" << cloud->points[pointIndices[i]].x
            << " " << cloud->points[pointIndices[i]
```

```

        ]]..y
        << " " << cloud->points[pointIndices[i]]..z
        << " (squared distance: " <<
            squaredDistances[i] << ")" << std::
            endl;
    }

// (inside a sphere of radius 3cm centered at the point).
if (kdtree.radiusSearch(point, 0.03, pointIndices, squaredDistances) > 0)
{
    std::cout << "Neighbors within 3cm:" << std::endl;
    for (size_t i = 0; i < pointIndices.size(); ++i)
        std::cout << "\t" << cloud->points[pointIndices[i]].x
            << " " << cloud->points[pointIndices[i]]..y
            << " " << cloud->points[pointIndices[i]]..z
            << " (squared distance: " <<
                squaredDistances[i] << ")" << std::
                endl;
}

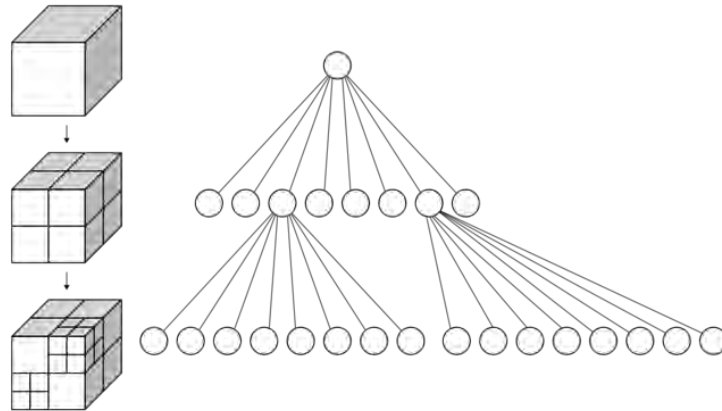
```

Συγκεκριμένα για τον υπολογισμό των κοντινότερων γειτόνων και εναλλακτικά θα μπορεί να χρησιμοποιηθεί πιο εξειδικευμένος αλγόριθμος με δέντρα που παρέχεται από την βιβλιοθήκη FLANN (Fast Library for Approximate Nearest Neighbors).

4.2.8 Δέντρο Octree

Όπως και με το k-tree, το octree είναι μία ιεραρχική δενδρική δομή δεδομένων, που οι ιδιότητες του και η δομή του βοηθάνε σε αναζητήσεις αλλά και σε διαδικασίες συμπίεσης και αποσυμπίεσης δεδομένων.

Κάθε κόμβος του octree που συνήθως αναφέρεται ως voxel (όρος που χρησιμοποιείται από σχεδιαστές 3D) και μπορεί να παρομοιαστεί με ένα τρισδιάστατο pixel, αποτελεί ένα κύβο με οκτώ παιδιά ή κανένα. Ο αρχικός κόμβος περιγράφει του ευρύτερο κύβο που περιλαμβάνει όλα τα σημεία. Σε κάθε επίπεδο του δέντρου, το πλήθος σημείων και η ανάλυση αυξάνεται μία τάξη μεγέθους. Με αυτό το τρόπο μπορούμε να ρυθμίσουμε την ανάλυση κάποιων ζώνης σημείων.



(α') Δέντρο Octree

Σχήμα 4.3

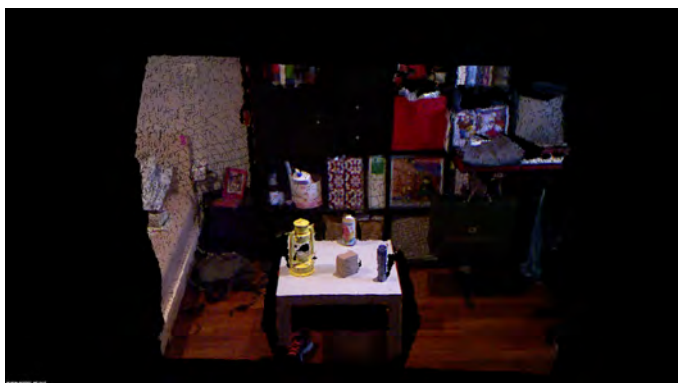
Κώδικας 4.6: Συμπίεση δεδομένων cloud

```
// Object for storing the point cloud.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointXYZ>::Ptr decompressedCloud(new pcl::PointCloud<pcl::PointXYZ>);

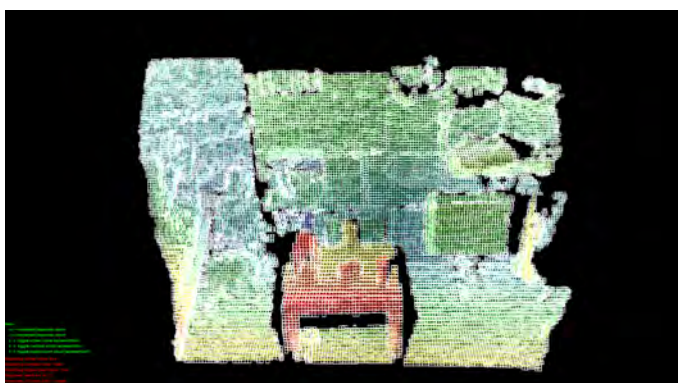
pcl::io::OctreePointCloudCompression<pcl::PointXYZ> octreeCompression(pcl::io::
    MED_RES_ONLINE_COMPRESSION_WITHOUT_COLOR, true);

std::stringstream compressedData;
// Compression
octreeCompression.encodePointCloud(cloud, compressedData);

// Decompression octreeCompression.decodePointCloud(compressedData,
    decompressedCloud);
```

(α') Pointcloud



(β') Octree

Σχήμα 4.4: Δέντρο Octree

Όπως φαίνεται στο παράδειγμα, τα σημεία του νέφους έχουν μειωθεί, κάτι που μπορεί να εμφανιστεί πολύ χρήσιμο στην επεξεργασία των δεδομένων του νέφους.

4.2.9 Φιλτράρισμα

Τα δεδομένα όπως συλλέγονται από τους αισθητήρες αφού μετατραπούν σε νέφος συνήθων χρειάζονται μια προ-επεξεργασία έτσι ώστε να αποτελέσουν καλή είσοδο για τους υπολογιστικούς αλγόριθμους. Συνήθως αυτό χρειάζεται λόγω μικρής ακρίβειας του αισθητήρα, λόγω θορύβου, ανακλάσεων και άλλων συνθηκών που μπορεί να προκαλέσουν λανθασμένες μετρήσεις. Από την άλλη, πολλές φορές χρειάζεται η επεξεργασία ή ανάλυση τμήματος του νέφους που σαρώθηκε από τον αισθητήρα και όχι όλης της σκηνής.

4.2.10 Φίλτρο Pass-through

Ένα pass-through φίλτρο όταν εφαρμοστεί σε ένα νέφος σημείων παρέχει τη δυνατότητα να αφαιρεθεί ένα συγκεκριμένο πλήθος από σημεία τα οποία δεν ανήκουν σε ένα συγκεκριμένο δοθέν εύρος. Το κριτήριο του εύρους μπορεί να εφαρμοστεί σε κάθε μία

από τις τρεις διαστάσεις συντεταγμένων που χαρακτηρίζουν κάθε σημείο. Παραδείγματος χάρη, προκειμένου να εξαχθούν σημεία που απέχουν περισσότερο από 1 μέτρο από το σημείο αναφοράς, μπορεί να εφαρμοστεί το φίλτρο στον άξονα z με εύρος [0,3]. Συνήθεις χρήσεις τέτοιου φίλτρου είναι η αφαίρεση σημείων που δεν θεωρούνται σημαντικά στο νέφος με προκαθορισμένα κριτήρια ή υπολογίζοντας το εύρος δυναμικά ανάλογα με τα χαρακτηριστικά των αντικειμένων μιας σκηνής και εφαρμόζοντας το φίλτρο δυναμικά.

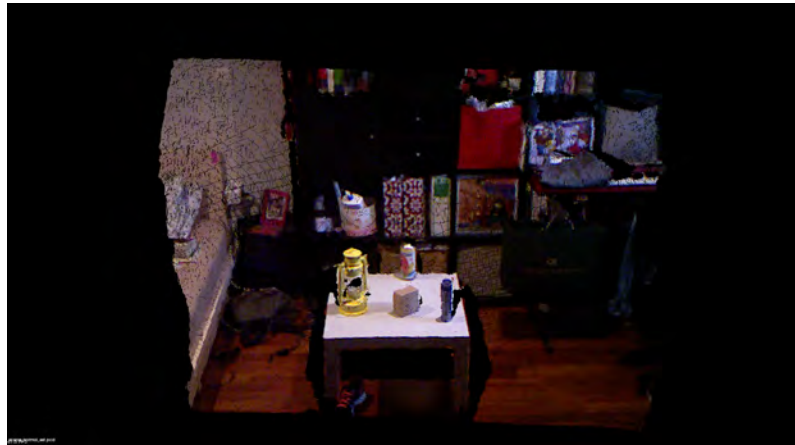
Κώδικας 4.7: Φίλτρο Pass-through

```
// Objects for storing the point clouds.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
    pcl::PointCloud<pcl::PointXYZ>::Ptr filteredCloud(new pcl::PointCloud<pcl::
    PointXYZ>);

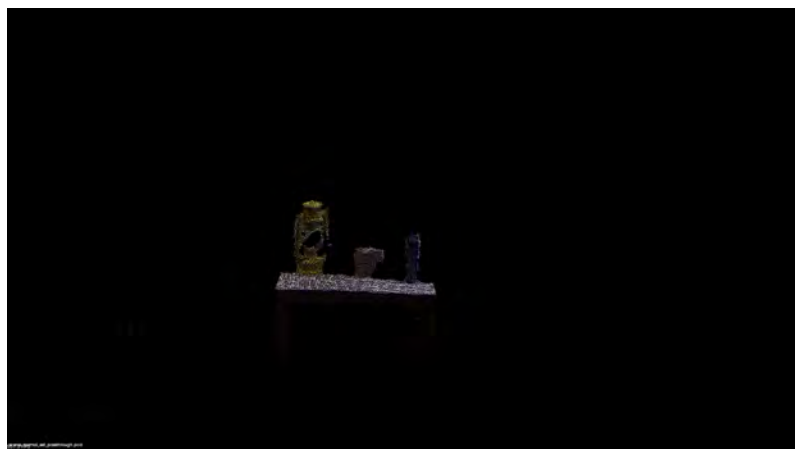
pcl::PassThrough<pcl::PointXYZ> filter;
filter.setInputCloud(cloud);

filter.setFilterFieldName("z");
filter.setFilterLimits(0.0, 3.0);

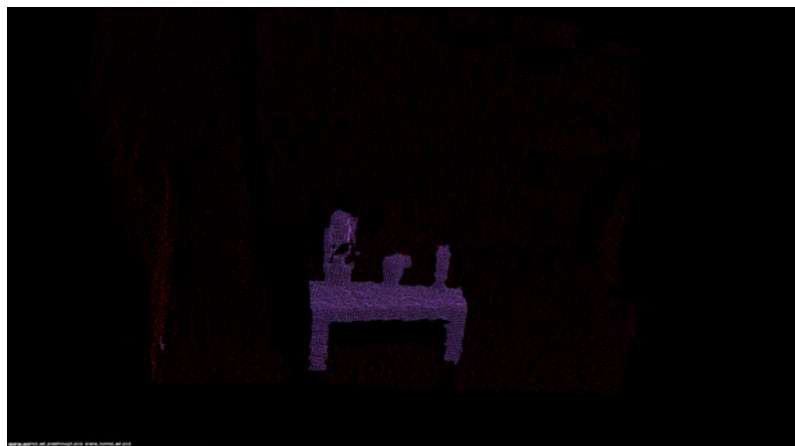
filter.filter(*filteredCloud);
```



(α') Pointcloud



(β') Φίλτρο στον άξονα z



(γ') Φίλτρο στον άξονα z



(δ') Φίλτρο στον άξονα z

4.2.11 Αφαίρεση οριακών σημείων

Τα οριακά σημεία ενός νέφους (outliers), είναι αυτά που εμφανίζονται οργανωμένα σε πολύ μικρά πλήθη και κατανομημένα μέσα στο νέφος χωρίς άμεση σύνδεση με τα πραγματικά αντικείμενα που αναπαριστώνται. Τέτοια σημεία είναι αποτέλεσμα κακών μετρήσεων και συνήθων απαιτείται να αφαιρεθούν προκειμένου οι υπολογισμοί που θα γίνουν πάνω σε ένα νέφος να μην εμφανίζουν σφάλματα, ενώ η αφαίρεση τους καθιστά και τους υπολογισμούς και τους αλγορίθμους που θα εφαρμοστούν ταχύτερους.

Αφαίρεση σε σφαιρική ακτίνα Η αφαίρεση με βάση μια σφαιρική ακτίνα είναι η πιο εύκολη μέθοδος αφαίρεσης των outliers. Η ακτίνα που ορίζεται αποτελεί ακτίνα γύρω από κάθε σημείο που θα υπολογιστούν τα γειτονικά σημεία. Με βάση ένα κατώφλι που θα οριστεί, σημεία που δεν έχουν πάνω από συγκεκριμένο αριθμό γειτονικών σημείων θα χαρακτηρίζονται οριακά και θα αφαιρούνται.

Κώδικας 4.8: Radius-based φιλτράρισμα

```
// Objects for storing the point clouds.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointXYZ>::Ptr filteredCloud(new pcl::PointCloud<pcl::
    PointXYZ>);

// Filter object.
pcl::RadiusOutlierRemoval<pcl::PointXYZ> filter;
filter.setInputCloud(cloud);

filter.setRadiusSearch(0.15);
filter.setMinNeighborsInRadius(10);

filter.filter(*filteredCloud);
```

Στατιστικός υπολογισμός outliers Η μέθοδος του στατιστικού υπολογισμού των outliers είναι μέθοδος με μεγαλύτερη πολυπλοκότητα αλλά και καλύτερης ακρίβειας. Για κάθε σημείο γίνεται υπολογισμός των K γειτόνων. Θεωρώντας ότι το αποτελεσμα ακολουθεί μία κανονική Γκαουσιανή κατανομή με μέσο μ και τυπική απόκλιση σ , τότε θεωρείται ασφαλής υπόθεση να αφαιρεθούν ως outliers όλα τα σημεία που αποκλίνουν από τη μέση απόκλιση του νέφους.

Κώδικας 4.9: Στατιστικός υπολογισμός outliers

```
// Objects for storing the point clouds.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointXYZ>::Ptr filteredCloud(new pcl::PointCloud<pcl::
    PointXYZ>);
```

```
// Filter object.
pcl::StatisticalOutlierRemoval<pcl::PointXYZ> filter;
filter.setInputCloud(cloud);

//number of neighbors
filter.setMeanK(50);
//larger than 1 standard deviation of the mean distance will be considered outliers.
filter.setStddevMulThresh(1.0);

filter.filter(*filteredCloud);
```

```
++eikona
```

4.2.12 Επαναδειγματοληψία (resampling)

Δειγματοληψία είναι χαρακτηριστικό του πλήθους των δειγμάτων που συλλέχθηκαν, στην περίπτωση ενός νέφους σημείων, επηρεάζει την πυκνότητα και το πλήθος των σημείων που συλλέχθηκαν στη μονάδα του χώρου. Με τον όρο επαναδειγματοληψία αναφερόμαστε στη διαδικασία αύξησης ή μείωσης των σημείων που αναπαριστούν ένα νέφος.

4.2.13 Υποδειγματοληψία (Downsampling)

Μία κάμερα βάθους όπως το Kinect έχοντας μία ανάλυση 640x480 σημείων παράγει νέφος σημείων με 307200 σημεία, ενώ πιο εξελιγμένες τεχνολογίες συλλέγουν πολλαπλάσια νούμερα. Ένας υπολογισμός πάνω σε κάθε σημείο του νέφους έχει πολυπλοκότητα $O(n)$, με n τα σημεία του νέφους. Παραδείγματος χάρη, η πολυπλοκότητα για τον υπολογισμό των k γειτόνων κάθε σημείο είναι ως αποτέλεσμα $O(nk)$. όπως γίνεται σαφές για απλούς υπολογισμούς σε μεγάλο πλήθος σημείων χρειάζεται μεγάλη επεξεργαστική ισχύ κάτι που δημιουργεί προβλήματα ειδικά σε εφαρμογές που χρειάζεται ανάλυση δεδομένων σε πραγματικό χρόνο.

Παρά τη μεγάλη επεξεργαστική ισχύ που απαιτείται, πολλοί αλγόριθμοι μπορούν να υλοποιηθούν σε μοντέρνα συστήματα με τρόπο που να κάνει χρήση της παραλληλίας που προσφέρουν οι γραφικοί επεξεργαστές (GPU) προσφέροντας τεράστια μείωση του χρόνου εκτέλεσης των αλγορίθμων, αλλά κάτι τέτοιο δεν είναι πάντα εφικτό.

Μία εναλλακτική προσέγγιση που προσφέρει καλύτερους χρόνους υπολογισμού είναι η μείωση της πολυπλοκότητας του νέφους άρα και των υπολογισμών που θα γίνουν σε αυτό. Τεχνικές που ήδη αναφέρονται παραπάνω όπως η αφαίρεση outliers και η εφαρμογή ενός Pass-through φίλτρου επιφέρουν τέτοιο αποτέλεσμα αλλά συνήθως

τεχνική χωρίς να θυσιαστεί το μέγεθος του χώρου που αναπαριστά το νέφος είναι η επαναδειγματοληψία που μειώνει το πλήθος των σημείων του νέφους (Downsampling). Το παραγόμενο νέφος, με εφαρμοζόμενη μία επιτρεπτή αναλογία υποδειγματοληψίας, παρόλο που απαρτίζεται από λιγότερα σημεία, αναπαριστά το αρχικό και αποτελεί αξιόπιστη είσοδο για τους αλγορίθμους υπολογισμού.

Η διαδικασία της υποδειγματοληψίας γίνεται με διαίρεση του νέφους σε voxels όπως και στην μέθοδο octtree. Το νέφος χβαντοποιείται σε πολλές εφαιπόμενες και ορθογραμμισμένες περιοχές με κυβικό σχήμα, που το πλήθος τους εξαρτάται ανάλογα από την επιθυμητή τελική ανάλυση. Στη συνέχεια όλα τα σημεία του κάθε Voxel επεξεργάζονται έτσι ώστε να παραμείνει μόνο ένα. Η πιο απλή προσέγγιση είναι να επιλεγεί τυχαία ένα από τα σημεία του κάθε voxel αλλά μπορεί να χρησιμοποιηθούν πολύ πιο ακριβείς μέθοδοι όπως η επιλογή του σημείου που αποτελεί το κεντρικού σημείου ενός νέφους (centroid).

Κώδικας 4.10: Downsampling

```
// Objects for storing the point clouds.
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);
pcl::PointCloud<pcl::PointXYZ>::Ptr filteredCloud(new pcl::PointCloud<pcl::
    PointXYZ>);

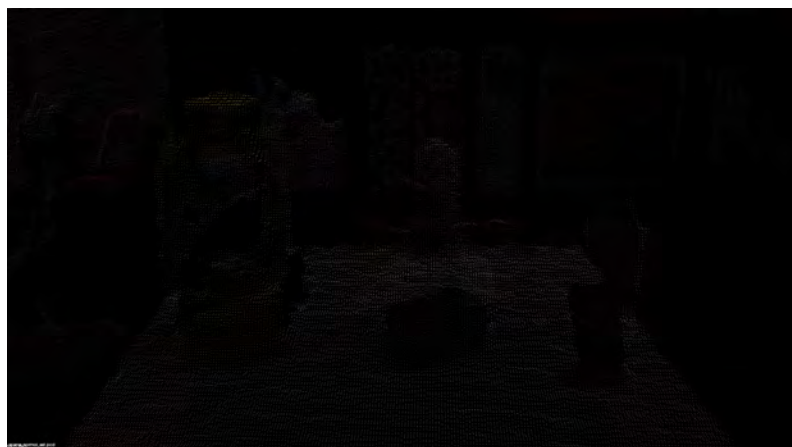
// Filter object.
pcl::VoxelGrid<pcl::PointXYZ> filter;
filter.setInputCloud(cloud);

//size of the voxel set to 1cm
filter.setLeafSize(0.01f, 0.01f, 0.01f);

filter.filter(*filteredCloud);
```



(α') Αρχικό Pointcloud

(β') Μετά από υποδειγματοληψία
Σχήμα 4.6: Downsampling

4.2.14 Υπερδειγματοληψία (Upsampling)

Η μέθοδος της υπερδειγματοληψίας είναι η αντίθετη διαδικασία από αυτή του downsampling με την διαφορά ότι προστίθεται πληροφορία στο νέφος. Η ειδικότερη διαφορά εδώ είναι πως αυτό γίνεται χωρίς την ύπαρξη περισσότερης πληροφορίας για την σκηνή που αναπαριστάται στο νέφος, άρα αποτελεί μία μέθοδο ανασύστασης του νέφους και κυρίως των επιφανειών που αυτό αναπαριστά παρεμβάλλοντας επιπλέον σημεία ανάμεσα στα υπάρχοντα.

Κώδικας 4.11: Upsampling

```
// Objects for storing the point clouds.  
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);  
pcl::PointCloud<pcl::PointXYZ>::Ptr filteredCloud(new pcl::PointCloud<pcl::  
    PointXYZ>);
```

```
// Filter object.
pcl::MovingLeastSquares<pcl::PointXYZ, pcl::PointXYZ> filter;
filter.setInputCloud(cloud);

pcl::search::KdTree<pcl::PointXYZ>::Ptr kdtree; filter.setSearchMethod(kdtree);

//radius of 3cm
filter.setSearchRadius(0.03);

filter.setUpsamplingMethod(pcl::MovingLeastSquares<pcl::PointXYZ, pcl::PointXYZ>::SAMPLE_LOCAL_PLANE);
filter.setUpsamplingRadius(0.03);
filter.setUpsamplingStepSize(0.02);

filter.filter(*filteredCloud);
```




(α') Αρχικό Pointcloud



(β') Μετά από υπερδειγματοληψία

Σχήμα 4.7: Upsampling

4.2.15 Plane Segmentation

Ο αλγόριθμος όπως αναλύθηκε στο προηγούμενο κεφάλαιο, βασίζεται στη μέθοδο RANSAC που αποτελεί μία επαναληπτική μέθοδος υπολογισμού παραμέτρων ενός μοντέλου από ένα σύνολο τυχαία επιλεγμένων δεδομένων.

Κώδικας 4.12: Plane Segmentation

```
// Objects for storing the point clouds.  
pcl::PointCloud<pcl::PointXYZ>::Ptr cloud(new pcl::PointCloud<pcl::PointXYZ>);  
pcl::PointCloud<pcl::PointXYZ>::Ptr plane(new pcl::PointCloud<pcl::PointXYZ>);  
pcl::PointCloud<pcl::PointXYZ>::Ptr convexHull(new pcl::PointCloud<pcl::  
    PointXYZ>);
```

```
pcl::PointCloud<pcl::PointXYZ>::Ptr objects(new pcl::PointCloud<pcl::PointXYZ>);

// find the plane model
pcl::ModelCoefficients::Ptr coefficients(new pcl::ModelCoefficients);
pcl::SACSegmentation<pcl::PointXYZ> segmentation;
segmentation.setInputCloud(cloud);
segmentation.setModelType(pcl::SACMODEL_PLANE);
segmentation.setMethodType(pcl::SAC_RANSAC);
segmentation.setDistanceThreshold(0.01);
segmentation.setOptimizeCoefficients(true);
pcl::PointIndices::Ptr planeIndices(new pcl::PointIndices);
segmentation.segment(*planeIndices, *coefficients);

// Copy the points of the plane to a new cloud.
pcl::ExtractIndices<pcl::PointXYZ> extract;
extract.setInputCloud(cloud);
extract.setIndices(planeIndices);
extract.filter(*plane);

// Retrieve the convex hull.
pcl::ConvexHull<pcl::PointXYZ> hull;
hull.setInputCloud(plane);
hull.setDimension(2);
hull.reconstruct(*convexHull);

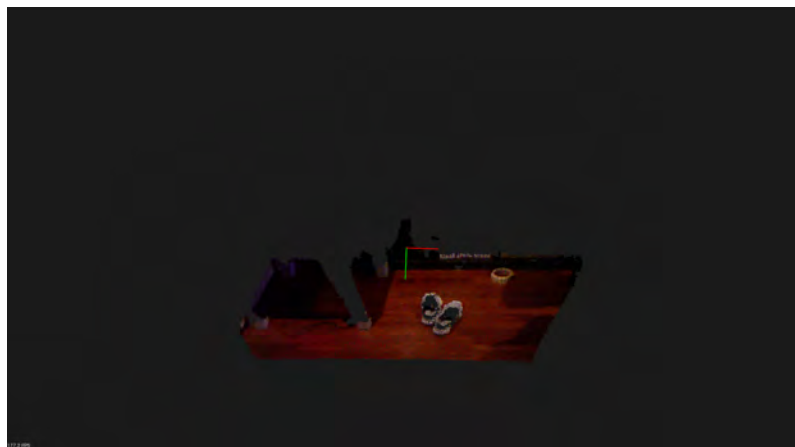
// Prism object.
pcl::ExtractPolygonalPrismData<pcl::PointXYZ> prism;
prism.setInputCloud(cloud);
prism.setInputPlanarHull(convexHull);

prism.setHeightLimits(0.0f, 0.1f);
pcl::PointIndices::Ptr objectIndices(new pcl::PointIndices);
prism.segment(*objectIndices);

// Get and show all points retrieved by the hull.
extract.setIndices(objectIndices);
extract.filter(*objects);
pcl::visualization::CloudViewer viewerObjects("Objects on table");
viewerObjects.showCloud(objects);
while (!viewerObjects.wasStopped()) {}
```



(α') Αρχικό Pointcloud



(β') Μετά από το φιλτράρισμα
Σχήμα 4.8: Plane Segmentation

Κεφάλαιο 5

Αποτελέσματα - Σύγκριση Αποτελεσμάτων

Παρακάτω φαίνονται αποτελέσματα από την εκτέλεση του κώδικα:

5.1 Εκπαίδευση συστήματος

5.1.1 Βιβλιοθήκη αντικειμένων

Τα 3D μοντέλα που εισάγονται στο σύστημα ως είσοδος για τη διαδικασία της εκπαίδευσης, αποτελούν τη βάση δεδομένων στην οποία θα προσπαθήσει το σύστημα να αναζητήσει και να αναγνωρίσει το αντικείμενο που βλέπει η κάμερα.

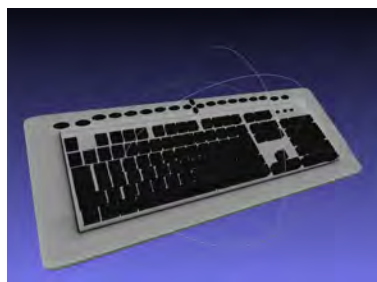
5.1.2 Εξαγωγή σκηνών από διάφορες οπτικές γωνίες

Από κάθε μοντέλο αντικειμένου που δίνεται στο σύστημα για εκπαίδευση, γίνεται προβολή του σε τρισδιάστατο χώρο με σταθερή εικονική κάμερα. Το αντικείμενο περιστρέφεται γύρω από το κέντρο του σε διάφορες γωνίες και εκτελούνται τα εξής βήματα:

- Εικονική σάρωση από την κάμερα
- Μετατροπή σε Pointcloud
- Εξαγωγή χαρακτηριστικών



(α') Ball



(β') Keyboard



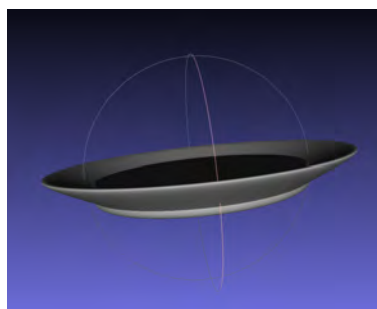
(γ') Lantern



(δ') Milk Box



(ε') Monitor



(ς') Plate



(ζ') Pliers



(η') Shoe



(θ') Soda

Σχήμα 5.1: Βιβλιοθήκη αντικειμένων



(α') Ball 3D Model



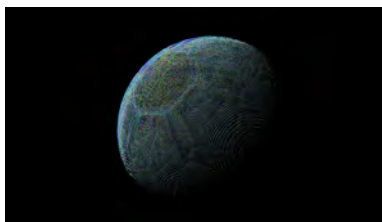
(β') Lantern 3D Model



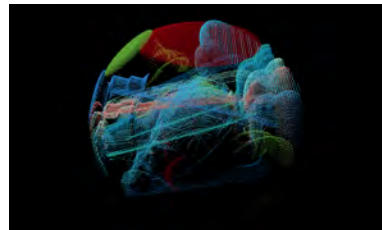
(γ') Ball 1st partial view



(δ') Lantern 1st partial view



(ε') Ball All partial views

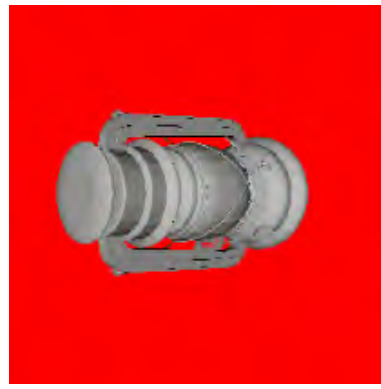


(ϛ') Lantern All partial views

Σχήμα 5.2: Εξαγωγή σχημών



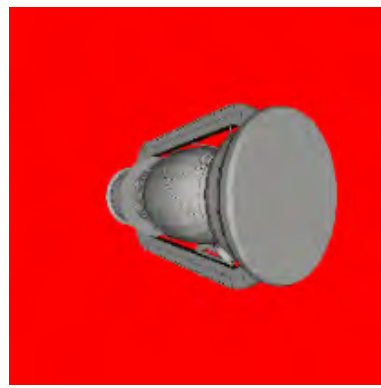
(α') Σκηνή με Φανάρι



(β') Σκηνή με Φανάρι



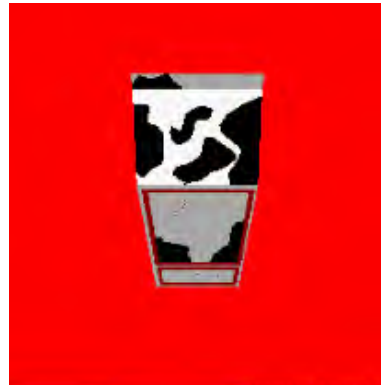
(γ') Σκηνή με Φανάρι



(δ') Σκηνή με Φανάρι



(ε') Σκηνή με κουτί



(ς') Σκηνή με κουτί



(ζ') Σκηνή με κουτί



(η') Σκηνή με κουτί

Σχήμα 5.3: Δημιουργία εικονικών σκημών από τα 3Δ αντικείμενα

5.1.3 Χώρος σάρωσης



(α') Σκηνή



(β') Επιφάνεια τοποθέτησης αντικειμένων
Σχήμα 5.4: Χώρος σάρωσης



(α) Σκηνή - Κάτοψη

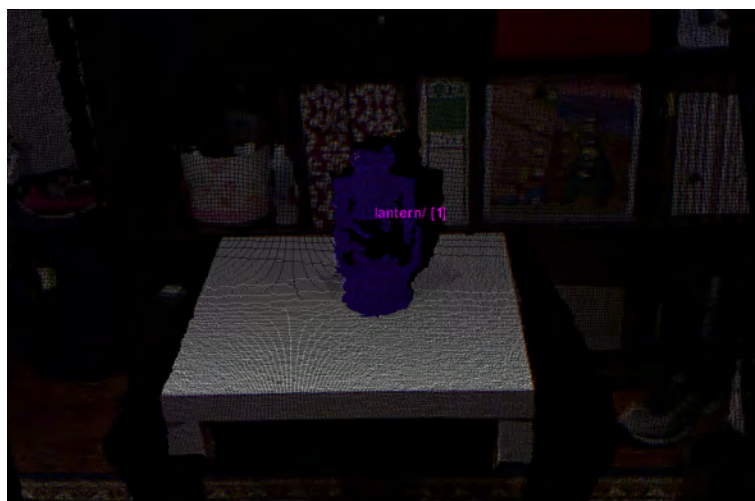


(β) Σκηνή 2

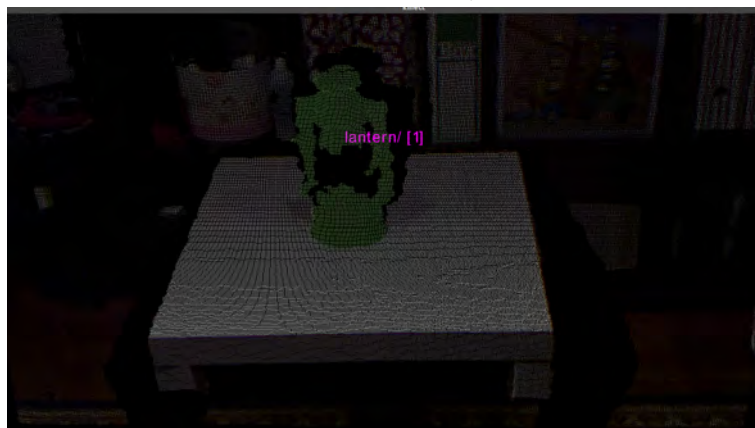
Σχήμα 5.5: Χώρος σάρωσης (2)

5.1.4 Αποτελέσματα αναγνώρισης

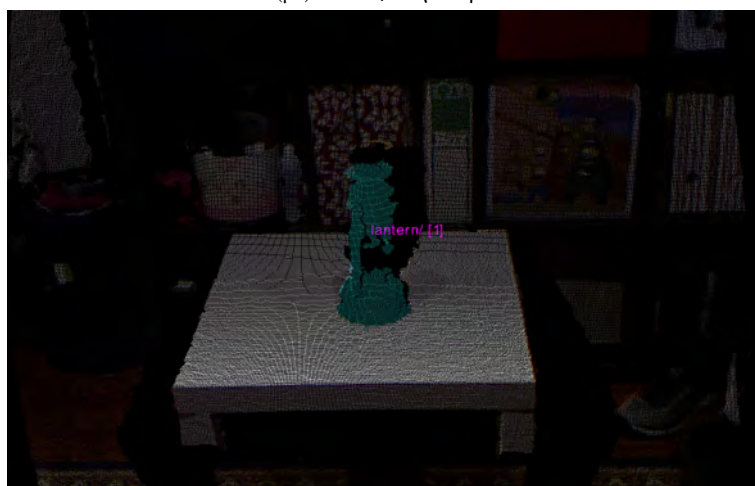
Σκηνή 1 - Ένα αντικείμενο στην επιφάνεια



(α') Αναγνώριση 1



(β') Αναγνώριση 2



(γ') Αναγνώριση 3

Σχήμα 5.6: Το φανάρι αναγνωρίζεται από 3 διαφορετικές στάσεις

Σκηνή 2 - Δύο αντικείμενα στην επιφάνεια



(α') Αναγνώριση 1



(β') Αναγνώριση 2



(γ') Αναγνώριση 3

Σχήμα 5.7: Το φανάρι αναγνωρίζεται από διαφορετικές στάσεις ακόμη και σε επικάλυψη στον ζ άξονα

Σχηνή 3 - Τρία αντικείμενα στην επιφάνεια



(α') Αναγνώριση 1



(β') Αναγνώριση 2



(γ') Αναγνώριση 3

Σχήμα 5.8: Τρία αντικείμενα στην επιφάνεια

Τα διαφορετικά αντικείμενα απομονώνονται σε ξεχωριστά Ποιντζλουντ και αναγνωρίζονται αυτόνομα από το δέντρο.

Σκηνή 4 - Ένα μικρό αντικείμενο στην επιφάνεια



(α') Αναγνώριση 1



(β') Αναγνώριση 2



(γ') Αναγνώριση 3

Σχήμα 5.9: Αναγνώριση μικρού αντικειμένου σε διαφορετικές στάσεις

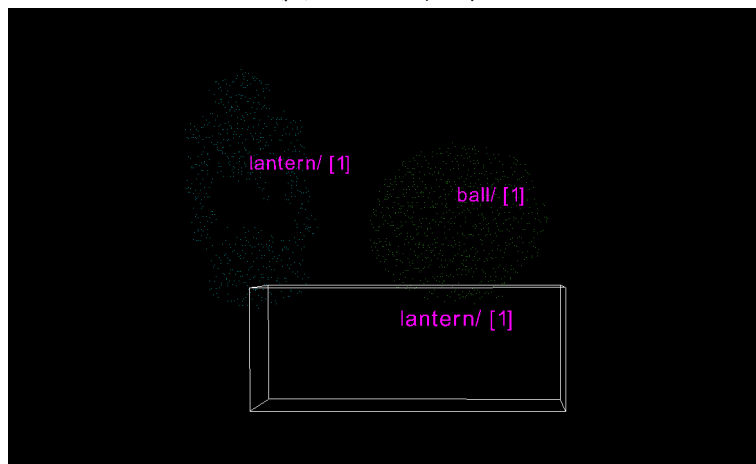
Σκηνή 5 - Σφάλμα αναγνώρισης με τρία αντικείμενα στην επιφάνεια



(α') Αναγνώριση 1



(β') Αναγνώριση 2



(γ') Αναγνώριση 3

Σχήμα 5.10: Σφάλμα αναγνώρισης

Σφάλμα αναγνώρισης: Το αντικείμενο (υπόδημα) διαφέρει στα χαρακτηριστικά από το πρότυπο και αναγνωρίζεται λανθασμένα ως φανάρι.

Σκηνή 6 - Δύο όμοια αντικείμενα στην επιφάνεια



(α') Αναγνώριση 1

Σχήμα 5.11: Τα δύο όμοια αντικείμενα αναγνωρίζονται ξεχωριστά ως φανάρια

Σκηνή 7 - Δύο αντικείμενα στην επιφάνεια με διαφορετική τοποθέτηση της κάμερας



(α') Αναγνώριση 1



(β') Αναγνώριση 2

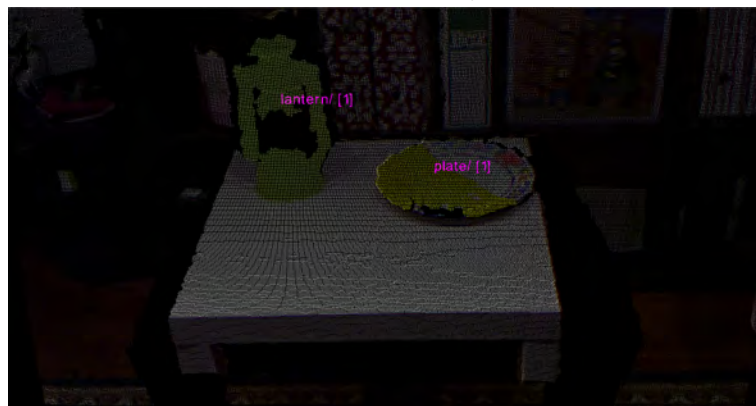
Σχήμα 5.12: Δύο αντικείμενα στην επιφάνεια

Τα αντικείμενα αναγνωρίζονται επιτυχώς και για τις δύο τοποθετήσεις της κάμερας.

Σκηνή 8 - Δύο αντικείμενα στην επιφάνεια με συμμετρία



(α') Αναγνώριση 1

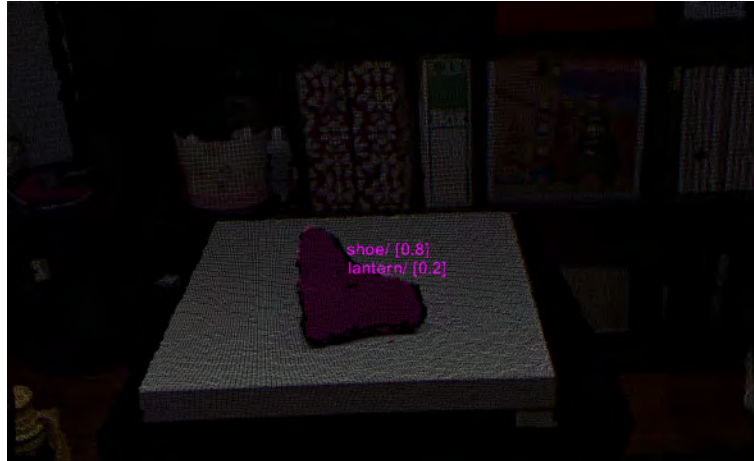


(β') Αναγνώριση 2

Σχήμα 5.13: Δύο αντικείμενα στην επιφάνεια με συμμετρία

Το αντικείμενα (πιάτο) αναγνωρίζεται επιτυχώς αλλά η στάση του δεν θα μπορούσε να καθοριστεί καθώς είναι συμμετρικό. Το χρώμα και το pattern του χρώματος δεν λαμβάνεται ως feature.

Σκηνή 9 - Αντικείμενα που δεν υπάρχουν στη βάση των αντικειμένων



(α') Αναγνώριση 1



(β') Αναγνώριση 2

Σχήμα 5.14: Αντικείμενα που δεν υπάρχουν στη βάση

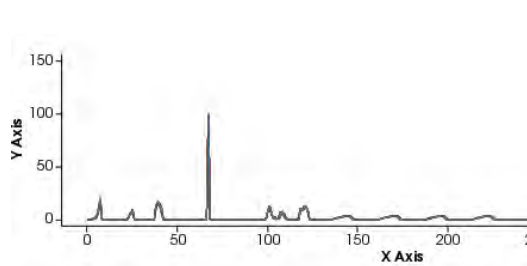
Αντικείμενα που δεν υπάρχουν στη βάση, αναγνωρίζονται λανθασμένα στο πιο κοντινό σε χαρακτηριστικά αντικείμενο.

Εκτέλεση κώδικα - Logs

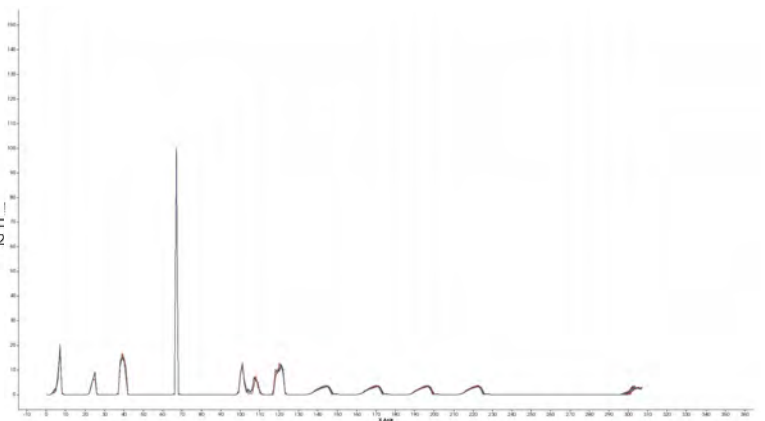
5.1.5 Διάγραμμα Χαρακτηριστικών



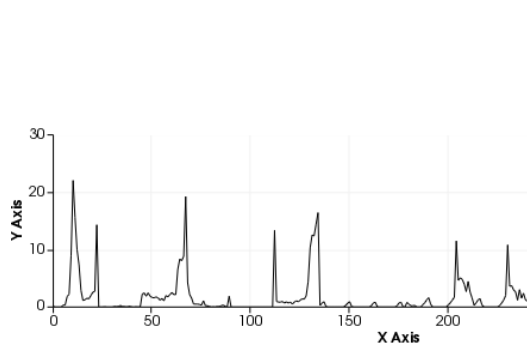
(α') Σκηνή



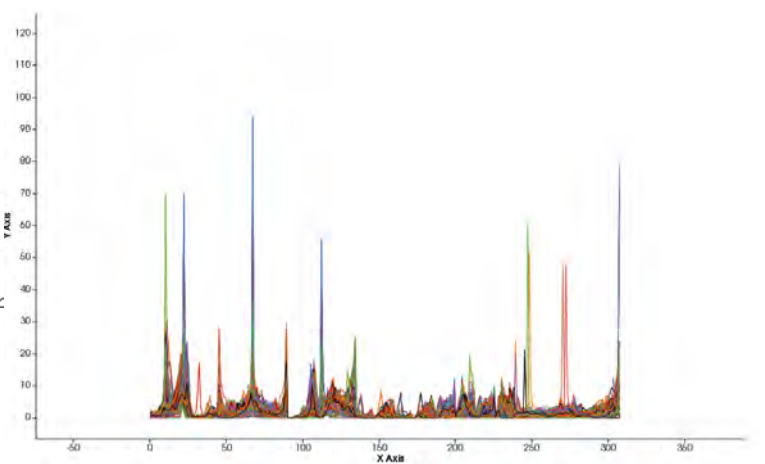
(β') Διάγραμμα Χαρακτηριστικών - Μπάλα



(γ') Διαγράμματα Αναφοράς



(δ') Διάγραμμα Χαρακτηριστικών - Φανάρι



(ε') Διαγράμματα Αναφοράς

Σχήμα 5.16: Διάγραμμα Χαρακτηριστικών

Το φανάρι αναγνωρίζεται ανάμεσα στις διαφορετικές πόζες που έχουν συλλεχθεί ως αναφορά κατά την εκπαίδευση του συστήματος

Κεφάλαιο 6

Επίλογος

Όπως φαίνεται από τα αποτελέσματα που παρουσιάστηκαν στο προηγούμενο κεφάλαιο, η αναγνώριση αντικειμένων είναι αρκετά ακριβής και όχι τόσο υπολογιστικά ακριβή, ώστε να μπορεί να πραγματοποιηθεί ακόμα και σε πραγματικό χρόνο. Παρόλο που οι δοκιμές έγιναν σε σχεδόν ιδανικό περιβάλλον η εφαρμογή δεν είναι επιρρεπής στο θόρυβο χάρις στο αρχικό φιλτράρισμα και μπορεί εύκολα να επεκταθεί σε μεγαλύτερους ή μικρότερους χώρους.

Επίσης η εφαρμογή είναι επεκτάσιμη και δίνει την δυνατότητα για προσθήκη περισσότερων αντικειμένων που μπορούν να αναγνωριστούν χωρίς μεγάλη επίπτωση στην ταχύτητα αναγνώρισης.

6.1 Επεκτάσεις

Η αναγνώριση αντικειμένων με πολλές εφαρμογές στη βιομηχανία, την επιστήμη, τις mobile εφαρμογές βρίσκεται ήδη σε πληθώρα δραστηριοτήτων της καθημερινότητας. Έτσι και η 3D αναγνώριση αποκτά τη θέση της με τεχνολογίες που γίνονται όλο και πιο διαδεδομένες, augmented reality, 3D printing,

Στους αλγόριθμους που αναπτύχθηκαν δεν συμπεριλήφθηκε το χρώμα ως χαρακτηριστικό που περιγράφει το κάθε αντικείμενο ξεχωριστά. Αυτό θα μπορούσε να δώσει επιπλέον δυνατότητες αναγνώρισης ίδιων γεωμετρικά αντικειμένων αλλά με διαφορετική απόχρωση.

6.2 Περιορισμοί

Οι αλγόριθμοι που αναπτύχθηκαν στα πλαίσια αυτής της διπλωματικής εργασίας μπορούν να επεκταθούν και σε αντικείμενα μεγαλύτερης κλίμακας ή μικρότερης με περισσότερες λεπτομέρειες. Όμως οι μικρο-ρυθμίσεις στον κώδικα αλλά και οι 3D

κάμερες που χρησιμοποιούνται μπορούν να αποτυπώσουν με ακρίβεια ώστε να μην χαθεί η λειτουργία του αλγορίθμου, αντικείμενα καθημερινής χρήσης και μεσαίου μεγέθους.

6.3 Συμπεράσματα

Ο τομέας της αναγνώρισης αντικειμένων με χρήση της τρισδιάστατης αναπαράστασης βελτιώνει και διευκολύνει σημαντικά την αναγνώριση και κατάτμηση όπως αυτή γινόταν από στατικές εικόνες ή βίντεο. Εφαρμογές τέτοιων τεχνολογιών μπορούν να χρησιμοποιηθούν στη παραγωγή & αποθήκευση προϊόντων και στη διεπαφή ανθρώπου με υπολογιστή, όπως επίσης και στη ψυχαγωγία.

Συγκεκριμένα για την υλοποίηση αυτής της εργασίας από τα αποτελέσματα γίνεται σαφές πως η εκπαίδευση με αρχεία 3D είναι εφικτή και μπορεί να χρησιμοποιηθεί με επιτυχία στην αναγνώριση των πραγματικών αντικειμένων βασιζόμενοι στα τρισδιάστατα μοντέλα που τα περιγράφουν. Τις περισσότερες φορές το τρισδιάστατο μοντέλο δεν χρειάζεται να δημιουργηθεί από σάρωση του πραγματικού αντικειμένου αλλά μπορεί ταχύτερα να δημιουργηθεί σε ειδικό λογισμικό ως συνδυασμός θεμελιωδών γεωμετρικών σχημάτων.

Από τα αποτελέσματα της εργασίας γίνεται σαφές πως η αναγνώριση χαρακτηριστικών με global περιγραφείς αλλά και η χρήση νέφους σημείων, δίνει πολλές δυνατότητες για αναγνώριση του αντικειμένου ακόμα και σε παραμορφωμένη ή ελλειπής μορφή εισόδου. Η αναγνώριση παρουσίασε μεγάλη ανοχή στο θόρυβο, είτε από το περιβάλλον και το φωτισμό, είτε από τα υπόλοιπα γειτονικά αντικείμενα.

Βιβλιογραφία

- [1] Aitor Aldoma, Federico Tombari, Radu Bogdan Rusu και Markus Vincze. Our-cvfh-oriented, unique and repeatable clustered viewpoint feature histogram for object recognition and 6dof pose estimation. Στο *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*, σελίδες 113–122. Springer, 2012.
- [2] Aitor Aldoma, Markus Vincze, Nico Blodow, David Gossow, Suat Gedikli, Radu Bogdan Rusu και Gary Bradski. Cad-model recognition and 6dof pose estimation using 3d cues. Στο *2011 IEEE international conference on computer vision workshops (ICCV workshops)*, σελίδες 585–592. IEEE, 2011.
- [3] Marc Alexa, Johannes Behr, Daniel Cohen-Or, Shachar Fleishman, David Levin και Claudio T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on visualization and computer graphics*, 9(1):3–15, 2003.
- [4] Luis A Alexandre. 3d descriptors for object and category recognition: a comparative evaluation. Στο *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*, τόμος 1, σελίδα 7, 2012.
- [5] Khaled Alhamzi, Mohammed Elmogy και Sherif Barakat. 3d object recognition based on local and global features using point cloud library. *International Journal of Advancements in Computing Technology*, 7(3):43, 2015.
- [6] Haider Ali, Faisal Shafait, Eirini Giannakidou, Athena Vakali, Nadia Figueroa, Theodoros Varvadoukas και Nikolaos Mavridis. Contextual object category recognition for rgb-d scene labeling. *Robotics and Autonomous Systems*, 62(2):241–256, 2014.
- [7] David G Lowe. Local feature view clustering for 3d object recognition. Στο *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, τόμος 1, σελίδες I–I. IEEE, 2001.

-
- [8] Zoltan Csaba Marton, Dejan Pangercic, Nico Blodow και Michael Beetz. Combined 2d–3d categorization and classification for multimodal perception systems. *The International Journal of Robotics Research*, 30(11):1378–1402, 2011.
- [9] Ajmal Mian, Mohammed Bennamoun και Robyn Owens. On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes. *International Journal of Computer Vision*, 89(2-3):348–361, 2010.
- [10] Chavdar Papazov και Darius Burschka. An efficient ransac for 3d object recognition in noisy and occluded scenes. Στο *Asian Conference on Computer Vision*, σελίδες 135–148. Springer, 2010.
- [11] Xiaofeng Ren, Dieter Fox και Kurt Konolige. Change their perception: Rgb-d for 3-d modeling and recognition. *IEEE Robotics & Automation Magazine*, 20(4):49–59, 2013.
- [12] Radu Bogdan Rusu, Gary Bradski, Romain Thibaux και John Hsu. Fast 3d recognition and pose using the viewpoint feature histogram. Στο *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, σελίδες 2155–2162. IEEE, 2010.
- [13] Santosh Thoduka, Stepan Pazekha, Alexander Moriarty και Gerhard K Kraetzschmar. Rgb-d-based features for recognition of textureless objects. Στο *Robot World Cup*, σελίδες 294–305. Springer, 2016.
- [14] Federico Tombari, Samuele Salti και Luigi Di Stefano. Unique signatures of histograms for local surface description. Στο *European conference on computer vision*, σελίδες 356–369. Springer, 2010.
- [15] Walter Wohlking, Aitor Aldoma, Radu B Rusu και Markus Vincze. 3dnet: Large-scale object class recognition from cad models. Στο *2012 IEEE international conference on robotics and automation*, σελίδες 5384–5391. IEEE, 2012.
- [16] Walter Wohlking και Markus Vincze. Ensemble of shape functions for 3d object classification. Στο *2011 IEEE international conference on robotics and biomimetics*, σελίδες 2987–2992. IEEE, 2011.

