



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

**Συμβολή στην Ανάπτυξη Μεθόδου Υβριδικής (τοπολογικής /  
μετρικής) Χαρτογράφησης και Ταυτόχρονου Εντοπισμού  
Θέσης Κινητού Ρομπότ με Χρήση Οπτικής Πληροφορίας**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

του

ΜΙΧΑΗΛ Δ. ΤΣΟΠΕΛΑ

**Επιβλέπων :** Κωνσταντίνος Σ. Τζαφέστας  
Επίκουρος Καθηγητής Ε.Μ.Π.

Αθήνα, Ιούλιος 2011





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ  
ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΗΜΑΤΩΝ, ΕΛΕΓΧΟΥ ΚΑΙ ΡΟΜΠΟΤΙΚΗΣ

**Συμβολή στην Ανάπτυξη Μεθόδου Υβριδικής (τοπολογικής /  
μετρικής) Χαρτογράφησης και Ταυτόχρονου Εντοπισμού  
Θέσης Κινητού Ρομπότ με Χρήση Οπτικής Πληροφορίας**

**ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ**

του

**ΜΙΧΑΗΛ Δ. ΤΣΟΠΕΛΑ**

**Επιβλέπων :** Κωνσταντίνος Σ. Τζαφέστας  
Επίκουρος Καθηγητής Ε.Μ.Π.

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 19<sup>η</sup> Ιουλίου 2011.

.....

Κωνσταντίνος Τζαφέστας

Επίκουρος Καθηγητής Ε.Μ.Π

.....

Πέτρος Μαραγκός

Καθηγητής Ε.Μ.Π

.....

Κωνσταντίνος Κυριακόπουλος

Καθηγητής Ε.Μ.Π

Αθήνα, Ιούλιος 2011

.....  
ΜΙΧΑΗΛ Δ. ΤΣΟΠΕΛΑΣ

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

Copyright © Μιχαήλ Τσόπελας, 2011.

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

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

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν τον συγγραφέα και δεν πρέπει να ερμηνευθεί ότι αντιπροσωπεύουν τις επίσημες θέσεις του Εθνικού Μετσόβιου Πολυτεχνείου.

## Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες σε όλους τους ανθρώπους, οι οποίοι συνέβαλαν στο να ολοκληρώσω την παρούσα Διπλωματική Εργασία.

Ιδιαίτερα, θα ήθελα να ευχαριστήσω τον επιβλέπων καθηγητή μου, Κωνσταντίνο Τζαφέστα, για την πολύτιμη καθοδήγησή και την συνεχή υποστήριξή του. Η συμβολή του υπήρξε καίρια, κυρίως μέσω των εποικοδομητικών διαλόγων και των εύστοχων παρατηρήσεών του, που με καθοδηγούσαν και μου έθεταν στόχους. Κυρίως όμως θέλω να τον ευχαριστήσω για την κατανόησή και την υπομονή του.

Επίσης, θα ήθελα να ευχαριστήσω θερμά τον Δόκτωρ Γεώργιο Μούστρη για την πραγματικά ανυπολόγιστη βοήθεια του. Οι συμβουλές του και ο κώδικας που μου παρείχε, κυρίως σε ό,τι αφορούσε το ρομπότ, ήταν καθοριστικές. Τον ευχαριστώ πολύ για τον χρόνο που μου διέθεσε. Χωρίς αυτόν, η εργασία αυτή δεν θα είχε ολοκληρωθεί με επιτυχία.

Τέλος, θέλω να ευχαριστήσω ιδιαίτερα την μητέρα μου, Έφη, που με την αγάπη της και την υποστήριξή της, μου έδωσε δύναμη και κουράγιο για να ολοκληρώσω την εργασία αυτή.

# Περίληψη

Ο σκοπός της παρούσας διπλωματικής εργασίας ήταν η ανάπτυξη και η επέκταση μιας μεθοδολογίας για την επίλυση του προβλήματος της Ταυτόχρονης Εκτίμησης Θέσης και Χαρτογράφησης (SLAM – Simultaneous Localization and Mapping) σε ένα αυτόνομο, κινούμενο ρομποτικό όχημα. Η τεχνική που αναπτύχθηκε στηρίζεται σε συνδυασμό αφενός της μετρικής πληροφορίας από τους κωδικοποιητές στους τροχούς του ρομπότ και αφετέρου της οπτικής πληροφορίας, που λαμβάνεται από μια απλή, μονοσκοπική, web κάμερα. Ο συνδυασμός των δύο ειδών πληροφορίας γίνεται με ένα Διακριτό Φίλτρο Bayes. Για την μοντελοποίηση του περιβάλλοντος επελέχθη η μέθοδος του τοπολογικού χάρτη, εμπλουτισμένου με την μετρική πληροφορία που μας ήταν διαθέσιμη.

Δεδομένου ότι η μέθοδος αυτή έχει αναπτυχθεί για την επίλυση του SLAM σε περιβάλλοντα εσωτερικά και με έντονη δομή, χρησιμοποιήσαμε κάποιες απλουστευτικές παραδοχές ως προς την κίνηση του ρομπότ ( ακολουθώντας τις παραδοχές μιας προηγούμενης εργασίας που χρησιμοποιούσε οπτική οδομετρία). Η κωδικοποίηση της οπτικής πληροφορίας, έγινε με χρήση των χαρακτηριστικών SURF (Speeded-Up Robust Features). Παράλληλα, χρησιμοποιήσαμε και ιδέες οπτικής κωδικοποίησης από προηγούμενες εργασίες, κυρίως τα ιστογράμματα χρωματικού τόνου και κορεσμού, προκειμένου να κάνουμε σύγκριση της επίδοσής τους. Αναπτύξαμε ένα απλό αλλά αποτελεσματικό μοντέλο παρατήρησης προκειμένου να συγκρίνουμε εικόνες, χρησιμοποιώντας τα χαρακτηριστικά SURF. Επιπρόσθετα, θέλοντας να αξιοποιήσουμε την μετρική πληροφορία, βελτιώσαμε το προηγούμενο μοντέλο κίνησης (βασισμένο σε οπτική οδομετρία), με ένα πιο σύνθετο, το οποίο πλέον λαμβάνει υπ' όψιν την αβεβαιότητα της οδομετρίας του ρομπότ – μια αλλαγή που βελτίωσε πολύ, την συνέργεια των βημάτων πρόβλεψης και παρατήρησης του Διακριτού Φίλτρου Bayes.

Διεξήχθη ένα πλήθος πειραμάτων στο εσωτερικό των κτηρίων της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών του ΕΜΠ. Τα πειράματα αυτά είχαν τρεις στόχους. Πρώτον, να αναδείξουν το βελτιωμένο μοντέλο παρατήρησης με χρήση των χαρακτηριστικών SURF, σε σχέση με αυτό των ιστογραμμάτων χρωματικού τόνου και κορεσμού. Δεύτερον, να επιβεβαιώσουν την ικανότητα του συστήματός μας να εντοπίζει επιτυχώς την θέση του μέσα στον χώρο, καθώς κατασκευάζει ένα αντιπροσωπευτικό χάρτη αυτού, όπως επίσης και την δυνατότητά του να αναγνωρίζει βρόχους διαφόρων μηκών. Στο σημείο αυτό, μας ενδιέφερε επιπλέον να δείξουμε πώς το νέο βελτιωμένο μοντέλο κίνησης και η αξιοποίηση της μετρικής πληροφορίας, μπορεί να βελτιώσει σημαντικά την επίδοση του αλγορίθμου. Τέλος, στόχος μας ήταν να εντοπίσουμε τις αδυναμίες του αλγορίθμου.

Η μεθοδολογία αυτή μπορεί να γίνει οδηγός για την υλοποίηση ενός εύρωστου συστήματος Ταυτόχρονης Εκτίμησης Θέσης και Χαρτογράφησης, βασισμένου σε μετρική πληροφορία από κατάλληλους αισθητήρες (πχ. Κωδικοποιητές τροχών) πάνω στο ρομπότ και σε οπτική πληροφορία από κάμερες, κατάλληλου για εφαρμογές σε δομημένα περιβάλλοντα. Πιστεύουμε ότι η μεθοδολογία αυτή μπορεί εύκολα να βελτιωθεί και να επεκταθεί, καθώς είναι δομημένη σε ανεξάρτητα τμήματα.

**Λέξεις Κλειδιά:** αυτόνομο κινούμενο ρομπότ, εκτίμηση θέσης, χαρτογράφηση, SLAM, Οπτικό SLAM, τοπολογικός χάρτης, ρομποτική όραση, μονοσκοπική κάμερα, ταίριασμα εικόνων, μοντέλο κίνησης, μοντέλο παρατήρησης χαρακτηριστικά SURF, ιστογράμματα χρώματος, διακριτό φίλτρο Bayes.

# Abstract

The purpose of this Diploma Thesis was the development and expansion of a methodology to perform Simultaneous Localization and Mapping (SLAM) with an autonomous mobile robotic vehicle. The proposed technique relies on the fusion of the metric information received from the robot's encoders and visual information extracted from a simple, monocular, web camera. The fusion of these types of sensory information was accomplished using a Discrete Bayes Filter. The method chosen for modeling the environment was based on a topological representation, combined with the metric information received.

Given the fact that this method was developed to perform SLAM in indoor and heavily structured environments, we applied a few simplifying assumptions regarding the robot's movement (following the assumptions of a previous Diploma Thesis, which relied solely on a visual odometry scheme to calculate the movement of the robot). Visual information is encoded using SURF features (Speeded-Up Robust Features) and its effectiveness is compared to the visual encoding ideas from a previous Diploma Thesis, where HIS color histograms had been proposed. A simple, yet effective observation model was developed for the purpose of comparing images using SURF features. Furthermore, in order to utilize the metric information, we enhanced the simple visual odometry motion model with a more complex one, taking into consideration the uncertainty of the robot's pose; an addition that greatly improved the synergy achieved between the prediction and update steps of the Discrete Bayes Filter.

Finally, experiments were conducted in the building of the Electrical and Computer Engineering School of the National Technical University of Athens. The purpose of these experiments was threefold. Firstly, to show the improved performance of the new observation model, based on SURF features, as compared to the one based on HIS histograms. Secondly, to confirm that our system is capable of successfully localizing itself in the workspace, whilst creating a representative map of its environment, as well as effectively recognizing loop closures of different lengths. At this point, we were also interested in showing that our new improved motion model could greatly enhance the effectiveness of the developed SLAM algorithm. Lastly, a series of experiments was conducted in order to reveal specific weaknesses of our method.

The methodology proposed in this Diploma Thesis, can be used as a guide for the implementation of a robust SLAM system, based on metric information from the wheel's encoders and visual information from a monocular camera, mainly for structured environments. We believe that this methodology can be easily improved and expanded, given its modular structure.

**Keywords:** autonomous mobile robot, localization, mapping, SLAM, Visual SLAM, topological map, robotic vision, image matching, monocular camera, motion model, observation model, SURF features, HIS histograms, discrete Bayes filter.

# Περιεχόμενα

1	ΕΙΣΑΓΩΓΗ.....	11
1.1	Η Ιστορία του Ρομπότ .....	11
1.2	Το πρόβλημα του SLAM.....	12
1.3	Αντικείμενο διπλωματικής .....	12
1.3.1	Συνεισφορά.....	13
1.4	Οργάνωση κειμένου .....	14
2	SLAM- Βιβλιογραφική Επισκόπηση.....	16
2.1	Αισθητήριες διατάξεις.....	16
2.2	Αυτοεντοπισμός θέσης .....	17
2.2.1	Καθολικός Εντοπισμός Θέσης (Global Position Tracking) .....	17
2.2.2	Τοπικός Εντοπισμός Θέσης (Local Position Tracking).....	22
2.3	Χαρτογράφηση (Mapping).....	22
2.3.1	Ερευνητικές Προκλήσεις Προβλημάτων Ρομποτικής Χαρτογράφησης. ....	23
2.3.2	Αλγόριθμοι Χαρτογράφησης.....	25
2.4	Μέθοδοι Επίλυσης του SLAM.....	26
2.4.1	Ορισμός του SLAM.....	26
2.4.2	Το φίλτρο Kalman .....	28
2.4.3	Αλγόριθμοι Μεγιστοποίησης Προσδοκίας (Expectation Maximization Algorithms) .....	30
2.4.4	Υβριδικές Μέθοδοι.....	32
2.4.5	SLAM με Φίλτρο Σωματιδίων .....	33
2.5	Οπτικό SLAM .....	34
3	Θεωρητικό Υπόβαθρο .....	36
3.1	Ταίριασμα Εικόνων (Image Correspondence).....	36
3.1.1	Βιβλιογραφική Επισκόπηση .....	37
3.1.2	Μέθοδοι Βασισμένες σε Συσχέτιση-Περιοχές (Correlation-Area Based Methods) 37	
3.1.3	Cross-Correlation Μέθοδοι .....	38
3.1.4	Fourier-Based Μέθοδοι .....	39
3.1.5	Μέθοδοι Αμοιβαίας Πληροφορίας .....	40
3.1.6	Μέθοδοι Βασισμένοι σε Χαρακτηριστικά (Feature-Based Methods) .....	41
3.1.7	Ιστογράμματα Εικόνων .....	51
3.2	Χάρτες .....	53



3.2.1	Τοπολογικοί Χάρτες-Μέθοδοι Αναπαράστασης.....	53
3.3	Πιθανοτική Προσέγγιση .....	55
3.3.1	Ο Κανόνας του Bayes.....	55
3.3.2	Το Φίλτρο Bayes .....	56
3.3.3	Υπόθεση Πλήρους Κατάστασης – Αλυσίδα Markov .....	57
3.3.4	Κατανομή Πιθανότητας στο Χώρο Κατάστασης .....	57
3.3.5	Μή Παραμετρικά Φίλτρα .....	58
4	Τοπολογικό SLAM.....	60
4.1	Επίλυση του SLAM με Διακριτό Φίλτρο Bayes – Μοντέλα Κίνησης και Παρατήρησης και το Διάνυσμα Κατάστασης .....	60
4.1.1	Μοντέλο Κίνησης.....	61
4.1.2	Μοντέλο Παρατήρησης.....	63
4.1.3	Το Διάνυσμα Κατάστασης .....	68
4.2	Το Διακριτό Φίλτρο Bayes .....	68
4.2.1	Βήμα Πρόβλεψης .....	69
4.2.2	Βήμα Ανανέωσης .....	70
5	Οπτικό SLAM με Κάμερα.....	71
5.1	Αναπαράσταση Κόμβων με Ιστογράμματα Εικόνων και Χαρακτηριστικά SURF. 71	
5.1.1	Ιστογράμματα Χρωματικού Τόνου και Κορεσμού – Κατακερματισμός Εικόνας 72	
5.1.2	Αναπαράσταση με Χαρακτηριστικά SURF .....	73
5.2	Τοπολογικό SLAM με βάση Οπτική και Οδομετρική Πληροφορία .....	74
5.2.1	Μοντέλο Κίνησης.....	74
5.2.2	Μοντέλο Παρατήρησης.....	82
5.3	Αναγνώριση Βρόχου και Ανανέωση Χάρτη .....	83
5.4	Συνοπτική Παρουσίαση του Αλγορίθμου SLAM .....	87
6	Πειραματική Αξιολόγηση.....	88
6.1	Παράμετροι Αξιολόγησης .....	88
6.2	Τρόπος Αξιολόγησης.....	89
6.3	Συγκριτική Αξιολόγηση των Μοντέλων Παρατήρησης με Ιστογράμματα Εικόνας και Χαρακτηριστικά SURF .....	89
6.3.1	Αναγνώριση Βρόχων.....	90
6.3.2	Εισαγωγή Νέου Κόμβου.....	92
6.3.3	Ο ρόλος του Κατωφλίου $\theta$ των χαρακτηριστικών SURF.....	94
6.3.4	Συγκριτική Αξιολόγηση Ιστογραμμάτων-SURF σε Ολοκληρωμένες Διαδρομές 96	

6.4	Πειραματική Διάταξη Τηλεχειριζόμενου Κινητού Ρομπότ για Συλλογή Δεδομένων 97	
6.5	Ολοκληρωμένο Πείραμα Αναγνώρισης Βρόχου.....	101
6.6	Η Συνεισφορά της Μετρικής Πληροφορίας – Συνέργεια Βήματος Πρόβλεψης και Βήματος Ανανέωσης.....	112
6.7	Επαναληψιμότητα .....	113
6.8	Προβλήματα .....	117
6.9	Σύνοψη Συμπερασμάτων Αξιολόγησης .....	119
7	Επίλογος.....	121
7.1	Ανακεφαλαίωση και Συμπεράσματα.....	121
7.2	Μελλοντικές Επεκτάσεις.....	121
8	Βιβλιογραφία.....	123

## 1.1 Η Ιστορία του Ρομπότ

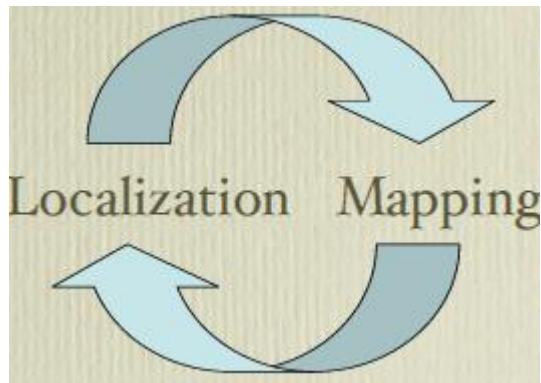
Ως ρομπότ, ορίζουμε οποιοδήποτε νοήμονα μηχανικό πράκτορα. Συνήθως, πρόκειται για μια ηλεκτρική-μηχανική συσκευή, η οποία ελέγχεται από κάποιο υπολογιστή και μπορεί να εκτελεί συγκεκριμένες λειτουργίες. Αν και η έννοια του ρομπότ μπορεί να εντοπιστεί από τα αρχαία χρόνια σε διάφορες θρησκείες του καθώς και σε κάποιες συσκευές, είναι σαφές ότι το ρομπότ με την έννοια που το αντιλαμβανόμαστε σήμερα, ήρθε πολύ αργότερα. Ο Leonardo da Vinci, σχεδίασε πρώτος ένα ανθρωπόμορφο ρομπότ το οποίο θα μπορούσε να κάνει κάποιες βασικές λειτουργίες, όπως το να είναι όρθιο ή να κουνάει τα χέρια του. Αργότερα στα τέλη του 19<sup>ου</sup> αιώνα ο Νίκολα Τέσλα, δημιουργεί ένα ρομπότ πολεμικού σκοπού, μια τηλεκατευθυνόμενη τορπίλη. Όμως το 1<sup>ο</sup> πραγματικά σύγχρονο ρομπότ δημιουργήθηκε το 1954 και χρησιμοποιήθηκε στην γραμμή παραγωγής της General Motors.

Σήμερα τα ρομπότ χωρίζονται σε δύο μεγάλες κατηγορίες:

- Εργοστασιακά ρομπότ. Χρησιμοποιούνται ως μέρος της γραμμής παραγωγής ενός εργοστασίου. Συνήθως αποτελούνται από ένα ρομποτικό βραχίονα με ένα τελικό εργαλείο. Επιτελούν μια πολύ συγκεκριμένη ενέργεια, που συνιστά ένα κομμάτι της γενικότερης παραγωγικής αλυσίδας.
- Κινητά και αυτόνομα ρομπότ. Η κατηγορία αυτή είναι πολύ ευρεία και περιλαμβάνει γενικά όλα τα ρομποτικά συστήματα με δυνατότητα κίνησης στο περιβάλλον, δηλαδή ρομπότ που δεν μένουν σε μια συγκεκριμένη θέση του χώρου. Τέτοια ρομπότ χρησιμοποιούνται για στρατιωτικούς, ερευνητικούς, ψυχαγωγικούς, ιατρικούς και άλλους σκοπούς. Έχουν την δυνατότητα να πηγαίνουν σε μέρη που είναι εξαιρετικά επικύνδυνα για τον άνθρωπο (λόγω θερμοκρασίας, ραδιενέργειας, ακτινοβολίας κα.). Συνήθως χαρακτηρίζονται από ένα βαθμό αυτονομίας, δηλαδή την δυνατότητα να αντιλαμβάνονται το περιβάλλον και να λαμβάνουν κατάλληλες αποφάσεις για την επίτευξη ενός στόχου. Προφανώς κάτι τέτοιο προϋποθέτει αισθητήρια όργανα, ικανά να παράσχουν στο ρομπότ την απαραίτητη πληροφορία. Μέχρι στιγμής τέτοια κινητά ρομπότ χρησιμοποιούνται με υψηλό κόστος για ιδιαίτερα σημαντικά project, όπως για παράδειγμα στην εξερεύνηση του διαστήματος. Τα τελευταία όμως χρόνια υπάρχει μια τάση για τη δημιουργία οικιακών ρομπότ, δηλαδή ρομπότ που θα απευθύνονται σε απλούς καταναλωτές. Φυσικά κάτι τέτοιο απαιτεί κατακόρυφη μείωση του κόστους παραγωγής, μεγάλο μέρος του οποίου αποτελεί το software και το hardware που απαιτεί το ρομπότ. Συνεπώς, χρειάζεται να βρεθούν τρόποι τα κινούμενα ρομπότ να αποκτήσουν όσο το δυνατόν πιο ευρεία αυτονομία, με το δυνατόν μικρότερο κόστος. Αυτό με τη σειρά του, προϋποθέτει τη δημιουργία αλγορίθμων, με μικρό κόστος υλοποίησης και ταυτόχρονα αποτελεσματικών. Αυτή η κατηγορία, δηλαδή των αυτόνομων κινητών ρομπότ, θα μας απασχολήσει στην παρούσα εργασία.

## 1.2 Το πρόβλημα του SLAM

Σε γενικές γραμμές, SLAM (Simultaneous Localization and Mapping) είναι το πρόβλημα όπου ένα ρομπότ πρέπει να εκτελέσει ταυτόχρονα δύο αποστολές: πρώτον, να χτίσει ένα χάρτη του περιβάλλοντος του και δεύτερον να αυτοεντοπίσει την θέση του στο περιβάλλον αυτό. Η δυσκολία του προβλήματος, έγκειται ακριβώς στο ότι αυτοί οι δύο στόχοι πρέπει να επιτευχθούν ταυτόχρονα. Σαφώς, η εκ των προτέρων τέλεια γνώση ενός εκ των δύο δηλαδή, είτε του χάρτη, είτε της θέσης του ρομπότ στο περιβάλλον, αυτομάτως καθιστά την επίλυση του άλλου σχεδόν τετριμμένη. Το SLAM αποτελεί κλασική περίπτωση του ερωτήματος «η κότα έκανε το αυγό ή το αυγό την κότα,»: από τη μία, η σωστή χαρτογράφηση απαιτεί τέλεια γνώση της θέσης του ρομπότ. Από την άλλη όμως, η εύρεση της θέσης του ρομπότ απαιτεί την τέλεια γνώση του χώρου, δηλαδή του χάρτη. Το SLAM θεωρείται από πολλούς ως μια αναγκαία δυνατότητα που πρέπει να έχει οποιοδήποτε αυτόνομο ρομπότ κινείται σε περιβάλλοντα όπου δεν υπάρχουν ακριβείς χάρτες και δεν είναι δυνατός ο αυτοεντοπισμός θέσης με άλλα μέσα.



Σχημα 1. Το SLAM είναι δύσκολο γιατί απαιτεί την ταυτόχρονη επίλυση δύο προβλημάτων, τα οποία μάλιστα αλληλοεπηρεάζονται.

## 1.3 Αντικείμενο διπλωματικής

Η παρούσα διπλωματική εργασία ασχολείται με την επίλυση του SLAM σε ένα τηλεχειριζόμενο μικρό ρομποτικό όχημα. Συγκεκριμένα, θα εφαρμόσουμε μια υβριδική μέθοδο εκτίμησης θέσης και χαρτογράφησης σε ένα αυτόνομο ρομποτικό όχημα. Υβριδική, διότι θα συνδυάζει τοπολογικό χάρτη με μετρική πληροφορία από τους αισθητήρες του ρομπότ. Συνίσταται δηλαδή, στην on-line αναπαράσταση του χώρου στον οποίο κινείται το ρομπότ κυρίως μέσω ενός τοπολογικού χάρτη, σε συνδυασμό με τον αυτοεντοπισμό της θέσης του ρομπότ στο πλαίσιο του χάρτη τον οποίο δημιουργεί. Αυτό επιτυγχάνεται αποκλειστικά με τη χρήση μιας μικρής κάμερας και της οδομετρίας του ρομπότ. Η πλούσια πληροφορία που παρέχει η κάμερα που είναι τοποθετημένη πάνω στο ρομπότ, συνδυάζεται

με στοιχεία οδομετρίας και δημιουργεί ένα χάρτη. Οι κόμβοι του χάρτη αυτού χαρακτηρίζονται από δύο συνιστώσες:

- Την οπτική πληροφορία από την κάμερα. Η οπτική πληροφορία μπορεί να είναι σε διάφορες μορφές (ολόκληρη η εικόνα, ιστογράμματα, περιοχές ενδιαφέροντος, συμβολοσειρές, χαρακτηριστικά σημεία κτλ).
- Τα στοιχεία της οδομετρίας: κάθε κόμβος χαρακτηρίζεται από μια θέση σε σχέση με το γενικό σύστημα συντεταγμένων και από τον προσανατολισμό.

Για την επίτευξη των παραπάνω, χρησιμοποιούμε ένα Διακριτό Φίλτρο Bayes. Η οπτική πληροφορία που χρησιμοποιούμε είναι τα χαρακτηριστικά SURF (Speeded-Up Robust Features).

### 1.3.1 Συνεισφορά

Η συνεισφορά της παρούσης διπλωματικής εργασίας μπορεί να συνοψισθεί στα ακόλουθα σημεία:

1. Μελέτη διαφόρων αποτελεσματικών μεθόδων για την επιτυχή σύγκριση εικόνων. Στόχος μας δεν ήταν τόσο η ταχύτητα, όσο το βέλτιστο αποτέλεσμα, όπως θα φανεί παρακάτω.
2. Σύγκριση διαφόρων μεθόδων ταιριάσματος εικόνων και αξιολόγηση της απόδοσής τους στο πρόβλημα του SLAM. Ιδιαίτερα, επικεντρωθήκαμε στην σύγκριση της επίδοσης των χρωματικών ιστογραμμάτων HSI και των χαρακτηριστικών SURF.
3. Δημιουργία ενός αλγορίθμου για σύγκριση χαρακτηριστικών SURF διαφορετικών εικόνων, προκειμένου να εξάγουμε ένα βαθμό ομοιότητας μεταξύ δύο εικόνων.
4. Επέκταση ενός ήδη υπάρχοντος αλγορίθμου SLAM βασισμένου αποκλειστικά σε οπτική πληροφορία, με ένταξη των νέων χαρακτηριστικών SURF και επιπλέον μετρικής πληροφορίας από τους αισθητήρες του ρομπότ.
5. Δημιουργία ενός μοντέλου αβεβαιότητας της κίνησης και ενός βήματος πρόβλεψης του Διακριτού Φίλτρου Bayes που εξαρτάται από αυτό. Ακόμα, αναπτύχθηκε μέθοδος αναγνώρισης βρόχου (loop closure), λαμβάνοντας επιπλέον πρόνοια για την ενσωμάτωση λειτουργιών διόρθωσης του χάρτη, με στόχο τη διατήρηση της συνέπειάς του και τη μείωση της αβεβαιότητας θέσης του ρομπότ, με ανάστροφη εφαρμογή του μοντέλου κίνησης.
6. Ανάπτυξη πλατφόρμας πειραματικών δοκιμών σε ρομποτικό όχημα τύπου Pioneer 3-DX, και ανάπτυξη διεπαφής τηλεχειρισμού μέσω απλής κινητής συσκευής. Αξιολόγηση της επίδοσης του νέου αλγορίθμου με πραγματικά πειραματικά δεδομένα που ελήφθησαν σε εσωτερικούς χώρους. Διαπίστωση ότι πράγματι ο

αλγόριθμος δημιουργεί ένα αξιόπιστο και αντιπροσωπευτικό χάρτη του περιβάλλοντος και παράλληλα μπορεί να αυτοεντοπίσει την θέση του μέσα σε αυτόν.

7. Υπόδειξη νέων κατευθύνσεων και πιθανών βελτιώσεων προκειμένου ο αλγόριθμος να εντάξει περισσότερη πληροφορία από αισθητήρες ή να διευρύνει το εύρος λειτουργίας του, μειώνοντας τις παραδοχές που απαιτεί (ιδιαίτερα της οπτικής οδομετρίας).

## 1.4 Οργάνωση κειμένου

Στο Κεφάλαιο 2 παρουσιάζουμε το πρόβλημα του SLAM και τα επιμέρους θέματα που εμπεριέχει. Μελετάμε ξεχωριστά τόσο το πρόβλημα του αυτοεντοπισμού θέσης (localization), όσο και το πρόβλημα της χαρτογράφησης (mapping). Επίσης, παρουσιάζουμε τις κύριες κατευθύνσεις της βιβλιογραφίας που επιλύουν το πρόβλημα του SLAM.

Στο Κεφάλαιο 3 ασχολούμαστε με κάποια θεωρητικά θέματα μοντελοποίησης του κόσμου (χάρτες) και εισάγουμε τον αναγνώστη στα θεωρητικά εργαλεία που θα χρησιμοποιήσουμε, όπως το Φίλτρο Bayes, την Υπόθεση Markov, και τα Φίλτρα Ιστογράμματος. Ακόμα, εξετάζουμε διάφορες μεθόδους που προσεγγίζουν το πρόβλημα της αντιστοίχισης (ταίριασμα) εικόνων. Το θέμα αυτό είναι πολύ ευρύ και μας απασχολεί ιδιαίτερα στην παρούσα εργασία που εξαρτάται άμεσα από την σύγκριση εικόνων, σε μια προσπάθεια να βελτιώσουμε την επίδοση του αλγορίθμου μας.

Στο Κεφάλαιο 4, εξηγούμε πώς το μοντέλο κίνησης και το μοντέλο παρατήρησης συνδυάζονται δημιουργώντας το Διακριτό Φίλτρο Bayes, δηλαδή τα βήματα πρόβλεψης και ανανέωσης αντίστοιχα. Ακόμα, αναλύουμε την μέθοδο που χρησιμοποιούμε για την σύγκριση των χαρακτηριστικών SURF και που αποτελεί το κύριο μοντέλο παρατήρησης του αλγορίθμου μας.

Στο Κεφάλαιο 5 παρουσιάζουμε το συνολικό αλγόριθμο επίλυσης SLAM που βασίζεται στη θεωρία του Κεφαλαίου 4, χρησιμοποιεί οπτική οδομετρία σε συνδυασμό με μετρική πληροφορία από τους αισθητήρες του ρομπότ, και δημιουργεί εν τέλει ένα τοπολογικό χάρτη που όμως εμπεριέχει μετρική πληροφορία από τους κωδικοποιητές του ρομπότ. Μοναδικό αισθητήριο όργανο είναι μια μονοσκοπική κάμερα ενώ η οπτική πληροφορία που χρησιμοποιείται είναι τα χαρακτηριστικά SURF.

Στο Κεφάλαιο 6 αναφερόμαστε στην πειραματική αξιολόγηση του αλγορίθμου. Αυτό περιλαμβάνει σύγκριση χαρακτηριστικών SURF και χρωματικών ιστογραμμάτων τόνου και κορεσμού, καθώς και μελέτη της επαναληψιμότητας της μεθόδου. Επίσης, εξετάζεται το κατά πόσον η μετρική πληροφορία που εισήχθη, βελτιώνει την επίδοση και την ευστάθεια της μεθόδου. Τέλος, επισημαίνονται τα προβλήματα και οι αδυναμίες της.

Στο Κεφάλαιο 7, γίνεται μια σύντομη ανακεφαλαίωση της μεθόδου που αναπτύχθηκε και προτείνονται νέες κατευθύνσεις και βελτιώσεις για μεταγενέστερες διπλωματικές εργασίες.

## 2 *SLAM- Βιβλιογραφική Επισκόπηση*

Με στόχο λοιπόν την δημιουργία ενός αυτόνομου ρομπότ, πρέπει όπως φάνηκε από τη διατύπωση του προβλήματος SLAM, να λύσουμε δύο υποπροβλήματα: το 1<sup>ο</sup> αφορά τον αυτοεντοπισμό της θέσης, ενώ το 2<sup>ο</sup> την χαρτογράφηση. Αυτά τα δύο θα βρίσκονται πάντα στην καρδιά του αλγορίθμου οποιουδήποτε πραγματικά αυτόνομου ρομπότ. Η επίλυση αυτών των προβλημάτων, έχει απασχολήσει του επιστήμονες για αρκετές δεκαετίες και συνεπώς έχουν αναπτυχθεί διάφορες μεθοδολογίες και προσεγγίσεις που τα επιλύουν. Όπως θα φανεί και στη συνέχεια, δεν υπάρχει μονοσήμαντος τρόπος να λύσει κανείς τέτοια προβλήματα. Οι παραδοχές, οι συνθήκες, τα διαθέσιμα μέσα, αλλά και το επιθυμητό αποτέλεσμα, είναι καθοριστικές παράμετροι για την μέθοδο και τον αλγόριθμο που χρησιμοποιείται. Στόχος είναι η ανάπτυξη συστημάτων χαρτογράφησης και πλοήγησης, που θα είναι εύρωστα και θα μπορούν να χρησιμοποιούνται σε κινούμενα ρομπότ που βρίσκονται σε μη εξερευνημένα (δηλαδή άγνωστα) και χωρίς ιδιαίτερη δομή περιβάλλοντα. Χαρακτηριστικά παραδείγματα αποτελούν τα οχήματα που στέλνονται να εξερευνήσουν μακρινούς πλανήτες, αυτόνομα υποβρύχια που στέλνονται για εξερεύνηση και ανακάλυψη νέων κοιτασμάτων, δηλαδή ρομπότ που βρίσκονται σε επικίνδυνα περιβάλλοντα, χωρίς πρότερη πληροφορία (χάρτες) για αυτά, ή ενδεχομένως κάποιους χάρτες μικρής ανάλυσης [1]. Είναι λοιπόν ιδιαίτερα σημαντικό, τα ρομπότ αυτά να μπορούν να στηρίζονται σε πολύ μεγάλο βαθμό στα δεδομένα που παίρνουν από τους αισθητήρες τους και όχι σε κάποια εκ των προτέρων δοσμένη πληροφορία. Στο παρόν κεφάλαιο, θα μιλήσουμε για κάποιες βασικές έννοιες και μεθόδους που απαρτίζουν το πρόβλημα του SLAM.

Στην Ενότητα 2.1, θα αναφερθούμε επιγραμματικά στις κύριες *αισθητήριες διατάξεις* ενός ρομπότ.

Εν συνεχεία, η Ενότητα 2.2 εξετάζει το πρόβλημα του *αυτοεντοπισμού θέσης (localization)* που άλλωστε αποτελεί και τη μια όψη του προβλήματος SLAM.

Στην Ενότητα 2.3, μελετάμε την άλλη όψη του SLAM, δηλαδή το πρόβλημα της *χαρτογράφησης (mapping)*.

Ακολούθως, στην Ενότητα 2.4 αναφερόμαστε στις πιο διαδεδομένες υπάρχουσες μεθόδους επίλυσης του SLAM.

Τέλος, η Ενότητα 2.5 αναφέρεται σε μερικές ενδιαφέρουσες προσεγγίσεις για την επίλυση του Οπτικού SLAM.

### 2.1 *Αισθητήριες διατάξεις*

Πρωτού προχωρήσουμε σε πιο ειδικά θέματα, είναι σημαντικό να αναφερθούμε στις πηγές πληροφορίας ενός ρομπότ, τους αισθητήρες. Η επίλυση ενός προβλήματος SLAM



προϋποθέτει γνώση για την κατάσταση του ρομπότ, συνεπώς οι αισθητήρες είναι απαραίτητοι. Υπάρχουν 2 βασικές κατηγορίες αισθητήρων:

- **Ιδιοδεκτικοί (proprioceptive)** αισθητήρες, είναι αυτοί που συλλέγουν πληροφορίες για την εσωτερική κατάσταση του ρομπότ. Τέτοιοι μπορεί να είναι επιταχυνσιόμετρα, γυροσκόπια, αισθητήρες, κωδικοποιητές κτλ. Στην παρούσα εργασία τέτοιοι αισθητήρες θα χρησιμοποιηθούν για την οδομετρία του ρομπότ.
- **Εξωτερικοί (exteroceptive)** αισθητήρες, είναι αυτοί που μετρούν εξωτερικά μεγέθη, δηλαδή μεγέθη που σχετίζονται με το περιβάλλον. Για παράδειγμα, αισθητήρες που μετρούν την απόσταση, την θερμοκρασία ή την ένταση της ακτινοβολίας ή που λαμβάνουν ήχους ή και εικόνα. Laser, infrared, sonar, κάμερες, μικρόφωνα αποτελούν περιπτώσεις τέτοιων αισθητήρων. Στην παρούσα εργασία ο μόνος εξωτερικός αισθητήρας που χρησιμοποιείται, είναι μια κάμερα.

*Το κλειδί στην επίλυση του SLAM δεν είναι άλλο από την εύρεση του κατάλληλου αλγορίθμου για τον σύνθεση (fusion) της πληροφορίας από αυτά τα δύο είδη αισθητήρων.*

## 2.2 Αυτοεντοπισμός θέσης

Ο αυτοεντοπισμός θέσης αποτελεί ένα πολύ σημαντικό ζήτημα για οποιοδήποτε αυτόνομο ρομπότ. Ακόμα, είναι απαραίτητο κομμάτι για κάθε αλγόριθμο που επιλύει το πρόβλημα του SLAM. Το πρόβλημα του αυτοεντοπισμού θέσης θεωρεί το χάρτη του κόσμου γνωστό και προσπαθεί να προσδιορίσει την θέση του ρομπότ σε αυτόν. Το πρόβλημα αυτό είναι ευρύ και επιδέχεται αρκετές προσεγγίσεις στην επίλυσή του [2]:

- **Γεωμετρικές Μέθοδοι.** Συνήθως χρησιμοποιούν ένα διδιάστατο διακριτό πλέγμα (*occupancy grid*). Ο αυτοεντοπισμός θέσης γίνεται προσπαθώντας να βρούμε την θέση του ρομπότ σε σχέση με τις συντεταγμένες του χάρτη-πλέγματος.
- **Τοπολογικές μέθοδοι.** Χρησιμοποιούν ένα *γράφο* και τον *πίνακα πρόσπτωσης* που του αντιστοιχεί για να περιγράψουν τον κόσμο. Προσπαθούν να εντοπίσουν τον κόμβο στον οποίο βρίσκεται το ρομπότ.
- **Υβριδικές μέθοδοι.** Αποτελούν συνδυασμό των δύο παραπάνω.

Ο αυτοεντοπισμός θέσης (*localization*), χωρίζεται σε δύο μεγάλες κατηγορίες.

### 2.2.1 Καθολικός Εντοπισμός Θέσης (Global Position Tracking)

Στο πρόβλημα αυτό, το ρομπότ πρέπει να εντοπίσει τη θέση του στο πλαίσιο ενός εκ των προτέρων γνωστού χάρτη. Βέβαια, εάν κανένας χάρτης δεν είναι εκ των προτέρων διαθέσιμος θα πρέπει πρώτα να κατασκευασθεί.

Υπάρχουν διάφορες μέθοδοι που μπορεί κανείς να χρησιμοποιήσει για τον καθολικό εντοπισμό θέσης.

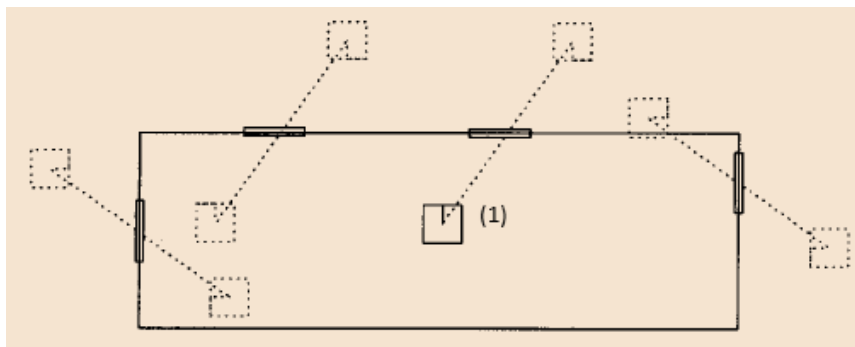
Η 1<sup>η</sup> μέθοδος είναι ίσως και η πιο διαισθητική και στηρίζεται στην χρήση *ορόσημων* (*landmarks*). Ως *ορόσημο* ορίζουμε οτιδήποτε μπορεί να ξεχωρίζει στον χώρο και να αποτελεί ένα ιδιαίτερο χαρακτηριστικό του. Ας πάρουμε για παράδειγμα τον χάρτη των πόλεων του κόσμου. Η Ακρόπολη αποτελεί ένα ορόσημο της Αθήνας, με την έννοια ότι εάν κάποιος άνθρωπος που βρίσκεται σε μια πόλη δει την Ακρόπολη, τότε μπορεί με μεγάλη σιγουριά να συνάγει ότι βρίσκεται στην Αθήνα. Φυσικά δεν θα μπορεί να πει ακριβώς σε ποιο μέρος της Αθήνας βρίσκεται. Η λογική αυτή λοιπόν μπορεί να εφαρμοστεί και στον εντοπισμό της καθολικής θέσης ενός ρομπότ. Τα συστήματα ανίχνευσης οροσήμων στηρίζονται είτε σε τεχνητά, είτε σε φυσικά χαρακτηριστικά στοιχεία του περιβάλλοντος. Τα τεχνητά ορόσημα ανιχνεύονται ευκολότερα και με μεγαλύτερη αξιοπιστία απ' ό,τι τα φυσικά. Όμως, τα τεχνητά ορόσημα απαιτούν μεταποίηση του περιβάλλοντος που σημαίνει ότι κάποιος θα πρέπει εκ των προτέρων να τα τοποθετήσει στον χώρο που πρόκειται να κινηθεί το ρομπότ. Αυτό αποτελεί μεγάλο περιοριστικό παράγοντα και ως εκ τούτου τα συστήματα που βασίζονται σε φυσικά ορόσημα συχνά προτιμούνται. Διάφορα χαρακτηριστικά στοιχεία έχουν κατά καιρούς χρησιμοποιηθεί ως φυσικά ορόσημα: γωνίες, πόρτες, φώτα οροφής, συστήματα εξαερισμού στο ταβάνι και χαρακτηριστικά κτίρια. Λόγω του ότι τα περισσότερα συστήματα που βασίζονται σε ορόσημα είναι σχεδιασμένα για συγκεκριμένα περιβάλλοντα, σπάνια μπορούν εύκολα να χρησιμοποιηθούν σε διαφορετικά περιβάλλοντα. Επίσης είναι πιθανό σε μη στατικά περιβάλλοντα διάφορα ορόσημα να μεταβάλλονται. Για παράδειγμα, μια πόρτα που πριν ήταν κλειστή και στη συνέχεια το ρομπότ την συναντά ανοιχτή, ενδέχεται να προκαλέσει σύγχυση, αφού το ρομπότ δεν θα μπορεί άμεσα να συνάγει εάν πρόκειται για μια καινούργια ανοιχτή πόρτα, ή για μια προηγουμένως κλειστή αλλά πλέον ανοιχτή πόρτα.

Μια 2<sup>η</sup> μέθοδος που περιγράφεται στο [1], στηρίζεται σε ένα Occupation Grid (πλέγμα κατάληψης). Σύμφωνα με τη μέθοδο αυτή, που μπορεί να χρησιμοποιηθεί τόσο για χαρτογράφηση όσο και για αυτοεντοπισμό θέσης, ο χώρος χωρίζεται σε πολλά τετράγωνα, κάθε ένα από τα οποία χαρακτηρίζεται από δύο πιθανοτικά (ή και ντετερμινιστικά) μεγέθη: άδειο (*empty*) ή κατειλημμένο (*occupied*). Ο σχεδιασμός τροχιάς γίνεται με βάση κυρίως τον αλγόριθμο A\*, προσπαθώντας να ελαχιστοποιήσουμε μια συνάρτηση κόστους που λαμβάνει υπ' όψιν της την αβεβαιότητα του πλέγματος.

Η 3<sup>η</sup> μεθοδολογία βασίζεται στο *Φίλτρο Κάλμαν* και το *Εκτεταμένο Φίλτρο Κάλμαν* (*Extended Kalman Filters-EKF*) που είναι η μη γραμμική έκδοση του απλού *Φίλτρου Κάλμαν*. Τα *Φίλτρα Κάλμαν* στηρίζονται στην υπόθεση, ότι εάν η αρχική κατάσταση και ο θόρυβος περιγράφονται από Gaussian κατανομές τότε η θέση του ρομπότ και ο χάρτης θα περιγράφονται και αυτές από Gaussian κατανομές. Το πρόβλημα των *Φίλτρων Kalman* είναι ότι δεν μπορούν να αναπαραστήσουν κατανομές με πολλαπλά μέγιστα (*multimodal*), φαινόμενο που θα προκύψει εάν συναντήσουμε ένα ορόσημο που μοιάζει με ένα άλλο ορόσημο του χάρτη μας. Με άλλα λόγια τα φίλτρα Κάλμαν δεν μπορούν να επιλύσουν το πρόβλημα της αντιστοίχισης ή συσχέτισης πληροφορίας (*correspondence- data association problem*) δηλαδή δεν μπορούν να αντιληφθούν εάν αισθητηριακά δεδομένα διαφορετικών χρονικών στιγμών, αντιστοιχούν στο ίδιο φυσικό αντικείμενο του κόσμου [3], [4]. Πρακτικά, αυτό σημαίνει ότι ενώ μπορεί να είμαστε σε ένα διαφορετικό σημείο, που όμως μοιάζει με κάποιο που επισκεφτήκαμε προηγουμένως, τότε κατά πάσα πιθανότητα το *Φίλτρο Κάλμαν* δεν θα μπορέσει να καταλάβει την διαφορά και συνεπώς θα έχουμε λανθασμένη εκτίμηση θέσης.

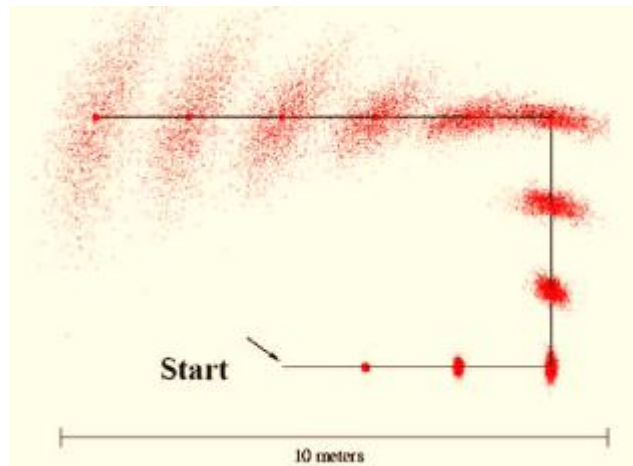
Μια άλλη μέθοδος που αναλύεται στο [5], βασίζεται και αυτή στο *Φίλτρο Κάλμαν*, όμως προσπαθεί να ξεπεράσει το παραπάνω πρόβλημα της αντιστοίχισης, δημιουργώντας πολλές

υποθέσεις σχετικά με την θέση του ρομπότ στον χώρο και εφαρμόζοντας τον αλγόριθμο του Φίλτρου Κάλμαν σε κάθε μια από αυτές, μέχρις ότου κάποια ξεχωρίσει και αναδειχθεί νικήτρια. Για παράδειγμα, εάν κάποιος μας πει ότι το ρομπότ βλέπει στα δεξιά του μια κλειστή πόρτα, με δεδομένο τον χώρο στο παραπάνω σχήμα, τότε αμέσως προκύπτουν 8 διαφορετικές υποθέσεις σχετικά με την πιθανή θέση του ρομπότ στον χώρο. Η συλλογή δεδομένων από τους αισθητήρες, τελικά αναδुकνει νικήτρια την υπόθεση (1), όπως φαίνεται στο Σχημα 2.



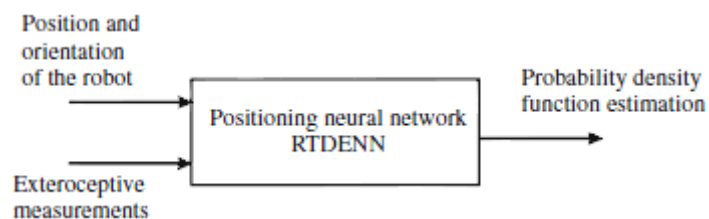
**Σχημα 2.** Αρχικά υπάρχουν 8 διαφορετικές θέσεις για τις οποίες το ρομπότ συναντά μια πόρτα στα δεξιά του. Το πρόβλημα λύνεται με συνδυασμό πληροφορίας από τους αισθητήρες, και αναδुकνει την (1) ως νικήτρια θέση.

Η 4<sup>η</sup> μεθοδολογία βασίζεται στο επονομαζόμενο φίλτρο σωματιδίων (particle filtering [6] και είναι ίσως η πιο κοινή. Δημιουργήθηκε για να καλύψει τα μειονεκτήματα άλλων μεθόδων όπως του φίλτρου Κάλμαν και των πλεγμάτων Markov. Το φίλτρο Κάλμαν για παράδειγμα, δεν μπορεί να επανέλθει (δηλαδή να επαναπροσδιορίσει τη θέση ή τον χάρτη) μετά από μια λανθασμένη εκτίμηση, ενώ τα πλέγματα Markov είναι πολύ απαιτητικά σε μνήμη και χρόνο. Πολύ σημαντικό είναι το γεγονός ότι ενώ τα Kalman φίλτρα περιορίζονται σε Gaussian θόρυβο και σε γραμμικά μοντέλα, με Gaussian αρχική κατάσταση, (θεωρώντας ότι εάν αυτά ισχύουν τότε η πιθανότητα  $p(x_k | Z_k)$ , όπου  $x_k$  είναι η θέση και  $Z_k$  η παρατήρηση της χρονικής στιγμής  $k$ , είναι και αυτή Gaussian), το Particle Filtering δεν κάνει καμία τέτοια παραδοχή. Επίσης η ταχύτητα εκτέλεσης και η ακρίβεια της μεθόδου εξαρτώνται από την επαναδειγματοληψία που κάνουμε πάνω στα σωματίδια (δηλαδή το πόσα σωματίδια διατηρούμε σε κάθε επανάληψη). Η μέθοδος αυτή, αντί να προσπαθεί να βρει αναλυτικά ολόκληρη την συνάρτηση πιθανότητας, διατηρεί μόνο ένα αριθμό από δείγματα παρμένα από αυτή. Κάθε δείγμα αποτελεί μια διάταξη (configuration) του ρομπότ στον χώρο. Σε κάθε επανάληψη του αλγορίθμου έχουμε δύο βήματα: στο 1<sup>ο</sup> βήμα, που αποτελεί το μοντέλο κίνησης, τα  $n$  σωματίδια(που εμπεριέχουν την θέση και τον προσανατολισμό του ρομπότ) υφίστανται ένα μετασχηματισμό με βάση το μοντέλο κίνησης του ρομπότ. Φτιάχνονται έτσι  $n$  νέα σωματίδια. Το 2<sup>ο</sup> βήμα υλοποιεί το μοντέλο παρατήρησης, όπου η πληροφορία από τους αισθητήρες χρησιμοποιείται για να «σταθμίσουμε» κάθε σωματίδιο με ένα βάρος που εξαρτάται από την πληροφορία που έχουμε. Τα σωματίδια με την μεγαλύτερη πιθανότητα επιζούν ενώ τα υπόλοιπα χάνονται.



**Σχημα 3.** Παράδειγμα εφαρμογής Φίλτρου Σωματιδίων για τον υπολογισμό της θέσης του ρομπότ. Παρατηρείστε ότι καθώς το ρομπότ προχωρά στον χώρο, αυξάνεται η αβεβαιότητα στη θέση του και τα σωματίδια αρχίζουν να σκορπίζουν όλο και περισσότερο.

Τα νευρωνικά δίκτυα έχουν και αυτά παίξει τον ρόλο τους στη προσπάθεια αυτή. Όπως φαίνεται στο [7], υπάρχουν διάφορες ιδέες από αυτοοργανούμενα δίκτυα SOM (Self-Organized Maps) που κατηγοριοποιούν την θέση του ρομπότ στο περιβάλλον, μέχρι δίκτυα RBF (Radial Basis Function) και MLP (Multi-layered Perceptron) τα οποία μαθαίνουν την εκ των προτέρων σχέση μεταξύ αισθητηριακών δεδομένων, και της θέσης του ρομπότ στο περιβάλλον. Στη συγκεκριμένη εργασία, συνδυάζονται η πρόβλεψη του dead reckoning για την θέση του ρομπότ και ενός μοντέλου θέσης (Positional Model) το οποίο στηρίζεται σε ένα νευρωνικό δίκτυο εντοπισμού θέσης (Positional Neural Network) το οποίο δέχεται ως είσοδο την θέση του ρομπότ και μετρήσεις των αισθητήρων, και δίνει ως έξοδο συναρτήσεις πυκνότητας πιθανότητας των μεταβλητών εισόδου. Σε περίπτωση ασυμφωνίας των δύο μοντέλων, γίνεται μια ευριστική εκτίμηση της θέσης του ρομπότ. Το πλεονέκτημα αυτής της προσέγγισης είναι η δυνατότητα προσαρμογής στο περιβάλλον, επιτρέποντας την χρήση της σε SLAM, καθώς και η σταθερότητά της.



**Σχημα 4.** Η βασική μορφή του νευρωνικού δικτύου, όπως περιγράφεται στο [7], για τον υπολογισμό της θέσης του ρομπότ. Πηγή: [7].

Επιπρόσθετα, στο [8], περιγράφεται μια μέθοδος με την οποία η θέση του ρομπότ εκτιμάται με βάση δεδομένα από laser scanners. Η μέθοδος αυτή γενικά είναι πολύ σταθερή και αποτελεσματική σε μή δομημένα και δυναμικά περιβάλλοντα σε αντίθεση με γεωμετρικές ή οπτικές μεθόδους. Επίσης, έχει λύση σε κλειστή μορφή με αποτέλεσμα να είναι ιδιαίτερα

γρήγορη και ακριβής. Η βασική ιδέα είναι ότι γίνεται μια προσπάθεια να ταιριάξουν διαδοχικές μετρήσεις από τα laser προκειμένου να υπολογιστεί η μετατόπιση του ρομπότ.

Τέλος υπάρχουν διάφορες μέθοδοι που βασίζονται αποκλειστικά σε οπτική πληροφορία. Οι μέθοδοι αυτές συνήθως χρησιμοποιούν ένα τοπολογικό χάρτη για την αναπαράσταση του χώρου. Για παράδειγμα στο [2], εξάγονται ιστογράμματα HSI, και γίνεται ένα βέλτιστο ταίριασμα του κάθε ιστογράμματος με τη βάση των αποθηκευμένων εικόνων. Στην συνέχεια γίνεται από κάθε ιστόγραμμα μια ψηφοφορία που αφορά τον κόμβο στον οποίο βρίσκεται το ρομπότ. Το πρόβλημα της μεθόδου είναι ότι πρώτον απαιτεί κάποιον χειριστή να οδηγήσει το ρομπότ σε ένα 1<sup>ο</sup> πέρασμα από τον χώρο προκειμένου να πάρει εικόνες του χώρου και δεύτερον ότι πρέπει κάποιος χειροκίνητα να αντιστοιχίσει τις εικόνες αυτές σε κόμβους ενός τοπολογικού χάρτη.

Στο [9] περιγράφεται μια μέθοδος όπου κατακόρυφες ακμές της εικόνας, εντοπίζονται μέσω ιστογραμμάτων και συνδυάζονται με την αναγνώριση χρωματικών περιοχών προκειμένου να δημιουργηθεί μια ακολουθία χαρακτήρων που θα περιγράψει την εν λόγω εικόνα. Στη συνέχεια, για να εντοπίσει το ρομπότ την θέση του, δεν έχει παρά να συγκρίνει την συμβολοακολουθία της τωρινής εικόνας, με αυτές που είναι αποθηκευμένες στη μνήμη του, και να βρει αυτή με το καλύτερο ταίριασμα.

Στο [10] χρησιμοποιούνται και πάλι τα ιστογράμματα από διάφορους χρωματικούς χώρους ως ορόσημα. Η θέση των οροσίων εντοπίζεται στο πρώτο πέρασμα από τον χώρο με χρήση στερεοσκοπίας σε σχέση με ένα σύστημα συντεταγμένων «εξερεύνησης». Στο επόμενο πέρασμα του ρομπότ, τα ορόσημα εντοπίζονται σε μια κάπως διαφορετική θέση. Έτσι η θέση του ρομπότ στον χώρο μπορεί να υπολογιστεί ελαχιστοποιώντας το Μέσο Τετραγωνικό Σφάλμα των θέσεων των οροσίων στα 2 αυτά περάσματα.

Στο [11], γίνεται τόσο τοπικός όσο και καθολικός εντοπισμός θέσης με χρήση χαρακτηριστικών SIFT που εξάγονται από εικόνες, χρησιμοποιώντας ένα σύστημα από τρεις κάμερες. Με κατάλληλη επεξεργασία των εικόνων από τις 3 κάμερες, είναι δυνατόν να βρει κανείς την μετατόπιση-στροφή του ρομπότ.

Στο [12], χρησιμοποιείται η ιδέα του αποτυπώματος. Κάθε ορόσημο του χάρτη, χαρακτηρίζεται από ένα μοναδικό αποτύπωμα, το οποίο εξάγεται από τις ακμές και τα τμήματα της εικόνας που χαρακτηρίζονται έντονα από κάποιο χρώμα. Το τελικό αποτέλεσμα είναι μια ακολουθία χαρακτήρων (string) το οποίο εν συνεχεία μπορεί να συγκριθεί με χρήση υπάρχοντων αλγορίθμων, με άλλες ακολουθίες προκειμένου να εντοπιστεί εκείνη με την οποία έχουν τις μικρότερες διαφορές.

Τέλος στο [13], χρησιμοποιώντας μια πανοραμική κάμερα, εξάγονται τα τοπικά χαρακτηριστικά (τύπου SURF-Speeded Up Robust Features) και γίνεται ένα ταίριασμα με τα χαρακτηριστικά που είναι αποθηκευμένα στο χάρτη του ρομπότ. Η συγκεκριμένη εργασία, χρησιμοποιεί και ένα είδος μνήμης, βραχυχρόνιας όσο και μακροχρόνιας, προκειμένου να μπορεί να αντιμετωπίσει δυναμικές αλλαγές στον περιβάλλοντα χώρο του ρομπότ.

## 2.2.2 Τοπικός Εντοπισμός Θέσης (Local Position Tracking)

Στο πρόβλημα αυτό, το ρομπότ πρέπει να εντοπίσει την θέση του «τοπικά», δηλαδή με βάση την προηγούμενη γνωστή θέση του και στοιχεία σχετικά με τον προσανατολισμό, την ταχύτητα και την επιτάχυνσή του. Είναι γνωστή και ως *Dead Reckoning*. Πρακτικά συλλέγει πληροφορίες από τους αισθητήρες του ρομπότ και βάση ενός κινητικού μοντέλου υπολογίζει την νέα θέση του ρομπότ δεδομένης της αμέσως προηγούμενης. Λέγεται *dead reckoning* διότι ουσιαστικά υπολογίζει την θέση «με κλειστά τα μάτια», χωρίς να λαμβάνει υπ' όψιν άλλη πληροφορία από το περιβάλλον. Αυτό σημαίνει ότι η μέθοδος αυτή εγγενώς εμπεριέχει ολοκληρωτικά σφάλματα, δηλαδή σφάλματα που με το πέρασ του χρόνου συσσωρεύονται. Έτσι, εάν χρησιμοποιηθεί χωρίς επιπλέον πληροφορία, είναι σχεδόν βέβαιο ότι μετά από λίγο χρόνο η πραγματική θέση του ρομπότ θα απέχει πολύ από την προβλεπόμενη μέσω *dead reckoning* θέση.

Ο κύριος λόγος συσσώρευσης σφαλμάτων οφείλεται στον θόρυβο των μετρήσεων, σε συστηματικά σφάλματα των αισθητήρων, σε πιθανή ολίσθηση των τροχών κατά την κίνηση του ρομπότ αλλά και σε πιθανή ανακρίβεια (λόγω προσεγγίσεων) του κινηματικού μοντέλου που χρησιμοποιείται. Φυσικά για μικρές αποστάσεις, η μέθοδος αυτή είναι ιδιαίτερα αποτελεσματική.

## 2.3 Χαρτογράφηση (Mapping)

Η χαρτογράφηση αποτελεί την 2<sup>η</sup> εργασία ενός αλγορίθμου SLAM και ένα απαραίτητο βήμα για την υλοποίηση ενός πραγματικά αυτόνομου, κινητού ρομπότ. Τα τελευταία χρόνια έχουν αναπτυχθεί πολλές μέθοδοι ικανές να φτιάξουν σε πραγματικό χρόνο, περίπλοκους χάρτες κυρίως σε εσωτερικά περιβάλλοντα. Πολλές από τις υπάρχουσες τεχνικές είναι ιδιαίτερα ανθεκτικές στον θόρυβο και μπορούν να αντιμετωπίσουν πλήθος στατικών αλλά και δυναμικών περιβάλλοντων. Γενικά υπάρχουν δύο μεγάλα κατηγορίες προβλημάτων χαρτογράφησης [14]. Η 1<sup>η</sup> κατηγορία θεωρεί το περιβάλλον στατικό, υπό την έννοια ότι μόνο το ρομπότ μπορεί να κινείται μέσα σε αυτό, ενώ ο χώρος γύρω του παραμένει αμετάβλητος. Συνεπώς μπορούμε με ασφάλεια να υποθέσουμε ότι σε διαδοχικά περάσματα από τα ίδια σημεία, δεν θα εντοπίζουμε αλλαγές. Η 2<sup>η</sup> κατηγορία αντιμετωπίζει ένα δυναμικό περιβάλλον, δηλαδή ένα περιβάλλον στο οποίο οι άνθρωποι και άλλα αντικείμενα αλλάζουν θέση. Φυσικά αυτό δημιουργεί μια επιπλέον δυσκολία στην κατασκευή του χάρτη γιατί το ρομπότ μπορεί να βγάλει ευκολότερα λάθος συμπεράσματα για την θέση στην οποία βρίσκεται.

Τα περιβάλλοντα, εκτός από δυναμικά και στατικά, χωρίζονται και σε περιβάλλοντα με ή χωρίς δομή. Η 1<sup>η</sup> κατηγορία, αποτελεί κυρίως κλειστούς χώρους, διαδρόμους, κτίρια και άλλα, ενώ η 2<sup>η</sup> κατηγορία αφορά κυρίως εξωτερικούς χώρους όπως ένα δάσος ή τον βυθό της θάλασσας.

Οι περισσότεροι αλγόριθμοι χαρτογράφησης είναι πιθανοτικοί. Κάποιοι, μπορούν να κατασκευάσουν τον χάρτη σε πραγματικό χρόνο, ενώ άλλοι απαιτούν πολλαπλά περάσματα από τα δεδομένα. Επίσης, υπάρχουν αλγόριθμοι που απαιτούν ακριβή γνώση της θέσης του

ρομπότ, ενώ κάποιοι άλλοι χρειάζονται μόνο δεδομένα από την οδομετρία. Τέλος, κάποιοι μπορούν να αντιμετωπίσουν εγγενώς το πρόβλημα της αντιστοίχισης δεδομένων διαφορετικών χρονικών στιγμών, ενώ άλλοι απαιτούν τα χαρακτηριστικά του χάρτη να κρατάνε κάποιες ιδιαίτερες υπογραφές για να μπορούν να ξεχωρίζουν.

Ιστορικά, η χαρτογράφηση έχει χωριστεί σε τρεις κατηγορίες, ανάλογα με το είδος χάρτη που δημιουργεί:

- *Μετρικοί χάρτες*. Οι χάρτες αυτοί προσπαθούν να περιγράψουν τα γεωμετρικά χαρακτηριστικά του περιβάλλοντος. Συνήθως είναι χάρτες πλέγματος, δηλαδή χάρτες αποτελούμενοι από τετράγωνα, τα οποία αντιπροσωπεύουν ένα κομμάτι του χώρου όσο μικρό ή μεγάλο θέλουμε.
- *Τοπολογικοί χάρτες*. Αναπαρίστανται με κόμβους και ένα πίνακα πρόπτωσης (συνήθως). Αναπαριστούν την διασύνδεση διαφόρων σημείων του κόσμου.
- *Χάρτες οροσήμεων*. Περιγράφουν τον κόσμο με βάση ιδιαίτερα χαρακτηριστικά, όπως ακμές, γωνίες κτλ.

Επίσης, υπάρχουν και υβριδικές κατηγορίες όπως τοπολογικοί-μετρικοί χάρτες που συνδυάζουν χαρακτηριστικά των δύο κατηγοριών. Η κάθε κατηγορία έχει τα πλεονεκτήματα και τα μειονεκτήματά της. Γενικά οι μετρικοί χάρτες είναι οι πιο ακριβείς και μπορούν να γίνουν όσο ακριβείς θέλουμε, δεδομένου ότι αντέχουμε το βαρύ υπολογιστικό κόστος που συνεπάγεται από μια πολύ υψηλή ανάλυση. Από την άλλη, η απλότητα και οι μικρές απαιτήσεις σε χώρο και χρόνο των τοπολογικών χαρτών, τους καθιστούν πολύ καλές εναλλακτικές σε περίπτωση απλών εργασιών, δομημένων περιβάλλοντων και γενικότερα σε προβλήματα όπου η μεγάλη ακρίβεια δεν είναι μείζονος σημασίας.

Μια άλλη κατηγοριοποίηση των χαρτών γίνεται ανάμεσα σε *κοσμο-κεντρικούς* και *ρομπο-κεντρικούς* χάρτες. Η 1<sup>η</sup> κατηγορία αναπαριστά τον κόσμο σε κάποιο παγκόσμιο σύστημα συντεταγμένων και τα χαρακτηριστικά του χάρτη δεν περιγράφονται από τα αισθητηριακά δεδομένα που οδήγησαν στην ανακάλυψή τους, αλλά στις «παγκόσμιες» συντεταγμένες του κοσμο-κεντρικού χάρτη. Η 2<sup>η</sup> κατηγορία περιγράφει τα πάντα στο χώρο των μετρήσεων, δηλαδή με βάση τα δεδομένα από τους αισθητήρες. Όπως είναι φυσικό οι κοσμο-κεντρικοί χάρτες έχουν επικρατήσει, κυρίως διότι οι ρομπο-κεντρικοί έχουν δύο μεγάλα μειονεκτήματα: πρώτον, δεν μπορούν να συμπεράνουν την γεωμετρία του χώρου από συνεχείς μετρήσεις. Δεύτερον, δεν μπορούν να λύσουν το πρόβλημα της αντιστοίχισης δεδομένων καθώς είναι πολύ πιθανό, ίδιες αισθητηριακές μετρήσεις να αντιστοιχούν σε διαφορετικά σημεία του χώρου. Παρ'όλα αυτά, χρησιμοποιούνται σε αλγορίθμους *Υβριδικού Οπτικού SLAM*, σε συνδυασμό πάντα με κοσμο-κεντρικούς χάρτες [15].

### **2.3.1 Ερευνητικές Προκλήσεις Προβλημάτων Ρομποτικής Χαρτογράφησης.**

#### **2.3.1.1 Θόρυβος μετρήσεων**

Ίσως το σημαντικότερο πρόβλημα της χαρτογράφησης είναι η ύπαρξη θορύβου στις μετρήσεις και μάλιστα συνεπικουρούμενη από το γεγονός ότι ο θόρυβος διαφορετικών μετρήσεων δεν είναι στατιστικά ανεξάρτητος. Οποιαδήποτε μέτρηση κάνουμε κατά την

χαρτογράφηση είναι εξαρτημένη από προηγούμενες μετρήσεις. Αυτό συμβαίνει επειδή τα σφάλματα κατά την χαρτογράφηση είναι ολοκληρωτικά, δηλαδή συσσωρεύονται με αποτέλεσμα να επηρεάζουν τις μελλοντικές μετρήσεις.

### **2.3.1.2 Διαστασιμότητα**

Γενικά το πρόβλημα της χαρτογράφησης είναι ένα πρόβλημα υψηλής διαστασιμότητας (dimensionality), χρειάζεται δηλαδή ένα πολύ μεγάλο αριθμό από μεταβλητές για να περιγράψει κατάλληλα τον χώρο. Φυσικά η διαστασιμότητα εξαρτάται από την επιλογή του χάρτη, από την ακρίβειά του, και από τις διαστάσεις του χώρου που θα περιγράψουμε. Ένας τοπολογικός χάρτης μπορεί να χρειαστεί μερικές δεκάδες νούμερα για να περιγραφεί, ενώ ένας μετρικός χάρτης ενδέχεται να χρειαστεί από μερικές χιλιάδες εάν είναι δισδιάστατος, μέχρι κάποια εκατομμύρια εάν είναι τριδιάστατος.

### **2.3.1.3 Το πρόβλημα της αντιστοίχισης (Correspondence- data association problem).**

Είναι το πρόβλημα του συμπερασμού για το εάν αισθητηριακά δεδομένα διαφορετικών χρονικών στιγμών, αντιστοιχούν σε ένα σημείο του χώρου. Η αναγνώριση βρόχου στηρίζεται ακριβώς στην επίλυση αυτού του προβλήματος. Το πρόβλημα αυτό είναι εξαιρετικά δύσκολο, διότι καθώς ο χρόνος περνάει, οι διαφορετικές υποθέσεις που πρέπει κανείς να κάνει για τις πιθανές θέσεις του ρομπότ με βάση τις μετρήσεις αυξάνονται εκθετικά. Το πρόβλημα αυτό θα μας απασχολήσει ιδιαίτερα στην συνέχεια όταν θα αναφερθούμε στο ταίριασμα εικόνων.

### **2.3.1.4 Μεταβαλλόμενο περιβάλλον.**

Όπως προαναφέραμε, σε πολλές περιπτώσεις ο χώρος που το ρομπότ καλείται να χαρτογραφήσει μπορεί να μεταβάλλεται. Το πρόβλημα περιπλέκεται αφάνταστα σε μια τέτοια περίπτωση, κυρίως επειδή δίνει την δυνατότητα για μια διαφορετική εξήγηση των φαινομενικά ασυνεπών(μη αναμενόμενων) μετρήσεων. Για παράδειγμα μπορεί κανείς να φανταστεί ότι το ρομπότ συναντά μια ανοιχτή πόρτα που προηγουμένως την είχε συναντήσει κλειστή. Αυτό έχει δύο εξηγήσεις: ή πρόκειται για την ίδια πόρτα που όμως πλέον είναι ανοιχτή, ή πρόκειται για μια εντελώς νέα, ανοιχτή πόρτα και συνεπώς το ρομπότ δεν είναι τελικά εκεί που νόμιζε ότι είναι [16].



### 2.3.2 Αλγόριθμοι Χαρτογράφησης

Στην βιβλιογραφία, υπάρχει ένα πλήθος αλγορίθμων που χρησιμοποιούνται για την χαρτογράφηση. Θα μπορούσαμε να κατηγοριοποιήσουμε τους αλγορίθμους είτε με βάση τα μαθηματικά εργαλεία που χρησιμοποιούν, είτε με βάση τον τρόπο που αναπαριστούν τον κόσμο, δηλαδή το είδος χάρτη που φτιάχνουν. Θα ξεκινήσουμε με βάση τον τρόπο αναπαράστασης του περιβάλλοντος :

#### 2.3.2.1 Τοπολογική Προσέγγιση

Οι προσεγγίσεις αυτές βασίζονται στην δημιουργία ενός τοπολογικού χάρτη. Οι λόγοι για τους οποίους φτιάχνουμε τοπολογικό αντί για μετρικό χάρτη είναι ποικίλοι.

- Αντιμετωπίζουμε ένα έντονα δομημένο περιβάλλον, σχετικά απλό στην δομή του.
- Δεν έχουμε τους απαραίτητους αισθητήρες, όπως laser finders, sonar, infrared κτλ, προκειμένου να κατασκευάσουμε μετρικό χάρτη.
- Δεν έχουμε την απαραίτητη υπολογιστική ισχύ για να αντιμετωπίσουμε το βαρύ φορτίο που απαιτεί η κατασκευή ενός ακριβέστερου αλλά πιο σύνθετου μετρικού χάρτη.
- Χρειαζόμαστε μια «αδρή» (υψηλού επιπέδου) περιγραφή του χώρου, σε μορφή διασυνδεδεμένων «περιοχών-κόμβων» για λόγους εφαρμογής αποδοτικών μεθόδων καθολικού σχεδιασμού δρόμου (mission – path planning).

Μια ιδιαίτερη προσέγγιση περιγράφεται στο [13], όπου δημιουργείται ένας τοπολογικός χάρτης βασισμένος σε τοπικά χαρακτηριστικά της εικόνας. Η εργασία αυτή εμπεριέχει μέσα στον αλγόριθμο έννοιες βραχυχρόνιας και μακροχρόνιας μνήμης για να αντιμετωπίζει αλλαγές του περιβάλλοντος. Στο [17], δεδομένα από τους αισθητήρες καταγράφονται και προβάλλονται σε ένα χώρο. Στη συνέχεια με χρήση μη επιβλεπόμενης μάθησης, γίνεται μια ομαδοποίηση των δεδομένων αυτών. Κάθε ομάδα αποτελεί ένα κόμβο του τοπολογικού χάρτη. Το [18], φτιάχνει ένα αυξητικό τοπολογικό χάρτη, δηλαδή ένα χάρτη που εξαπλώνεται σε πραγματικό χρόνο καθώς το ρομπότ εξερευνά τον χώρο. Το μοντέλο είναι βασισμένο στους νόμους της φυσικής. Κάθε εικόνα που λαμβάνεται, αποτελεί ένα σωματίδιο το οποίο συνδέεται με «ελατήρια» με γειτονικά του σωματίδια. Το μήκος του ελατηρίου καθορίζεται από την Ευκλείδεια απόσταση των υπογραφών Fourier των γειτονικών εικόνων.

#### 2.3.2.2 Προσέγγιση Πλέγματος Κατάληψης

Τέτοιοι αλγόριθμοι όπως φαίνονται στο [1], αναπαριστούν το χώρο ως ένα πλέγμα οποιασδήποτε ακρίβειας επιθυμούμε. Το πλέγμα αποτελείται από τετράγωνα, τα οποία γενικά χαρακτηρίζονται από δύο πιθανότητες: πιθανότητα να είναι άδειο, και πιθανότητα να είναι κατειλημμένο. Τέτοιοι αλγόριθμοι απαιτούν αισθητήρες sonar ή laser. Στηρίζονται σε φίλτρα Bayes για να υπολογίσουν την a-posteriori πιθανότητα του πλέγματος. Είναι πολύ δημοφιλείς και απλοί στην κατασκευή τους [16]. Παρ'όλα, αυτά σε μεγάλα περιβάλλοντα, τα σφάλματα

διακριτοποίησης, οι ανάγκες σε αποθηκευτικό χώρο, και ο χρόνος επεξεργασίας αποκτούν καίρια σημασία και μειώνουν την αποτελεσματικότητά τους.

### 2.3.2.3 Προσέγγιση Χαρακτηριστικών.

Στην περίπτωση αυτή, το περιβάλλον μοντελοποιείται με βάση χαρακτηριστικά όπως ακμές, γωνίες ή άλλα χαρακτηριστικά. Ίσως η πιο σημαντική εργασία στον χώρο αυτό περιγράφεται στο [19]. Στην εργασία αυτή, δίνεται μια μαθηματική περιγραφή ενός χάρτη χαρακτηριστικών: ο χάρτης αυτός, που αποτελείται από ένα σύνολο χαρακτηριστικών, περιγράφεται από μια μέση τιμή, δηλαδή το πού πιστεύουμε ότι βρίσκονται τα χαρακτηριστικά σημεία του περιβάλλοντος και από ένα πίνακα συμμεταβλητότητας που εκφράζει το βαθμό της αβεβαιότητας που χαρακτηρίζει την παραπάνω μέση τιμή. Η εργασία αυτή χρησιμοποιεί ένα EKF προκειμένου να συνθέσει δεδομένα οδομετρίας και αισθητήρων και να φτιάξει ένα χάρτη με όσο το δυνατόν μικρότερη αβεβαιότητα. Το [20], στην ίδια λογική, χρησιμοποιεί τρία είδη χαρακτηριστικών για την μοντελοποίηση του περιβάλλοντος: ένα σημείο, μια γραμμή και ένα κύκλο. Κάθε ένα από αυτά τα χαρακτηριστικά έχει το δικό του διάνισμα που το περιγράφει. Και πάλι εδώ χρησιμοποιείται ένα EKF.

Τα προβλήματα του Εκτεταμένου φίλτρου Κάλμαν, δηλαδή η τετραγωνική ως προς τα χαρακτηριστικά πολυπλοκότητα  $O(K^2)$  ( $K$ = αριθμός χαρακτηριστικών) και η αδυναμία αντιμετώπισης του προβλήματος της αντιστοίχισης πληροφορίας, οδήγησαν στην ανάπτυξη νέων μεθόδων. Μια από αυτές, η μέθοδος FastSLAM που βασίζεται στο Rao-Blackwellized Particle Filter, περιγράφεται στα [21], [22]. Η ιδέα είναι απλή: υποθέτουμε ότι έχουμε  $M$  δυνατές θέσεις (σωματίδια) του ρομπότ στον χώρο, κάθε μια από τις οποίες εξελίσσεται ανεξάρτητα από τις άλλες και η οποία περιγράφει  $K$  ορόσημα του χώρου. Κάθε σωματίδιο διατηρεί την δική του εκτίμηση για την θέση των οροσήμων, ανεξάρτητα από τα άλλα σωματίδια. Έτσι για  $M$  θέσεις και  $K$  χαρακτηριστικά, θα έχουμε  $M*K$  φίλτρα Κάλμαν διάστασης 2. Αυτό κάνει την πολυπλοκότητα  $O(MK)$  αλλά έχει αποδειχτεί στην πράξη είναι  $O(M \log K)$ .

## 2.4 Μέθοδοι Επίλυσης του SLAM

Στην παρούσα ενότητα, αφού ορίσουμε το πρόβλημα του SLAM, θα αναφερθούμε στις κύριες προσεγγίσεις του SLAM από την διεθνή βιβλιογραφία.

Στην Ενότητα 2.4.1 θα ορίσουμε μαθηματικά το πρόβλημα του SLAM και θα εξηγήσουμε διαισθητικά το πώς ο θόρυβος το καθιστά ένα ιδιαίτερα δύσκολο πρόβλημα.

Στις Ενότητες 2.4.2 - 2.4.5 θα εξετάσουμε τις πιο σημαντικές προσεγγίσεις για την επίλυση του SLAM.

### 2.4.1 Ορισμός του SLAM

Το πρόβλημα του SLAM μπορεί να περιγραφεί ως μια πιθανοτική αλυσίδα Markov. Ορίζουμε την θέση του ρομπότ τη στιγμή  $t$ , ως  $s_t$ . Εάν δουλεύουμε στο επίπεδο, η θέση ορίζεται ως ένα σημείο  $x, y$  πάνω στους άξονες και ως κατεύθυνση. Το περιβάλλον του ρομπότ υποτίθεται ότι αποτελείται από  $K$  ακίνητα ορόσημα κάθε ένα από τα οποία χαρακτηρίζεται από την θέση του  $\theta_i$  στο χώρο. Συμβολίζουμε με  $\theta$  το σύνολο των οροσήμων του χάρτη μας.

Η θέση του ρομπότ γενικά εξελίσσεται με βάση κάποιο πιθανοτικό μοντέλο το οποίο ονομάζεται *μοντέλο κίνησης*:

$$p(s_t | u_t, s_{t-1}) \quad (1)$$

Για την χαρτογράφηση του περιβάλλοντος, το ρομπότ εντοπίζει ορόσημα. Έστω  $z_t$  η παρατήρηση την χρονική στιγμή  $t$ , η οποία γενικά χαρακτηρίζεται από το *μοντέλο παρατήρησης*:

$$p(z_t | s_t, \theta_{n_t}, n_t) \quad (2)$$

Όπου  $n_t$  είναι ο δείκτης του ορόσημου που αντιλαμβάνεται το ρομπότ.

Με λίγα λόγια, ως SLAM ορίζουμε το πρόβλημα του εντοπισμού της θέσης όλων των οροσήμων  $\theta$  και θέσεων του ρομπότ  $s_t$  από τις μετρήσεις  $z^t = z_1, z_2, \dots, z_t$  και τους ελέγχους  $u^t = u_1, u_2, \dots, u_t$ . Αυτό μπορεί να εκφραστεί και ως εξής:

$$p(s^t, \theta | z^t, u^t) \quad (3)$$

Ένα σημαντικό ζήτημα του SLAM είναι η *συσχέτιση της πληροφορίας (Data Association)*. Στην πράξη, η αντιστοίχιση  $n^t = [n_1, n_2, \dots, n_t]$  μεταξύ των παρατηρήσεων και των οροσήμων είναι σπανίως γνωστή. Έτσι, όποτε γίνεται μια νέα παρατήρηση, αυτή πρέπει είτε να αντιστοιχιστεί με ένα υπάρχον ορόσημο, είτε να θεωρηθεί ότι προέρχεται από κάποιο νέο ορόσημο. Εάν αυτή η αντιστοίχιση δεν είναι προφανής, τότε η επιλογή της λανθασμένης συσχέτισης μπορεί να οδηγήσει σε καταστροφικά λάθη.

Γενικά δύο παράγοντες συνεισφέρουν στην αβεβαιότητα στο πρόβλημα του SLAM: ο *θόρυβος μετρήσεων* και ο *θόρυβος κίνησης* [22]. Και οι δύο οδηγούν σε διαφορετικού είδους αμφισημίες κατά την συσχέτιση της πληροφορίας. Θόρυβος στο μοντέλο παρατήρησης θα οδηγήσει σε μεγαλύτερη αβεβαιότητα στη θέση των οροσήμων. Αυτό με τη σειρά του οδηγεί σε αμφισημία μέτρησης, δηλαδή στην σύγχυση μεταξύ γειτονικών οροσήμων. Λάθη που οφείλονται σε αμφισημίες παρατήρησης έχουν συνήθως μικρή επίδραση στο σφάλμα εκτίμησης, γιατί η παρατήρηση θα μπορούσε κάλλιστα να προέρχεται από διαφορετικό ορόσημο.

Αντίθετα η αμφισημία που προέρχεται από τον θόρυβο κίνησης θα έχει πολύ χειρότερα αποτελέσματα. Υψηλός θόρυβος στην κίνηση του ρομπότ θα οδηγήσει σε μεγαλύτερη

αβεβαιότητα σχετικά με την θέση του ρομπότ. Εάν η αβεβαιότητα αυτή, αυξηθεί σε μεγάλο βαθμό, τότε είναι δυνατόν διαφορετικές πιθανές θέσεις του ρομπότ, να οδηγήσουν σε τελείως διαφορετικές συσχετίσεις πληροφορίας κατά τις επακόλουθες παρατηρήσεις. Εάν ο αλγόριθμος SLAM επιλέξει λάθος συσχέτιση για κάποιο ορόσημο λόγω αβεβαιότητας θέσης, είναι σχεδόν βέβαιο ότι και οι υπόλοιπες συσχετίσεις θα είναι λάθος.

Με βάση τα παραπάνω θα δούμε τώρα τις βασικές προσεγγίσεις που επιλύουν τα παραπάνω προβλήματα.

## 2.4.2 Το φίλτρο Kalman

Η πιο παλιά και κλασική προσέγγιση είναι βασισμένη στο *Φίλτρο Kalman* [23] και είναι άρρηκτα συνδεδεμένη με το πρόβλημα του SLAM.

Το φίλτρο Kalman είναι ένα φίλτρο Bayes που περιγράφει πιθανότητες  $p(s_t, m | z^t, u^t)$  χρησιμοποιώντας Gaussian κατανομές. Εδώ το Gaussian μοντέλο είναι το διάνυσμα  $x$ , το οποίο περιέχει την θέση του ρομπότ  $s$  και τον χάρτη  $m$ , δηλαδή:

$$x_t = (s_t, m)^T \quad (4)$$

Εν γένει η θέση του ρομπότ στο επίπεδο, ορίζεται από τρεις αριθμούς: δύο για τις Καρτεσιανές συντεταγμένες και ένα για τον προσανατολισμό στο επίπεδο, συνεπώς  $s = (s_x, s_y, s_\theta)$ .

Από την άλλη ο χάρτης περιγράφεται από τις Καρτεσιανές συντεταγμένες των χαρακτηριστικών του, όπως αυτά ορίζονται (για παράδειγμα αντικείμενα, γωνίες κτλ.). Εάν έχουμε  $K$  τέτοια ορόσημα, τότε το διάνυσμα κατάστασης έχει μέγεθος  $2K+3$ :

$$x_t = (s_{x,t}, s_{y,t}, s_{\theta,t}, m_{1,x,t}, m_{2,x,t}, \dots, m_{k,x,t}, m_{k,y,t})^T \quad (5)$$

Συνήθως, η Gaussian που περιγράφει την κατανομή του διανύσματος κατάστασης περιγράφεται από μια μέση τιμή  $\mu$  και ένα πίνακα συμμεταβλητότητας  $\Sigma$ .

Το φίλτρο Kalman βασίζεται πάνω σε τρεις υποθέσεις:

1. Πρώτον, ότι η επόμενη κατάσταση είναι γραμμική με Gaussian θόρυβο.
2. Δεύτερον, ότι το ίδιο ισχύει και για το μοντέλο αντίληψης-παρατήρησης.
3. Τρίτον, ότι η αρχική κατάσταση έχει Gaussian αβεβαιότητα.

Η 1<sup>η</sup> υπόθεση μας λέει ότι η κατάσταση  $s_t$  και  $m_t$  εξαρτώνται γραμμικά από τις προηγούμενες καταστάσεις  $s_{t-1}$ ,  $m_{t-1}$ . Για τον χάρτη που θεωρούμε ότι δεν αλλάζει αυτό προφανώς ισχύει. Για τη θέση του ρομπότ όμως, κάτι τέτοιο δεν είναι εν γένει σωστό, αφού εμπλέκονται τριγωνομετρικές και άρα μη γραμμικές συναρτήσεις που εμπλέκουν και την είσοδο  $u_t$ . Έτσι οδηγούμαστε σε μια γραμμικοποίηση χρησιμοποιώντας ανάπτυγμα Taylor, που μας δίνει το Εκτεταμένο Φίλτρο Κάλμαν. Έτσι έχουμε:

$$p(x | u, x') = Ax' + Bu + \varepsilon_{control} \quad (6)$$

Θεωρώντας αντίστοιχη σχέση και για το μοντέλο παρατήρησης, έχουμε το ακόλουθο ενδιαφέρον αποτέλεσμα: Η σχέση του Bayes για την a posteriori πιθανότητα

$$p(s_t, m | z^t, u^t) = \eta p(z_t | s_t, m) \int p(s_t | u_t, s_{t-1}) p(s_{t-1}, m | z^{t-1}, u^{t-1}) ds_{t-1} \quad (7)$$

είναι ισοδύναμη με τις γνωστές σχέσεις του φίλτρου Κάλμαν, δηλαδή:

$$\begin{aligned} \mu_{t-1}' &= \mu_{t-1} + Bu_t \\ \Sigma_{t-1}' &= \Sigma_{t-1} + \Sigma_{control} \\ K_t &= \Sigma_{t-1}' C^T (C \Sigma_{t-1}' C^T + \Sigma_{measure})^{-1} \\ \mu_t &= \mu_{t-1}' + K_t (z - C \mu_{t-1}') \\ \Sigma_t &= (I - K_t C) \Sigma_{t-1}' \end{aligned} \quad (8)$$

Στην πράξη, ο αριθμός των ορόσημων του χάρτη κυμαίνεται από μερικές δεκάδες μέχρι μερικές εκατοντάδες. Ο αριθμός των οροσήμων που διατηρούνται στο χάρτη, μπορεί να μένει σταθερός ή να μεγαλώνει δυναμικά καθώς νέες περιοχές εξερευνούνται.

Το φίλτρο Κάλμαν έχει ένα πολύ σημαντικό περιορισμό. Εκτός ότι υποθέτει Gaussian θόρυβο και γενικότερα Gaussian κατανομές, υποστηρίζει μόνο κατανομές με ένα μόνο μέγιστο. Αυτός ο μαθηματικός περιορισμός που επιβάλλει το Φίλτρο Κάλμαν είναι ιδιαίτερα αυστηρός, ιδιαίτερα σε περιπτώσεις όπου δύο διαφορετικά ορόσημα δεν μπορούν να διαχωριστούν (για παράδειγμα δύο ίδιες πόρτες σε διαφορετικά σημεία του χώρου). Μια τέτοια μέτρηση θα δημιουργήσει μια κατανομή η οποία θα έχει μέγιστα σε τουλάχιστον δύο σημεία του χώρου, οπότε η υπόθεση Κάλμαν για απλή (με ένα μέγιστο) Gaussian κατανομή, καταρρέει. Συνεπώς το πρόβλημα της *αντιστοίχισης (correspondence)* δεν αντιμετωπίζεται από το φίλτρο Κάλμαν.

Το παραπάνω ζήτημα, εγείρει και άλλα προβλήματα. Μια αποτελεσματική υλοποίηση του φίλτρου Κάλμαν απαιτεί ένα αραιό σύνολο οροσήμων, που θα είναι επαρκώς διαχωρίσιμα και αναγνωρίσιμα είτε λόγω θέσης, είτε λόγω αισθητηριακών δεδομένων. Διαφορετικά, λάθη στην αναγνώριση οδηγούν σε ανεπανόρθωτη αστοχία του αλγορίθμου χαρτογράφησης δεδομένου ότι δεν υπάρχει δυνατότητα για πίσω βήματα. Για αυτό τον λόγο, αλγόριθμοι βασισμένοι στο φίλτρο Κάλμαν, συνήθως αγνοούν ένα μεγάλο μέρος των αισθητηριακών δεδομένων, και δίνουν σαν αποτέλεσμα την θέση των οροσήμων, χωρίς όμως ιδιαίτερη γεωμετρική-μετρική πληροφορία για το περιβάλλον.

Τα προβλήματα του παραπάνω αλγορίθμου οδήγησαν σε διάφορες βελτιώσεις, που στόχευαν κυρίως στο να λύσουν το πρόβλημα της αντιστοίχισης. Χαρακτηριστική προσπάθεια είναι ο αλγόριθμος Lu/Milios [24]. Βασίζεται σε δύο βήματα. Στο πρώτο, το φίλτρο Κάλμαν υπολογίζει τις a posteriori πιθανότητες για το διάλυσμα κατάστασης, και στο δεύτερο, όπου πολλαπλές μετρήσεις από laser συσχετίζονται μεταξύ τους. Η συσχέτιση γίνεται με βάση τη μέγιστη πιθανοφάνεια των δεδομένων, δηλαδή ο αλγόριθμος απλά ζευγαρώνει γειτονικές μετρήσεις. Καθώς η μια φάση διαδέχεται την άλλη, η αντιστοίχιση υπολογίζεται

επαναληπτικά, δίνοντας τη δυνατότητα επαναφοράς από λάθος αντιστοιχίσεις. Όμως το γεγονός ότι χρησιμοποιεί την μέγιστη πιθανοφάνεια για να μαντέψει τις αντιστοιχίσεις, αντί να υπολογίζει πλήρως την a posteriori πιθανότητα, επιβάλλει διάφορους περιορισμούς. Στην πράξη, ο αλγόριθμος δουλεύει πολύ καλά όταν τα σφάλματα στις αρχικές επαναλήψεις είναι μικρά (<2 μέτρα). Μεγάλα σφάλματα, όπως αυτά που συναντώνται σε κυκλικά περιβάλλοντα, οδηγούν σε λάθος αποτελέσματα. Επίσης, ο αλγόριθμος απαιτεί πολλαπλά περάσματα από τα δεδομένα, οπότε δεν μπορεί να χρησιμοποιηθεί σε πραγματικό χρόνο.

### 2.4.3 Αλγόριθμοι Μεγιστοποίησης Προσδοκίας (Expectation Maximization Algorithms)

Αυτή η οικογένεια αλγορίθμων, αναπτύχθηκε ως εναλλακτική του φίλτρου Κάλμαν και περιγράφεται αναλυτικά στο [25]. Βασίζονται στην υπολογισμό της μεγιστοποίησης της πιθανοφάνειας με έμμεσες(latent)-όχι άμεσα παρατηρήσιμες μεταβλητές. Το πλεονέκτημα αυτών των αλγορίθμων είναι ότι αντιμετωπίζουν ιδιαίτερα αποτελεσματικά το πρόβλημα της αντιστοίχισης ακόμα και σε ιδιαίτερα δύσκολα περιβάλλοντα, όπου όλα σχεδόν τα χαρακτηριστικά είναι ίδια και δεν μπορούν να διαχωριστούν με μια ματιά.

Οι αλγόριθμοι αυτοί δεν έχουν το στοιχείο της αβεβαιότητας στη μορφή του φίλτρου Κάλμαν, δηλαδή χρησιμοποιώντας Gaussian πιθανότητες. Αντίθετα εφαρμόζουν μια αναρρίχηση λόφου (hill climbing) στον χώρο των χαρτών, ψάχνοντας τον πιο πιθανό χάρτη. Αυτό προϋποθέτει πολλαπλά περάσματα από τα δεδομένα, συνεπώς ο χάρτης δεν κατασκευάζεται αυξητικά όπως στα φίλτρα Κάλμαν.

Οι αλγόριθμοι EM(expectation minimization), εκμεταλλεύονται κάτι που αναφέραμε νωρίτερα: το SLAM έχει το χαρακτηριστικό του ερωτήματος «η κότα έκανε το αυγό ή το αυγό την κότα;». Αυτό σημαίνει ότι εκμεταλλεύονται από τη μία ότι η κατασκευή του χάρτη είναι σχετικά απλή υπόθεση εάν η τροχιά του ρομπότ είναι γνωστή, και από την άλλη ότι ο πιθανοτικός υπολογισμός της θέσης του ρομπότ είναι εφικτός εάν ο χάρτης είναι δεδομένος. Με βάση αυτή την παρατήρηση ο αλγόριθμος EM δουλεύει σε δύο αλληλοδιαδεχόμενα βήματα: το *βήμα προσδοκίας* (expectation step) ή E-step, όπου η a posteriori πιθανότητα της θέσης του ρομπότ υπολογίζεται με δεδομένο τον χάρτη, και το *βήμα μεγιστοποίησης* (maximization step) ή M-step, όπου υπολογίζεται ο πιθανότερος χάρτης, με δεδομένες αυτές τις προσδοκίες της θέσης. Το αποτέλεσμα είναι μια σειρά από όλο και πιο λεπτομερείς και σωστούς χάρτες  $m^{[0]}, m^{[1]}, \dots$ , όπου ο αρχικός χάρτης  $m^{[0]}$  είναι κενός.

#### 2.4.3.1 E-step

Τυπικά, η συνάρτηση που μεγιστοποιείται είναι η προσδοκία(μέση τιμή) πάνω στην συνδυασμένη λογαριθμική πιθανοφάνεια των δεδομένων  $d^t$  και της τροχιάς του ρομπότ  $s^t = \{s_1, \dots, s_t\}$ :

$$m^{[i+1]} = \arg \max_m E_{s^t} [\log p(d^t, s^t | m) | m^{[i]}, d^t] \quad (9)$$

η οποία προκύπτει από την σχέση του φίλτρου Bayes που αναφέραμε πιο πριν. Όπως φαίνεται, μέρος της πιθανοφάνειας που μεγιστοποιείται, είναι και η τροχιά του ρομπότ,  $s^t$ . Όμως επειδή στην χαρτογράφηση (βήμα E) η τροχιά είναι άγνωστη, η παραπάνω εξίσωση υπολογίζει την μέση τιμή γύρω από όλα τα δυνατά μονοπάτια του ρομπότ. Έτσι, ουσιαστικά προκύπτει η ακόλουθη σχέση:

$$m^{[i+1]} = \arg \max_m \sum_{\tau} \int p(s_{\tau} | m^{[i]}, d^t) \log p(z_{\tau} | s_{\tau}, m) ds_{\tau} \quad (10)$$

Ο όρος  $p(s_{\tau} | m^{[i]}, d^t)$ , είναι η δεσμευμένη πιθανότητα της θέσης  $s^t$  με δεδομένα, την πληροφορία  $d^t$  και τον  $i$ -οστό χάρτη  $m^{[i]}$ . Το ενδιαφέρον εδώ, είναι ότι για τον υπολογισμό της θέσης την στιγμή  $\tau$ , χρησιμοποιούμε πληροφορία από όλο το διάστημα  $\{1, \dots, t\}$ , ακόμα και όπου  $t > \tau$ .

Έχει αποδειχθεί ότι ένας τέτοιος υπολογισμός μπορεί να γίνει σε δύο βήματα: το πρώτο ξεκινά για χρόνο από  $\{0, \dots, \tau\}$ , υπολογίζει την  $p(s_{\tau} | m^{[i]}, d^{\tau})$ . Το δεύτερο πηγαίνει πίσω στον χρόνο δηλαδή υπολογίζει την  $p(s_{\tau} | m^{[i]}, d^{\tau+1}, \dots, d^t)$ . Εάν πολλαπλασιάσουμε και κανονικοποιήσουμε τις δύο πιθανότητες, τότε παίρνουμε την τελική πιθανότητα  $p(s_{\tau} | m^{[i]}, d^t)$ .

### 2.4.3.2 M-step

Το βήμα αυτό ουσιαστικά αποτελείται από την μεγιστοποίηση της (10). Γνωρίζοντας πλέον όλες τις πιθανότητες  $p(s_{\tau} | m^{[i]}, d^t)$  από το προηγούμενο βήμα, θέλουμε να μεγιστοποιήσουμε τον λογάριθμο της πιθανότητας των αισθητηριακών δεδομένων  $\log p(z_{\tau} | s_{\tau}, m)$  για όλα τα  $\tau$ , και όλες τις πιθανές θέσεις  $s^t$ . Δυστυχώς δεν υπάρχει κλειστή λύση για ένα τέτοιο πρόβλημα τόσο πολλών διαστάσεων. Μια λύση είναι να λύσουμε το πρόβλημα για κάθε σημείο  $(x, y)$  ανεξάρτητα. Αυτό προϋποθέτει τον κβαντισμό του χάρτη σε ένα πεπερασμένο αριθμό περιοχών, για παράδειγμα ένα πλέγμα. Ο χάρτης που προκύπτει μπορεί να είναι είτε ντετερμινιστικός με την έννοια ότι κάθε σημείο παίρνει τιμές 0 ή 1, είτε πιθανοτικός.

Ο αλγόριθμος EM έχει το τεράστιο πλεονέκτημα ότι αντιμετωπίζει το πρόβλημα της αντιστοιχίσης. Αυτό το επιτυγχάνει με το να εντοπίζει ξανά και ξανά την θέση του ρομπότ σε σχέση με τον χάρτη στο βήμα E. Κάθε διαφορετική θέση που εξετάζεται ουσιαστικά αντιστοιχεί σε διαφορετικές υποθέσεις σε σχέση με την θέση του ρομπότ και άρα υπονοεί διαφορετικές αντιστοιχίσεις. Στη συνέχεια στο βήμα M αυτές οι αντιστοιχίσεις μεταφράζονται σε χαρακτηριστικά του χάρτη.

#### 2.4.4 Υβριδικές Μέθοδοι

Οι υβριδικές μέθοδοι προσπαθούν να απλοποιήσουν το πρόβλημα, δανειζόμενες χαρακτηριστικά τόσο από το φίλτρο Κάλμαν, όσο και από τον αλγόριθμο EM. Η βασική ιδέα είναι η κατασκευή του χάρτη σε βήματα, καθώς λαμβάνονται νέα δεδομένα από τους αισθητήρες, χωρίς να κρατάμε στοιχεία αβεβαιότητας. Ουσιαστικά αυτό είναι ίδιο με το βήμα M του αλγορίθμου EM, χωρίς να έχει όμως προηγηθεί το βήμα E.

Μαθηματικά, αυτό που γίνεται είναι να δημιουργούμε μια ακολουθία από βέλτιστους χάρτες  $m_1^*, m_2^*, \dots$ , μαζί με μια ακολουθία από τις πιθανότερες θέσεις  $s_1^*, s_2^*, \dots$ . Ο t-οστός χάρτης και η αντίστοιχη θέση κατασκευάζονται από τον t-1 χάρτη και θέση, μέσω της ακόλουθης σχέσης:

$$\langle m_t^*, s_t^* \rangle = \arg \max_{m_t, s_t} p(z_t | s_t, m_t) p(s_t, m_t | u_t, s_{t-1}^*, m_{t-1}^*) \quad (11)$$

, η οποία προκύπτει από την βασική σχέση του φίλτρου Bayes:

$$p(s_t, m_t | z', u') = \eta p(z_t | s_t, m_t) \int \int p(s_t, m_t | u_t, s_{t-1}, m_{t-1}) p(s_{t-1}, m_{t-1} | z^{t-1}, u^{t-1}) ds_{t-1} dm_{t-1} \quad (12)$$

, εάν υποθέσουμε ότι ο (t-1) χάρτης και θέση, είναι γνωστά.

Πρακτικά αυτό που χρειάζεται κάθε φορά που λαμβάνονται νέα δεδομένα είναι να ψάξει κανείς σε όλες τις θέσεις και να βρει αυτές που μεγιστοποιούν την παραπάνω πιθανότητα. Το γεγονός ότι δεν διατηρεί κάποια αβεβαιότητα, συνεπάγεται ότι όταν σε κάποιο βήμα βρεθεί ο βέλτιστος χάρτης και θέση, αυτά πλέον θεωρούνται σταθερά για πάντα και δεν μπορούν να αλλάξουν λόγω για παράδειγμα μελλοντικών δεδομένων. Επομένως, κυκλικά περιβάλλοντα δεν μπορούν να χαρτογραφηθούν.

Παρ'όλα αυτά έχουν δημιουργηθεί αλγόριθμοι που ξεπερνούν αυτό το πρόβλημα. Η κύρια ιδέα είναι να φτιάχουν βήμα-βήμα τον χάρτη, διατηρώντας όμως παράλληλα και μια κατανομή για τις πιθανές θέσεις του ρομπότ, αντίθετα με τις προηγούμενες υβριδικές μεθόδους, που θεωρούσαν δεδομένη την θέση του ρομπότ και τον χάρτη. Έτσι όταν το ρομπότ ανακαλύψει ένα κυκλικό περιβάλλον, χρησιμοποιεί αυτή την αβεβαιότητα για να επαναπροσδιορίσει την θέση του στο χάρτη. Έχοντας πλέον την θέση, δεν έχει παρά να εξαπλώσει το σφάλμα ομοιόμορφα στον κύκλο του χάρτη. Η μέθοδος αυτή έχει το πλεονέκτημα ότι διατηρεί μόνο ένα χάρτη κάτι που είναι καλύτερο υπολογιστικά, ενώ παράλληλα έχει την δυνατότητα να διορθώνει προς τα πίσω τον χάρτη, όταν εντοπίσει κάποια ασυμβατότητα. Δυστυχώς, παρουσιάζει μερικά μειονεκτήματα: για παράδειγμα η διόρθωση του χάρτη προς τα πίσω μπορεί εάν γίνει λάθος να αποδειχτεί καταστροφική. Ακόμα, εμπολεωμένοι βρόγχοι, που δημιουργούν πολύ σύνθετες αμφισημίες δεν μπορούν να αντιμετωπιστούν. Τέλος, ο αλγόριθμος δεν είναι απαραίτητα εκτελέσιμος σε πραγματικό χρόνο καθώς εξαρτάται από το μέγεθος του βρόγχου.



#### 2.4.5 SLAM με Φίλτρο Σωματιδίων

Το φίλτρο σωματιδίων αποτελεί μια εναλλακτική πρόταση στους αλγορίθμους που βασίζονται στο EKF. Χαρακτηριστική εργασία στον χώρο αυτό είναι η μέθοδος FastSLAM που περιγράφεται στο[22]. Η ιδέα είναι να εντάξουμε το πρόβλημα του SLAM σε ένα πρόβλημα αυτοενοπισμού θέσης και εντοπισμού  $K$  ανεξάρτητων οροσίων δεσμευμένα από την εκτίμηση θέσης του ρομπότ.

Στην μέθοδο αυτή ξεκινάμε από κάποιο αριθμό σωματιδίων που αναπαριστούν διάφορες πιθανές θέσεις του ρομπότ στον χώρο κατάστασης. Το κάθε σωματίδιο περιέχει  $K$  ανεξάρτητα φίλτρα Κάλμαν που προσδιορίζουν την εκτίμηση θέσης των οροσίων με βάση την τροχιά τους. Επειδή κάθε σωματίδιο αντιπροσωπεύει και μια διαφορετική εκτίμηση θέσης του ρομπότ, η συσχέτιση των δεδομένων μπορεί να αντιμετωπιστεί ξεχωριστά για κάθε σωματίδιο. Αυτό προσφέρει δύο πλεονεκτήματα. Πρώτον, ο θόρυβος από την κίνηση του ρομπότ δεν επηρεάζει την ακρίβεια της συσχέτισης δεδομένων. Δεύτερον, εάν παρατηρήσεις συσχετίζονται σωστά σε κάποια σωματίδια και λανθασμένα σε κάποια άλλα, τότε τα λανθασμένα σωματίδια θα λάβουν μικρότερες πιθανότητες και δεν θα επιλεγούν στην επόμενη δειγματοληψία. Με αυτό τον τρόπο ο αλγόριθμος του φίλτρου σωματιδίων λειτουργεί όπως οι εξελικτικοί αλγόριθμοι, βελτιώνοντας τα καλά δείγματα, και πετώντας τα λανθασμένα.

Ας δούμε λίγο πιο μαθηματικά το σκεπτικό. Κάθε χρονική στιγμή, διατηρείται ένα σύνολο σωματιδίων  $S_t$  που περιγράφουν την κατανομή  $p(s^t | z^t, u^t)$ . Κάθε σωματίδιο  $s^{t,[m]} \in S_t$ , αποτελεί μια υπόθεση για την τροχιά του ρομπότ:

$$S_t = \{s^{t,[m]}\}_m = \{s_1^{[m]}, \dots, s_t^{[m]}\}_m \quad (13)$$

Το σύνολο  $S_t$  υπολογίζεται σε κάθε βήμα από το σύνολο  $S_{t-1}$ , την είσοδο ελέγχου  $u_t$ , και την μέτρηση  $z_t$ . Αρχικά, κάθε σωματίδιο  $s^{t,[m]} \in S_{t-1}$  χρησιμοποιείται για την εκτίμηση της θέσης του ρομπότ τη στιγμή  $t$ :

$s_t^{[m]} \sim p(s_t | u_t, s_{t-1}^{[m]})$ , το οποίο πρακτικά προκύπτει αν δειγματοληπτίσουμε το μοντέλο κίνησης του ρομπότ. Στη συνέχεια αυτή η εκτίμηση προστίθεται σε ένα σύνολο προσωρινών σωματιδίων. Αν υποθέσουμε ότι το σύνολο  $S_{t-1}$  ακολουθεί κατανομή  $p(s^{t-1} | z^{t-1}, u^{t-1})$  τότε το νέο σωματίδιο που προκύπτει θα έχει κατανομή:

$p(s^t | z^{t-1}, u^t)$ , που ονομάζεται *κατανομή πρότασης (proposal distribution)*.

Αφού δημιουργηθούν  $M$  σωματίδια με τον παραπάνω τρόπο, στη συνέχεια το νέο σύνολο  $S_t$  προκύπτει από δειγματοληψία (με αντικατάσταση) από το προσωρινό σύνολο. Κάθε σωματίδιο  $s^{t,[m]} \in S_t$  επιλέγεται με μια πιθανότητα ανάλογη με ένα συντελεστή-βάρος:

$$w_t^{[m]} = \frac{\text{target dist.}}{\text{proposal dist.}} = \frac{p(s^{t,[m]} | z^t, u^t)}{p(s^{t,[m]} | z^{t-1}, u^t)} \quad (14)$$

Τα σωματίδια αυτά δεν είναι τίποτε άλλο από μια δειγματοληψία της κατανομής πιθανότητας που προκύπτει από το μοντέλο κίνησης του ρομπότ.

Πέρα από την εκτίμηση θέσης που μόλις περιγράψαμε, πρέπει κανείς να βρει και την θέση των οροσήμεων. Η μέθοδος FastSLAM αναπαριστά στην υπό συνθήκη εκτίμηση της θέσης αυτής ως  $p(\theta_i | s^t, z^t, u^t, n^t)$  και την υπολογίζει με τη βοήθεια φίλτρων Kalman. Κάθε φίλτρο Kalman είναι συνδεδεμένο με ένα σωματίδιο το οποίο περιγράφει την θέση του ρομπότ. Συγκεκριμένα, η συνολική εκτίμηση πιθανότητας των μονοπατιών και των θέσεων των οροσήμεων, αναπαρίσταται ως εξής:

$$S_t = \{s^{t,[m]}, \mu_1^{[m]}, \Sigma_1^{[m]}, \dots, \mu_k^{[m]}, \Sigma_k^{[m]}\}_m \quad (15)$$

όπου  $\mu_i^{[m]}, \Sigma_i^{[m]}$  είναι η μέση τιμή και η απόκλιση της Gaussian που περιγράφει το  $i$ -οστό ορόσημο  $\theta_i$ , που αντιστοιχεί στο  $m$ -οστό σωματίδιο. Ο υπολογισμός της  $p(\theta_i | s^t, z^t, u^t, n^t)$  είναι απλός και εξαρτάται από το αν το ορόσημο παρατηρήθηκε ή όχι. Εάν ναι, τότε:

$$p(\theta_i | s^t, z^t, u^t, n^t) \propto p(z_t | \theta_{n_t}, s_t, n_t) p(\theta_{n_t} | s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \quad (16)$$

Εάν όχι, τότε:

$$p(\theta_{i \neq n_t} | s^t, z^t, u^t, n^t) = p(\theta_i | s^{t-1}, z^{t-1}, u^{t-1}, n^{t-1}) \quad (17)$$

Στη μέθοδο αυτή, η (16) υπολογίζεται με την βοήθεια ενός EKF. Αυτό σημαίνει ότι χρησιμοποιείται μια γραμμικοποιημένη μορφή του μοντέλου παρατήρησης  $p(z_t | s_t, \theta_{n_t}, n_t)$ .

Η μεγάλη διαφορά μεταξύ του πώς ο FastSLAM χρησιμοποιεί το φίλτρο Kalman και της παραδοσιακής μορφής του αλγορίθμου SLAM είναι ότι ο πρώτος χρειάζεται να υπολογίσει μόνο μια Gaussian διάσταση δύο. Αντίθετα ο αλγόριθμος SLAM βασισμένος στο EKF πρέπει να υπολογίζει  $2K + 3$  παραμέτρους.

## 2.5 Οπτικό SLAM

Τα τελευταία χρόνια, έχουν γίνει πολλές προσπάθειες για την επίλυση του SLAM με χρήση οπτικής πληροφορίας από κάμερες. Ένας από τους βασικούς λόγους που συμβαίνει αυτό, είναι το γεγονός ότι η κάμερα μας προσφέρει πολύ πλούσια πληροφορία, την οποία όπως θα δούμε και στη συνέχεια μπορούμε να την επεξεργαστούμε με διάφορους τρόπους. Επίσης, η κάμερα ως αισθητήριο όργανο, είναι πολύ λιγότερο επιρρεπής σε σφάλματα τόσο συστηματικά όσο και θορύβου, και συνεπώς είναι πολύ πιο αξιόπιστη στις μετρήσεις της.

Το κύριο στοιχείο που διαφοροποιεί τις διάφορες προσεγγίσεις σε ένα αλγόριθμο οπτικού SLAM, είναι το πώς κανείς συγκρίνει εικόνες. Σε αυτό το χώρο υπάρχουν πολλές ιδέες.

Στο [26], γίνεται μια προσπάθεια για ένα απλό και γρήγορο αλγόριθμο ο οποίος επιτυγχάνει πολύ καλά αποτελέσματα. Η σύγκριση των εικόνων γίνεται με βάση τα ιστογράμματα χρησιμοποιώντας ένα σύστημα ψηφοφορίας. Το κύριο τμήμα του αλγορίθμου χρησιμοποιεί τη μέθοδο RatSLAM ( η οποία θα χρειαζόταν αρκετό χρόνο για να περιγραφεί και δεν θα

μας απασχολήσει). Εντυπωσιακά επίσης είναι τα αποτελέσματα από την χρήση της οπτικής οδομετρίας η οποία χρησιμοποιείται για τον υπολογισμό του προσανατολισμού και για την δημιουργία ενός ψευδο-μετρικού χάρτη.

Μια άλλη πολύ ενδιαφέρουσα προσέγγιση παρουσιάζεται στο [27]. Εκεί χρησιμοποιούνται χαρακτηριστικά τύπου SURF τα οποία έχουν κάποιες πολύ σημαντικές ιδιότητες, όπως ανεξαρτησία από στροφή και κλίμακα. Η καρδιά του αλγορίθμου είναι ένα EKF το οποίο τελικά υπολογίζει την θέση του ρομπότ και των οροσήμων. Τα SURF χρησιμοποιούνται για την σύγκριση εικόνων, με βάση την Ευκλείδεια απόσταση και υλοποιούν πρακτικά το μοντέλο παρατήρησης του φίλτρου.

Το [11] προτείνει την δημιουργία τοπολογικού χάρτη καθώς θα είναι λιγότερο ευαίσθητος σε λάθη οδομετρίας και θέσης. Προτείνεται μια μέθοδος προκειμένου να κατασκευαστεί αυξητικά ο χάρτης χρησιμοποιώντας μια πανοραμική κάμερα, από τα δεδομένα της οποίας εξάγονται χαρακτηριστικά. Για τον σκοπό αυτό χρησιμοποιείται ο αλγόριθμος KLT (Kanade-Lucas-Tomashi) [28], ο οποίος επιτρέπει την εύρεση του καλύτερου ταιριάσματος μεταξύ δύο εικόνων.

Ακόμα, το [29] προτείνει μια μεθοδολογία για την δημιουργία ενός υβριδικού χάρτη κλειστού χώρου. Ο υβριδικός χάρτης έχει δύο επίπεδα αναπαράστασης: 1) ο απόλυτος χάρτης, ο οποίος αποτελείται από απολύτως διακριτά οπτικά χαρακτηριστικά (για παράδειγμα χαρακτηριστικά SIFT) και σχεδιάζεται σε σχέση με καθολικές συντεταγμένες και 2) ο σχετικός χάρτης που δημιουργείται σε σχέση με τις ρομποκεντρικές (robotcentric) συντεταγμένες.

Τέλος, μια πολύ ενδιαφέρουσα εργασία παρουσιάζεται στο [30]. Η βασική ιδέα, είναι ο εντοπισμός αντικειμένων από τις εικόνες, χρησιμοποιώντας ένα συνδυασμό ιστογραμμάτων HSI και άλλων χαρακτηριστικών όπως SIFT χαρακτηριστικά. Χρησιμοποιείται ένας χάρτης πλέγματος ο οποίος κατασκευάζεται με τη βοήθεια υπέρυθρων. Το πλεονέκτημα της μεθόδου είναι η ταχύτητα σε σχέση με άλλες μεθόδους που χρησιμοποιούν γωνίες ή ακμές για την αναγνώριση εικόνων, καθώς τα αντικείμενα που εντοπίζονται από την μέθοδο αυτή, είναι πολύ λιγότερα και έτσι είναι πολύ ευκολότερο να τα ταιριάξουμε.

## 3

# Θεωρητικό Υπόβαθρο

Το κεφάλαιο αυτό καλύπτει σημαντικά κομμάτια του αλγορίθμου SLAM: αναπαράσταση περιβάλλοντος, ταίριασμα εικόνων και πιθανοτικά φίλτρα.

Στην Ενότητα 3.1 αναφερόμαστε σε μεθόδους ταιριάσματος εικόνων. Αναδुकνείουμε την πλούσια βιβλιογραφία και την πληθώρα μεθοδολογιών που υπάρχει στον τομέα αυτό. Το ταίριασμα εικόνων, όπως θα φανεί και στην συνέχεια, είναι πολύ σημαντικό και θα μας απασχολήσει σε μεγάλο βαθμό.

Στην Ενότητα 3.2, γίνεται μια αναφορά στην αναπαράσταση των χαρτών και δη των τοπολογικών αναπαραστάσεων, που άλλωστε θα χρησιμοποιήσουμε στον αλγόριθμό μας.

Τέλος, στην Ενότητα 3.3 γίνεται μια αναφορά στο κύριο μαθηματικό εργαλείο που χρησιμοποιείται από την παρούσα εργασία: το Φίλτρο Bayes. Πολύ σημαντική όπως θα φανεί και στην συνέχεια είναι η Υπόθεση Πλήρους Κατάστασης καθώς καθιστά υπολογιστικά εφικτή, την επίλυση ενός τέτοιου προβλήματος.

### 3.1 Ταίριασμα Εικόνων (Image Correspondence)

Ένα πολύ βασικό χαρακτηριστικό ενός αλγορίθμου οπτικού SLAM είναι η ικανότητα συσχέτισης δύο εικόνων, το λεγόμενο πρόβλημα του ταιριάσματος εικόνων (*image correspondence*). Πιο συγκεκριμένα, ας φανταστούμε ότι περπατάμε σε ένα χώρο παίρνοντας εικόνες. Σε κάποια στιγμή, επιστρέφουμε σε ένα σημείο από το οποίο έχουμε ξαναπεράσει. Είναι λογικό να περιμένουμε ότι η εικόνα που βλέπουμε τόσο στο 1<sup>ο</sup> όσο και στο 2<sup>ο</sup> πέρασμα από το ίδιο σημείο, θα είναι σε γενικές γραμμές οι ίδιες. Αυτό, μπορεί να ακούγεται απλό στην θεωρία, αλλά στην πράξη υπάρχουν πολλοί λόγοι για τους οποίους μπορεί οι δύο εικόνες να μην είναι και τόσο όμοιες, ή τουλάχιστον να μην μπορούμε εύκολα να ανακαλύψουμε ότι όντως αντιστοιχούν στο ίδιο σημείο του χώρου. Για παράδειγμα, αλλαγές στον φωτισμό ή στην κίνηση μας καθώς παίρνουμε την εικόνα, ή ακόμα και σε δυναμική αλλαγή του περιβάλλοντος, αποτελούν μερικούς παράγοντες που δυσκολεύουν το πρόβλημα αυτό.

Το πρόβλημα του του *image correspondence* είναι υποπρόβλημα του *image registration*. Το τελευταίο αναζητά το πώς μπορεί να γίνει η βέλτιστη συγχώνευση δύο εικόνων σε μια (*overlay*). Λέγοντας ότι το πρόβλημα του *correspondence* είναι υποπρόβλημα του *registration*, εννοούμε ότι το εάν λύσει κανείς το 2<sup>ο</sup>, έχει λύσει και το 1<sup>ο</sup> [31].

Το γενικότερο πρόβλημα του *image registration*, αποτελείται από 4 βήματα:

1. Εντοπισμός χαρακτηριστικών των εικόνων (feature detection). Όπως θα φανεί στη συνέχεια, τα χαρακτηριστικά μπορεί να αναφέρονται είτε σε περιοχές, είτε σε σημεία της εικόνας.

2. Ταίριασμα των χαρακτηριστικών(feature matching). Εδώ, προσπαθούμε να βρούμε το κατά πόσο χαρακτηριστικά των δύο εικόνων, σχετίζονται μεταξύ τους. Εδώ πρακτικά τελειώνει το πρόβλημα του *image correspondence*.
3. Υπολογισμός του μοντέλου μετασχηματισμού. Με βάση το βήμα 2, βρίσκονται οι παράμετροι των συναρτήσεων αντιστοίχισης προκειμένου η μια εικόνα να ταιριάζει (για παράδειγμα με έναν αφινικό μετασχηματισμό) με την άλλη.
4. Τέλος γίνεται ο μετασχηματισμός της εικόνας, προκειμένου να ταυτιστεί με την άλλη.

Στην ανάλυση που ακολουθεί, θα μας απασχολήσουν μόνο τα δύο πρώτα βήματα.

### 3.1.1 Βιβλιογραφική Επισκόπηση

Βλέπουμε ότι το να μπορούμε να πούμε το κατά πόσο δύο εικόνες ταιριάζουν, δηλαδή αντιστοιχούν στο ίδιο σημείο του χώρου, είναι ένα ιδιαίτερα σημαντικό πρόβλημα. Εδώ και καιρό, γίνονται προσπάθειες στην επιστημονική κοινότητα προκειμένου να μπορεί κανείς να αντιστοιχίζει εικόνες αυτόματα, δηλαδή χωρίς καμία επέμβαση ανθρώπινου παράγοντα. Κάτι τέτοιο είναι πάρα πολύ σημαντικό σε τομείς όπως η υπολογιστική όραση, η ιατρική, ή και η ασφάλεια (για παράδειγμα με την ταυτοποίηση δακτυλικών αποτυπωμάτων). Στο πρόβλημα του *Οπτικού SLAM*, αυτή η ανάγκη είναι εμφανής καθώς πρέπει να μπορούμε βάσει των εικόνων να εξαγάγουμε συμπεράσματα για το πού μπορεί να βρίσκεται το ρομπότ.

Γενικά υπάρχουν δύο κύριες προσεγγίσεις για το πρόβλημα του *image correspondence*, κυρίως ως προς το τρόπο εντοπισμού χαρακτηριστικών, τις οποίες θα αναλύσουμε στη συνέχεια:

- *Correlation/Area based.*
- *Feature based.*

### 3.1.2 Μέθοδοι Βασισμένες σε Συσχέτιση-Περιοχές (Correlation-Area Based Methods)

Η 1<sup>η</sup> κατηγορία μεθόδων είναι ίσως η πιο προφανής και διαισθητικά σωστή: για να βρει κανείς εάν δύο εικόνες ταιριάζουν, αρκεί να εντοπίσει περιοχές των εικόνων που είναι περίπου ίδιες. Στη συναφή βιβλιογραφία αναφέρονται πάρα πολλές μέθοδοι που προσεγγίζουν το πρόβλημα του *image correspondence* με τον τρόπο αυτό. Οι μέθοδοι που βασίζονται σε περιοχές των εικόνων (area based), ουσιαστικά αποφεύγουν το 1<sup>ο</sup> βήμα που αναφέραμε στην εισαγωγή της 3<sup>ης</sup> ενότητας, περνώντας κατευθείαν στο βήμα του feature matching [31].

Οι μέθοδοι αυτές, ουσιαστικά συνδυάζουν τα βήματα 1 και 2 που αναφέραμε πιο πάνω.

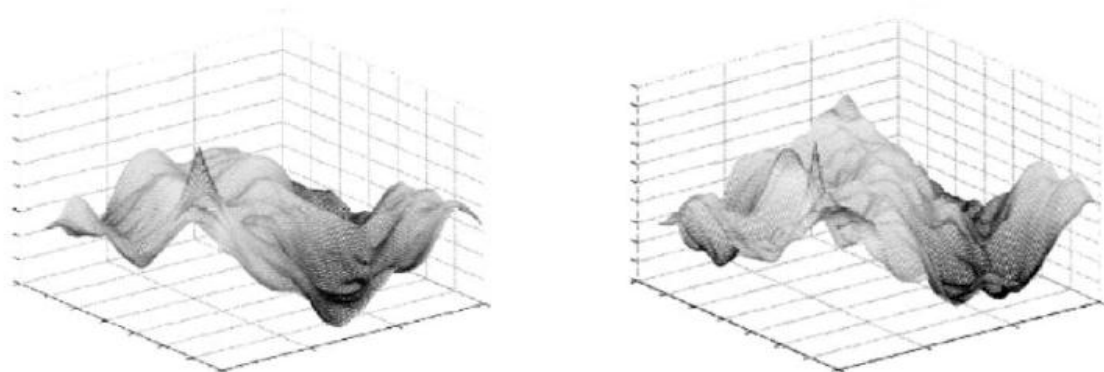
Η βασική λογική των μεθόδων αυτών είναι η χρήση παραθύρων των εικόνων που στη συνέχεια συγκρίνονται μεταξύ τους. Με τη σειρά τους, οι μέθοδοι αυτές χωρίζονται σε διάφορες κατηγορίες:

### 3.1.3 Cross-Correlation Μέθοδοι

Η πιο χαρακτηριστική μορφή του Cross-Correlation περιγράφεται στα [31], [32]. Για παράδειγμα στο [32], γίνεται μια αναφορά σε μια πολύ απλή μορφή ταιριάσματος εικόνων: οι δύο εικόνες  $I_1, I_2$  χωρίζονται σε τμήματα  $n \times n$  pixel  $I_1^i, I_2^i$  και στη συνέχεια υπολογίζεται ένας κανονικοποιημένος δείκτης ταιριάσματος ως εξής:

$$CC = \frac{\sum_{i=1}^n (I_1^i - \bar{I}_1)(I_2^i - \bar{I}_2)}{\sqrt{\sum_{i=1}^n (I_1^i - \bar{I}_1)^2} \sqrt{\sum_{i=1}^n (I_2^i - \bar{I}_2)^2}} \quad (18)$$

, όπου  $\bar{I}$ , η μέση τιμή των pixels.



Σχημα 5.

Πάνω: οι δύο εικόνες προς ταίριασμα.

Κάτω: το αποτέλεσμα του normalized cross correlation, αριστερά σε red-red κανάλια, ενώ δεξιά σε red-blue.

Πηγή: [31]

Μια τέτοια προσέγγιση, αν και σταθερή σε γραμμικές αλλαγές της φωτεινότητας και πιθανώς σε μικρές περιστροφές και αλλαγές κλίμακας, δεν είναι καθόλου εύρωστη (robust) σε μη γραμμικά φαινόμενα καθώς και στον θόρυβο. Επίσης, ανεπιθύμητο είναι το γεγονός ότι μεμονωμένα pixels μπορεί να επηρεάσουν πολύ τον βαθμό ομοιότητας. Γενικά, κανείς επιδιώκει να ελαχιστοποιήσει την επίδραση πολύ διαφορετικών pixel (outliers) δίνοντας τους μικρό βάρος, μεγιστοποιώντας όμως παράλληλα την βαρύτητα των σημαντικών (inliers) δειγμάτων [32]. Ένας τρόπος αντιμετώπισης αυτού του φαινομένου είναι η χρήση κατωφλίων. Ένα κατώφλι, συνήθως δοσμένο από τον χρήστη, είναι αυτό που καθορίζει εάν μια τιμή θεωρείται έγκυρη (inlier) και συνεπώς λαμβάνεται υπ' όψιν, ή άκυρη (outlier) και αγνοείται. Μια άλλη αντιμετώπιση, μετασχηματίζει την εικόνα από pixel φωτεινότητας, σε pixel τάξης και ονομάζεται *μετασχηματισμός τάξης (rank transform)*: κάθε pixel του παραθύρου παίρνει μια τιμή, που εξαρτάται από το πόσα γειτονικά του pixel έχουν μικρότερη τιμή από αυτό. Αυτό απεμπλέκει την οποιαδήποτε εν συνεχεία σύγκριση των εικόνων από απόλυτες τιμές φωτεινότητας και κατ' επέκταση από outliers.

Άλλες μέθοδοι προσπαθούν να αντιστοιχίσουν εικόνες που έχουν υποστεί αφινικούς μετασχηματισμούς περιγράφονται στα [33], [34] και βασίζονται σε πιο σύνθετες και υπολογιστικά απαιτητικές τεχνικές. Το [33] υποθέτει διάφορους γεωμετρικούς μετασχηματισμούς για το κάθε παράθυρο της εξεταζόμενης εικόνας. Το [34] χρησιμοποιεί ένα αλγόριθμο τύπου «διαίρει και βασίλευε» σε συνδυασμό με το cross-correlation για να αντιστοιχίζει εικόνες με αλλαγές στην προοπτική αλλά και ατέλειες του φακού της κάμερας.

Στο [35], γίνεται χρήση όχι των αρχικών εικόνων (με pixel φωτεινότητας), αλλά δυαδικών τους, που βασίζονται σε ανίχνευση ακμών και η σύγκριση γινόταν με χρήση του μέτρου Hausdorff. Το αποτέλεσμα ήταν πολύ καλύτερο από την μέθοδο που περιγράφεται στο [32], ειδικά με εικόνες που είχαν υποστεί αφινικούς μετασχηματισμούς.

### 3.1.4 Fourier-Based Μέθοδοι

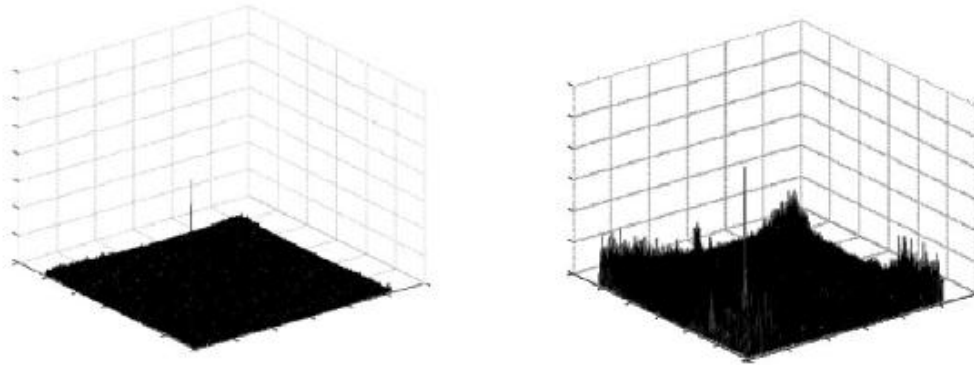
Η υπολογιστική πολυπλοκότητα της μεθόδου Cross-Correlation, σε συνδυασμό με την ύπαρξη θορύβου που εξαρτάται από την συχνότητα, συχνά επιβάλλει την χρήση άλλων μεθόδων βασισμένων σε μετασχηματισμούς Fourier.

Εάν  $f$  είναι η αρχική εικόνα και  $g$  η καινούργια, τότε η βασική ιδέα είναι ο υπολογισμός του:

$$R = \frac{F(f)F^*(g)}{|F(f)F^*(g)|} \quad (19)$$

και μετά ο εντοπισμός του σημείου της εικόνας όπου

$$(i, j) = \arg \max_{(i, j)} R^{-1} \quad (20).$$



**Σχημα 6. Παράδειγμα Fourier based correlation. Στα αριστερά, ταίριασμα red-red καναλιών, ενώ στα δεξιά red-blue. Παρατηρούμε το τύπου dirac μέγιστο και στα δύο σχήματα.**

Πηγή: [31]

Η παραπάνω μορφή είναι κατάλληλη μόνο για εικόνες που έχουν υποστεί κάποια μετατόπιση μόνο. Παρ'όλα αυτά, στο [36] γίνεται μια επέκταση που επιτρέπει και περιστροφή μεταξύ των εικόνων. Περαιτέρω βελτίωση έγινε στα [37], [38], οπότε και προστέθηκε η δυνατότητα για ταίριασμα εικόνων με διαφορετική κλίμακα.

### 3.1.5 Μέθοδοι Αμοιβαίας Πληροφορίας

Η πιο πρόσφατη μέθοδος προτάθηκε μόλις το 1995 από τον Viola και έχει εκτεταμένη εφαρμογή σε multi modal προβλήματα, δηλαδή όταν οι εικόνες που συγκρίνουμε έχουν ληφθεί από διαφορετικά όργανα-αισθητήρες. Οι εφαρμογές αυτές, σχετίζονται για παράδειγμα με το να παίρνουμε συνδυασμό μαυρόασπρων και εικόνων με χρώματα, ή όταν σε ιατρικές εικόνες καλούμαστε να συνδυάσουμε εικόνες από MRI (Magnetic Resonance Image) ή υπέρηχους, με πληροφορία από αισθητήρες που παρακολουθούν την λειτουργία του σώματος, όπως PET (Photon Emission Tomography) ή SPECT (Single Photon Emission Computed Tomography) [31].

Το ταίριασμα multi modal εικόνων είναι το πιο δύσκολο πρόβλημα και είναι υπολογιστικά πολυσύνθετο. Η αρχική ιδέα προέρχεται από την θεωρία της πληροφορίας και προσπαθεί να βρει το κατά πόσο δύο σύνολα δεδομένων είναι μεταξύ τους στατιστικά εξαρτημένα. Στο [39], μια μέθοδος εμπνευσμένη από τις δυνάμεις που συναντάμε στη φύση, προσπαθεί να βρει τις παραμέτρους μετατόπισης και περιστροφής προκειμένου να ταιριάζει δύο εικόνες.

Στο [40] γίνεται μια σύγκριση μεταξύ διαφόρων μεθόδων ταιριάσματος εικόνων, μεταξύ αυτών: κανονικοποιημένη συσχέτιση (normalized cross-correlation), αμοιβαία πληροφορία (mutual information), εντροπία εικόνας διαφοράς (entropy of difference image), συσχέτιση κλίσης (gradient correlation) και άλλες. Το συμπέρασμα που βγήκε με χρήση εικόνων κυρίως από την ιατρική, είναι ότι η μέθοδος αμοιβαίας πληροφορίας είχε τα μεγαλύτερα ποσοστά αποτυχίας στο ταίριασμα. Αυτό οφείλεται πρώτον, στο ότι αν και η αμοιβαία πληροφορία είναι χρήσιμη σε 3-D εικόνες εντούτοις αποτυγχάνει στο ταίριασμα εικόνων 2-D σε 3-D ή το



αντίστροφο και δεύτερον στο ότι η μέθοδος δεν υποθέτει γραμμική σχέση φωτεινότητας μεταξύ δύο εικόνων παρ'όλο που στην συγκεκριμένη εφαρμογή αυτή η γραμμική σχέση υπήρχε εκ κατασκευής.

### 3.1.5.1 Μέθοδοι Βελτιστοποίησης

Σε αυτές τις μεθόδους, προσπαθούμε να βρούμε το ελάχιστο ενός μέτρου ανομοιότητας, ή το μέγιστο ενός μέτρου ομοιότητας μεταξύ των δύο εικόνων. Σε αυτή την περίπτωση, το πρόβλημα είναι πολυδιάστατο και έχει τόση διαστασιμότητα, όσοι και οι βαθμοί ελευθερίας του γεωμετρικού μετασχηματισμού που συνδέει τις δύο εικόνες. Ο μόνος τρόπος να εξασφαλίσουμε την εύρεση του ολικού ακρότατου, είναι μια εξαντλητική έρευνα στον χώρο λύσεων. Το τελευταίο χρησιμοποιείται κυρίως σε περιπτώσεις που ψάχνουμε μόνο την μετατόπιση μεταξύ των δύο εικόνων.

Εάν αναζητάμε πιο σύνθετους μετασχηματισμούς τότε είναι αναγκαία η χρήση εξελιγμένων αλγορίθμων βελτιστοποίησης. Το [41] χρησιμοποιεί την μέθοδο της *ταχύτερης κατάβασης* (*Gradient Descent*) για την μεγιστοποίηση της αμοιβαίας πληροφορίας μεταξύ των δύο εικόνων. Το [42] υιοθετεί την χρήση της μεθόδου Levenberg-Marquardt, προκειμένου να ελαχιστοποιήσει την απόκλιση της φωτεινότητας αντίστοιχων pixel. Τέλος, το [43] χρησιμοποιεί τη μέθοδο της *προσωμειωμένης απόπτωσης* (*simulated annealing*), προκειμένου να ελαχιστοποιηθεί ένα μέτρο ανομοιότητας ανάμεσα σε ζευγάρια σημείων των εικόνων.

### 3.1.6 Μέθοδοι Βασισμένοι σε Χαρακτηριστικά (Feature-Based Methods)

Η δεύτερη κατηγορία μεθόδων, στηρίζεται στην εξαγωγή από κάθε εικόνα μόνο ορισμένων κρίσιμων σημείων ή και υποπεριοχών της. Έτσι στο 2<sup>ο</sup> βήμα του image registration δεν απαιτείται η σύγκριση ολόκληρων των εικόνων, παρά μόνο των κρίσιμων σημείων τους. Υπάρχει πληθώρα σημείων που μπορούν να θεωρηθούν κρίσιμα-χαρακτηριστικά μιας εικόνας [31].

- Περιοχές της εικόνας (*Χαρακτηριστικές Περιοχές*). Τέτοιες μπορεί να είναι για παράδειγμα αντικείμενα, δάση, λίμνες ή άλλα που αποτελούν ιδιαίτερα χαρακτηριστικά μιας εικόνας.
- Ακμές της εικόνας (*Χαρακτηριστικές Γραμμές*). Περιοχές με έντονη αλλαγή της φωτεινότητας προς μια κατεύθυνση είναι συνήθως ιδιαίτερα σημαντικές, και μπορεί να αποτελούν όρια μεταξύ περιοχών, δρόμους, ακτογραμμές κτλ.
- Σημεία της εικόνας (*Χαρακτηριστικά Σημεία*). Τέτοια χαρακτηριστικά μπορεί να είναι για παράδειγμα, γωνίες περιοχών, σημεία διασταύρωσης γραμμών ή σημεία πάνω σε καμπύλες έντονης καμπυλότητας.

Αυτό που χρειαζόμαστε σε τέτοια χαρακτηριστικά, είναι κυρίως το να είναι διακριτά μεταξύ τους, να καλύπτουν όσο το δυνατόν μεγαλύτερο μέρος της εικόνας και να εντοπίζονται αποτελεσματικά και αποδοτικά και στις δύο εικόνες. Επιπρόσθετα, πρέπει να είναι όσο το δυνατόν πιο σταθερά και εύρωστα, δηλαδή να διατηρούν αναλλοίωτα τα χαρακτηριστικά τους, ανεξάρτητα του χρόνου και πιθανών μετασχηματισμών της εικόνας.

Η ευρωστία των χαρακτηριστικών είναι αναγκαία για να μπορούμε να κάνουμε έγκυρη σύγκριση μεταξύ των χαρακτηριστικών δύο εικόνων. Κάτι τέτοιο πρέπει να διασφαλίζεται από τον αλγόριθμο που εντοπίζει τα χαρακτηριστικά αυτά και ο οποίος θα πρέπει να βρίσκει ένα επαρκή αριθμό σημείων ανεξάρτητα από αλλαγές στην φωτεινότητα, την γεωμετρία, τον θόρυβο κτλ. Στη συνέχεια θα κάνουμε μια σύντομη επισκόπηση των μεθόδων που αφορούν τον εντοπισμό όλων των τύπων χαρακτηριστικών που αναφέραμε λίγο πιο πάνω.

### 3.1.6.1 Χαρακτηριστικές Περιοχές

Συνήθως αποτελούν κλειστές περιοχές (*Blobs*) της εικόνας, με μεγάλη αντίθεση και αρκετά μεγάλο μέγεθος. Οι περιοχές αυτές, όντας κλειστές, μπορούν να αναπαρασταθούν από το κέντρο βάρους τους, το οποίο είναι εν γένει αμετάβλητο ως προς περιστροφή, αλλαγή κλίμακας και γενικά ανεκτικό στην επίδραση του θορύβου [31]. Κύρια μέθοδος, εύρεσης τέτοιων περιοχών, είναι η *κατάτμηση (segmentation)* των εικόνων.

Στο [44], γίνεται μια εκτενής αναφορά και σύγκριση των μεθόδων κατάτμησης, που όπως γίνεται σαφές επηρεάζουν σε μεγάλο βαθμό την δυνατότητα ταιριάσματος των εικόνων. Ένας διαδομένος τρόπος, είναι η χρήση κατωφλίων. Το κατώφλι μπορεί να εξάγεται με βάση όλη την εικόνα (χρησιμοποιώντας για παράδειγμα ιστογράμματα), ή σε τοπικό επίπεδο για κάθε παράθυρο της εικόνας. Επίσης υπάρχει η *διεπίπεδη (two level thresholding)* και η *πολυεπίπεδη κατωφλιοποίηση (multilevel thresholding)*: η πρώτη, προσπαθεί να χωρίσει την εικόνα σε δύο τμήματα, το αντικείμενο και το φόντο. Η δεύτερη προσπαθεί ουσιαστικά να κάνει το ίδιο πράγμα, αλλά χρησιμοποιεί πολλαπλά κατώφλια για να βρει όλα τα αντικείμενα που πιθανώς να έχουν διαφορετικά χαρακτηριστικά. Στην τελευταία περίπτωση, κανείς προσπαθεί να εφαρμόσει στην εικόνα πολλαπλές κατωφλιοποιήσεις, ελπίζοντας να εξάγει διαφορετικές περιοχές της εικόνας με διαφορετική φωτεινότητα (ενδεχομένως πολλαπλά αντικείμενα). Η εύρεση των κατάλληλων κατωφλίων είναι στην γενική περίπτωση ένα εξαιρετικά σύνθετο πρόβλημα και το γεγονός ότι στηρίζονται μόνο σε ιστογράμματα τις καθιστά μη πρακτικές, κυρίως διότι δεν λαμβάνουν υπ' όψιν χωρική πληροφορία. Άλλες μέθοδοι λαμβάνουν υπ' όψιν αυτή την πληροφορία, αλλά τελικά και πάλι να κατωφλιοποιούν με βάση το ιστόγραμμα. Γενικά, όλες οι παραπάνω μέθοδοι είναι πολύ ευαίσθητες σε θόρυβο και σε κακό φωτισμό.

Μια άλλη κατηγορία μεθόδων, είναι οι επαναληπτικές μέθοδοι. Μια από αυτές, η *χαλάρωση*, κάνει παράλληλα την κατάταξη σε μια κατηγορία για κάθε pixel, στηριζόμενη είτε σε πιθανοτικά είτε σε ασαφή μοντέλα. Μια τέτοια μέθοδος περιγράφεται στο [45], όπου επαναληπτικά γινόταν μια βελτίωση της κατάτμησης με στόχο το βέλτιστο ταίριασμα των εικόνων.

Τέλος, έχουν αναπτυχθεί και μέθοδοι με χρήση νευρωνικών δικτύων. Η παράλληλη επεξεργασία σε συνδυασμό με την πολύπλοκη αρχιτεκτονική, κάνουν ένα τέτοιο σύστημα

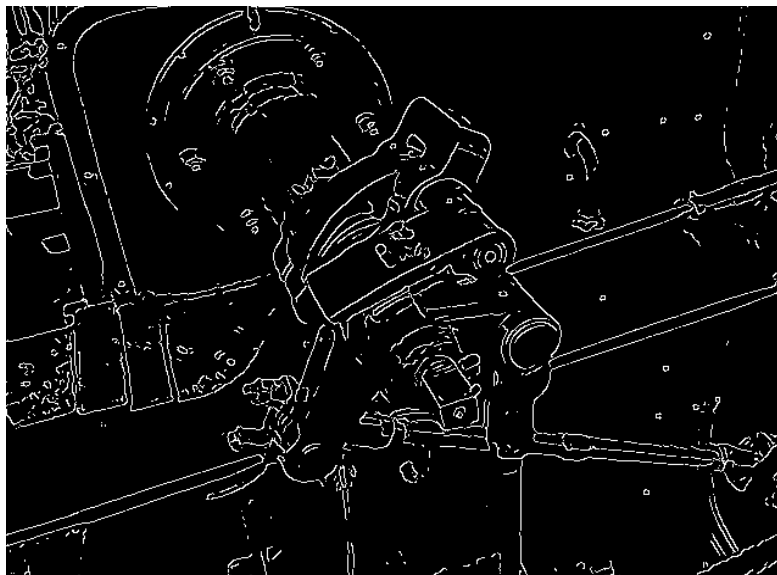
γρήγορο και ανεκτικό σε θόρυβο. Μια βασική ιδέα είναι η χρήση ενός πολυεπίπεδου δικτύου, με τόσες εισόδους, όσα τα χαρακτηριστικά του κάθε pixel, και εξόδους όσες οι κλάσεις. Η εκπαίδευση γίνεται με χρήση backpropagation μέχρις ότου το σύστημα μπορεί να καταφλιώσει σωστά την εικόνα. Φυσικά, κάτι τέτοιο προϋποθέτει την εκτενή εκπαίδευση με χρήση μιας μεγάλης βάσης δεδομένων από εικόνες. Επίσης, έχουν χρησιμοποιηθεί και δίκτυα Hopfield, των οποίων η συνάρτηση ενέργειας είναι φτιαγμένη έτσι ώστε να ελαχιστοποιείται και να είναι ευσταθής όταν εξάγει συμπαγείς περιοχές από μια εικόνα.

Στο [46], αναλύονται δύο μέθοδοι με τις οποίες μπορούν να εξαχθούν περιοχές σταθερές ως προς αφινικούς μετασχηματισμούς. Η 1<sup>η</sup> μέθοδος, ξεκινά από γωνίες που εντοπίζει με τον Harris Corner Detector, και από εκεί βρίσκει ακμές ενώ η 2<sup>η</sup> βασίζεται καθαρά στην φωτεινότητα των pixels.

### 3.1.6.2 Χαρακτηριστικές Γραμμές

Τέτοια χαρακτηριστικά μπορεί να αποτελούν οποιοσδήποτε γραμμές της εικόνας, συνήθως το περίβλημα των αντικειμένων. Συνήθως αναπαρίστανται είτε με τα δύο ακριανά σημεία τους, είτε με κάποια ενδιάμεσα.

Υπάρχουν διάφοροι αλγόριθμοι για τον εντοπισμό τέτοιων γραμμών-ακμών. Ο Canny detector [47] που φαίνεται στο Σχημα 7, ενώ και αυτός που βασίζεται στη Laplacian of Gaussian (LoG) [48] είναι από τους πιο διαδεδομένους.



Σχημα 7.Αποτέλεσμα του Canny edge detector. Πηγή:  
[http://en.wikipedia.org/wiki/Canny\\_edge\\_detector](http://en.wikipedia.org/wiki/Canny_edge_detector)

### 3.1.6.3 Χαρακτηριστικά σημεία

Τα σημεία αποτελούν το τελευταίο είδος χαρακτηριστικού μιας εικόνας που θα μελετήσουμε. Υπάρχουν πολλές μέθοδοι για την εξαγωγή τέτοιων σημείων. Στο [49] χαρακτηριστικά σημεία εντοπίζονται στις διασταυρώσεις ακμών της εικόνας. Τα [50] και [51] χρησιμοποιούν διασταυρώσεις δρόμων ως τέτοια σημεία.

Γενικά, οι περισσότεροι αλγόριθμοι θεωρούν ως σημείο, την διασταύρωση δύο ή περισσότερων γραμμών, ή ως το κέντρο βάρους μιας κλειστής περιοχής. Οι γωνίες αποτελούν μια ειδική περίπτωση τέτοιων χαρακτηριστικών και γενικά δεν είναι εύκολο να ορίσει κανείς μαθηματικά μια γωνία αν και διαισθητικά, γωνία είναι ένα σημείο με μεγάλη καμπυλότητα στο σύνορο μιας περιοχής. Η γωνία, είναι ένα από τα πιο σημαντικά χαρακτηριστικά σημεία μιας εικόνας, όμως η δυσκολία ενός αυστηρού ορισμού έχει οδηγήσει σε διάφορους αλγορίθμους εντοπισμού γωνιών. Το [52] χρησιμοποιεί την μέθοδο SUSAN (Small Univalued Segment Assimilating Nucleus), που εντοπίζει ταυτόχρονα ακμές, γωνίες και διασταυρώσεις χωρίς περιορισμούς στην διεύθυνση και επιχειρεί μια σύγκριση με άλλες αντίστοιχες μεθόδους (Sobel, Canny, Prewitt).

Στο [53] γίνεται μια ανάλυση των διαφόρων μεθόδων εντοπισμού γωνιών. Η 1<sup>η</sup> κατηγορία μεθόδων, βασίζονται σε ταίριασμα προτύπων: η βασική ιδέα είναι ο υπολογισμός της ομοιότητας μεταξύ ενός πρότυπου παραθύρου, και όλων των παραθύρων ίδιου μεγέθους, της εικόνας. Οι γωνίες είναι τα μέγιστα αυτής της συνάρτησης ομοιότητας. Η 2<sup>η</sup> κατηγορία είναι οι γεωμετρικές μέθοδοι που με τη σειρά τους διακρίνονται σε, τοπολογικές μεθόδους (όπως ο Harris corner detector), σε συνδεδεμένες με ακμές (edge related) και σε μεθόδους αυτοσυσχέτισης.

#### 3.1.6.3.1 Αλγόριθμοι Εντοπισμού Χαρακτηριστικών Σημείων

Δεδομένης της ευρείας χρήσης των χαρακτηριστικών σημείων από πολλές εργασίες που ασχολούνται με την σύγκριση εικόνων, κρίνουμε σκόπιμο να αναφερθούμε σε κάποιες πολύ δημοφιλείς μεθόδους εντοπισμού τέτοιων σημείων.

Μια πρώτη κατηγορία αλγορίθμων εντοπισμού γωνιών, δουλεύουν σε τοπικό επίπεδο πάνω σε μια εικόνα  $I$ , με κάποιο πίνακα δομής  $C$ . Ο πίνακας αυτός, έχει γενικά την μορφή:

$$C(x, y) = \begin{bmatrix} \left(\frac{\partial I}{\partial x}\right)^2 & \frac{\partial I^2}{\partial x \partial y} \\ \frac{\partial I^2}{\partial x \partial y} & \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \quad (21)$$

Οι ιδιοτιμές του παραπάνω πίνακα έχουν ιδιαίτερη σημασία στον εντοπισμό γωνιών. Αν  $\lambda_1, \lambda_2$  οι ιδιοτιμές του για κάθε σημείο  $(x, y)$  της εικόνας, τότε:

- Αν  $\lambda_1 = \lambda_2 = 0$ , τότε η περιοχή είναι ομοιογενής και συνεπώς δεν υπάρχουν γωνίες.
- Αν  $\lambda_1 > \lambda_2 = 0$ , τότε βρισκόμαστε σε ακμή, με κατεύθυνση που εξαρτάται από την ιδιοτιμή  $\lambda_1$ .
- Αν  $\lambda_1 \geq \lambda_2 > 0$ , τότε λογικά βρισκόμαστε πάνω σε μια γωνία. Όσο μεγαλύτερα είναι τα  $\lambda_1, \lambda_2$ , τόσο πιο μεγάλη αντίθεση υπάρχει γύρω από το υποψήφιο σημείο.

❖ Μια πρώτη ιδέα είναι ο *Harris corner detector*.

Ο αλγόριθμος αυτός δουλεύει με μια παραλλαγή του πίνακα C, που αναφέραμε πιο πάνω. Συγκεκριμένα:

$$C_{harris} = \sigma_D^2 g(x, \sigma_I) * \begin{bmatrix} I_x^2(x, \sigma_D) & I_x(x, \sigma_D)I_y(x, \sigma_D) \\ I_x(x, \sigma_D)I_y(x, \sigma_D) & I_y^2(x, \sigma_D) \end{bmatrix}$$

$$I_x(x, \sigma_D) = \frac{\partial}{\partial x} g(\sigma_D) * I(x), \text{ (όμοια για τις άλλες μερικές παραγώγους)} \quad (22)$$

$$g(\sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

Στη συνέχεια υπολογίζει την *συνάρτηση γωνιότητας*:

$$r(x, y) = \det(C_{harris}(x, y)) - k \times \text{trace}(C_{harris})^2 \quad (23)$$

όπου  $k$  μια σταθερά επιλεγμένη από τον χρήστη.

Εν συνεχεία γίνεται μια κατωφλιοποίηση προκειμένου να αφαιρεθούν αδύναμες γωνίες. Γενικά, σημεία όπου η συνάρτηση γωνιότητας παίρνει μεγάλες τιμές, αποτελούν υποψήφιες γωνίες και άρα χαρακτηριστικά σημεία.

Το πρόβλημα του αλγορίθμου αυτού είναι ότι δεν είναι ανεκτικός σε αλλαγές κλίμακας. Για το λόγο αυτό, στο [54] γίνεται μια βελτίωση, όπου υπολογίζονται πολλοί πίνακες C σε πολλαπλές κλίμακες (δηλαδή, φιλτράρισμα από Γκαουσιανές με διαφορετικά  $\sigma$ ). Δηλαδή:

$\sigma_n = \xi^n \sigma_0$ , με  $\sigma_0$  μια αρχική κλίμακα και  $\xi$  μια σταθερά. Ο πίνακας  $C(x, y; \sigma_I, \sigma_D)$ , υπολογίζεται για  $\sigma_I = \sigma_n$ ,  $\sigma_D = s\sigma_1$ , όπου  $s$  μια σταθερά. Ο υπολογισμός αυτός, γίνεται για αρκετές κλίμακες  $\sigma_n$ . Έτσι, σε μια διδιάστατη εικόνα δημιουργείται μια  $3^n$  διάσταση, αυτή της κλίμακας, ο λεγόμενος *χώρος κλίμακας (scale space)*. Με τον τρόπο αυτό, ουσιαστικά αναζητάμε πλέον γωνίες, σε όλο αυτό το τριδιάστατο χώρο. Για κάθε

κλίμακα εντοπίζονται τα σημεία που η συνάρτηση γωνιότητας είναι μέγιστη. Έπειτα για όλα αυτά τα σημεία υπολογίζεται η ποσότητα

$$\Delta_n(x, \sigma_n) = \sigma_n^2 |L_{xx}(x, \sigma_n) + L_{yy}(x, \sigma_n)| \quad (24)$$

Τα σημεία όπου η  $\Delta$  παρουσιάζει ακρότατο σε σχέση με τις δύο γειτονικές κλίμακες, θεωρούνται γωνίες.

- ❖ Η 2<sup>η</sup> πολύ σημαντική μέθοδος εντοπισμού χαρακτηριστικών σημείων, είναι ο λεγόμενος *KLT (Kanade-Lucas-Tomasi) corner detector* [55], που είναι λίγο μεταγενέστερος του Harris corner detector. Η λογική του αλγορίθμου είναι η εξής:

Για κάθε σημείο της εικόνας, υπολογίζουμε τον ακόλουθο πίνακα:

$$C_{KLT}(x, y) = \begin{bmatrix} \sum \sum_R \left(\frac{\partial I}{\partial x}\right)^2 & \sum \sum_R \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} \\ \sum \sum_R \frac{\partial I}{\partial x} \frac{\partial I}{\partial y} & \sum \sum_R \left(\frac{\partial I}{\partial y}\right)^2 \end{bmatrix} \quad (25)$$

όπου R είναι μια γειτονιά του σημείου (x,y).

Στη συνέχεια, υπολογίζουμε την ελάχιστη ιδιοτιμή του πίνακα C, έστω  $\lambda_2$ . Εάν  $\lambda_2 > \lambda_{\min}$ , τότε αποθηκεύουμε το σημείο (x,y) σε μία λίστα υποψήφιων γωνιών. Ταξινομούμε την λίστα με φθίνουσα σειρά, βάσει των  $\lambda_2$ . Τέλος, επιλέγουμε σημεία από τη λίστα ξεκινώντας από αυτά με το μεγαλύτερο  $\lambda_2$ . Όσα άλλα σημεία εμπίπτουν μέσα στην περιοχή R επιλεγμένων σημείων απορρίπτονται. Όπως και ο Harris detector, έτσι και ο KLT χρειάζεται δύο παραμέτρους ορισμένες από τον χρήστη: το κατώφλι  $\lambda_{\min}$ , και το μέγεθος του παραθύρου R.

- ❖ Μια ακόμα πολύ ενδιαφέρουσα μέθοδος, εντοπίζει όχι μόνο γωνίες, αλλά γενικότερα σημεία της εικόνας που είναι ιδιαίτερα χαρακτηριστικά. Η μέθοδος αυτή, *SIFT (Scale Invariant Feature Transform) keypoint detector*, προτάθηκε το 1999 στο [56]. Το σημαντικότερο στοιχείο των εντοπιζόμενων χαρακτηριστικών, είναι η σταθερότητα που παρουσιάζουν σε αφινικούς μετασχηματισμούς, αλλαγές κλίμακας και εν μέρει σε αλλαγές φωτισμού και άλλων παραμορφώσεων. Η διαφορά της μεθόδου αυτής από τις δύο προηγούμενες μεθόδους έγκειται στην αναπαράσταση των χαρακτηριστικών σημείων: ενώ προηγουμένως, τα σημεία ήταν απλά ένα διάνυσμα (x,y) στον χώρο της εικόνας, πλέον κάθε σημείο περιγράφεται από ένα πολύ πιο σύνθετο διάνυσμα στο οποίο θα αναφερθούμε στη συνέχεια. Τα βήματα της μεθόδου είναι τα εξής:

- Συνέλιξη της εικόνας με Gaussian με τυπικές αποκλίσεις  $\{\sigma_1, \sigma_2, \dots\}$ , όπου γενικά  $\sigma_{j+1} = k\sigma_j$ . Αποτέλεσμα είναι οι ομαλοποιημένες εικόνες  $\{G(x,y,\sigma_1), G(x,y,\sigma_2), \dots\}$ .

- Οι εικόνες γειτονικής κλίμακας αφαιρούνται για να δώσουν 3 ή 4 DoG(Difference of Gaussian). Δηλαδή:

$$D(x, y, \sigma_j) = G(x, y, \sigma_{j+1}) - G(x, y, \sigma_j) \quad (26)$$

- Πηγαίνουμε ξανά στο 1<sup>ο</sup> βήμα, αυτή τη φορά όμως υποδειγματοληπτούμε την αρχική εικόνα, συνεχίζοντας στο 2<sup>ο</sup> βήμα. Παίρνουμε έτσι ένα σύνολο από DoG στον χώρο κλίμακας.
- Κάθε σημείο των DoG εξετάζεται: χαρακτηριστικό, είναι ένα σημείο όπου η DoG στο σημείο αυτό ακουμπά σε ακρότατο σε μια περιοχή 3x3 γύρω από το σημείο, σε σχέση με τις γειτονικές κλίμακες.

Ακολουθεί μια κατωφλιοποίηση και απόρριψη μερικών υποψήφιων σημείων. Σε κάθε σημείο αντιστοιχίζεται μια συγκεκριμένη κλίση(gradient) και διεύθυνση, που κανονικοποιούνται για να ενισχύσουν την σταθερότητά τους.

Μια ενδιαφέρουσα επέκταση των SIFT, είναι τα χαρακτηριστικά SURF (Speeded-Up Robust Features) [57]. Στηρίζονται στην λογική των SIFT, μόνο που διακρίνονται για τον ταχύτερο υπολογισμό τους, εκμεταλλευόμενα *ολοκληρωτικές εικόνες(integral images)*. Ακόμα, πειράματα έχουν δείξει ότι τα χαρακτηριστικά αυτά είναι πιο εύρωστα σε περισσότερους μετασχηματισμούς εικόνων απ'ότι τα SIFT. Στην συνέχεια, θα αναφερθούμε αναλυτικά στον αλγόριθμο εντοπισμού τέτοιων σημείων καθώς θα τα χρησιμοποιήσουμε και αργότερα στην εργασία μας.

#### 3.1.6.4 Χαρακτηριστικά SURF(Speeded-Up Robust Features)

Ο αλγόριθμος που εντοπίζει τέτοια χαρακτηριστικά, βασίζεται στον πίνακα Hessian (όπως και ο Hessian-Laplace detector), αλλά δεν ομαλοποιεί καθώς κινείται στον χώρο κλίμακας. Επίσης, η διακρίνουσα του Hessian χρησιμοποιείται για την επιλογή τόσο των συντεταγμένων όσο και της κλίμακας του χαρακτηριστικού σημείου. Ο πίνακας αυτός, προσεγγίζεται με την χρήση τετράγωνων (box-type) φίλτρων, που φαίνονται στο Σχημα 8, για τον υπολογισμό των μερικών παραγώγων στις διευθύνσεις  $x, y, xy$ . Τα 9x9 φίλτρα προσεγγίζουν Γκαουσιανές 2<sup>ης</sup> τάξης μερικές παραγώγους ( $L_{xx}, L_{yy}, L_{xy}$ ) με  $\sigma=1.2$ . Θα ονομάζουμε αυτές τις προσεγγίσεις  $D_{xx}, D_{yy}, D_{xy}$ . Για τις δύο πρώτες οι άσπρες περιοχές έχουν βάρος -1, οι μαύρες +2 και οι υπόλοιπες δεν λαμβάνονται υπ'όψιν. Για την τελευταία, οι μαύρες έχουν βάρος +1 και οι άσπρες -1.

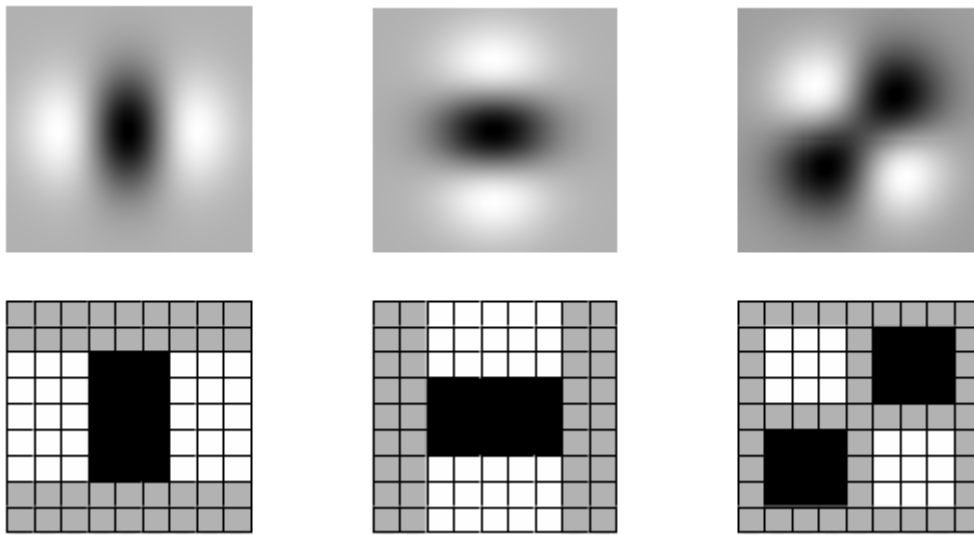
Τα βάρη διατηρούνται σε απλή μορφή, όμως πρέπει να είναι ισορροπημένα. Για αυτό, χρησιμοποιείται ένας παράγοντας  $w$  όταν υπολογίζουμε την διακρίνουσα του Hessian. Μάλιστα, θεωρητικά ο παράγοντας αυτός πρέπει να αλλάζει ανάλογα με την κλίμακα αλλά στην πράξη διατηρείται σταθερός.

Τελικά λοιπόν θα έχουμε:

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \quad (27)$$

Στη συνέχεια αναζητούμε τοπικά μέγιστα της παραπάνω συνάρτησης στον τρισδιάστατο χώρο της εικόνας (x-, y-, και κλίμακα). Ο αλγόριθμος SURF διαφέρει από άλλους στο ότι προσομοιάζει τον χώρο κλίμακας με το να εφαρμόζει πάνω στην εικόνα, φίλτρα διαφορετικού μεγέθους, αντί να υποδειγματοληπτεί την εικόνα και να εφαρμόζει Γκαουσιανό φιλτράρισμα.

Η χαμηλότερη κλίμακα που εξετάζουμε είναι το αποτέλεσμα του φίλτρου 9x9 που φαίνεται στο Σχημα 8. Οι άλλες κλίμακες λαμβάνονται εάν μεγαλώσουμε το φίλτρο διατηρώντας σταθερές τις αναλογίες του. Προφανώς όσο μεγαλώνει το φίλτρο, τόσο μεγαλώνει και η κλίμακα. Η κατασκευή του χώρου κλίμακας είναι παρόμοια με αυτή που χρησιμοποιείται στα SIFT/DoG, συνεπώς ο αριθμός  $n_o$  των οκτάβων, τα διαστήματα  $n_i$  της κάθε οκτάβας και το βήμα δειγματοληψίας  $s$  πρέπει να καθοριστούν.



Σχημα 8. Προσεγγίσεις της LoG, σε x-, y- και xy- διευθύνσεις. Πηγή: Stefan Ameling, “An Introduction to Local Feature Extraction, Description and Tracking in Matching Applications”.

Για να υπολογιστούν χαρακτηριστικά σημεία που είναι αναλλοίωτα σε περιστροφή και κλίμακα, πρέπει να γίνουν τρία βήματα. Πρώτον όλες οι τιμές κάτω από ένα κατώφλι  $t$ , αφαιρούνται. Στη συνέχεια, γίνεται non-maxima suppression συγκρίνοντας κάθε pixel με τους 26 γείτονες του (8 στην ίδια κλίμακα και 9 στις δύο γειτονικές), κρατώντας μόνο αυτά που είναι μεγαλύτερα από τους γείτονές τους. Τέλος, γίνεται μια παρεμβολή της γειτονικής πληροφορίας για να πετύχουμε ακρίβεια sub-pixel. Έτσι, η διακρίνουσα του Hessian,  $H(x, \sigma)$ , εκφράζεται σαν ανάπτυγμα Taylor 2<sup>ου</sup> βαθμού, γύρω από το σημείο που εντοπίστηκε.

$$H(x) = H + \frac{\partial H^T}{\partial x} x + \frac{1}{2} \frac{\partial^2 H^T}{\partial x^2} x \quad (28)$$

Η θέση του ακραίου σημείου  $\hat{x} = (x, y, \sigma)$  υπολογίζεται εάν πάρουμε τις παραγώγους της παραπάνω συνάρτησης και τις θέσουμε ίσες με το μηδέν. Δηλαδή:



$$\hat{x} = -\frac{\partial^2 H^{-1}}{\partial x^2} \frac{\partial H}{\partial x} = -\begin{pmatrix} d_{xx} & d_{yx} & d_{\sigma x} \\ d_{xy} & d_{yy} & d_{\sigma y} \\ d_{x\sigma} & d_{y\sigma} & d_{\sigma\sigma} \end{pmatrix}^{-1} \begin{pmatrix} d_x \\ d_y \\ d_\sigma \end{pmatrix} \quad (29)$$

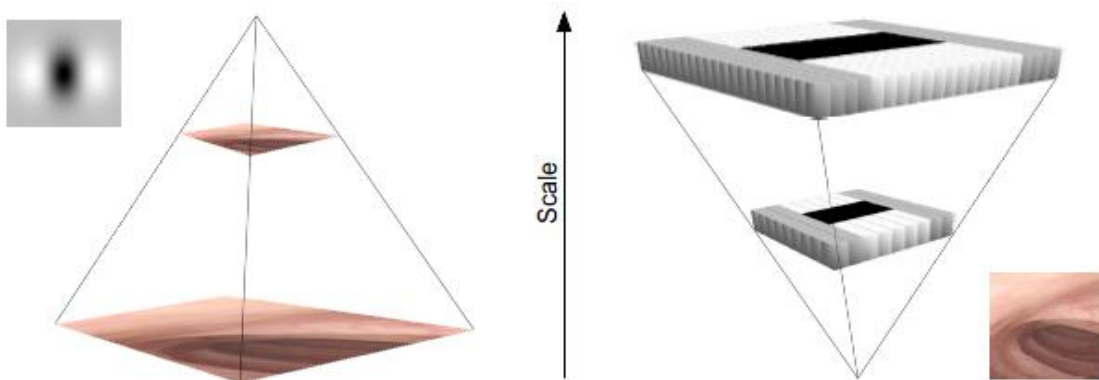
Εδώ, το  $d_x$  αναφέρεται στην  $\frac{\partial I}{\partial x}$  και το  $d_{xx}$  στην  $\frac{\partial^2 I}{\partial x^2}$ , όπου I, η εικόνα.

Για τη βελτίωση της απόδοσης του αλγορίθμου, χρησιμοποιούνται ολοκληρωτικές-αθροιστικές εικόνες (*integral images*). Μια ολοκληρωτική εικόνα  $I_\Sigma(x)$  πρακτικά είναι:

$$I_\Sigma(x) = \sum_{i=0}^{i \leq x} \sum_{j=0}^{j \leq y} I(i, j) \quad (30)$$

Χρειάζονται μόλις τρεις προσθέσεις για τον υπολογισμό του αθροίσματος των pixel μιας οποιασδήποτε παραλληλόγραμμης περιοχής. Αυτό έχει σημαντική βελτίωση στην ταχύτητα του αλγορίθμου, καθώς ο υπολογισμός των αθροισμάτων είναι ανεξάρτητος του χρόνου. Κατ'επέκταση ο υπολογισμός των φίλτρων διαφόρων μεγεθών είναι και αυτός ανεξάρτητος του χρόνου.

Η παραπάνω διαδικασία εξασφαλίζει το αναλλοίωτο ως προς την κλίμακα. Μένει να εξασφαλίσουμε το ίδιο και για την περιστροφή. Αυτό, επιτυγχάνεται με το να βρούμε τον κυρίαρχο προσανατολισμό του χαρακτηριστικού σημείου. Ο προσανατολισμός αυτός, εντοπίζεται με τον υπολογισμό των αποκρίσεων των κυματίων Haar (Haar wavelets) μεγέθους  $4\sigma$  (εδώ  $\sigma$  είναι η κλίμακα που εντοπίσαμε το χαρακτηριστικό σημείο), για ένα σύνολο pixel σε ακτίνα  $6\sigma$  γύρω από το χαρακτηριστικό σημείο.



**Σημια 9.** Αριστερά: Παραδοσιακή προσέγγιση του χώρου κλίμακας. Δεξιά: Τα SURF αφήνουν την εικόνα αμετάβλητη και αλλάζουν μόνο το μέγεθος του φίλτρου. Πηγή: Stefan Ameling, “An Introduction to Local Feature Extraction, Description and Tracking in Matching Applications”.

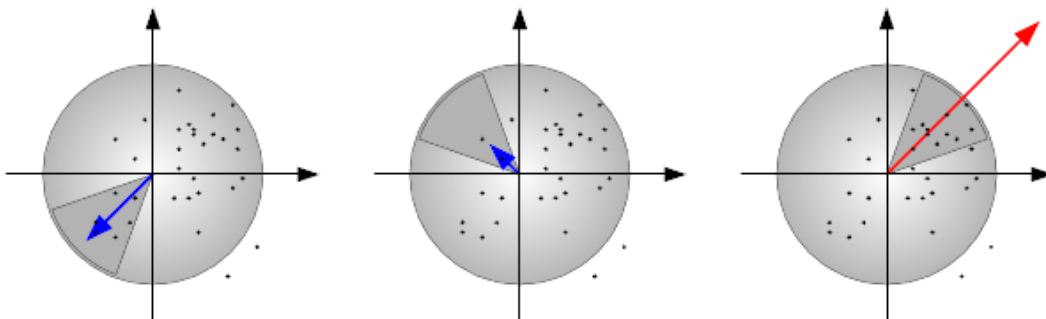


**Σχημα 10.** Το αριστερό φίλτρο υπολογίζει την παράγωγο στην  $y$ - διεύθυνση, ενώ το δεξί στην  $x$ - διεύθυνση. Πηγή: Stefan Ameling, “An Introduction to Local Feature Extraction, Description and Tracking in Matching Applications”.

Τα κυμάτια Haar είναι φίλτρα που χρησιμοποιούνται όπως φαίνεται και στο Σχημα 10 για τον υπολογισμό των μερικών παραγώγων. Οι μαύρες περιοχές έχουν βάρος  $+1$ , ενώ οι άσπρες  $-1$ . Οι αποκρίσεις υπολογίζονται χρησιμοποιώντας ολοκληρωτικές εικόνες για μεγαλύτερη ταχύτητα. Για τον υπολογισμό της διεύθυνσης, δεν χρησιμοποιούνται όλα τα pixel στον κύκλο ακτίνας  $6\sigma$ . Αντίθετα γίνεται μια δειγματοληψία με βήμα  $\sigma$ . Στη συνέχεια κάθε μια από τις επιλεγμένες αποκρίσεις, σταθμίζεται με μια Γκαουσιανή με κέντρο το χαρακτηριστικό σημείο και απόκλιση  $2.5\sigma$ . Ο προσανατολισμός επιλέγεται με το να περιστρέφουμε ένα κυκλικό τμήμα που καλύπτει γωνία  $\frac{\pi}{3}$  γύρω από το χαρακτηριστικό σημείο. Σε κάθε θέση που δείχνει αυτό το κυκλικό τμήμα, οι αποκρίσεις στους  $x$ - και  $y$ - άξονες αθροίζονται και φτιάχνουν ένα νέο διάνυσμα. Το μεγαλύτερο από αυτά τα διανύσματα, καθορίζει και τον κυρίαρχο προσανατολισμό όπως φαίνεται στο Σχημα 11.

Το βασικό που απομένει είναι η κατασκευή του SURF-Descriptor, δηλαδή του διανύσματος που περιγράφει το χαρακτηριστικό σημείο. Πρακτικά, περιέχει την πληροφορία για τον τρόπο που κατανέμονται οι εντάσεις των γειτονικών pixel μέσα στην γειτονιά του χαρακτηριστικού σημείου, που εξαρτάται από την κλίμακα. Για να επιτύχουμε το αναλλοίωτο ως προς την κλίμακα, όλοι οι υπολογισμοί βασίζονται σε μετρήσεις σχετικές με την κλίμακα  $\sigma$  στην οποία εντοπίσαμε το χαρακτηριστικό σημείο.

Το 1<sup>ο</sup> βήμα είναι ο σχεδιασμός ενός τετράγωνου παραθύρου μεγέθους  $20\sigma$  γύρω από το χαρακτηριστικό σημείο, προσανατολισμένο προς τον κυρίαρχο προσανατολισμό, έτσι που όλοι οι υπολογισμοί που ακολουθούν να γίνονται σε σχέση με αυτόν. Το παράθυρο χωρίζεται σε 16 υποπεριοχές και σε κάθε μια από αυτές, υπολογίζονται οι αποκρίσεις σε σωμάτια Haar



**Σχημα 11.** Η διαδικασία ανάθεσης προσανατολισμού για τα χαρακτηριστικό SURF σημείο. Πηγή: Stefan Ameling, “An Introduction to Local Feature Extraction, Description and Tracking in Matching Applications”.

μεγέθους  $2\sigma$  για 25 κανονικά κατανομημένα σημεία.

Έπειτα για κάθε υποπεριοχή υπολογίζονται τα  $\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|$ , όπου  $d_x, d_y$  είναι οι αποκρίσεις των σωμάτων Haar στις οριζόντιες και κάθετες διευθύνσεις, σε σχέση πάντα με τον κυρίαρχο προσανατολισμό. Συνεπώς έχουμε ένα τελικό διάνυσμα  $4 \cdot 16 = 64$  το οποίο αποτελεί τον περιγραφέα του SURF χαρακτηριστικού σημείου. Η διαδικασία αυτή εξασφαλίζει το αναλλοίωτο σε περιστροφή, αλλαγή κλίμακας, φωτεινότητα και μετά από κανονικοποίηση, σε αντίθεση.

Μια σημαντική παρατήρηση η οποία πρέπει να γίνει εδώ είναι ότι γενικά, μέθοδοι που βασίζονται στον εντοπισμό χαρακτηριστικών σημείων, για να είναι πετυχημένες πρέπει να εφαρμόζονται σε εικόνες που έχουν κάποια χαρακτηριστικά και εύκολα εντοπίσιμα αντικείμενα, προκειμένου να είναι αποτελεσματικές. Για παράδειγμα, κατάλληλες εικόνες μπορεί να σχετίζονται με δρόμους, δωμάτια, τοπία και άλλα, ενώ οι πλέον ακατάλληλες εικόνες είναι οι ιατρικές. Στις τελευταίες, χρησιμοποιούνται είτε area-based μέθοδοι, ή διαφορετικά υπάρχει ανθρώπινη επίβλεψη και καθοδήγηση, που εισάγει επιπλέον χαρακτηριστικά.

### 3.1.7 Ιστογράμματα Εικόνων

Μια διαφορετική προσέγγιση στην σύγκριση δύο εικόνων, δεν βασίζεται σε χαρακτηριστικά αντικείμενα, σημεία, γωνίες, ή περιοχές, αλλά σε μια εντελώς διαφορετική αναπαράσταση της εικόνας μέσω ιστογραμμάτων. Υπάρχουν πολλές μορφές ιστογραμμάτων, ανάλογα με τον χρωματικό χώρο στον οποίο δουλεύουμε, για παράδειγμα RGB ή HIS.

Έστω λοιπόν  $H$  το ιστόγραμμα της εικόνας. Το μέγεθος  $n$  του  $H$  εξαρτάται από το πόσες τιμές μπορεί να πάρει το κάθε pixel της εικόνας. Για 8bit στον χρωματικό χώρο RGB θα έχουμε  $n = 2^8 = 256$  στάθμες.

Έστω τώρα  $I_{(x,y)}$  η εικόνα. Ο υπολογισμός του ιστογράμματος της, είναι πολύ απλός και γίνεται ως εξής:

1. Αρχικά ισχύει  $H(i) = 0, \forall i = 1, \dots, n$ .
2. Για κάθε σημείο  $(x, y)$  της εικόνας, εφαρμόζουμε:  $H_{(I(x,y))} + 1$

Τα ιστογράμματα εικόνων, μπορούν εν συνεχεία να συγκριθούν και να βγάλουν διάφορα αποτελέσματα ομοιότητας μεταξύ των αντίστοιχων εικόνων, ανάλογα με τη σχέση ομοιότητας που εφαρμόζουμε.

Μια επιτυχημένη μέθοδος εφαρμογής σύγκρισης ιστογραμμάτων για αντιστοίχιση εικόνων, παρουσιάζεται στο [58]. Όπως περιγράφεται εκεί, η διαδικασία αναγνώρισης του χώρου στον οποίο πάρθηκε μια εικόνα, έχει ως εξής:

1. Αρχικά φτιάχνουμε το ιστόγραμμα της εικόνας. Πρακτικά δημιουργούμε 6 ιστογράμματα, για κάθε χρωματικό χώρο: red, green, blue, hue, saturation, value-intensity.
2. Στη συνέχεια για καθένα από αυτά τα ιστογράμματα, ψάχνουμε στη βάση των εικόνων μας, το καλύτερο ταίριασμα. Υπάρχουν πολλοί τρόποι να αποφασίσουμε για τον βαθμό ομοιότητας μεταξύ δύο ιστογραμμάτων: νόρμες L1, L2, τομή ιστογραμμάτων, απόκλιση Jeffrey, στατιστική μέθοδος  $\chi^2$ . Επίσης, υπάρχουν μέθοδοι για 1 προς 1 ταίριασμα, όπως τετραγωνική απόσταση, απόσταση ταιριάσματος και η απόσταση Kolmogorov-Smirnov. Τα πειράματα έδειξαν ότι η καλύτερη μετρική για να συγκρίνουμε δύο ιστογράμματα, είναι η απόκλιση Jeffrey. Εάν  $H, K$  τα δύο ιστογράμματα τότε η απόκλιση Jeffrey ορίζεται ως εξής:

$$d(H, K) = \sum_i (h_i \log(\frac{2h_i}{h_i + k_i}) + k_i \log(\frac{2k_i}{h_i + k_i})), \quad (31)$$

όπου  $h_i, k_i$  η  $i$ -οστή στάθμη.

Βλέπουμε ότι όταν  $h_i \approx k_i$  τότε το άθροισμα για την  $i$ -οστή στάθμη είναι περίπου μηδέν.

3. Κάθε χρωματικός χώρος, «ψηφίζει» το σημείο στον χώρο στον οποίο θεωρεί ότι τραβήχτηκε η εικόνα, και δίνει ένα μέτρο εμπιστοσύνης για αυτή την απόφαση. Πρακτικά αυτό σημαίνει ότι κάθε χρωματικό κανάλι ψηφίζει την καλύτερη εικόνα (αυτή με την μικρότερη ελάχιστη απόκλιση). Ακόμα υπολογίζεται ένα μέτρο εμπιστοσύνης  $c_b$  όπου  $b$  το συγκεκριμένο χρωματικό κανάλι. Θα ισχύει:

$$c_b = 1 - \frac{d_m}{\min_{i \neq m} \{d_i\}}, \quad \text{όπου}$$

$d_m$  : η ελάχιστη απόσταση από όλες τις εικόνες της βάσης

$d_i$  : η ελάχιστη απόσταση για όλες τις άλλες εικόνες της βάσης

Προφανώς το διάστημα εμπιστοσύνης παίρνει τιμές μεταξύ 0 και 1. Όσο πιο μεγάλο, τόσο πιο βέβαιο είναι το χρωματικό κανάλι για την επιλογή του. Εάν για παράδειγμα όμως υπάρχει και μια  $2^{\text{η}}$  εικόνα που έχει εξίσου καλό ταίριασμα, τότε ο βαθμός εμπιστοσύνης είναι μικρός. Στο τέλος του βήματος αυτού, θα έχουμε λοιπόν 6 βαθμούς εμπιστοσύνης (ή γενικά τόσους όσα τα χρωματικά κανάλια που χρησιμοποιούμε).

4. Στο τελευταίο βήμα, τα χρωματικά κανάλια «ψηφίζουν». Κάθε κανάλι έχει ένα κατώφλι, το οποίο ο βαθμός εμπιστοσύνης του, πρέπει να ξεπερνά για να ψηφίσει. Υπάρχουν τρία ενδεχόμενα:
  - i. Όλα τα κανάλια ψηφίζουν την ίδια εικόνα. Τότε ο συνολικός δείκτης εμπιστοσύνης υπολογίζεται από την σχέση:

$$c_{total} = \sum_{b=1}^6 (c_b - \tau_b), \text{ όπου } \tau_b \text{ το κατώφλι εμπιστοσύνης για το } b \text{ χρωματικό}$$

κανάλι. Εάν ο τελικός βαθμός ξεπερνά ένα (μικρό) κατώφλι τότε θεωρούμε ότι βρήκαμε σωστά την μετάβαση.

- ii. Τα χρωματικά κανάλια ψηφίζουν διαφορετικές εικόνες, οπότε το σύστημα είναι μπερδεμένο και δεν μπορεί να αποφασίσει.
- iii. Κανένα χρωματικό κανάλι δεν ξεπερνά το κατώφλι και συνεπώς δεν μπορεί να ψηφίσει. Το σύστημα τότε είναι αβέβαιο.

## 3.2 Χάρτες

Για την μοντελοποίηση και επίλυση του προβλήματος SLAM είναι αναγκαίο όπως φάνηκε και παραπάνω να μπορέσει κανείς με κάποιο τρόπο να περιγράψει το περιβάλλον και τον χώρο. Όπως είπαμε, υπάρχουν τρία είδη χαρτών: *τοπολογικοί*, *μετρικοί* και *υβριδικοί*.

Οι τοπολογικοί χάρτες χρησιμοποιούν γράφους για να αναπαραστήσουν το περιβάλλον, και δεν χρησιμοποιούν καθόλου γεωμετρική ή μετρική πληροφορία. Με άλλα λόγια δεν ενδιαφέρονται για την ακριβή περιγραφή του περιβάλλοντος, αλλά αναπαριστούν τον κόσμο σε ένα πιο αφηρημένο (υψηλότερο) επίπεδο. Αυτή η ανάγκη προκύπτει συχνά και ιδιαίτερα στην καθημερινότητά μας: για παράδειγμα, όταν περιγράφουμε ένα σπίτι δεν συνηθίζουμε να εξετάζουμε τις διαστάσεις των δωματίων, αλλά μας ενδιαφέρει το ποια είναι τα δωμάτια, πού βρίσκονται και πώς συνδέονται μεταξύ τους. Συνεπώς, στους τοπολογικούς χάρτες έχουμε κόμβους που αναπαριστούν διακριτά σημεία-χώρους του περιβάλλοντος και ακμές που δείχνουν τις συνδέσεις μεταξύ των κόμβων αυτών. Τα βασικά πλεονεκτήματα των τοπολογικών χαρτών είναι η απλότητα της κατασκευής τους, η ανεξαρτησία τους από σφάλματα οδομετρίας, η ευκολία με την οποία αντιμετωπίζουν βρόχους κ.α. Στο [59] υπάρχει μια πολύ αναλυτική και μαθηματική ανάλυση των τοπολογικών χαρτών.

Από την άλλη, οι μετρικοί χάρτες εμπεριέχουν μετρική πληροφορία και χρησιμοποιούν διάφορα μοντέλα: γράφους, διαγράμματα Voronoi, πλέγματα κατάληψης, μοντέλα τμημάτων, μοντέλα κόμβων και μοντέλα κυρτών πολυγώνων. Χρησιμοποιούνται όταν θέλει κανείς να έχει την ακριβή θέση του ρομπότ σε ένα σύστημα συντεταγμένων και μια όσο γίνεται πιο λεπτομερή αναπαράσταση του περιβάλλοντος. Όμως είναι ιδιαίτερα ευαίσθητοι σε σφάλματα οδομετρίας και εν γένει πολύ πιο δύσκολοι στην επεξεργασία και στον χειρισμό τους.

Τέλος, οι υβριδικοί χάρτες συνδυάζουν χαρακτηριστικά μετρικών και τοπολογικών προσεγγίσεων.

### 3.2.1 Τοπολογικοί Χάρτες-Μέθοδοι Αναπαράστασης

Όπως αναφέραμε, ένας τοπολογικός χάρτης αναπαρίσταται με την βοήθεια ενός γράφου. Ο γράφος αποτελείται από ένα σύνολο κόμβων  $V$  που συνδεόνται μέσω ενός συνόλου ακμών  $E$ . Μια ακμή συνήθως συμβολίζεται ως  $(e_1, e_2)$  και υποδηλώνει ότι ο κόμβος  $e_1$  συνδέεται με τον κόμβο  $e_2$ . Οι κόμβοι αυτοί ονομάζονται άκρα της ακμής και καλούνται γείτονες. Οι γράφοι

διακρίνονται σε *κατευθυνόμενους* και *μη κατευθυνόμενους*. Κατευθυνόμενος, είναι ένας γράφος στον οποίο ο συμβολισμός  $(e_1, e_2)$  έχει αυστηρή σημασία: υπάρχει ακμή που να συνδέει τον κόμβο  $e_1$  με τον κόμβο  $e_2$ , χωρίς απαραίτητα να ισχύει το αντίστροφο. Αντίθετα σε ένα μη κατευθυνόμενο γράφο ισχύει:  $(e_1, e_2) \Leftrightarrow (e_2, e_1)$ . Επίσης, ενδέχεται κάθε ακμή να χαρακτηρίζεται από κάποιο βάρος. Τέτοια βάρη συνήθως μοντελοποιούν φυσικά μεγέθη, όπως απόσταση, χρόνος, κόστος, χωρητικότητα, κέρδος κτλ.

Στην επιστήμη των υπολογιστών υπάρχουν δύο βασικοί τρόποι αναπαράστασης ενός γράφου. Ο πρώτος είναι με τον Πίνακα Γειννίασης. Ουσιαστικά είναι ένας πίνακας  $A = [a_{i,j}]_{N \times N}$ , όπου  $N$  είναι ο αριθμός των κόμβων. Θα ισχύει:

$$a_{i,j} = \begin{cases} 1, & \text{αν } (v_i, v_j) \in E \\ 0, & \text{αλλιώς} \end{cases} \quad (32)$$

Εάν ο γράφος είναι μη κατευθυνόμενος τότε ο πίνακας  $A$  είναι συμμετρικός. Εάν οι ακμές χαρακτηρίζονται από βάρη τότε τα στοιχεία του πίνακα  $a_{i,j}$  αντικαθίστανται από τα βάρη αυτά. Στα θετικά της αναπαράστασης αυτής είναι η ευκολία συντήρησης και υλοποίησής του, καθώς και ο σταθερός χρόνος για τον έλεγχο υπάρξης ακμής. Από την άλλη οι απαιτήσεις σε μνήμη είναι αρκετά μεγάλες, της τάξης του  $O(N^2)$ .

Μια άλλη μορφή αναπαράστασης είναι ο Πίνακας Πρόσπτωσης. Εδώ έχουμε ένα πίνακα  $B = [b_{i,j}]_{N \times M}$ , όπου  $N$  ο αριθμός των κόμβων και  $M$  ο αριθμός των ακμών. Για μη κατευθυνόμενους γράφους, θα ισχύει:

$$b_{i,j} = \begin{cases} 1, & \text{αν ο κόμβος } v_i \text{ ακουμπά στην ακμή } x_j \\ 0, & \text{αλλιώς} \end{cases} \quad (33)$$

Εάν έχουμε κατευθυνόμενο γράφο, τότε:

$$b_{i,j} = \begin{cases} 1, & \text{αν η ακμή } x_j \text{ εισέρχεται στον κόμβο } v_i \\ -1, & \text{αν η ακμή } x_j \text{ εξέρχεται από τον κόμβο } v_i \\ 0, & \text{αλλιώς} \end{cases} \quad (34)$$

με την σύμβαση των -1 και 1 να εξαρτάται από την υλοποίηση.

Αντιλαμβάνεται κανείς ότι η χρήση ενός εκ των δύο πινάκων αναπαράστασης, εξαρτάται από την μορφή του γράφου που θέλουμε να αναπαραστήσουμε. Εάν ο γράφος έχει πολλούς κόμβους με λίγες ακμές, τότε ο Πίνακας Πρόσπτωσης έχει μικρότερες απαιτήσεις σε μνήμη. Εάν ο γράφος έχει πολλές ακμές, τότε καλύτερα κανείς να χρησιμοποιήσει τον Πίνακα Γειννίασης του οποίου το μέγεθος εξαρτάται αποκλειστικά από τον αριθμό κόμβων.

Τέλος, υπάρχουν δύο αντίστοιχες αναπαραστάσεις με χρήση λιστών αντί για πινάκων, δηλαδή η Λίστα Γειννίασης και Λίστα Πρόσπτωσης. Το πλεονέκτημα των λιστών είναι ότι εν γένει απαιτούν λιγότερη μνήμη ειδικά για αραιούς γράφους. Επίσης είναι πιο εύκολος ο εντοπισμός των γειτόνων ενός γράφου. Από την άλλη, ο έλεγχος υπάρξης μιας ακμής δεν έχει πλέον σταθερό χρόνο, ενώ η υλοποίηση μιας λίστας είναι πιο δύσκολη από αυτή ενός πίνακα.

### 3.3 Πιθανοτική Προσέγγιση

Είδαμε παραπάνω πως όλοι οι αλγόριθμοι χαρτογράφησης (και σε ένα βαθμό οι αλγόριθμοι αυτοεντοπισμού θέσης) είναι μη ντετερμινιστικοί, δηλαδή φτιάχνουν πιθανοτικά μοντέλα του ρομπότ και του περιβάλλοντος (χάρτη) και χρησιμοποιούν τα μοντέλα αυτά για να μετατρέψουν τις αισθητηριακές ενδείξεις σε ένα χάρτη [23].

Αυτό συμβαίνει επειδή το πρόβλημα της χαρτογράφησης και άρα του SLAM χαρακτηρίζεται εγγενώς από αβεβαιότητα και θόρυβο μετρήσεων στην οδομετρία και τους αισθητήρες (laser, infrared κτλ). Οι πιθανοτικοί αλγόριθμοι προσεγγίζουν το πρόβλημα μοντελοποιώντας με ρητό τρόπο τους διάφορους θορύβους και τις επιπτώσεις τους στις μετρήσεις.

Το πρόβλημα της χαρτογράφησης, είναι τόσο δύσκολο που η επιστημονική κοινότητα έχει αναγκασθεί να το προσεγγίσει στην πλειοψηφία των περιπτώσεων με ένα μαθηματικό εργαλείο: τον Κανόνα του Bayes.

#### 3.3.1 Ο Κανόνας του Bayes

Ο κανόνας του Bayes είναι ιδιαίτερα σημαντικό μαθηματικό εργαλείο και χρησιμοποιείται σε πάρα πολλές εφαρμογές. Ο Κανόνας του Bayes είναι ο εξής:

$$p(x | d) = \eta p(d | x) p(x) \quad (35)$$

Αυτό το οποίο υποδηλώνει πρακτικά ο κανόνας αυτός είναι το εξής: Η πιθανότητα να έχει προκύψει μια κατάσταση (ενδεχόμενο)  $x$ , δεδομένης μιας ληφθείσας παρατήρησης  $d$ , ισούται με την πιθανότητα να έχει προκύψει η κατάσταση  $x$ , επί την πιθανότητα να ληφθεί η παρατήρηση  $d$  όντας στην ενδεχόμενη κατάσταση  $x$ . Στη ρομποτική, εάν θεωρήσουμε ως  $x$  μια ποσότητα που ψάχνουμε (για παράδειγμα, την θέση) και  $d$  τα αισθητηριακά δεδομένα (οδομετρία, μετρητές απόστασης), ο Κανόνας του Bayes μας λέει ότι το πρόβλημα μπορεί να λυθεί εάν πολλαπλασιάσουμε δύο όρους:  $p(d | x)$  και  $p(x)$ . Ο 1<sup>ος</sup> όρος είναι η πιθανότητα του να παρατηρήσουμε την μέτρηση  $d$  υποθέτοντας ότι βρισκόμαστε στην θέση  $x$ . Έτσι, ο 1<sup>ος</sup> όρος ονομάζεται και *μοντέλο παραγωγής*, αφού περιγράφει την διαδικασία παραγωγής μετρήσεων σε διάφορους κόσμους-χάρτες  $x$ . Ο 2<sup>ος</sup> όρος ονομάζεται *ο πρότερος (the prior)*. Περιγράφει την πιθανότητα που δίνουμε στο να είμαστε στη θέση  $x$ , χωρίς να έχει έρθει καμία επιπλέον πληροφορία από αισθητήρες. Τέλος, το  $\eta$  είναι μια σταθερά κανονικοποίησης που εξασφαλίζει ότι το συνολικό αποτέλεσμα είναι όντως μια κατανομή πιθανότητας.

Ας δούμε λίγο πιο αναλυτικά τί συμβαίνει: στην χαρτογράφηση τα δεδομένα έρχονται διαδοχικά. Ας υποθέσουμε δύο μορφές δεδομένων: αισθητηριακά δεδομένα  $z$  και δεδομένα ελέγχου  $u$ . Υποθέτουμε ότι τα δεδομένα έρχονται με την ακόλουθη σειρά:

$$z_1, u_1, z_2, u_2, \dots$$

Οι δείκτες υπονοούν χρόνο. Συγκεκριμένα  $z_t$  το αισθητηριακό δεδομένο τη στιγμή  $t$ , ενώ  $u_t$  είναι ο εντολή κίνησης το διάστημα  $[t-1, t)$ . Κάποιες φορές, χρησιμοποιούνται δεδομένα οδομετρίας αντί για εντολές κίνησης γιατί αυτά αναπαριστούν καλύτερα την πραγματική κίνηση του ρομπότ. Το ζήτημα λοιπόν είναι να βρούμε πώς θα επεξεργαστούμε τα παραπάνω δεδομένα. Ο πιο διαδεδομένος τρόπος είναι το Φίλτρο Bayes.

### 3.3.2 Το Φίλτρο Bayes

Στο τομέα της ρομποτικής χαρτογράφησης, το κυρίαρχο σχήμα προσέγγισης για να συμψηφίζει δεδομένα της μορφής  $z_1, u_1, z_2, u_2, \dots$ , είναι το *Φίλτρο Bayes*, που συνδέεται άμεσα με το *Φίλτρο Κάλμαν*, τα *Κρυφά Μαρκοβιανά Μοντέλα*, τα *Δυναμικά Δίκτυα Bayes* και οι *Μερικώς Παρατηρήσιμες Μαρκοβιανές Διαδικασίες Απόφασης*. Πρακτικά το Φίλτρο Bayes επεκτείνει τον Κανόνα του Bayes σε διαχρονικά (temporal) προβλήματα εκτίμησης. Είναι ένας αναδρομικός εκτιμητής που υπολογίζει μια ακολουθία posterior πιθανοτήτων πάνω σε ποσότητες που δεν μπορούν να παρατηρηθούν άμεσα, όπως ένας χάρτης. Ας ονομάσουμε την ποσότητα αυτή *κατάσταση* και ας τη συμβολίσουμε ως  $x_t$ , όπου  $t$  ο χρόνος. Το Φίλτρο Bayes υπολογίζει την δεσμευμένη πιθανότητα της κατάστασης  $x_t$  με βάση τον ακόλουθο τύπο:

$$p(x_t | z^t, u^t) = \eta p(z_t | x_t) \int p(x_t | u_t, x_{t-1}) p(x_{t-1} | z^{t-1}, u^{t-1}) dx_{t-1} \quad (36)$$

Εδώ, ο συμβολισμός  $z^t, u^t$  σημαίνει ότι:

$$z^t = \{z_1, z_2, \dots, z_t\}$$

$$u^t = \{u_1, u_2, \dots, u_t\}$$

Η εξίσωση (36) βλέπουμε ότι είναι αναδρομική, δηλαδή η πιθανότητα  $p(x_t | z^t, u^t)$  υπολογίζεται με βάση την ίδια πιθανότητα που υπολογίστηκε στο προηγούμενο βήμα.

Μια πολύ σημαντική παρατήρηση είναι ότι η ποσότητα  $x_t$  πρέπει να περιλαμβάνει όλα εκείνα τα μεγέθη που επηρεάζουν τις αισθητηριακές μετρήσεις. Στην ρομποτική χαρτογράφηση αυτά είναι εν γένει δύο: ο χάρτης και η θέση του ρομπότ σε αυτόν. Για αυτό, όταν χρησιμοποιούμε πιθανοτικές τεχνικές, η χαρτογράφηση είναι ένα πρόβλημα που τόσο ο χάρτης όσο και η θέση πρέπει να υπολογιστούν ταυτόχρονα. Έτσι αν συμβολίσουμε ως  $s$  την θέση του ρομπότ και ως  $m$  τον χάρτη, θα έχουμε:

$$p(s_t, m_t | z^t, u^t) = \eta p(z_t | s_t, m_t) \int \int p(s_t, m_t | u_t, s_{t-1}, m_{t-1}) p(s_{t-1}, m_{t-1} | z^{t-1}, u^{t-1}) ds_{t-1} dm_{t-1} \quad (37)$$

Επειδή όμως οι περισσότεροι αλγόριθμοι υποθέτουν στατικό κόσμο, μπορούμε να αφαιρέσουμε τον δείκτη  $t$  όταν αναφερόμαστε στον χάρτη  $m$ . Συνεπώς θα έχουμε:

$$p(s_t, m | z^t, u^t) = \eta p(z_t | s_t, m) \int p(s_t, m | u_t, s_{t-1}, m) p(s_{t-1}, m | z^{t-1}, u^{t-1}) ds_{t-1} \quad (38)$$



Η διαφορά της (37) από την (38) είναι πρακτικά τεράστια. Η (38) δεν απαιτεί πλέον ολοκλήρωση στο σύνολο των πιθανών χαρτών. Μια τέτοια ολοκλήρωση θα ήταν πρακτικά αδύνατη γιατί οι χάρτες έχουν συνήθως απαγορευτική διαστασιμότητα.

Για να την υπολογίσουμε την (38) χρειαζόμαστε δύο πιθανότητες: την  $p(s_t | u_t, s_{t-1})$  και την  $p(z_t | s_t, m)$ . Οι κατανομές αυτές υποθέτουμε ότι είναι ανεξάρτητες από τον χρόνο. Έτσι συνήθως τις γράφουμε ως  $p(s | u, s')$  και  $p(z | s, m)$ . Και οι δύο είναι μοντέλα παραγωγής του ρομπότ και του περιβάλλοντός του. Η  $p(z | s, m)$  είναι το *μοντέλο αντίληψης ή παρατήρησης*, δηλαδή το πως οι αισθητήρες του ρομπότ αντιλαμβάνονται το περιβάλλον δεδομένης της θέσης του ρομπότ και του χάρτη. Η  $p(s | u, s')$  περιγράφει το πως ο έλεγχος  $u$  επιδρά στην κατάσταση  $s$ , δηλαδή μας λέει τι θα γίνει αν το ρομπότ όντας στην κατάσταση  $s'$  του ασκηθεί ένας έλεγχος (εντολή κίνησης)  $u$ . Συνήθως, η σχέση αυτή ονομάζεται *μοντέλο κίνησης* του ρομπότ.

### 3.3.3 Υπόθεση Πλήρους Κατάστασης – Αλυσίδα Markov

Στην παραπάνω ανάλυση εξηγήσαμε την σημασία του να αφαιρέσουμε τον δείκτη του χρόνου όταν αναφερόμαστε στον χάρτη. Τώρα θα κάνουμε μια ακόμα παραδοχή η οποία είναι καθοριστικής σημασίας για την πρακτική υλοποίηση ενός αλγορίθμου χαρτογράφησης. Η σχέση (38) λέει ότι πρέπει να ολοκληρώσουμε για το σύνολο των παρελθόντων θέσεων του ρομπότ στον χάρτη. Αυτό θα ήταν καταστροφικό για την υλοποίηση της (38) σε πραγματικό χρόνο. Στην πράξη θεωρούμε ότι ισχύει η *Υπόθεση Πλήρους Κατάστασης*. Συγκεκριμένα θεωρούμε ότι κάθε χρονική στιγμή, η μετάβαση από την κατάσταση  $x_t$  στην κατάσταση  $x_{t+1}$  εξαρτάται μόνο από την προηγούμενη (τελευταία) κατάσταση. Αυτό σημαίνει ότι δεν χρειάζεται να λάβουμε υπ'όψιν όλες τις προηγούμενες θέσεις του ρομπότ. Πρακτικά, θεωρούμε ότι κάθε κατάσταση  $x_t$  εμπεριέχει και ενσωματώνει όλες τις προηγούμενες παρατηρήσεις και εντολές ελέγχου του ρομπότ.

Εκφράζοντας το παραπάνω μαθηματικά, και χρησιμοποιώντας τον παραπάνω συμβολισμό θα έχουμε ότι πλέον  $p(x_t | z^t, u^t) \Rightarrow p(x_t | z_t, u_t)$  και  $p(z_t | x^t) \Rightarrow p(z_t | x_t)$ .

Συνεπώς η μεταβολή του διανύσματος κατάστασης, είναι μια *Μαρκοβιανή Αλυσίδα*.

### 3.3.4 Κατανομή Πιθανότητας στο Χώρο Κατάστασης

Μία ακόμη σημαντική έννοια των πιθανοτικών προσεγγίσεων στον τομέα της ρομποτικής, είναι αυτή του μέτρου πεποίθησης. Το μέτρο πεποίθησης αντικατοπτρίζει την εσωτερική γνώση του ρομπότ για την κατάσταση του περιβάλλοντος αλλά και του ιδίου. Ωστόσο, η κατάσταση είναι ένα μέγεθος που δεν μπορεί να μετρηθεί απευθείας. Τουναντίον, το ρομποτικό όχημα οφείλει να συμπεράνει τη θέση του και το χάρτη από τα δεδομένα εισόδου. Υπάρχει, επομένως, ο διαχωρισμός μεταξύ της πραγματικής κατάστασης και του εσωτερικού μέτρου πεποίθησης του ρομπότ αναφορικά με αυτή. Οι πιθανοτικές τεχνικές αναπαριστούν τα

μέτρα πεποιθήσης μέσω κατανομών πιθανότητας υπό συνθήκες. Μία κατανομή πιθανότητας αποδίδει μία πιθανότητα (ή τιμή βάρους) σε κάθε δυνατή υπόθεση για την πραγματική κατάσταση. Κατ' ουσίαν, τα μέτρα πεποιθήσης είναι εκ των υστέρων κατανομές πιθανότητας επί των μεταβλητών κατάστασης, υπό τη συνθήκη των διαθέσιμων δεδομένων.

Στο πλαίσιο ενός αλγορίθμου, η υλοποίηση της κατανομής πιθανότητας μπορεί να γίνει με πολλούς διαφορετικούς τρόπους καθέννας από τους οποίους στηρίζεται σε διαφορετικές υποθέσεις όσον αφορά στις πιθανότητες παρατήρησης και μετάβασης, καθώς και στο μέτρο πεποιθήσης της αρχικής κατάστασης. Η διαφορά αυτή στα δομικά χαρακτηριστικά κάθε τεχνικής οδηγεί σε αλγορίθμους με διαφορετικές υπολογιστικές απαιτήσεις και αξιοπιστία. Οι δύο κύριες κατηγορίες στις οποίες μπορούν να διαχωριστούν οι διάφορες κατανομές πιθανότητας είναι οι παραμετρικές και οι μη παραμετρικές. Οι παραμετρικές κατανομές – με πλέον διαδεδομένη την κατανομή Gauss – οδηγούν σε υλοποιήσεις με φίλτρα που χρησιμοποιούν ένα σύνολο από παραμέτρους για την απόδοση των πιθανοτήτων στο εύρος του πεδίου ορισμού των κατανομών, όπως για παράδειγμα τα φίλτρα Κάλμαν. Στην παρούσα διπλωματική ωστόσο, θα μας απασχολήσουν υλοποιήσεις που καταλήγουν σε μη παραμετρικά φίλτρα.

### 3.3.5 Μή Παραμετρικά Φίλτρα

Κύριο χαρακτηριστικό των μή παραμετρικών φίλτρων είναι η χρήση ενός πεπερασμένου πλήθους διακριτών τιμών για την περιγραφή του χώρου κατάστασης (για παράδειγμα της θέσης του ρομπότ). Αυτό μπορεί να γίνει με διάφορους τρόπους. Κάποιες μέθοδοι προσεγγίζουν τον χώρο κατάστασης αντλώντας δείγματα από μια εκ των υστέρων κατανομή πιθανότητας. Άλλες, χωρίζουν το χώρο σε υποπεριοχές, κάθε μια από τις οποίες περιγράφεται από μια διακριτή τιμή. Το τεράστιο πλεονέκτημα των μεθόδων αυτών είναι ότι η ακρίβεια καθορίζεται από το σχεδιαστή. Με άλλα λόγια το πλήθος των συντελεστών είναι ελεύθερη επιλογή. Για παράδειγμα, στο *φίλτρο σωματιδίων* η ακρίβεια της προσέγγισης του χώρου κατάστασης εξαρτάται από τον αριθμό των σωματιδίων που εξελίσσονται κάθε χρονική στιγμή. Το *φίλτρο σωματιδίων* (particle filtering) είναι μια προσέγγιση που χρησιμοποιείται σε μεγάλο βαθμό και είναι η μια από τις δύο βασικές προσεγγίσεις. Τα *Φίλτρα Ιστογράμματος* είναι η άλλη κύρια προσέγγιση.

#### 3.3.5.1 Φίλτρο Ιστογράμματος

Τα Φίλτρα Ιστογράμματος αποτελούν την δεύτερη μή παραμετρική προσέγγιση εκτίμησης κατανομής πιθανότητας ενός συνεχούς χώρου μέσω διακριτοποίησης (πεπερασμένων τιμών). Στη μέθοδο αυτή, ο χώρος χωρίζεται σε ένα πλήθος περιοχών (πεπερασμένο), και η αναπαράσταση της κατανομής πιθανότητας γίνεται με χρήση ενός ιστογράμματος. Το ιστογράμμα αναθέτει σε κάθε περιοχή μια μοναδική πιθανότητα να βρίσκεται το ρομπότ σε αυτή. Στην εργασία αυτή θα ασχοληθούμε με μια υποπερίπτωση του Φίλτρου Ιστογράμματος, που αφορά τον διακριτό χώρο, το λεγόμενο Διακριτό Φίλτρο Bayes.

### **3.3.5.2 Φίλτρο Σωματιδίων**

Για μια αρκετά αναλυτική παρουσίαση του Φίλτρου Σωματιδίων, ο αναγνώστης παραπέμπεται στην ενότητα 2.4.5.

Έχοντας αναλύσει τα βασικά θεωρητικά ζητήματα που αφορούν την χαρτογράφηση και τον αυτοεντοπισμό θέσης, καθώς και την πιθανοτική προσέγγιση και το ταίριασμα εικόνων, ερχόμαστε τώρα να εξετάσουμε το συνολικό πρόβλημα του *SLAM* (*Simultaneous Localisation and Mapping*). Θα αναλύσουμε κάποιες βασικές συνιστώσες του προβλήματος, όπως το *μοντέλο κίνησης*, το *μοντέλο παρατήρησης*, το *διάνυσμα κατάστασης* ενώ στη συνέχεια θα αναλύσουμε θεωρητικά το *Διακριτό Φίλτρο Bayes* και το πώς αυτό θα μας βοηθήσει στην επίλυση του προβλήματος.

Συγκεκριμένα στην Ενότητα 4.1 θα εξετάσουμε το πρόβλημα του SLAM υπό το πρίσμα του Διακριτού Φίλτρου Bayes. Όπως θα δούμε, ένας αλγόριθμος οπτικού SLAM έχει δύο σημαντικές συνιστώσες: το μοντέλο κίνησης και το μοντέλο παρατήρησης. Τα μοντέλα αυτά, ουσιαστικά περιγράφουν την αλληλεπίδραση του ρομπότ με τον κόσμο, αφού αποτελούν τους νόμους που διέπουν την κίνηση του ρομπότ στον χώρο, και τον τρόπο που αντιλαμβάνεται το περιβάλλον του αντίστοιχα.

Στην ενότητα 4.2, θα μελετήσουμε την εφαρμογή αυτών των δύο μοντέλων στο Διακριτό Φίλτρο Bayes, δηλαδή το *βήμα πρόβλεψης* (που σχετίζεται άμεσα με το μοντέλο κίνησης), και το *βήμα ανανέωσης* (που σχετίζεται άμεσα με το μοντέλο παρατήρησης).

#### **4.1 Επίλυση του SLAM με Διακριτό Φίλτρο Bayes – Μοντέλα Κίνησης και Παρατήρησης και το Διάνυσμα Κατάστασης**

Στην Ενότητα 2.4.1 αναφερθήκαμε εισαγωγικά στο πρόβλημα του SLAM. Εδώ θα αναλύσουμε πιο λεπτομερώς το πώς θεωρητικά δουλεύει ένας αλγόριθμος SLAM βασισμένος στο Διακριτό Φίλτρο Bayes. Δεν θα μας απασχολήσει κάποια συγκεκριμένη μέθοδος χαρτογράφησης ή αυτοεντοπισμού θέσης, καθώς θεωρητικά όλες όσες αναφέραμε παραπάνω μπορούν να χρησιμοποιηθούν, αλλά τα βήματα που χρησιμοποιούν οι περισσότεροι αλγόριθμοι SLAM που βασίζονται στο Διακριτό Φίλτρο Bayes.

Το SLAM είναι εγγενώς ένα πάρα πολύ δύσκολο πρόβλημα με πολύ μεγάλη αβεβαιότητα. Προσπαθούμε να φτιάξουμε ένα χάρτη, χωρίς να γνωρίζουμε την ακριβή θέση του ρομπότ και ταυτόχρονα προσπαθούμε να εντοπίσουμε την θέση του ρομπότ μέσα στον χάρτη, χωρίς όμως ο χάρτης να είναι απόλυτα ακριβής. Εάν ένα από τα δύο – ακριβής θέση, ή τέλειος χάρτης - καταστεί γνωστό με κάποιο τρόπο, τότε το νέο πρόβλημα είναι τετριμένο σε σχέση με το αρχικό. Τελικός στόχος του SLAM είναι να υπολογίσει την πιθανότητα

$$p(s_t, m_t \mid z_t, u_t) \quad (39)$$

όπου  $s_i$  η θέση του ρομπότ,  $m_i$  ο χάρτης ή τα ορόσημα,  $z_i$  οι παρατηρήσεις και  $u_i$  στοιχεία οδομετρίας, ή εντολές ελέγχου. Με άλλα λόγια, στόχος είναι για κάθε χρονική στιγμή να βρούμε τον χάρτη και την θέση του ρομπότ μέσα σε αυτόν, με δεδομένα τις παρατηρήσεις και την οδομετρία (ή τις εντολές ελέγχου).

Το ρομπότ ξεκινά την εξερεύνηση από μια γνωστή ή άγνωστη κατάσταση (εμείς θα θεωρήσουμε το αρχικό σημείο γνωστό με ικανοποιητική ακρίβεια) και κινείται στον χώρο (αυτόνομο ή με δικές μας εντολές). Καθώς το ρομπότ κινείται στον χώρο, θεωρούμε ότι έρχονται αισθητηριακά δεδομένα: πληροφορία οδομετρίας από τους κωδικοποιητές του ρομπότ, εικόνα από την κάμερα και πιθανώς άλλες πληροφορίες. Ένας αλγόριθμος SLAM, βασισμένος στο Διακριτικό Φίλτρο Bayes λειτουργεί σε δύο βήματα.

Το 1<sup>ο</sup> βήμα που ονομάζεται *βήμα πρόβλεψης*, είναι η εκτίμηση της νέας θέσης του ρομπότ, η οποία υπολογίζεται βάσει της προηγούμενης και του μοντέλου κίνησης. Η εκτίμηση αυτή, έχει εγγενώς μια αβεβαιότητα –την οποία μπορούμε να μοντελοποιήσουμε ή να την αγνοήσουμε. Ο χάρτης παραμένει αμετάβλητος σε αυτό το βήμα.

Το 2<sup>ο</sup> βήμα, που ονομάζεται *βήμα ανανέωσης*, αφορά την επεξεργασία και ενσωμάτωση της επιπλέον πληροφορίας από τους αισθητήρες. Η πληροφορία αυτή προσπαθεί να αντιστοιχισθεί με κάποιο χώρο (grid σε ένα μετρικό χάρτη, ή κόμβο σε ένα τοπολογικό) στον υπάρχοντα χάρτη. Εδώ προφανώς υπάρχουν δύο ενδεχόμενα: το 1<sup>ο</sup> είναι η περίπτωση του να μην μπορούμε να αντιστοιχίσουμε αυτήν την πληροφορία στον υπάρχοντα χάρτη. Αυτό συνεπάγεται ότι για να συμπεριλάβουμε την νέα αυτή πληροφορία, πρέπει να επεκτείνουμε τον χάρτη - εκτός εάν για κάποιο λόγο επιλέξουμε να την αγνοήσουμε, όπως αρκετές φορές μπορεί να συμβαίνει σε έναν αλγόριθμο Φίλτρου Kalman. Το 2<sup>ο</sup> ενδεχόμενο είναι ότι η πληροφορία παραπέμπει σε κάποιο χαρακτηριστικό του χάρτη. Τότε συνήθως έχουμε την δυνατότητα να ενσωματώσουμε την πληροφορία αυτή, και να αυξήσουμε την ακρίβεια του χάρτη, και κατ'επέκταση την ακρίβεια της θέσης του ρομπότ μέσα σε αυτόν.

Τα δύο παραπάνω βήματα εκτελούνται συνεχώς με το ένα να διαδέχεται το άλλο. Το μεν βήμα πρόβλεψης, βασίζεται στο μοντέλο κίνησης, ενώ το βήμα ανανέωσης βασίζεται στο μοντέλο παρατήρησης.

Στη συνέχεια, θα μελετήσουμε τα μοντέλα κίνησης και παρατήρησης. Σε ό,τι αφορά το μοντέλο κίνησης, θα αναφερθούμε σε διάφορες προσεγγίσεις που γίνονται στην διεθνή βιβλιογραφία. Για το μοντέλο παρατήρησης, θα κάνουμε μια αναφορά στην δουλειά που έκανε ο Γ. Καλπάκας και στη συνέχεια θα περιγράψουμε την επέκταση που κάναμε στο κομμάτι αυτό, χρησιμοποιώντας τα χαρακτηριστικά SURF.

#### 4.1.1 Μοντέλο Κίνησης

Έστω ότι η θέση ενός ρομποτικού οχήματος, συμβολίζεται με  $\varphi = (x, y, \theta)$ . Το μοντέλο κίνησης προσπαθεί κάθε χρονική στιγμή να υπολογίσει την νέα θέση  $\varphi'$ , ή πιο μαθηματικά το  $P(\varphi' | \varphi, o)$ , όπου  $o = (d, t)$  είναι τα στοιχεία της οδομετρίας για την μεταφορική και περιστροφική κίνηση του ρομπότ αντίστοιχα.

Η ακρίβεια του μοντέλου κίνησης, καθορίζεται από τον σχεδιαστή και είναι πολύ κρίσιμη καθώς θέτει όρια στον θόρυβο που μπορούμε να μοντελοποιήσουμε, στην ακρίβεια των αποτελεσμάτων, και στις επιτρεπτές κινήσεις που θεωρούμε για το ρομπότ. Εμείς θα εξετάσουμε διάφορες μορφές κινηματικών μοντέλων για διαφορεικά ρομπότ.

Η 1<sup>η</sup> περίπτωση είναι η ιδανική:

$$\begin{aligned}x' &= x + D \cos(\theta + T) \\y' &= y + D \sin(\theta + T) \\ \theta' &= \theta + T \bmod 2\pi\end{aligned}\tag{40}$$

όπου  $D$  είναι η πραγματική απόσταση που διένυσε το ρομπότ και  $T$  είναι η στροφή. Η αδυναμία ενός τέτοιου μοντέλου είναι ότι απλούστατα υποθέτει ανεξαρτησία περιστροφικής και μεταφορικής κίνησης. Δηλαδή το ρομπότ ή απλώς στρίβει ή κινείται μόνο ευθεία.

Μια βελτίωση αυτού του μοντέλου, είναι να επιτρέψει ταυτόχρονη περιστροφική και μεταφορική κίνηση:

$$\begin{aligned}x' &= x + D \cos(\theta + T / 2) \\y' &= y + D \sin(\theta + T / 2) \\ \theta' &= \theta + T \bmod 2\pi\end{aligned}\tag{41}$$

Το μοντέλο αυτό υποθέτει ότι η περιστροφική ταχύτητα του ρομπότ είναι σταθερή καθόλη τη διάρκεια της κίνησης και ότι το ρομπότ κινείται μόνο στη κατεύθυνση που κοιτάζει. Δεν λαμβάνει υπ'όψιν του ότι στην πράξη η απόσταση που θα διανύσει, δεν είναι ευθεία αλλά τόξο. Παρ'όλα αυτά, για μικρό  $T$  το σφάλμα είναι μικρό και μπορεί να θεωρηθεί θόρυβος.

Ένα ακόμα καλύτερο μοντέλο, θα πρέπει να λάβει υπ'όψιν την δυνατότητα του ρομπότ να κινήθει σε μια κατεύθυνση όπου δεν καθορίζεται από την αρχική και τελική του γωνία. Ένα τέτοιο μοντέλο θα μπορεί να αντιμετωπίσει μεταβαλλόμενες περιστροφικές ταχύτητες:

$$\begin{aligned}x' &= x + D \cos \theta^* \\y' &= y + D \sin \theta^* \\ \theta' &= \theta + T \bmod 2\pi\end{aligned}\tag{42}$$

όπου  $\theta^*$  είναι η πραγματική γωνία κίνησης του ρομπότ. Όπως βλέπουμε σε αυτή την περίπτωση, η κατεύθυνση κίνησης έχει εκφραστεί ανεξάρτητα από τα  $\theta$  και  $T$ , κάτι που επιτρέπει κίνηση σε κατεύθυνση διαφορετική από εκείνη που κοιτάζει το ρομπότ. Στην πράξη βέβαια, αυτό προϋποθέτει ότι το ρομπότ διαθέτει κωδικοποιητές στους τροχούς που μπορούν να διαβαστούν ανεξάρτητα, προκειμένου να υπολογίσουμε την παράμετρο  $\theta^*$ .

Στην πράξη το τελευταίο μοντέλο, αλλάζει και χρησιμοποιείται στην ακόλουθη μορφή:

$$\begin{aligned}
x' &= x + D \cos\left(\theta + \frac{T}{2}\right) + C \cos\left(\theta + \frac{T + \pi}{2}\right) \\
y' &= y + D \sin\left(\theta + \frac{T}{2}\right) + C \sin\left(\theta + \frac{T + \pi}{2}\right) \\
\theta' &= \theta + T \bmod 2\pi
\end{aligned} \tag{43}$$

Αυτό που έγινε, είναι να προσεγγίσουμε το  $\theta^*$  με το  $(\theta + \frac{T}{2})$ , το οποίο ονομάζουμε και ως κύριο άξονα κατεύθυνση. Το C είναι μια επιπλέον μεταβλητή μεταφορικής κίνησης που περιγράφει την κίνηση στην διεύθυνση που είναι κάθετη στον κύριο άξονα κατεύθυνσης.

Το παραπάνω μοντέλο κίνησης μπορεί εύκολα να συμπεριλάβει θόρυβο. Αναμένουμε οι πραγματικές τιμές D, T να είναι κανονικά κατανομημένες γύρω από τις τιμές d, t της οδομετρίας, με την μέση τιμή να εξαρτάται γραμμικά από τα d, t ενώ την διασπορά να εξαρτάται από το άθροισμα των τετραγώνων τους. Δηλαδή:

$$\begin{aligned}
D &\sim N(d\mu_{D_d} + t\mu_{D_t}, d^2\sigma_{D_d}^2 + t^2\sigma_{D_t}^2) \\
T &\sim N(d\mu_{T_d} + t\mu_{T_t}, d^2\sigma_{T_d}^2 + t^2\sigma_{T_t}^2)
\end{aligned} \tag{44}$$

όπου  $\mu_{A_b}$ , είναι ο συντελεστής που δείχνει την επιρροή του όρου οδομετρίας b, στη μέση τιμή της κατανομής του A.

Για μια πιο αναλυτική παρουσίαση περί μοντέλων κίνησης και μοντελοποίησης του θορύβου, ο αναγνώστης παραπέμπεται στο [60].

#### 4.1.2 Μοντέλο Παρατήρησης

Το μοντέλο παρατήρησης αποτελεί ένα ιδιαίτερα σημαντικό κομμάτι του αλγορίθμου για την επίλυση του SLAM καθώς εκεί γίνεται η επεξεργασία των οπτικών αισθητηριακών δεδομένων, προκειμένου να εκτιμηθεί η θέση του ρομπότ και να σχεδιασθεί ο χάρτης του περιβάλλοντος. Μαθηματικά το μοντέλο παρατήρησης προσπαθεί να υπολογίσει την πιθανότητα  $p(z_t | x_k)$ , όπου  $z_t$  είναι η παρατήρηση τη στιγμή t και  $x_k$  είναι στην περίπτωσή μας ο κ-οστός κόμβος του χάρτη. Αυτή είναι η πιθανότητα να έχουμε την παρατήρηση  $z_t$  με δεδομένο ότι βρισκόμαστε στον κόμβο  $x_k$ .

Στην παρούσα εργασία, ο μόνος αισθητήρας που χρησιμοποιείται για αυτό το σκοπό είναι η μονοσκοπική κάμερα που τοποθετείται πάνω στο ρομποτικό όχημα. Η μονοσκοπική αυτή κάμερα παίρνει εικόνες του χώρου ακριβώς στην ευθεία κατεύθυνσης κίνησης του ρομπότ.

Το μοντέλο παρατήρησης αποτέλεσε ένα σημαντικό κομμάτι της εργασίας αυτής. Για το λόγο αυτό θα χωρίσουμε την ενότητα αυτή σε δύο μέρη. Στο 1<sup>ο</sup>, θα γίνει μια σύντομη παρουσίαση της προσπάθειας που έγινε από τον Γεώργιο Καλπάκα στην διπλωματική του εργασία, με

τίτλο «Εκτίμηση Θέσης και Τοπολογική Χαρτογράφηση σε Αυτόνομο Κινούμενο Ρομποτικό Όχημα με βάση Οπτική Πληροφορία» και αφορά ένα μοντέλο παρατήρησης βασιζόμενο στα χρωματικά ιστογράμματα εικόνας. Στο 2<sup>ο</sup> μέρος, θα αναλύσουμε μια επέκταση και βελτίωση αυτής, με την οποία ασχοληθήκαμε ιδιαίτερα στην παρούσα εργασία

#### 4.1.2.1 Σύγκριση Εικόνων με Ιστογράμματα HSI

Η εικόνα που λαμβάνεται από την κάμερα, μετατρέπεται σε τρία μονοδιάστατα ιστογράμματα, που αντιστοιχούν στα κανάλια  $H, S, I$  (τόνος, κορεσμός και φωτεινότητα αντίστοιχα). Η χρήση αυτού του χρωματικού χώρου γίνεται γιατί διαχωρίζει τα δύο πιο σημαντικά χαρακτηριστικά χρωματικού τόνου και κορεσμού από την εύκολα μεταβαλλόμενη και θορυβώδη χρωματική ένταση, η οποία γενικά δεν λαμβάνεται υπ' όψιν στη υπόλοιπη διαδικασία.

Στην Ενότητα 3.1.7 περιγράψαμε το πώς λαμβάνουμε το ιστόγραμμα μιας εικόνας. Έστω λοιπόν  $H$  το ιστόγραμμα της εικόνας που αντιπροσωπεύει την πληροφορία του περιβάλλοντος από την τρέχουσα θέση του ρομπότ:

$$H = (H_H \ H_S \ H_I)^T, \text{ όπου ισχύει}$$

$$H_{i_{1 \times 256}}, i \in [H, S, I]$$

Μιλήσαμε παραπάνω για τον υπολογισμό της δεσμευμένης πιθανότητας  $p(z_t | x_k)$ . Στην προκειμένη περίπτωση, για τον ορισμό της, χρησιμοποιούμε ένα μέτρο ομοιότητας μεταξύ δύο εικόνων, και συγκεκριμένα την *Απόκλιση Jeffrey*. Εάν τώρα ονομάσουμε  $H_z = (H_{z,H}, H_{z,S}, H_{z,I})^T$  το ιστόγραμμα της τρέχουσας παρατήρησης και  $H_k = (H_{k,H}, H_{k,S}, H_{k,I})^T$  το ιστόγραμμα που αναφέρεται στην εκτιμώμενη θέση του ρομπότ, τότε θα έχουμε, ότι η πιθανότητα να λάβει το ρομπότ την παρατήρηση  $z$ , δεδομένης της θέσης  $x_k$ , είναι:

$$p(z_t | x_k) = \frac{1}{2} \sum_{i=H,S} \text{JeffDiv}(i) \quad (45)$$

$$\text{JeffDiv}(ch) = \sum_{i=1}^{256} h_{z,ch,i} \log\left(\frac{2h_{z,ch,i}}{h_{z,ch,i} + h_{k,ch,i}}\right) + h_{k,ch,i} \log\left(\frac{2h_{k,ch,i}}{h_{z,ch,i} + h_{k,ch,i}}\right) \quad (46)$$



όπου  $h_{x,y,i}$  είναι η τιμή της  $i$ -οστής στάθμης του καναλιού  $Y$  του ιστογράμματος  $H_x$ , όπου  $x \in [z, k]$  και  $Y \in [H, S]$ .

#### 4.1.2.2 *Επέκταση του Μοντέλου Παρατήρησης- Σύγκριση Εικόνων με χαρακτηριστικά SURF (Speeded-Up Robust Features)*

Στην Ενότητα 3.1.6.4 περιγράψαμε τον αλγόριθμο εντοπισμού χαρακτηριστικών σημείων τύπου SURF. Όπως είπαμε τα σημεία αυτά παρουσιάζουν ιδιαίτερη ανοχή σε περιστροφή, αλλαγή κλίμακας, φωτεινότητα και αντίθεση. Αυτά τα χαρακτηριστικά, καθιστούν τα σημεία αυτά μια καλή βάση για να συγκρίνει κανείς εικόνες. Μάλιστα κάτι τέτοιο έχει γίνει στα [61], [56] και σε διάφορες άλλες εργασίες ταιριάσματος εικόνων, αναγνώρισης αντικειμένων και αλγορίθμους SLAM, όπως στο [62].

Η βασική ιδέα όταν δουλεύει κανείς με τέτοια χαρακτηριστικά σημεία από διαφορετικές εικόνες και θέλει να τα συγκρίνει προκειμένου να εντοπίσει κάποιο αντικείμενο, ή να δει εάν αντιστοιχούν στο ίδιο εν τέλει σημείο είναι η ακόλουθη: Έστω ότι έχουμε μια εικόνα  $I_1$  που αντιστοιχεί σε μια νέα παρατήρηση, και την εικόνα  $I_2$  που αντιστοιχεί σε μια εικόνα με την οποία θέλουμε να την συγκρίνουμε (ας υποθέσουμε ότι αντιστοιχεί στον κόμβο  $x_k$ ) και ακόμα έστω  $F_1, F_2$  το σύνολο των χαρακτηριστικών σημείων τους, αντίστοιχα. Για παράδειγμα, όπως είπαμε η  $I_1$  μπορεί να είναι μια νέα παρατήρηση του ρομπότ από την τρέχουσα θέση του, ενώ η  $I_2$  η εικόνα που αντιστοιχεί σε μια εκτιμώμενη θέση. Για κάθε στοιχείο του  $F_1$  (ή για κάποια από τα στοιχεία του, επιλεγμένα με κάποιο τρόπο) που δεν είναι τίποτε άλλο από ένα διάνυσμα 64 ή 128 στοιχείων, χρησιμοποιούμε την Ευκλείδεια απόσταση για να υπολογίσουμε την διαφορά του με όλα τα στοιχεία του  $F_2$ . Μαθηματικά, αυτό σημαίνει ότι αν  $p, q^j$  είναι δύο διανύσματα SURF-Descriptors από τα σύνολα  $F_1, F_2$ , αντίστοιχα, τότε θα έχουμε ότι η Ευκλείδεια απόσταση  $d_p^j$  του διανύσματος  $p$  από το διάνυσμα  $q^j$ , είναι :

$$d_p^j = \sqrt{\sum_{i=1}^n (q_i^j - p_i)^T (q_i^j - p_i)} \quad (47)$$

όπου εδώ,  $n$  είναι το μέγεθος του διανύσματος-περιγραφητή, και ο δείκτης  $j$  δείχνει το  $j$ -οστό διάνυσμα του συνόλου  $F_2$ . Τελικά καταλήγουμε σε ένα διάνυσμα

$$D_p = \{d_p^1, d_p^2, \dots, d_p^k\}, \text{ όπου } k \text{ είναι ο αριθμός των διανυσμάτων του συνόλου } F_2.$$

Το διάνυσμα αυτό εκφράζει τις απόστασεις του  $p$  του συνόλου  $F_1$ , από όλα τα διανύσματα του  $F_2$ .

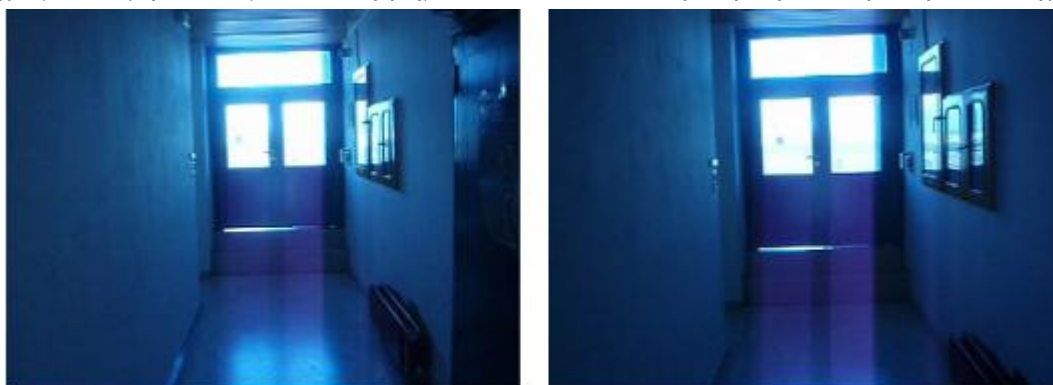
Δύο χαρακτηριστικά σημεία  $p, q^x$ , λέμε ότι ταιριάζουν, εάν έχουν την μικρότερη Ευκλείδεια απόσταση  $d_p^{m_1}$  και ταυτόχρονα ισχύει:

$$\frac{d_p^{m_1}}{d_p^{m_2}} \leq \theta \quad (48)$$

όπου  $d_p^{m_2}$  είναι η 2<sup>η</sup> μικρότερη απόσταση του χαρακτηριστικού σημείου  $p$ , από το σύνολο των σημείων  $F_2$ , και  $\theta$  είναι ένα κατώφλι που επιλέγουμε. Στο [61], ο David Lowe προτείνει  $\theta = 0.8$ . Μετά από πειράματα όμως βρήκαμε ότι στην δική μας περίπτωση, το 0.8 είναι κάπως αισιόδοξο και μπορεί να οδηγήσει σε λάθος συμπεράσματα, για αυτό και χρησιμοποιούμε μικρότερο κατώφλι, μεταξύ 0.65-0.75.

Από την παραπάνω ανάλυση, είναι εύκολο να δει κανείς ότι ουσιαστικά αυτό που κάνουμε είναι να ψάχνουμε εξαντλητικά για να βρούμε το καλύτερο ταιρίασμα. Άλλωστε, στην πράξη δεν υπάρχουν αλγόριθμοι που μπορούν να βρουν τον ακριβή κοντινότερο γείτονα σε χώρους υψηλής διαστασιμότητας, εκτός από την εξαντλητική έρευνα [61]. Διάφορες βελτιώσεις έχουν προταθεί, οι οποίες οδηγούν σε σωστό αποτέλεσμα με κάποια πιθανότητα, όπως ο αλγόριθμος Best-Bin-First (BBF), ο οποίος περιγράφεται στο [61] και βασίζεται σε k-d δέντρα. Για την επιτάχυνση της διαδικασίας, συγκρίνουμε μόνο τα χαρακτηριστικά εκείνα SURF που έχουν το ίδιο πρόσημο στην Laplacian τους. Η Laplacian ξεχωρίζει φωτεινά τμήματα της εικόνας από σκοτεινό φόντο συνεπώς σημεία με διαφορετική Laplacian δεν μπορούν ποτέ να ταιριάζουν.

Η παραπάνω ανάλυση περιγράφει πότε δύο χαρακτηριστικά σημεία SURF ταιριάζουν μεταξύ τους. Απώτερος στόχος είναι να αποφανθούμε εάν τελικά δύο εικόνες ταιριάζουν ή γενικότερα να εκτιμήσουμε τον βαθμό στον οποίο ταιριάζουν. Είναι σαφές και από τα όσα έχουμε αναφέρει, ότι γενικά το ζήτημα του αν δύο εικόνες ταιριάζουν, δηλαδή αντιστοιχούν

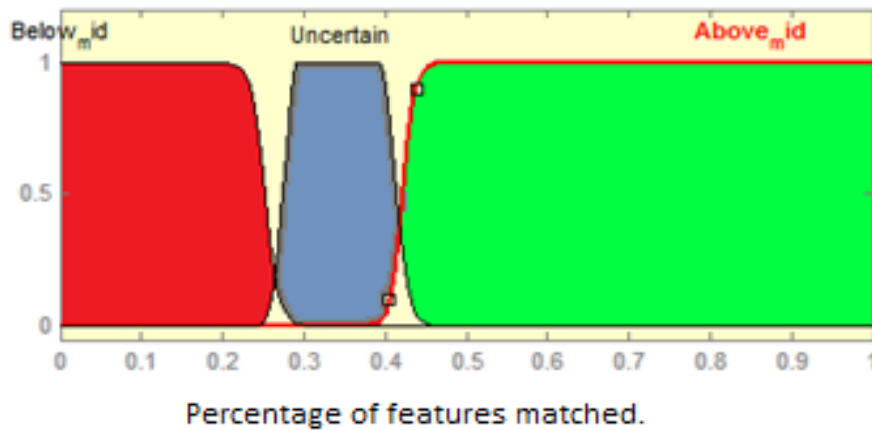


**Σχημα 12.** Δύο εικόνες που μοιάζουν πάρα πολύ. Στην δεξιά, η πόρτα δεν φαίνεται πια. Αντιστοιχούν στον ίδιο τοπολογικό κόμβο του χάρτη;

στο ίδιο χώρο του περιβάλλοντος είναι αρκετά υποκειμενικό και για τον λόγο αυτό ακόμα και οι άνθρωποι μπορεί να έχουμε διαφορετική άποψη. Για παράδειγμα, αντιστοιχούν οι δύο εικόνες στο Σχημα 12 στον ίδιο χώρο του περιβάλλοντος;

Τέτοια παραδείγματα προκύπτουν συνέχεια σε ένα τέτοιο περιβάλλον. Σε μια προσπάθεια να μοντελοποιήσουμε αυτή την υποκειμενικότητα και αβεβαιότητα του προβλήματος, καταλήξαμε στην λύση της ασαφούς λογικής. Στόχος μας φυσικά είναι να καταλήξουμε σε μια τιμή για την πιθανότητα κατανομής  $p(z_t | x_k)$ . Συνεπώς η έξοδος του συστήματος της ασαφούς λογικής θα είναι η τιμή αυτής της πιθανότητας.

Το ζήτημα είναι η είσοδος του συστήματος αυτού. Μια απλή και πολύ αποτελεσματική λογική είναι η εξής: Η είσοδος του συστήματος, είναι το ποσοστό των χαρακτηριστικών σημείων της τελευταίας παρατήρησης ( $z_t$ ), που ταιρίαξαν με χαρακτηριστικά σημεία της εικόνας της εκτιμώμενης θέσης ( $x_k$ ). Έτσι, αν  $N$  είναι ο αριθμός των σημείων της τελευταίας παρατήρησης  $z_t$ , και  $M$  ο αριθμός των σημείων που καταφέραμε να ταιριάξουμε, τότε, η είσοδος στο σύστημα είναι το ποσοστό:



Σχημα 13.. Η είσοδος του συστήματος ασαφούς λογικής.

$$Fuzzy\ Input = \frac{M}{N} \quad (49)$$

Μετά από πειραματισμό, βρήκαμε ότι αν μια εικόνα καταφέρει να ταιριάξει επιτυχώς τουλάχιστον το 40-50% των σημείων της, με μια άλλη, τότε οι δύο αυτές εικόνες ταιριάζουν. Λιγότερο από 20%, σημαίνει ότι οι εικόνες δεν ταιριάζουν, ενώ οτιδήποτε μεταξύ 20-40% είναι αβέβαιο, αλλά στην πλειοψηφία των περιπτώσεων θεωρούμε (και γενικά ισχύει) ότι οι εικόνες ταιριάζουν.

### 4.1.3 Το Διάνυσμα Κατάστασης

Το διάνυσμα κατάστασης, είναι όλα εκείνα τα μεγέθη τα οποία προσπαθεί να υπολογίσει ο αλγόριθμος του SLAM. Στην περίπτωση μας, το διάνυσμα κατάστασης περιλαμβάνει την θέση του ρομπότ καθώς και τον χάρτη. Ας δούμε αναλυτικά τί σημαίνει αυτό.

Επειδή το ρομπότ θεωρούμε ότι κινείται στο επίπεδο, η θέση του περιγράφεται από τρία νούμερα: η θέση  $(x, y)$  στο επίπεδο και ο προσανατολισμός του.

Σε ό,τι αφορά τον χάρτη, η αναπαράσταση του κόσμου που θα υιοθετήσουμε καθ'όλη τη μελέτη μας, είναι τοπολογική. Συνεπώς, ο χάρτης αποτελείται από διακριτούς κόμβους, καθένας εκ των οποίων αντιπροσωπεύει μια διακριτή περιοχή του χώρου. Στόχος μας είναι να χρησιμοποιήσουμε το Διακριτό Φίλτρο Bayes για να υπολογίσουμε την πιθανότητα να βρίσκεται το ρομπότ σε ένα δεδομένο κόμβο του χάρτη, καθώς και να βρούμε το πώς αυτοί οι κόμβοι συνδέονται μεταξύ τους (τοπολογία). Μαθηματικά, αυτό συμβολίζεται ως εξής:

$$\xi(k) \in R^{(1+N^2)} : \xi(k) = [x_{pos}(k) \ x_{node,1} \ x_{node,2} \ \dots \ x_{node,N}]^T \quad (50)$$

όπου  $x_{pos}(k)$  είναι ο εκτιμώμενος κόμβος που βρίσκεται το ρομπότ τη στιγμή  $k$ , και  $x_{node,i}$  είναι το  $1 \times N$  διάνυσμα (όπου  $N$  πλήθος κόμβων του χάρτη) που περιέχει τις σχέσεις γειτνίασης του  $i$ -οστού κόμβου, με όλους τους υπόλοιπους.

## 4.2 Το Διακριτό Φίλτρο Bayes

Όπως αναφέραμε προηγουμένως, για την επίλυση του προβλήματος SLAM και την εκτίμηση σε κάθε χρονική στιγμή του διανύσματος κατάστασης θα χρησιμοποιήσουμε ένα μη παραμετρικό φίλτρο βασισμένο στο Διακριτό Φίλτρο Bayes.

Το φίλτρο αυτό, είναι ένα κλασικό εργαλείο για τον υπολογισμό των πιθανοτήτων διαφόρων ενδεχομένων, προκειμένου να μπορέσει ένα ρομπότ να εντοπίσει την θέση του και τον προσανατολισμό του. Το φίλτρο διαιρεί τον χώρο κατάστασης σε διακριτές υποπεριοχές, κάτι που ταιριάζει απόλυτα με την τοπολογική προσέγγιση του χάρτη μας. Κάθε διακριτή περιοχή του φίλτρου, αντιστοιχεί σε ένα κόμβο του χάρτη. Σε κάθε κόμβο ανατίθεται μια πιθανότητα που εκφράζει την πεποίθησή μας ότι βρίσκεται το ρομπότ εκεί. Φυσικά η ακρίβεια με την οποία εντοπίζουμε τη θέση του ρομπότ στον χάρτη εξαρτάται από την πυκνότητα των κόμβων. Κατά περίπτωση θα πρέπει να αναζητάται η βέλτιστη πυκνότητα. Γενικά, δεν θέλουμε ούτε πολύ μεγάλη ούτε πολύ μικρή πυκνότητα κόμβων: στην 1<sup>η</sup> περίπτωση θα έχουμε μεγάλη σύγχυση καθώς γειτονικοί κόμβοι θα έχουν πολύ μεγάλη

συσχέτιση, ενώ στη  $2^{\text{η}}$  θα έχουμε πολύ μικρή ακρίβεια στην θέση του ρομπότ, η οποία ενδέχεται να μην βοηθάει ιδιαίτερα.

Πρακτικά το φίλτρο πραγματοποιεί μια εκτίμηση του διανύσματος κατάστασης, συνεπώς σε αντιστοιχία με την ενότητα 4.1.3, θα έχουμε:

$$p(k) \in R^{(N+N^2)} : p(k) = [p_r(k) P_{node,1} P_{node,2} \dots P_{node,N}]^T \quad (51)$$

Όπου το  $p_r(k)$  είναι το  $1 \times N$  διάνυσμα που περιέχει την διακριτή κατανομή πιθανότητας να βρίσκεται το ρομπότ σε κάθε ένα από τους  $N$  κόμβους τη στιγμή  $k$ . Επιπλέον,  $P_{node,i}$  είναι το  $1 \times N$  διάνυσμα που περιέχει τις πιθανότητες γεινιάσης μεταξύ του  $i$ -οστού κόμβου του χάρτη και όλων των υπολοίπων. Στην πράξη βέβαια, ο αλγόριθμός μας δεν υπολογίζει τις πιθανότητες  $P_{node,i}$  αλλά απλώς βρίσκει τον τελικό γράφο που δείχνει τις σχέσεις μεταξύ των κόμβων. Πρακτικά δηλαδή φτιάχνουμε ένα ντετερμινιστικό Πίνακα Γεινιάσης. Για τον υπολογισμό των παραπάνω ποσοτήτων, το Φίλτρο Bayes εφαρμόζει διαδοχικά βήματα πρόβλεψης και ανανέωσης. Το Φίλτρο Bayes πραγματοποιεί τα βήματα αυτά, με τη βοήθεια των μοντέλων κίνησης και παρατήρησης που αναφέραμε στην Ενότητα 4.1.1 και Ενότητα 4.1.2 αντίστοιχα.

#### 4.2.1 Βήμα Πρόβλεψης

Το βήμα πρόβλεψης, προβλέπει την νέα θέση του ρομπότ την χρονική στιγμή  $t$  γνωρίζοντας μόνο την προηγούμενη θέση την χρονική στιγμή  $t-1$ , και είτε τις εντολές ελέγχου, είτε πληροφορία από την οδομετρία. Κατ'αναλογία με το φίλτρο Bayes της ενότητας 3.3.2, η πιθανότητα να βρίσκεται το ρομπότ στον κόμβο  $i$  την χρονική στιγμή  $t$  είναι:

$$p(x_{i,t} | z_{t-1}, x_{t-1}, u_t) = \sum_n \underbrace{p(x_{i,t} | x_{n,t-1}, u_t)}_{\text{μοντέλο κίνησης}} p(x_{n,t-1} | z_{t-1}, x_{t-2}, u_{t-1}) \quad (52)$$

Στη παραπάνω σχέση, ο  $1^{\text{ος}}$  όρος είναι το μοντέλο κίνησης του ρομπότ, ενώ ο  $2^{\text{ος}}$  είναι ο όρος που είχε υπολογιστεί την προηγούμενη χρονική στιγμή και αφορά την πιθανότητα να βρίσκεται το ρομπότ στον κόμβο  $n$  την χρονική στιγμή  $t-1$ .

Στην πράξη, ο αλγόριθμος σε κάθε χρονική στιγμή θεωρεί ότι το ρομπότ βρίσκεται σε ένα συγκεκριμένο κόμβο και ότι εφ'όσον βρίσκεται σε αυτόν τότε έχει μηδενική πιθανότητα να βρίσκεται σε κάποιον άλλο. Με άλλα λόγια, το παραπάνω άθροισμα τελικά εκφυλίζεται σε ένα όρο αφού μόνο για ένα  $n$  θα ισχύει  $p(x_{n,t-1} | z_{t-1}, x_{t-2}, u_{t-1}) \neq 0$  : για τον κόμβο στον οποίο αποφασίσαμε ότι βρισκόμασταν την στιγμή  $t-1$ .

#### 4.2.2 Βήμα Ανανέωσης

Στο βήμα αυτό, το φίλτρο ενσωματώνει πλέον στην εκτίμηση πιθανότητας και την νέα παρατήρηση  $z_t$  που παίρνει το ρομπότ από το περιβάλλον του, την στιγμή  $t$ . Το βήμα ανανέωσης ορίζεται γενικά ως εξής:

$$p(x_{i,t} | z_t, u_t, x_{t-1}) = \underbrace{\eta p(z_t | x_{i,t}, z_{t-1}, u_t)}_{\text{μοντέλο παρατήρησης}} \overbrace{p(x_{i,t} | z_{t-1}, u_t, x_{t-1})}^{\text{βήμα πρόβλεψης}}$$

(53)

Αξίζει να σημειωθεί ότι λόγω της Υπόθεσης Markov, γράφουμε τελικά  $p(z_t | x_{i,t})$  αντί για  $p(z_t | x^t, z^{t-1}, u^{t-1})$  (θυμίζουμε ότι  $z^{t-1} = \{z_1, z_2, \dots, z_{t-1}\}$ ) αφού όλες οι προηγούμενες παρατηρήσεις, θέσεις και εντολές ελέγχου, θεωρούμε ότι έχουν ενσωματωθεί και δεν παίζουν ρόλο στην καινούργια κατάσταση.

## 5 *Οπτικό SLAM με Κάμερα*

Έχοντας περιγράψει το πρόβλημα του SLAM και ορίσει τις επιμέρους διαστάσεις του, στο παρόν κεφάλαιο θα παρουσιάσουμε την μεθοδολογία επίλυσής του. Η μεθοδολογία που χρησιμοποιήθηκε, είναι βασισμένη στη διπλωματική εργασία του Γ. Καλπάκα με τίτλο «Εκτίμηση Θέσης και Τοπολογική Χαρτογράφηση σε Αυτόνομο Κινούμενο Ρομποτικό Όχημα με βάση Οπτική Πληροφορία». Σε αδρές γραμμές, η μεθοδολογία αυτή υλοποιούσε Τοπολογικό SLAM με χρήση μιας κάμερας, χωρίς μετρική πληροφορία. Το θεωρητικό εργαλείο στο οποίο στηριζόταν ο αλγόριθμος ήταν το Διακριτό Φίλτρο Bayes. Θα αναφερθούμε συνοπτικά σε καίρια σημεία της μεθοδολογίας αυτής, ενώ θα αναλύσουμε τα νέα στοιχεία που εισάγαμε, επεκτείνοντας την μεθοδολογία με στόχο τη βελτίωση της επίδοσης του αλγορίθμου και την διεξαγωγή πειραμάτων με ρομπότ.

Θα χωρίσουμε το κεφάλαιο αυτό σε τρία μέρη. Στην Ενότητα 5.1, θα αναφερθούμε στην αναπαράσταση των κόμβων του χάρτη.

Στην Ενότητα 5.2, θα αναφερθούμε στην μορφή των μοντέλων κίνησης και παρατήρησης του Διακριτού Φίλτρου Bayes, που υλοποιεί η παρούσα μεθοδολογία.

Επιπλέον, στην Ενότητα 5.3, θα περιγράψουμε το πώς ακριβώς αντιμετωπίζει ο αλγόριθμός την αναγνώριση ενός βρόχου.

Τέλος, στην Ενότητα 5.4 θα αναφέρουμε συνοπτικά τα βήματα του αλγορίθμου (για πιο αναλυτική παρουσίαση ο αναγνώστης παραπέμπεται στην εργασία «Εκτίμηση Θέσης και Τοπολογική Χαρτογράφηση σε Αυτόνομο Κινούμενο Ρομποτικό Όχημα με βάση Οπτική Πληροφορία»).

### **5.1 Αναπαράσταση Κόμβων με Ιστογράμματα Εικόνων και Χαρακτηριστικά SURF**

Ο χάρτης τον οποίο κατασκευάζει το ρομπότ καθώς κινείται στο περιβάλλον αποτελείται από ένα σύνολο κόμβων και ένα σύνολο σχέσεων γειτνίασης. Το τελευταίο υλοποιείται με ένα Πίνακα Γειτνίασης.

Κάθε κόμβος χαρακτηρίζεται από ένα σύνολο μεγεθών, με τα σημαντικότερα να είναι η θέση, ο προσανατολισμός και η οπτική πληροφορία. Η οπτική πληροφορία του κάθε κόμβου, αποτελείται από δύο συνιστώσες που εξάγονται από την εικόνα που αντιστοιχεί στον αντίστοιχο κόμβο: η 1<sup>η</sup> είναι τα ιστογράμματα χρωματικού τόνου και κορεσμού της εικόνας. Τα ιστογράμματα αυτά, κατασκευάζονται ταχύτατα και χρειάζονται μικρό χώρο αποθήκευσης. Επίσης, ο συγκεκριμένος χρωματικός χώρος παρουσιάζει διαχωρισμό βασικών χρωματικών συνιστωσών τόνου και κορεσμού, από την φωτεινότητα, κάτι που προσδίδει σταθερότητα στη μέθοδο και ανεξαρτησία από την φωτεινότητα. Η 2<sup>η</sup> συνιστώσα είναι τα χαρακτηριστικά SURF της εικόνας. Κάθε κόμβος χαρακτηρίζεται από ένα πίνακα

διανυσμάτων  $F_{n \times 64}$ , όπου  $n$  ο αριθμός των χαρακτηριστικών σημείων. Κάθε σημείο περιγράφεται από ένα διάνυσμα μήκους 64. Ο εντοπισμός τέτοιων σημείων, περιγράφεται αναλυτικά στην Ενότητα 3.1.6.4.

### 5.1.1 Ιστογράμματα Χρωματικού Τόνου και Κορεσμού – Κατακερματισμός Εικόνας

Στην Ενότητα 3.1.7 περιγράψαμε τη διαδικασία εξαγωγής του ιστογράμματος μιας εικόνας.

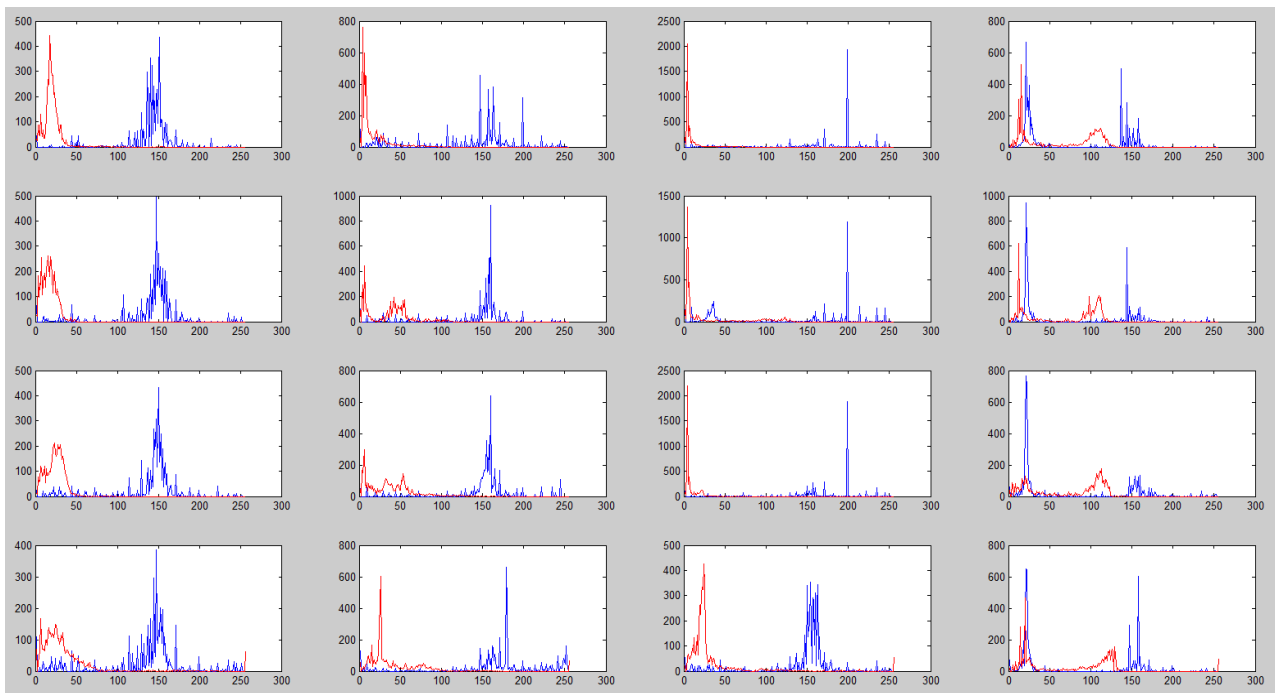
Η απλούστερη λογική για την εξαγωγή του ιστογράμματος είναι η επεξεργασία απ' ευθείας ολόκληρης της εικόνας και η εξαγωγή ενός μόνο ιστογράμματος για κάθε χρωματικό χώρο (τόνο και κορεσμό). Αυτό ναί μεν είναι πολύ απλό, αλλά έχει ένα σημαντικό μειονέκτημα: ενδέχεται ακόμα και μια μικρή αλλαγή στην εικόνα, για παράδειγμα ένα αντικείμενο που προστίθεται ή αφαιρείται, να αλλάξει σε μεγάλο βαθμό το τελικό ιστόγραμμα. Σε μια τέτοια περίπτωση, μικρές αλλαγές στις εικόνες που αντιστοιχούν στο ίδιο σημείο του χώρου, θα άλλαζαν άρδην το αποτέλεσμα της σύγκρισης και ενδέχεται να παίρναμε λάθος αποτελέσματα από το μοντέλο παρατήρησης. Αυτό το φαινόμενο, ουσιαστικά θα καθιστούσε αδύνατη την αναγνώριση βρόχων.

Ένας τρόπος να ξεπεραστεί αυτό το πρόβλημα, είναι να μην υπολογίσει κανείς απευθείας το ιστόγραμμα όλης της εικόνας, αλλά να χωρίσει πρώτα την εικόνα σε επιμέρους τμήματα, και να υπολογίσει στη συνέχεια το ιστόγραμμα για καθένα από αυτά. Εν συνεχεία δεν θα είχε παρά να συγκρίνει ομόλογα τμήματα (τμήματα που βρίσκονται στις ίδιες περιοχές των εικόνων) χρησιμοποιώντας την *Απόκλιση Jeffrey*, και εν τέλει να βγάλει ένα συνολικό βαθμό ομοιότητας. Βέβαια, ένα ερώτημα είναι το σε πόσα τμήματα πρέπει να χωρίσουμε την εικόνα για να είναι αποτελεσματική μια τέτοια μέθοδος. Εάν τα τμήματα είναι πολύ λίγα, τότε ίσως το πρόβλημα που αναφέραμε παραπάνω να μην επιλύεται. Εάν είναι πάρα πολλά, τότε το κόστος υπολογισμού και σύγκρισης όλων αυτών των ιστογραμμάτων θα αυξηθεί πάρα πολύ.



Σχημα 14. Η εικόνα που αντιστοιχεί στην εισαγωγή ενός κόμβου.





**Σχημα 15.Εσωτερική αναπαράσταση της εικόνας. Κάθε τμήμα από τα 16, περιγράφεται από ένα ιστογράμμο Hue(μπλε) και ένα ιστογράμμο Saturation(κόκκινο).**

Έχει δειχθεί πειραματικά, ότι ο χωρισμός της εικόνας σε  $4 \times 4 = 16$  τμήματα είναι ο βέλτιστος.

Στο Σχημα 15 φαίνεται το αποτέλεσμα για την εικόνα στο Σχημα 14. Όπως βλέπουμε, έχουμε συνολικά 16 τμήματα της αρχικής εικόνας, για καθένα από τα οποία έχουμε υπολογίσει τα ιστογράμματα χρωματικού τόνου και κορεσμού.

### 5.1.2 Αναπαράσταση με Χαρακτηριστικά SURF

Σαν επέκταση της παραπάνω μεθοδολογίας, ταυτόχρονα με τον υπολογισμό των ιστογραμμάτων της εικόνας, γίνεται και εντοπισμός των χαρακτηριστικών σημείων SURF που αυτή εμπεριέχει. Ο αλγόριθμος που χρησιμοποιείται είναι αυτός που περιγράφηκε στην Ενότητα 3.1.2.4 . Στο περιβάλλον που εκτελέσαμε τα πειράματα, ο αλγόριθμος αυτός εντοπίζει για κάθε εικόνα από 50 μέχρι 200 χαρακτηριστικά σημεία. Όσο πιο πλούσια είναι η εικόνα σε πληροφορία (αντικείμενα, χρώματα) τόσο πιο πολλά σημεία εντοπίζονται.

Πρακτικά αυτό που αποθηκεύει ο αλγόριθμος, είναι τα διανύσματα (με μήκος 64 ή 128 στοιχεία) που περιγράφουν αυτά τα σημεία. Ένα παράδειγμα φαίνεται στο Σχημα 16. Γενικά, σημεία SURF αποτελούν μεγάλες ακμές της εικόνας, ή έντονες γωνίες. Εδώ, να σημειώσουμε ότι κάποια σημεία SURF που εντοπίζονται, είναι πιο ισχυρά από κάποια άλλα, με την έννοια ότι εντοπίζονται από τον αλγόριθμο σε υψηλότερες κλίμακες.



Σημια 16. Χαρακτηριστικά SURF που εντοπίζονται και αποθηκεύονται ως αναπαράσταση του κόμβου.

## 5.2 Τοπολογικό SLAM με βάση Οπτική και Οδομετρική Πληροφορία

Στην ενότητα αυτή, θα αναλύσουμε τα μοντέλα κίνησης και παρατήρησης του αλγορίθμου. Τα μοντέλα αυτά, αποτελούν την καρδιά του Διακριτού Φίλτρου Bayes με την έννοια ότι υπολογίζουν την διακριτή κατανομή πιθανότητας που αντιστοιχεί σε κάθε κόμβο του χάρτη, και συνεπώς πραγματοποιούν την εκτίμηση θέσης και την ανανέωση του χάρτη με βάση τις νέες παρατηρήσεις.

Στην Ενότητα 5.2.1 θα αναλύσουμε το *μοντέλο κίνησης*: θα μελετήσουμε την αρχική μορφή του μοντέλου κίνησης βασισμένου αποκλειστικά σε οπτική πληροφορία, και στην συνέχεια, θα δούμε την επέκταση που έγινε εισάγοντας επιπλέον μετρική πληροφορία από τους κωδικοποιητές του ρομπότ. Στην Ενότητα 5.2.2 αναφερόμαστε στο *μοντέλο παρατήρησης* και στις μορφές που μπορεί να πάρει.

### 5.2.1 Μοντέλο Κίνησης

Σε αντιστοιχία με όσα αναφέραμε στην Ενότητα 5.1, το μοντέλο κίνησης προσπαθεί σε κάθε στιγμή να υπολογίσει την ποσότητα  $p(x_t | u_t, x_{t-1})$ , όπου  $x_t, x_{t-1}$  είναι η τρέχουσα και η προηγούμενη εκτίμηση θέσης, αντίστοιχα, και  $u_t$  είναι είτε η εντολή κίνησης είτε η πληροφορία από την οδομετρία για το χρονικό διάστημα  $(t-1, t]$ .

### **5.2.1.1 Μοντέλο Κίνησης Βασισμένο στην Οπτική Οδομετρία**

Η προηγούμενη εργασία είχε υιοθετήσει ένα μοντέλο κίνησης για το ρομπότ το οποίο ήταν βασισμένο στην οπτική οδομετρία. Το γεγονός αυτό, το καθιστά ανεξάρτητο από ιδιοδεκτικούς αισθητήρες του ρομπότ ή άλλες εξωτερικές πηγές μετρικής πληροφορίας. Αυτό, έρχεται βέβαια σε συμφωνία με τον τοπολογικό χάρτη ο οποίος εν γένει δεν χρειάζεται μετρική πληροφορία για να κατασκευαστεί.

Η μέθοδος βασίζεται σε γενικές γραμμές, σε μια προτεινόμενη μεθοδολογία οπτικής οδομετρίας που παρουσιάζεται στο [26]. Το πώς ακριβώς δουλεύει περιγράφεται σε άλλες εργασίες, και δεν θα μας απασχολήσει ιδιαίτερα εδώ. Αυτό που αξίζει να αναφέρουμε είναι ότι μπορεί και δουλεύει σε πραγματικό χρόνο. Βέβαια, η μέθοδος αυτή επιβάλλει κάποιους περιορισμούς και παραδοχές, στους οποίους θα αναφερθούμε καθώς παίζουν σημαντικό ρόλο σε αυτή την εργασία.

#### **5.2.1.1.1 Παραδοχές-Περιορισμοί της Οπτικής Οδομετρίας**

Η 1<sup>η</sup> παραδοχή είναι ότι το ρομπότ κινείται σε επίπεδο χώρο, συνεπώς δεν υπάρχουν ράμπες, σκαλιά ή άλλες κλίσεις.

Η 2<sup>η</sup> παραδοχή, σχετίζεται με τις δυνατές κινήσεις του ρομπότ στον χώρο. Κάθε στιγμή, θεωρούμε ότι το ρομπότ εκτελεί μόνο μια από τέσσερις δυνατές κινήσεις: κίνηση προς τα εμπρός, σταμάτημα, περιστροφή δεξιά, ή περιστροφή αριστερά.

Η 3<sup>η</sup> παραδοχή λέει ότι το ρομπότ κινείται με σταθερή ταχύτητα. Τυχόν παραβίαση αυτής της συνθήκης μπορεί να οδηγήσει σε εσφαλμένη εκτίμηση θέσης, αλλά δεν θα δημιουργήσει ιδιαίτερα προβλήματα.

Η 4<sup>η</sup> και σημαντικότερη παραδοχή, είναι ότι ο χώρος στον οποίο κινείται το ρομπότ αποτελείται από διαδρόμους, σε ορθογώνια μεταξύ τους διάταξη. Έτσι το ρομπότ κινείται σε μόνο τέσσερις διαφορετικές διευθύνσεις με βήμα  $90^\circ$ . Επίσης, το ρομπότ πρέπει να πραγματοποιήσει μια στάση μεταξύ δύο διαδοχικών περιστροφών. Η παραδοχή αυτή, και μεν είναι δεσμευτική, αλλά στο περιβάλλον που δοκιμάσαμε τον αλγόριθμο, δεν δημιουργεί κανένα πρόβλημα.

#### **5.2.1.1.2 Εισάγοντας Εικονική Μετρική Πληροφορία για το Μοντέλο Κίνησης**

Είναι σαφές ότι με μια μονοσκοπική κάμερα, οποιαδήποτε μέθοδος οπτικής οδομετρίας δεν θα μπορούσε ποτέ να υπολογίσει βάθος στην κίνηση και συνεπώς να μας δώσει πληροφορία για το πόσο μπροστά κινήθηκε το ρομπότ. Βεβαίως υπάρχουν αλγόριθμοι που επιτυγχάνουν κάτι τέτοιο, όμως βασίζονται σε τουλάχιστον δύο κάμερες, ή στην παρακολούθηση κάποιων χαρακτηριστικών από καρέ σε καρέ.

Εάν δεν θέλουμε να χρησιμοποιήσουμε ή δεν έχουμε περισσότερες από μια κάμερες, τότε πρέπει να καταφύγουμε σε άλλες μεθόδους. Άλλωστε κάτι τέτοιο είναι αναγκαίο αφού το μοντέλο κίνησης προϋποθέτει την ύπαρξη κάποιας μορφής μετρικής πληροφορίας. Διαφορετικά, η μόνη λύση θα ήταν να μην υπάρχει βήμα πρόβλεψης (και κατ'επέκταση μοντέλο κίνησης), αλλά ο αλγόριθμος του τοπολογικού SLAM απλά να συγκρίνει εικόνες (συνεπώς να υπάρχει μόνο το βήμα ανανέωσης). Αυτό θα μπορούσε να γίνει σε ένα αλγόριθμο αυτοεντοπισμού θέσης, όπως άλλωστε είδαμε, αλλά δεν θα αρκούσε για ένα ολοκληρωμένο αλγόριθμο SLAM.

Μια λύση στο πρόβλημα που περιγράψαμε, είναι να δώσουμε μια εικονική μετρική πληροφορία, την οποία το [26] περιγράφει ως κίνηση στο *χώρο αντίληψης* (*perceptual space*) αντί για τον *φυσικό* (*physical*) *χώρο*, όπως περίπου οι μέλισσες χρησιμοποιούν την οπτική ροή για να επιτύχουν *ολοκλήρωση τροχιάς* (*path integration*).

Αναλυτικότερα, η ιδέα είναι η εξής: έστω ότι έχουμε δύο διαδοχικά καρέ. Στο 1<sup>ο</sup> βήμα, για δύο γειτονικά καρέ αθροίζουμε κατά στήλες την φωτεινότητα των pixel και κανονικοποιούμε, δημιουργώντας δύο πίνακες αθροισμάτων (ένα για κάθε καρέ). Εν συνεχεία βρίσκουμε το πόσο πρέπει να ολισθήσουμε τον ένα πίνακα, προκειμένου να ελαχιστοποιήσουμε την απόλυτη διαφορά τους. Με δεδομένη αυτή την ολίσθηση (θεωρώντας δηλαδή ότι έχουμε αποσυνδέσει την επίδραση της περιστροφής), βρίσκουμε το άθροισμα των διαφορών των δύο πινάκων και κανονικοποιούμε το αποτέλεσμα. Το αποτέλεσμα είναι μια τιμή εικονικής ταχύτητας. Όπως φαίνεται από τα αποτελέσματα της εργασίας, η μέθοδος αυτή οδηγεί στην δημιουργία πολύ καλών χαρτών.

Ένας άλλος, απλούστερος τρόπος να γίνει κάτι τέτοιο, είναι να μετράμε τα καρέ στα οποία παρατηρούμε κίνηση προς τα εμπρός. Όσο περισσότερα καρέ συμμετείχαν στην μεταφορική κίνηση, τόσο περισσότερο θεωρούμε ότι κινήθηκε το ρομπότ. Αυτή η απλή ιδέα είναι αρκετά αποτελεσματική στην δημιουργία της μετρικής ψευδαίσθησης και αυτή ακριβώς χρησιμοποιήθηκε από την εργασία του Γ. Καλπάκα, «Εκτίμηση Θέσης και Τοπολογική Χαρτογράφηση σε Αυτόνομο Κινούμενο Ρομποτικό Όχημα με βάση Οπτική Πληροφορία».

Τελικά λοιπόν και χρησιμοποιώντας τις παραδοχές της Ενότητας 5.2.1.1.1, το μοντέλο κίνησης  $p(x_i, y_i, \theta_i | u_i, x_{i-1})$ , όπου με  $x_i$  και  $x_{i-1}$  συμβολίζονται οι εκτιμήσεις θέσης του ρομποτικού οχήματος τις χρονικές στιγμές  $t$  και  $t-1$  αντίστοιχα και το  $u_i$  εκφράζει την εκτελούμενη κίνηση για το διάστημα  $(t-1, t]$ , είναι το εξής:

$$p(x_i, y_i, \theta_i | x_o, y_o, \theta_o) = \begin{cases} 0 & : \theta_i \neq \theta_o \\ \frac{1}{2\pi\sigma^2} \exp\left(-\frac{(x_i - x_o)^2 + (y_i - y_o)^2}{2\sigma^2}\right) & : \theta_i = \theta_o \end{cases} \quad (54)$$

όπου:

- $x_o, y_o, \theta_o \rightarrow$  Συντεταγμένες και Προσανατολισμός της εκτιμώμενης διάταξης του ρομπότ
- $x_i, y_i, \theta_i \rightarrow$  Συντεταγμένες και Προσανατολισμός του  $i$ -οστού κόμβου του χάρτη
- $\sigma \rightarrow$  Τυπική απόκλιση της κατανομής (ίδια και για τις δύο διαστάσεις)

Η σχέση (54) υλοποιεί το βήμα πρόβλεψης του Διακριτού Φίλτρου Bayes, δηλαδή σε κάθε βήμα και για κάθε κόμβο του χάρτη, αναθέτει μια εκτίμηση για την πιθανότητα να βρίσκεται το ρομπότ στον κόμβο αυτό, με δεδομένη την θέση και τον προσανατολισμό του ρομπότ.

Όπως φαίνεται από την (54), το ρομποτικό όχημα έχει 0% πιθανότητα να βρεθεί σε κάποιον κόμβο με διαφορετικό προσανατολισμό από τον τρέχοντα. Η πιθανότητα να βρεθεί σε κόμβο με τον τρέχοντα προσανατολισμό είναι αντιστρόφως ανάλογη της απόστασης του κόμβου από την εκτιμώμενη θέση του ρομπότ και ορίζεται ως η τιμή μιας σταθερού εύρους δισδιάστασης κατανομής Gauss με κέντρο την εκτιμώμενη θέση του ρομπότ.

### 5.2.1.2 *Επέκταση του Μοντέλου Κίνησης με Μετρική Πληροφορία*

Στην Ενότητα 5.2.1.1 είδαμε την μορφή του μοντέλου κίνησης βασισμένου αποκλειστικά σε οπτική πληροφορία. Στην παρούσα εργασία επεκτείνουμε το μοντέλο αυτό χρησιμοποιώντας την μετρική πληροφορία που μας διέθεταν οι κωδικοποιητές στους τροχούς του ρομπότ, εγκαταλείποντας φυσικά την εικονική μετρική πληροφορία της οπτικής οδομετρίας και του προηγούμενου μοντέλου κίνησης. Βέβαια, διατηρήσαμε τις ίδιες παραδοχές της οπτικής οδομετρίας, με κυριότερη αυτή της ύπαρξης μόνο τεσσάρων κατευθύνσεων στην κίνηση του ρομπότ. Στόχος μας είναι η παραγωγή ενός μοντέλου κίνησης, το οποίο σεβόμενο τις παραπάνω παραδοχές, θα προβλέπει την νέα θέση του ρομπότ γνωρίζοντας την προηγούμενη θέση και στοιχεία οδομετρίας, ενώ παράλληλα θα διατηρεί και μια αβεβαιότητα στην κίνηση αυτή.

Σε κάθε βήμα έχουμε ως δεδομένα την θέση του ρομπότ  $x_{t-1}$  και δεδομένα οδομετρίας  $(\Delta_x^{odo}, \Delta_y^{odo}, \Delta_\varphi^{odo})$ . Συνεπώς, η νέα θέση του ρομπότ υπολογίζεται με τη βοήθεια της προηγούμενης θέσης και της οδομετρίας, θεωρώντας πάντοτε τέσσερις προσανατολισμούς κίνησης. Δηλαδή πάντοτε το ρομπότ κινείται ή στον άξονα  $x$  ή στον άξονα  $y$ , αλλά ποτέ και στους δύο ταυτόχρονα. Ας εξηγήσουμε λίγο πιο μαθηματικά αυτό που λέμε:

Γενικά ένα μοντέλο κίνησης βασισμένο στην οδομετρία είναι το εξής:

$$\begin{pmatrix} x' \\ y' \\ \varphi' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \varphi \end{pmatrix} + \begin{pmatrix} \cos(\varphi + \frac{\Delta_\varphi^{odo}}{2}) & -\sin(\varphi + \frac{\Delta_\varphi^{odo}}{2}) & 0 \\ \sin(\varphi + \frac{\Delta_\varphi^{odo}}{2}) & \cos(\varphi + \frac{\Delta_\varphi^{odo}}{2}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta_x^{odo} \\ \Delta_y^{odo} \\ \Delta_\varphi^{odo} \end{pmatrix} \quad (55)$$

Με βάση τις παραδοχές που έχουμε κάνει, προσαρμόζουμε το παραπάνω μοντέλο στο πραγματικό μοντέλο κίνησης του ρομπότ, και έτσι θα έχουμε:

$$\begin{pmatrix} x' \\ y' \\ \varphi' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \varphi \end{pmatrix} + \begin{pmatrix} \cos(\varphi + \frac{\Delta_\varphi^{odo}}{2}) & 0 & 0 \\ 0 & \sin(\varphi + \frac{\Delta_\varphi^{odo}}{2}) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta_x^{odo} \\ \Delta_y^{odo} \\ \Delta_\varphi^{odo} \end{pmatrix} \quad (56)$$

Αφού σε κάθε χρονική στιγμή έχουμε κίνηση μόνο στον ένα άξονα, αυτό σημαίνει ότι μόνο ένα εκ των  $\Delta_x^{odo}, \Delta_y^{odo}$  θα είναι διάφορο από μηδέν και ότι η κίνηση πάνω στους δύο άξονες είναι ασυσχέτιστη. Τέλος, με βάση την παραδοχή των τεσσάρων κατευθύνσεων θα έχουμε  $\Delta_\varphi^{odo} = 0$ , και συνεπώς θα ισχύει:  $\varphi = 0^\circ$  ή  $90^\circ$  ή  $180^\circ$  ή  $270^\circ$ . Για  $\varphi = 0^\circ$  ή  $180^\circ$  έχουμε κίνηση πάνω στον άξονα x, ενώ διαφορετικά έχουμε κίνηση στον άξονα y.

$$\begin{pmatrix} x' \\ y' \\ \varphi' \end{pmatrix} = \begin{pmatrix} x \\ y \\ \varphi \end{pmatrix} + \begin{pmatrix} \cos(\varphi) & 0 & 0 \\ 0 & \sin(\varphi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \Delta_x^{odo} \\ \Delta_y^{odo} \\ 0 \end{pmatrix} \quad (57)$$

Το παραπάνω μοντέλο της (57) υπολογίζει την νέα θέση του ρομπότ με βάση την προηγούμενη και την οδομετρία. Μένει ακόμα να εισάγουμε και την αβεβαιότητα στην κίνηση αυτή. Για να το κάνουμε αυτό χρησιμοποιούμε ένα Γκαουσιανό Πιθανοτικό Μοντέλο. Στόχος μας σε κάθε χρονική στιγμή, είναι να βρούμε την νέα Γκαουσιανή κατανομή αβεβαιότητας της θέσης του ρομπότ, γνωρίζοντας την προηγούμενη. Η ιδέα είναι απλή:

Η Γκαουσιανή χρειάζεται δύο μεγέθη για να προσδιοριστεί πλήρως. Πρώτον, την μέση τιμή η οποία είναι απλώς το αποτέλεσμα της (57) για την τρέχουσα θέση του ρομπότ. Δεύτερον, τη τυπική απόκλιση  $\sigma_x, \sigma_y$ . Για να τη βρούμε, θα ξεκινήσουμε από το γενικότερο μοντέλο της (55) και στην συνέχεια θα καταλήξουμε στο ζητούμενο.

Έστω λοιπόν  $\Sigma$  ο διαγώνιος πίνακας που στη διαγώνιό του έχει τις τρεις αποκλίσεις των παραμέτρων της οδομετρίας,  $\sigma_{\Delta_x^{odo}}, \sigma_{\Delta_y^{odo}}, \sigma_{\Delta_\varphi^{odo}}$ . Θα ισχύει:

$$\Sigma = \begin{bmatrix} \sigma_{\Delta_x^{odo}} & 0 & 0 \\ 0 & \sigma_{\Delta_y^{odo}} & 0 \\ 0 & 0 & \sigma_{\Delta_\varphi^{odo}} \end{bmatrix} \quad (58)$$

$$\begin{aligned}\sigma_{\Delta_x^{odo}} &= \Delta\sigma_{\Delta_y^{odo}} = a_1 \sqrt{(\Delta_x^{odo})^2 + (\Delta_y^{odo})^2} + a_2 |\Delta_\varphi^{odo}| \\ \sigma_{\Delta_\varphi^{odo}} &= a_3 \sqrt{(\Delta_x^{odo})^2 + (\Delta_y^{odo})^2} + a_4 |\Delta_\varphi^{odo}| \end{aligned} \quad (59)$$

Τελικά θα έχουμε για τον πίνακα  $C$  των αποκλίσεων της Γκαουσιανής κατανομής της νέας θέσης του ρομπότ:

$$\Delta C = J \Sigma J^T \quad (60)$$

όπου  $J$  είναι η Ιακωβιανή της (5). Η (60) μας λέει ότι η μεταβολή της αβεβαιότητας στη νέα θέση του ρομπότ, εξαρτάται από την αβεβαιότητα στην οδομετρία που εκφράζει ο πίνακας  $\Sigma$  και από το κινηματικό μοντέλο, που εκφράζει η Ιακωβιανή.

Στην δική μας περίπτωση τα πράγματα είναι αρκετά απλοποιημένα. Αρχικά, αγνοούμε οποιοδήποτε σφάλμα στον προσανατολισμό καθώς στην πράξη, όπως αναφέραμε παραπάνω, δεν υπάρχει. Συνεπώς  $\sigma_{\Delta_\varphi^{odo}} = 0$ . Επίσης στον υπολογισμό της (59), αγνοούμε τον όρο  $|\Delta_\varphi^{odo}|$  καθώς η κβάντιση των τεσσάρων προσανατολισμών αφαιρεί οποιαδήποτε επίδραση του όρου στο σφάλμα θέσης. Τέλος, η υπόθεση ότι το ρομπότ κινείται είτε στον άξονα  $x$  είτε στον  $y$ , σημαίνει ότι η αβεβαιότητα για την θέση αυξάνεται σε κάθε χρονική στιγμή είτε για τον άξονα  $x$  είτε για τον άξονα  $y$ , αλλά ποτέ και για τους δύο ταυτόχρονα.

Τα παραπάνω σημαίνουν ότι η (59), τελικά καταλήγει στην ακόλουθη μορφή:

$$\begin{aligned}\sigma_{\Delta_x^{odo}} &= a_1 |\Delta_x^{odo}| \\ \sigma_{\Delta_y^{odo}} &= a_1 |\Delta_y^{odo}| \\ \sigma_{\Delta_\varphi^{odo}} &= 0 \end{aligned} \quad (61)$$

Ακόμα, με βάση την (57), η Ιακωβιανή  $J$  θα έχει την εξής μορφή:

$$J = \begin{bmatrix} \cos(\varphi) & 0 & 0 \\ 0 & \sin(\varphi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (62)$$

Συνεπώς με βάση τις (58), (61), (62) μπορούμε να υπολογίσουμε την (60).

Έτσι, η αβεβαιότητα θέσης του ρομπότ εξελίσσεται με βάση την ακόλουθη σχέση:

$$\begin{aligned}\sigma_x' &= \sigma_x + \Delta C_{(1,1)} \\ \sigma_y' &= \sigma_y + \Delta C_{(2,2)}\end{aligned}\quad (63)$$

όπου  $\Delta C_{(1,1)}$  είναι το 1<sup>ο</sup> διαγώνιο στοιχείο και το  $\Delta C_{(2,2)}$  το 2<sup>ο</sup> διαγώνιο στοιχείο του πίνακα  $\Delta C$ .

Έτσι λοιπόν με τις (57) και (63) μπορούμε να βρούμε την νέα θέση του ρομπότ και την νέα αβεβαιότητα αυτής της θέσης, αντίστοιχα. Δηλαδή, οι σχέσεις αυτές υπολογίζουν την πιθανότητα  $p(x_i | u_i, x_{i-1})$ .

### 5.2.1.3 Βήμα Πρόβλεψης – Επέκταση με Μετρική Πληροφορία

Έχοντας εξηγήσει το μοντέλο κίνησης, που περιγράφει το πώς κινείται το ρομπότ στον χώρο και εισάγει μια αβεβαιότητα στην κίνηση αυτή, ερχόμαστε τώρα να υλοποιήσουμε το βήμα πρόβλεψης του Διακριτού Φίλτρου Bayes.

Το βήμα αυτό, βρίσκει την πιθανότητα μετάβασης σε κάποιον κόμβο του χάρτη, με θέση  $(x_i, y_i, \theta_i)$ , δεδομένου ότι βρισκόμαστε σε μια θέση  $(x_0, y_0, \theta_0)$ , με άλλα λόγια υπολογίζει την κατανομή  $p(x_i | u_i, x_{i-1})$ . Η λογική λύση θα ήταν να χρησιμοποιήσουμε και εδώ μια διδιάστατη Γκαουσιανή κατανομή (όπως κάναμε και για την μοντελοποίηση της αβεβαιότητας της θέσης του ρομπότ).

Η Γκαουσιανή αυτή κατανομή ουσιαστικά περιγράφει τον χώρο στον οποίο αναζητούμε την πιθανή θέση του ρομπότ. Προφανώς, όσο μεγαλύτερη είναι η αβεβαιότητά μας για την θέση του ρομπότ, τόσο μεγαλύτερος είναι ο χώρος αναζήτησης (και άρα τόσο περισσότερους κόμβους του χάρτη εξετάζουμε). Το μοντέλο που χρησιμοποιούμε παρουσιάζει μηδενική ανοχή ως προς τον προσανατολισμό: με άλλα λόγια, αναζητούμε κόμβους που έχουν τον ίδιο προσανατολισμό με τον τρέχοντα προσανατολισμό του ρομπότ. Επομένως, κόμβοι με διαφορετικό προσανατολισμό έχουν μηδενική πιθανότητα και απορρίπτονται. Έτσι λοιπόν, καταλήγουμε στην ακόλουθη μορφή της κατανομής πιθανότητας:

$$p(x_i, y_i, \theta_i | x_o, y_o, \theta_o) = \begin{cases} 0 & : \theta_i \neq \theta_o \\ \frac{1}{2\pi k^2 \sigma_x \sigma_y} \exp\left(-\frac{(x_i - x_o)^2 (y_i - y_o)^2}{2k^2 \sigma_x \sigma_y}\right) & : \theta_i = \theta_o \end{cases}\quad (64)$$



,όπου:

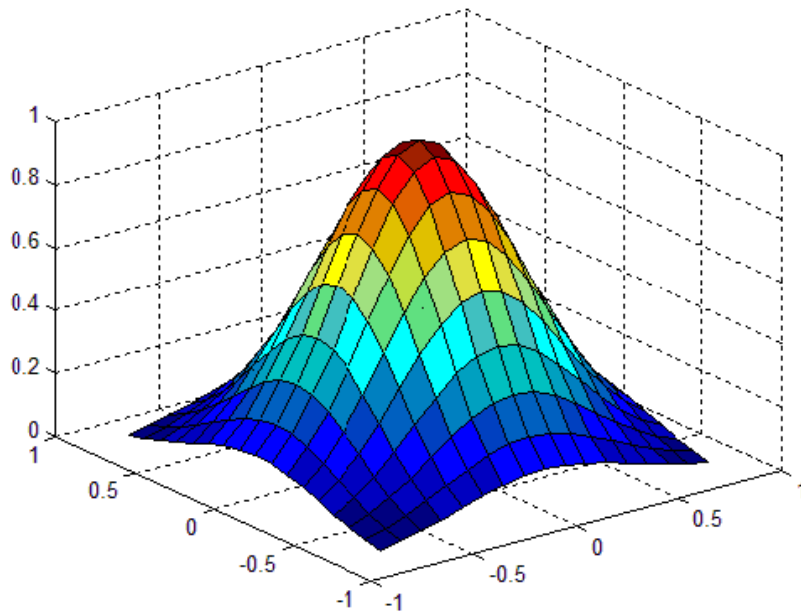
$x_o, y_o, \theta_o \rightarrow$  Συντεταγμένες και Προσανατολισμός της εκτιμώμενης διάταξης του ρομπότ

$x_i, y_i, \theta_i \rightarrow$  Συντεταγμένες και Προσανατολισμός του  $i$ -οστού κόμβου του χάρτη

$\sigma_x, \sigma_y \rightarrow$  Τυπική απόκλιση της θέσης του ρομπότ (αβεβαιότητα θέσης)

$k \rightarrow$  Παράμετρος που καθορίζει το εύρος της αναζήτησης ( $k = 5$ )

Η διαφορά των δύο μοντέλων κίνησης είναι σαφής. Το μοντέλο κίνησης που βασίζεται στην οπτική οδομετρία, δεν εισάγει καμία αβεβαιότητα ως προς την θέση του ρομπότ και αυτό φαίνεται και στην σχέση (54) όπου ουσιαστικά η Gaussian του βήματος πρόβλεψης είναι σταθερή σε εύρος. Αντίθετα, το μοντέλο κίνησης που εμπεριέχει μετρική αβεβαιότητα, εισάγει μια αβεβαιότητα στην θέση, όπως ορίζεται από τις σχέσεις (57) και (63), που



**Σχημα 17. Η συνάρτηση κατανομής πιθανότητας του μοντέλου κίνησης. Πηγή: Γ. Καλπάκας, «Εκτίμηση Θέσης και Τοπολογική Χαρτογράφηση σε Αυτόνομο Κινούμενο Ρομποτικό Όχημα με βάση Οπτική Πληροφορία.**

αντιπροσωπεύουν την μέση τιμή και την απόκλιση της θέσης του ρομπότ. Επιπλέον το βήμα πρόβλεψης της (64), μεταβάλλει την Gaussian κατανομή αναζήτησης στον χώρο ανάλογα με την αβεβαιότητα της θέσης του ρομπότ. Αυτό, όπως θα φανεί στην συνέχεια είναι πολύ σημαντικό χαρακτηριστικό.

Είναι ιδιαίτερα σημαντικό να αντιληφθούμε ότι, με δεδομένη την φύση των κόμβων του χάρτη, οι οποίοι αντιπροσωπεύουν περισσότερο περιοχές του χώρου και λιγότερο συγκεκριμένα σημεία του, όταν λέμε ότι βρισκόμαστε στον  $i$ -οστό κόμβο εννοούμε ότι βρισκόμαστε στην περιοχή του  $i$ -οστού κόμβου και όχι ακριβώς πάνω στις συντεταγμένες του.

## 5.2.2 Μοντέλο Παρατήρησης

Το μοντέλο παρατήρησης υπολογίζει την δεσμευμένη πιθανότητα  $p(z_t | x_k)$ , δηλαδή την πιθανότητα να παρατηρήσουμε τα χαρακτηριστικά  $z_t$  του περιβάλλοντος, όταν βρισκόμαστε στον κόμβο  $k$ .

Όπως είπαμε, ο βασικός αλγόριθμος στην εργασία του Γ. Καλπάκα «Εκτίμηση Θέσης και Τοπολογική Χαρτογράφηση σε Αυτόνομο Κινούμενο Ρομποτικό Όχημα με βάση Οπτική Πληροφορία», χρησιμοποιούσε τα ιστογράμματα χρωματικού τόνου και κορεσμού για τον υπολογισμό της πιθανότητας  $p(z_t | x_k)$ . Δηλαδή, στην περίπτωση αυτή ισχύει:

$$p(z_t | x_k) = p^{histogram}(z_t | x_k) \quad (65)$$

Το μοντέλο αυτό επεκτείνεται στην παρούσα εργασία με την προσθήκη των χαρακτηριστικών SURF, όπως περιγράφηκε στην Ενότητα 3.1.6.4.. Πραγματοποιήσαμε μια σειρά από πειράματα για να συγκρίνουμε τις επιδόσεις τόσο των ιστογραμμάτων, όσο και των χαρακτηριστικών SURF.

Πειραματιστήκαμε με διάφορες μορφές της συνάρτησης  $p(z_t | x_k)$ , είτε χρησιμοποιώντας μόνο τα ιστογράμματα εικόνων (βασίζομενοι στην Απόκλιση Jeffrey), είτε μόνο τα χαρακτηριστικά SURF, είτε και τα δύο. Στην τελευταία περίπτωση, χρησιμοποιήσαμε την ακόλουθη μορφή για το μοντέλο παρατήρησης:

$$p(z_t | x_k) = \begin{cases} \frac{p^{histogram}(z_t | x_k) + p^{SURF}(z_t | x_k)}{2} \\ \text{ή} \\ \max(p^{histogram}(z_t | x_k), p^{SURF}(z_t | x_k)) \end{cases} \quad (66)$$

,όπου η συνάρτηση  $p^{SURF}$  περιγράφεται στην ενότητα 4.1.2.2.

Σε ό,τι αφορά την  $p^{histogram}$ , είδαμε στην ενότητα 4.1.2.1 το πώς εφαρμόζουμε το κριτήριο Jeffrey για να συγκρίνουμε δύο ιστογράμματα και να βγάλουμε ένα μέτρο ομοιότητας. Με βάση τα όσα είπαμε στην ενότητα 5.1.1 για τον κατακερματισμό εικόνων, θα έχουμε ότι για τον υπολογισμό της τελικής  $p^{histogram}$  χρειάζονται δύο βήματα:

1. Αρχικά, εξάγεται ένα μέτρο ομοιότητας (Απόκλιση Jeffrey) για κάθε ένα από τα 16 ζεύγη ομόλογων τμημάτων των δύο υπό σύγκριση εικόνων. Ομόλογα ονομάζουμε τα τμήματα που βρίσκονται στην ίδια θέση. Στην πράξη λαμβάνουμε δύο μέτρα ομοιότητας για κάθε ζεύγος τμημάτων, για τις συνιστώσες Χρωματικός Τόνος (Hue) και Κορεσμός (Saturation), οπότε το συνολικό μέτρο ομοιότητας των δύο ομόλογων τμημάτων είναι ο μέσος όρος των μέτρων των δύο συνιστωσών.
2. Στη συνέχεια υπολογίζεται ο μέσος όρος των μέτρων κάθε ζεύγους ομόλογων τμημάτων των δύο εικόνων, από τον οποίο προκύπτει το συνολικό μέτρο ομοιότητας των δύο εικόνων.

Όπως θα φανεί και από τα πειράματα που διεξήχθησαν, η επίδοση του μοντέλου παρατήρησης με χρήση των χαρακτηριστικών SURF, ήταν πολύ καλύτερη από αυτή των ιστογραμμάτων χρωματικού τόνου και κορεσμού.

### 5.2.2.1 Βήμα Ανανέωσης

Η σχέση (64) μας δίνει μια πρώτη εκτίμηση της πιθανότητας το ρομπότ να βρίσκεται στον  $k$ -οστό κόμβο. Η εκτίμηση αυτή, βασίζεται μόνο σε δεδομένα οδομετρίας. Το βήμα ανανέωσης βελτιώνει αυτή την εκτίμηση, συνυπολογίζοντας και την οπτική πληροφορία. Σε συμφωνία με την Ενότητα 4.2.2, η μορφή του βήματος ανανέωσης θα είναι η εξής:

$$p(x_{i,t} | z_t, u_t, x_{t-1}) = \underbrace{p(z_t | x_i)}_{\text{μοντέλο παρατήρησης}} \overbrace{p(x_{i,t} | z_{t-1}, u_t, x_{t-1})}^{\text{βήμα πρόβλεψης}} \quad (67)$$

Ο  $1^{\text{ος}}$  όρος είναι η σχέση(66), ενώ ο  $2^{\text{ος}}$  όρος η σχέση (64).

Συνολικά, η σχέση (67) μας δίνει την πιθανότητα να βρίσκεται το ρομπότ στον  $i$ -οστό κόμβο την χρονική στιγμή  $t$ , δοσμένων των δεδομένων της οδομετρίας  $u_t$  και της οπτικής πληροφορίας  $z_t$ , και γνωρίζοντας φυσικά την προηγούμενη θέση του  $x_{t-1}$

## 5.3 Αναγνώριση Βρόχου και Ανανέωση Χάρτη

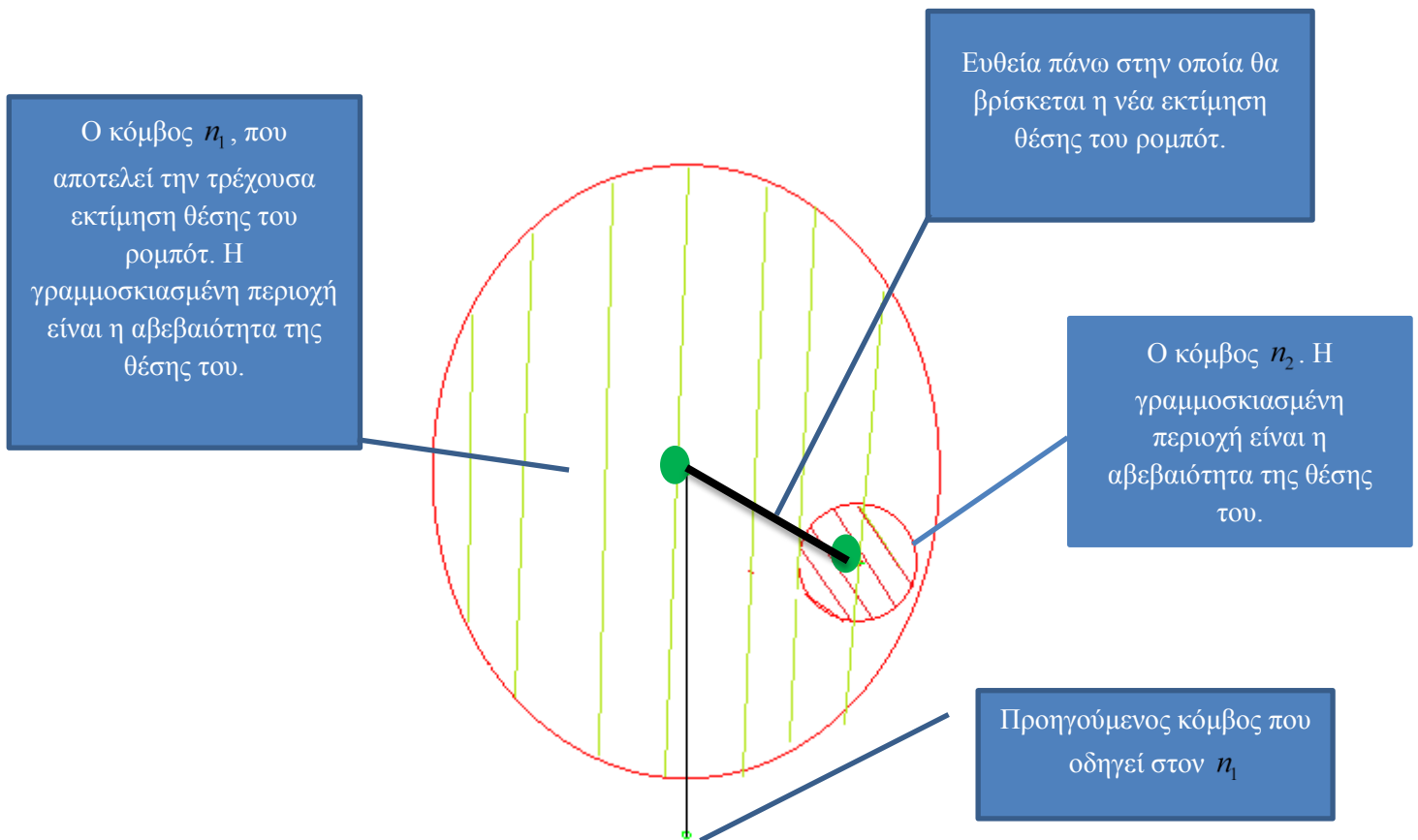
Η ολοκλήρωση ενός βρόχου και η αναγνώρισή του από τον αλγόριθμο του SLAM είναι καίριας σημασίας για την ορθή λειτουργία του. Στον αλγόριθμό μας, εξαιτίας και της απουσίας οποιασδήποτε άλλης πληροφορίας, η αναγνώριση ενός βρόχου, είναι ο μόνος τρόπος να διορθώσουμε τυχόν ανακρίβειες του χάρτη καθώς και να μειώσουμε την αβεβαιότητα του σε ό,τι αφορά τις θέσεις των κόμβων και του ρομπότ.

Η ανάλυση που ακολουθεί αφορά την περίπτωση όπου αναγνωρίζουμε ένα βρόχο που ολοκληρώνουν δύο κόμβοι για 1<sup>η</sup> φορά (με άλλα λόγια, πριν από την αναγνώριση του βρόχου οι δύο αυτοί κόμβοι δεν είχαν καμία ακμή που να τους συνδέει).

Ας φανταστούμε ότι το ρομπότ κινείται στον χώρο. Σε κάθε βήμα του, η αβεβαιότητά μας για την πραγματική θέση του μεγαλώνει με βάση το μοντέλο κίνησης όπως αναλύθηκε στην Ενότητα 5.2.1.2. Έτσι, κάθε κόμβος που εισάγεται στον χάρτη χαρακτηρίζεται από μια αβεβαιότητα σε ό,τι αφορά την θέση του και η αβεβαιότητα αυτή είναι μεγαλύτερη από αυτή των προηγούμενων κόμβων. Ας υποθέσουμε ότι κάποια χρονική στιγμή, που ο αλγόριθμος έχει μια εκτιμώμενη θέση του ρομπότ, αντιλαμβάνεται ότι η τρέχουσα παρατήρηση (εικόνα) αντιστοιχεί σε προηγούμενη θέση (κόμβο). Είναι λογικό, ότι ο κόμβος αυτός, έχοντας προστεθεί στον χάρτη στο παρελθόν θα έχει μικρότερη αβεβαιότητα θέσης από την τρέχουσα του ρομπότ – κάτι τέτοιο θα συμβεί σίγουρα εάν κλείνουμε βρόχο για 1<sup>η</sup> φορά. Ας δούμε λίγο πιο μαθηματικά την ανάλυσή αυτή:

Έστω ότι ο κόμβος  $n_1$  αποτελεί την τρέχουσα εκτιμώμενη θέση του ρομπότ  $(\hat{x}, \hat{y})$  και έχει μια Γκαουσσιανή αβεβαιότητα βασισμένη στο μοντέλο κίνησης που χαρακτηρίζεται από τα εξής μεγέθη:  $(\hat{x}, \hat{y}, \hat{\sigma}_x, \hat{\sigma}_y)$  όπου  $\hat{x}, \hat{y}$ , η μέση τιμή της θέσης της Γκαουσσιανής και  $\hat{\sigma}_x, \hat{\sigma}_y$  η τυπική απόκλιση της.

Ακόμα, έστω ότι η τρέχουσα παρατήρηση, μας οδηγεί στην αναγνώριση ενός βρόχου, δηλαδή ο αλγόριθμος βλέπει ότι το ρομπότ βρίσκεται στην περιοχή ενός παλαιότερου κόμβου, έστω του  $n_2$ , στη θέση  $(x, y)$ , ο οποίος θα χαρακτηρίζεται με την σειρά του από αντίστοιχα μεγέθη αβεβαιότητας:  $(x, y, \sigma_x, \sigma_y)$ .



Σημια 18. Απεικόνιση της ανακατάταξης των κόμβων, στην αναγνώριση βρόχου.

Κατ'αναλογία με το φίλτρο Κάλμαν, η κατάσταση αυτή, όπου έχουμε μια εκτίμηση θέσης με βεβαιότητα  $\hat{\sigma}_x, \hat{\sigma}_y$  και εν συνεχεία μια καινούργια παρατήρηση με αβεβαιότητα  $\sigma_x, \sigma_y$ , μπορεί να μειώσει την συνολική αβεβαιότητα και να βελτιώσει την εκτίμησή μας.

Σε μια τέτοια περίπτωση λοιπόν, αρχικά υπολογίζουμε τα κέρδη του φίλτρου Κάλμαν για τους άξονες  $x, y$ .

$$\begin{aligned}
 gain_x &= \frac{\hat{\sigma}_x}{\hat{\sigma}_x + \sigma_x} \\
 gain_y &= \frac{\hat{\sigma}_y}{\hat{\sigma}_y + \sigma_y}
 \end{aligned}
 \tag{68}$$

Η σχέσεις αυτές μας λένε ότι αν η ακρίβεια του νέου κόμβου είναι πολύ μεγάλη σε σχέση με την ακρίβεια της τρέχουσας εκτιμώμενης θέσης, δηλαδή  $\sigma_x \ll \hat{\sigma}_x$  και/ή  $\sigma_y \ll \hat{\sigma}_y$  (η αβεβαιότητα του νέου κόμβου είναι πολύ μικρή σε σχέση με την αβεβαιότητα της τρέχουσας εκτιμώμενης θέσης), τότε αυτά τα κέρδη παίρνουν τιμές πολύ κοντά στην μονάδα.

Ακολούθως, γίνεται η διόρθωση της αβεβαιότητας της θέσης δηλαδή:

$$\begin{aligned}\widehat{\sigma}_x' &= \widehat{\sigma}_x - gain_x \cdot \widehat{\sigma}_x \\ \widehat{\sigma}_y' &= \widehat{\sigma}_y - gain_y \cdot \widehat{\sigma}_y\end{aligned}\quad (69)$$

Τώρα μένουν δύο ζητήματα:

1. Το πρώτο σχετίζεται με τη διόρθωση της θέσης των κόμβων που συμμετέχουν σε αυτό τον βρόχο. Ας ονομάσουμε, χρησιμοποιώντας τους προηγούμενους συμβολισμούς,  $n_1$  τον κόμβο που αντιστοιχεί στην τελευταία εκτίμηση θέσης, ακριβώς πριν την αναγνώριση του βρόχου και  $n_2$  τον κόμβο που επιστρέφουμε κατά την ολοκλήρωση του βρόχου. Έχοντας υπολογίσει πιο πριν τα κέρδη  $gain_x, gain_y$ , μπορούμε να υπολογίσουμε μια νέα βελτιωμένη εκτίμηση θέσης του ρομπότ, η οποία θα βρίσκεται πάνω στην ευθεία που ενώνει τους κόμβους  $n_1, n_2$ . Όσο πιο κοντά στην μονάδα είναι τα κέρδη που υπολογίστηκαν, τόσο πιο κοντά στον κόμβο  $n_2$  θα είναι η νέα εκτίμηση θέσης, αφού στην περίπτωση αυτή, η αβεβαιότητα του  $n_2$  είναι πολύ μικρή συγκρινόμενη με του  $n_1$ .

Αρχικά υπολογίζουμε τις διαφορές στις θέσεις των δύο κόμβων, πάνω στους δύο άξονες κίνησης του ρομπότ:

$$\begin{aligned}Dx &= x - \widehat{x} \\ Dy &= y - \widehat{y}\end{aligned}\quad (70)$$

Εν συνεχεία διορθώνουμε τις θέσεις των δύο κόμβων με βάση τις ακόλουθες σχέσεις:

$$\begin{cases} \widehat{x}' = \widehat{x} + Dx \cdot gain_x \\ x' = x - (1 - gain_x) \cdot Dx \\ \widehat{y}' = \widehat{y} + Dy \cdot gain_y \\ y' = y - (1 - gain_y) \cdot Dy \end{cases}\quad (71)$$

Έτσι οι δύο κόμβοι συνδυάζονται σε ένα νέο κόμβο, με καινούργια θέση πάνω στην ευθεία που συνδέει τους κόμβους  $n_1, n_2$ . Όσο μικρότερη η αβεβαιότητα του  $n_2$  τόσο πιο κοντά του θα είναι η νέα εκτιμώμενη θέση του ρομπότ.

Τέλος στο βήμα αυτό, πρέπει να ανανεώσουμε την θέση και άλλων κόμβων του χάρτη. Συγκεκριμένα, στην υλοποίησή μας επεξεργαζόμαστε μόνο τους κόμβους που συμμετέχουν στον βρόχο, συνεπώς κάθε ολοκλήρωση βρόχου εξετάζεται εντελώς ανεξάρτητα από άλλες προηγούμενες. Οι κόμβοι αυτοί αλλάζουν θέση προκειμένου να διατηρηθεί ίδια η βασική δομή του χάρτη.

2. Τέλος, πρέπει είναι να διορθώσουμε τις αβεβαιότητες της θέσης των κόμβων που συμμετείχαν στον βρόχο. Παραπάνω είδαμε πώς υπολογίζουμε την νέα αβεβαιότητα των  $n_1, n_2$ . Η μείωση της αβεβαιότητας του κόμβου  $n_1$  είναι λογικό να οδηγήσει σε μείωση της αβεβαιότητας και των κόμβων που προηγούνται αυτού. Αυτή η μείωση γίνεται με εφαρμογή του μοντέλου κίνησης του ρομπότ, προς τα πίσω, δηλαδή ξεκινώντας από τον  $n_1$  και πηγαίνοντας προς τα πίσω στο μονοπάτι του ρομπότ. Η εφαρμογή του αναστροφου κινηματικού μοντέλου σταματά όταν δούμε ότι πλέον δεν μας συμφέρει να την συνεχίσουμε, δηλαδή όταν δούμε ότι η αβεβαιότητα των κόμβων βάση του μοντέλου κίνησης είναι μεγαλύτερη από την αρχική τους (πριν την αναγνώριση του βρόχου).

## 5.4 Συνοπτική Παρουσίαση του Αλγορίθμου SLAM

Αφού παρουσιάσαμε τα σημαντικότερα κομμάτια της μεθόδου που αναπτύξαμε, κρίναμε σκόπιμο να παρουσιάσουμε συνοπτικά τα βήματα του αλγορίθμου εντοπισμού βρόχου και τοπολογικής χαρτογράφησης και αυτοεντοπισμού θέσης, ο οποίος προτείνεται στην παρούσα εργασία:

1. Αρχικοποίηση: εισαγωγή αρχικού κόμβου, αρχικοποίηση παραμέτρων.
2. Λήψη πληροφορίας:
  - a. Επεξεργασία εικόνας: Ανάλογα με το μοντέλο παρατήρησης, εξαγωγή χαρακτηριστικών SURF από την εικόνα και/ή ιστογραμμάτων χρωματικού τόνου και κορεσμού.
  - b. Επεξεργασία μετρικής πληροφορίας: εφαρμογή του μοντέλου κίνησης για τον υπολογισμό της νέας θέσης και αβεβαιότητας θέσης του ρομπότ.
3. Εφαρμογή βημάτων πρόβλεψης και ανανέωσης για όλους τους κόμβους του χάρτη. Υπάρχουν τρία ενδεχόμενα:
  - a. Ο τρέχων κόμβος αναδουκνείται νικητής, δηλαδή πιθανότητα να φιλοξενεί το ρομπότ μεγαλύτερη ή ίση από 0,6. Πέραν από την ενημέρωση για το μονοπάτι που ακολουθεί το ρομπότ, δεν γίνεται κάποια άλλη ενέργεια.
  - b. Κανένας κόμβος δεν ξεπερνά το ελάχιστο κατώφλι πιθανότητας (0.49). Εισαγωγή νέου κόμβου στην τρέχουσα θέση του ρομπότ και ενημέρωση του χάρτη και του Πίνακα Γειτνίασης.
  - c. Αναγνώριση βρόχου. Γίνονται όλες οι ενέργειες που περιγράψαμε στην Ενότητα 5.3 για το πώς ανανεώνονται οι κόμβοι και ο χάρτης.

Επιστροφή στο βήμα 2.

## 6

# Πειραματική Αξιολόγηση

Στο κεφάλαιο αυτό, θα εξετάσουμε την πειραματική εφαρμογή και αξιολόγηση των μεθοδολογιών που αναπτύχθηκαν στα προηγούμενα κεφάλαια, και ειδικά του αλγορίθμου υβριδικής τοπολογικής/μετρικής χαρτογράφησης και εντοπισμού θέσης που αναπτύχθηκε στην παρούσα εργασία, και χρησιμοποιεί μονοσκοπική οπτική και οδομετρική αισθητηριακή πληροφορία..

Ειδικότερα, στην Ενότητα 6.1 θα αναφερθούμε στις βασικές παραμέτρους αξιολόγησης του αλγορίθμου SLAM και στην Ενότητα 6.2 στο πώς αξιολογούμε τα αποτελέσματα του.

Ακόμα, στην Ενότητα 6.3 θα συγκρίνουμε τα αποτελέσματα του μοντέλου παρατήρησης χρησιμοποιώντας είτε μόνο ιστογράμματα, είτε μόνο χαρακτηριστικά SURF. Επίσης, θα δείξουμε πώς η επιλογή του κατωφλίου για το μοντέλο παρατήρησης των χαρακτηριστικών SURF επηρεάζει τη συμπεριφορά του αλγορίθμου.

Η Ενότητα 6.4 περιγράφει την διασύνδεση με το ρομπότ και την πειραματική διάταξη.

Ακολούθως, στην Ενότητα 6.5 θα επιχειρήσουμε να παρουσιάσουμε τα αποτελέσματα ενός ολοκληρωμένου πειράματος με αναγνώριση βρόχου, που έγινε χρησιμοποιώντας το ρομπότ Pioneer 3-DX του Εργαστηρίου Ρομποτικής της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Στην Ενότητα 6.6, θα δείξουμε το πώς η μετρική πληροφορία μπορεί να μας βοηθήσει και να βελτιώσει τον αλγόριθμό μας, σε σχέση με μεθόδους καθαρά τοπολογικές, δηλαδή χωρίς καμία ενσωμάτωση πραγματικής μετρικής πληροφορίας.

Επίσης, στην Ενότητα 6.7, θα αξιολογήσουμε την επαναληψιμότητα του αλγορίθμου, δηλαδή την ικανότητα του να δίνει όμοια αποτελέσματα όταν εκτελείται κάτω από περίπου ίδιες συνθήκες και δεδομένα.

Τέλος, στην Ενότητα 6.8, θα εξετάσουμε μειονεκτήματα και προβλήματα της μεθόδου μας.

### 6.1 Παράμετροι Αξιολόγησης

Στο πλαίσιο της εργασίας που υλοποιήσαμε, τα κύρια χαρακτηριστικά που αξιολογήσαμε και προσπαθήσαμε να βελτιώσουμε κινούνται στους ακόλουθους άξονες:

- Η δυνατότητα αξιόπιστης αναγνώρισης ενός βρόχου και ο τρόπος με τον οποίο διορθώνεται ο χάρτης σε μια τέτοια περίπτωση.
- Η δυνατότητα σωστής εκτίμησης θέσης από το ρομπότ. Πρέπει ανά πάσα στιγμή να υπολογίζουμε σωστά τον κόμβο του χάρτη στον οποίο βρίσκεται το ρομπότ.
- Η δημιουργία ενός σωστού τοπολογικά και, κατά το δυνατόν μετρικά, χάρτη.



- Η διακριτοποίηση του χώρου που επιτυγχάνει η τοποθέτηση κόμβων, να είναι κατά το δυνατόν όσο πιο κοντά σε μια ανθρώπινη προσέγγιση.
- Ο αριθμός των κόμβων να είναι κατά το δυνατόν μικρότερος, αναπαριστώντας σωστά και επαρκώς το περιβάλλον.

Αν και η εφαρμογή σε πραγματικό χρόνο είναι μια σημαντική παράμετρος, προσπαθήσαμε να διατηρήσουμε τον χρόνο εκτέλεσης του αλγορίθμου σε χαμηλά επίπεδα. Όμως η δυνατότητα επεξεργασίας των δεδομένων σε off-line χρόνο, επικέντρωσε την προσοχή μας περισσότερο στην κατά το δυνατόν σωστή λειτουργία του αλγορίθμου, παρά στην ταχύτητα εκτέλεσης.

## 6.2 Τρόπος Αξιολόγησης

Όπως αναφέραμε στην Ενότητα 6.1, στόχος είναι η χαρτογράφηση που κάνουμε να συμφωνεί κατά το δυνατόν με αυτή που θα έκανε ένας ανθρώπινος παρατηρητής. Αυτό συνεπάγεται σωστή εισαγωγή νέων κόμβων, ακριβή αυτοεντοπισμό θέσης μέσα στον χάρτη και αναγνώριση βρόχων.

Για να δούμε εάν ένα τέτοιο μοντέλο παρατήρησης (με τα χαρακτηριστικά SURF) μπορεί να αναγνωρίζει με επιτυχία βρόχους, ελήφθη ένα σύνολο από εικόνες στο εσωτερικό των κτιρίων της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, που αφορούσαν το ίδιο σημείο, αλλά με διαφορές μεταξύ τους, είτε στην γωνία λήψης, είτε στην θέση λήψης, είτε στο περιεχόμενο. Στόχος μας ήταν να δούμε εάν το μοντέλο παρατήρησης μπορεί να αναγνωρίσει πρακτικά ίδιες περιοχές, που όμως διαφέρουν σε κλίμακα ή γωνία λήψης.

Επίσης, εξετάσαμε τον συνολικό αριθμό κόμβων που δημιουργούνται για διάφορες διαδρομές στον χώρο των κτιρίων, υιοθετώντας διάφορα μοντέλα παρατήρησης. Στόχος εδώ, δεν ήταν η αναγνώριση βρόχων, αλλά το να εξετάσουμε το πόσο καλά τοποθετούνται οι κόμβοι, καθώς και τον συνολικό αριθμό τους.

Τέλος, ενδιαφερθήκαμε για την συνολική απόδοση του αλγορίθμου, πραγματοποιώντας διαδρομές με το ρομπότ και εξετάζοντας συνολικά την απόδοσή του με βάση τις παραμέτρους της Ενότητας 6.1.

## 6.3 Συγκριτική Αξιολόγηση των Μοντέλων Παρατήρησης με Ιστογράμματα Εικόνας και Χαρακτηριστικά SURF

Στην ενότητα 4.1.2 είδαμε πώς λειτουργούν τα Ιστογράμματα Εικόνας και τα Χαρακτηριστικά SURF αντίστοιχα, και πώς παράγουν μια δεσμευμένη πιθανότητα  $p(z_t | x_k)$  όπου  $z_t$  η τρέχουσα εικόνα-παρατήρηση και  $x_k$  είναι ο  $k$ -οστός κόμβος στον που υποθέτουμε ότι είναι η εκτιμώμενη θέση του ρομπότ. Με βάση το μοντέλο παρατήρησης, ισχύει πάντα

$p(z_t | x_k) \in [0, 1]$ . Στην συνέχεια θα συγκρίνουμε τα μοντέλα παρατήρησης των χαρακτηριστικών SURF και των ιστογραμμάτων χρωματικού τόνου και κορεσμού σε διάφορους τομείς. Συγκεκριμένα, θα ασχοληθούμε με την αναγνώριση βρόχων, την εισαγωγή νέων κόμβων και των συνολικό αριθμό κόμβων που εισάγουν τα δύο μοντέλα παρατήρησης. Τονίζουμε ότι στα παρακάτω πειράματα έχουμε αφαιρέσει οποιαδήποτε επιρροή της μετρικής πληροφορίας στο τελικό αποτέλεσμα.

### 6.3.1 Αναγνώριση Βρόχων

Στην ενότητα αυτή θα εργαστούμε προσομοιώνοντας μια διαδρομή του ρομπότ, όπου έχοντας εισάγει ένα κόμβο (άρα και μια εικόνα που αντιστοιχεί σε αυτόν), υποθέτουμε ότι επιστρέφει στο ίδιο περίπου σημείο, αλλά με διαφορετική γωνία, σε ελαφρώς διαφορετική θέση, και πιθανώς βλέπει και κάποια διαφορετικά αντικείμενα (για παράδειγμα, ένα αντικείμενο που υπήρχε στο 1<sup>ο</sup> πέρασμα, δεν υπάρχει στο 2<sup>ο</sup>). Εξετάζουμε λοιπόν την δυνατότητα του συστήματος, να εντοπίσει ή όχι την ολοκλήρωση βρόχου. Θα συγκρίνουμε τα αποτελέσματα με χρήση μόνο ιστογραμμάτων εικόνας ή μόνο χαρακτηριστικών SURF.

Πρέπει να τονιστεί ότι η αναγνώριση ολοκλήρωσης βρόχου είναι το πιο σημαντικό κομμάτι του αλγορίθμου. Η αναγνώριση αυτή μπορεί να είναι αποτυχημένη με δύο τρόπους: ο 1<sup>ος</sup> είναι να έχουμε όντως ολοκληρώσει βρόχο, αλλά το σύστημα να μην τον αναγνωρίσει. Ο 2<sup>ος</sup> είναι να μην έχουμε κλείσει βρόχο αλλά το σύστημα να νομίσει ότι έχουμε. Αυτό πρακτικά είναι το πρόβλημα της αντιστοίχισης το οποίο τονίσαμε ιδιαίτερα στις πρώτες ενότητες: σημεία του χώρου που ναι μεν είναι διαφορετικά (χωρικά) αλλά μοιάζουν πάρα πολύ. Επειδή η αντιμετώπιση μιας τέτοιας κατάστασης απαιτεί την ενσωμάτωση μετρικής πληροφορίας, που να μας πληροφορεί για παράδειγμα ότι αυτά τα δύο σημεία (εικόνες) έχουν παρθεί από εντελώς διαφορετικό χώρο στον χάρτη, δεν θα την εξετάσουμε εδώ (καθώς μελετάμε μόνο το μοντέλο παρατήρησης).

Στην πλειοψηφία των περιπτώσεων (και όπως θα φανεί από τα παρακάτω παραδείγματα), τα χαρακτηριστικά SURF αναγνωρίζουν επιτυχώς ότι πρόκειται για δύο πρακτικά ίδιες εικόνες ενώ τα ιστογράμματα στις περισσότερες περιπτώσεις αποτυγχάνουν. Για τα χαρακτηριστικά SURF το κατώφλι που χρησιμοποιήθηκε στο ταίριασμα με τη μέθοδο του *κοντινότερου γείτονα* (Nearest Neighbour) είναι  $\theta = 0.75$ . Όπως θα δούμε παρακάτω, η παράμετρος αυτή



Σχήμα 19. Αριστερά (α): Αρχική θέση. Δεξιά (β): Επιστροφή στην ίδια θέση αλλά με λίγο διαφορετικό προσανατολισμό ο οποίος όμως αλλάζει αρκετά το περιεχόμενο της εικόνας.



**Σχημα 20. Αριστερά (α): Αρχική θέση. Δεξιά (β): Επιστροφή στην ίδια θέση αλλά με λίγο διαφορετικό προσανατολισμό ο οποίος όμως αλλάζει αρκετά το περιεχόμενο της εικόνας.**

παίζει ρόλο στην επίδοση των χαρακτηριστικών SURF.

Στα Σχήματα 19, 20, 21, τα χαρακτηριστικά SURF καταφέρνουν να αναγνωρίσουν το βρόχο, ενώ τα ιστογράμματα χρωματικού τόνου και κορεσμού αποτυγχάνουν. Χαρακτηριστικό του εκάστοτε ζεύγους εικόνων στα σχήματα αυτά, είναι ότι έχουνε ληφθεί από την ίδια ακριβώς



**Σχημα 21. Αριστερά (α): Αρχική θέση. Δεξιά (β): Επιστροφή στην ίδια θέση αλλά με λίγο διαφορετικό προσανατολισμό ο οποίος όμως αλλάζει αρκετά το περιεχόμενο της εικόνας.**

θέση, με μια όμως περιστροφή που μεταβάλλει το  $\frac{1}{4}$  περίπου της εικόνας.

Για το Σχήμα 22, βλέπουμε ότι απλά υπάρχει μια αλλαγή θέσης ανάμεσα στις δύο λήψεις. Σε αυτή την περίπτωση, και οι δύο μέθοδοι πετυχαίνουν να αναγνωρίσουν το βρόχο.

Γενικά, σαν παρατήρηση πρέπει να πούμε ότι τα χαρακτηριστικά SURF σε όσα πειράματα πραγματοποιήσαμε, ήταν πολύ αποτελεσματικά σε ότι αφορά την αναγνώριση βρόχων. Τα ιστογράμματα, αν και ταχύτερα σε χρόνο εκτέλεσης, δεν είναι πολύ αποτελεσματικά σε αυτό το ρόλο. Με βάση τα παραπάνω παραδείγματα, βλέπουμε ότι εάν αλλάξει το περιεχόμενο του  $\frac{1}{4}$ ου της εικόνας (όπως άλλωστε συμβαίνει στα Σχήματα 19,20,21 όπου έχουμε μια περιστροφή αριστερά με αποτέλεσμα να αλλάξει το δεξί μέρος της εικόνας), τότε τα ιστογράμματα αποτυγχάνουν στην αναγνώριση του βρόχου.

Στον ακόλουθο πίνακα, παραθέτουμε τους βαθμούς ομοιότητας που έδωσαν τα μοντέλα παρατήρησης των Ιστογραμμάτων και των χαρακτηριστικών SURF, για τα παραπάνω σχήματα.

Σχήμα	Histogram Correlation	SURF Correlation
19(α)-19(β)	0,413404	0,949972
20(α)-20(β)	0,3799	0,95
21(α)-21(β)	0,4282	0,9639
22(α)-22(β)	0,5232	0,95



Σχημα 22. Αριστερά (α): Αρχική θέση. Δεξιά (β): Επιστροφή στην αρχική θέση με μια μικρή μετατόπιση.

### 6.3.2 Εισαγωγή Νέου Κόμβου

Η ενότητα αυτή ασχολείται με τις διαφορές των δύο μεθόδων κατά την εισαγωγή νέων κόμβων. Όπως είπαμε, ο μεγάλος αριθμός κόμβων δεν είναι επιθυμητός, γιατί δεν προσφέρει καλή διακριτοποίηση του χώρου. Νέοι κόμβοι προστίθενται όταν κριθεί από το μοντέλο παρατήρησης σε συνδυασμό με το μοντέλο κίνησης, ότι κανένας από τους υπάρχοντες κόμβους δεν αντιστοιχεί στην τρέχουσα παρατήρηση, δηλαδή όταν η τελική πιθανότητα είναι μικρότερη από ένα κατώφλι (εδώ, 0.49). Με δεδομένο ότι θέλουμε να συγκρίνουμε αποκλειστικά τα μοντέλα παρατήρησης στον τομέα της εισαγωγής νέων κόμβων, τα πειράματα αυτά έγιναν απαλλαγμένα από την επιρροή του μοντέλου κίνησης. Τα αποτελέσματα βασίζονται αποκλειστικά στο μοντέλο παρατήρησης, και άρα στην σταθερότητα των δύο μεθόδων: ιστογραμμάτων και χαρακτηριστικών SURF. Στόχος είναι να δούμε τις ποιοτικές διαφορές στην εισαγωγή κόμβων ανάμεσα στις δύο μεθόδους. Φυσικά, το πρόβλημα της τοποθέτησης νέων κόμβων και κατ'επέκταση της απόφασης για το εάν δύο εικόνες ταιριάζουν ή όχι, είναι ένα ζήτημα αρκετά υποκειμενικό.

Γενικά, τα ιστογράμματα ως λιγότερο ευσταθής μέθοδος τείνουν να παράγουν περισσότερους κόμβους. Βέβαια όπως θα δείξουμε, και τα χαρακτηριστικά SURF έχουν κάποιες αδυναμίες και σε μερικές περιπτώσεις παράγουν και αυτά πολλούς κόμβους. Στο σύνολο όμως, υπερτερούν σαφώς των ιστογραμμάτων.

Ακόμα, στην συντριπτική πλειοψηφία των περιπτώσεων, σημαντικά νέα στοιχεία του περιβάλλοντος, όπως πόρτες, πυροσβεστήρες, ντουλάπια, δημιουργούν νέους κόμβους ανεξαρτήτως μεθόδου του μοντέλου παρατήρησης.

Τα χαρακτηριστικά SURF τείνουν να καθυστερούν περισσότερο την εισαγωγή κόμβων από τα ιστογράμματα και έτσι απλώνουν περισσότερο τους κόμβους στον χώρο. Στο Σχημα 23,



**Σχημα 23.** Αριστερά: Η αρχική εικόνα(κόμβος). Μέση: Ο αμέσως επόμενος κόμβος που τοποθετείται με βάση τα ιστογράμματα. Δεξιά: Ο αμέσως επόμενος κόμβος (από τον αρχικό) που τοποθετείται με βάση τα χαρακτηριστικά SURF.

βλέπουμε αρχικά την 1<sup>η</sup> εικόνα (στιγμιότυπο 1), που αντιστοιχεί και στον αρχικό κόμβο σε μια διαδρομή που ακολουθήσαμε. Στη μεσαία και τη δεξιά εικόνα, βλέπουμε την παρατήρηση που οδηγεί τα ιστογράμματα και τα χαρακτηριστικά SURF αντίστοιχα, στο να εισάγουν ένα νέο κόμβο. Παρατηρούμε ότι στη δεξιά εικόνα, έχουμε προχωρήσει περισσότερο απ'ότι στην μεσαία και πλέον στην δεξιά εικόνα, η 1<sup>η</sup> κόκκινη πόρτα δεν φαίνεται.

Ένα ακόμα παράδειγμα που εμφανίζεται στη μέση της διαδρομής, φαίνεται στο Σχημα 24.



**Σχημα 24.** Πάνω αριστερά(α): Κόμβος που εισάγουν τα ιστογράμματα. Πάνω δεξιά (β): Αμέσως επόμενος κόμβος που εισάγουν τα ιστογράμματα. Κάτω αριστερά(γ): Κόμβος στο ίδιο περίπου μέρος με τον (α), εισηγμένος από τα χαρακτηριστικά SURF. Κάτω δεξιά(δ): Αμέσως επόμενος κόμβος σε σχέση με τον (γ), εισηγμένος από τα χαρακτηριστικά SURF.

Όπως φαίνεται, με ίδιο σημείο εκκίνησης, τα χαρακτηριστικά SURF έχουν πιο αραιούς κόμβους σε σχέση με τα ιστογράμματα. Επιλέξαμε τα συγκεκριμένα παραδείγματα κυρίως λόγω της αντιπροσωπευτικότητάς τους. Η τάση που φαίνεται στα παραδείγματα αυτά, επιβεβαιώνεται σε όλα τα πειράματα που εκτελέσαμε. Σε επόμενη ενότητα, θα δούμε και σε συνολικό βαθμό τη διαφορά στους αριθμούς κόμβων.

### 6.3.3 Ο ρόλος του Κατώφλιου $\theta$ των χαρακτηριστικών SURF

Στην ενότητα 4.1.2.2 αναφερθήκαμε στο κατώφλι  $\theta$  το οποίο χρησιμοποιείται από τον αλγόριθμο που ταιριάζει χαρακτηριστικά SURF από διαφορετικές εικόνες. Το κατώφλι αυτό παίζει σημαντικό ρόλο στο πώς ανταποκρίνεται ο αλγόριθμος. Γενικά όσο μικρότερο το κατώφλι, τόσο αυστηρότερο και το ταίριασμα των χαρακτηριστικών σημείων. Για παράδειγμα, αν  $\theta = 0.5$  αυτό σημαίνει ότι για να ταιριάξουν δύο χαρακτηριστικά σημεία, θα πρέπει η Ευκλείδεια απόστασή τους να είναι το πολύ η μισή από την Ευκλείδεια απόσταση του ίδιου σημείου με το 2<sup>ο</sup> καλύτερο ταίριασμα. Όσο μεγαλύτερο το κατώφλι, τόσο πιο εύκολα ταιριάζουν τα σημεία. Με το να αλλάζει κανείς το κατώφλι, επηρεάζει άμεσα τον αριθμό των εισαγόμενων κόμβων: μικρότερο κατώφλι γενικά συνεπάγεται περισσότερους κόμβους, ενώ μεγαλύτερο κατώφλι συνεπάγεται λιγότερους κόμβους, με το ρίσκο όμως να έχουμε λάθος ταίριασμα μεταξύ των εικόνων.

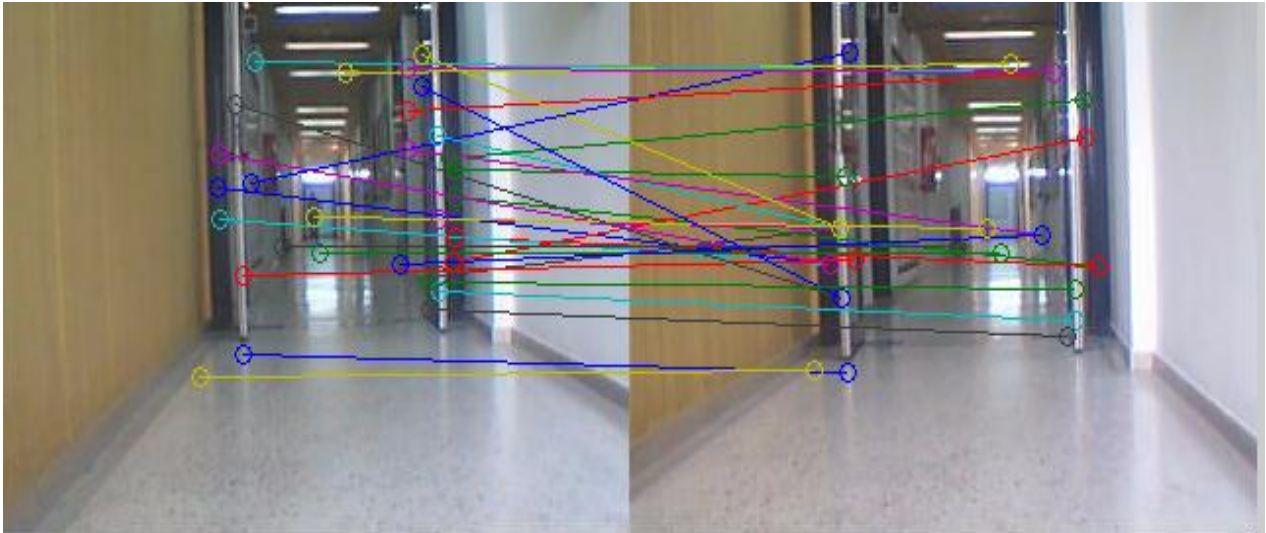
Συνεπώς, η τιμή του κατωφλίου είναι ένας συμβιβασμός ανάμεσα στην ανάγκη να ταιριάζουμε σημεία προκειμένου να βγάλουμε συμπεράσμα για το αν δύο εικόνες ταυτίζονται, και στο να μην γίνονται λάθος ταυτοποιήσεις που μπορεί να έχουν καταστροφικά αποτελέσματα στην χαρτογράφηση και τον εντοπισμό θέσης.

Οι τιμές που χρησιμοποιήσαμε για το κατώφλι ήταν αρκετά μικρές, μεταξύ 0.65-0.75.

Για να δείξουμε πώς το κατώφλι επηρεάζει πρακτικά την απόδοση του αλγορίθμου, παραθέτουμε κάποια χαρακτηριστικά παραδείγματα.

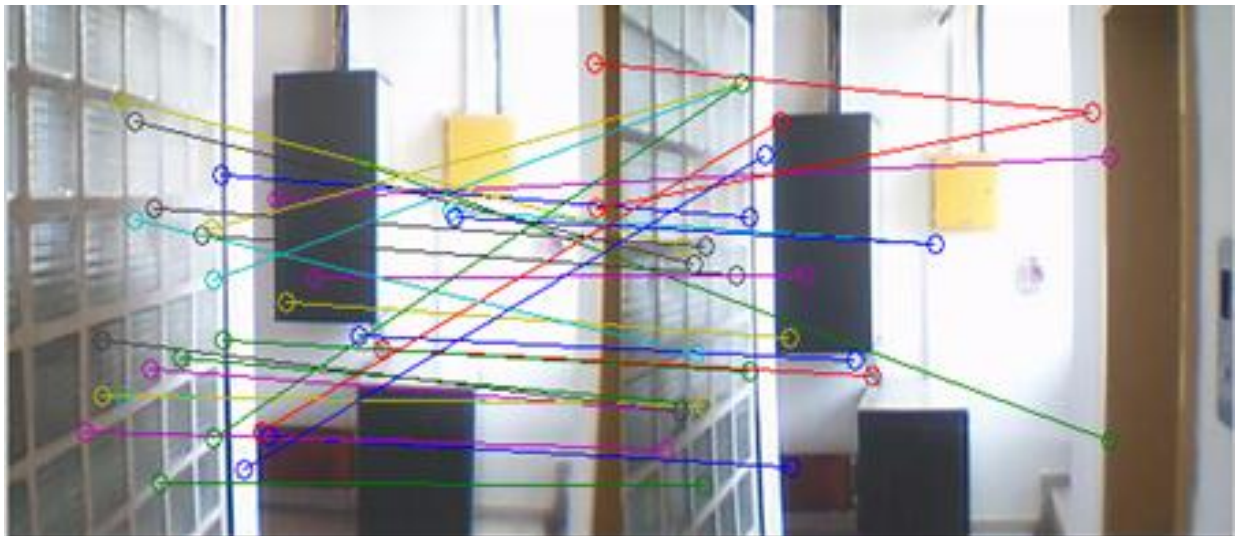
Οι ακόλουθες δύο εικόνες αποτελούν στιγμιότυπα από μια διαδρομή που πραγματοποιήσαμε με το ρομπότ, και αντιστοιχούν σε δύο παρατηρήσεις του περιβάλλοντος από την οπτική γωνία του ρομπότ. Στο Σχημα 25 βλέπουμε πώς τα χαρακτηριστικά ανάμεσα στις δύο εικόνες ταιριάζουν μεταξύ τους. Εδώ, ισχύει για το κατώφλι  $\theta = 0.75$ .





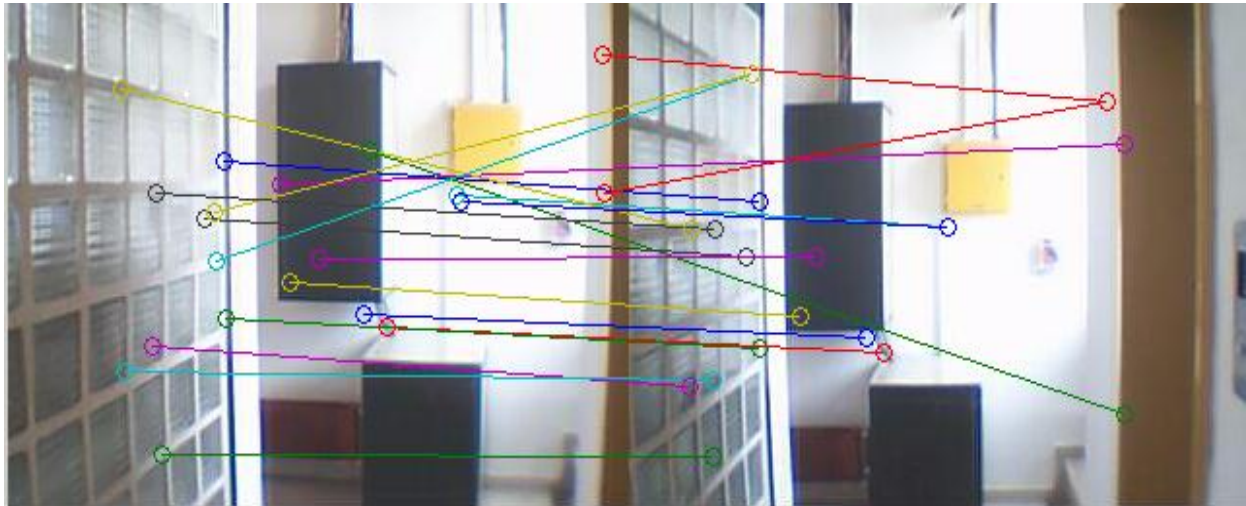
Σχημα 25.Αριστερά (α): Η αρχική θέση. Δεξιά (β): Επιστροφή στην αρχική θέση με διαφορετική γωνία.

Βλέπουμε γενικά ότι τα περισσότερα σημεία έχουν ταιριάξει σωστά, όμως υπάρχουν και μερικά λάθη στην αντιστοίχιση. Το ίδιο φαίνεται και στο ακόλουθο παράδειγμα ( $\theta = 0.75$ ):



Σχημα 26.Αντιστοίχιση χαρακτηριστικών SURF. Προσέξτε ότι κάποια σημεία δεν αντιστοιχίζονται σωστά.

Το φαινόμενο αυτό μπορεί να αντιμετωπιστεί εάν κανείς αυστηρήνει το κριτήριο, δηλαδή μικρώνει το κατώφλι. Έτσι για  $\theta = 0.65$  έχουμε:



Σχημα 27. Αντιστοίχιση χαρακτηριστικών με μικρότερο κατώφλι. Πλέον έχουν ταιριάξει λιγότερα σημεία, αλλά στην συντριπτική πλειοψηφία η αντιστοίχιση είναι τέλεια.

Παρατηρούμε τώρα πως ναι μεν έχουν ταιριάξει λιγότερα σημεία, όμως πλέον έχουμε και λιγότερα λάθη στις αντιστοιχίσεις.

#### 6.3.4 Συγκριτική Αξιολόγηση Ιστογραμμάτων-SURF σε Ολοκληρωμένες Διαδρομές

Έχοντας εξετάσει ποιοτικά, τις διαφορές στην επίδοση τόσο στην αναγνώριση βρόχων, όσο και στην εισαγωγή νέων κόμβων, θα δούμε τώρα την συνολική επίδοση των ιστογραμμάτων και των χαρακτηριστικών SURF σε ολοκληρωμένες διαδρομές.

Σε ένα σύνολο 10 διαδρομών, αναφέρουμε τα αποτελέσματα των 6 πιο αντιπροσωπευτικών. Από αυτές τις 6, οι 3 έγιναν με χρήση μιας κάμερας χειρός λαμβάνοντας βίντεο κατά την βάδιση, ενώ οι άλλες 3 έγιναν με χρήση του ρομπότ Pioneer 3, με την κάμερα τοποθετημένη πάνω σε ένα φορητό υπολογιστή, που με τη σειρά του βρισκόταν πάνω στο ρομπότ (θα αναφερθούμε στην διασύνδεση με το ρομπότ στην επόμενη ενότητα). Στα πειράματα που δεν χρησιμοποιήθηκε το ρομπότ, και άρα δεν είχαμε δεδομένα οδομετρίας, χρησιμοποιήθηκε αποκλειστικά οπτική οδομετρία. Σε όλα τα πειράματα χρησιμοποιήθηκαν οι ίδιες παράμετροι, ειδικά αυτές που αφορούν τα κατώφλια παραμονής στον ίδιο κόμβο (0.60) και το κατώφλι επίσκεψης παλαιότερου κόμβου (0.49).



Αριθμός Κόμβων		
HIS Color Histograms	SURF	Ποσοστιαία Διαφορά Κόμβων (%)
36	19	-47%
36	23	-36%
42	33	-21%
25	15	-40%
23	17	-26%
29	19	-34%

Τα παραπάνω μεγέθη υποδηλώνουν μια μέση μείωση του αριθμού των κόμβων, της τάξης του 34%. Σε όλες τις περιπτώσεις, τα χαρακτηριστικά SURF πέτυχαν μειωμένο αριθμό κόμβων και καλύτερη κατανομή τους στο χώρο.

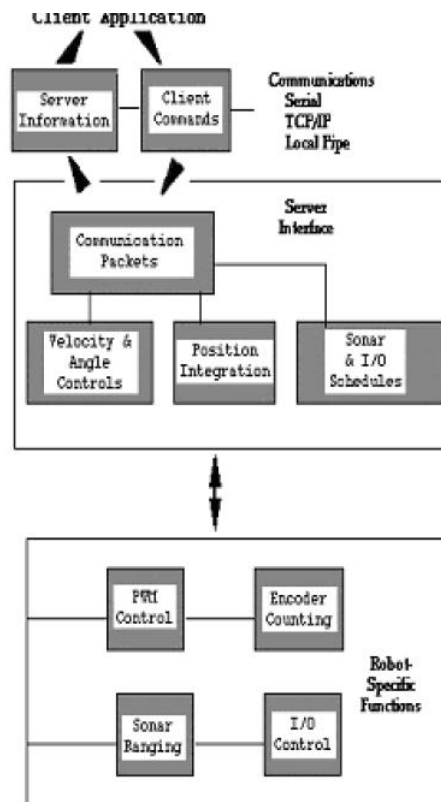
## 6.4 Πειραματική Διάταξη Τηλεχειριζόμενου Κινητού Ρομπότ για Συλλογή Δεδομένων

### 6.4.1.1 Σύντομη Περιγραφή του Ρομπότ

Για την διεξαγωγή πολλών πειραμάτων, συμπεριλαμβανομένου και αυτού που ακολουθεί στην Ενότητα 6.5, χρησιμοποιήθηκε το ρομπότ Pioneer 3-DX της ActivMedia Robotics. Το συγκεκριμένο μοντέλο εμφανίστηκε το 2003. Το ρομπότ αυτό είναι πολύ δημοφιλές και προσφέρει πολλές δυνατότητες στον χρήστη. Χρησιμοποιεί DC κινητήρες, ενώ διαθέτει κωδικοποιητές στους τροχούς και αισθητήρες sonar. Έχει διαφορετικό δύο τροχών καθώς και ένα ακόμα τροχό στο πίσω μέρος για ευστάθεια και καλύτερο έλεγχο. Επίσης δίνεται η δυνατότητα για προσθήκη επιπλέον εξαρτημάτων, όπως αυτοματοποιημένα συστήματα για επαναφόρτιση, αισθητήρες απόστασης laser, αρπάγη πέντε βαθμών ελευθερίας, κάμερες και άλλα.

Το λειτουργικό του σύστημα είναι το ActivMedia Robotics Operating System (AROS) και συνοδεύεται από πληθώρα λογισμικού σε C++, Java, Python.

### 6.4.1.2 Η Επικοινωνία με το Ρομπότ



**Σχημα 28. Σχηματική απεικόνιση της αρχιτεκτονικής επικοινωνίας με το ρομπότ.**

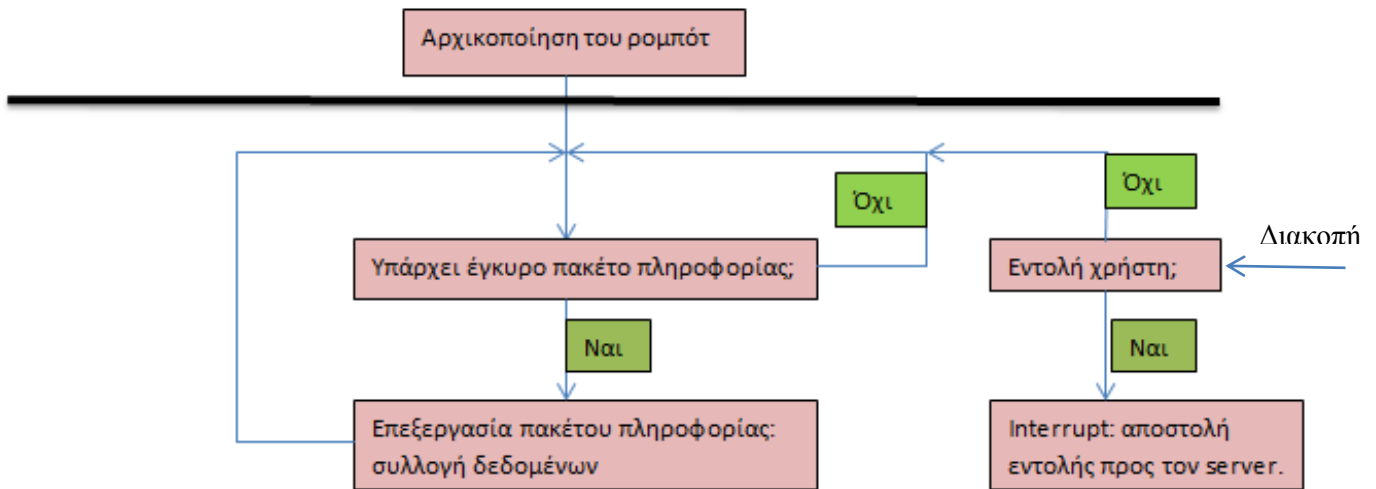
Η σύνδεση του φορητού υπολογιστή με το ρομπότ έγινε με καλώδιο Serial to USB. Για την επικοινωνία, το ρομπότ χρησιμοποιεί μια αρχιτεκτονική client-server όπου πρακτικά ο ελεγκτής του ρομπότ αναλαμβάνει τον έλεγχο όλων των λειτουργιών χαμηλού επιπέδου, όπως για παράδειγμα τον χειρισμό των κινητήρων και την συλλογή δεδομένων από τους αισθητήρες.

Με δεδομένη την παραπάνω αρχιτεκτονική απαιτείται εν συνεχεία μόνο ο προγραμματισμός του client, δηλαδή το κομμάτι που αφορά τις εντολές του χρήστη και την επεξεργασία των πακέτων πληροφορίας. Το πρωτόκολλο επικοινωνίας με το ρομπότ χρησιμοποιεί δύο είδη τέτοιων πακέτων πληροφορίας. Το πρώτο αφορά εντολές από τον πελάτη στο server του ρομπότ, ενώ το δεύτερο είναι πακέτα πληροφορίας από τον server στον πελάτη.

Τα πακέτα εντολών αποτελούνται από 7 μέχρι 9 bytes ανάλογα με την εντολή και αφορούν εντολές ελέγχου προς τους κινητήρες, τους αισθητήρες και άλλες λειτουργίες του ρομπότ. Κάθε πακέτο που στέλνεται πρέπει να συνοδεύεται και από 2 bytes ελέγχου (checksum). Τα bytes αυτά υπολογίζονται τόσο από τον πελάτη κατά την αποστολή του πακέτου, όσο και από τον server κατά την λήψη του. Εάν διαφέρουν, τότε υπάρχει σφάλμα κατά την αποστολή και το πακέτο θεωρείται άκυρο.

Οι εντολές κίνησης παρουσιάζουν το μεγαλύτερο ενδιαφέρον και μας απασχόλησαν ιδιαίτερα. Όλες οι εντολές κίνησης δίνονται σε μονάδες μέτρησης, ανεξάρτητες του ρομπότ, δηλαδή σε χιλιοστά και μοίρες. Στη συνέχεια το λειτουργικό σύστημα του ρομπότ μετατρέπει αυτά τα μεγέθη, σε αντίστοιχα νούμερα των κωδικοποιητών του χρησιμοποιώντας κάποιες δεδομένες παραμέτρους. Ο server επιστρέφει την τρέχουσα θέση και ταχύτητα του ρομπότ σε μονάδες ανεξάρτητες του ρομπότ.

Τα πακέτα πληροφορίας στέλνονται ανά τακτά χρονικά διαστήματα από τον server προς τον πελάτη. Περιέχουν πληροφορίες για την κατάσταση των κινητήρων, για τις μετρήσεις των



Σχημα 29. Αλγόριθμος επικοινωνίας με το ρομπότ κατά την διεξαγωγή των πειραμάτων.

κωδικοποιητών στους τροχούς, για την κατεύθυνση του ρομπότ, την ταχύτητα των τροχών, καθώς και μετρήσεις από τα sonar.

Η υλοποίηση του πελάτη έγινε εξ' ολοκλήρου σε περιβάλλον Matlab, αν και οι κατασκευαστές παρέχουν λογισμικό επικοινωνίας μέσω Java. Όπως είπαμε, η σύνδεση έγινε με καλώδιο Serial to USB. Οι εντολές δινόντουσαν μέσω ενός κινητού τηλεφώνου συνδεδεμένου με bluetooth προς το Matlab, το οποίο στην συνέχεια προωθούσε την εντολή στο ρομπότ.

Κατά την διεξαγωγή των πειραμάτων, χρησιμοποιήσαμε τέσσερις εντολές κίνησης: Κίνηση προς τα εμπρός, κίνηση προς τα πίσω, στροφή δεξιά και στροφή αριστερά.

Για την ολοκληρωμένη επικοινωνία με το ρομπότ, χρειαζόμασταν ουσιαστικά τις ακόλουθες ρουτίνες:

1. Η 1<sup>η</sup> ρουτίνα, έλεγχε εάν υπήρχε έγκυρο πακέτο πληροφορίας από το ρομπότ. Υπολόγιζε το checksum των εισερχόμενων bytes και το συνέκρινε με αυτό που έστελνε ο server του ρομπότ. Εάν τα δύο checksums διέφεραν, τότε το πακέτο απορριπτόταν.
2. Η 2<sup>η</sup> ρουτίνα, εκτελείται όταν υπήρχε έγκυρο πακέτο. Σε αυτή, επεξεργαζόμασταν το πακέτο, παίρνοντας όλη την απαραίτητη πληροφορία από το ρομπότ. Συγκεκριμένα συλλέγαμε πληροφορίες για την θέση και τον προσανατολισμό του.
3. Η 3<sup>η</sup> ρουτίνα, ήταν υπεύθυνη για την λήψη εντολών από το κίνητο τηλέφωνο (ή από κάποιο άλλο μέσο, όπως το πληκτρολόγιο). Η ρουτίνα αυτή εκτελείται με interrupts: οποιαδήποτε εντολή από τον χρήστη προς το ρομπότ, είχε προτεραιότητα και έκανε διακοπή για να εκτελεστεί αυτή η ρουτίνα.
4. Τέλος, υπήρχαν και άλλες δύο μικρότερες ρουτίνες. Η μία υπολόγιζε το checksum, ενώ η δεύτερη έστελνε τις εντολές προς το ρομπότ, σύμφωνα με το πρωτόκολλο επικοινωνίας.

Στο Σχημα 29, φαίνεται η μορφή του αλγορίθμου που υλοποιήσαμε για την επικοινωνία με το ρομπότ.

#### **6.4.1.3 Η Πειραματική Διάταξη**

Η διεξαγωγή των πειραμάτων έγινε με χρήση του ρομπότ, ενός φορητού υπολογιστή, μιας απλής web κάμερας, και ενός κινητού τηλεφώνου.

Ο φορητός υπολογιστής, στον οποίο τοποθετήθηκε η κάμερα, στηρίχθηκε πάνω στο ρομπότ. Στην διεξαγωγή των πειραμάτων μας ενδιέφερε η συλλογή δύο ειδών δεδομένων: εικόνας και οδομετρίας του ρομπότ. Κατά την συλλογή των δεδομένων, κάθε εικόνα συνδεόταν με μια θέση πάνω στο επίπεδο, και ένα προσανατολισμό.

Ο αλγόριθμος της επικοινωνίας είναι ακριβώς αυτός που φαίνεται στο Σχημα 29, με τον επιπλέον φόρτο της συλλογής των εικόνων. Έτσι, με ρυθμό 15 καρέ το δευτερόλεπτο, εκτελείτο μια ρουτίνα η οποία ελάμβανε την τρέχουσα εικόνα της κάμερας και την αποθήκευε μαζί την αντίστοιχη μετρική πληροφορία. Στο τέλος του πειράματος, το σύνολο των εικόνων που είχαν συλλεχθεί, μετατρέπονταν σε μορφή video.

Στο Σχημα 30, φαίνεται η πειραματική διάταξη.



**Σχημα 30. Η πειραματική διάταξη: ρομπωτικό όχημα Pioneer 3-DX, με τοποθετημένο φορητό υπολογιστή και USB κάμερα.**

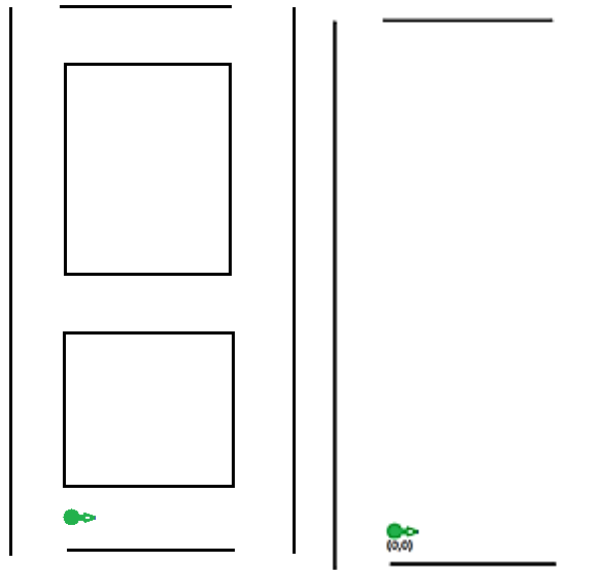
## 6.5 Ολοκληρωμένο Πείραμα Αναγνώρισης Βρόχου

Στην ενότητα αυτή, θα παρουσιάσουμε βήμα προς βήμα, το πώς λειτούργησε ο αλγόριθμος κατά την διεξαγωγή ενός πειράματος, στον 3<sup>ο</sup> όροφο των Παλαιών Κτιρίων της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών στο ΕΜΠ.

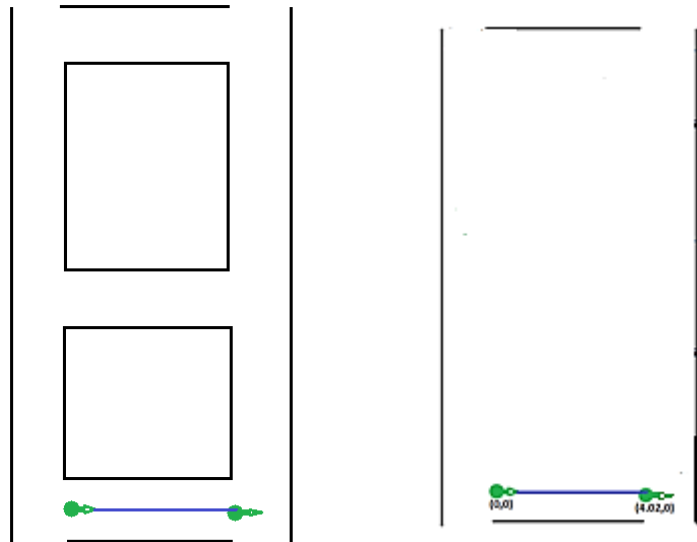
Στη διάρκεια του πειράματος συλλέξαμε εικόνες με ρυθμό 15 καρτέ το δευτερόλεπτο. Κάθε καρτέ συνδέθηκε με οδομετρική πληροφορία από τους μετρητές του ρομπότ. Η επεξεργασία των δεδομένων έγινε offline. Η διαδρομή διήρκεσε 3 λεπτά, ήταν περίπου 50 μέτρα και αποτελούσε ένα μεγάλο παραλληλόγραμμο. Στο μοντέλο παρατήρησης χρησιμοποιήσαμε αποκλειστικά χαρακτηριστικά SURF. Η επεξεργασία των δεδομένων έγινε και με χρήση μόνο των ιστογραμμάτων χρωματικού τόνου και κορεσμού, προκειμένου να διαπιστωθεί εάν ο αριθμός των κόμβων στην περίπτωση αυτή θα ήταν μεγαλύτερος από αυτόν που προέκυψε με τη χρήση των χαρακτηριστικών SURF.

Στο 1<sup>ο</sup> μέρος της παρουσίασης, παρουσιάζουμε τρία στοιχεία: στα αριστερά βλέπουμε το εκάστοτε στιγμιότυπο που οδήγησε στην δημιουργία ενός νέου κόμβου στον χάρτη. Στη μέση βλέπουμε το πραγματικό χάρτη και την σχετική θέση του ρομπότ μέσα σε αυτόν (προσεγγιστικά, καθώς δεν διαθέταμε GPS ή κάποιο άλλο τρόπο να βρούμε την ακριβή θέση του ρομπότ μέσα στον χώρο). Τέλος, στο δεξί μέρος, βλέπουμε τον χάρτη όπως τον σχεδίασε ο αλγόριθμος. Ο χάρτης αυτός εμπεριέχει μετρική πληροφορία.

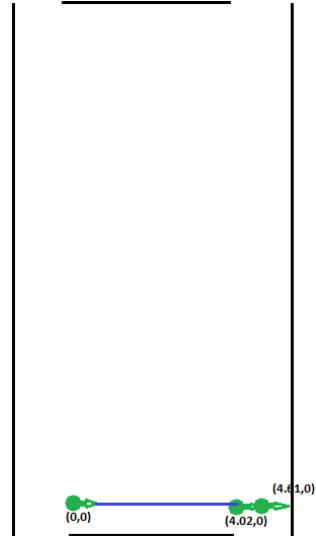
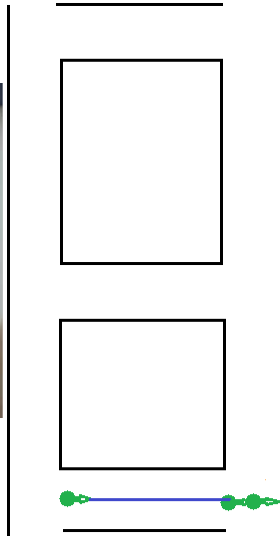
Στο σημείο αυτό, πρέπει να αναφέρουμε ότι στην διαδρομή που πραγματοποίησε το ρομπότ είχαμε τοποθετήσει δύο αντικείμενα, τα οποία φαίνονται Σημια 31 (το χάρτινο κουτί) και στο Σημια 35 (η μαύρη τσάντα). Τα αντικείμενα αυτά λειτούργησαν ως τεχνητά ορόσημα, βελτιώνοντας την επίδοση του αλγορίθμου, με το να παρέχουν επιπλέον οπτική πληροφορία σε σημεία του χώρου όπου δεν παρουσιάζουν ενδιαφέρον (για παράδειγμα, λευκοί τείχοι).



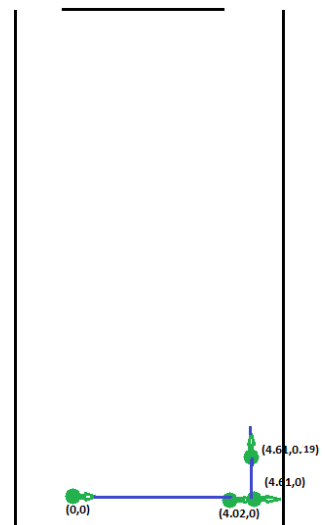
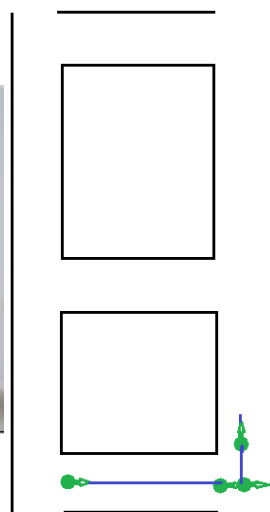
Σχημα 31. Στιγμιότυπο 1.



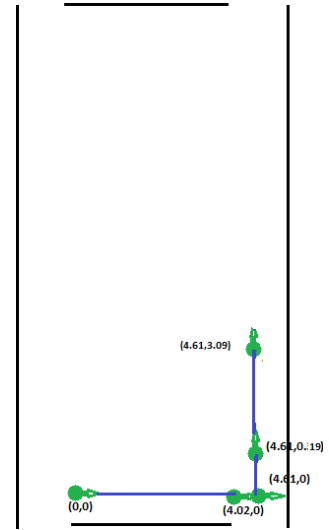
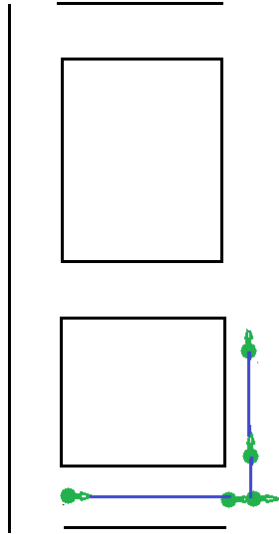
Σχημα 32. Στιγμιότυπο 271



Σχημα 33. Στιγμιότυπο 316.



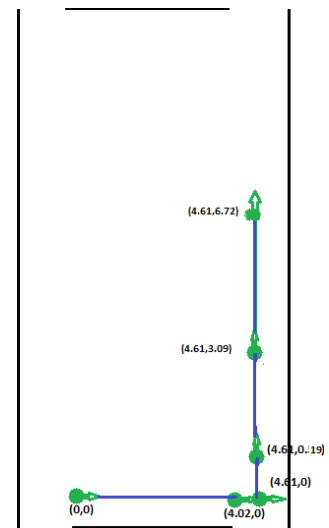
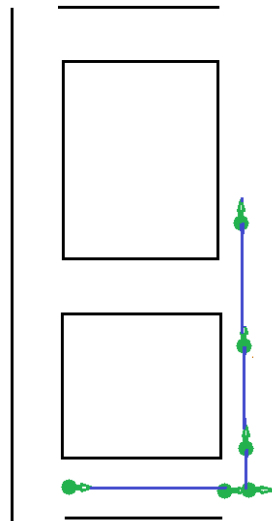
Σχημα 34.Στιγμιότυπο 401.



Σχημα 35. Στιγμιότυπο 596.



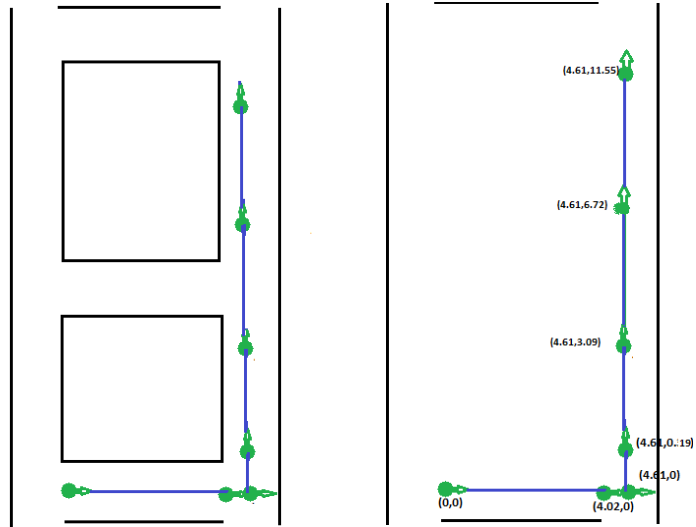
Σχημα 36.Στιγμιότυπο 818.



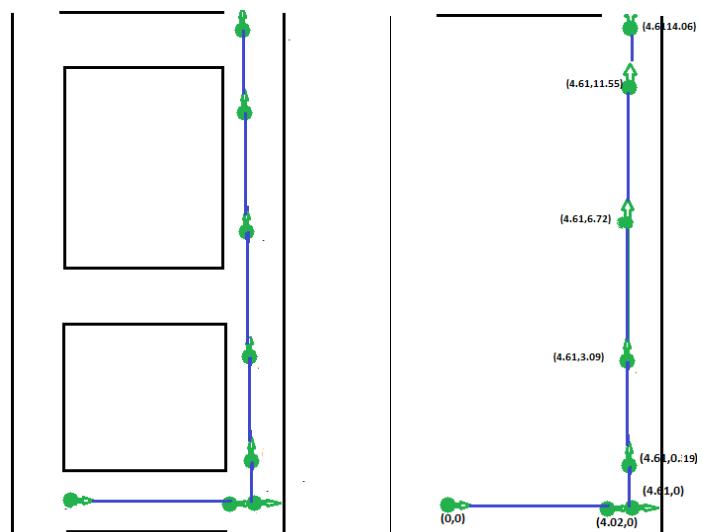


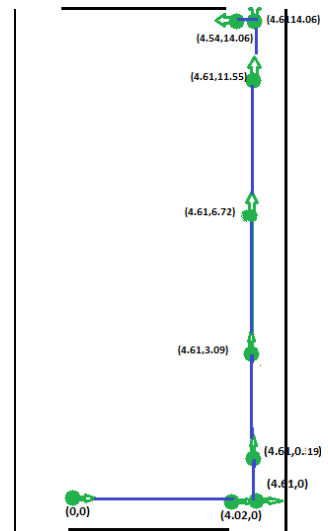
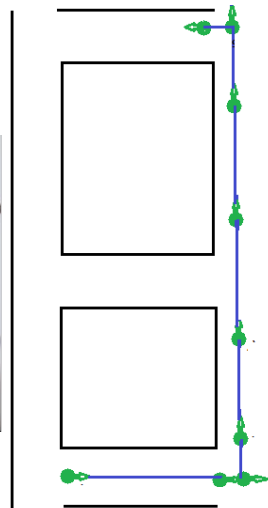
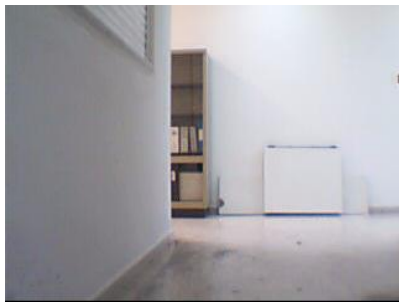


Σχημα 37. Στιγμιότυπο 1108

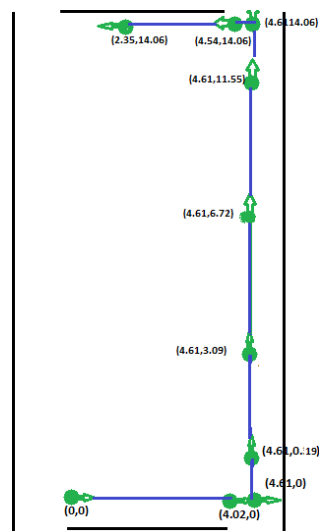
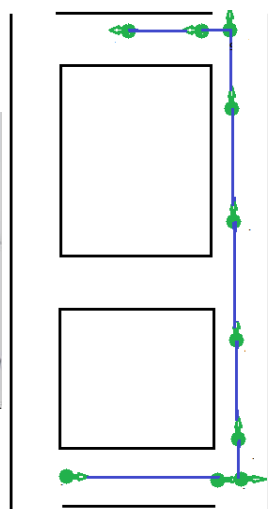
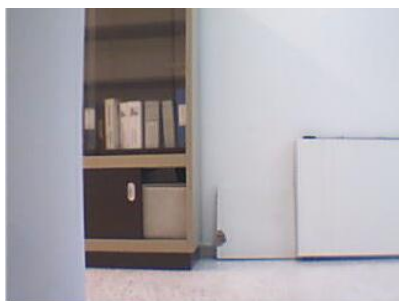


Σχημα 38. Στιγμιότυπο 1268

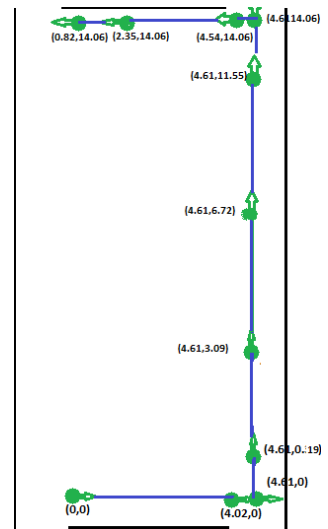
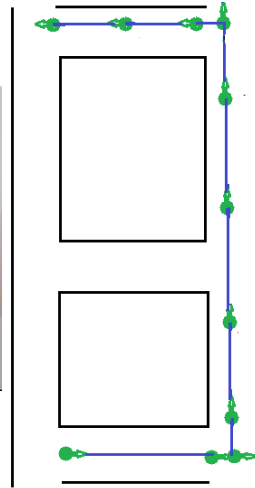




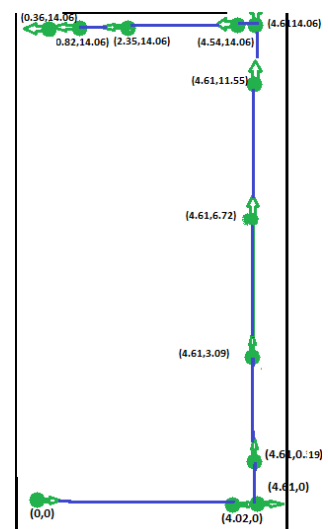
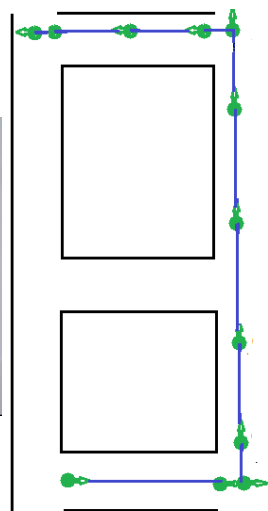
Σχημα 39. Στιγμιότυπο 1372



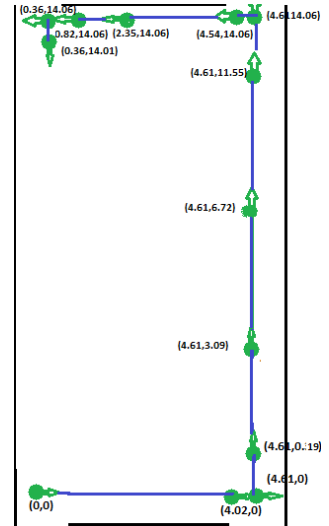
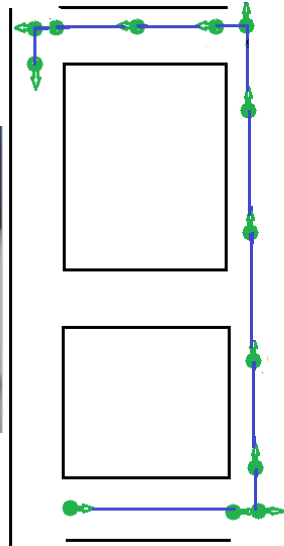
Σχημα 40. Στιγμιότυπο 1501



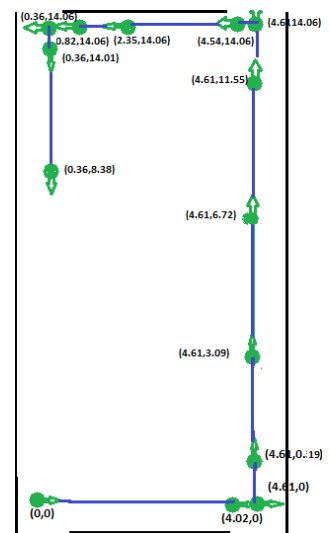
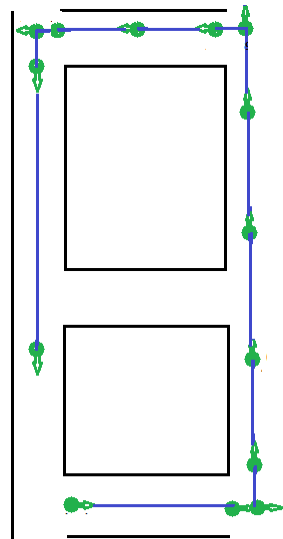
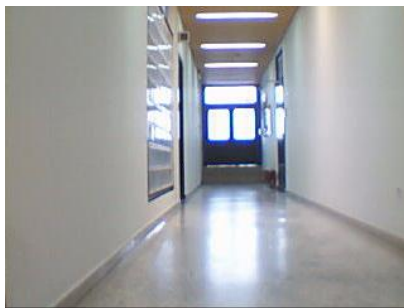
Σχημα 41. Στυγμιότυπο 1611



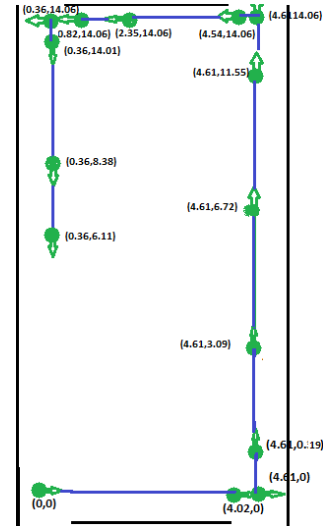
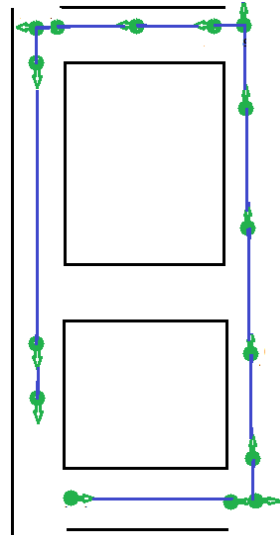
Σχημα 42. Στυγμιότυπο 1645



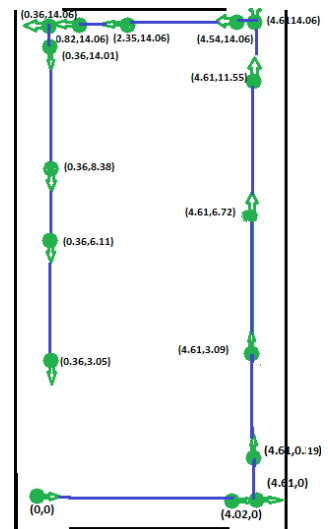
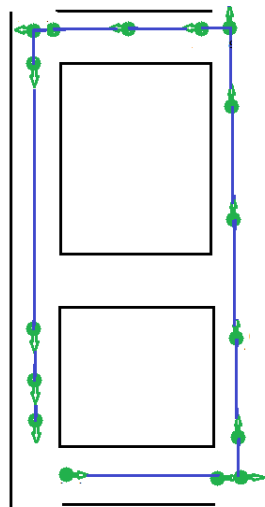
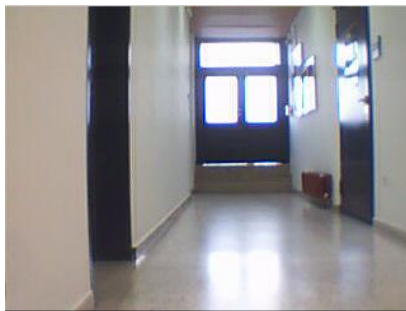
Σχημα 43. Στιγμιότυπο 1739



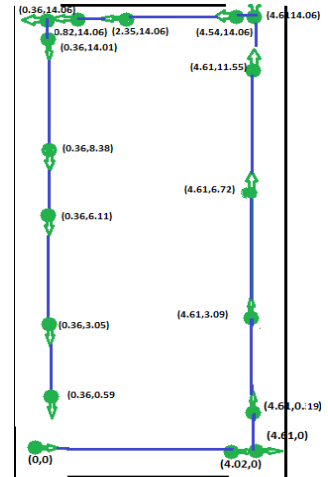
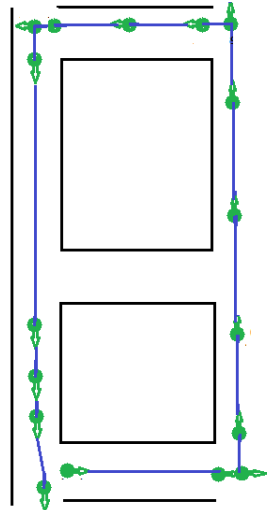
Σχημα 44. Στιγμιότυπο 2066



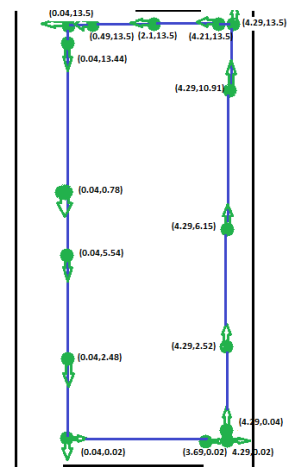
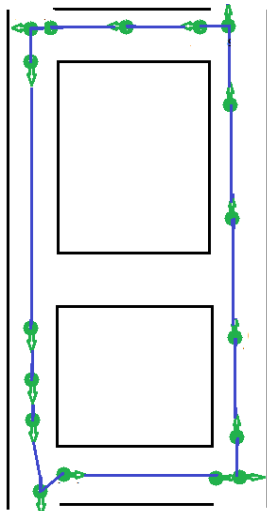
Σχημα 45. Στιγμιότυπο 2178



Σχημα 46. Στιγμιότυπο 2342



Σχημα 47. Στιγμιότυπο 2502



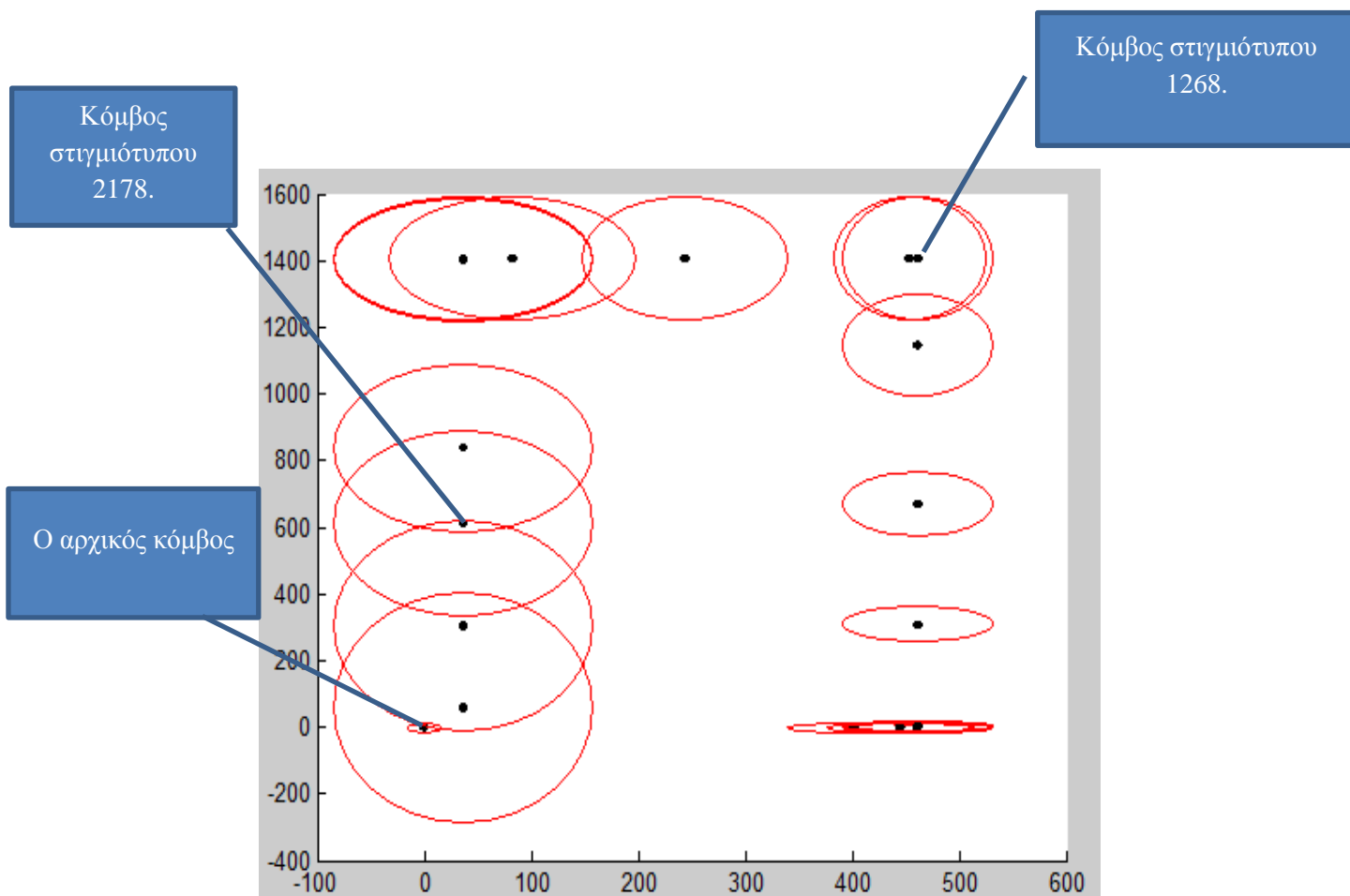
Σχημα 48. Στιγμιότυπο 2569

Βλέπουμε ότι με το στιγμιότυπο 2569 στο Σχημα 48, πραγματοποιείται η αναγνώριση του βρόχου. Το στιγμιότυπο 2569 μοιάζει εξαιρετικά με το 1<sup>ο</sup> στιγμιότυπο που αντιστοιχεί στον αρχικό κόμβο. Από το Σχημα 49 βλέπουμε ότι ο αρχικός κόμβος είναι μέσα στον χώρο αναζήτησης του βήματος πρόβλεψης. Ταυτόχρονα, σύμφωνα με το μοντέλο παρατήρησης οι τα δύο στιγμιότυπα ταιριάζουν.

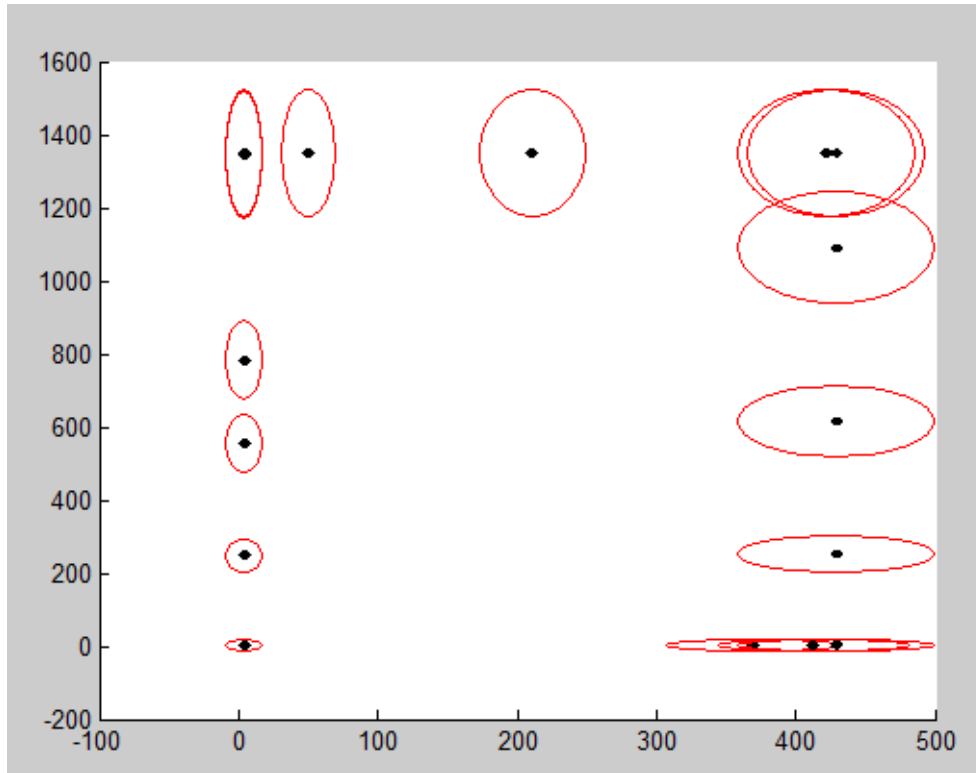
Μετά την αναγνώριση, οι δύο τελευταίοι κόμβοι τοποθετούνται στο ίδιο σημείο, το οποίο θεωρούμε ως την τρέχουσα θέση του ρομπότ. Ταυτόχρονα, διορθώνονται οι θέσεις των προηγούμενων κόμβων για να διατηρηθεί η συνέπεια του χάρτη, δηλαδή οι σχετικές αποστάσεις των υπόλοιπων κόμβων. Τέλος, ανανεώνουμε τις αβεβαιότητες των κόμβων, με βάση το ανάστροφο κινηματικό μοντέλο. Οι διαφορές φαίνονται στο Σχημα 49 και στο Σχημα 50 ( η κλίμακα είναι σε εκατοστά). Με κόκκινο συμβολίζουμε τον χώρο της διασδιάστατης Gaussian αβεβαιότητας του κάθε κόμβου (εύρους  $3\sigma_x, 3\sigma_y$ ).

Συνολικά τοποθετήθηκαν 19 κόμβοι. Με τα ίδια δεδομένα, αλλά με το μοντέλο παρατήρησης που βασίζεται στα ιστογράμματα, τοποθετήθηκαν συνολικά 29 κόμβοι.

Στο Σχημα 50 βλέπουμε την μορφή του τελικού χάρτη όπως φαίνεται στο περιβάλλον του Matlab.



Σχημα 49. Ο χάρτης πριν την αναγνώριση βρόχου.



Σχημα 50. Ο χάρτης μετά την αναγνώριση του βρόχου. Παρατηρείστε τις ελαφρώς διαφορετικές θέσεις των κόμβων και την διαφορά στις αβεβαιότητες της θέσης τους.

## 6.6 Η Συνεισφορά της Μετρικής Πληροφορίας – Συνέργεια Βήματος Πρόβλεψης και Βήματος Ανανέωσης

Κάτι που πρέπει κανείς να παρατηρήσει είναι η συνεισφορά της μετρικής πληροφορίας, και το μοντέλο αβεβαιότητας της κίνησης στο αποτέλεσμα του αλγορίθμου. Όπως είδαμε στην Ενότητα 5.2.1.2, η αβεβαιότητα θέσης του ρομπότ επηρεάζει άμεσα το εύρος αναζήτησης κόμβων στον ήδη υπάρχοντα χάρτη. Με άλλα λόγια, μικρή αβεβαιότητα θέσης σημαίνει ότι ψάχνουμε κοντά στην τρέχουσα εκτιμώμενη θέση του ρομπότ, ενώ μεγάλη αβεβαιότητα σημαίνει διευρυμένη περιοχή αναζήτησης.

Αυτή η λογική αναζήτησης στο χώρο του χάρτη ως το 1<sup>ο</sup> βήμα του διακριτού φίλτρου Bayes μας δίνει την δυνατότητα να αντιμετωπίζουμε το πρόβλημα της αντιστοίχισης εικόνων που μοιάζουν πολύ, αλλά τελικά αντιστοιχούν σε διαφορετικά σημεία του χώρου. Αλγοριθμικά, αυτό σημαίνει ότι αν εκτελούσαμε μόνο το βήμα παρατήρησης (ή είχαμε ένα βήμα πρόβλεψης που είχε μεγάλο εύρος αναζήτησης) τότε θα αντιστοιχίζαμε δύο εικόνες που μοιάζουν πολύ, αλλά αντιστοιχούν σε διαφορετικά σημεία του χάρτη, στον ίδιο κόμβο. Ένα τέτοιο λάθος μπορεί να αποβεί καταστροφικό για τον αλγόριθμο μας.

Για παράδειγμα στο Σχημα 51 βλέπουμε δύο διαφορετικά στιγμιότυπα που μοιάζουν πάρα πολύ. Συγκεκριμένα, το μοντέλο παρατήρησης με τα χαρακτηριστικά SURF μας δίνει



$p_{SURF} = 0.6351$ , που σημαίνει ότι οι εικόνες ταιριάζουν. Από το Σχημα 49, βλέπουμε ότι η αβεβαιότητα του κόμβου που αντιστοιχεί στο στιγμιότυπο 2178 είναι μεν μεγάλη, αλλά όχι αρκετά μεγάλη για να λάβει υπ' όψιν το βήμα πρόβλεψης τον κόμβο που αντιστοιχεί στο



**Σχημα 51.** Αριστερά: Το στιγμιότυπο 1268 από την προηγούμενη διαδρομή. Δεξιά: Το στιγμιότυπο 2178 της ίδιας διαδρομής.

στιγμιότυπο 1268. Αυτή η μετρική συσχέτιση των δύο κόμβων, μας επιτρέπει να διαχωρίσουμε τις δύο εικόνες, και τελικά να πάρουμε την σωστή απόφαση και να μην θεωρήσουμε ότι βρισκόμαστε στον κόμβο του στιγμιότυπου 1268, ενώ έχουμε σαν τρέχουσα παρατήρηση το στιγμιότυπο 2178.

Το βασικό συμπέρασμα από το παραπάνω παράδειγμα, είναι ότι το μεταβαλλόμενο εύρος αναζήτησης στο βήμα πρόβλεψης παίζει ιδιαίτερα σημαντικό ρόλο στη συνέργεια των βημάτων πρόβλεψης και ανανέωσης και βοηθά στην επίλυση του προβλήματος της αντιστοίχισης. Καθώς η απόσταση που διανύει το ρομπότ μεγαλώνει, μεγαλώνει αντίστοιχα και ο χώρος αναζήτησης.

Μπορούμε να φανταστούμε τί θα γινόταν εάν το εύρος αναζήτησης ήταν πάντα σταθερό. Εάν το εύρος αυτό ήταν ιδιαίτερα μεγάλο για να μας επιτρέψει μια αρκετά μεγάλη αναζήτηση στον χώρο του χάρτη, κάτι ιδιαίτερα επιθυμητό σε μια μεγάλη διαδρομή του ρομπότ, τότε ο αλγόριθμος θα αναγνώριζε ως ολοκλήρωση βρόχου τις εικόνες στο Σχημα 51.

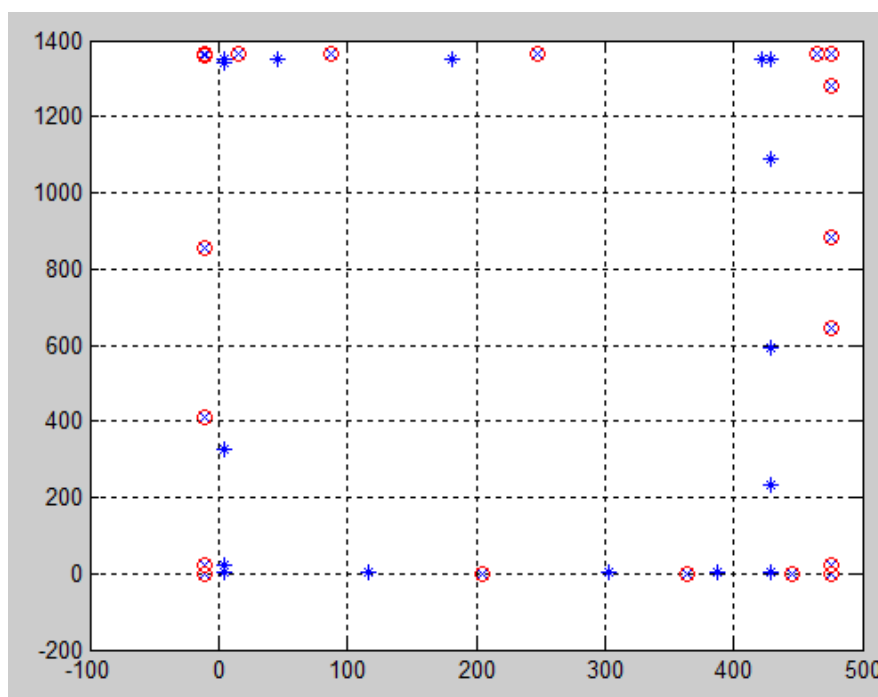
Αντίθετα, με μικρό εύρος αναζήτησης πιθανώς να αγγούσαμε κάποιους βρόχους, χάνοντας έτσι πολύτιμη πληροφορία και ταυτόχρονα θα προσθέταμε πολύ περισσότερους κόμβους στον χάρτη. Έτσι λοιπόν, προκύπτει ότι η συμπεριφορά ενός αλγορίθμου χωρίς μετρική πληροφορία θα ήταν απόλυτα εξαρτώμενη από το εύρος έρευνας του χάρτη που θα επιλέγαμε εκ των προτέρων να έχουμε.

## 6.7 Επαναληψιμότητα

Είναι σημαντικό για τον αλγόριθμό μας να είναι συνεπής, δηλαδή για παρόμοια δεδομένα να δίνει αντιστοίχως παρόμοια αποτελέσματα. Για να εξετάσουμε το κατά πόσο κάτι τέτοιο ισχύει, κάναμε την ίδια διαδρομή με αυτή που δείξαμε στην Ενότητα 6.5 ακόμα δύο φορές. Φυσικά οι δύο διαδρομές δεν ήταν ακριβώς ίδιες, αλλά διέφεραν σε ό,τι αφορούσε το ακριβές μονοπάτι του ρομπότ και τον προσανατολισμό του σε κάθε στιγμή. Ιδιαίτερα σημαντική είναι η απουσία από την 2<sup>η</sup> διαδρομή όλων των τεχνητών οροσήμων που υπήρχαν στην 1<sup>η</sup>. Τα ορόσημα αυτά ήταν το χάρτινο κουτί και η μαύρη τσάντα.

Μας ενδιαφέρουν κυρίως δύο χαρακτηριστικά:

- Πρώτον, η συνέπεια στην μορφή του χάρτη, δηλαδή οι κόμβοι στις δύο διελεύσεις να είναι περίπου στα ίδια σημεία δίνοντας την ίδια μορφή στον χάρτη.
- Δεύτερον, ο αριθμός των κόμβων στα δύο πειράματα θα πρέπει να είναι ίδιος ή σχεδόν ίδιος. Εάν ο αριθμός αυτός διέφερε σε μεγάλο βαθμό, αυτό θα υποδήλωνε αστάθεια και έλλειψη ευρωστίας στον αλγόριθμο.



Σχημα 52. Οι χάρτες των δύο ασυσχέτιστων διαδρομών. Με αστερίσκους, οι κόμβοι από την 1η διαδρομή (με χρήση τεχνητών ορόσημων). Με κύκλους, οι κόμβοι της 2ης διαδρομής (χωρίς τεχνητά ορόσημα).

Στο Σχημα 52 φαίνονται τα αποτελέσματα από τα δύο περάσματα: αποτελέσματα ιδιαίτερα ενθαρρυντικά. Το 1<sup>ο</sup> πείραμα δημιούργησε ένα χάρτη 19 κόμβων, ενώ το 2<sup>ο</sup> πείραμα 25 κόμβους. Η διαφορά στον αριθμό των κόμβων οφείλεται κυρίως στην έλλειψη κατά το 2<sup>ο</sup> πείραμα του χάρτινου κουτιού, που υπήρχε στο 1<sup>ο</sup>, κάτι που θα εξηγήσουμε αμέσως παρακάτω.

Στο Σχημα 53, φαίνονται οι 2<sup>οι</sup> κόμβοι των δύο διαδρομών. Είναι αρκετά κοντά, αλλά όχι στο ίδιο σημείο. Το σημαντικό εδώ φαίνεται στο Σχημα 54. Όπως βλέπουμε, η 1<sup>η</sup> διαδρομή έχει στρίψει ήδη από τον 6<sup>ο</sup> κόμβο, ενώ η 2<sup>η</sup> διαδρομή (χωρίς τα ορόσημα), έχει μείνει πιο πίσω.

Η εξέλιξη του 2<sup>ου</sup> πειράματος διαφοροποιείται ως ένα βαθμό λόγω της έλλειψης του κουτιού. Αυτό συμβαίνει επειδή στα επόμενα καρέ του 2<sup>ου</sup> πειράματος η εικόνα που παίρνουμε καθώς πλησιάζουμε τον τοίχο είναι κενή περιεχομένου, αφού φαίνεται μόνο το λευκό χρώμα του τοίχου. Οι επιπτώσεις του φαινομένου αυτού στο 2<sup>ο</sup> πείραμα είναι αρκετά σημαντικές και εξηγούν γιατί υπάρχει αυτή η διαφορά των κόμβων. Μάλιστα στην 1<sup>η</sup> ευθεία της διαδρομής, η δοκιμή με τα ορόσημα εισήγαγε 5 κόμβους, ενώ η δοκιμή χωρίς τα ορόσημα, εισήγαγε 9 δηλαδή σχεδόν τους διπλάσιους. Βλέπουμε δηλαδή ότι το μεγαλύτερο

μέρος της διαφοράς στον αριθμό των κόμβων, εξηγείται απλά και μόνο από το επιπλέον ορόσημο που τοποθετήσαμε στο περιβάλλον.

Το φαινόμενο αυτό είναι ιδιαίτερα έντονο με τη χρήση του ρομπότ, καθώς η κάμερα βρίσκεται κοντά στο πάτωμα και η εικόνα που παίρνουμε είναι κάποιες φορές χωρίς ιδιαίτερη πληροφορία/περιεχόμενο. Αντίθετα, από την οπτική γωνία ενός ανθρώπινου παρατηρητή, το πρόβλημα αυτό δεν είναι τόσο έντονο.

Τα συμπεράσματα από το πείραμα αυτό είναι πολύ σημαντικά.

- Πρώτον, βλέπουμε ότι η προτεινόμενη μεθοδολογία, παρουσιάζει συνέπεια σε παρόμοιες αλλά ασυσχέτιστες μεταξύ τους διαδρομές.
- Δεύτερον, αναδεικνύεται η σημασία και η δύναμη της πλούσιας οπτικής πληροφορίας στην επίδοση του αλγορίθμου. Όσο πιο πλούσιο είναι το περιβάλλον σε οπτική πληροφορία, τόσο καλύτερα θα αποδώσει το μοντέλο παρατήρησης.
- Τρίτον, τα τεχνητά ορόσημα, εάν είναι δυνατόν να τοποθετηθούν, μπορούν να ενισχύσουν σημαντικά την επίδοση του αλγορίθμου παρέχοντας επιπλέον οπτική πληροφορία.



**Σχημα 53. Αριστερά: Ο 2ος κόμβος της διαδρομής με τα ορόσημα. Δεξιά: Ο 2ος κόμβος της διαδρομής χωρίς ορόσημα.**



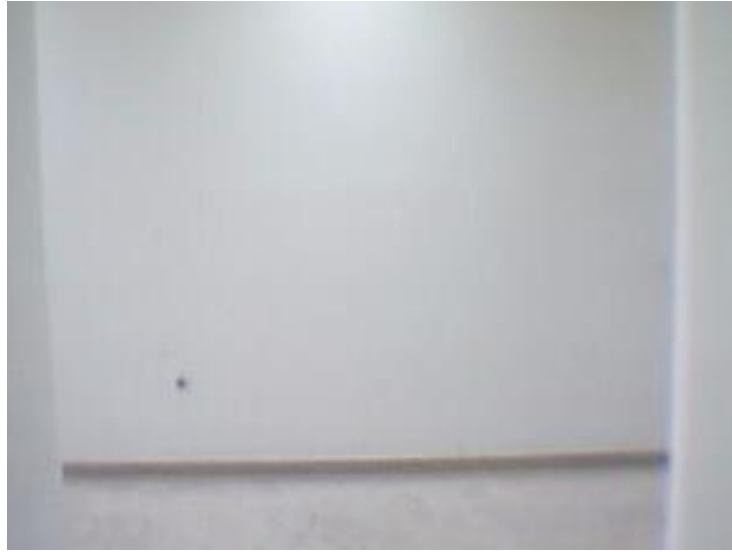
**Σχημα 54. Από πάνω προς τα κάτω: 2ος, 3ος, 4ος, 5ος και 6ος κόμβος.  
Αριστερά: Πέρασμα με ορόσημα. Δεξιά: Πέρασμα χωρίς ορόσημα.**

## 6.8 Προβλήματα

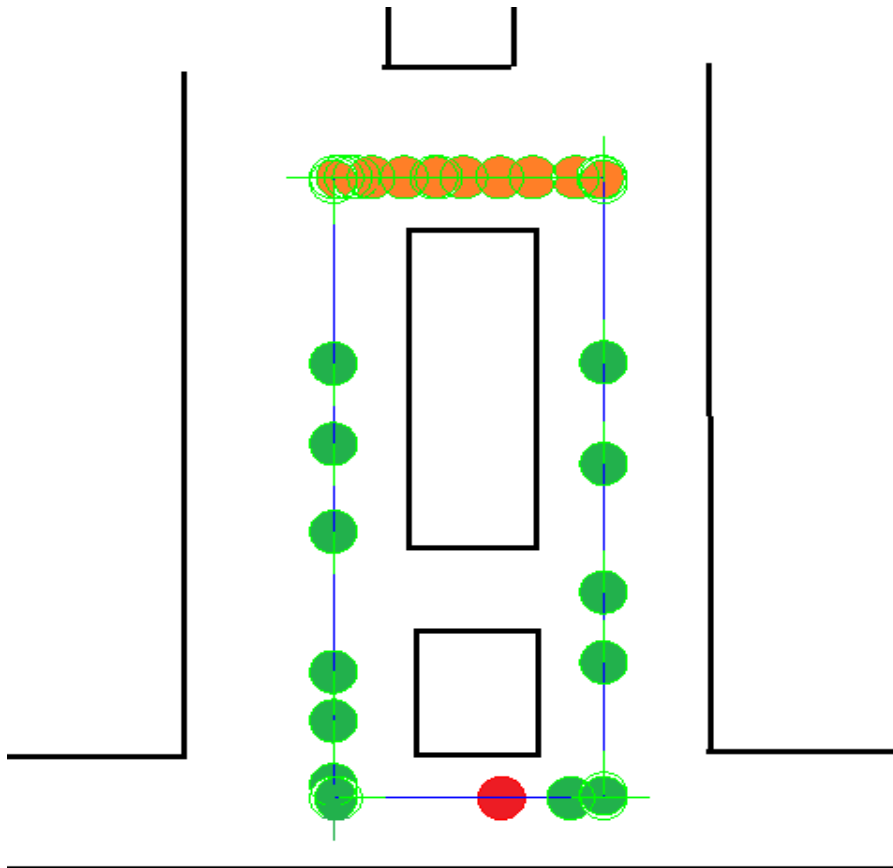
Παρ'όλη την ελπιδοφόρα βελτίωση που επιτυγχάνουν τα χαρακτηριστικά SURF κατά την αναγνώριση και το ταίριασμα των εικόνων, παρουσιάζουν κάποια μειονεκτήματα.

Το βασικότερο είναι ότι ο αλγόριθμος εντοπισμού των χαρακτηριστικών SURF δουλεύει υπό μια κύρια προϋπόθεση: η εικόνα πρέπει να έχει όσο το δυνατόν περισσότερη πληροφορία. Σε διαφορετική περίπτωση, ο αλγόριθμος μας αδυνατεί να εντοπίσει εύλογο (>40) αριθμό σημείων επαρκή για σύγκριση με άλλες εικόνες και σε μια τέτοια περίπτωση ανατρέχουμε στη μέθοδο των ιστογραμμάτων για να συγκρίνουμε τις εικόνες. Όταν συμβεί κάτι τέτοιο, η επίδοση του αλγορίθμου για τα συγκεκριμένα καρέ πέφτει αισθητά και μπορεί κανείς εύκολα να παρατηρήσει την διαφορά στην πυκνότητα των κόμβων σε μια τέτοια περίπτωση. Για να καταστήσουμε σαφές το παραπάνω πρόβλημα, θα παρουσιάσουμε τον χάρτη που προέκυψε από ένα πείραμα χωρίς μετρική πληροφορία από το ρομπότ, καθώς το *video* το τραβήξαμε με μια κάμερα, με το χέρι. Το μοντέλο παρατήρησης βασίζεται και εδώ αποκλειστικά στα χαρακτηριστικά SURF.

Η διαδρομή αυτή, οδηγεί στην δημιουργία ενός χάρτη που φαίνεται στο Σχημα 56. Με κόκκινο συμβολίζουμε τον κόμβο στον οποίο επιστρέψαμε και αναγνώρισε ο αλγόριθμος ως ολοκλήρωση βρόχου. Το χαρακτηριστικό που θέλουμε να παρατηρήσουμε εδώ, είναι οι κόμβοι που έχουν σημειωθεί με καφέ χρώμα. Οι κόμβοι αυτοί αντιστοιχούν σε παρατηρήσεις του τοίχου που φαίνεται στο Σχημα 55. Επειδή ο τοίχος αυτός δεν διαθέτει ιδιαίτερη οπτική πληροφορία, ο αλγόριθμος εντοπισμού χαρακτηριστικών SURF δεν μπόρεσε να εντοπίσει επαρκή αριθμό χαρακτηριστικών σημείων. Σε μια τέτοια περίπτωση, αναγκάζομαστε να χρησιμοποιήσουμε το μοντέλο παρατήρησης των ιστογραμμάτων χρωματικού τόνου και κορεσμού, το οποίο όμως αδυνατεί να αντιληφθεί ότι πρακτικά βρίσκεται στον ίδιο κόμβο. Έτσι εξηγείται η μεγάλη πυκνότητα των καφέ κόμβων που εισάγονται. Να σημειώσουμε ότι επειδή ο χάρτης αυτός δημιουργήθηκε χωρίς το μοντέλο κίνησης του ρομπότ, αλλά μόνο με χρήση οπτικής οδομετρίας, η μετρική πληροφορία της απόστασης προέρχεται μόνο από τον αριθμό των καρέ που συμμετέχουν στην ευθύγραμμη κίνηση όπως εξηγήσαμε στην Ενότητα 5.2.1.1.2.



Σχημα 55. Πολλαπλά στιγμιότυπα του τοίχου αυτού, δεν μπόρεσαν να παράγουν χαρακτηριστικά SURF.



Σχημα 56. Ο χάρτης, όπως δημιουργήθηκε από τον αλγόριθμο SLAM. Με κόκκινο συμβολίζεται ο κόμβος που ολοκληρώθηκε ο βρόχος. Με καφέ, οι κόμβοι όπου τα χαρακτηριστικά SURF δεν δούλεψαν.

Επιπρόσθετα, στην παρούσα μορφή του, ο αλγόριθμος υποστηρίζει αποτελεσματικά την χαρτογράφηση και τον αυτοεντοπισμό θέσης στο 1<sup>ο</sup> πέρασμα από μια περιοχή, αλλά εμφανίζει αδυναμίες σε περίπτωση που το ρομπότ περάσει από ήδη χαρτογραφημένη περιοχή. Αυτό έχει σαν αποτέλεσμα την προσθήκη νέων κόμβων σε περιοχές που αντιπροσωπεύουν κάποιοι ήδη υπάρχοντες. Σε μια τέτοια περίπτωση μπορεί να δημιουργηθεί μια αναντιστοιχία του χάρτη με την πραγματική δομή του χώρου εργασίας. Το πρόβλημα αυτό είναι αισθητά μειωμένο όταν κανείς χρησιμοποιεί τα χαρακτηριστικά SURF λόγω της ευρωστίας της μεθόδου, όμως στην παρούσα υλοποίηση, δεν μπορέσαμε να αντιμετωπίσουμε με επιτυχία το πρόβλημα που προκύπτει ορισμένες φορές.

Ένα ακόμα πρόβλημα των χαρακτηριστικών SURF είναι ο χρόνος επεξεργασίας τους στην παρούσα υλοποίηση. Η παρούσα μορφή της υλοποίησης δεν ενδύκνεται για online επεξεργασία, για αυτό και τα περισσότερα πειράματα έγιναν offline. Ο σχετικά αργός χρόνος επεξεργασίας, οφείλεται κυρίως στην χρήση του Matlab το οποίο δεν ενδείκνυται για τέτοιες εφαρμογές πραγματικού χρόνου.

Τέλος, ένα σημαντικό μειονέκτημα του αλγορίθμου, είναι ότι το σύστημα δεν μπορεί ανακάμψει σε περίπτωση λανθασμένης αναγνώρισης κόμβου (εάν δηλαδή αναγνωρίσει λανθασμένα ένα βρόχο). Σε μια τέτοια περίπτωση η διόρθωση ολόκληρου του χάρτη από μελλοντικές σωστές αναγνώρισεις δεν μπορεί να επιτευχθεί. Για τον λόγο αυτό, η συνέργεια του μοντέλου κίνησης (και κατ'επέκταση του βήματος πρόβλεψης) με το μοντέλο παρατήρησης (και κατ'επέκταση το βήμα ανανέωσης) πρέπει να εξασφαλίσουμε ότι είναι εξαιρετικά ισχυρή. Το τελευταίο, θεωρούμε ότι σε μεγάλο βαθμό η εργασία αυτή το επιτυγχάνει.

## 6.9 Σύνοψη Συμπερασμάτων Αξιολόγησης

Συνοψίζοντας τα αποτελέσματα που προέκυψαν από την πειραματική επαλήθευση της προτεινόμενης μεθοδολογίας, συμπεραίνεται ότι ο αλγόριθμος που υλοποιήθηκε:

1. Λύνει το πρόβλημα του SLAM σε δομημένα εσωτερικά περιβάλλοντα, με κάποιες παραδοχές στην κίνηση του ρομπότ.
2. Είναι πολύ αποτελεσματικός στο να καθορίζει τις θέσεις των νέων κόμβων, ώστε οι ποιοτικές διαφορές που θα έχουν, να αντικατοπτρίζουν τα οπτικά χαρακτηριστικά των περιοχών που αντιπροσωπεύουν.
3. Αντιμετωπίζει με επιτυχία το πρόβλημα του correspondence, δηλαδή της δυνατότητας να διακρίνει πότε όμοιες εικόνες αντιστοιχούν σε διαφορετικά σημεία του χάρτη, χάρη στη συνέργεια των βημάτων πρόβλεψης και παρατήρησης.
4. Αναγνωρίζει με πολύ μεγάλη επιτυχία την ολοκλήρωση βρόχων σε διαδρομές διαφορετικού μήκους, και μπορεί να διορθώνει αποτελεσματικά τους κόμβους του χάρτη, τόσο την θέση τους, όσο και την αβεβαιότητα αυτής.
5. Παρουσιάζει συνέπεια μεταξύ ασυχέτιστων μεταξύ τους διαδρομών, ακόμα με αλλαγές στο περιβάλλον.

6. Μπορεί να διεκπεραιώσει αποτελεσματικά όλες τις απαιτούμενες διεργασίες για την επίλυση του SLAM, βασιζόμενος σε μια απλή, φθηνή, μονοσκοπική κάμερα, και στην οδομετρία του ρομπότ (που είναι άμεσα και εύκολα διαθέσιμη).

Με βάση τα παραπάνω, και παρά τις όποιες αδυναμίες ή περιορισμούς που εισήχθησαν, η παρούσα μεθοδολογία, αποδεικνύεται κατάλληλη για την επίλυση του SLAM με ένα αυτόνομο, κινούμενο ρομποτικό όχημα, το οποίο είναι εξοπλισμένο με μια απλή κάμερα και κινείται σε εσωτερικό, κτιριακό περιβάλλον.



Ολοκληρώνοντας την παρούσα διπλωματική εργασία, θα κάνουμε μια ανακεφαλαίωση της μεθόδου που αναπτύξαμε και θα αναφερθούμε στα συμπεράσματα που προκύπτουν. Ακόμα, θα προτείνουμε πιθανές βελτιώσεις και επεκτάσεις της μεθόδου.

## 7.1 Ανακεφαλαίωση και Συμπεράσματα

Όπως αναφέραμε και στην εισαγωγή αυτής της εργασίας, στόχος μας ήταν η ανάπτυξη και η πειραματική αξιολόγηση μιας μεθοδολογίας τοπολογικού SLAM, χρησιμοποιώντας πρωτίστως οπτική πληροφορία.

Επεκτείναμε την προϋπάρχουσα μεθοδολογία χρησιμοποιώντας χαρακτηριστικά SURF και εφαρμόζοντας μια απλή τεχνική διαστάρωσης χαρακτηριστικών μεταξύ των εικόνων. Διατηρώντας τις παραδοχές της οπτικής οδομετρίας για την κίνηση του ρομπότ, εμπλουτίσαμε την εισερχόμενη πληροφορία, παίρνοντας και μετρική πληροφορία από τους κωδικοποιητές του ρομπότ στους τροχούς. Επιπροσθέτως, δημιουργήσαμε αποτελεσματικό μοντέλο κίνησης το οποίο σέβεται τις παραδοχές κίνησης και εισάγει αβεβαιότητα στην θέση του ρομπότ με την μορφή μιας κατανομής Gauss. Το γεγονός αυτό έδωσε την δυνατότητα να εμπλουτίσουμε τον χάρτη μας και να κάνουμε πολύ καλύτερη αναζήτηση στο χώρο του χάρτη μεταβάλλοντας το εύρος της Γκαουσιανής κατανομής ανάλογα με την αβεβαιότητα θέσης.

Τέλος, διεξάγαμε πειράματα με στόχο να αξιολογήσουμε την επίδοση της νέας μεθόδου, και ιδιαίτερα του νέου μοντέλο παρατήρησης σε συνδυασμό με το μοντέλο κίνησης. Όπως προκύπτει αναλύοντας τα πειραματικά δεδομένα, η μετρική πληροφορία ενισχύει σε μεγάλο βαθμό την αποτελεσματικότητα της μεθόδου, ενώ τα χαρακτηριστικά SURF της δίνουν σταθερότητα και ευρωστία.

## 7.2 Μελλοντικές Επεκτάσεις

Τα θετικά αποτελέσματα από τα πειράματα και τις δοκιμές που διεξήχθησαν, παρέχουν θετική ένδειξη ότι η μεθοδολογία που έχει αναπτυχθεί μπορεί να χρησιμοποιηθεί με επιτυχία στην επίλυση του Οπτικού SLAM. Υπάρχουν όμως μερικές βελτιώσεις που, κατά την γνώμη μας, θα βελτίωναν σημαντικά την επίδοση της μεθόδου.

Η πρώτη βελτίωση θα ήταν η άρση των περιορισμών που προκύπτουν από την οπτική οδομετρία. Αυτό μπορεί να γίνει εν γένει με δύο τρόπους: ο 1<sup>ος</sup>, είναι η υιοθέτηση ενός συστήματος οπτικής οδομετρίας όπως περιγράφεται στο [26], η οποία εν γένει επιτρέπει τυχαίο προσανατολισμό. Αυτή, σε συνδυασμό με μετρική πληροφορία από τους

κωδικοποιητές του ρομπότ, θα μπορούσε να δώσει πολύ καλά αποτελέσματα, χωρίς ενδεχομένως την παραδοχή των τεσσάρων διευθύνσεων. Ο 2<sup>ος</sup> τρόπος, είναι η υιοθέτηση της μεθόδου του Feature Tracking μεταξύ διαδοχικών καρέ. Η μέθοδος αυτή είναι αρκετά διαδεδομένη, και ουσιαστικά εντοπίζει την κίνηση της κάμερας μεταξύ των καρέ, παρακολουθώντας την μεταβολή στη θέση διαφόρων χαρακτηριστικών.

Η δεύτερη σημαντική βελτίωση σχετίζεται με το μοντέλο παρατήρησης. Δείξαμε πόσο πολύ πιο σταθερά και εύρωστα είναι τα χαρακτηριστικά SURF σε σχέση με τα ιστογράμματα HSI. Παρ'όλα αυτά, το μοντέλο παρατήρησης που φτιάξαμε, και μεν αποδίδει στο περιβάλλον εργασίας μας, και με τις παραδοχές κίνησης που κάναμε, είναι όμως αρκετά απλοϊκό και μπορεί να υποπέσει σε λάθη υπό διαφορετικές συνθήκες εργασίας. Επίσης, είναι αρκετά αργό. Αυτό μπορεί να διορθωθεί χρησιμοποιώντας κάποιες πιθανόντικες μεθόδους ταύτισης χαρακτηριστικών SURF, όπως για παράδειγμα ο αλγόριθμος RANSAC. Πιστεύουμε πάντως ότι η υιοθέτηση χαρακτηριστικών SURF είναι ένα βήμα προς την σωστή κατεύθυνση, το οποίο επιδέχεται βελτίωσης.

Επιπλέον, σημαντική θα είναι η εξασφάλιση ότι ο αλγόριθμος μπορεί να λειτουργήσει απολύτως σωστά κάτω από πολλαπλά περάσματα από τον ίδιο χάρτη. Θεωρούμε ότι με την ευρωστία της νέας μεθόδου κάναμε ένα σημαντικό βήμα, αλλά απομένει ακόμα σημαντική ερευνητική εργασία στον τομέα αυτό.

Ακόμα, ενδιαφέρουσα θα ήταν και η χρήση πληροφορίας από άλλους αισθητήρες, όπως laser ή IR. Η σύνθεση της επιπλέον πληροφορίας ενδεχομένως να βελτιώνει σημαντικά την επίδοση της μεθόδου. Επιπρόσθετα, θα εμπλούτιζε την πληροφορία του χάρτη, δείχνοντας για παράδειγμα την γεωμετρία του περιβάλλοντος, απεικονίζοντας εμπόδια, τοίχους και άλλα χαρακτηριστικά. Προφανώς, ένας τέτοιος χάρτης θα ήταν πιο πλήρης και κατατοπιστικός, αναπαριστώντας καλύτερα το περιβάλλον του ρομπότ. Επιπλέον, χρήση τέτοιων αισθητήρων απόστασης, θα έδινε την δυνατότητα για την δημιουργία υβριδικών χαρτών πάνω στις ιδέες του [15]. Ο συνδυασμός ενός κοσμοκεντρικού χάρτη (δηλαδή σε «παγκόσμιες» συντεταγμένες), με ένα ρομποκεντρικό χάρτη, δηλαδή ένα χάρτη βασισμένο στα νούμερα των παρατηρήσεων και των αισθητήρων, θα απέφερε σημαντικά αποτελέσματα. Από την μία θα είχε κανείς την οπτική γωνία ενός ουδέτερου, μακρινού παρατηρητή, χάρη στον κοσμοκεντρικό χάρτη και από την άλλη θα είχε την οπτική γωνία του ίδιου του ρομπότ μέσω του ρομποκεντρικού χάρτη. Σαφώς η πολυπλοκότητα αυξάνεται, αλλά ανοίγονται νέες δυνατότητες από τη στιγμή που μπορούμε να συνδυάσουμε την πληροφορία από τους δύο χάρτες και να εξάγουμε συμπεράσματα για την θέση του ρομπότ αλλά και για τους ίδιους τους χάρτες.

Τέλος, ενδεχόμενη αλλαγή στο είδος της κάμερας να βελτιώνει σε μεγάλο βαθμό την επίδοση της μεθόδου. Σε πολλές εργασίες, ειδικά σε αυτές που αξιοποιούν χαρακτηριστικά σημεία της εικόνας, όπως τα SURF, οι πανοραμικές κάμερες είναι ιδιαίτερα αποτελεσματικές γιατί προσφέρουν πλουσιότερη οπτική πληροφορία, δεδομένου ότι λαμβάνουν εικόνα από ψηλά.

- [1] A. Elfes, "A sonar-based mapping and navigation system," *Proceedings IEEE International Conference on, Robotics and Automation*, pp. 1151-1156, 1986.
- [2] I. Ulrich and I. Nourbakhsh, "Appearance-based place recognition for topological localization," *IEEE International Conference on Robotics and Automation*, pp. 1023-1029 vol.2, 2000.
- [3] A. Ranganathan, E. Menegatti, and F. Dellaert, "Bayesian inference in the space of topological maps," *Robotics, IEEE Transactions on Robotics and Automation*, vol. 22, pp. 92-107, 2006.
- [4] G. Lakemeyer and B. Nebel, *Exploring artificial intelligence in the new millennium*, Morgan Kaufmann, 2003.
- [5] P. Jensfelt and S. Kristensen, "Active global localization for a mobile robot using multiple hypothesis tracking," *IEEE Transactions on Robotics and Automation*, vol. 17, pp. 748-760, 2001.
- [6] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," *IEEE International Conference on Robotics and Automation*, pp. 1322-1328 vol.2, 1999.
- [7] Á.S. Miralles and M.Á.S. Bobi, "A Neural-based Model for Fast Continuous and Global Robot Location," *Journal of Intelligent and Robotic Systems*, vol. 46, pp. 221-243, 2006.
- [8] P. Nunez, R. R. Vazquez-Martin, A. Bandera, and F. Sandoval, "Fast Laser Scan Matching Approach Based on Adaptive Curvature Estimation for Mobile Robots," *Robotica*, vol. 27, pp. 469-479, 2009.
- [9] P. Lamon, I. Nourbakhsh, B. Jensen, and R. Siegwart, "Deriving and matching image fingerprint sequences for mobile robot localization," *IEEE International Conference on Robotics and Automation*, pp. 1609-1614 vol.2, 2001.
- [10] L. A Cornelissen and F. C. A. Groen, "Automatic color landmark detection and retrieval for robot navigation," Feb. 2008.
- [11] S. Se, D. Lowe, και J. Little, Συγγρ., 'Local and global localization for mobile robots using visual landmarks', in *Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No.01CH37180)*, Maui, HI, USA, σσ. 414-420.
- [12] P. Lamon, I. Nourbakhsh, B. Jensen, και R. Siegwart, Συγγρ., 'Deriving and matching image fingerprint sequences for mobile robot localization', in *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No.01CH37164)*, Seoul, South Korea, σσ. 1609-1614.
- [13] F. Dayoub and T. Duckett, "An adaptive appearance-based map for long-term topological localization of mobile robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2008.*, pp. 3364-3369, 2008.
- [14] X.Z. Zhang, A.B. Rad, Y.K. Wong, G.Q. Huang, Y.L. Ip, and K.M. Chow, "A Comparative Study of Three Mapping Methodologies," *Journal of Intelligent and Robotic Systems*, vol. 49, pp. 385-395, 2007.

- [15] Sunghwan Ahn, Wan Kyun Chung, και Sang-Rok Oh, Συγγρ., ‘Construction of hybrid visual map for indoor SLAM’, in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, San Diego, CA, USA, pp. 1695-1701, 2007.
- [16] S. Thrun: “Robotic Mapping: A Survey”, Book: Exploring artificial intelligence in the new millennium, ISBN:1-55860-811-7, pp.1-35 February 2002
- [17] Greg Grudic και Jane Mulligan, Συγγρ., ‘Topological Mapping with Multiple Visual Manifolds’. 2005.
- [18] L. F. Rojo, L. Paya, O. Reinoso, J. M. Marin, και A. Gil, Συγγρ., ‘Comparison of mapping techniques in appearance-based topological maps creation’, *IEEE Conference on Emerging Technologies and Factory Automation (ETFA)*, pp. 1-7, 2010.
- [19] R. Smith, M. Self, και P. Cheeseman, Συγγρ., ‘Estimating uncertain spatial relationships in robotics’, in *Robotics and Automation. Proceedings. 1987 IEEE International Conference on*, 1987, τ. 4, σ. 850.
- [20] J. J. Leonard και H. F. Durrant-Whyte, Συγγρ., ‘Simultaneous map building and localization for an autonomous mobile robot’, *Proceedings IROS '91. IEEE/RSJ International Workshop on Intelligent Robots and Systems '91*, , pp. 1442-1447 vol.3, 1991.
- [21] M. Montemerlo και S. Thrun, Συγγρ., ‘Simultaneous localization and mapping with unknown data association using FastSLAM’, *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, vol. 2, pp. 1985-1991 vol.2, 2003.
- [22] Michael Montemerlo, Συγγρ., ‘FastSLAM: A factored solution to the simultaneous localization and mapping problem’. AAI, 2002.
- [23] S. Thrun, Συγγρ., ‘Robotic mapping: a survey’, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., pp. 1–35, 2003.
- [24] F. Lu και E. Milios, Συγγρ., ‘Globally Consistent Range Scan Alignment for Environment Mapping’, *Autonomous Robots*, τ. 4, τχ. 4, σσ. 333-349, Οκτωβρίου. 1997.
- [25] A. Dempster, N. Laird, και D. Rubin, Συγγρ., ‘Maximum likelihood from incomplete data via the {EM} algorithm (with discussion)’, *Journal of the Royal Statistical Society B*, τ. 39, σσ. 1-38, 1977.
- [26] Michael Milford και Gordon Wyeth, Συγγρ., ‘Featureless Vehicle-Based Visual SLAM with a Consumer Camera’. 2008.
- [27] Zhanyu Zhang, Yalou Huang, Chao Li, και Yewei Kang, Συγγρ., ‘Monocular vision simultaneous localization and mapping using SURF’, in *2008 7th World Congress on Intelligent Control and Automation*, Chongqing, China, 2008, σσ. 1651-1656.
- [28] Carlo Tomasi Takeo και Takeo Kanade, Συγγρ., ‘Detection and Tracking of Point Features’. 1991.
- [29] Sunghwan Ahn και Wan Kyun Chung, Συγγρ., ‘Efficient SLAM algorithm with hybrid visual map in an indoor environment’, in *2007 International Conference on Control, Automation and Systems*, Seoul, South Korea, 2007, σσ. 663-667.
- [30] Yong-Ju Lee και Jae-Bok Song, Συγγρ., ‘Visual SLAM in indoor environments using autonomous detection and registration of objects’, in *2008 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Seoul, 2008, σσ. 671-676.
- [31] B. Zitov και J. Flusser, Συγγρ., ‘Image registration methods: a survey’, *Image and Vision Computing*, τ. 21, τχ. 11, σσ. 977-1000, Οκτωβρίου. 2003.

- [32] D. N. Bhat και S. K. Nayar, Συγγρ., ‘Ordinal measures for image correspondence’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 20, τχ. 4, σσ. 415-423, Απριλίου. 1998.
- [33] R. Berthilsson, Συγγρ., ‘Affine correlation’, in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, Brisbane, Qld., Australia, σσ. 1458-1460.
- [34] A. Simper, Συγγρ., ‘Correcting general band-to-band misregistrations’, in *Proceedings of 3rd IEEE International Conference on Image Processing*, Lausanne, Switzerland, σσ. 597-600.
- [35] D. P. Huttenlocher, G. A. Klanderman, και W. J. Rucklidge, Συγγρ., ‘Comparing images using the Hausdorff distance’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 15, τχ. 9, σσ. 850-863, Σεπτεμβρίου. 1993.
- [36] E. De Castro και C. Morandi, Συγγρ., ‘Registration of Translated and Rotated Images Using Finite Fourier Transforms’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 9, τχ. 5, σσ. 700-703, Σεπτεμβρίου. 1987.
- [37] T. M. Lehmann, Συγγρ., ‘A two-stage algorithm for model-based registration of medical images’, in *Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No.98EX170)*, Brisbane, Qld., Australia, σσ. 344-351.
- [38] Qin-Sheng Chen, M. Defrise, και F. Deconinck, Συγγρ., ‘Symmetric phase-only matched filtering of Fourier-Mellin transforms for image registration and recognition’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 16, τχ. 12, σσ. 1156-1168, Δεκεμβρίου. 1994.
- [39] A. Sadri και A. Beheshti, Συγγρ., ‘Image registration based on physical forces’, in *2008 15th International Conference on Systems, Signals and Image Processing*, Bratislava, Slovakia, 2008, σσ. 201-204.
- [40] G. P. Penney, J. Weese, J. A. Little, P. Desmedt, D. L. G. Hill, και D. J. Hawkes, Συγγρ., ‘A comparison of similarity measures for use in 2-D-3-D medical image registration’, *IEEE Transactions on Medical Imaging*, τ. 17, τχ. 4, σσ. 586-595, Αυγούστου. 1998.
- [41] P. Viola και W. M. Wells, Συγγρ., ‘Alignment by maximization of mutual information’, in *Proceedings of IEEE International Conference on Computer Vision*, Cambridge, MA, USA, σσ. 16-23.
- [42] H. S. Sawhney και R. Kumar, Συγγρ., ‘True multi-image alignment and its application to mosaicing and lens distortion correction’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 21, τχ. 3, σσ. 235-243, Μαρτίου. 1999.
- [43] J. P. Pascual Starink και E. Backer, Συγγρ., ‘Finding point correspondences using simulated annealing’, *Pattern Recognition*, τ. 28, τχ. 2, σσ. 231-240, Φεβρουαρίου. 1995.
- [44] N. R. Pal και S. K. Pal, Συγγρ., ‘A review on image segmentation techniques’, *Pattern Recognition*, τ. 26, τχ. 9, σσ. 1277-1294, Σεπτεμβρίου. 1993.
- [45] A. Goshtasby, G. Stockman, και C. Page, Συγγρ., ‘A Region-Based Approach to Digital Image Registration with Subpixel Accuracy’, *IEEE Transactions on Geoscience and Remote Sensing*, τ. 24, τχ. 3, σσ. 390-399, Μαΐου. 1986.
- [46] T. Tuytelaars και L. Van Gool, Συγγρ., ‘Matching Widely Separated Views Based on Affine Invariant Regions’, *International Journal of Computer Vision*, τ. 59, τχ. 1, σσ. 61-85, Αυγούστου. 2004.

- [47] J. Canny, Συγγρ., ‘A Computational Approach to Edge Detection’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 8, τχ. 6, σσ. 679-698, Νοεμβρίου. 1986.
- [48] D. Marr και E. Hildreth, Συγγρ., ‘Theory of Edge Detection’, *Proceedings of the Royal Society B: Biological Sciences*, τ. 207, τχ. 1167, σσ. 187-217, Φεβρουαρίου. 1980.
- [49] G. Stockman, S. Kopstein, και S. Benett, Συγγρ., ‘Matching Images to Models for Registration and Object Detection via Clustering’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, τ. 4, τχ. 3, σσ. 229-241, Μαΐου. 1982.
- [50] M. Roux, Συγγρ., ‘Automatic registration of SPOT images and digitized maps’, in *Proceedings of 3rd IEEE International Conference on Image Processing*, Lausanne, Switzerland, σσ. 625-628.
- [51] S. Grove και R. Tonjes, Συγγρ., ‘A knowledge based approach to automatic image registration’, in *Proceedings of International Conference on Image Processing*, Santa Barbara, CA, USA, σσ. 228-231.
- [52] Weng Muyun και He Mingyi, Συγγρ., ‘Image Feature Detection and Matching Based on SUSAN Method’, in *First International Conference on Innovative Computing, Information and Control - Volume I (ICICIC'06)*, Beijing, China, σσ. 322-325.
- [53] Z. Zheng, H. Wang, και E. Khwang Teoh, Συγγρ., ‘Analysis of gray level corner detection’, *Pattern Recognition Letters*, τ. 20, τχ. 2, σσ. 149-162, Φεβρουαρίου. 1999.
- [54] K. Mikolajczyk και C. Schmid, Συγγρ., ‘Indexing based on scale invariant interest points’, in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, Vancouver, BC, Canada, σσ. 525-531.
- [55] ‘Scientific Commons: Detection and Tracking of Point Features (1991), 1991 [Carlo Tomasi Takeo, Takeo Kanade]’. [Online]. Available: <http://en.scientificcommons.org/42984610>. [Accessed: 31-Μαρτίου-2011].
- [56] D. G. Lowe, Συγγρ., ‘Object recognition from local scale-invariant features’, in *Proceedings of the Seventh IEEE International Conference on Computer Vision*, Kerkyra, Greece, 1999, σσ. 1150-1157 vol.2.
- [57] H. Bay, T. Tuytelaars, και L. Gool, Συγγρ., ‘SURF: Speeded Up Robust Features’, in *Computer Vision – ECCV 2006*, τ. 3951, A. Leonardis, H. Bischof, και A. Pinz, Επιμ. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, σσ. 404-417.
- [58] I. Ulrich και I. Nourbakhsh, Συγγρ., ‘Appearance-based place recognition for topological localization’, in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, San Francisco, CA, USA, σσ. 1023-1029.
- [59] E. Remolina και B. Kuipers, Συγγρ., ‘Towards a general theory of topological maps’, *Artificial Intelligence*, τ. 152, τχ. 1, σσ. 47-104, 2004.
- [60] A. I. Eliazar και R. Parr, Συγγρ., ‘Learning probabilistic motion models for mobile robots’, in *Twenty-first international conference on Machine learning - ICML '04*, Banff, Alberta, Canada, 2004, σ. 32.
- [61] D. G. Lowe, Συγγρ., ‘Distinctive Image Features from Scale-Invariant Keypoints’, *International Journal of Computer Vision*, τ. 60, τχ. 2, σσ. 91-110, Νοεμβρίου. 2004.

- [62] Yin-Tien Wang, Duan-Yan Hung, και Sheng-Hsien Cheng, Συγγρ., 'Image feature tracker for SLAM with monocular vision', in *SICE Annual Conference 2010, Proceedings of*, 2010, σσ. 2300-2307.
- [63] Michael Milford και Gordon Wyeth, Συγγρ., 'Featureless Vehicle-Based Visual SLAM with a Consumer Camera'. 2008.

## ΠΑΡΑΡΤΗΜΑ

### *Τεχνικές Λεπτομέρειες Υλοποίησης*

Στο παρόν Παράρτημα παρουσιάζονται συνοπτικά οι τεχνικές λεπτομέρειες της υλοποίησης, ο υλικοτεχνικός εξοπλισμός και το λογισμικό που χρησιμοποιήθηκε.

#### ❖ Υλικό

Για τις ανάγκες της παρούσας διπλωματικής εργασίας χρησιμοποιήθηκε φορητός, προσωπικός υπολογιστή ASUS G53J με τα εξής χαρακτηριστικά:

- Επεξεργαστής: Intel Core i7-740QM, 1.73GHz
- Μνήμη (RAM): 4 GB
- Λειτουργικό Σύστημα: Windows 7 Home Premium, 64 bit

Για την λήψη της οπτικής πληροφορίας, χρησιμοποιήθηκε η κάμερα Lenovo Q350 USB PC Camera.

Το αυτόνομο, κινούμενο ρομποτικό όχημα που χρησιμοποιήθηκε ήταν το Pioneer 3-DX της ActiveMedia Robotics, που μου παραχώρησε το Εργαστήριο Ρομποτικής της Σχολής Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών.

Η διασύνδεση μας με τον υπολογιστή για την έκδοση εντολών προς το ρομπότ, έγινε με τη χρήση ενός κινητού Nokia 3500 classic, μέσω bluetooth.

#### ❖ Λογισμικό

Για την συγγραφή της εργασίας αυτής, χρησιμοποιήθηκε το πρόγραμμα επεξεργασίας κειμένου Microsoft Office Word 2010.

Ο σχεδιασμός και η υλοποίηση της μεθοδολογίας που αναπτύξαμε έγινε σε περιβάλλον Matlab 2010B. Ομοίως και για τα πειράματα και την αξιολόγηση.