



Σχολή Θετικών Επιστημών & Τεχνολογίας
ΠΛΗΡΟΦΟΡΙΚΗ

Πτυχιακή Εργασία

**Βελτιστοποίηση τοποθέτησης αισθητήρων LiDAR για την
αποτύπωση 3D περιβαλλόντων**

Δημήτρης Ζαρολιάγκης

Επιβλέπων καθηγητής: Χρήστος-Νικόλαος Αναγνωστόπουλος

Πάτρα, Μάιος 2026

Η παρούσα εργασία αποτελεί πνευματική ιδιοκτησία του φοιτητή («συγγραφέας/δημιουργός») που την εκπόνησε. Στο πλαίσιο της πολιτικής ανοικτής πρόσβασης ο συγγραφέας/δημιουργός εκχωρεί στο ΕΑΠ, μη αποκλειστική άδεια χρήσης του δικαιώματος αναπαραγωγής, προσαρμογής, δημόσιου δανεισμού, παρουσίασης στο κοινό και ψηφιακής διάχυσής τους διεθνώς, σε ηλεκτρονική μορφή και σε οποιοδήποτε μέσο, για διδακτικούς και ερευνητικούς σκοπούς, άνευ ανταλλάγματος και για όλο το χρόνο διάρκειας των δικαιωμάτων πνευματικής ιδιοκτησίας. Η ανοικτή πρόσβαση στο πλήρες κείμενο για μελέτη και ανάγνωση δεν σημαίνει καθ' οιονδήποτε τρόπο παραχώρηση δικαιωμάτων διανοητικής ιδιοκτησίας του συγγραφέα/δημιουργού ούτε επιτρέπει την αναπαραγωγή, αναδημοσίευση, αντιγραφή, αποθήκευση, πώληση, εμπορική χρήση, μετάδοση, διανομή, έκδοση, εκτέλεση, «μεταφόρτωση» (downloading), «ανάρτηση» (uploading), μετάφραση, τροποποίηση με οποιονδήποτε τρόπο, τμηματικά ή περιληπτικά της εργασίας, χωρίς τη ρητή προηγούμενη έγγραφη συναίνεση του συγγραφέα/δημιουργού. Ο συγγραφέας/δημιουργός διατηρεί το σύνολο των ηθικών και περιουσιακών του δικαιωμάτων.



Βελτιστοποίηση τοποθέτησης αισθητήρων LiDAR για την αποτύπωση 3Δ περιβαλλόντων

Δημήτρης Ζαρολιάγκης

Επιβλέπων Καθηγητής:
Χρήστος-Νικόλαος Αναγνωστόπουλος
Καθηγητής, Πανεπιστήμιο Αιγαίου

Πάτρα, Μάιος 2026

Ευχαριστίες

Η παρούσα πτυχιακή εργασία δεν θα είχε ολοκληρωθεί χωρίς την συνεχή καθοδήγηση και ενθάρρυνση του επιβλέποντα Καθηγητή μου Δρ Χρήστου-Νικόλαου Αναγνωστόπουλου, ο οποίος ενέπνευσε το ενδιαφέρον μου στο θέμα και μοιράστηκε μαζί μου, γενναιόδωρα, τις ιδέες του. Τον ευχαριστώ θερμά, όχι μόνο για την πολύτιμη επίβλεψή του, αλλά και γιατί μου άφησε μεγάλα περιθώρια ελευθερίας, ώστε να πάρω πρωτοβουλίες και να πειραματιστώ με τις δικές μου ιδέες.

Καθώς με την εργασία αυτή ολοκληρώνονται οι σπουδές μου στο πρόγραμμα Πληροφορικής του Ελληνικού Ανοικτού Πανεπιστημίου, ευχαριστώ όλους τους διδάσκοντες για όσα μου πρόσφεραν στη διάρκεια των σπουδών μου.

Τέλος, οφείλω ένα ευχαριστώ στην οικογένεια και τους φίλους μου για την υποστήριξή τους τα τελευταία τέσσερα χρόνια.

Περίληψη

Η παρούσα Πτυχιακή Εργασία έχει ως σημείο αφετηρίας το πρόβλημα της «γκαλερί έργων τέχνης» (Art Gallery Problem, AGP), το οποίο διατυπώθηκε για πρώτη φορά από τον Victor Klee το 1973. Αναπτύχθηκε η εφαρμογή Scannalyze, προκειμένου να βελτιστοποιηθεί η τοποθέτηση αισθητήρων LiDAR για την αποτύπωση τρισδιάστατων χώρων. Η εφαρμογή επιτρέπει την σαφή και οπτικοποιημένη απεικόνιση του ζητήματος της ορατότητας, την αξιολόγηση και σύγκριση των μεθόδων κάλυψης σε διδιάστατους και τρισδιάστατους χώρους όσον αφορά την αποτελεσματικότητα και την υπολογιστική πολυπλοκότητά τους, και τον πειραματισμό με διάφορες γεωμετρίες και παραμέτρους.

Η εργασία συνοψίζει τις κυριότερες προσεγγίσεις για το πρόβλημα της κάλυψης διδιάστατων και τρισδιάστατων χώρων, και επισημαίνει τα σημεία που καθιστούν πιο προκλητική υπολογιστικά την επεξεργασία των τρισδιάστατων χώρων. Παρουσιάζει αναλυτικά αποτελέσματα από πειράματα που διεξήχθησαν και στις δύο διαστάσεις και προτείνει κατευθύνσεις για μελλοντική έρευνα.

Λέξεις – Κλειδιά

Αποτύπωση 3Δ χώρων, Βελτιστοποίηση, Νέφος σημείων, LiDAR, Next-Best-View, Point Cloud Based Coverage, Ray Tracing.

Abstract

This project takes as its starting point the "Art Gallery Problem" (AGP), which was first formulated by Victor Klee in 1973. The "Scannalyze" application was developed to optimize the placement of LiDAR sensors for the 3D mapping of spaces. The application allows for a clear and visual representation of the visibility issue, the evaluation and comparison of coverage methods in 2D and 3D spaces, regarding both their efficiency and computational complexity, and experimentation with various geometries and parameters.

The thesis summarizes the main approaches to the problem of covering 2D and 3D spaces and highlights the factors that make the processing of 3D spaces more challenging computationally. It presents detailed results from experiments conducted in both dimensions and proposes directions for future research.

Keywords

3D Mapping of Spaces, LiDAR, Next-Best-View, Optimization, Point Cloud, Point Cloud Based Coverage, Ray Tracing.

Περιεχόμενα

Ευχαριστίες	iv
Περίληψη.....	v
Abstract	vi
Περιεχόμενα	vii
Κατάλογος Εικόνων / Σχημάτων	ix
Κατάλογος Πινάκων	x
Συντομογραφίες & Ακρωνύμια.....	xi
1. Εισαγωγή.....	1
1.1 Κύριες Προσεγγίσεις στο 2Δ Πρόβλημα	2
1.1.1 Θεωρητικές ή συνδυαστικές προσεγγίσεις	2
1.1.2 Αλγοριθμικές ή υπολογιστικές προσεγγίσεις	2
1.1.3 Γεωμετρικές προσεγγίσεις	3
1.1.4 Προσεγγίσεις βελτιστοποίησης και μαθηματικού προγραμματισμού	3
1.1.5 Ευρετικές και μεταερευτικές προσεγγίσεις.....	3
1.1.6 Εφαρμοσμένες προσεγγίσεις.....	3
1.2 Κύριες προσεγγίσεις στο 3Δ Πρόβλημα.....	4
1.2.1 Πολυεδρικές και Γεωμετρικές Προσεγγίσεις.....	4
1.2.2 Προσεγγίσεις Ογκοστοιχείων	4
1.2.3 Προσεγγίσεις Νέφους Σημείων	4
1.3 Προτεινόμενη Μεθοδολογία Επίλυσης του Προβλήματος	5
1.4 Οργάνωση του τόμου.....	6
2 Θεωρητικό Υπόβαθρο.....	8
2.1 Μαθηματική Διατύπωση του Προβλήματος 2Δ	8
2.2 Προσεγγίσεις	9
2.3 Αλγόριθμος Ray Tracing	12
2.4 Αλγόριθμος Next-Best-View	15
2.5 Μαθηματική Διατύπωση του Προβλήματος 3Δ	19
2.6 Αλγόριθμος Point Cloud Based Coverage	19
3 Ανάλυση Απαιτήσεων και Σχεδιασμός Εφαρμογής	22
4 Υλοποίηση.....	27
4.1 Γενική Υλοποίηση	27
4.2 Επεξεργασία 2Δ Χώρων.....	27
4.3 Παραδειγματική Εκτέλεση για 2Δ χώρους.....	28
4.4 Επεξεργασία 3Δ χώρων	43
4.4.1 Η Κατάρρευση της Τριγωνοποίησης και το Πολύεδρο του Schönhardt	43
4.4.2 Πολυπλοκότητα Ορατότητας και Ελικοειδείς Επιφάνειες	43
4.4.3 Υπολογιστική Πολυπλοκότητα και NP-Hardness.....	44
4.4.4 Αξιολόγηση Συνεισφοράς Θέσεων Φύλαξης μέσω Τιμής Shapley	45
4.4.5 Επισημάνσεις για την υλοποίηση	47
4.5 Παραδειγματική Εκτέλεση 3Δ χώρων	49
5 Αποτελέσματα.....	53
5.1 Αποτελέσματα για 2Δ χώρους	53
5.2 Αποτελέσματα για 3Δ χώρους	58

5.3	Πρόσθετα Πειράματα για 3Δ χώρους	61
5.3.1	Πείραμα με στεγανοποιημένο μοντέλο με ανοιχτά παράθυρα και πόρτες	61
5.3.2	Πείραμα με πραγματικό χώρο με αρχείο .xyz	62
6	Συμπεράσματα και Μελλοντικές Επεκτάσεις	66
7	Βιβλιογραφία.....	71
	Παράρτημα Α: Κώδικας Πτυχιακής Εργασίας	76

Κατάλογος Εικόνων / Σχημάτων

Εικόνα 1: Παράδειγμα Ray Tracing χωρίς εσωτερικά εμπόδια	14
Εικόνα 2: Παράδειγμα Ray Tracing με εμπόδιο μια κολώνα	14
Εικόνα 3: Παράδειγμα Next-Best-View	17
Εικόνα 4: Παράδειγμα Next-Best-View με εμπόδιο	17
Εικόνα 5: Παράδειγμα Next-Best-View αποτέλεσμα	18
Εικόνα 6: Διάγραμμα Περιπτώσεων Χρήσης Scannalyze	22
Εικόνα 7: Αρχική Οθόνη εφαρμογής Scannalyze.....	28
Εικόνα 8: Εισαγωγή Χώρου 2Δ (α)	29
Εικόνα 9: Εισαγωγή 2Δ Χώρου (β)	29
Εικόνα 10: Κάτοψη σε μορφή πλέγματος.....	30
Εικόνα 11: Επισήμανση ορίων χώρου από χρήστη	31
Εικόνα 12: Αυτόματη ανίχνευση εξωτερικού χώρου	32
Εικόνα 13: Παραμετροποίηση Angle Step σαρωτών.....	32
Εικόνα 14: Παραμετροποίηση Maximum Range σαρωτών.....	33
Εικόνα 15: Μήνυμα Scannalyze για καλές επιλογές παραμέτρων.....	33
Εικόνα 16: Μήνυμα Scannalyze για κακές επιλογές παραμέτρων.....	34
Εικόνα 17: Ολοκλήρωση παραμετροποίησης	34
Εικόνα 18: Ημιαυτόματη τοποθέτηση σαρωτών.....	35
Εικόνα 19: Τοποθέτηση πρώτου σαρωτή από το χρήστη	36
Εικόνα 20: Επιλογή επανατοποθέτησης.....	36
Εικόνα 21: Αποτέλεσμα ημιαυτόματης τοποθέτησης σαρωτών.....	37
Εικόνα 22: Κλάδεμα σαρωτών μετά από ημιαυτόματη τοποθέτηση	37
Εικόνα 23: Αυτόματη τοποθέτηση σαρωτών	38
Εικόνα 24: Αποτέλεσμα αυτόματης τοποθέτησης μετά το κλάδεμα.....	39
Εικόνα 25: Αποθήκευση 2Δ χώρου με σαρωτές	40
Εικόνα 26: Επιτυχής αποθήκευση	40
Εικόνα 27: Εξαγωγή αποτελέσματος ως εικόνα	41
Εικόνα 28: Το αποτέλεσμα που εξήχθη.....	41
Εικόνα 29: Σελίδα με αποθηκευμένους χώρους	42
Εικόνα 30: Επεξεργασία αποθηκευμένου χώρου.....	42
Εικόνα 31: Επιλογή αρχείου .obj.....	49
Εικόνα 32: Στεγανοποιημένο 3Δ μοντέλο.....	50
Εικόνα 33: Προσδιορισμός παραμέτρων και εκτέλεση PCBC	50
Εικόνα 34: Τοποθέτηση σαρωτών στο νέφος σημείων	51
Εικόνα 35: Εναλλακτική όψη του αποτελέσματος	51
Εικόνα 36: Παραγόμενη αναφορά τύπου Sharpley	52
Εικόνα 37: Περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Α.....	56
Εικόνα 38: Περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Β	57
Εικόνα 39: Περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Γ.....	57
Εικόνα 40: Χώρος με ανοιχτά παράθυρα/πόρτες	61
Εικόνα 41: Σάρωση σε χώρο με ανοιχτά παράθυρα/πόρτες	62
Εικόνα 42: 3Δ μοντέλο ναού ως .xyz στο MeshLab	63
Εικόνα 43: Αποτελέσματα πείραμα 1 με .xyz	63

Κατάλογος Πινάκων

Πίνακας 1: Δείγματα κατόψεων για τα πειράματα σε 2Δ χώρους.....	53
Πίνακας 2: Σύνοψη αποτελεσμάτων πειραμάτων	55
Πίνακας 3: Δείγματα μοντέλων για τα πειράματα σε 3Δ χώρους.....	58
Πίνακας 4: Αποτελέσματα πειραμάτων για χώρους 3Δ.....	59
Πίνακας 5: Νέφη σημείων με σαρωτές για χώρους 3Δ.....	60

Συντομογραφίες & Ακρωνύμια

AGP	Art Gallery Problem
CNN	Convolutional Neural Network
ILP	Integer Linear Programming
LiDAR	Light Detection and Ranging
MIP	Mixed Integer Programming
PCBC	Point Cloud Based Coverage
ΛΑ	Λειτουργική Απαίτηση

1. Εισαγωγή

Η παρούσα Πτυχιακή Εργασία έχει ως σημείο αφετηρίας το πρόβλημα της «γκαλερί έργων τέχνης» (Art Gallery Problem, AGP), το οποίο διατυπώθηκε για πρώτη φορά από τον Victor Klee το 1973: Δεδομένου ενός πολυγώνου που αναπαριστάνει την κάτοψη ενός μουσείου, ποιος είναι ο κατάλληλος αριθμός θέσεων στις οποίες πρέπει να τοποθετηθούν φύλακες, ώστε κάθε σημείο εντός του πολυγώνου να είναι ορατό από τουλάχιστον ένα φύλακα.

Το πρόβλημα αυτό αποτέλεσε αφορμή για πολλή έρευνα στο πεδίο, κυρίως, της υπολογιστικής γεωμετρίας (ενδεικτικά (O'Rourke, 1987)) και είναι πολύ σημαντικό για πλειάδα εφαρμογών. Άλλες σχετικές περιοχές συμπεριλαμβάνουν:

- Ρομποτική: τοποθέτηση καθαριστικών, περιπολικών ή επιθεωρητικών ρομπότ για πλήρη εποπτεία ενός χώρου, ή/και σχεδιασμό της διαδρομής τους για κάλυψη ορατότητας.
- Αναζήτηση και Διάσωση: βέλτιστη τοποθέτηση ομάδων ή drones για την κάλυψη όλης της περιοχής έρευνας.
- Επιτήρηση και Ασφάλεια: βέλτιστη τοποθέτηση καμερών ασφαλείας σε διάφορους χώρους ή/και σχεδιασμός βέλτιστης διαδρομής ομάδων περιπολίας.
- Περιβαλλοντική Παρακολούθηση: τοποθέτηση αισθητήρων για παρακολούθηση ρύπανσης, πυρκαγιών ή άγριας ζωής.
- Χωροταξία και Αστικός Σχεδιασμός: τοποθέτηση υποδομών (πύργοι παρατήρησης, φωτισμού ή σταθμοί έκτακτης ανάγκης).
- Ασύρματα Δίκτυα Αισθητήρων: σχεδιασμός θέσεων αισθητήρων για πλήρη κάλυψη σήματος ή δεδομένων.

Η ανάγκη για ακριβή και βελτιστοποιημένη αποτύπωση 3Δ περιβαλλόντων μέσω αισθητήρων LiDAR γίνεται ολοένα και πιο επιτακτική σε πεδία όπως η ψηφιοποίηση, η τεκμηρίωση και η ανάδειξη μνημείων πολιτιστικής κληρονομιάς. Χαρακτηριστικό παράδειγμα αποτελεί η συνδυαστική χρήση επίγειας σάρωσης λέιζερ (TLS) και φωτογραμμετρίας από UAV για την πλήρη και λεπτομερή 3Δ αναπαράσταση σύνθετων αρχιτεκτονικών δομών μεγάλης κλίμακας (Zachos & Anagnostopoulos, 2024). Στο πλαίσιο

αυτό, η βελτιστοποίηση της τοποθέτησης των αισθητήρων είναι κρίσιμη για την εξάλειψη των τυφλών σημείων (occlusions) και τη διασφάλιση της πληρότητας των δεδομένων.

1.1 Κύριες Προσεγγίσεις στο 2Δ Πρόβλημα

Το πρόβλημα όπως περιγράφηκε παραπάνω αναδεικνύει δύο ζητήματα, αυτό της ορατότητας ενός σημείου του χώρου από άλλα σημεία του χώρου, και το επακόλουθο ζήτημα της συνολικής κάλυψης του χώρου (ή του μέρους του χώρου που πρέπει να καλυφθεί). Οι προσεγγίσεις που έχουν αναπτυχθεί μπορούν να κατηγοριοποιηθούν ανάλογα με τη φύση της ανάλυσης που πραγματοποιούν και τους στόχους που επιδιώκουν. Οι κυριότερες από αυτές συνοψίζονται εδώ, και κάποιες εξετάζονται αναλυτικότερα στο Κεφάλαιο 2.

1.1.1 Θεωρητικές ή συνδυαστικές προσεγγίσεις

Οι θεωρητικές μέθοδοι επικεντρώνονται στην εξέταση των μαθηματικών χαρακτηριστικών του ζητήματος και στην απόδειξη ορίων και θεωρημάτων σχετικά με τον ελάχιστο αριθμό φυλάκων που απαιτείται για την πλήρη ασφάλεια ενός πολυγώνου. Το θεώρημα του Chvátal (1975) απέδειξε ότι οποιοδήποτε πολύγωνο με n κορυφές μπορεί να προστατευθεί επαρκώς με όχι περισσότερους από $\lfloor n/3 \rfloor$ φρουρούς, αριθμός που αντιπροσωπεύει το βέλτιστο όριο. Αργότερα, ο Fisk (1978) παρείχε μια απόδειξη χρησιμοποιώντας τριγωνοποίηση πολυγώνων, ενώ οι O'Rourke (1987) και Shermer (1992) επέκτειναν τα ευρήματα σε συγκεκριμένες κατηγορίες πολυγώνων (ορθογώνια, μονοτονικά, αστέρια).

1.1.2 Αλγοριθμικές ή υπολογιστικές προσεγγίσεις

Οι αλγοριθμικές τεχνικές (Lee, 1986; Ghosh, 2007; Eidenbenz *et al.*, 2001) επιδιώκουν να αντιμετωπίσουν το ζήτημα χρησιμοποιώντας υπολογιστικές μεθόδους, είτε μέσω ακριβών αλγορίθμων είτε μέσω προσεγγίσεων. Η βασική ιδέα περιλαμβάνει τον προσδιορισμό του χώρου ορατότητας και την κατασκευή ενός γραφήματος ορατότητας, το οποίο βοηθά στη δημιουργία πιθανών τοποθετήσεων φυλάκων. Πολλές μέθοδοι αυτού του είδους βασίζονται στη μετατροπή του προβλήματος της γκαλερί τέχνης σε πρόβλημα κάλυψης συνόλου, το οποίο αναγνωρίζεται ως NP-πλήρες.

1.1.3 Γεωμετρικές προσεγγίσεις

Οι γεωμετρικές μέθοδοι (Ghosh, 2007; O'Rourke, 2017) αξιοποιούν τα χαρακτηριστικά ορατότητας και τις γεωμετρικές διαμορφώσεις του πολυγώνου. Αυτές οι μέθοδοι περιλαμβάνουν τον υπολογισμό των πολυγώνων ορατότητας, τον διαχωρισμό του χώρου με τριγωνοποίηση ή κυρτή αποσύνθεση και τη δημιουργία ενός γραφήματος ορατότητας. Η κατανόηση αυτών των δομών διευκολύνει την αποτελεσματική ανάλυση των περιοχών που μπορεί να παρατηρήσει κάθε πιθανός φύλακας.

1.1.4 Προσεγγίσεις βελτιστοποίησης και μαθηματικού προγραμματισμού

Στις στρατηγικές βελτιστοποίησης, το πρόβλημα της γκαλερί τέχνης αντιμετωπίζεται με ακέραιο γραμμικό προγραμματισμό (ILP) ή μικτό ακέραιο προγραμματισμό (MIP). Αυτές οι τεχνικές εστιάζουν στην ελαχιστοποίηση του αριθμού των φυλάκων, εφαρμόζοντας περιορισμούς ορατότητας που εκφράζονται μαθηματικά (Kröller *et al.*, 2012; Tozoni *et al.*, 2016).

1.1.5 Ευρετικές και μεταευρετικές προσεγγίσεις

Οι ευρετικές και μεταευρετικές στρατηγικές χρησιμοποιούνται όπου οι ακριβείς μέθοδοι είναι υπερβολικά υπολογιστικά εντατικές. Για παράδειγμα, γενετικοί αλγόριθμοι, προσομοιωμένη ανόπτηση και τεχνικές βελτιστοποίησης σμήνους σωματιδίων έχουν χρησιμοποιηθεί για τη βελτίωση της κάλυψης ορατότητας (Bajuelos *et al.*, 2008; Amit *et al.*, 2007).

1.1.6 Εφαρμοσμένες προσεγγίσεις

Στις εφαρμοσμένες προσεγγίσεις, το πρόβλημα τροποποιείται ώστε να ταιριάζει σε πραγματικά σενάρια, λαμβάνοντας υπόψη παράγοντες όπως ενδεχόμενοι φυσικοί περιορισμοί, εμπόδια ή τρισδιάστατα σχήματα. Αυτές οι μέθοδοι είναι χρήσιμες σε τομείς όπως η ρομποτική παρακολούθηση, η τοποθέτηση καμερών ασφαλείας και η διαχείριση ασύρματων αισθητήρων (Huang *et al.*, 2005; Bodor *et al.*, 2007).

1.2 Κύριες προσεγγίσεις στο 3Δ Πρόβλημα

Η επέκταση του προβλήματος της γκαλερί έργων τέχνης σε τρισδιάστατους χώρους οδηγεί σε ένα σημαντικά πιο σύνθετο πεδίο έρευνας, όπου η ορατότητα και η κάλυψη εξαρτώνται από την τρισδιάστατη γεωμετρία, τις επιφάνειες πολλαπλών επιπέδων, τα εμπόδια και τον όγκο του περιβάλλοντος. Το 3Δ πρόβλημα είναι γνωστό ότι είναι ακόμη πιο δύσκολο από το αντίστοιχο 2Δ, καθώς πολλές διατυπώσεις του έχουν αποδειχθεί NP-δύσκολες ή ακόμη και μη προσεγγίσιμες εντός πολυωνυμικών παραγόντων σε συγκεκριμένες περιπτώσεις (Eidenbenz et al., 2001). Οι κυριότερες προσεγγίσεις στο 3Δ πρόβλημα συνοψίζονται εδώ και περιγράφεται πιο αναλυτικά αυτή που θα χρησιμοποιηθεί στην εργασία στο Κεφάλαιο 2.

1.2.1 Πολυεδρικές και Γεωμετρικές Προσεγγίσεις

Το περιβάλλον αναπαρίσταται ως πολυεδρικό μοντέλο, και η ορατότητα υπολογίζεται μέσω πολυγωνικών ή πολυεδρικών συμμετριών. Μελέτες όπως των Durand et al. (2002) αξιοποιούν πολυεδρικές δομές για τον υπολογισμό πολυέδρων ορατότητας, και 3D ray-shooting. Αν και θεωρητικά ακριβείς, αυτές οι προσεγγίσεις σπάνια εφαρμόζονται αυτούσιες σε μεγάλα περιβάλλοντα λόγω της αυξημένης υπολογιστικής πολυπλοκότητας

1.2.2 Προσεγγίσεις Ογκοστοιχείων

Οι μέθοδοι ογκοστοιχείων διακριτοποιούν τον χώρο σε ογκοστοιχεία (voxels), επιτρέποντας τον υπολογισμό της ορατότητας μέσω ray casting μέσα στο πλέγμα ογκοστοιχείων. Χρησιμοποιούνται ευρέως στη ρομποτική, την αυτόνομη πλοήγηση και προσομοιώσεις (Hornung et al., 2013). Υλοποιούνται σχετικά απλά και προσαρμόζονται καλά σε πολύπλοκα περιβάλλοντα, αλλά έχουν υψηλό υπολογιστικό κόστος, ανάλογα με την ανάλυση του πλέγματος.

1.2.3 Προσεγγίσεις Νέφους Σημείων

Οι μέθοδοι αυτές αντιμετωπίζουν το 3D περιβάλλον ως νέφος σημείων (π.χ. από LiDAR ή 3D scanning). Αποτελούν την πιο σύγχρονη και πρακτική προσέγγιση για προβλήματα

κάλυψης 3D, καθώς αποφεύγουν τη γεωμετρική πολυπλοκότητα των πολυεδρικών μοντέλων και την υπερβολική διακριτοποίηση των πλεγμάτων ογκοστοιχείων. Χρησιμοποιούν ray tracing πάνω σε point clouds (Kolluri, 2008), μέτρα επιφανειακής κάλυψης (Masuda, 2002), και επιλογή viewpoints για μέγιστη κάλυψη επιφανειών (Delmerico et al., 2018).

1.3 Προτεινόμενη Μεθοδολογία Επίλυσης του Προβλήματος

Σε αυτήν την Πτυχιακή Εργασία προτείνεται μια πρακτική και πειραματική μεθοδολογία για την επίλυση του προβλήματος της γκαλερί τέχνης, τόσο σε διδιάστατους όσο και σε τρισδιάστατους χώρους, με στόχο την κατανόηση και την αξιολόγηση διάφορων στρατηγικών κάλυψης και ορατότητας.

Η επίλυση του προβλήματος θα υλοποιηθεί σε δύο φάσεις:

1. Η πρώτη φάση αφορά χώρους σε δύο διαστάσεις (2D) (ουσιαστικά κατόψεις), οι οποίες είναι γεωμετρικά λιγότερο περίπλοκες και ευκολότερες στην απεικόνιση.
2. Η δεύτερη φάση αφορά χώρους σε τρεις διαστάσεις (3D), για να διερευνηθεί η εφαρμογή των αλγορίθμων σε πιο αληθοφανή περιβάλλοντα, όπως περιοχές με όγκο ή επιφάνειες πολλαπλών επιπέδων.

Στην διδιάστατη μορφή του προβλήματος, η κύρια ιδέα είναι να προσδιοριστεί η περιοχή ορατότητας κάθε δυνητικού φύλακα και να επιλεγεί ο ελάχιστος αριθμός θέσεων οι οποίες θα εγγυώνται την πλήρη κάλυψη του πολυγώνου. Για την αξιολόγηση της ορατότητας συνιστάται η χρήση αλγορίθμων Ray Tracing και Next-Best-View. Ο συνδυασμός των αλγορίθμων Ray Tracing και Next-Best-View αναμένεται να επιτρέψει την προοδευτική κάλυψη του διδιάστατου χώρου, ενώ παράλληλα θα οπτικοποιεί περιοχές οι οποίες δεν είναι ακόμη ορατές, προσφέροντας μια σαφή μέθοδο για την εύρεση της βέλτιστης τοποθέτησης των φυλάκων στο χώρο.

Στον τρισδιάστατο χώρο, όπου η έννοια της ορατότητας και της κάλυψης γίνεται πιο περίπλοκη εξαιτίας της παρουσίας όγκου, εμποδίων και επιφανειών σε διάφορα βάθη και ύψη. Για την αντιμετώπιση του προβλήματος στις τρεις διαστάσεις, συνιστάται η χρήση της μεθόδου Point Cloud Based Coverage, η οποία περιλαμβάνει τη μοντελοποίηση του χώρου ως νέφος σημείων (point cloud). Αυτή η προσέγγιση είναι ιδιαίτερα κατάλληλη για

περιβάλλοντα που δημιουργούνται από μοντέλα 3D (.obj, .xyz) ή σαρώσεις, καθώς κάθε επιφάνεια στο χώρο αντιπροσωπεύεται από μια πυκνή συλλογή σημείων, εξαλείφοντας την ανάγκη διακριτοποίησης σε voxels ή ορθογώνια τμήματα.

Η προτεινόμενη μεθοδολογία αποσκοπεί:

- στην σαφή και οπτικοποιημένη απεικόνιση του ζητήματος της ορατότητας,
- στην αξιολόγηση και σύγκριση των μεθόδων κάλυψης σε 2D και 3D όσον αφορά την αποτελεσματικότητα και την υπολογιστική πολυπλοκότητά τους,
- στην ανάπτυξη μιας λειτουργικής εφαρμογής που θα επιτρέπει τον πειραματισμό με διάφορες γεωμετρίες και παραμέτρους.

Αυτή η προσέγγιση συνδυάζει:

- απλότητα υλοποίησης και κατανόησης, καθιστώντας την προσιτή σε όσους δεν διαθέτουν εξειδικευμένες γνώσεις,
- επεκτασιμότητα, επιτρέποντας την προσθήκη πιο σύνθετων αλγορίθμων ή νέων τύπων εισαγωγής δεδομένων (π.χ. LiDAR data),
- πρακτική αξία, με πιθανές χρήσεις σε διάφορα σενάρια του πραγματικού κόσμου, όπως η επιτήρηση χώρου, ο χωροταξιακός σχεδιασμός ή η εικονική πραγματικότητα.

Με αυτόν τον τρόπο, η μετάβαση από το 2D στάδιο στο 3D στάδιο της εργασίας πραγματοποιείται ομαλά, με την ίδια βασική φιλοσοφία κάλυψης (Ray Tracing και Next-Best-View), μόνο που στον τρισδιάστατο χώρο η προσέγγιση εφαρμόζεται σε πιο ρεαλιστικά 3D δεδομένα νέφους σημείων.

1.4 Οργάνωση του τόμου

Το Κεφάλαιο 2 παρουσιάζει τη μαθηματική διατύπωση του προβλήματος τόσο στις δύο όσο και στις τρεις διαστάσεις και τους κυριότερους αλγόριθμους που χρησιμοποιούνται στην υλοποίηση της εφαρμογής. Το Κεφάλαιο 3 παρουσιάζει την ανάλυση των απαιτήσεων και το σχεδιασμό της εφαρμογής. Το Κεφάλαιο 4 περιέχει επισημάνσεις, αρχικά γενικά για όλη την εφαρμογή, και κατόπιν ξεχωριστά για το τμήμα της που αφορά τους χώρους στις δύο διαστάσεις και στις τρεις διαστάσεις. Για κάθε διάσταση δίνεται μια παραδειγματική εκτέλεση της εφαρμογής και εξηγούνται οι παράμετροι που χρησιμοποιήθηκαν. Το

Κεφάλαιο 5 παρουσιάζει περισσότερα αποτελέσματα από το κάθε τμήμα της εφαρμογής και σχολιάζει τα ευρήματα. Τέλος στο Κεφάλαιο 6 συνοψίζονται τα βασικά συμπεράσματα της εργασίας και προτείνονται μελλοντικές επεκτάσεις.

2 Θεωρητικό Υπόβαθρο

Στο Κεφάλαιο αυτό, παρουσιάζεται το θεωρητικό υπόβαθρο για το πρόβλημα της γκαλερί έργων τέχνης, και οι κυριότερες προσεγγίσεις για τη λύση του.

2.1 Μαθηματική Διατύπωση του Προβλήματος 2Δ

Δεδομένου ενός απλού πολυγώνου P με n κορυφές, ζητείται ο ελάχιστος αριθμός φυλάκων που μπορούν να τοποθετηθούν εντός του P , έτσι ώστε κάθε σημείο του P να είναι ορατό από τουλάχιστον έναν φύλακα.

Συμβολίζουμε ως:

$P \subseteq \mathbb{R}^2$ την πολυγωνική περιοχή που πρέπει να φυλαχθεί

$V = \{v_1, v_2, \dots, v_n\}$ τις κορυφές του πολυγώνου P

$G \subseteq P$ το σύνολο των υποψήφιων θέσεων φυλάκων εντός του P

$T \subseteq P$ το σύνολο των σημείων εντός του P που πρέπει οπωσδήποτε να είναι ορατά (δηλαδή να φυλαχθούν). Όταν το πρόβλημα τίθεται ως συνεχές, δηλαδή πρέπει να φυλαχθεί όλο το P , έχουμε $T=P$.

Για κάθε σημείο τοποθέτησης φύλακα $g \in G$ και κάθε σημείο $t \in T$ που πρέπει να φυλαχθεί, ορίζεται η σχέση ορατότητας

$$vis(g, t) = \begin{cases} 1 & \text{αν το γραμμικό τμήμα } gt \in P \\ 0 & \text{διαφορετικά} \end{cases}$$

Η σχέση αυτή ορίζει έναν διμερή γράφο ορατότητας μεταξύ των σημείων στους οποίους μπορούν να τοποθετηθούν φύλακες G και των σημείων που πρέπει οπωσδήποτε να φυλαχθούν T .

Για κάθε υποψήφια θέση τοποθέτησης φύλακα $g \in G$ ορίζεται μια δυαδική μεταβλητή

$$x_g = \begin{cases} 1 & \text{αν τοποθετείται φύλακας στο } g \\ 0 & \text{διαφορετικά} \end{cases}$$

Προκειμένου κάθε σημείο $t \in T$ να είναι ορατό από τουλάχιστον ένα φύλακα πρέπει να ισχύει:

$$\sum_{g \in G} vis(g, t) x_g \geq 1 \quad \forall t \in T$$

Και στόχος είναι να ελαχιστοποιηθεί η ποσότητα

$$\sum_{g \in G} x_g$$

Εναλλακτικά, το πρόβλημα μπορεί να διατυπωθεί μαθηματικά χρησιμοποιώντας την έννοια του συνόλου κάλυψης. Συμβολίζουμε ως:

$P \subset \mathbb{R}^2$ το πολύγωνο που πρέπει να φυλαχθεί/καλυφθεί.

$G \subseteq P$ το σύνολο των υποψηφίων θέσεων τοποθέτησης φυλάκων εντός του P

U το σύνολο των σημείων εντός του πολυγώνου P που πρέπει να καλυφθούν. Στην συνεχή εκδοχή $U = P$, ενώ στην διακριτή έχουμε $U = T$, όπου T είναι το σύνολο των διακριτών σημείων που πρέπει να καλυφθούν και $T \subseteq P$.

Για κάθε σημείο τοποθέτησης φύλακα $g \in G$ ορίζουμε το σύνολο ορατότητάς του

$$V_g = \{t \in U: gt \subseteq P\}$$

Οπότε κάθε $V_g \subseteq U$ περιέχει τα σημεία που είναι ορατά και μπορούν να καλυφθούν αν τοποθετηθεί φύλακας στη θέση g , και το σύνολο των συνόλων ορατότητας είναι:

$$V = \{V_g: g \in G\}$$

Ζητείται το μικρότερο υποσύνολο του V , έτσι ώστε η ένωση των στοιχείων του να αντιστοιχεί στο U , δηλαδή αναζητείται

$$C \subseteq V \text{ τέτοιο ώστε } \bigcup_{C_i \in C} C_i = U \text{ και } |C| \text{ ελάχιστο}$$

Ισοδύναμα, αναζητείται ελάχιστο υποσύνολο των θέσεων φύλαξης $S \subseteq G$, τέτοιο ώστε $\bigcup_{g \in S} V_g = U$.

2.2 Προσεγγίσεις

Σύμφωνα με το θεώρημα του Chvátal (1975) οποιοδήποτε πολύγωνο P με n κορυφές μπορεί να προστατευθεί επαρκώς με όχι περισσότερους από $\lceil n/3 \rceil$ φύλακες. Η απόδειξη χρησιμοποιεί δύο βασικές γεωμετρικές έννοιες:

- Τριγωνοποίηση: Κάθε απλό πολύγωνο που αποτελείται από n κορυφές μπορεί να χωριστεί σε $n-2$ μη επικαλυπτόμενα τρίγωνα χρησιμοποιώντας διαγώνιες που δεν τέμνονται.
- Χρωματισμός με 3 χρώματα: Στις κορυφές ενός τριγωνοποιημένου πολυγώνου μπορούν να αποδοθούν τρία χρώματα με τέτοιο τρόπο, ώστε καμία γειτονική κορυφή να μην έχει το ίδιο χρώμα. Η κλάση του χρώματος με τα λιγότερα μέλη υποδεικνύει το πλήθος των φυλάκων που απαιτούνται.

Δεδομένου ότι κάθε τρίγωνο περιέχει μία κορυφή από κάθε χρώμα, η επιλογή όλων των κορυφών ενός χρώματος εγγυάται ότι κάθε τρίγωνο καλύπτεται από τουλάχιστον μία επιλεγμένη κορυφή. Επιπλέον, εφόσον υπάρχουν μόνο τρία διαθέσιμα χρώματα, τουλάχιστον ένα χρώμα εφαρμόζεται σε $\lfloor n/3 \rfloor$ κορυφές το πολύ. Επομένως, δεν απαιτούνται περισσότεροι από $\lfloor n/3 \rfloor$ φύλακες.

Υπάρχουν διάφορες παραλλαγές και επεκτάσεις του θεωρήματος του Chvátal, ενδεικτικά:

- Ορθογώνια πολύγωνα: οι άκρες του πολυγώνου είναι ευθυγραμμισμένες με τους άξονες (είτε κάθετους είτε οριζόντιους), οπότε απαιτούνται το πολύ $\lfloor n/4 \rfloor$ φύλακες.
- Πολύγωνα με τρύπες (περιέχουν άλλα πολύγωνα, εμπόδια): αυτά είναι πιο περίπλοκα, και αντιστοιχούν πιο ρεαλιστικά σε αυτό που αντιλαμβανόμαστε ως κάτοψη ενός χώρου, με αποτέλεσμα να απαιτούνται περισσότεροι φύλακες.
- Κινητοί φύλακες: οι φύλακες επιτρέπεται να κινούνται κατά μήκος των ακμών ή των διαγωνίων, γεγονός που μπορεί να οδηγήσει σε μείωση του αριθμού τους.
- Τρισδιάστατοι χώροι: το πρόβλημα επεκτείνεται στα πολύεδρα, πράγμα σημαντικά πιο περίπλοκο και εν μέρει ακόμα άλυτο.

Στις αρχικές υπολογιστικές υλοποιήσεις του θεωρήματος του Chvátal, η τριγωνοποίηση είχε πολυπλοκότητα $O(n^2)$, αλλά ο αλγόριθμος Chazelle (1991) πέτυχε γραμμική πολυπλοκότητα, και στην υλοποίηση του Seidel (1991) έχουμε $O(n \log n)$.

Το θεώρημα του Chvátal έχει θεωρητική αξία, γιατί προσδιορίζει το άνω όριο στον αριθμό των φυλάκων που απαιτούνται για ένα πολύγωνο n κορυφών, αλλά δεν προσδιορίζει πώς μπορεί να υπολογιστεί αυτός ο αριθμός αποτελεσματικά. Οι πρώτοι επιτυχείς βασίστηκαν στα βήματα που χρησιμοποιήθηκαν για την απόδειξη του θεωρήματος του Chvátal, αλλά εφαρμόζονταν καλύτερα σε απλά πολύγωνα, και, συνήθως, δεν ήταν εγγυημένη η ελαχιστοποίηση του αριθμού των φυλάκων.

Στις πιο σύγχρονες προσεγγίσεις το πρόβλημα αντιμετωπίζεται ως πρόβλημα βελτιστοποίησης, και δεν χρησιμοποιείται άμεσα το θεώρημα του Chvátal, αλλά διατηρείται η βασική του διαίσθηση ότι πρόκειται για πρόβλημα κάλυψης μιας περιοχής χρησιμοποιώντας σύνολα ορατότητας (visibility sets). Αυτές οι προσεγγίσεις περιλαμβάνουν:

- Ακέραιο γραμμικό προγραμματισμό (Integer Linear Programming, ILP), όπου οι πιθανές θέσεις φυλάκων και τα σύνολα ορατότητας είναι περιορισμοί (Ahmed, *et al.*, 2018; Couto, *et al.*, 2009; Tozoni, *et al.*, 2009). Οι προσεγγίσεις αυτές είναι από τις πιο ακριβείς μεθόδους, αλλά έχουν σημαντικούς περιορισμούς. Ενώ το πρόβλημα τίθεται σε συνεχή χώρο, στις προσεγγίσεις αυτές απαιτείται διακριτοποίηση, οπότε, ακόμα κι αν η λύση που βρίσκεται είναι ακριβής για το διακριτό χώρο, δεν είναι βέλτιστη για τον συνεχή χώρο. Επίσης, οι προσεγγίσεις αυτές είναι κατάλληλες για πολύγωνα με σχετικά λίγες κορυφές, καθώς η ορατότητα κάθε υποψήφιου φύλακα πρέπει να υπολογιστεί εκ των προτέρων, και αυτό απαιτεί $O(n^3)$.
- Διατυπώσεις του προβλήματος με σύνολα κάλυψης, όπου κάθε φύλακας καλύπτει ένα υποσύνολο του πολυγώνου και ο στόχος είναι η κάλυψη όλου του πολυγώνου με τον μικρότερο αριθμό φυλάκων (Eidenbenz, *et al.*, 2001; Kröller, *et al.*, 2012). Και αυτές οι προσεγγίσεις έχουν παρόμοιους περιορισμούς με τις προσεγγίσεις ILP.
- Ευρετικές και μεταευρετικές προσεγγίσεις: αυτές επικεντρώνονται στην εύρεση σχεδόν βέλτιστων τοποθετήσεων φυλάκων χωρίς να επιλύουν ολόκληρο το μοντέλο. Ο Ghosh (2010) πρότείνει τη χρήση ευρετικών μεθόδων που επιλέγουν επαναληπτικά εκείνους τους φύλακες που καλύπτουν τις μεγαλύτερες ορατές περιοχές. Αν και αυτές οι μέθοδοι είναι απλές και γρήγορες, βασίζονται σε μεγάλο βαθμό στην διακριτοποίηση και μπορεί να οδηγήσουν στην τοποθέτηση περιττών φυλάκων. Ο Gonzales-Banos (2001) πρότείνει έναν τυχαίο αλγόριθμο που επιλέγει φύλακες με βάση την ικανότητά τους να καλύπτουν τα περισσότερα προηγούμενως ακάλυπτα σημεία, ενισχύοντας την επεκτασιμότητα, αλλά ενδεχομένως οδηγώντας σε τοπικά βέλτιστα αποτελέσματα, και μεταβλητότητα στα αποτελέσματα σε διαφορετικές εκτελέσεις. Στο (Bajuelos, *et al.*, 2008) χρησιμοποιήθηκαν γενετικοί αλγόριθμοι για να εξελίξουν πληθυσμούς συνόλων φυλάκων μέσω διασταύρωσης και μετάλλαξης λαμβάνοντας υπόψη την ποιότητα της κάλυψης. Το (Terhar, *et al.*, 2021) παρουσιάζει έναν εξελικτικό αλγόριθμο που βασίζεται σε βελτιστοποίηση σμήνους σωματιδίων (Particle Swarm Optimization, PSO)

για την επίλυση του προβλήματος. Παράγονται προσεγγιστικές, και όχι ακριβείς, λύσεις, και η μέθοδος που προτείνεται είναι σχετικά αργή για μεγάλα ή πολύπλοκα πολύγωνα.

Στη συνέχεια παρουσιάζονται πιο αναλυτικά δύο σύγχρονοι αλγόριθμοι για την πρακτική επίλυση του προβλήματος της γκαλερί τέχνης στις δύο διαστάσεις, καθώς συνδυάζουν υπολογιστική αποδοτικότητα με ακρίβεια στον προσδιορισμό ορατότητας. Ο αλγόριθμος Ray Tracing (Glassner, 1989) υπολογίζει τα σύνολα κάλυψης που απαιτούνται από τις μεθόδους βελτιστοποίησης, ενώ ο αλγόριθμος Next-Best-View (NBV) (Pito, 1999) προσφέρει μια επαναληπτική στρατηγική για τη δυναμική τοποθέτηση φυλάκων, με βάση την προοδευτική ελαχιστοποίηση της μη ορατής περιοχής.

2.3 Αλγόριθμος Ray Tracing

Ο αλγόριθμος Ray Tracing (ή «Ανίχνευσης Ακτίνων») (Glassner, 1989) είναι μία από τις κλασικότερες τεχνικές που χρησιμοποιούνται για την προσομοίωση ορατότητας. Η βασική του ιδέα περιλαμβάνει την εκπομπή ακτίνων από ένα συγκεκριμένο σημείο (όπως η θέση ενός φύλακα) προς όλες τις δυνατές κατευθύνσεις, και τον προσδιορισμό των ακτίνων που συναντούν εμπόδια. Αυτή η διαδικασία επιτρέπει τον υπολογισμό των ορατών περιοχών από κάθε σημείο του χώρου. Αν και αυτή η μέθοδος χρησιμοποιείται συνήθως στα γραφικά υπολογιστών για ρεαλιστική απεικόνιση του φωτός (Glassner, 1989), μπορεί επίσης να χρησιμοποιηθεί στο πρόβλημα της γκαλερί έργων τέχνης για τον υπολογισμό των πολυγώνων ορατότητας (visibility polygons) (Woo *et al.*, 1999).

Ο αλγόριθμος Ray Tracing δέχεται ως είσοδο (input):

- τη γεωμετρία του χώρου (π.χ. πολύγωνο, σύνολο τοιχωμάτων),
- τη θέση του φύλακα (ένα σημείο στο επίπεδο),
- τον αριθμό ή την πυκνότητα των ακτίνων που θα εκπέμπονται.

Και παράγει σαν έξοδο (output):

- το σύνολο των σημείων ή των τμημάτων του χώρου τα οποία είναι ορατά από έναν φύλακα.

Πιο συγκεκριμένα, από τη θέση που βρίσκεται ο φύλακας G , με συντεταγμένες (x_g, y_g) , εκπέμπονται ακτίνες σε όλες τις πιθανές κατευθύνσεις (από 0° έως 360°). Για κάθε ακτίνα, υπολογίζεται το πρώτο εμπόδιο (τοίχος, γωνία, αντικείμενο) στο οποίο προσκρούει. Το

σημείο πρόσκρουσης καταγράφεται ως το όριο της ορατότητας του φύλακα προς εκείνη την κατεύθυνση. Συνδέοντας διαδοχικά όλα τα σημεία πρόσκρουσης, σχηματίζεται το πολύγωνο ορατότητας του φύλακα G. Η ακρίβεια του αποτελέσματος εξαρτάται από το πλήθος των ακτίνων που εκπέμπονται, δηλαδή όσο περισσότερες είναι οι ακτίνες, τόσο πιο επακριβές είναι το περίγραμμα του πολυγώνου ορατότητας. Φυσικά, η ακρίβεια του αποτελέσματος εξαρτάται και από την εμβέλεια των ακτίνων.

Ακολουθεί ενδεικτικός ψευδοκώδικας του αλγορίθμου Ray Tracing:

Algorithm RayTracing

Input: polygon P, guard position G, number of rays N

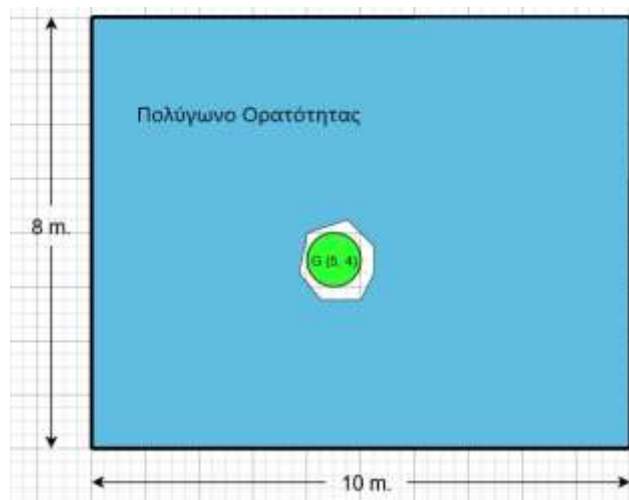
Output: visibility polygon V

```
V = [] //αρχικά άδεια λίστα
for i = 0 to N-1 do
    θ = 2π * i / N; //κατεύθυνση ακτίνας
    r = castRay(G, θ, P) //υπολογισμός σημείου πρόσκρουσης
    add r to V
end for
return polygon(V)
```

Ψευδοκώδικας 1: Αλγόριθμος Ray Tracing

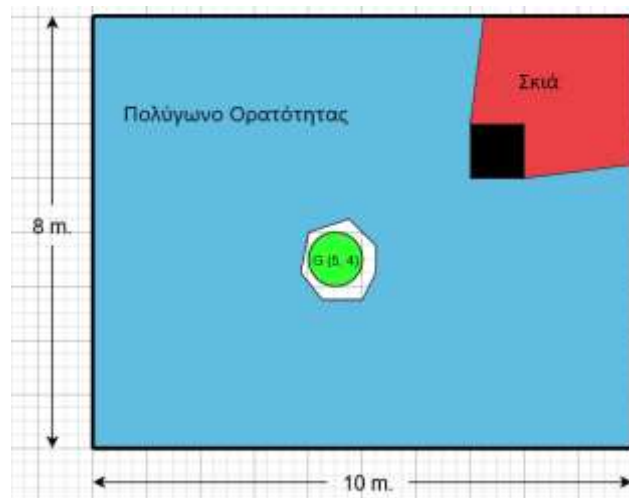
Η συνάρτηση $\text{castRay}(G, \theta, P)$ επιστρέφει το κοντινότερο σημείο P του πολυγώνου στο οποίο προσκρούει η ακτίνα που ξεκινά από το σημείο G με γωνία θ .

Για παράδειγμα, έστω ένα ορθογώνιο δωμάτιο διαστάσεων 10×8 m, με έναν φύλακα τοποθετημένο στις συντεταγμένες (5, 4). Ο αλγόριθμος Ray Tracing εκπέμπει 360 ακτίνες, μία για κάθε μοίρα ενός κύκλου. Κάθε ακτίνα επεκτείνεται μέχρι να συναντήσει έναν τοίχο και τότε σημειώνεται το σημείο επαφής. Συνδέοντας όλα αυτά τα σημεία, δημιουργείται το πολύγωνο ορατότητας. Στην περίπτωση που ο φύλακας βρίσκεται στο κέντρο, η ορατότητα καλύπτει όλο το δωμάτιο, όπως φαίνεται στην Εικόνα 1:



Εικόνα 1: Παράδειγμα Ray Tracing χωρίς εσωτερικά εμπόδια

Ωστόσο, εάν υπάρχει ένα εμπόδιο στο εσωτερικό του δωματίου, όπως μια κολόνα, το πολύγωνο ορατότητας θα εμφανίζει μια «σκιά» πίσω από την κολόνα, πράγμα που σημαίνει ότι θα υπάρχουν περιοχές που θα παραμένουν αόρατες στον φύλακα, όπως φαίνεται στην Εικόνα 2:



Εικόνα 2: Παράδειγμα Ray Tracing με εμπόδιο μια κολόνα

Ο αλγόριθμος Ray Tracing προσφέρει μια διαισθητική και προσαρμόσιμη προσέγγιση για την αξιολόγηση της ορατότητας στο πρόβλημα της γκαλερί τέχνης, με στόχο τον προσδιορισμό των περιοχών εντός ενός πολυγωνικού χώρου που είναι ορατές από έναν ή περισσότερους φύλακες. Ο αλγόριθμος μπορεί να διαχειριστεί αποτελεσματικά μη κυρτά σχήματα και πολλαπλά εμπόδια με απλό τρόπο, χωρίς την ανάγκη για πολύπλοκες μαθηματικές διατυπώσεις. Ο Pito (1999) επεσήμανε ότι η δειγματοληψία ορατότητας με

βάση την ανίχνευση ακτίνων είναι εννοιολογικά απλή και μπορεί εύκολα να προσαρμοστεί για επαυξητική ή βασισμένη σε αισθητήρες ανίχνευση χώρου, αν και η ακρίβειά της επηρεάζεται σε μεγάλο βαθμό από την γωνιακή ανάλυση και τον όγκο των ακτίνων που χρησιμοποιούνται. Επιπλέον οι de Berg et al. (2008) σημειώνουν ότι, ενώ η ανίχνευση ακτίνων αντιμετωπίζει το πρόβλημα της ορατότητας με σαφείς γεωμετρικούς όρους, είναι πιο απαιτητική από υπολογιστική άποψη σε σύγκριση με τις αναλυτικές μεθόδους που παράγουν ακριβή πολύγωνα ορατότητας. Συνοψίζοντας, η ανίχνευση ακτίνων εξακολουθεί να είναι μια πολύτιμη και συχνά χρησιμοποιούμενη μέθοδος για την ανάλυση της ορατότητας και για την επίδειξη του προβλήματος της γκαλερί τέχνης, ιδιαίτερα σε περιβάλλοντα προσομοίωσης ή οπτικοποίησης.

2.4 Αλγόριθμος Next-Best-View

Ο αλγόριθμος Next-Best-View (NBV) είναι μια επαναληπτική μέθοδος επιλογής θέσεων παρατήρησης (Conolly, 1985). Σε κάθε επανάληψη, επιλέγεται η επόμενη βέλτιστη θέση παρατήρησης (viewpoint) για τη μεγιστοποίηση της ορατότητας των ακάλυπτων περιοχών στο περιβάλλον. Αυτή η διαδικασία συνεχίζεται μέχρι οι επιλεγμένες θέσεις να εγγυώνται την πλήρη κάλυψη του χώρου. Η τεχνική αυτή χρησιμοποιείται συνήθως στην ρομποτική όραση και στη σάρωση τρισδιάστατων αντικειμένων, όπου η επίτευξη πλήρους χαρτογράφησης μιας περιοχής απαιτεί συνεχείς προσαρμογές στη θέση του αισθητήρα (Pito, 1999; Scott *et al.*, 2003).

Ο αλγόριθμος NBV δέχεται ως είσοδο (input):

- την περιγραφή του χώρου (π.χ. πολύγωνο, τρισδιάστατο μοντέλο),
- τις ήδη ορατές περιοχές ή τα δεδομένα που έχουν συλλεχθεί μέχρι στιγμής,
- ένα σύνολο υποψήφιων θέσεων παρατήρησης (viewpoints) ή ένα πλέγμα σημείων στο οποίο μπορεί να τοποθετηθεί ένας φύλακας.

Και παράγει σαν έξοδο (output):

- μια ακολουθία βέλτιστων θέσεων παρατήρησης που συνδυαστικά εγγυώνται τη μέγιστη δυνατή κάλυψη του χώρου.
- προαιρετικά, τον βαθμό κάλυψης (coverage ratio) μετά από κάθε επανάληψη.

Η βασική ιδέα του NBV είναι η σταδιακή βελτίωση της ορατότητας μέσω μιας σειράς επιλογών σημείων. Αρχικά, υποθέτουμε ότι καμία περιοχή δεν είναι ορατή. Ο αλγόριθμος υπολογίζει ποια τμήματα του χώρου είναι ήδη ορατά/γνωστά. Έπειτα, εντοπίζει πιθανά σημεία που μπορούν να αποκαλύψουν περιοχές που δεν ήταν ορατές προηγουμένως. Αξιολογεί κάθε πιθανή θέση σύμφωνα με ένα συγκεκριμένο κριτήριο (π.χ. το εμβαδόν της νέας ορατής περιοχής). Επιλέγει τη θέση που παρέχει την υψηλότερη απόδοση, δηλαδή τη μέγιστη νέα κάλυψη, και, τέλος, ενημερώνει το σύνολο ορατών περιοχών, και επαναλαμβάνει την ίδια διαδικασία έως ότου δεν υπάρχουν άλλες κρυφές περιοχές. Αυτή η προσέγγιση μπορεί να θεωρηθεί ως ένας κυκλικός μηχανισμός λήψης αποφάσεων, όπου κάθε νέα θέση παρατήρησης αντιπροσωπεύει την «επόμενη καλύτερη» απόφαση (Next-Best-View) για την επίτευξη πλήρους κάλυψης της περιοχής.

Ακολουθεί ενδεικτικός ψευδοκώδικας του αλγορίθμου NBV:

Algorithm NextBestView

Input: environment E, candidate viewpoints V, visibility function vis()

Output: ordered list of viewpoints NBV

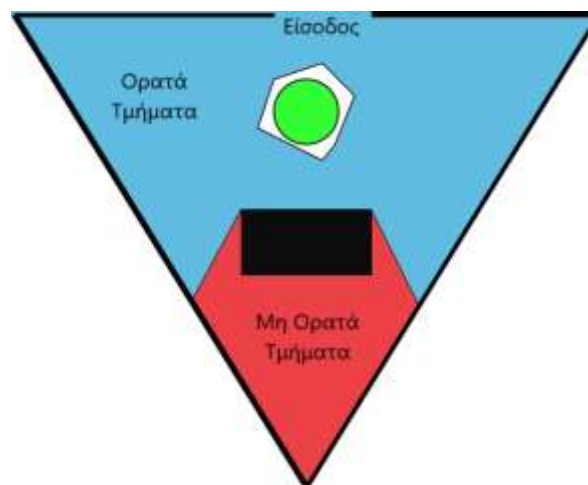
```
NBV = [] //άδεια λίστα
U = entire area of E //μη ορατή περιοχή

while U is not empty do
    best_view = None
    best_gain = 0
    for each v in V do
        new_visible = vis(v) ∩ U
        gain = area(new_visible)
        if gain > best_gain then
            best_gain = gain
            best_view = v
        end if
    end for
    add best_view to NBV
    U = U \ vis(best_view) //ενημέρωση μη ορατής περιοχής
end while
return NBV
```

Ψευδοκώδικας 2: Αλγόριθμος Next-Best-View

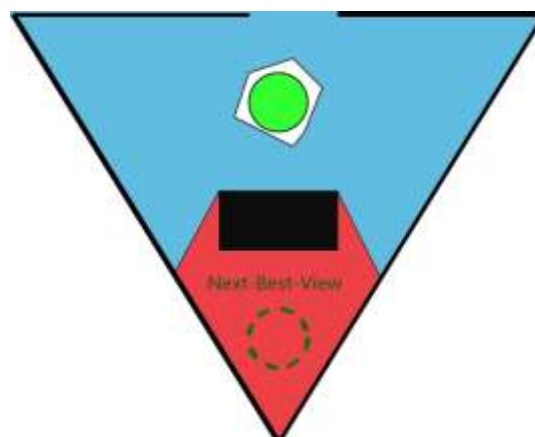
Η συνάρτηση $vis(v)$ επιστρέφει την περιοχή του χώρου η οποία είναι ορατή από τη θέση v . Το κριτήριο βελτιστοποίησης (*gain*) μπορεί να εκφράζει το ποσοστό νέας κάλυψης ή άλλες μετρικές, όπως η απόσταση ή το κόστος μετακίνησης.

Έστω ότι ο χώρος αποτελείται από ένα δωμάτιο με τρεις τοίχους και μια κεντρική κολώνα που εμποδίζει την ορατότητα, όπως φαίνεται στην Εικόνα 3. Στην αρχή, ο φύλακας τοποθετείται σε μια συγκεκριμένη θέση (π.χ. κοντά στην είσοδο του δωματίου). Ο αλγόριθμος προσδιορίζει τα τμήματα του δωματίου που είναι ορατά και μη ορατά.



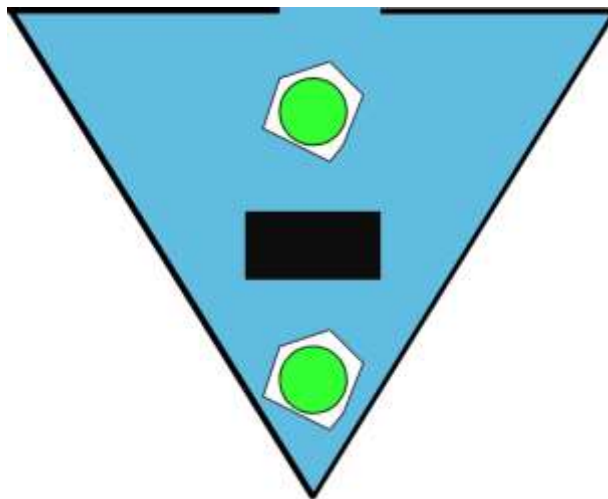
Εικόνα 3: Παράδειγμα Next-Best-View

Στη συνέχεια, εντοπίζει τη θέση που παρέχει την μεγαλύτερη πρόσθετη ορατότητα (π.χ. η περιοχή πίσω από την κολώνα). Αυτή η θέση ορίζεται ως η επόμενη βέλτιστη θέση παρατήρησης (Next-Best-View), όπως φαίνεται στην Εικόνα 4:



Εικόνα 4: Παράδειγμα Next-Best-View με εμπόδιο

Η διαδικασία επαναλαμβάνεται έως ότου καλυφθεί κάθε μέρος του δωματίου. Το τελικό αποτέλεσμα είναι μια ακολουθία θέσεων, στις οποίες αν τοποθετηθούν φύλακες ή σαρωτές, εξασφαλίζεται η πλήρης ορατότητα του χώρου, χρησιμοποιώντας τον ελάχιστο δυνατό αριθμό θέσεων παρατήρησης, όπως φαίνεται στην Εικόνα 5:



Εικόνα 5: Παράδειγμα Next-Best-View αποτέλεσμα

Ο αλγόριθμος Next-Best-View (NBV) προσφέρει μια δομημένη προσέγγιση για την αντιμετώπιση του προβλήματος της γκαλερί τέχνης, εντοπίζοντας σταδιακά σημεία παρατήρησης που βελτιώνουν την ορατότητα εκείνων των περιοχών που δεν καλύπτονταν προηγουμένως. Το κύριο πλεονέκτημά του έγκειται στην ικανότητά του να κατατάσσει τα σημεία παρατήρησης ανάλογα με την πληροφοριακή τους απόδοση, διευκολύνοντας την αποτελεσματική εξερεύνηση περίπλοκων, μη κυρτών πολυγώνων χωρίς την ανάγκη εξαντλητικής αξιολόγησης όλων των πιθανών τοποθετήσεων φυλάκων (Pito, 1999). Αυτή η μέθοδος είναι ιδιαίτερα επωφελής σε δυναμικά ή μερικώς παρατηρήσιμα περιβάλλοντα, όπου η κάλυψη πρέπει να προσαρμόζεται καθώς νέες περιοχές ανακαλύπτονται. Ο αλγόριθμος μπορεί επίσης να λάβει υπόψη τους περιορισμούς των αισθητήρων και τα ενδεχόμενα εμπόδια, καθιστώντας τον ευρέως εφαρμόσιμο σε τομείς όπως η ρομποτική επιτήρηση και οι εργασίες τρισδιάστατης ανακατασκευής (Zeng, *et al.*, 2020). Ωστόσο, ο αλγόριθμος NBV έχει ορισμένους περιορισμούς. Η επιτυχία του εξαρτάται σε μεγάλο βαθμό από την ευρετική μέθοδο που χρησιμοποιείται για την επιλογή του επόμενου σημείου θέασης, και αναποτελεσματικές ευρετικές μέθοδοι μπορεί να οδηγήσουν σε περιττές θεάσεις ή ατελή κάλυψη. Επιπλέον, ο αλγόριθμος NBV μπορεί να είναι υπολογιστικά απαιτητικός, ιδιαίτερα σε περιβάλλοντα που παρουσιάζουν πολυάριθμα υποψήφια σημεία

θέασης, καθώς το καθένα πρέπει να αξιολογείται ως προς τη συμβολή του στη συνολική ορατότητα (Pito, 1999; de Berg, *et al.*, 2008). Ανεξάρτητα από αυτούς τους περιορισμούς, ο αλγόριθμος NBV εξακολουθεί να αποτελεί ένα ευέλικτο και αποτελεσματικό πλαίσιο για την ανάλυση της σταδιακής ορατότητας τόσο σε θεωρητικές έρευνες όσο και σε πρακτικές εφαρμογές.

2.5 Μαθηματική Διατύπωση του Προβλήματος 3Δ

Στην τρισδιάστατη εκδοχή του προβλήματος, το ζητούμενο είναι να φυλαχθεί ένα πολύεδρο από τον μικρότερο κατά το δυνατόν αριθμό φυλάκων. Η μαθηματική διατύπωση είναι ίδια όπως και στην περίπτωση του προβλήματος 2Δ.

Συμβολίζουμε ως:

$P \subseteq \mathbb{R}^3$ μια κλειστή πολυεδρική περιοχή, η οποία μπορεί να διαθέτει εσωτερικούς τοίχους
 $G \subseteq P$ το σύνολο των υποψήφιων θέσεων τοποθέτησης φυλάκων εντός του P
 $T \subseteq P$ το σύνολο των θέσεων που πρέπει να φυλαχθούν (συνήθως $T=P$, αλλά έχουμε διακριτοποίηση του χώρου).

Ένας φύλακας g τοποθετημένος σε κάποια από τις θέσεις G επιτηρεί μια θέση t , αν το γραμμικό τμήμα μεταξύ τους βρίσκεται εντός του P και δεν διακόπτεται από τοίχο. Έτσι η σχέση ορατότητας μεταξύ g και t , ορίζεται όπως στην περίπτωση του προβλήματος 2Δ ως

$$vis(g, t) = 1 \Leftrightarrow gt \subset P$$

Το σύνολο ορατότητας κάθε υποψήφιας θέσης φύλακα ορίζεται όπως και στην 2Δ εκδοχή του προβλήματος, ως το σύνολο των σημείων που επιτηρούνται από αυτή τη θέση, δηλαδή

$$V_g = \{t \in T: vis(g, t) = 1\}$$

Το σύνολο ορατότητας κάθε φύλακα είναι πολύεδρο στην εκδοχή 3Δ του προβλήματος.

Το ζητούμενο είναι να επιλεγεί το μικρότερο δυνατό υποσύνολο των υποψήφιων θέσεων φύλαξης $S \subseteq G$ έτσι ώστε όλες οι προς φύλαξη θέσεις T να καλύπτονται:

$$\bigcup_{g \in S} V_g = T \text{ και } \min |S|$$

2.6 Αλγόριθμος Point Cloud Based Coverage

Η προσέγγιση Point Cloud Based Coverage χρησιμοποιείται συνήθως σε τομείς όπως η ρομποτική χαρτογράφηση, η τρισδιάστατη ανακατασκευή αντικειμένων και η ανάλυση ορατότητας σε τρισδιάστατα περιβάλλοντα (Yamauchi, 1997; Kriegel *et al.*, 2011; Delmerico *et al.*, 2018).

Η βασική ιδέα είναι να αντιμετωπίζεται κάθε σημείο στο νέφος σημείων ως ένα μικρό τμήμα επιφάνειας ή περιοχής που πρέπει να «δει» ο φύλακας ή ο σαρωτής. Ο στόχος είναι να επιλεγεί το μικρότερο δυνατό σύνολο σημείων παρατήρησης που μπορεί να μεγιστοποιήσει την ορατότητα των σημείων εντός του νέφους.

Ο αλγόριθμος δέχεται ως είσοδο (input):

- ένα σύνολο σημείων που αναπαριστά τον τρισδιάστατο χώρο,
- ένα σύνολο υποψήφιων θέσεων παρατήρησης (viewpoints),
- παραμέτρους πεδίου θέασης (Field of View, FOV) και μέγιστης απόστασης ορατότητας.

Και παράγει σαν έξοδο (output):

- το σύνολο των θέσεων παρατήρησης που καλύπτουν το μέγιστο δυνατό ποσοστό σημείων στο point cloud,
- το βαθμό κάλυψης (το ποσοστό των σημείων που έγιναν ορατά).

Πιο συγκεκριμένα, ο χώρος εισάγεται από ένα αρχείο .xyz ή .obj και μετατρέπεται σε νέφος σημείων. Για κάθε πιθανή θέση παρατήρησης, η ορατότητα αξιολογείται με προσομοίωση ακτίνων που εκπέμπονται προς τα σημεία του νέφους (συνδυασμός με Ray Tracing). Υπολογίζεται το ποσοστό των σημείων που είναι ορατά από κάθε θέση και επιλέγεται η θέση που παρέχει την μεγαλύτερη κάλυψη (δηλαδή τον μεγαλύτερο αριθμό σημείων που δεν καλύπτονταν προηγουμένως). Η διαδικασία επαναλαμβάνεται έως ότου καλυφθεί το μέγιστο δυνατό ποσοστό σημείων ή έως ότου επιτευχθεί ένας καθορισμένος αριθμός σαρωτών. Ως αποτέλεσμα, υπολογίζεται η κάλυψη χρησιμοποιώντας πραγματικά δεδομένα γεωμετρίας 3D, χωρίς να χρειάζεται τμηματοποίηση του χώρου σε πλέγμα voxel, βελτιώνοντας έτσι την ακρίβεια και την ευελιξία.

Ακολουθεί ενδεικτικός ψευδοκώδικας:

```
Algorithm PointCloudCoverage
```

```
Input: point cloud P, candidate viewpoints V, visibility function vis()
```

```
Output: selected viewpoints S
```

```
S = [] //άδεια λίστα  
U = P //μη ορατά σημεία
```

```
while U is not empty do
```

```
best_view = None
best_gain = 0
for each v in V do
    visible_points = vis(v, P)
    new_visible = visible_points  $\cap$  U
    gain = |new_visible|
    if gain > best_gain then
        best_gain = gain
        best_view = v
    end if
end for
add best_view to S
U = U \ vis(best_view, P) //ενημέρωση μη ορατών σημείων
end while
return S
```

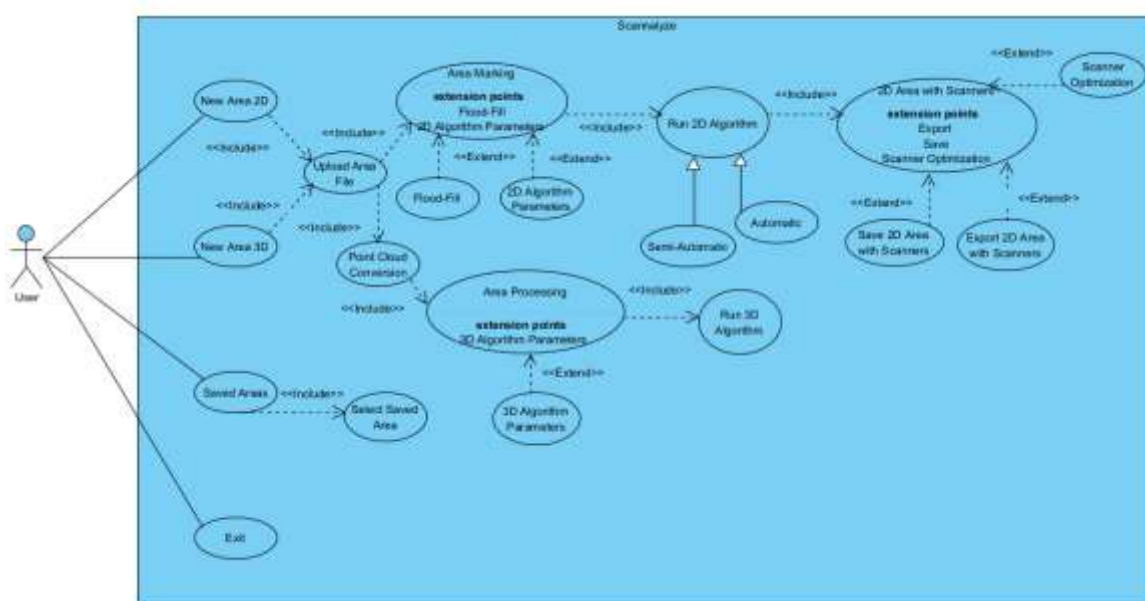
Ψευδοκώδικας 3: Αλγόριθμος Point Cloud Based Coverage

Η συνάρτηση $\text{vis}(v, P)$ υπολογίζει ποια σημεία του νέφους είναι ορατά από το viewpoint v , χρησιμοποιώντας έναν βασικό αλγόριθμο Ray Tracing για την ανίχνευση εμποδίων.

Για παράδειγμα, έστω ότι ο χρήστης εισάγει ένα τρισδιάστατο μοντέλο δωματίου μέσω αρχείου .fbx. Το μοντέλο μετατρέπεται σε ένα νέφος σημείων που περιέχει περίπου 50.000 σημεία, τα οποία αντιπροσωπεύουν στοιχεία όπως τοίχους, έπιπλα και δάπεδο. Ο αλγόριθμος προσδιορίζει πιθανές θέσεις φυλάκων στις άκρες ή στη μέση του δωματίου και αξιολογεί πόσα σημεία από το νέφος είναι ορατά από κάθε μία από αυτές τις θέσεις. Η θέση που παρατηρεί τον μεγαλύτερο αριθμό σημείων ορίζεται ως η πρώτη θέση φύλακα/σαρωτή. Αυτή η διαδικασία επαναλαμβάνεται (παρόμοια με τον αλγόριθμο NBV) έως ότου είναι ορατό σχεδόν ολόκληρο το νέφος σημείων. Το αποτέλεσμα αποτελείται από μια σειρά βέλτιστων θέσεων παρατήρησης στον τρισδιάστατο χώρο, που εγγυώνται τη μέγιστη ορατότητα της περιοχής με τον ελάχιστο αριθμό σαρωτών.

3 Ανάλυση Απαιτήσεων και Σχεδιασμός Εφαρμογής

Στα πλαίσια της παρούσας πτυχιακής εργασίας αναπτύσσεται μια εφαρμογή (Scannalyze) μέσω της οποίας εντοπίζονται οι κατάλληλες θέσεις τοποθέτησης φυλάκων σε διδιάστατους και τρισδιάστατους χώρους. Αρχικά περιγράφεται η εφαρμογή μέσω του διαγράμματος περιπτώσεων χρήσης το οποίο παρουσιάζεται στην Εικόνα 6 και εξηγείται στην συνέχεια με αναφορά στις λειτουργικές απαιτήσεις (ΛΑ).



Εικόνα 6: Διάγραμμα Περιπτώσεων Χρήσης Scannalyze

Εκκίνηση Εφαρμογής

ΛΑ-1: Κατά την εκκίνηση της εφαρμογής, το σύστημα πρέπει να εμφανίζει την αρχική οθόνη.

ΛΑ-2: Η αρχική οθόνη πρέπει να παρέχει στον χρήστη τις ακόλουθες επιλογές:

- Μπάρα πλοήγησης
- Εισαγωγή νέου 2D χώρου (New Area 2D)
- Εισαγωγή νέου 3D χώρου (New Area 3D)
- Προβολή αποθηκευμένων χώρων (Saved Areas)
- Τερματισμός εφαρμογής (Exit)

ΛΑ-3: Το σύστημα πρέπει να επιτρέπει την επιλογή λειτουργίας μέσω γραφικής διεπαφής χρήστη (GUI).

ΛΑ-4: Σε περίπτωση επιλογής «Exit», το σύστημα πρέπει να τερματίζει τη λειτουργία του με ασφαλή τρόπο.

Μπάρα πλοήγησης

ΛΑ-1: Η μπάρα πλοήγησης πρέπει να υπάρχει σε κάθε σελίδα της εφαρμογής.

ΛΑ-2: Η μπάρα πλοήγησης πρέπει να παρέχει στον χρήστη τις ακόλουθες επιλογές:

- να επιστρέψει στην αρχική οθόνη της εφαρμογής (Home)
- να διαβάσει πληροφορίες για την εφαρμογή (About)
- να βρει τα στοιχεία επικοινωνίας του δημιουργού (Contact)

Εισαγωγή νέου 2D χώρου (New Area 2D)

ΛΑ-1: Το σύστημα πρέπει να ζητά από τον χρήστη να ανεβάσει ένα αρχείο με την κάτοψη του χώρου.

ΛΑ-2: Το αρχείο με την κάτοψη πρέπει να φορτώνεται σε grid μορφή και ο χρήστης πρέπει να μπορεί:

- να μαρκάρει τοίχους και εμπόδια μέσα στον χώρο
- να μαρκάρει τον χώρο εκτός κάτοψης (εφόσον αυτός υπάρχει)

ΛΑ-3: Το σύστημα πρέπει να ζητά από τον χρήστη να εισάγει επιθυμητές τιμές στις παραμέτρους με τις οποίες θα εκτελεστεί ο αλγόριθμος τοποθέτησης σαρωτών και να παρέχει προεπιλεγμένες τιμές:

- Angle Step (απόσταση μεταξύ των ακτινών που θα εκπέμπει ο κάθε σαρωτής)
- Maximum Range (εμβέλεια των ακτινών του κάθε σαρωτή)

ΛΑ-4: Το σύστημα οφείλει να ενημερώνει τον χρήστη για την ποιότητα των αποτελεσμάτων που θα προκύψουν με βάση τις τιμές των παραμέτρων που εκείνος επέλεξε.

ΛΑ-5: Με το πάτημα του κουμπιού «Scannalyze», το σύστημα πρέπει να αποθηκεύει προσωρινά τον μαρκαρισμένο 2D χώρο και τις τιμές των παραμέτρων για επεξεργασία, να ενημερώνει τον χρήστη με σχετικό μήνυμα και να τον οδηγεί στη σελίδα τοποθέτησης των σαρωτών.

Τοποθέτηση σαρωτών σε 2D χώρο

ΛΑ-1: Το σύστημα πρέπει να φορτώνει τον μαρκαρισμένο χώρο που εισήγαγε ο χρήστης, χωρίς τοποθετημένους σαρωτές, ακόμα.

ΛΑ-2: Το σύστημα πρέπει να παρέχει στον χρήστη τη δυνατότητα επιλογής μεταξύ ημιαυτόματης τοποθέτησης σαρωτών και αυτόματης τοποθέτησης σαρωτών.

Ημιαυτόματη τοποθέτηση σαρωτών σε 2D χώρο

ΛΑ-1: Το σύστημα πρέπει να ζητά από τον χρήστη να τοποθετήσει εκείνος όπου επιθυμεί τους πρώτους 2 σαρωτές μέσα στον χώρο.

ΛΑ-2: Ο χρήστης πρέπει να μπορεί:

- να τοποθετήσει έως και 2 σαρωτές μέσα στον ελεύθερο εσωτερικό χώρο της κάτοψης
- να αφαιρέσει έναν ή περισσότερους τοποθετημένους σαρωτές για να κάνει επανατοποθέτηση
- να δει τον συνολικό αριθμό τοποθετημένων σαρωτών καθώς και το ποσοστό κάλυψης του χώρου

ΛΑ-3: Αφού τοποθετήσει ο χρήστης χειροκίνητα τους πρώτους 2 σαρωτές, το σύστημα πρέπει να λάβει υπόψη του τις τοποθεσίες αυτών των 2 σαρωτών και να τοποθετήσει αυτόματα όσους ακόμα χρειάζονται για την πλήρη κάλυψη του χώρου εκτελώντας τους αλγόριθμους για τους διδιάστατους χώρους (Ray Tracing και Next-Best-View), σύμφωνα με τις παραμέτρους που είχε εισάγει ο χρήστης νωρίτερα.

Αυτόματη τοποθέτηση σαρωτών σε 2D χώρο

ΛΑ-1: Το σύστημα πρέπει να εκτελεί τους αλγόριθμους τοποθέτησης σαρωτών για διδιάστατους χώρους (Ray Tracing και Next-Best-View), σύμφωνα με τις παραμέτρους που είχε εισάγει ο χρήστης νωρίτερα.

ΛΑ-2: Ο χρήστης πρέπει να μπορεί να δει τον συνολικό αριθμό τοποθετημένων σαρωτών καθώς και το ποσοστό κάλυψης του χώρου.

Οπτικοποίηση αποτελεσμάτων 2D

ΛΑ-1: Το σύστημα πρέπει να εμφανίζει τον χώρο που εισήγαγε ο χρήστης μαζί με τα αποτελέσματα κάθε μεθόδου τοποθέτησης (ημιαυτόματης ή αυτόματης).

ΛΑ-2: Οι θέσεις των σαρωτών πρέπει να απεικονίζονται με σαφή και διακριτό τρόπο, καθώς και το πολύγωνο ορατότητας (visibility polygon) του καθενός από αυτούς.

ΛΑ-3: Το σύστημα πρέπει να επιτρέπει την επιστροφή στο στάδιο εισαγωγής του διδιάστατου χώρου για τροποποιήσεις.

ΛΑ-4: Το σύστημα πρέπει να παρέχει τη δυνατότητα βελτιστοποίησης του αποτελέσματος, επανεξετάζοντας τον χώρο με τους τοποθετημένους σαρωτές και αφαιρώντας όσους είναι περιττοί και δεν συμβάλουν στην πλήρη κάλυψη του χώρου, εφόσον αυτοί υπάρχουν.

ΛΑ-5: Ο χρήστης πρέπει να ενημερώνεται:

- για τις συντεταγμένες των σαρωτών που αφαιρέθηκαν
- για την ενδεχόμενη μη εύρεση περιττών σαρωτών

Αποθήκευση και Εξαγωγή

ΛΑ-1: Το σύστημα πρέπει να παρέχει δυνατότητα αποθήκευσης του χώρου και των αποτελεσμάτων σε αρχείο.

ΛΑ-2: Τα αποθηκευμένα δεδομένα πρέπει να περιλαμβάνουν:

- τη γεωμετρία του χώρου
- τις τελικές θέσεις των σαρωτών
- τα πολύγωνα ορατότητας των σαρωτών

ΛΑ-3: Το σύστημα πρέπει να ζητά από τον χρήστη να ονομάσει το αρχείο πριν το αποθηκεύσει και να τον ενημερώνει αν η αποθήκευση ήταν επιτυχής ή όχι.

ΛΑ-4: Το σύστημα πρέπει να επιτρέπει την εξαγωγή του τελικού αποτελέσματος ως εικόνα (π.χ. PNG).

ΛΑ-5: Ο χρήστης πρέπει να μπορεί να επιστρέψει στην αρχική οθόνη μετά την αποθήκευση ή την εξαγωγή του αποτελέσματος.

Προβολή αποθηκευμένων 2D χώρων (Saved Areas)

ΛΑ-1: Το σύστημα πρέπει να εμφανίζει στον χρήστη μια λίστα με όλους τους διδιάστατους χώρους στους οποίους έχει ήδη τοποθετήσει σαρωτές και έχει αποθηκεύσει σε αλφαβητική σειρά.

ΛΑ-2: Το σύστημα πρέπει να επιτρέπει στον χρήστη να ανοίξει έναν αποθηκευμένο χώρο και να δει τους σαρωτές που έχει τοποθετήσει ή να επανατοποθετήσει σαρωτές και γενικότερα να επεξεργαστεί ξανά τον χώρο.

Εισαγωγή νέου 3D χώρου (New Area 3D)

ΛΑ-1: Το σύστημα πρέπει να ζητά από τον χρήστη να ανεβάσει ένα αρχείο του τρισδιάστατου χώρου.

ΛΑ-2: Το σύστημα πρέπει να μετατρέπει τον τρισδιάστατο χώρο σε νέφος σημείων.

ΛΑ-3: Το σύστημα πρέπει να ζητά από τον χρήστη να εισάγει επιθυμητές τιμές στις παραμέτρους με τις οποίες θα εκτελεστεί ο αλγόριθμος τοποθέτησης σαρωτών και να παρέχει προεπιλεγμένες τιμές:

- Maximum Range (εμβέλεια των ακτινών του κάθε σαρωτή)
- Ideal Coverage Percent (επιθυμητό ποσοστό κάλυψης)

ΛΑ-4: Με το πάτημα του κουμπιού «Run Scanner Placement», το σύστημα πρέπει να τρέχει τον αλγόριθμο τοποθέτησης σαρωτών PCBC και να εμφανίζει μήνυμα «Placing Scanners...» όσο ο χρήστης περιμένει να φορτώσει το τελικό αποτέλεσμα.

Οπτικοποίηση αποτελεσμάτων 3D

ΛΑ-1: Το σύστημα πρέπει να εμφανίζει τον χώρο που εισήγαγε ο χρήστης μαζί με τους τοποθετημένους σαρωτές σε ξεχωριστό παράθυρο.

ΛΑ-2: Το σύστημα πρέπει να παρέχει πληροφορίες στον χρήστη σχετικά με το αποτέλεσμα που βλέπει στην οθόνη του (π.χ. οι κόκκινες σφαίρες αποτελούν τους σαρωτές, οι γαλάζιες κουκίδες είναι τα ορατά σημεία, κλπ.)

ΛΑ-3: Το σύστημα πρέπει να επιτρέπει στον χρήστη να περιηγηθεί μέσα στο παράθυρο του αποτελέσματος χρησιμοποιώντας το ποντίκι του υπολογιστή.

ΛΑ-4: Το σύστημα πρέπει να επιστρέφει το τελικό ποσοστό κάλυψης του χώρου.

ΛΑ-5: Το σύστημα πρέπει να επιτρέπει την επιστροφή στο στάδιο εισαγωγής του τρισδιάστατου χώρου για τροποποιήσεις και επανεκτέλεση του αλγόριθμου.

ΛΑ-5: Το σύστημα πρέπει να παράγει και να επιστρέφει αναφορά τύπου Shapley με τη συνεισφορά κάθε σαρωτή στην συνολική κάλυψη του χώρου.

4 Υλοποίηση

Στο κεφάλαιο αυτό εξηγούνται λεπτομέρειες της υλοποίησης της εφαρμογής που κρίθηκαν άξιες λόγου και περιγράφεται η λειτουργία της με παραδειγματικές εκτελέσεις.

4.1 Γενική Υλοποίηση

Η εφαρμογή Scannalyze υλοποιήθηκε σε προσωπικό υπολογιστή με λειτουργικό σύστημα Windows 10 και επεξεργαστή Intel Core i5 10th Gen. Η υλοποίηση έγινε στη γλώσσα προγραμματισμού Python με χρήση κατάλληλων βιβλιοθηκών για γραφικά, γεωμετρικούς υπολογισμούς, διαχείριση αρχείων κλπ. Η διεπαφή χρήστη υλοποιήθηκε με τη χρήση HTML, CSS και JavaScript. Στην τρέχουσα έκδοση οι κατόψεις διδιάστατων χώρων εισάγονται ως PNG, αλλά σχετικά εύκολα αυτό μπορεί να επεκταθεί ώστε να περιλαμβάνει άλλους τύπους αρχείων εικόνων. Τα αρχεία αυτά, μετά την φόρτωσή τους και την επισήμανση (μαρκάρισμα) από τον χρήστη, αποθηκεύονται προσωρινά ως CSV. Οι διδιάστατοι χώροι, μετά την επεξεργασία τους και την τοποθέτηση σαρωτών αποθηκεύονται ως αρχεία JSON, ώστε να είναι εφικτή η επανεπεξεργασία τους.

4.2 Επεξεργασία 2Δ Χώρων

Όπως προαναφέρθηκε στο Κεφάλαιο 2 οι αλγόριθμοι που υλοποιήθηκαν για την τοποθέτηση σαρωτών σε χώρους δύο διαστάσεων είναι οι Ray Tracing και Next-Best-View, που εξηγήθηκαν στις ενότητες 2.3 και 2.4.

Κατά την υλοποίηση των αλγορίθμων στην εφαρμογή Scannalyze προβλέφθηκε ένα προαιρετικό στάδιο προεπεξεργασίας καθώς και ένα προαιρετικό στάδιο εκ των υστέρων επεξεργασίας, προκειμένου να συγκριθούν τα αποτελέσματα μεταξύ διαφορετικών σεναρίων τοποθέτησης σαρωτών και να εξαχθούν συμπεράσματα.

Συγκεκριμένα, σε σχέση με το στάδιο προεπεξεργασίας:

Ο χρήστης της εφαρμογής έχει την δυνατότητα να επιλέξει την τοποθέτηση σαρωτών από το σύστημα, δηλαδή την εκτέλεση των αλγορίθμων με βάση κάποιες αρχικές παραμέτρους ή default τιμές (σενάριο 1, αυτόματη τοποθέτηση σαρωτών).

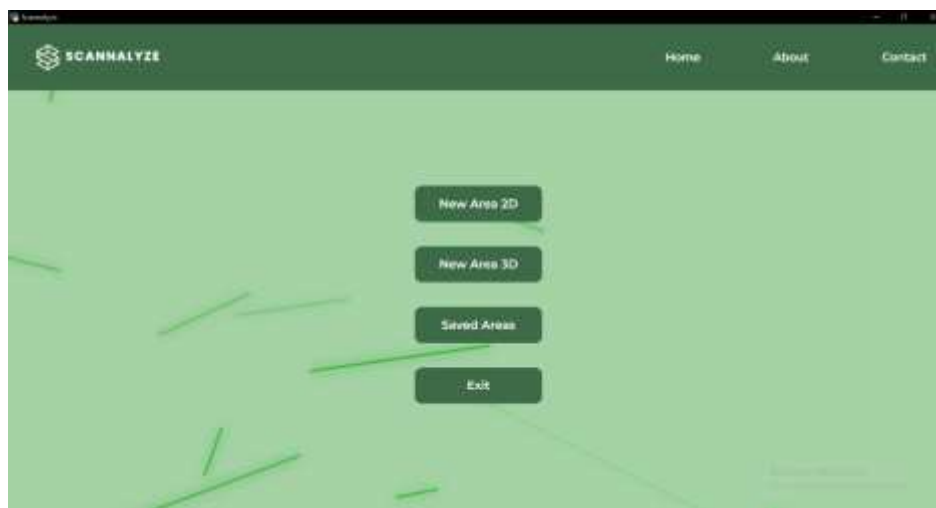
Επιπλέον, ο χρήστης μπορεί να επιλέξει να τοποθετήσει αρχικά ο ίδιος δύο σαρωτές στον χώρο, βασισμένος στο προσωπικό του κριτήριο καθώς παρατηρεί την κάτοψη, και κατόπιν να εκτελεστούν οι αλγόριθμοι με τις παραμέτρους που έχουν οριστεί ή default τιμές (σενάριο 2, ημιαυτόματη τοποθέτηση σαρωτών).

Ανεξάρτητα από τις επιλογές του χρήστη για την προαιρετική εφαρμογή του σταδίου προεπεξεργασίας, μετά την εκτέλεση των αλγορίθμων που έχουν υλοποιηθεί, ο χρήστης έχει την δυνατότητα να επιλέξει την εφαρμογή ενός σταδίου εκ των υστέρων επεξεργασίας. Συγκεκριμένα, ο χρήστης μπορεί να ζητήσει την βελτιστοποίηση του πλήθους των σαρωτών που έχουν υπολογιστεί, χωρίς να μειωθεί το ποσοστό κάλυψης του χώρου. Αυτή η εκ των υστέρων επεξεργασία προστέθηκε σε μια προσπάθεια να επιτευχθεί το θεωρητικό βέλτιστο που παρέχει το θεώρημα του Chvátal, καθώς είναι δυνατόν ένα ή περισσότερα σημεία του χώρου να καλύπτονται από περισσότερους από έναν σαρωτές. Διαισθητικά η ιδέα είναι ότι σαρωτές που καλύπτουν ένα σημείο το οποίο καλύπτεται και από άλλους σαρωτές, δεν απαιτείται να συμπεριληφθούν.

Τα παραπάνω καθίστανται σαφέστερα μέσω της παραδειγματικής εκτέλεσης της εφαρμογής στην επόμενη ενότητα.

4.3 Παραδειγματική Εκτέλεση για 2Δ χώρους

Κατά την εκκίνηση της εφαρμογής, ο χρήστης βρίσκεται στην αρχική οθόνη (Εικόνα 7), η οποία περιλαμβάνει μια μπάρα πλοήγησης στο πάνω μέρος και τέσσερα κουμπιά στο κέντρο:

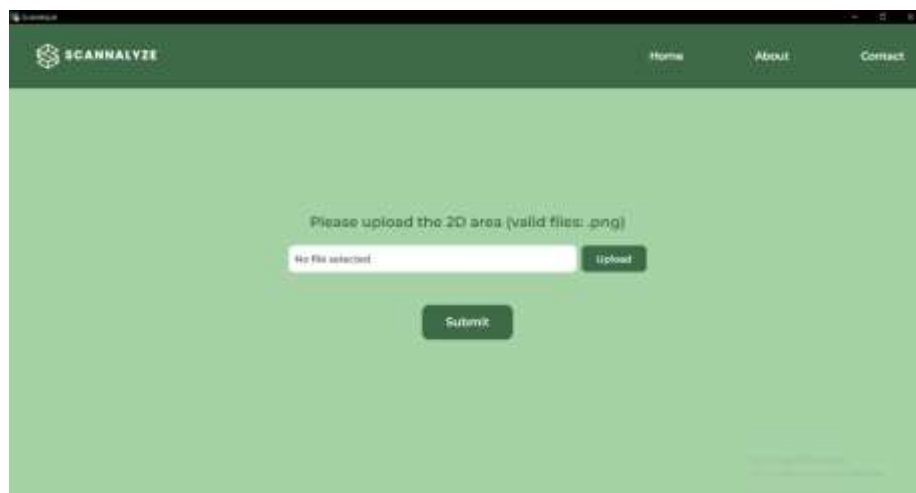


Εικόνα 7: Αρχική Οθόνη εφαρμογής Scannalyze

Στην μπάρα πλοήγησης, το κουμπί “Home” οδηγεί στην αρχική οθόνη της εφαρμογής. Τα κουμπιά “About” και “Contact” βρίσκονται εκεί καθαρά για λόγους σχεδίασης και δεν έχουν κάποια σημαντική λειτουργικότητα.

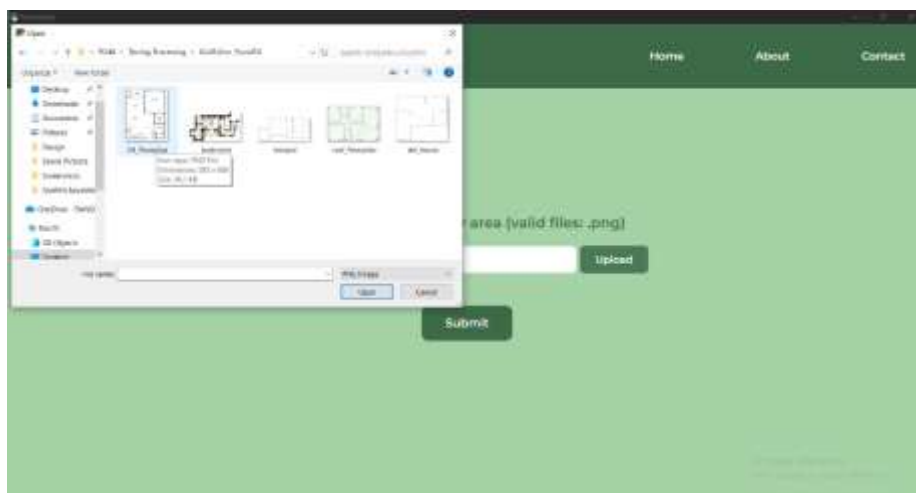
Το κουμπί “New Area 2D” επιτρέπει στον χρήστη να εισάγει έναν διδιάστατο χώρο και να τοποθετήσει σαρωτές για την κάλυψή του. Αντίστοιχα, το κουμπί “New Area 3D” εκτελεί την ίδια λειτουργία για τρισδιάστατους χώρους. Το κουμπί “Saved Areas” δίνει τη δυνατότητα στον χρήστη να δει και να επανεπεξεργαστεί, εφόσον ο ίδιος το επιθυμεί, χώρους (διδιάστατους και τρισδιάστατους) με τοποθετημένους σαρωτές που έχει αποθηκεύσει και τέλος, το κουμπί “Exit” τερματίζει την εφαρμογή.

Επιλέγοντας το “New Area 2D”, ο χρήστης οδηγείται στην σελίδα εισαγωγής ενός νέου διδιάστατου χώρου (Εικόνα 8):



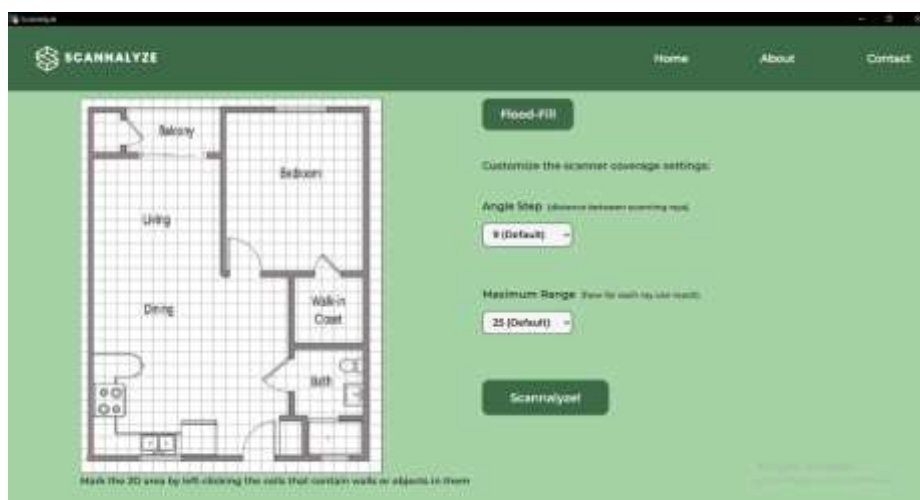
Εικόνα 8: Εισαγωγή Χώρου 2Δ (α)

Η εφαρμογή του ζητά να εισάγει τον διδιάστατο χώρο πατώντας το κουμπί “Upload” (Εικόνα 9). Τα επιτρεπτά αρχεία είναι τύπου PNG, προς το παρόν:



Εικόνα 9: Εισαγωγή 2Δ Χώρου (β)

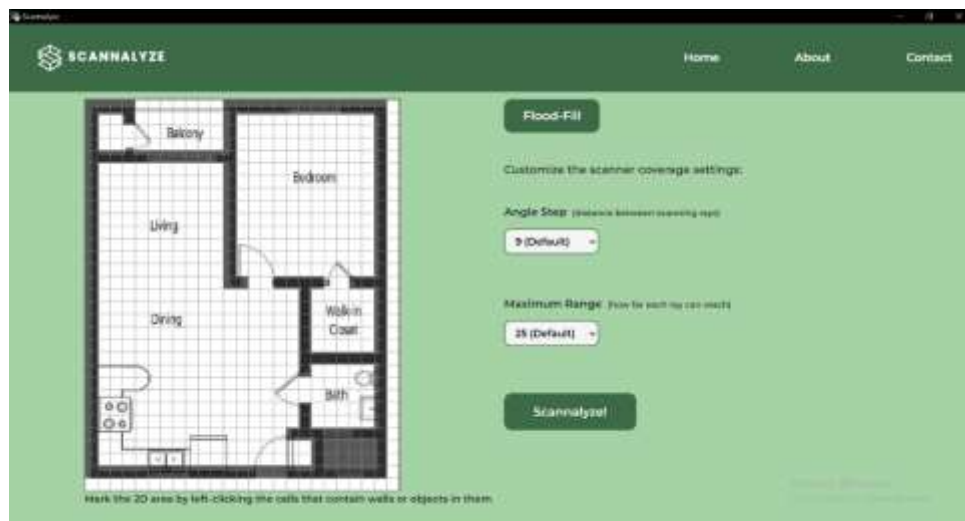
Πατώντας το κουμπί “Submit” ο χρήστης επιβεβαιώνει την εισαγωγή του διδιάστατου χώρου και οδηγείται στη σελίδα προεπεξεργασίας του χώρου πριν την τοποθέτηση σαρωτών (Εικόνα 10):



Εικόνα 10: Κάτοψη σε μορφή πλέγματος

Στο αριστερό μέρος αυτής της σελίδας έχει φορτωθεί η κάτοψη που εισήγαγε ο χρήστης με μορφή πλέγματος (grid). Στην αρχική έκδοση της εφαρμογής η ανάλυση του κάθε κελιού ήταν 30 x 30 pixels. Στην παρούσα έκδοση για βελτιωμένη ανάλυση κάθε κελί είναι 20 x 20 pixels. Θα ήταν ενδιαφέρον σε μελλοντική επέκταση της εφαρμογής το μέγεθος κελιού να παραμετροποιηθεί, ώστε να προσδιορίζεται από τον χρήστη.

Αφού η κάτοψη φορτωθεί σε μορφή πλέγματος, η εφαρμογή ζητά από τον χρήστη να μαρκάρει την κάτοψη επιλέγοντας με αριστερό κλικ τα κελιά που περιέχουν τοίχους ή αντικείμενα. Εάν ο χρήστης μαρκάρει άθελά του κάποιο κελί, τότε για να αναιρέσει την ενέργειά του μπορεί απλώς να ξαναπατήσει αριστερό κλικ. Το τελικό αποτέλεσμα παρουσιάζεται στην Εικόνα 11:



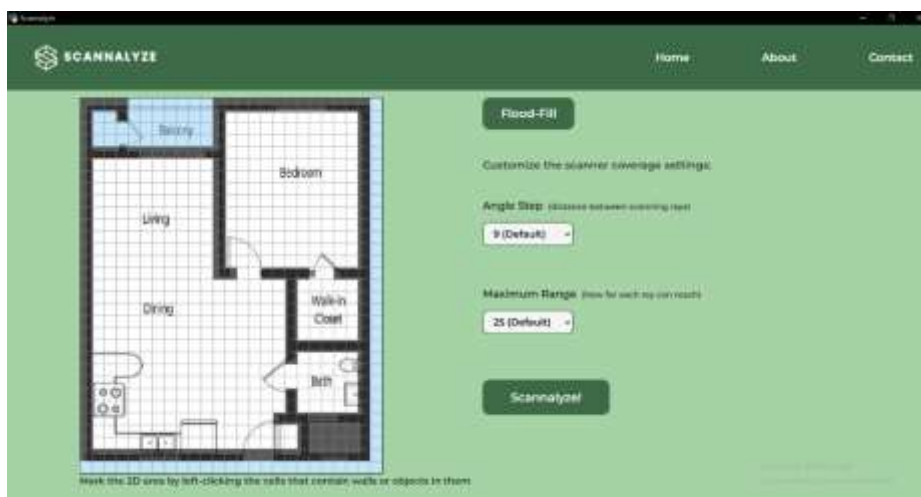
Εικόνα 11: Επισήμανση ορίων χώρου από χρήστη

Το μαρκάρισμα της κάτοψης είναι μια διαδικασία που εξαρτάται αποκλειστικά από τον χρήστη και από το πόσο αυστηρός ή επιεικής θέλει να είναι ο ίδιος με την επισήμανση του χώρου. Στο συγκεκριμένο παράδειγμα έγιναν οι εξής επιλογές/παραδοχές:

- δύο ελεύθερα κελιά για κάθε πόρτα,
- κανένας σαρωτής δεν θα τοποθετηθεί στο μπαλκόνι (πάνω αριστερά),
- κανένας σαρωτής δεν θα τοποθετηθεί στη μπανιέρα (κάτω δεξιά).

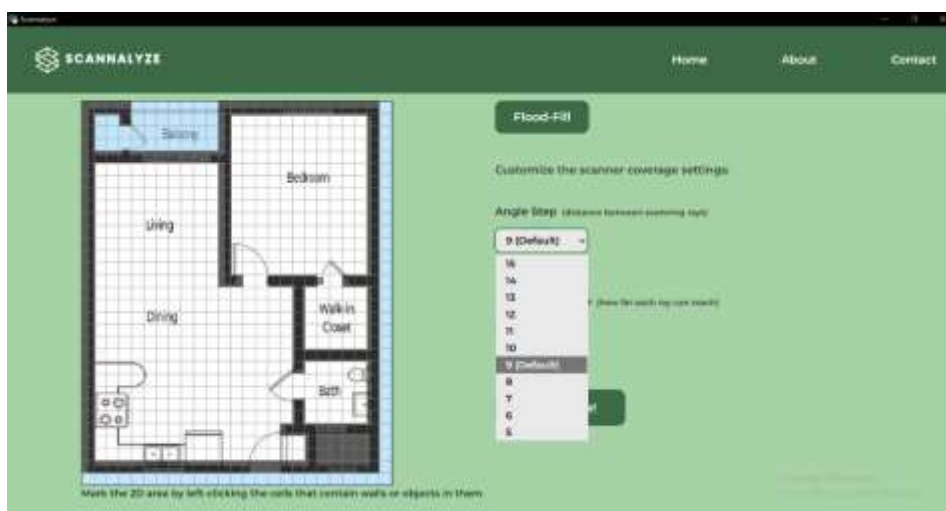
Φυσικά, η απαίτηση της εφαρμογής να επισημάνει ο χρήστης την κάτοψη είναι κοπιώδης και καλό θα ήταν να αποφευχθεί. Όμως, καθώς η έμφαση βρίσκεται στην δημιουργία μιας κατά το δυνατόν ολοκληρωμένης εφαρμογής, ενδεχομένως χωρίς τέλεια διεπαφή, αποφασίστηκε να μην αναλωθεί περισσότερος κόπος στην βελτίωση αυτού του ζητήματος, το οποίο σίγουρα αποτελεί στοιχείο για μελλοντική επέκταση.

Για κατόψεις όπως η παραπάνω, οι οποίες έχουν και περισσευούμενο εξωτερικό χώρο (τα άσπρα κελιά στην δεξιά και κάτω πλευρά και στην πάνω αριστερά πλευρά), δίνεται η δυνατότητα στον χρήστη να μαρκάρει και αυτόν τον χώρο, καθώς μπορεί να μην περιέχει τοίχους ή αντικείμενα, όμως είναι εκτός κάτοψης και δεν επιτρέπεται η τοποθέτηση σαρωτών. Επομένως, με το κουμπί “Flood-Fill” στο δεξί μέρος της οθόνης, η εφαρμογή αναγνωρίζει την κάτοψη που έχει μαρκάρει ο χρήστης και αυτόματα μαρκάρει με διαφορετικό χρώμα τον εξωτερικό χώρο (Εικόνα 12):



Εικόνα 12: Αυτόματη ανίχνευση εξωτερικού χώρου

Το επόμενο βήμα είναι να επιλέξει ο χρήστης τις παραμέτρους με τις οποίες επιθυμεί να εκτελεστούν οι αλγόριθμοι τοποθέτησης σαρωτών. Το Angle Step παριστάνει την απόσταση μεταξύ των ακτίνων που εκπέμπει κάθε σαρωτής, δηλαδή το πόσο πυκνές είναι (Εικόνα 13). Όπως είναι φυσικό, όσο πυκνότερες είναι οι ακτίνες κάθε σαρωτή (μικρές τιμές Angle Step) τόσο καλύτερη κάλυψη του χώρου αναμένουμε να πραγματοποιείται. Υποθέτουμε πανομοιότυπους σαρωτές, δηλαδή το Angle Step προσδιορίζεται από τον χρήστη με τιμή (ή αποκτά default τιμή) κοινή για όλους τους σαρωτές.



Εικόνα 13: Παραμετροποίηση Angle Step σαρωτών

Αντίστοιχα, το Maximum Range αντιπροσωπεύει την μέγιστη απόσταση (το μέγιστο πλήθος κελιών) που θα καλύπτουν οι ακτίνες του κάθε σαρωτή, δηλαδή την εμβέλειά του (Εικόνα 14). Διαισθητικά αναμένουμε ότι υψηλές τιμές της εμβέλειας κάθε σαρωτή

επιτυγχάνουν καλύτερη κάλυψη του χώρου. Και σε αυτήν την περίπτωση υποθέτουμε πανομοιότυπους σαρωτές, με κοινή τιμή για το Maximum Range. Ένας ευφυής χρήστης, για παράδειγμα, δεν θα επέλεγε τιμή μικρότερη από την μεγαλύτερη διάσταση του χώρου. Βέβαια, συχνά αναμένουμε ότι οι επιλογές για τις τιμές αυτές θα προσδιορίζονται από τον τύπο του διαθέσιμου σαρωτή.



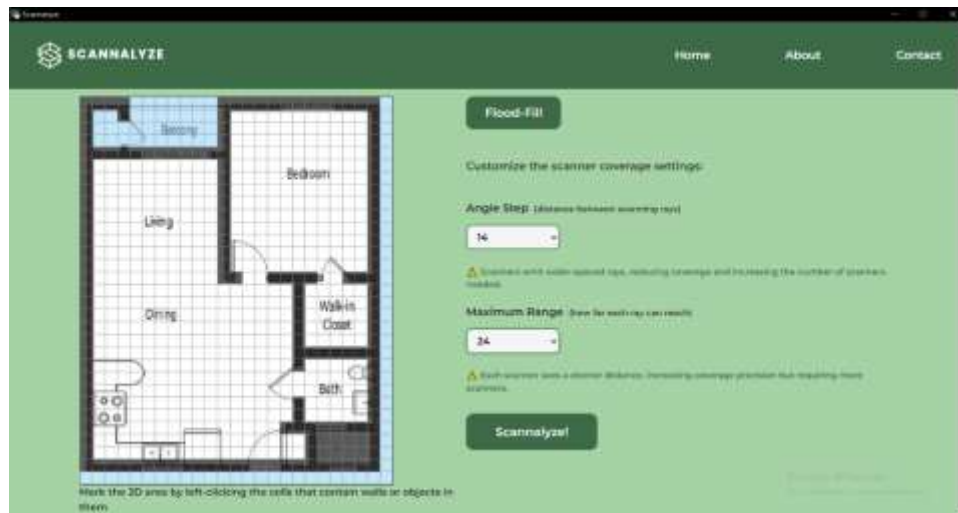
Εικόνα 14: Παραμετροποίηση Maximum Range σαρωτών

Ο συνδυασμός χαμηλής τιμής στο Angle Step και υψηλής τιμής στο Maximum Range δίνει βέλτιστα αποτελέσματα, καθώς οι ακτίνες γίνονται πολύ πυκνές και ανιχνεύουν μια μεγάλη απόσταση από κελιά, με αποτέλεσμα να καλύπτεται πλήρως ο χώρος με τον ελάχιστο δυνατό αριθμό σαρωτών (Εικόνα 15):



Εικόνα 15: Μήνυμα Scannalyze για καλές επιλογές παραμέτρων

Ο συνδυασμός υψηλής τιμής στο Angle Step και χαμηλής τιμής στο Maximum Range δεν εξασφαλίζει πλήρη κάλυψη του χώρου και παράλληλα αυξάνει τον αριθμό των σαρωτών (Εικόνα 16):



Εικόνα 16: Μήνυμα Scannalyze για κακές επιλογές παραμέτρων

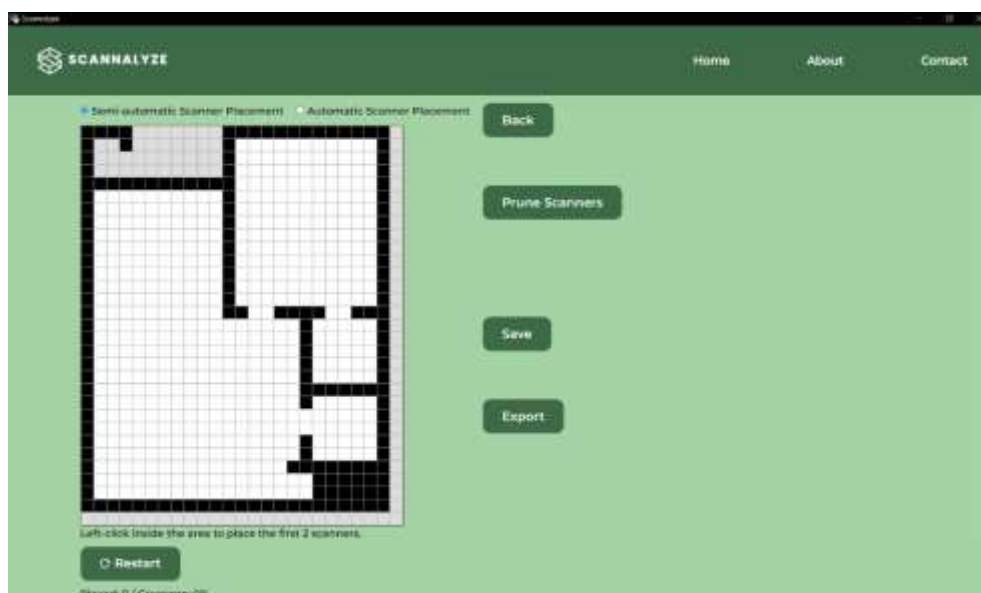
Επισημαίνεται ότι οι προειδοποιήσεις που βλέπει ο χρήστης στην Εικόνα ... δεν τον αποτρέπουν από το να προχωρήσει με την τοποθέτηση σαρωτών, καθώς μπορεί κάποιος χρήστης να επιθυμεί να εκτελέσει τον αλγόριθμο με αυτά τα δεδομένα για δικούς του λόγους. Ωστόσο, τον προειδοποιούν πως το αποτέλεσμα που θα λάβει δεν θα είναι το βέλτιστο δυνατό.

Πατώντας το κουμπί “Scannalyze!” ο χρήστης ενημερώνεται πως το στάδιο της παραμετροποίησης ολοκληρώθηκε επιτυχώς και είναι πλέον έτοιμος να προχωρήσει σε τοποθέτηση σαρωτών (Εικόνα 17):



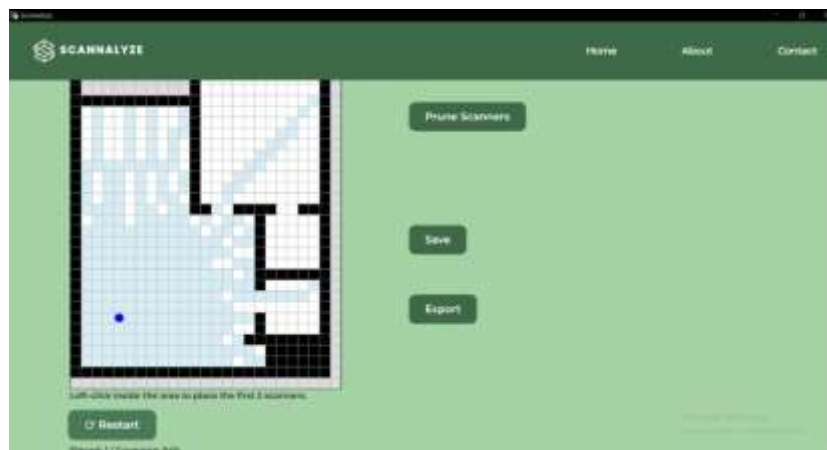
Εικόνα 17: Ολοκλήρωση παραμετροποίησης

Στο αριστερό μέρος της νέας σελίδας βρίσκεται πλέον η μαρκαρισμένη κάτοψη, η οποία είναι άδεια και δεν περιέχει προς το παρόν σαρωτές. Εδώ ο χρήστης έχει την επιλογή να ενεργοποιήσει αυτόματη (σενάριο 1) ή ημιαυτόματη (σενάριο 2) τοποθέτηση σαρωτών. Η προεπιλεγμένη μέθοδος είναι η ημιαυτόματη. Επιλέγοντας ημιαυτόματη τοποθέτηση σαρωτών, η εφαρμογή περιμένει τον χρήστη να τοποθετήσει τους πρώτους 2 σαρωτές όπου επιθυμεί εκείνος μέσα στο χώρο και ύστερα, λαμβάνοντας υπόψη αυτούς τους 2 τοποθετημένους σαρωτές, τοποθετεί αυτόματα όσους επιπλέον χρειάζονται για την πλήρη κάλυψη του χώρου (Εικόνα 18), εκτελώντας τους αλγορίθμους:



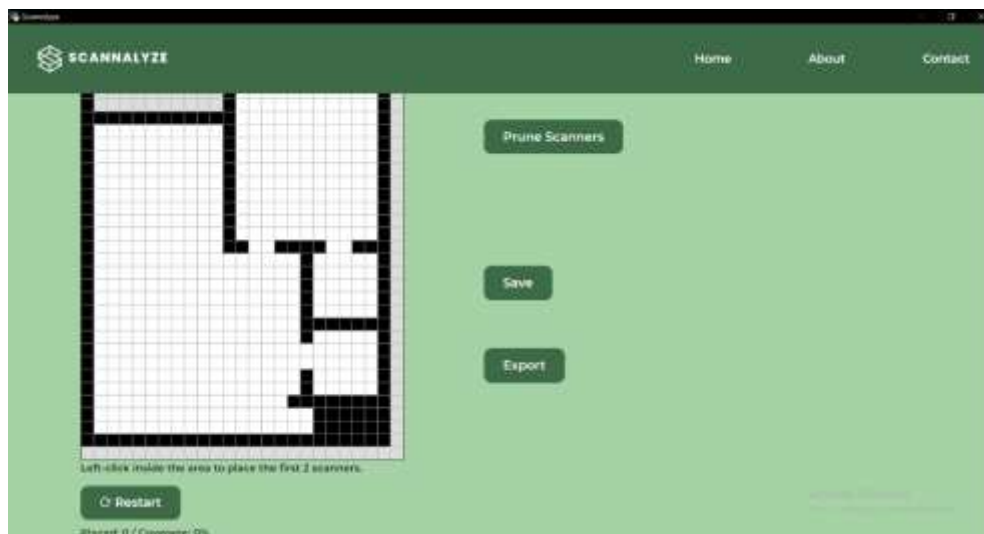
Εικόνα 18: Ημιαυτόματη τοποθέτηση σαρωτών

Ο χρήστης κάνει αριστερό κλικ μέσα στο κελί στο οποίο επιθυμεί να τοποθετήσει τον πρώτο σαρωτή (Εικόνα 19):



Εικόνα 19: Τοποθέτηση πρώτου σαρωτή από το χρήστη

Στην περίπτωση που δεν είναι ευχαριστημένος από την τοποθεσία του σαρωτή ή αν έκανε κάποιο λάθος κατά την τοποθέτηση, τότε μπορεί να αναιρέσει αυτήν την ενέργεια με το κουμπί “Restart” που βρίσκεται κάτω από την κάτοψη, το οποίο μόλις ενεργοποιηθεί αφαιρεί από τον χώρο όσους σαρωτές υπάρχουν (Εικόνα 20):



Εικόνα 20: Επιλογή επανατοποθέτησης

Ταυτόχρονα, μηδενίζονται ξανά οι δύο μετρητές που υπάρχουν κάτω αριστερά, δηλαδή αυτός που μετράει το πλήθος των συνολικά τοποθετημένων σαρωτών (Placed) και αυτός που μετράει το ποσοστό κάλυψης του χώρου (Coverage).

Ο χρήστης μπορεί να επανατοποθετήσει τους πρώτους δύο σαρωτές (κυκλωμένοι με κόκκινο) και να αφήσει τον αλγόριθμο να τοποθετήσει τους υπόλοιπους (Εικόνα 21):



Εικόνα 21: Αποτέλεσμα ημιαντόματης τοποθέτησης σαρωτών

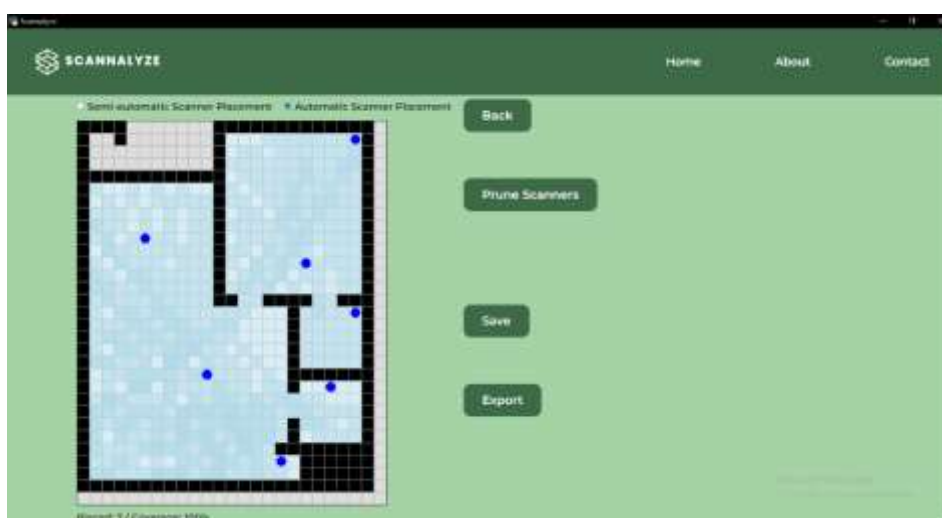
Στην παραπάνω περίπτωση χρειάστηκαν συνολικά 7 σαρωτές για την πλήρη κάλυψη του χώρου. Ωστόσο, με το κουμπί “Prune Scanners” που βρίσκεται στη δεξιά πλευρά της οθόνης, αυτός ο αριθμός σαρωτών ίσως να μπορεί να βελτιωθεί, δηλαδή εδώ δίνεται η δυνατότητα στον χρήστη να επεξεργαστεί εκ των υστέρων το αποτέλεσμα της εκτέλεσης των αλγορίθμων. Το συγκεκριμένο κουμπί όταν ενεργοποιείται ελέγχει εάν υπάρχουν πλεοναστικοί σαρωτές, οι οποίοι αν αφαιρούνταν δεν θα επηρεαζόταν η συνολική κάλυψη του χώρου. Εάν υπάρχουν τέτοιοι αχρείαστοι σαρωτές τους κλαδεύει (prune), και επιστρέφει τις συντεταγμένες τους, διαφορετικά ενημερώνει τον χρήστη πως δεν υπήρξαν πλεοναστικοί σαρωτές. Πράγματι, σε αυτό το παράδειγμα εκτέλεσης, ο ένας εκ των δύο σαρωτών που τοποθέτησε ο χρήστης ήταν αχρείαστος γι’ αυτό και η εφαρμογή των αφαιρέσε όπως φαίνεται στην Εικόνα 22 :



Εικόνα 22: Κλάδεμα σαρωτών μετά από ημιαντόματη τοποθέτηση

Επομένως, το τελικό πλήθος σαρωτών που χρειάστηκαν μετά από ημιαυτόματη τοποθέτηση και εκ των υστέρων επεξεργασία είναι 6.

Εκτός από την ημιαυτόματη τοποθέτηση, υπάρχει η δυνατότητα και εντελώς αυτόματης τοποθέτησης σαρωτών, χωρίς να χρειαστεί η συμβολή του χρήστη στην διαδικασία με την αρχική τοποθέτηση δύο σαρωτών. Για να ενεργοποιηθεί η αυτόματη τοποθέτηση θα πρέπει ο χρήστης να επιλέξει πάνω αριστερά το κουμπί “Automatic Scanner Placement” (Εικόνα 23):



Εικόνα 23: Αυτόματη τοποθέτηση σαρωτών

Με αυτήν την μέθοδο τοποθέτησης για τον ίδιο χώρο που παρουσιάστηκε νωρίτερα χρειάστηκαν συνολικά 7 σαρωτές. Όπως και προηγουμένως υπάρχει το ενδεχόμενο αυτοί οι σαρωτές να μειωθούν ή να παραμείνουν 7 αφού ενεργοποιηθεί το “Prune Scanners”, εφόσον ο χρήστης επιθυμεί εκ των υστέρων επεξεργασία του αποτελέσματος. Στην Εικόνα 24 παρατηρείται το εναλλακτικό σενάριο, κατά το οποίο κανείς σαρωτής δεν αφαιρέθηκε, καθώς όλοι τους κρίθηκαν απαραίτητοι:



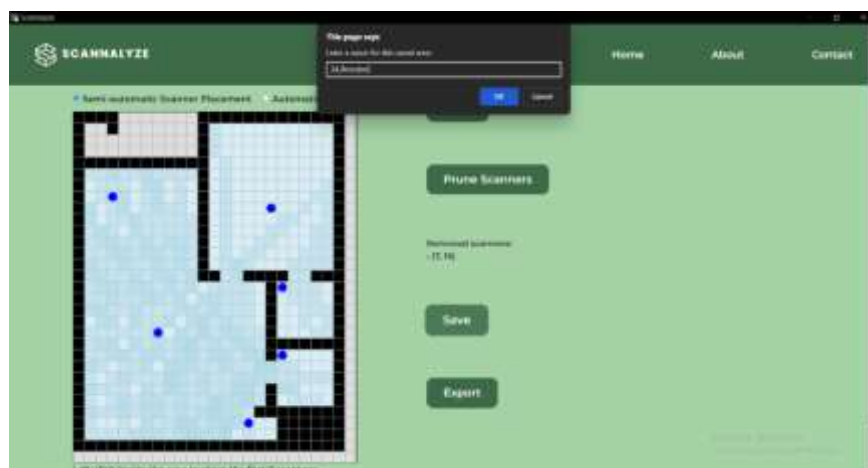
Εικόνα 24: Αποτέλεσμα αυτόματης τοποθέτησης μετά το κλάδεμα

Διαπιστώνουμε ότι για τον δεδομένο χώρο που χρησιμοποιείται σε αυτό το παράδειγμα εκτέλεσης, η ημιαυτόματη τοποθέτηση σαρωτών σε συνδυασμό με την εκ των υστέρων επεξεργασία επιτυγχάνει καλύτερο αποτέλεσμα (6 σαρωτές) σε σύγκριση με την αυτόματη τοποθέτηση σαρωτών σε συνδυασμό με την εκ των υστέρων επεξεργασία. Αναλυτικά αποτελέσματα από πειραματισμό με διάφορες κατόψεις και σχετική συζήτηση περιλαμβάνονται στην ενότητα 5.1.

Τα υπόλοιπα κουμπιά που υπάρχουν στη σελίδα είναι το “Back”, το “Save” και το “Export”.

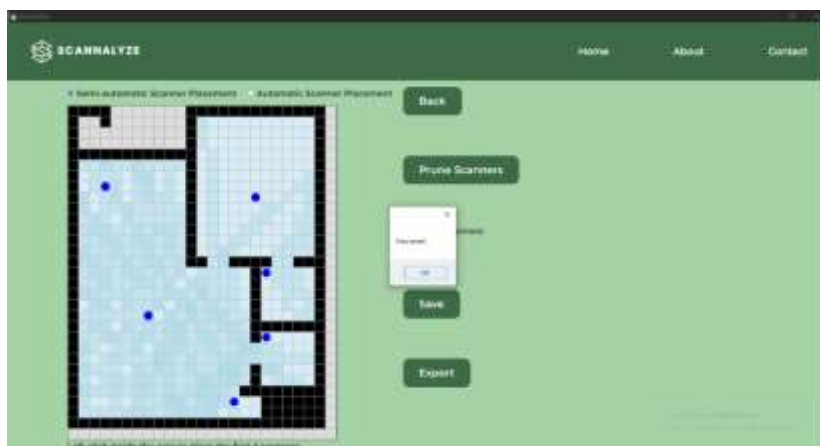
Το κουμπί “Back” ουσιαστικά επιστρέφει τον χρήστη στην προηγούμενη σελίδα προεπεξεργασίας της κάτοψης, σε περίπτωση που επιθυμεί να τρέξει τον αλγόριθμο με διαφορετικούς παραμέτρους ή εάν συνειδητοποίησε πως έκανε κάποιο λάθος κατά το μαρκάρισμα του χώρου.

Το κουμπί “Save” ζητάει πρώτα από τον χρήστη να δώσει ένα όνομα στην κάτοψη με τους τοποθετημένους σαρωτές (Εικόνα 25):



Εικόνα 25: Αποθήκευση 2Δ χώρου με σαρωτές

Και ύστερα τον ενημερώνει για την επιτυχή αποθήκευση (Εικόνα 26):

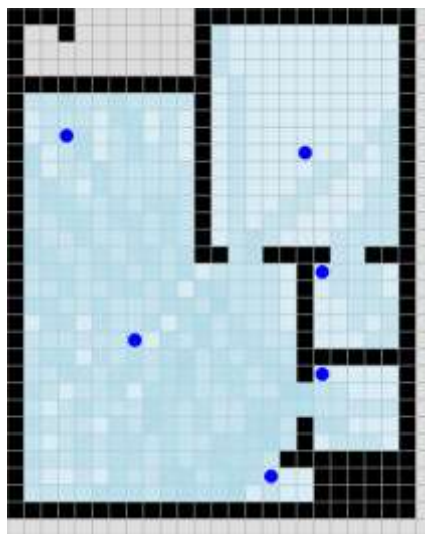


Εικόνα 26: Επιτυχής αποθήκευση

Αντίστοιχα, το κουμπί “Export” εξάγει την τελική κάτοψη με τους τοποθετημένους σαρωτές σαν εικόνα (αρχείο PNG) και ενημερώνει τον χρήστη (Εικόνα 27):

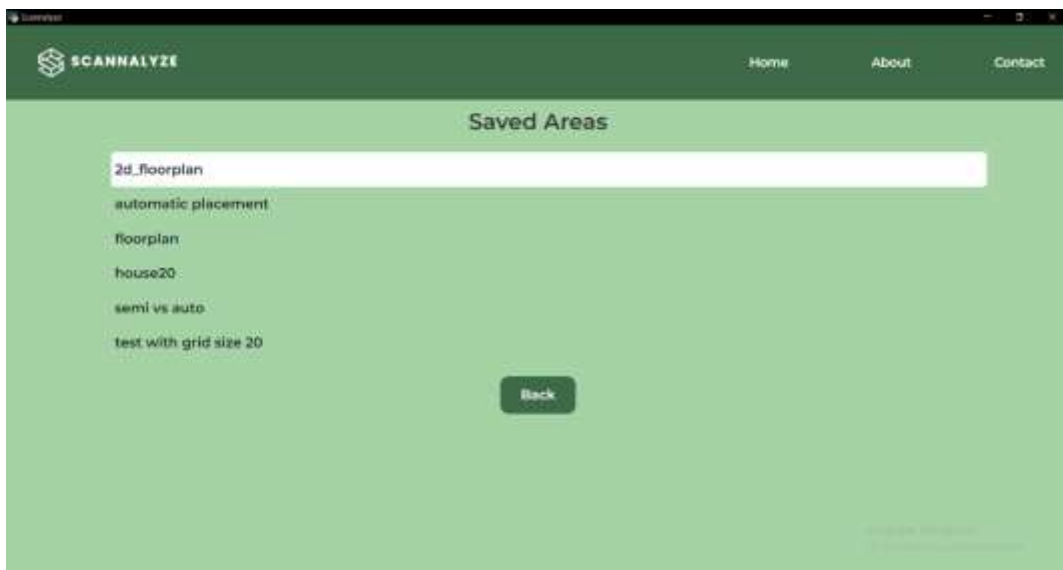


Εικόνα 27: Εξαγωγή αποτελέσματος ως εικόνα



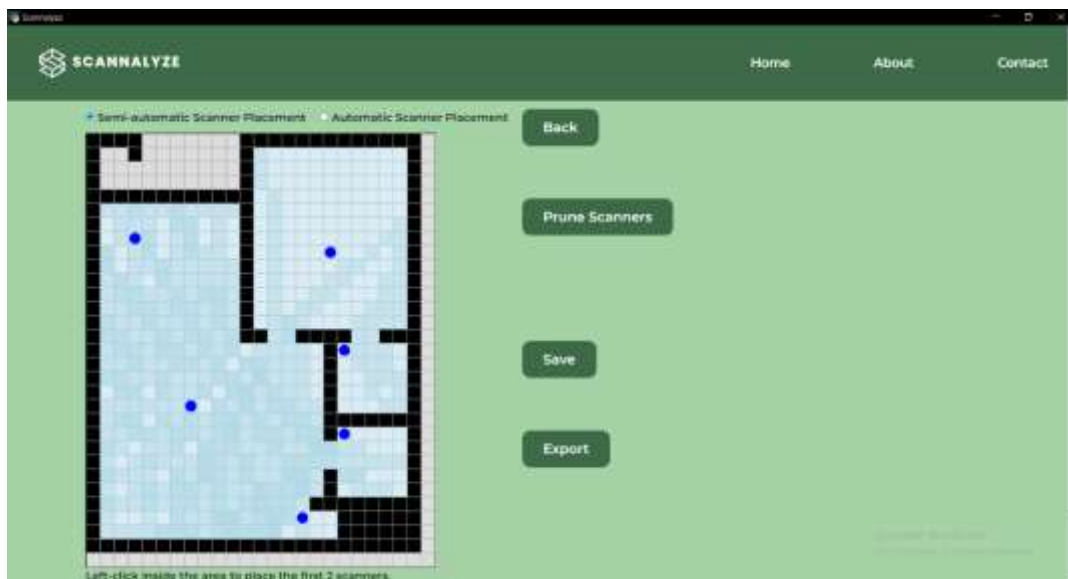
Εικόνα 28: Το αποτέλεσμα που εξήχθη

Πίσω στην αρχική οθόνη, επιλέγοντας το κουμπί “Saved Areas” ο χρήστης μπορεί να δει όλους τους χώρους (διδιάστατους και τρισδιάστατους) που έχει αποθηκεύσει με τους τοποθετημένους σαρωτές (Εικόνα 29):



Εικόνα 29: Σελίδα με αποθηκευμένους χώρους

Κάνοντας αριστερό κλικ σε έναν από αυτούς του επιτρέπεται να δει τους τοποθετημένους σαρωτές και εάν το επιθυμεί να κάνει επανατοποθέτηση ή περεταίρω επεξεργασία (Εικόνα 30):



Εικόνα 30: Επεξεργασία αποθηκευμένου χώρου

4.4 Επεξεργασία 3Δ χώρων

Μεταβαίνοντας από το διδιάστατο (2Δ) στο τρισδιάστατο (3Δ) επίπεδο, το πρόβλημα υφίσταται μια θεμελιώδη μεταβολή, καθώς η γεωμετρική και τοπολογική πολυπλοκότητα κλιμακώνεται εκθετικά. Ενώ στα πολυγωνικά σχήματα 2Δ η ορατότητα περιορίζεται από μονοδιάστατες ακμές, στον τρισδιάστατο χώρο τα πολύεδρα εισάγουν αλληλεπιδράσεις μεταξύ εδρών, ακμών και κορυφών που δημιουργούν σύνθετες "σκιές" και μη κυρτά οπτικά πεδία. Η μαθηματική απόδειξη ότι δεν υφίσταται πλέον ένα γραμμικό άνω όριο για τον αριθμό των φυλάκων ως προς τον αριθμό των κορυφών n , αναδεικνύει το πρόβλημα στις τρεις διαστάσεις ως ένα από τα πιο προκλητικά προβλήματα της Υπολογιστικής Γεωμετρίας. Στην ενότητα αυτή εξηγείται η αυξημένη πολυπλοκότητα που παρουσιάζει η επεξεργασία 3Δ χώρων και εισάγεται η έννοια της συνεισφοράς κάθε θέσης φύλαξης στην συνολική κάλυψη του χώρου, εμπνευσμένη από την θεωρία παιγνίων.

4.4.1 Η Κατάρρευση της Τριγωνοποίησης και το Πολύεδρο του Schönhardt

Το θεώρημα του Chvatal, που παρέχει ένα θεωρητικό μέγιστο για τον απαιτούμενο αριθμό φυλάκων προκειμένου να καλυφθεί ένα πολύγωνο, βασίζεται για την απόδειξή του, όπως είδαμε, στην τριγωνοποίηση. Κάθε απλό πολύγωνο P με n κορυφές επιδέχεται τριγωνοποίηση σε $(n-2)$ τρίγωνα. Στις τρεις διαστάσεις, η κατάσταση μεταβάλλεται άρδην. Όπως επισημαίνει ο O'Rourke (1987), ένα πολύεδρο δεν είναι πάντα δυνατόν να διαμεριστεί σε τετράεδρα χωρίς την προσθήκη επιπλέον κορυφών (Steiner points). Το κλασικό παράδειγμα είναι το Πολύεδρο του Schönhardt, το οποίο προκύπτει από τη στρέψη ενός τριγωνικού πρίσματος. Σε αυτή τη δομή, καμία εσωτερική διαγώνιος δεν μπορεί να συνδέσει δύο μη γειτονικές κορυφές χωρίς να τέμνει το όριο του πολυέδρου (Ruppert *et al.*, 1992).

Η αδυναμία αυτή σημαίνει ότι απαιτούνται αλγόριθμοι που χρησιμοποιούν πιο σύνθετες μεθόδους για την κάλυψη, όπως η αποσύνθεση σε κυρτά υποσύνολα, η οποία όμως είναι NP-hard για τρισδιάστατα σώματα (Chazelle, 1984).

4.4.2 Πολυπλοκότητα Ορατότητας και Ελικοειδείς Επιφάνειες

Στον διδιάστατο χώρο, το πολύγωνο ορατότητας ορίζεται από ευθύγραμμα τμήματα που διέρχονται από ακμές. Στον τρισδιάστατο χώρο η ορατότητα γίνεται μη γραμμική. Ενώ στις δύο διαστάσεις η ορατότητα από μια ακμή είναι πάντα ένα πολύγωνο, στις τρεις διαστάσεις η αλληλεπίδραση δύο λοξών ακμών δημιουργεί ελικοειδείς επιφάνειες («ruled

surfaces»). Αυτές οι επιφάνειες ορίζουν το όριο του ορατού όγκου, καθιστώντας τον ακριβή αναλυτικό υπολογισμό της ορατότητας εξαιρετικά επίπονο.

Ο Seidel απέδειξε ότι υπάρχουν πολύεδρα όπου ακόμη και αν τοποθετήσουμε έναν φύλακα σε κάθε κορυφή, ορισμένα σημεία στο εσωτερικό παραμένουν μη ορατά. Αυτό καταρρίπτει την ιδέα ότι ο αριθμός των φυλάκων συνδέεται απλά και γραμμικά με τον αριθμό των κορυφών n (O'Rourke, 1987). Έχει αποδειχθεί ότι για ορισμένες κλάσεις πολυέδρων (όπως τα ορθογώνια πολύεδρα), ο απαιτούμενος αριθμός φυλάκων είναι τουλάχιστον $\Omega(n^{\frac{3}{2}})$, ανάλογα με τη γεωμετρία (Urrutia, 2000).

Η έρευνα των Plantinga και Dyer (1987) ανέδειξε ότι ο αριθμός των τοπολογικά διαφορετικών όψεων ενός πολυέδρου μπορεί να φτάσει το $O(n^9)$ στην πλήρη του μορφή, αν και για κυρτά πολύεδρα περιορίζεται στο $O(n^2)$. Αυτή η εκθετική αύξηση της πληροφορίας καθιστά τις αναλυτικές λύσεις απαγορευτικές για εφαρμογές πραγματικού χρόνου.

Η διαισθητική προσέγγιση ότι περισσότερες κορυφές σημαίνουν αναλογικά περισσότερους φύλακες καταρρίπτεται. Στο 3Δ επίπεδο, μπορούμε να κατασκευάσουμε πολύεδρα όπου ο αριθμός των απαιτούμενων φυλάκων αυξάνεται τετραγωνικά ως προς n . Η ύπαρξη στενών «σχισμών» σε έναν τρισδιάστατο όγκο μπορεί να απομονώσει οπτικά τεράστιες περιοχές, απαιτώντας έναν φύλακα για κάθε μικροσκοπική δίοδο. Αυτό οδηγεί στο συμπέρασμα ότι η πολυπλοκότητα του προβλήματος στις τρεις διαστάσεις δεν εξαρτάται μόνο από τον αριθμό των κορυφών, αλλά κυρίως από την τοπολογική διάταξη των ακμών και την παρουσία οπτικών φραγμών που δεν υπάρχουν στις δύο διαστάσεις (Urrutia 2000). Συμπερασματικά, η μετάβαση από το διδιάστατο στο τρισδιάστατο επίπεδο δεν αποτελεί απλή γενίκευση, αλλά μια ριζική μεταβολή στην υπολογιστική φύση του προβλήματος. Ενώ στο 2Δ επίπεδο η ορατότητα είναι προβλέψιμη και περιορίζεται από τοπολογικά απλούς κανόνες, στο 3Δ επίπεδο το πρόβλημα καθίσταται συχνά υπολογιστικά μη επιλύσιμο με ακριβείς μεθόδους.

4.4.3 Υπολογιστική Πολυπλοκότητα και NP-Hardness

Στον τρισδιάστατο χώρο, το πρόβλημα κλιμακώνεται σε επίπεδα πολυπλοκότητας που υπερβαίνουν την απλή κατηγορία NP. Ενώ στις δυο διαστάσεις το πρόβλημα είναι NP-hard

(Yiakoumettis et al., 2013), πολλές παραλλαγές του προβλήματος ανήκουν στην κλάση 3R-complete (Abrahamsen et al., 2018). Αυτό σημαίνει ότι το πρόβλημα είναι το λιγότερο τόσο δύσκολο όσο το να αποφασιστεί αν ένα σύστημα πολυωνυμικών εξισώσεων με πραγματικούς συντελεστές έχει λύση, καθιστώντας την εύρεση ακριβούς βέλτιστης λύσης πρακτικά ανέφικτη για σύνθετα πολύεδρα. Λόγω της πολυπλοκότητας, το πρόβλημα στις τρεις διαστάσεις θεωρείται APX-hard, γεγονός που υποδηλώνει ότι δεν υπάρχει αλγόριθμος προσέγγισης με σταθερό λόγο σε πολυωνυμικό χρόνο, εκτός αν $P = NP$. Συνεπώς, η γεωμετρική πολυπλοκότητα των ελικοειδών επιφανειών που αναφέρθηκαν παραπάνω οδηγεί στην ανάγκη εγκατάλειψης της αναλυτικής γεωμετρίας υπέρ της διακριτοποίησης.

Μετατρέποντας το συνεχές χώρο σε ένα νέφος σημείων (point cloud), το πρόβλημα μετασχηματίζεται σε μια διακριτή παραλλαγή του Προβλήματος Κάλυψης Συνόλου (Set Cover Problem) (Alsadik et al., 2014). Αυτό επιτρέπει τη χρήση άπληστων αλγορίθμων που προσφέρουν μια προσέγγιση $O(\log n)$, η οποία είναι θεωρητικά η βέλτιστη δυνατή σε πολυωνυμικό χρόνο (Yiakoumettis et al., 2013).

Η χρήση νέφους σημείων επιτρέπει την ανάλυση ορατότητας σε πραγματικά περιβάλλοντα χωρίς την ανάγκη κατασκευής σύνθετων πολυεδρικών επιφανειών, κάτι που είναι υπολογιστικά δαπανηρό και ευάλωτο σε σφάλματα ακρίβειας (Alsadik et al., 2014). Έτσι οι $O(n^9)$ τοπολογικές όψεις αντικαθίστανται από m διακριτά σημεία. Χρησιμοποιείται Ray-Casting αντί για αναλυτικές τομές επιφανειών, και αξιοποιούνται δομές δεδομένων όπως τα KD-trees ή Octrees για ταχεία αναζήτηση γειτονίας και έλεγχο ορατότητας.

4.4.4 Αξιολόγηση Συνεισφοράς Θέσεων Φύλαξης μέσω Τιμής Shapley

Η τιμή Shapley αποτελεί μια θεμελιώδη μέθοδο κατανομής κερδών στη θεωρία συνεργατικών παιγνίων, η οποία εισήχθη από τον Lloyd Shapley το 1952. Στόχος της είναι ο δίκαιος επιμερισμός της συνολικής αξίας που παράγεται από μια ομάδα παικτών (συμμαχία), λαμβάνοντας υπόψη τη μέση οριακή συνεισφορά κάθε παίκτη σε όλες τις δυνατές υποσυμμαχίες.

Μαθηματικά, για ένα παίγνιο (N, v) , όπου N είναι το σύνολο των παικτών και v η χαρακτηριστική συνάρτηση που αποδίδει αξία σε κάθε υποσύνολο παικτών, η τιμή Shapley για τον παίκτη i ορίζεται ως:

$$\varphi_i(v) = \sum_{S \subseteq N - \{i\}} \frac{|S|! (n - |S| - 1)!}{n!} [v(S \cup \{i\}) - v(S)]$$

όπου

n : Το συνολικό πλήθος των παικτών.

S : Ένας συνασπισμός παικτών που δεν περιλαμβάνει τον i .

$v(S \cup \{i\}) - v(S)$: Η οριακή συνεισφορά του παίκτη i στον συνασπισμό S .

Στο πλαίσιο του προβλήματός μας, οι "παίκτες" αντιστοιχούν στις θέσεις των φυλάκων $G = \{g_1, g_2, \dots, g_k\}$ και η συνάρτηση αξίας $v(S)$ ορίζεται ως ο συνολικός όγκος (ή η επιφάνεια) του χώρου που καλύπτεται οπτικά από το σύνολο των φυλάκων S .

Η χρήση της τιμής Shapley επιτρέπει την ποσοτικοποίηση της "σημαντικότητας" κάθε θέσης, επιλύοντας το πρόβλημα της αλληλοκάλυψης. Αν δύο φύλακες καλύπτουν την ίδια περιοχή, η τιμή Shapley θα κατανείμει την αξία αυτής της περιοχής και στους δύο, αντί να την προσμετρήσει αυθαίρετα μόνο σε έναν. Αυτό είναι ιδιαίτερα κρίσιμο σε 3D περιβάλλοντα με πολύπλοκη γεωμετρία και τυφλά σημεία, όπου η στρατηγική τοποθέτηση είναι απαραίτητη για τη βελτιστοποίηση του κόστους.

Η αξιολόγηση των θέσεων φύλαξης μέσω της τιμής Shapley προσφέρει τρία βασικά πλεονεκτήματα:

- Δικαιοσύνη: Κάθε θέση φύλακα βαθμολογείται με βάση το πόσο αυξάνει την ορατότητα σε συνδυασμό με τους υπόλοιπους.
- Αναγνώριση Κρίσιμων Σημείων: Θέσεις που προσφέρουν μοναδική ορατότητα σε δυσπρόσιτα σημεία λαμβάνουν υψηλότερη τιμή Shapley.
- Βελτιστοποίηση Δικτύου θέσεων: Επιτρέπει την αφαίρεση πλεοναζόντων φυλάκων των οποίων η συνεισφορά στην κάλυψη, με βάση την τιμή Shapley, είναι οριακή ή μηδενική.

Στον αλγόριθμο PCBC, η ορατότητα δεν αντιμετωπίζεται ως συνεχής όγκος, αλλά ως ένα σύνολο διακριτών σημείων $P = \{p_1, p_2, \dots, p_m\}$. Κάθε υποψήφια θέση φύλακα g_j καλύπτει ένα υποσύνολο αυτών των σημείων $C_j \subseteq P$.

Σε αυτή την περίπτωση, η συνάρτηση $v(S)$ για έναν συνασπισμό φυλάκων S ορίζεται ως το πλήθος των μοναδικών σημείων που καλύπτονται:

$$v(S) = \left| \bigcup_{\{g_j \in S\}} C_j \right|$$

Η Τιμή Shapley αποκαλύπτει ποιοι φύλακες είναι "αναντικατάστατοι" και ποιοι είναι απλώς "βοηθητικοί". Αν ένα σημείο p_k καλύπτεται μόνο από τον φύλακα g_i τότε η συνεισφορά αυτού του σημείου προστίθεται εξολοκλήρου στην τιμή ϕ_i . Αν ένα σημείο καλύπτεται από k φύλακες, η αξία του σημείου μοιράζεται ισομερώς στους k φύλακες. Αν ένας φύλακας έχει πολύ χαμηλό ϕ_i (π.χ. $< 5\%$ του συνολικού), μπορεί να δικαιολογηθεί η αφαίρεσή του για λόγους κόστους, ακόμα και αν ο PCBC τον είχε αρχικά συμπεριλάβει.

Προκειμένου να διασφαλιστεί η υπολογιστική βιωσιμότητα του μοντέλου, αποφεύγεται ο προσδιορισμός των ακριβών τιμών Shapley, ο οποίος αποτελεί πρόβλημα υψηλής συνδυαστικής πολυπλοκότητας, και υιοθετείται αντ' αυτού μια εμπνευσμένη από την έννοια της τιμής Shapley μετρική που αποτιμά τη συνεισφορά κάθε θέσης φύλαξης στην συνολική κάλυψη του χώρου.

4.4.5 Επιστημάνσεις για την υλοποίηση

Για την υλοποίηση του τμήματος της εφαρμογής που αφορά την επεξεργασία χώρων 3Δ υιοθετήθηκε μια αυξητική προσέγγιση.

Ο αλγόριθμος PCBC αρχικά εφαρμόστηκε σε αρχεία .obj επειδή η μετατροπή αρχείων .dae και .fbx σε νέφος σημείων ήταν περίπλοκη. Το πρότυπο .obj αποτελεί το σημείο αναφοράς στη βιομηχανία για τον ορισμό της γεωμετρίας, καθώς περιγράφει ένα τρισδιάστατο αντικείμενο χρησιμοποιώντας κορυφές, αλλά το πιο σημαντικό είναι ότι συνδέει αυτές τις κορυφές για να σχηματίσει πολυγωνικές έδρες (faces), συνήθως τρίγωνα ή τετράπλευρα. Το κύριο μειονέκτημά του είναι ότι πρόκειται για μορφή κειμένου (ASCII), γεγονός που καθιστά το μέγεθος των αρχείων τεράστιο σε σύγκριση με σύγχρονα δυαδικά πρότυπα, όπως το .glb ή το .fbx.

Αργότερα, όπως περιγράφεται στην ενότητα 5.3, διενεργήθηκαν πειράματα και σε αρχεία .xyz και τα αποτελέσματα συγκρίθηκαν με αυτά που προέκυψαν από τα αρχεία .obj.

Δεδομένου ενός νέφους σημείων ο αλγόριθμος επιλέγει τυχαία 500 υποψήφιες θέσεις σάρωσης. Σε περίπλοκα μοντέλα αυτό το πλήθος σημείων είναι ενδεχομένως μικρό προκειμένου να έχουμε τέλεια κάλυψη αλλά η αύξηση (πχ διπλασιασμός ή τετραπλασιασμός του πλήθους των σημείων) καθιστά τον υπολογισμό πιο βαρύ και την εφαρμογή πιο αργή.

Ο χώρος στον οποίο εφαρμόζεται ο αλγόριθμος πρέπει να είναι στεγανοποιημένος, δηλαδή το τρισδιάστατο μοντέλο πρέπει να είναι κλειστό, χωρίς τρύπες, ασυνέχειες ή επιφάνειες που λείπουν. Αυτό είναι απαραίτητο για τους εξής λόγους:

- Διάκριση εσωτερικού/εξωτερικού χώρου: Για να υπολογίσουμε την κάλυψη, πρέπει να γνωρίζουμε ποια σημεία του νέφους βρίσκονται "μέσα" στον χώρο. Αν το μοντέλο δεν είναι στεγανοποιημένο, ο αλγόριθμος αποτυγχάνει να προσδιορίσει αν ένα σημείο ανήκει στον όγκο που πρέπει να καλυφθεί ή αν "διαρρέει" στο άπειρο.
- Ακρίβεια Ορατότητας: Η κάλυψη βασίζεται στην ορατότητα. Αν υπάρχουν ρωγμές στις επιφάνειες οι ακτίνες ορατότητας μπορεί να περάσουν μέσα από τοίχους που κανονικά θα έπρεπε να είναι αδιαφανείς. Αυτό οδηγεί σε ψευδώς θετικά αποτελέσματα, όπου ένας φύλακας φαίνεται να «βλέπει» σημεία σε διπλανά δωμάτια περνώντας μέσα από τις γεωμετρικές ασυνέχειες.
- Ομοιομορφία Δειγματοληψίας: Για να είναι το νέφος σημείων αντιπροσωπευτικό, τα σημεία πρέπει να κατανέμονται σε όλες τις επιφάνειες. Σε μη στεγανά μοντέλα, η παραγωγή του νέφους σημείων μπορεί να αφήσει κενά σε κρίσιμες γωνίες, καθιστώντας τον υπολογισμό της κάλυψης ελλιπή.

Για να διακρίνουμε το «μέσα» από το «έξω» σε μια επιφάνεια, χρησιμοποιούμε τα κάθετα διανύσματα (normals) των εδρών σε συνδυασμό με το εσωτερικό γινόμενο (dot product). Κάθε έδρα ενός αρχείου .obj έχει ένα προσανατολισμένο διάνυσμα που δείχνει προς την εξωτερική πλευρά. Αν θεωρήσουμε ένα διάνυσμα n που ξεκινά από ένα σημείο του χώρου προς την έδρα και το διάνυσμα της κάθετης n , το πρόσημο του εσωτερικού τους γινομένου $n \cdot n$ μας αποκαλύπτει τη σχετική τους θέση: αν το αποτέλεσμα είναι θετικό, το σημείο βρίσκεται στην εξωτερική πλευρά, ενώ αν είναι αρνητικό, βρίσκεται στην εσωτερική.

Επίσης, ιδανικά, απαιτείται το μοντέλο να μην περιλαμβάνει εξωτερικά αντικείμενα γιατί το νέφος σημείων που δημιουργείται θα περιλαμβάνει και σημεία για αυτά τα εξωτερικά αντικείμενα.

Αρχικά πειραματιστήκαμε χωρίς συγκεκριμένες υποθέσεις για το ύψος στο οποίο επιτρέπεται να τοποθετηθούν σαρωτές, αλλά στην τελευταία εκδοχή της υλοποίησης υιοθετήθηκε η υπόθεση ότι αυτοί τοποθετούνται σε ύψος 1.10 – 1.70 μέτρα, όπως συμβαίνει συνήθως στην πραγματικότητα. Προφανώς για μεγαλύτερους σε ύψος χώρους (όπως συχνά συμβαίνει σε δημόσια κτίρια) θα πρέπει να αναθεωρηθεί αυτή η υπόθεση.

Προκειμένου να τερματίζει σε εύλογο χρόνο η προσπάθεια εντοπισμού των κατάλληλων θέσεων φύλαξης προσδιορίζουμε ένα μέγιστο επιτρεπόμενο πλήθος σαρωτών (για παράδειγμα πειραματιστήκαμε με 8, 10, 15, 20 και σε ορισμένες περιπτώσεις 25 σαρωτές). Η εφαρμογή προσαρμόστηκε ώστε ο χρήστης να μπορεί να προσδιορίσει το επιθυμητό ποσοστό κάλυψης (το οποίο σπάνια επιτυγχάνεται) καθώς η εμπνευσμένη από τις τιμές Shapley αποτίμηση της τελικής τοποθέτησης των σαρωτών μπορεί να οδηγήσει τον χρήστη στο να προτιμήσει χαμηλότερο ποσοστό κάλυψης έναντι μικρότερου πλήθους θέσεων σάρωσης, δηλαδή σε μια αποτίμηση κόστους-οφέλους.

Για την καλύτερη οπτικοποίηση των αποτελεσμάτων παράγεται ένα heatmap κάλυψης, όπου κάθε σημείο του νέφους χρωματίζεται ανάλογα με την κάλυψή του από θέση σάρωσης (γαλάζιο, όταν καλύπτεται από έναν σαρωτή, μπλε, όταν καλύπτεται από δύο ή περισσότερους σαρωτές, και γκρι, όταν δεν καλύπτεται από κανέναν σαρωτή).

4.5 Παραδειγματική Εκτέλεση 3Δ χώρων

Κατά την εκκίνηση της εφαρμογής, όπως παρουσιάστηκε στην Εικόνα 7, ο χρήστης επιλέγει αυτή τη φορά «New Area 3D» και ακολουθεί το παράθυρο της εφαρμογής όπου καλείται να ανεβάσει αρχείο .obj, όπως φαίνεται στην Εικόνα 31



Εικόνα 31: Επιλογή αρχείου .obj

Για την παραδειγματική εκτέλεση που παρουσιάζουμε εδώ χρησιμοποιούμε το στεγανοποιημένο τρισδιάστατο μοντέλο που απεικονίζεται στην Εικόνα 32.



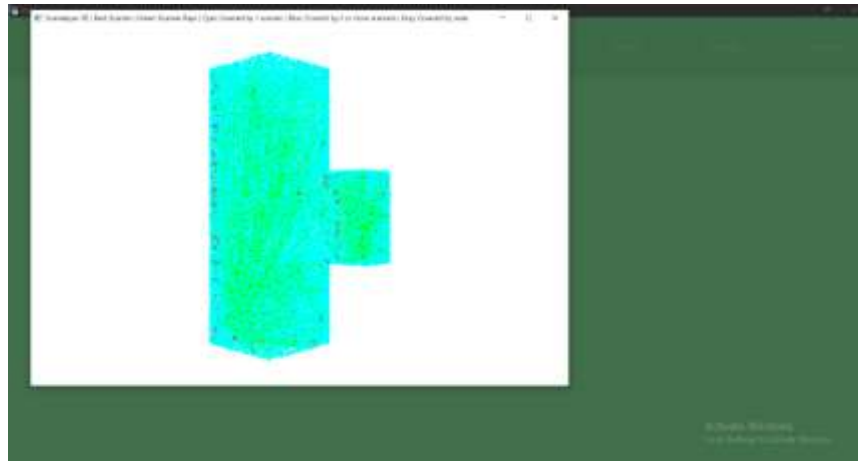
Εικόνα 32: Στεγανοποιημένο 3Δ μοντέλο

Στη συνέχεια ο χρήστης παραμετροποιεί τον αλγόριθμο προκειμένου να παραχθεί νέφος σημείων και να τοποθετηθούν οι σαρωτές. Οι τιμές των διαφόρων παραμέτρων πριν την εκτέλεση του PCBC προσδιορίζονται στην οθόνη της εφαρμογής που απεικονίζεται στην Εικόνα 33.



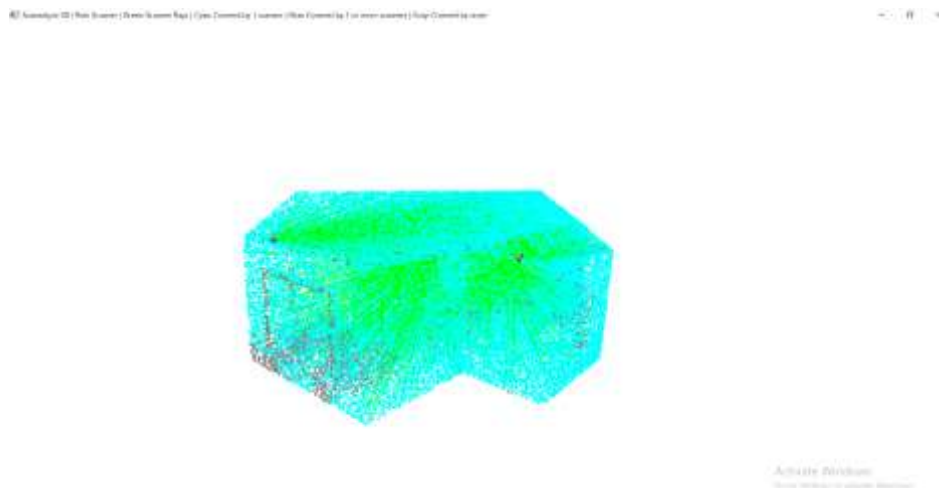
Εικόνα 33: Προσδιορισμός παραμέτρων και εκτέλεση PCBC

Στην παραδειγματική εκτέλεση που παρουσιάζεται εδώ επελέγη $\text{max range} = 10\text{m}$ και επιθυμητό ποσοστό κάλυψης 95%. Το μέγιστο πλήθος σαρωτών προσδιορίστηκε σε 15. Μετά από την εκτέλεση του PCBC το αποτέλεσμα που παράγεται με την τοποθέτηση των σαρωτών στο νέφος σημείων που αντιστοιχεί στο μοντέλο που δόθηκε ως είσοδος παρουσιάζεται στην Εικόνα 34.



Εικόνα 34: Τοποθέτηση σαρωτών στο νέφος σημείων

Μια εναλλακτική όψη του αποτελέσματος παρατίθεται στην Εικόνα 35. Με κόκκινο χρώμα εμφανίζονται οι σαρωτές (ή εναλλακτικά, οι θέσεις σάρωσης), με πράσινο χρώμα εμφανίζονται οι ακτίνες σάρωσης, με γαλάζιο χρώμα τα σημεία του νέφους που καλύπτονται από έναν σαρωτή, με μπλε τα σημεία του νέφους που καλύπτονται από δύο ή περισσότερους σαρωτές, και με γκρι χρώμα τα σημεία του νέφους που δεν καλύπτονται από κανέναν σαρωτή. Στην εκτέλεση που παρουσιάζεται εδώ επετεύχθη κάλυψη 91,95% με 5 σαρωτές.



Εικόνα 35: Εναλλακτική όψη του αποτελέσματος

Η αναφορά που παράγει η εφαρμογή για το πλήθος των σαρωτών που χρησιμοποιήθηκαν, την συνεισφορά του καθενός στην κάλυψη του χώρου, και το τελικό ποσοστό κάλυψης που επιτεύχθηκε παρουσιάζεται στην Εικόνα 36.

SHAPLEY-INSPIRED REPORT

```
=====
Scanner 1: 51.43% συνεισφορά στην κάλυψη του χώρου
Scanner 2: 18.36% συνεισφορά στην κάλυψη του χώρου
Scanner 3: 17.06% συνεισφορά στην κάλυψη του χώρου
Scanner 4: 3.02% συνεισφορά στην κάλυψη του χώρου
Scanner 5: 2.08% συνεισφορά στην κάλυψη του χώρου
```

Εικόνα 36: Παραγόμενη αναφορά τύπου Shapley

Παρατηρούμε ότι ο πρώτος σαρωτής επιτυγχάνει πάνω από την μισή κάλυψη του χώρου, στη συνέχεια ο δεύτερος και ο τρίτος συνεισφέρουν σημαντικά ποσοστά, και τέλος ο τέταρτος και ο πέμπτος συνεισφέρουν σχετικά λίγο στην κάλυψη του χώρου. Έτσι, θα μπορούσε ο χρήστης να εξετάσει κατά πόσο πράγματι επιθυμεί 95% κάλυψη του χώρου, όπως προσδιορίστηκε αρχικά, ή θα μπορούσε αυτό το ποσοστό να υποβαθμιστεί προκειμένου να μειωθεί ο συνολικός αριθμός των σαρωτών (ή αντίστοιχα των σαρώσεων).

5 Αποτελέσματα

5.1 Αποτελέσματα για 2Δ χώρους

Στην ενότητα αυτή συνοψίζουμε κάποια πειραματικά αποτελέσματα και συζητούμε τα ευρήματα.

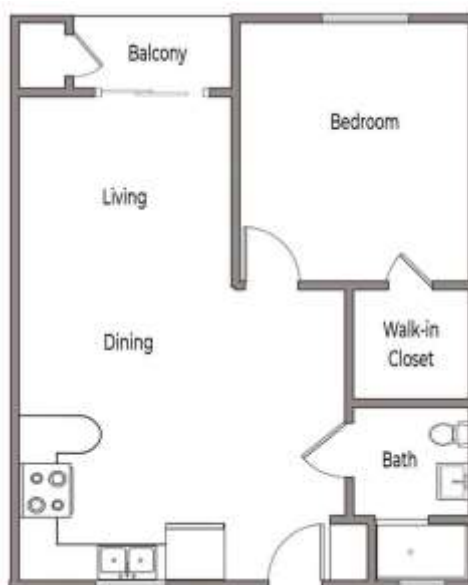
Η εφαρμογή εκτελέστηκε σε πλήθος κατοψεων, κάποιων πραγματικών που επελέγησαν τυχαία από το διαδίκτυο και κάποιων συνθετικών που κατασκευάστηκαν με το εργαλείο draw.io για να πειραματιστούμε με διαφορετικές διατάξεις χώρων (κλειστά πολύγωνα, πολύγωνα με τρύπες). Ο Πίνακας 1 παρουσιάζει ενδεικτικά κάποιους από τους χώρους με τους οποίους εκτελέσαμε πειράματα.

Πίνακας 1: Δείγματα κατοψεων για τα πειράματα σε 2Δ χώρους

Όνομα Κάτοψης

Κάτοψη Α

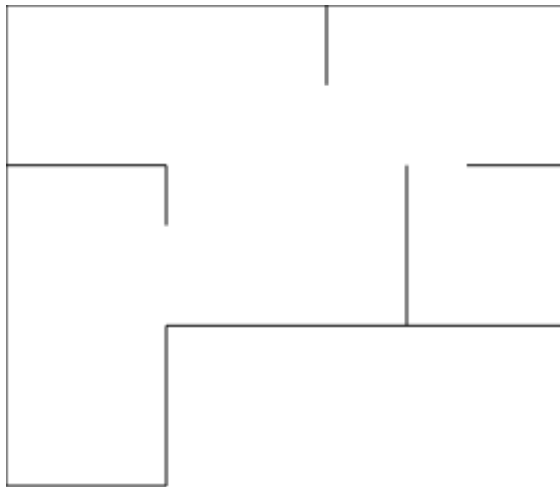
Κάτοψη



Κάτοψη Β



Κάτοψη Γ



Τα πειράματα εκτελέστηκαν για κάθε κάτοψη στις παρακάτω περιπτώσεις:

1. Αυτόματη τοποθέτηση σαρωτών
2. Αυτόματη τοποθέτηση σαρωτών και εκ των υστέρων επεξεργασία
3. Ημιαυτόματη τοποθέτηση σαρωτών
4. Ημιαυτόματη τοποθέτηση σαρωτών και εκ των υστέρων επεξεργασία

Σε όλα τα πειράματα χρησιμοποιήθηκαν βέλτιστες τιμές για τις παραμέτρους Angle Step (5) και Maximum Range (30).

Ο Πίνακας 2 συνοψίζει το πλήθος των σαρωτών που χρειάστηκαν για την κάλυψη κάθε χώρου για κάθε περίπτωση:

Πίνακας 2: Σύνοψη αποτελεσμάτων πειραμάτων

Κάτοψη	Περίπτωση 1	Περίπτωση 2	Περίπτωση 3	Περίπτωση 4
A	6	6	6	5
B	7	7	7	7
Γ	4	4	5	4

Από τα αποτελέσματα αυτά μπορούμε να εξάγουμε τα ακόλουθα συμπεράσματα:

1. Η περίπτωση 4, δηλαδή ο συνδυασμός ημιαυτόματης τοποθέτησης σαρωτών και της εκ των υστέρων επεξεργασίας τους απέδωσε για όλες τις κατόψεις τουλάχιστον το ίδιο καλό αποτέλεσμα.
2. Δεν διαφαίνεται να βελτιώνεται χάρη στην εκ των υστέρων επεξεργασία των σαρωτών το αποτέλεσμα της αυτόματης σάρωσης για όλες τις κατόψεις.
3. Η ημιαυτόματη τοποθέτηση σαρωτών, παρά την αναμενόμενη διαισθητικά χρησιμότητά της, για τις κατόψεις που εξετάστηκαν παραπάνω δεν παρήγαγε καλύτερα αποτελέσματα σε σύγκριση με την αυτόματη, και μάλιστα στην Κάτοψη Γ παρήγαγε χειρότερο αποτέλεσμα.
4. Σε ένα μεγάλο χώρο, όπως αυτός που αντιπροσωπεύει η Κάτοψη Β, δεν φαίνεται να μεταβάλλεται το αποτέλεσμα ανεξάρτητα από την περίπτωση που χρησιμοποιείται.

Τα αποτελέσματα αυτά δεν απομειώνουν την αξία της εφαρμογής, καθώς τελικά σε όλες τις περιπτώσεις επιτυγχάνεται πλήρης κάλυψη του κάθε χώρου. Απλώς, καταδεικνύεται ότι δεν είναι πάντα εφικτό να επιτευχθεί το θεωρητικό βέλτιστο, γεγονός που ισχύει για τα περισσότερα προβλήματα βελτιστοποίησης.

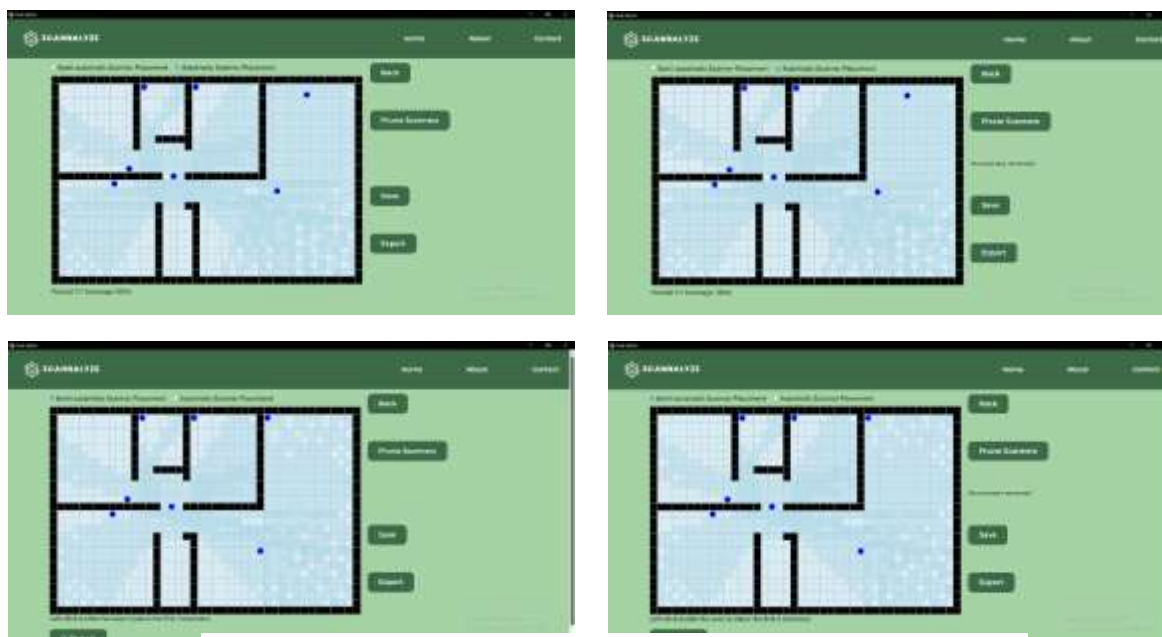
Στη συνέχεια, παρατίθενται τα στιγμιότυπα οθόνης των αποτελεσμάτων για κάθε κάτοψη. Στην Εικόνα 37 παρουσιάζονται όλες οι περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Α. Στο πάνω μέρος, αριστερά και δεξιά, διακρίνεται η αυτόματη τοποθέτηση και η αυτόματη τοποθέτηση με εκ των υστέρων επεξεργασία, αντίστοιχα, ενώ στο κάτω μέρος,

αριστερά και δεξιά διακρίνεται η ημιαυτόματη τοποθέτηση και η ημιαυτόματη τοποθέτηση με εκ των υστέρων επεξεργασία:



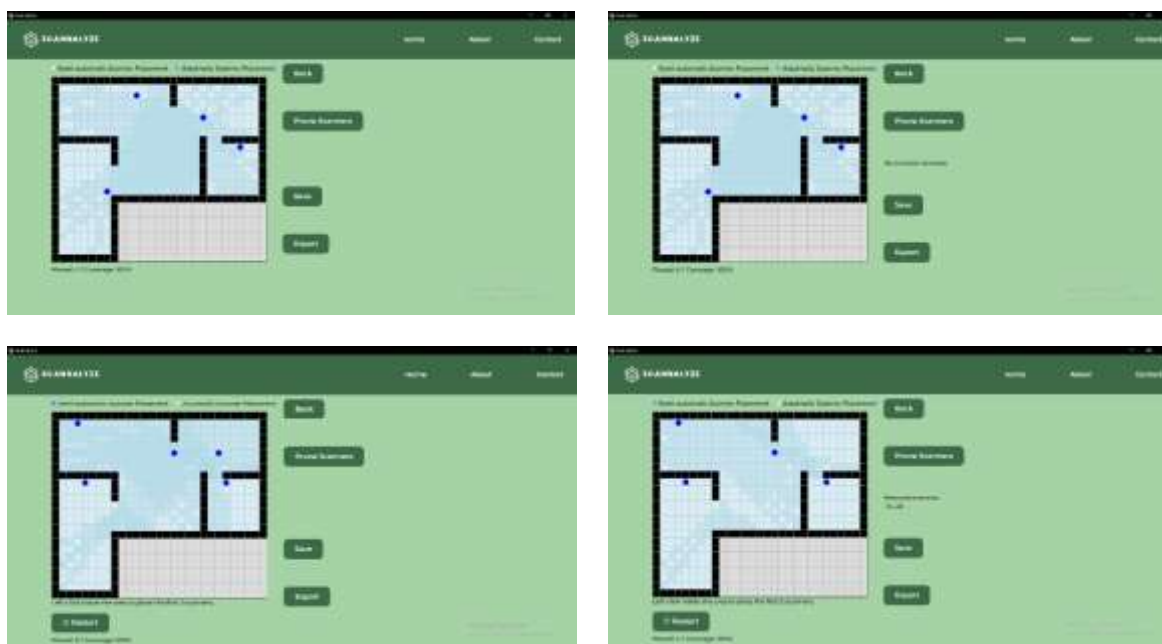
Εικόνα 37: Περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Α

Στην Εικόνα 38 παρουσιάζονται όλες οι περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Β. Στο πάνω μέρος, αριστερά και δεξιά, διακρίνεται η αυτόματη τοποθέτηση και η αυτόματη τοποθέτηση με εκ των υστέρων επεξεργασία, αντίστοιχα, ενώ στο κάτω μέρος, αριστερά και δεξιά διακρίνεται η ημιαυτόματη τοποθέτηση και η ημιαυτόματη τοποθέτηση με εκ των υστέρων επεξεργασία:



Εικόνα 38: Περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Β

Τέλος, στην Εικόνα 39 παρουσιάζονται όλες οι περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Γ. Στο πάνω μέρος, αριστερά και δεξιά, διακρίνεται η αυτόματη τοποθέτηση και η αυτόματη τοποθέτηση με εκ των υστέρων επεξεργασία, αντίστοιχα, ενώ στο κάτω μέρος, αριστερά και δεξιά διακρίνεται η ημιαυτόματη τοποθέτηση και η ημιαυτόματη τοποθέτηση με εκ των υστέρων επεξεργασία:



Εικόνα 39: Περιπτώσεις τοποθέτησης σαρωτών για την Κάτοψη Γ

Σημειώνεται ότι η διαφορά στον γαλάζιο χρωματισμό των κελιών υποδηλώνει το πλήθος των σαρωτών που καλύπτουν ένα κελί, δηλαδή έντονα χρωματισμένα κελιά καλύπτονται από περισσότερους από έναν σαρωτές.

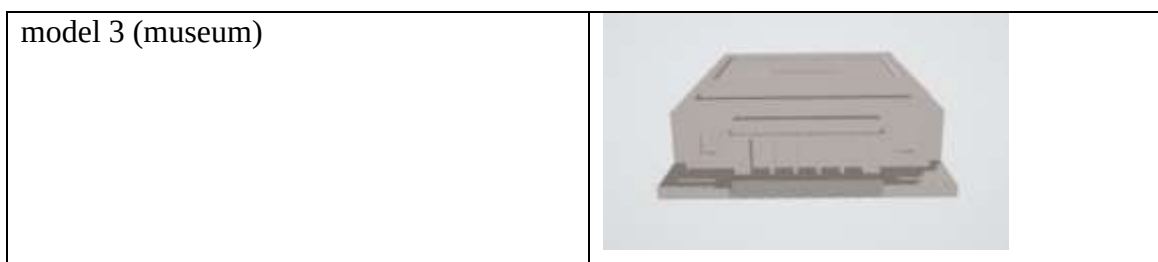
5.2 Αποτελέσματα για 3Δ χώρους

Στην ενότητα αυτή παρουσιάζουμε κάποια από τα βασικά πειράματα που εκτελέσαμε με χώρους 3Δ, αρχικά σε μοντέλα .obj. Γενικά, η εύρεση κατάλληλων μοντέλων για πειραματισμό ήταν μια δύσκολη διαδικασία, καθώς τα περισσότερα μοντέλα που διατίθενται ελεύθερα στο διαδίκτυο δεν είναι στεγανοποιημένα, ή είναι υπερβολικά μεγάλα ως αρχεία, ή αποτελούν εικόνες προσομοίωσης φυσικού περιβάλλοντος και περιέχουν άσχετη πληροφορία (ενδεικτικά, ανθρώπους, κήπους, τοπίο κλπ). Τα περισσότερα μοντέλα που χρησιμοποιήσαμε αντλήθηκαν από την ιστοσελίδα www.3dwarehouse.sketchup.com που περιλαμβάνει κυρίως collada (.dae) αρχεία. Αυτά μετατράπηκαν σε .obj χρησιμοποιώντας το www.imagetostl.com.

Ο Πίνακας 3 παρουσιάζει ενδεικτικά τρία μοντέλα που χρησιμοποιήθηκαν για τα πειράματα.

Πίνακας 3: Δείγματα μοντέλων για τα πειράματα σε 3Δ χώρους

Όνομα Μοντέλου	Μοντέλο
model 1 (house)	
model 2 (house)	



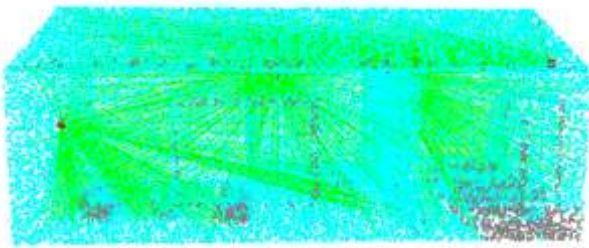
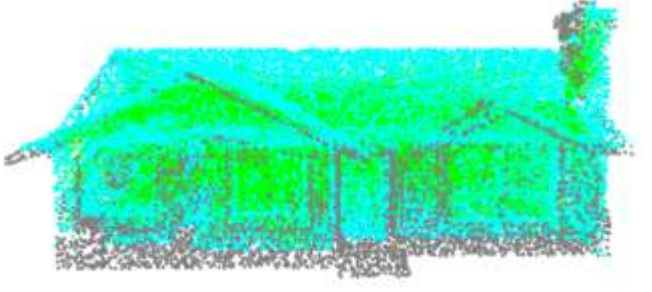
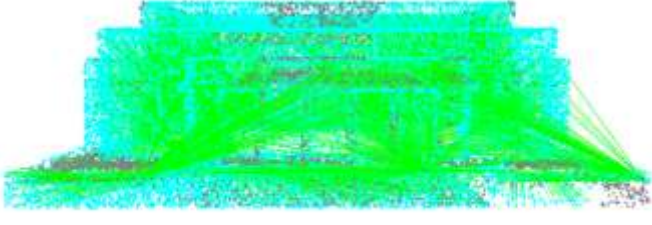
Σε όλα τα πειράματα χρησιμοποιήθηκαν οι ίδιες παράμετροι, μέγιστη εμβέλεια σαρωτή (Maximum Range) 12m, επιθυμητό ποσοστό κάλυψης 90%, και μέγιστο πλήθος σαρώσεων 15. Ο Πίνακας 4 συνοψίζει τα αποτελέσματα των πειραμάτων, δηλαδή το τελικό πλήθος σαρώσεων που χρειάστηκαν για να επιτευχθεί το ποσοστό κάλυψης που αναφέρεται και η αναφορά τύπου Shapley όπου αποτυπώνεται η συνεισφορά κάθε σαρωτή.

Πίνακας 4: Αποτελέσματα πειραμάτων για χώρους 3Δ

Μοντέλο	Τελικό Πλήθος Σαρώσεων	Ποσοστό Κάλυψης	Αναφορά τύπου Shapley
model 1	5	92.07%	SHAPLEY-INSPIRED REPORT for model.obj ***** Scanner 1: 51.2% συνεισφορά στην κάλυψη του χώρου Scanner 2: 18.57% συνεισφορά στην κάλυψη του χώρου Scanner 3: 16.99% συνεισφορά στην κάλυψη του χώρου Scanner 4: 3.82% συνεισφορά στην κάλυψη του χώρου Scanner 5: 2.29% συνεισφορά στην κάλυψη του χώρου
model 2	15	67.71%	SHAPLEY-INSPIRED REPORT for house.obj ***** Scanner 1: 25.93% συνεισφορά στην κάλυψη του χώρου Scanner 2: 18.46% συνεισφορά στην κάλυψη του χώρου Scanner 3: 3.73% συνεισφορά στην κάλυψη του χώρου Scanner 4: 3.58% συνεισφορά στην κάλυψη του χώρου Scanner 5: 3.53% συνεισφορά στην κάλυψη του χώρου Scanner 6: 2.87% συνεισφορά στην κάλυψη του χώρου Scanner 7: 2.51% συνεισφορά στην κάλυψη του χώρου Scanner 8: 2.87% συνεισφορά στην κάλυψη του χώρου Scanner 9: 2.63% συνεισφορά στην κάλυψη του χώρου Scanner 10: 2.21% συνεισφορά στην κάλυψη του χώρου Scanner 11: 1.76% συνεισφορά στην κάλυψη του χώρου Scanner 12: 1.68% συνεισφορά στην κάλυψη του χώρου Scanner 13: 1.42% συνεισφορά στην κάλυψη του χώρου Scanner 14: 1.2% συνεισφορά στην κάλυψη του χώρου Scanner 15: 1.31% συνεισφορά στην κάλυψη του χώρου
model 3	15	88.14%	SHAPLEY-INSPIRED REPORT for museum.obj ***** Scanner 1: 19.73% συνεισφορά στην κάλυψη του χώρου Scanner 2: 18.26% συνεισφορά στην κάλυψη του χώρου Scanner 3: 13.25% συνεισφορά στην κάλυψη του χώρου Scanner 4: 6.49% συνεισφορά στην κάλυψη του χώρου Scanner 5: 5.53% συνεισφορά στην κάλυψη του χώρου Scanner 6: 4.69% συνεισφορά στην κάλυψη του χώρου Scanner 7: 4.56% συνεισφορά στην κάλυψη του χώρου Scanner 8: 4.3% συνεισφορά στην κάλυψη του χώρου Scanner 9: 2.31% συνεισφορά στην κάλυψη του χώρου Scanner 10: 2.11% συνεισφορά στην κάλυψη του χώρου Scanner 11: 1.79% συνεισφορά στην κάλυψη του χώρου Scanner 12: 1.55% συνεισφορά στην κάλυψη του χώρου Scanner 13: 1.28% συνεισφορά στην κάλυψη του χώρου Scanner 14: 1.27% συνεισφορά στην κάλυψη του χώρου Scanner 15: 1.81% συνεισφορά στην κάλυψη του χώρου

Ο Πίνακας 5 παρουσιάζει τα νέφη σημείων με τους τοποθετημένους σαρωτές για κάθε πείραμα.

Πίνακας 5: Νέφη σημείων με σαρωτές για χώρους 3Δ

Μοντέλο	Νέφος Σημείων με σαρωτές
model 1	
model 2	
model 3	

Από τα αποτελέσματα αυτά μπορούμε να εξάγουμε τα ακόλουθα συμπεράσματα:

1. Σε ορισμένες περιπτώσεις, όπως στο μοντέλο 1, που έχει πιο «κυβική» μορφή, αρκούν πολύ λιγότερες από το μέγιστο πλήθος σαρώσεων, για να επιτευχθεί ικανοποιητικό σε σχέση με το επιθυμητό ποσοστό κάλυψης. Μάλιστα, όπως αναδεικνύει η αναφορά τύπου Shapley για το συγκεκριμένο μοντέλο, θα αρκούσαν και 4 σαρώσεις (επιτυγχάνοντας περίπου 89% έναντι του επιθυμητού 90%). Στις περιπτώσεις των μοντέλων 2 και 3 εξαντλήθηκε το μέγιστο πλήθος σαρώσεων και ενδεχομένως να έχουμε καλύτερα αποτελέσματα αυξάνοντας την μέγιστη εμβέλεια σαρωτών.
2. Το μοντέλο 3 αναδεικνύει την επισήμανση που έγινε νωρίτερα σχετικά με την καταλληλότητα των μοντέλων. Αντιστοιχεί σε ένα κτίριο μουσείου το οποίο

περιλαμβάνει εξωτερική σκάλα και πλατύσκαλο, με αποτέλεσμα κάποιοι σαρωτές να τοποθετούνται εξωτερικά του κτιρίου προκειμένου να καλύψουν και αυτά τα σημεία.

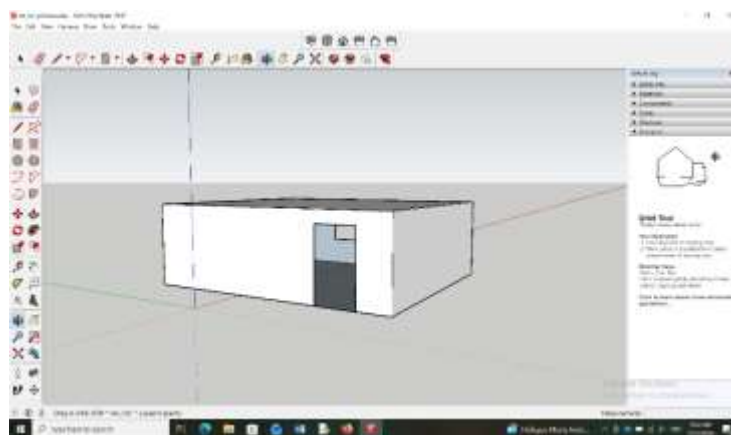
3. Το μοντέλο 2 που αντιστοιχεί σε πιο ρεαλιστικό σπίτι, με εσωτερικά δωμάτια, μοιάζει στη φιλοσοφία του με το μοντέλο 1 αλλά απαιτεί περισσότερους σαρωτές.

Προκειμένου να εξετάσουμε τη συμπεριφορά του αλγορίθμου PCBC όπως υλοποιήθηκε σε άλλα ρεαλιστικά σενάρια, πραγματοποιήθηκαν πρόσθετα πειράματα που περιγράφονται στην επόμενη ενότητα.

5.3 Πρόσθετα Πειράματα για 3Δ χώρους

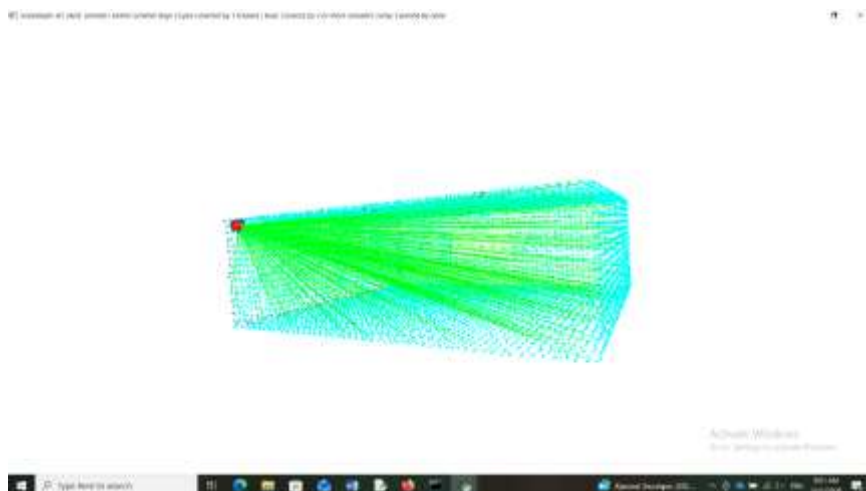
5.3.1 Πείραμα με στεγανοποιημένο μοντέλο με ανοιχτά παράθυρα και πόρτες

Σε αυτό το πείραμα εξετάσαμε τη συμπεριφορά του αλγορίθμου σε μοντέλο που έχει ανοιχτό παράθυρο και πόρτα. Ο χώρος που χρησιμοποιήθηκε απεικονίζεται στην Εικόνα 40 και σχεδιάστηκε με το εργαλείο SketchUp 2017.



Εικόνα 40: Χώρος με ανοιχτά παράθυρα/πόρτες

Επετεύχθη ποσοστό κάλυψης 98.53% με μόλις δύο σαρωτές, και ακόμα και με μέγιστη εμβέλεια σαρωτή τα 50 m οι ακτίνες σάρωσης παρέμειναν εντός του νέφους σημείων, όπως φαίνεται στην Εικόνα 41. Αυτό συμβαίνει επειδή ο αλγόριθμος PCBC αναζητά σημεία τα οποία θα χτυπήσουν οι ακτίνες σάρωσης, και αφού δεν υπάρχουν τέτοια σημεία στον εξωτερικό χώρο, οι ακτίνες δεν πρόκειται να ξεφύγουν εκτός του μοντέλου.

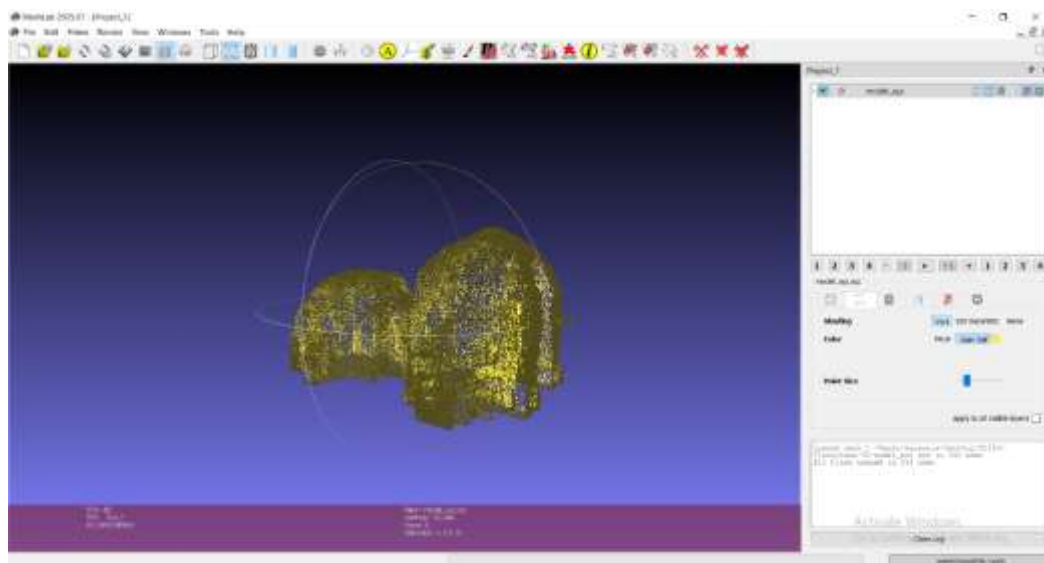


Εικόνα 41: Σάρωση σε χώρο με ανοιχτά παράθυρα/πόρτες

5.3.2 Πείραμα με πραγματικό χώρο με αρχείο .xyz

Σε αυτό το πείραμα εξετάσαμε τη συμπεριφορά του αλγορίθμου όταν χρησιμοποιείται .xyz αρχείο για το μοντέλο του χώρου, αντί για .obj. Τα αρχεία .xyz αποτελούν την απλούστερη μορφή ενός νέφους σημείων, καθώς στην πραγματικότητα είναι απλώς μια λίστα συντεταγμένων (x, y, z) στον τρισδιάστατο χώρο. Δεν περιλαμβάνεται πληροφορία εδρών (επιφανειών), παρά μόνο ένα νέφος από ασύνδετες κουκκίδες. Αποτελούν το πρότυπο για σαρώσεις LiDAR, αρχιτεκτονικές αποτυπώσεις και ανεπεξέργαστα δεδομένα αισθητήρων, με βασικό πλεονέκτημα ότι είναι ο πιο άμεσος τρόπος καταγραφής αυτού ακριβώς που «βλέπει» ένας σαρωτής λέιζερ σε έναν φυσικό χώρο.

Το .xyz μοντέλο που χρησιμοποιήσαμε αντιστοιχεί σε έναν ναό, όπως απεικονίζεται με τη βοήθεια του MeshLab στην Εικόνα 42. Αρχικά εκτελέστηκαν δύο πειράματα με το .xyz νέφος σημείων, με τις ακόλουθες παραμέτρους και αποτελέσματα:



Εικόνα 42: 3D μοντέλο ναού ως .xyz στο MeshLab

Στο πρώτο πείραμα, χρησιμοποιήθηκε Maximum Range = 5m, Target Coverage = 95% και μέγιστο πλήθος σαρωτών = 15, και επετεύχθη ποσοστό κάλυψης μόλις 55,26%, με την αναφορά τύπου Shapley που απεικονίζεται στην Εικόνα 43.

SHAPLEY-INSPIRED-REPORT	
Scanner-1:	32.94% συνεισφορά στην κάλυψη του χώρου
Scanner-2:	10.25% συνεισφορά στην κάλυψη του χώρου
Scanner-3:	4.18% συνεισφορά στην κάλυψη του χώρου
Scanner-4:	2.3% συνεισφορά στην κάλυψη του χώρου
Scanner-5:	1.31% συνεισφορά στην κάλυψη του χώρου
Scanner-6:	1.15% συνεισφορά στην κάλυψη του χώρου
Scanner-7:	0.66% συνεισφορά στην κάλυψη του χώρου
Scanner-8:	0.54% συνεισφορά στην κάλυψη του χώρου
Scanner-9:	0.45% συνεισφορά στην κάλυψη του χώρου
Scanner-10:	0.36% συνεισφορά στην κάλυψη του χώρου
Scanner-11:	0.33% συνεισφορά στην κάλυψη του χώρου
Scanner-12:	0.22% συνεισφορά στην κάλυψη του χώρου
Scanner-13:	0.22% συνεισφορά στην κάλυψη του χώρου
Scanner-14:	0.19% συνεισφορά στην κάλυψη του χώρου
Scanner-15:	0.17% συνεισφορά στην κάλυψη του χώρου

Εικόνα 43: Αποτελέσματα πείραμα 1 με .xyz

Ορισμένες ακτίνες περνούσαν μέσα από τοίχους, έβγαιναν στον εξωτερικό χώρο και ξαναπερνούσαν στο εσωτερικό μέσα από άλλους τοίχους. Αυτό συμβαίνει επειδή στα

αρχεία .xyz, ο τοίχος είναι απλώς μια “κουρτίνα” από κουκκίδες. Αν η ακτίνα περάσει από το κενό ανάμεσα σε δύο κουκκίδες, το Ray Casting δεν καταλαβαίνει ότι χτύπησε τοίχο και συνεχίζει στο επόμενο δωμάτιο. Η γεωμετρία του ναού είναι γεμάτη κολώνες και στενούς διαδρόμους και έτσι περιορίζεται η οριακή συνεισφορά κάθε σάρωσης. Με $\text{max_range} = 5\text{m}$, οι σαρωτές έχουν περιορισμένη “εμβέλεια” σε έναν χώρο που είναι αρκετά μεγάλος. Οι κολώνες δημιουργούν τεράστιες “σκιές”. Αν ένας σαρωτής μπει σε ένα σημείο, βλέπει μόνο ένα μικρό τμήμα του διαδρόμου πριν η επόμενη κολώνα του κόψει την ορατότητα.

Τροποποιήθηκε ελαφρώς ο αλγόριθμος PCBC ώστε να περιοριστεί ο αυθαίρετος προσανατολισμός των σαρωτών και να οριστεί ένα πιο αυστηρό όριο εμβέλειας για τις ακτίνες τους, προκειμένου να βελτιωθούν οι επιδόσεις, και διεξήχθησαν ξανά τα δύο πειράματα με τον τροποποιημένο κώδικα, με ίδιες τιμές παραμέτρων για max_range και επιθυμητό ποσοστό κάλυψης, και μέγιστο όριο σαρώσεων 20. Τα αποτελέσματα παρουσίασαν πτώση, συγκεκριμένα για το πρώτο πείραμα επετεύχθη κάλυψη 33,26% και για το δεύτερο 33,87%.

Η σημαντική μείωση του ποσοστού κάλυψης (από το αρχικό και για τα δύο πειράματα περίπου 55% στο νέο και για τα δύο πειράματα περίπου 33%) μετά την τροποποίηση του αλγορίθμου PCBC, φαίνεται ακατανόητη αλλά δεν υποδηλώνει μείωση της αποτελεσματικότητας, αντιθέτως υποδηλώνει αύξηση της ακρίβειας της μέτρησης.

Συγκεκριμένα, η διαφαινόμενη πτώση στο ποσοστό κάλυψης που επιτυγχάνεται παρά την τροποποίηση του αλγορίθμου οφείλεται στους εξής παράγοντες:

- Εξάλειψη των ψευδώς θετικών μετρήσεων: Στην αρχική μορφή του πειράματος, το υψηλότερο ποσοστό ήταν πλασματικό. Οι ακτίνες που διέφευγαν από τις “κουρτίνες” των .xyz σημείων κατέγραφαν ως “καλυμμένα” σημεία απομακρυσμένα δωμάτια ή εξωτερικούς χώρους, τα οποία στην πραγματικότητα ήταν αόρατα. Με την επιβολή αυστηρότερου ορίου εμβέλειας και περιορισμό του προσανατολισμού, ο αλγόριθμος έπαψε να προσμετρά αυτή την ορατότητα.
- Συσχετισμός εμβέλειας και γεωμετρίας: Η γεωμετρία του ναού, λόγω των στενών διαδρόμων και των κολόνων, απαιτεί πολύ μεγάλη πυκνότητα σαρώσεων αν η εμβέλεια της κάθε ακτίνας περιοριστεί. Όταν ορίστηκε το αυστηρό όριο εμβέλειας για να σταματήσει η διαρροή ακτινών ανάμεσα από κουκκίδες, οι σαρωτές έχασαν

τη δυνατότητα να "βλέπουν" μακριά μέσα στους διαδρόμους, με αποτέλεσμα να απαιτούνται πολύ περισσότερες από 20 σαρώσεις για να προσεγγίσουν το επιθυμητό 95%.

- Αντιστάθμιση ακρίβειας - ποσότητας: Η τροποποίηση ανέδειξε την πραγματική πρόκληση των .xyz αρχείων: για να έχουμε έγκυρα αποτελέσματα χωρίς διαρροές ακτίνων, θυσιάζεται η συνολική κάλυψη ανά σάρωση. Το χαμηλό 33% αντικατοπτρίζει την πραγματική ορατότητα εντός των ορίων του φυσικού εμποδίου, απαλλαγμένη από τα σφάλματα διαρροής των ακτίνων.

Μετατρέποντας το .xyz αρχείο σε .obj χρησιμοποιώντας το MeshLab και ξανατρέχοντας την αρχική εκδοχή της υλοποίησης για .obj αρχεία με παραμέτρους 10 m για την εμβέλεια των σαρωτών, επιθυμητό ποσοστό κάλυψης 90% και μέγιστο πλήθος σαρώσεων τις 15, επετεύχθη ποσοστό κάλυψης 38.27% για τον ναό, που είναι λογικό δεδομένου του μεγάλου όγκου του ναού, ενώ με εμβέλεια σαρωτών τα 25 m επετεύχθη ποσοστό κάλυψης σημαντικά καλύτερο, 67,48%.

6 Συμπεράσματα και Μελλοντικές Επεκτάσεις

Στο Κεφάλαιο αυτό γίνεται μια αποτίμηση των επιτευγμάτων της Πτυχιακής Εργασίας, με σημείο αναφοράς τους αρχικούς στόχους που τέθηκαν, και προτείνονται κατευθύνσεις για μελλοντική έρευνα και επέκταση της εφαρμογής που αναπτύχθηκε.

Στόχος της Πτυχιακής Εργασίας, όπως είχε τεθεί στην Εισαγωγή, ήταν η σαφής και οπτικοποιημένη απεικόνιση του ζητήματος της ορατότητας, η αξιολόγηση και σύγκριση των μεθόδων κάλυψης σε διδιάστατους και τρισδιάστατους χώρους όσον αφορά την αποτελεσματικότητα και την υπολογιστική πολυπλοκότητά τους, και η ανάπτυξη μιας λειτουργικής εφαρμογής που θα επιτρέπει τον πειραματισμό με διάφορες γεωμετρίες και παραμέτρους.

Στην παρούσα Πτυχιακή Εργασία αναπτύχθηκε η εφαρμογή Scannalyze, προκειμένου να βελτιστοποιηθεί η τοποθέτηση αισθητήρων LiDAR για την αποτύπωση τρισδιάστατων χώρων. Ως αφετηρία του προβλήματος και για την καλύτερη διαισθητική κατανόησή του, χρησιμοποιήθηκε το πρόβλημα της γκαλερί έργων τέχνης που πρωτοδιατυπώθηκε από τον Klee το 1973, και αφορούσε την εύρεση βέλτιστων θέσεων για την οπτική κάλυψη των σημείων κλειστών πολυγώνων, δηλαδή στις δύο διαστάσεις. Επομένως, η εφαρμογή αρχικά αναπτύχθηκε για πειραματισμό με τις δύο διαστάσεις, και με στόχο να επιτυγχάνεται ικανοποιητική (αν όχι βέλτιστη) τοποθέτηση αισθητήρων, δεδομένων κατόψεων. Οι αλγόριθμοι που υλοποιήθηκαν για την τοποθέτηση σαρωτών σε χώρους δύο διαστάσεων είναι οι Ray Tracing και Next-Best-View, και εκτελέστηκαν πειράματα με δύο γενικά σενάρια:

Στο πρώτο σενάριο οι αλγόριθμοι εκτελούνται χωρίς καμία παρέμβαση του χρήστη, με αφετηρία την κάτοψη του διδιάστατου χώρου. Στο δεύτερο σενάριο, υπάρχει η δυνατότητα παρέμβασης είτε κατά το στάδιο εκκίνησης της εφαρμογής των αλγορίθμων, όταν ο χρήστης τοποθετεί αρχικά δύο σαρωτές, είτε/και κατά την ολοκλήρωση της τοποθέτησης των σαρωτών από τους αλγορίθμους, οπότε υπάρχει η δυνατότητα εκ των υστέρων επεξεργασίας του αποτελέσματος, με σκοπό ενδεχόμενο κλάδεμα του πλήθους των σαρωτών για να επιτευχθεί βέλτιστο (κατά το δυνατόν) πλήθος, χωρίς να θυσιάζεται το ποσοστό κάλυψης.

Εκτελέστηκαν και πειράματα με συνδυασμούς αυτών των βασικών σεναρίων και παρουσιάστηκαν αναλυτικά τα αποτελέσματά τους στο Κεφάλαιο 5. Το ευρύτερο συμπέρασμα που εξήχθη από τον πειραματισμό στις δύο διαστάσεις είναι ότι, γενικά, είναι δύσκολο να επιτευχθεί το θεωρητικό βέλτιστο πλήθος σημείων σάρωσης για πολύγωνα n κορυφών, που έχει προσδιοριστεί στη βιβλιογραφία ως $\lfloor n/3 \rfloor$. Ειδικότερα, με βάση τα πειράματα που διεξήχθησαν, διαφαίνεται ότι ο συνδυασμός της ανθρώπινης διαίσθησης (όπου ο χρήστης τοποθετεί τους πρώτους δύο σαρωτές) με την εκ των υστέρων επεξεργασία για το κλάδεμα του πλήθους των σαρωτών, απέδωσε καλύτερα από την πλήρως αυτόματη διαδικασία. Μάλιστα το κλάδεμα είναι κρίσιμο για να προσεγγιστεί το θεωρητικό βέλτιστο του Chvátal, καθώς εξαλείφει τις επικαλύψεις που δημιουργούν οι άπληστοι αλγόριθμοι.

Κατά την ανάπτυξη αυτού του τμήματος της εφαρμογής, ο χρήστης πρέπει να επισημάνει χειροκίνητα το περίγραμμα του διδιάστατου χώρου που επιθυμεί να καλυφθεί, και αυτό δεν είναι βέλτιστο. Μια μελλοντική επέκταση θα μπορούσε να ασχοληθεί με την αυτόματη αναγνώριση τοίχων. Επίσης, υποτέθηκαν πανομοιότυποι σταθεροί σαρωτές, καθώς για την αποτύπωση του χώρου δύο διαστάσεων μας ενδιαφέρει ουσιαστικά το πλήθος των θέσεων στους οποίους θα τοποθετηθεί ένας, στην πραγματικότητα, σαρωτής. Για το ευρύτερο πρόβλημα της παρακολούθησης ενός χώρου, για παράδειγμα σε εφαρμογές Ρομποτικής στην επιθεώρηση χώρων ή την οργάνωση υποδομών ασφάλειας και διάσωσης, μια προφανής κατεύθυνση για μελλοντική επέκταση της μελέτης θα ήταν ο πειραματισμός με διαφορετικών δυνατοτήτων σαρωτές, και ενδεχομένως με τη δυνατότητα να μεταβάλουν την θέση τους γύρω από κάποια σημεία του χώρου (κινητοί). Επίσης, στην τρέχουσα εκδοχή της εφαρμογής, ο χρήστης δεν έχει την δυνατότητα να υποδείξει θέσεις στις οποίες απαγορεύεται να τοποθετηθούν σαρωτές (για παράδειγμα επειδή υπάρχουν εμπόδια ή αποτελούν αφύσικες επιλογές), και αυτή είναι μια επιπλέον υπόθεση που θα μπορούσε να αναθεωρηθεί σε μελλοντικό στάδιο. Στην τρέχουσα εκδοχή της εφαρμογής το μέγεθος κελιού για την απεικόνιση των διδιάστατων χώρων με μορφή πλέγματος είναι σταθερό, και μελλοντικά αυτό θα μπορούσε να παραμετροποιηθεί, ώστε να υπάρχει η δυνατότητα ο χρήστης να πειραματίζεται για την ίδια κάτοψη με διαφορετικές επιλογές προκειμένου, ενδεχομένως, να προσεγγίσει περισσότερο το θεωρητικό βέλτιστο. Τέλος, η εφαρμογή θα μπορούσε να επεκταθεί με περισσότερους αλγόριθμους για τις δύο διαστάσεις, και ο χρήστης θα μπορούσε να έχει τη δυνατότητα επιλογής του αλγορίθμου που θα εφαρμοστεί.

Κατά την δεύτερη φάση ανάπτυξης της εφαρμογής ο στόχος της Πτυχιακής Εργασίας ήταν να διερευνηθεί η εφαρμογή των αλγορίθμων σε πιο αληθοφανή περιβάλλοντα, όπως περιοχές με όγκο ή επιφάνειες πολλαπλών επιπέδων, και επομένως σε χώρους τριών διαστάσεων. Στην ενότητα 4.4 αναπτύχθηκαν εκτενώς οι λόγοι για τους οποίους η μετάβαση από τις δύο στις τρεις διαστάσεις κλιμακώνει την πολυπλοκότητα του προβλήματος. Για να αποφευχθεί η ανάγκη διακριτοποίησης χώρων προτιμήθηκε η εφαρμογή του αλγορίθμου Point Cloud Based Coverage, που μοντελοποιεί τους τρισδιάστατους χώρους ως νέφη σημείων. Κατά την παραγωγή των αποτελεσμάτων υιοθετήθηκε μια προσέγγιση εμπνευσμένη από τη Θεωρία Παιγνίων, ώστε να αποτυπώνεται η συνεισφορά κάθε θέσης σάρωσης στη συνολική κάλυψη του τρισδιάστατου χώρου, με μια τύπου Shapley τιμή.

Η εύρεση κατάλληλων μοντέλων για πειραματισμό ήταν μια δύσκολη διαδικασία, καθώς τα περισσότερα μοντέλα που διατίθενται ελεύθερα στο διαδίκτυο δεν είναι στεγανοποιημένα, ή είναι υπερβολικά μεγάλα ως αρχεία, ή αποτελούν εικόνες προσομοίωσης φυσικού περιβάλλοντος και περιέχουν άσχετη πληροφορία (ενδεικτικά, ανθρώπους, κήπους, τοπίο κλπ). Στην τρέχουσα εκδοχή της εφαρμογής ο αλγόριθμος επιλέγει τυχαία 500 υποψήφιες θέσεις σάρωσης. Σε περίπλοκα μοντέλα αυτό το πλήθος σημείων είναι ενδεχομένως μικρό, προκειμένου να έχουμε τέλεια κάλυψη, αλλά η αύξηση (π.χ. διπλασιασμός ή τετραπλασιασμός του πλήθους των σημείων) καθιστά τον υπολογισμό πιο βαρύ και την εφαρμογή πιο αργή.

Αρχικά πειραματιστήκαμε χωρίς συγκεκριμένες υποθέσεις για το ύψος στο οποίο επιτρέπεται να τοποθετηθούν σαρωτές, αλλά στην τελευταία εκδοχή της υλοποίησης υιοθετήθηκε η υπόθεση ότι αυτοί τοποθετούνται σε ύψος 1.10 – 1.70 μέτρα, όπως συμβαίνει συνήθως στην πραγματικότητα. Προφανώς για μεγαλύτερους σε ύψος χώρους (όπως συχνά συμβαίνει σε δημόσια κτίρια) θα πρέπει να αναθεωρηθεί αυτή η υπόθεση.

Προκειμένου να τερματίζει σε εύλογο χρόνο η προσπάθεια εντοπισμού των κατάλληλων θέσεων σάρωσης προσδιορίστηκε ένα μέγιστο επιτρεπόμενο πλήθος σαρωτών (για παράδειγμα, έγιναν πειράματα με 8, 10, 15, 20, και σε ορισμένες περιπτώσεις 25 σαρωτές). Σε μελλοντική εκδοχή της εφαρμογής αυτό θα μπορούσε να αποτελεί μια ακόμα παράμετρο εισόδου που θα μπορεί να προσδιορίζει ο χρήστης. Επίσης, η τρέχουσα εκδοχή της εφαρμογής υποστηρίζει αρχεία .obj και .xyz, και προφανώς σε μελλοντική εκδοχή θα

μπορούσαν να προστεθούν και άλλοι τύποι αρχείων. Τέλος, στην τρέχουσα εκδοχή της εφαρμογής δεν αποθηκεύονται οι επεξεργασμένοι τρισδιάστατοι χώροι μαζί με το αποτέλεσμα της τοποθέτησης σαρωτών, καθώς εάν αποθηκευτούν όπως οι διδιάστατοι σε αρχεία JSON, η ανακατασκευή από αυτά τα αρχεία νέφους σημείων για πιθανή επανεπεξεργασία είναι πολύπλοκη. Μελλοντικά, θα μπορούσε να διερευνηθεί η επιλογή κάποιου καταλληλότερου τρόπου αποθήκευσης (π.χ. με καταγραφή των συντεταγμένων των θέσεων σάρωσης στο νέφος σημείων).

Μερικές πιο ενδιαφέρουσες πτυχές για μελλοντική επέκταση του τμήματος της εφαρμογής για τρισδιάστατους χώρους περιλαμβάνουν:

- Δυναμικός επαναπροσδιορισμός του στόχου κάλυψης: Αντί ο αλγόριθμος να αξιολογεί το σύνολο του νέφους σημείων για κάλυψη, μετά από τον εντοπισμό μιας θέσης σάρωσης, ο ακάλυπτος χώρος θα αποτελεί τον μοναδικό ενεργό στόχο. Με αυτόν τον τρόπο ενδεχομένως να πετύχουμε οι σαρωτές να μην ανταγωνίζονται για τις ίδιες περιοχές (γεγονός που αντανακλάται από τις υψηλές τιμές συνεισφοράς στην κάλυψη που έχουν οι πρώτοι 1-2 σαρωτές), και να μειωθεί η επικάλυψη μεταξύ σαρωτών, ώστε ουσιαστικά να καταστεί πιο αποδοτική κάθε θέση σάρωσης.
- Η κάλυψη ενός σημείου από έναν σαρωτή είναι προς το παρόν Boolean (0, 1). Αλλά θα μπορούσε να εξεταστεί κατά πόσο η υιοθέτηση κάποιων κατάλληλων κριτηρίων θα επέτρεπε πιο λεπτομερή αποτύπωση της πληροφορίας αυτής, και ένα τέτοιο κριτήριο θα μπορούσε, ενδεχομένως, να είναι η γωνία πρόσπτωσης των ακτινών σάρωσης σε ένα σημείο του νέφους.
- Ευφύεστερη επιλογή υποψηφίων θέσεων σάρωσης. Στην τρέχουσα εκδοχή επιλέγονται τυχαία 500 σημεία του συνολικού νέφους που αντιστοιχεί στον τρισδιάστατο χώρο. Θα μπορούσαμε να δημιουργήσουμε ένα τρισδιάστατο πλέγμα και ο αλγόριθμος να δίνει προτεραιότητα σε σημεία που βρίσκονται κοντά σε ανοιχτούς χώρους ή κεντρικά σημεία. Αυτό θα ήταν κάπως αντίστοιχο με το στάδιο της προεπεξεργασίας ενός χώρου που προβλέφθηκε για την περίπτωση των χώρων δύο διαστάσεων. Προς την κατεύθυνση αυτή, η βιβλιογραφία προτείνει τη χρήση προηγμένων μεθόδων μη επιβλεπόμενης ομαδοποίησης δεδομένων (Plagianakos, 2014) για τη δυναμική διαμέριση του χώρου, σε συνδυασμό με μεταερευτικούς αλγορίθμους όπως η

Βελτιστοποίηση Σμήνους Σωματιδίων (PSO) και η Διαφορική Εξέλιξη (Epitropakis et al., 2012), οι οποίοι επιτρέπουν την αυτόματη χαρτογράφηση σύνθετων τοπολογιών και τον εντοπισμό βέλτιστων θέσεων κάλυψης χωρίς την ανάγκη τυχαίας δειγματοληψίας.

Τέλος, φυσικά, θα μπορούσε το τμήμα της εφαρμογής που ασχολείται με τους τρισδιάστατους χώρους να εμπλουτιστεί με περισσότερους αλγόριθμους ώστε ο χρήστης να επιλέγει με ποιους επιθυμεί να πειραματιστεί, δεδομένου κάποιου τύπου αρχείου. Η πιο σημαντική, όμως, μελλοντική αξιοποίηση της εφαρμογής είναι η δοκιμή της στο πεδίο, δηλαδή η τοποθέτηση σαρωτών σε πραγματικό χώρο με βάση τις υποδείξεις της, και η σύγκριση του ποσοστού κάλυψης που επιτυγχάνεται στην πράξη.

Η ενσωμάτωση τεχνητής νοημοσύνης στο πρόβλημα που μελετήθηκε θα αποτελούσε πολύ ενδιαφέρουσα επέκταση της παρούσας εργασίας. Θα μπορούσε, για παράδειγμα, να μελετηθεί η αυτοματοποιημένη εισαγωγή χαρτών, ώστε με τη χρήση συνελκτικών νευρωνικών δικτύων (Convolutional Neural Networks, CNNs) να αναγνωρίζονται αυτόματα τοίχοι και εμπόδια από αρχιτεκτονικά σχέδια PDF, εξαλείφοντας τη χειροκίνητη διαδικασία σχεδίασης (Dodge et al., 2017).

Επίσης, θα μπορούσε να διερευνηθεί κατά πόσο ένας πράκτορας ενισχυτικής μάθησης (Reinforcement Learning Agent) θα μπορούσε να εκπαιδευτεί σε περιβάλλοντα προσομοίωσης ώστε να βρίσκει τον ελάχιστο αριθμό σαρωτών μέσω δοκιμής και σφάλματος, ανακαλύπτοντας συχνά μη προφανείς στρατηγικές τοποθέτησης (Liu et al., 2023).

Με την ενσωμάτωση μοντέλων μηχανικής μάθησης για την αυτοματοποιημένη αναγνώριση περιβάλλοντος και ενισχυτικής μάθησης για τη δυναμική βελτιστοποίηση, το Scannalyze μπορεί να εξελιχθεί ως ένα εργαλείο υποστήριξης αποφάσεων με πλήρη χωρική επίγνωση.

7 Βιβλιογραφία

1. Abrahamsen M., Adamaszek A. , and Miltzow T. (2021). The Art Gallery Problem is $\exists\mathbb{R}$ -complete. *J. ACM* 69, 1, Article 4 (February 2022), 70 pages. <https://doi.org/10.1145/3486220>
2. Ahmed, W. and Rushdi, A. M. A. (2018). A novel minimization method for sensor deployment via heuristic 2-sat solution. *Sir Syed University Research Journal of Engineering & Technology*, 7(1), 7. <https://doi.org/10.33317/ssurj.v7i1.37>
3. Alsadik, B., Gerke, M., and Vosselman, G.: Visibility analysis of point cloud in close range photogrammetry, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci.*, II-5, 9–16, <https://doi.org/10.5194/isprsannals-II-5-9-2014>
4. Amit, Y., Mitchell, J., Packer, E. (2007). Locating Guards for Visibility Coverage of Polygons. *International Journal of Computational Geometry & Applications*. <https://doi.org/10.1142/S0218195910003451>
5. Bajuelos D., Canales C., Hernández P., Martins A. (2008). Optimizing the minimum vertex guard set on simple polygons via a genetic algorithm. *WSEAS Transactions on Information Science and Applications*, 5(11): 1584-1596. https://www.researchgate.net/publication/43761422_Optimizing_the_Minimum_Vertex_Guard_Set_on_Simple_Polygons_via_a_Genetic_Algorithm
6. de Berg, M., Cheong, O., van Kreveld, M., & Overmars, M. (2008). *Computational Geometry: Algorithms and Applications* (3rd ed.). Springer. <https://doi.org/10.1007/978-3-540-77974-2>
7. Bodor, R., Drenner, A., Schrater, P., Papanikolopoulos, N. (2007). Optimal Camera Placement for Automated Surveillance Tasks. *Journal of Intelligent and Robotic Systems*. 50. 257-295. <https://doi.org/10.1007/s10846-007-9164-7>
8. Chazelle B. (1984). Convex partitions of polyhedra: a lower bound and worst-case optimal algorithm. *SIAM J. Comput.* 13, 3 (August 1984), 488–507. <https://doi.org/10.1137/0213031>
9. Chazelle, B. (1991). Triangulating a simple polygon in linear time. *Discrete & Computational Geometry*, 6(3), 485-524. <https://doi.org/10.1007/BF02574703>
10. Chvátal, V. (1975). A combinatorial theorem in plane geometry. *Journal of Combinatorial Theory*, Series B, 18(1), 39–41. [https://doi.org/10.1016/0095-8956\(75\)90061-1](https://doi.org/10.1016/0095-8956(75)90061-1)
11. Connolly, C. (1985). The determination of next best views. In *Proceedings 1985 IEEE international conference on robotics and automation* (Vol. 2, pp. 432-435). IEEE. <https://doi.org/10.1109/ROBOT.1985.1087372>.

12. Couto, M. C., Rezende, P. J. d., & Souza, C. C. d. (2009). An ip solution to the art gallery problem. *Proceedings of the Twenty-Fifth Annual Symposium on Computational Geometry*, 88-89. <https://doi.org/10.1145/1542362.1542378>
13. Delmerico, J.A., Isler, S., Sabzevari, R., & Scaramuzza, D. (2018). A comparison of volumetric information gain metrics for active 3D object reconstruction. *Autonomous Robots*, 42 (2), 197 - 208. <https://doi.org/10.1007/s10514-017-9634-0>
14. Dodge, S., J. Xu and B. Stenger, "Parsing floor plan images," 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA), Nagoya, Japan, 2017, pp. 358-361, <https://doi.org/10.23919/MVA.2017.7986875>
15. Durand, F., Drettakis, G., & Thollot, J. (2002). Conservative visibility preprocessing using extended projections. *ACM Transactions on Graphics*. <https://doi.org/10.1145/344779.344891>
16. Eidenbenz, S., Stamm, C. & Widmayer, P. Inapproximability Results for Guarding Polygons and Terrains. *Algorithmica* 31, 79–113 (2001). <https://doi.org/10.1007/s00453-001-0040-8>
17. Epitropakis, M. G., Plagianakos, V. P., & Vrahatis, M. N. (2012). Evolving cognitive and social experience in particle swarm optimization through differential evolution: a hybrid approach. *Information Sciences*, 216, 50-92. <https://doi.org/10.1016/j.ins.2012.05.017>
18. Fisk, S. (1978). A short proof of Chvátal's watchman theorem. *Journal of Combinatorial Theory*, Series B, 24(3), 374. [https://doi.org/10.1016/0095-8956\(78\)90059-X](https://doi.org/10.1016/0095-8956(78)90059-X)
19. Ghosh, S.K. (2007). *Visibility Algorithms in the Plane*. Cambridge University Press. https://assets.cambridge.org/97805218/75745/frontmatter/9780521875745_frontmatter.pdf
20. Ghosh, S. K. (2010). Approximation algorithms for art gallery problems in polygons. *Discrete Applied Mathematics*, 158(6), 718-722. <https://doi.org/10.1016/j.dam.2009.12.004>
21. Glassner, A. S. (Ed.). (1989). *An introduction to ray tracing*. Morgan Kaufmann. <https://www.cs.cmu.edu/~fp/courses/graphics/cmu-only/ray1.pdf>
22. González-Banos, H. (2001). A randomized art-gallery algorithm for sensor placement. In *Proceedings of the seventeenth annual symposium on Computational geometry (SCG '01)*. Association for Computing Machinery, New York, NY, USA, 232–240. <https://doi.org/10.1145/378583.378674>
23. Huang, CF., Tseng, YC. (2005). The Coverage Problem in a Wireless Sensor Network. *Mobile Netw Appl* 10, 519–528. <https://doi.org/10.1007/s11036-005-1564-y>

24. Hornung, A., Wurm, K.M., Bennewitz, M. et al. (2013). OctoMap: an efficient probabilistic 3D mapping framework based on octrees. *Auton Robot* 34, 189–206 (2013). <https://doi.org/10.1007/s10514-012-9321-0>
25. Kolluri, R. (2008). Provably Good Moving Least Squares. *ACM Transactions on Algorithms (TALG)*, Volume 4, Issue 2, Article No.: 18, Pages 1 – 25. <https://doi.org/10.1145/1361192.1361195>
26. Kriegel, S., & Bodenmliller, T. & Suppa, M., & Hirzinger, G. (2011). A surface-based Next-Best-View approach for automated 3D model completion of unknown objects. *Proceedings of ICRA 2011*. 4869 - 4874. <https://doi.org/10.1109/ICRA.2011.5979947>
27. Kröller, A., Baumgartner, T., Fekete, S. P., & Schmidt, C. (2012). Exact solutions and bounds for general art gallery problems. *Journal of Experimental Algorithmics (JEA)*, 17, 2-1. <https://doi.org/10.1145/2133803.2184449>
28. Lee, D.T and Lin, A.K (1986). Computational complexity of art gallery problems. *IEEE Trans. Inf. Theor.* 32, 2 (March 1986), 276–282. <https://doi.org/10.1109/TIT.1986.1057165>
29. Liu, Z., Deshpande, M., X. Qi, D. Zhao, R. Madhivanan, A. Sen, Learning to Explore (L2E): Deep Reinforcement Learning-based Autonomous Exploration for Household Robot, in: Robotics: Science and Systems Workshop on Robot Representations for Scene Understanding, Reasoning, and Planning, 2023. <https://mit-spark.github.io/robotRepresentations-RSS2023/assets/papers/2.pdf>
30. Masuda, T. (2002). Registration and Integration of Multiple Range Images by Matching Signed Distance Fields for Object Shape Modeling. *Computer Vision and Image Understanding*, 87(1-3):51-65. <https://doi.org/10.1006/cviu.2002.0982>
31. O'Rourke, J. (1987), *Art Gallery Theorems and Algorithms*, Oxford University Press. https://www.science.smith.edu/~jorourke/books/ArtGalleryTheorems/Art_Gallery_Full_Book.pdf
32. O'Rourke, J. (2017). Visibility. In *Handbook of discrete and computational geometry* (pp. 875-896). Chapman and Hall/CRC. <https://doi.org/10.1201/9781315119601>
33. Pito, R. (1999). A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on pattern analysis and machine intelligence*, 21(10), 1016-1030. <https://doi.org/10.1109/34.799908>
34. Plagianakos, V.P. "Unsupervised clustering and multi-optima evolutionary search," 2014 *IEEE Congress on Evolutionary Computation (CEC)*, Beijing, China, 2014, pp. 2383-2390, <https://doi.org/10.1109/CEC.2014.6900431>

35. Plantinga, H., Dyer, C.R. Visibility, occlusion, and the aspect graph. *Int J Comput Vision* 5, 137–160 (1990). <https://doi.org/10.1007/BF00054919>
36. Ruppert, J.; Seidel, R. (1992), "On the difficulty of triangulating three-dimensional nonconvex polyhedra", *Discrete & Computational Geometry*, 7 (3): 227–253, <https://doi.org/10.1007/BF02187840>
37. Scott, W. R., Roth, G., & Rivest, J.F., (2003). View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.* 35, 1 (March 2003), 64–96. <https://doi.org/10.1145/641865.641868>
38. Seidel, R. (1991). A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry*, 1(1), 51-64. [https://doi.org/10.1016/0925-7721\(91\)90012-4](https://doi.org/10.1016/0925-7721(91)90012-4)
39. Shapley, L. S. (1952). *A Value for n-person Games* (pp. 307–317). RAND Corporation. <https://doi.org/10.1017/CBO9780511528446.003>
40. Shermer, T.C. (1992). Recent results in art galleries. *Proceedings of the IEEE*, 80(9), 1384–1399. <https://doi.org/10.1109/5.163407>
41. Terhar, F. and Icking, C. (2021). "The Sectional Art Gallery and an Evolutionary Algorithm for Approaching Its Minimum Point Guard Problem," *2021 IEEE Congress on Evolutionary Computation (CEC)*, Kraków, Poland, pp. 1390-1397. <https://doi.org/10.1109/CEC45853.2021.9504843>
42. Tozoni D., De Rezende P., and De Souza C. (2016). Algorithm 966: A Practical Iterative Algorithm for the Art Gallery Problem Using Integer Linear Programming. *ACM Trans. Math. Softw.* 43, 2, Article 16 (June 2017). <https://doi.org/10.1145/2890491>
43. Urrutia, Jorge. (2000). Art Gallery and Illumination Problems. *Handbook of Computational Geometry*. <https://doi.org/10.1016/B978-044482537-7/50023-1>
44. Woo, M., Neider, J., Davis, T., & Shreiner, D. (1999). *OpenGL programming guide: the official guide to learning OpenGL, version 1.2*. Addison-Wesley Longman Publishing Co., Inc.. <https://dl.acm.org/doi/book/10.5555/554539>
45. Yiakoumettis, C., Doulamis, N., Miaoulis, G., & Ghazanfarpour, D. (2013). Active learning of user's preferences estimation towards a personalized 3D navigation of geo-referenced scenes. *GeoInformatica*, 18(1), 27-62. <https://doi.org/10.1007/s10707-013-0176-0>
46. Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. *Proceedings 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. 'Towards New Computational Principles for Robotics and Automation'*, 146-151. <https://ieeexplore.ieee.org/document/613851>

47. Zachos A. and Christos-Nikolaos Anagnostopoulos. 2024. Using TLS, UAV, and MR Methodologies for 3D Modelling and Historical Recreation of Religious Heritage Monuments. *J. Comput. Cult. Herit.* 17, 4, Article 56 (December 2024), 23 pages. <https://doi.org/10.1145/3679021>
48. Zeng, R., Wen, Y., Zhao, W., & Liu, Y. J. (2020). View planning in robot active vision: A survey of systems, algorithms, and applications. *Computational Visual Media*, 6(3), 225-245. <https://doi.org/10.1007/s41095-020-0179-3>

Παράρτημα Α: Κώδικας Πτυχιακής Εργασίας

Ο κώδικας της εφαρμογής Scannalyze βρίσκεται στον σύνδεσμο:

<https://drive.google.com/drive/folders/1CPvlihMofB8nscD4DRG3LcFjk4mDHeTB?usp=sharing>

Υπεύθυνη Δήλωση Συγγραφέα:

Δηλώνω ρητά ότι, σύμφωνα με το άρθρο 8 του Ν.1599/1986, η παρούσα εργασία αποτελεί αποκλειστικά προϊόν προσωπικής μου εργασίας, δεν προσβάλλει κάθε μορφής δικαιώματα διανοητικής ιδιοκτησίας, προσωπικότητας και προσωπικών δεδομένων τρίτων, δεν περιέχει έργα/εισφορές τρίτων για τα οποία απαιτείται άδεια των δημιουργών/δικαιούχων και δεν είναι προϊόν μερικής ή ολικής αντιγραφής, οι πηγές δε που χρησιμοποιήθηκαν περιορίζονται στις βιβλιογραφικές αναφορές και μόνον και πληρούν τους κανόνες της επιστημονικής παράθεσης.