



Εθνικό Μετσόβιο Πολυτεχνείο

Διπλωματική Εργασία

---

Μέτρηση γωνίας περιστροφής άξονα  
βιομηχανικού ρομποτικού βραχίονα  
με χρήση καμερών χαμηλού κόστους

---

Συγγραφέας:  
Α. Καραγιαννίδης

Επιβλέπων:  
Δρ. Γ.Χ Βοσνιάκος

Οκτώβριος 2012

Αφιερώνεται στην  
οικογένεια μου

## Ευχαριστίες

Η παρούσα διπλωματική εργασία εκπονείται ως μέρος των υποχρεώσεων μου για την λήψη του διπλώματος του Μηχανολόγου Μηχανικού από το Εθνικό Μετσόβιο Πολυτεχνείο. Στο σημείο αυτό νιώθω την ανάγκη να εκφράσω τις θερμές μου ευχαριστίες στον αναπληρωτή καθηγητή Δρ. Χριστόφορο Βοσνιάκο για την συμπαράσταση και την βοήθεια του στην επίλυση των προβλημάτων που συναντήθηκαν κατά την εκπόνηση της εργασίας μου. Οι γνώσεις και η εμπειρία του ήταν πολύτιμα βοηθήματα στην προσπάθειά μου.

Επίσης θα ήθελα να ευχαριστήσω τα παιδιά του τομέα κατεργασιών και ιδιαίτερα τον Νίκο Κοντολάτη για το ευχάριστο κλίμα, τις εποικοδομητικές συζητήσεις αλλά και την βοήθεια που μου προσέφεραν όλον αυτό τον καιρό. Επιπλέον ευχαριστώ τους φίλους μου για όλες αυτές τις ωραίες στιγμές που μοιραστήχαμε στα όμορφα χρόνια του πολυτεχνείου. Ιδιαίτερες ευχαριστίες πηγαινούν στον φίλο και συνάδελφο Αντώνη Αγγελίδη με τον οποίο εκπονήσαμε μαζί τις διπλωματικές μας εργασίες.

Τέλος θα ήθελα να κάνω ιδιαίτερη αναφορά σε κάποια κοντινά μου πρόσωπα. Στην Ζωή η οποία μου συμπαραστάθηκε όλα αυτά τα χρόνια αλλά και πάνω απ' όλα στους γονείς μου στους οποίους και αφιερώνεται αυτή η εργασία.

Καραγιαννίδης Άρης  
27 Σεπτεμβρίου 2012

# Περιεχόμενα

<b>1</b>	<b>Εισαγωγή</b>	<b>5</b>
1.1	Διατύπωση του προβλήματος . . . . .	5
1.2	Σημασία επίλυσης και πιθανές εφαρμογές . . . . .	6
1.3	Προηγούμενη Σχετική Έρευνα . . . . .	6
<b>2</b>	<b>Στοιχεία Τεχνητής Όρασης</b>	<b>9</b>
2.1	Βασικές έννοιες . . . . .	9
2.1.1	Αναπαράσταση εικόνας και εικονοστοιχεία . . . . .	9
2.1.2	Κάμερες . . . . .	10
2.1.3	Μοντέλο Pinhole . . . . .	10
2.2	Οπτική . . . . .	11
2.2.1	Φακοί - Τύποι φακών . . . . .	12
2.2.2	Σφάλματα και παραμορφώσεις φακών . . . . .	13
2.3	Αισθητήρες φωτογραφίας . . . . .	14
2.3.1	Αισθητήρες CCD . . . . .	14
2.3.2	Αισθητήρες CMOS . . . . .	14
2.4	Αναπαράσταση Χρωμάτων - Colorspaces . . . . .	14
2.4.1	Ανάλυση χρώματος - Ακρίβεια εικόνας . . . . .	17
2.4.2	Το μοντέλο RGB . . . . .	18
2.4.3	Τα μοντέλα HSV και HSL . . . . .	18
2.5	Επεξεργασία φωτογραφίας και φίλτρα . . . . .	25
2.5.1	Αποπαραμόρφωση εικόνας . . . . .	25
2.5.2	Συνέλιξη . . . . .	29
2.5.3	Θόλωση εικόνας - Blurring . . . . .	29
2.5.4	Μορφολογία εικόνας . . . . .	32
2.5.5	Διαχωρισμός - Threshold . . . . .	32
2.5.6	Παράγωγοι εικόνας - Εύρεση ακμών . . . . .	32
2.5.7	Μετασχηματισμοί Hough . . . . .	37
<b>3</b>	<b>Στοιχεία Ρομποτικής</b>	<b>42</b>
3.1	Τύποι Αρθρώσεων . . . . .	42
3.2	Ομογενείς μετασχηματισμοί . . . . .	42
3.2.1	Συμβάσεις στροφών . . . . .	44
3.3	Παράμετροι Denavit–Hartenberg . . . . .	44
3.3.1	Ενδιάμεσα και ακραία μέλη αλυσίδας . . . . .	45
3.3.2	Μετασχηματισμός Μέλους . . . . .	45

<b>4</b>	<b>Διατύπωση και επίλυση του προβλήματος</b>	<b>47</b>
4.1	Εξέταση του προβλήματος . . . . .	47
4.2	Προδιαγραφές . . . . .	48
4.3	Γενικός τρόπος επίλυσης . . . . .	48
4.4	Πρώτο μοντέλο . . . . .	48
4.4.1	Εύρεση χάρτη . . . . .	48
4.4.2	Αποτυχία μοντέλου . . . . .	49
4.5	Δεύτερο Μοντέλο . . . . .	49
4.5.1	Εύρεση χάρτη . . . . .	49
4.5.2	Αποτυχία δεύτερου μοντέλου . . . . .	55
4.6	Τρίτο Μοντέλο . . . . .	55
4.7	Πραγματικός Χάρτης . . . . .	58
4.7.1	Ερμηνεία χάρτη . . . . .	60
<b>5</b>	<b>Υλοποίηση</b>	<b>61</b>
5.1	Επιλογή εργαλείων . . . . .	61
5.2	Αρχιτεκτονική της επίλυσης . . . . .	62
5.2.1	Κύρια ρουτίνα . . . . .	62
5.2.2	Κοινές υπορουτίνες . . . . .	64
5.2.3	Κλάση Χάρτη . . . . .	65
5.3	Διάταξη . . . . .	65
5.4	Λοιπές Λεπτομέρειες Υλοποίησης . . . . .	69
5.5	Εργαλείο λήψης μετρήσεων . . . . .	69
<b>6</b>	<b>Αποτελέσματα</b>	<b>70</b>
6.1	Μετρήσεις . . . . .	70
6.1.1	Εξοπλισμός μετρήσεων . . . . .	70
6.1.2	Διαδικασία μετρήσεων . . . . .	70
6.2	Αποτελέσματα . . . . .	74
6.3	Σφάλματα . . . . .	79
6.3.1	Φωτισμός . . . . .	79
6.3.2	Παραμορφώσεις στόχου . . . . .	80
6.3.3	Απόκλιση θέσης . . . . .	80
6.3.4	Τοποθέτηση καμερών και ευαισθησία . . . . .	80
6.4	Πειραματικός προσδιορισμός σφαλμάτων . . . . .	82
<b>7</b>	<b>Συμπεράσματα</b>	<b>84</b>
7.1	Συμπεράσματα . . . . .	84
7.2	Μελλοντική έρευνα . . . . .	85
7.2.1	Τεχνητή όραση . . . . .	85
7.2.2	Βελτίωση συστήματος αντίχενωσης . . . . .	85
<b>A'</b>	<b>Κώδικες</b>	<b>87</b>

# Κατάλογος σχημάτων

2.1	Μοντέλο Pinhole . . . . .	10
2.2	Θετικός φακός . . . . .	12
2.3	Αρνητικός φακός . . . . .	12
2.4	Πεδίο βαρελοειδούς παραμόρφωσης . . . . .	15
2.5	Πεδίο ρομβοειδούς παραμόρφωσης . . . . .	15
2.6	Πεδίο εφραπτομενικής παραμόρφωσης . . . . .	16
2.7	Ολικό πεδίο παραμόρφωσης . . . . .	16
2.8	Οπτικές παραμορφώσεις από ευρυγώνιο φακό . . . . .	17
2.9	Ανάλυση αποχρώσεων γκριζου . . . . .	18
2.10	Ανάλυση χρωματικών αποχρώσεων στο μοντέλο χρώματος RGB. . . . .	19
2.11	Ανάμιξη βασικών χρωμάτων στο μοντέλο RGB. . . . .	19
2.12	Έγχρωμη φωτογραφία και τα RGB κανάλια της. . . . .	20
2.13	Άνω επίπεδο του κυλίνδρου HSV. . . . .	22
2.14	Τομή του κυλίνδρου HSV στην τιμή $V = 0.5$ . . . . .	22
2.15	Τομή του κυλίνδρου HSL στην τιμή $L = 0.5$ . . . . .	22
2.16	Έγχρωμη φωτογραφία και τα HSV κανάλια της. . . . .	23
2.17	Έγχρωμη φωτογραφία και τα HSL κανάλια της. . . . .	24
2.18	Φωτογραφία πριν και μετά την ρύθμιση κάμερας . . . . .	30
2.19	Πυρήνας συνέλιξης . . . . .	31
2.20	Παράθεση πυρήνα πάνω στην επεξεργαζόμενη εικόνα . . . . .	31
2.21	Κατανομές Gauss για διάφορες τιμές τυπικής απόκλισης . . . . .	33
2.22	Σύγκριση θόλωσης μέσου όρου και θόλωσης Gauss . . . . .	34
2.23	Τύποι διαχωρισμού . . . . .	35
2.24	Δυαδικός διαχωρισμός για διάφορες τιμές αποκοπής. . . . .	36
2.25	Εφαρμογή τελεστή Laplace. . . . .	38
2.26	Εφαρμογή τελεστών Sobel. . . . .	39
2.27	Απλές εφαρμογές Hough. . . . .	40
2.28	Απλές εφαρμογές Hough(συνέχεια). . . . .	41
3.1	Μετασχηματισμός πλαισίου . . . . .	43
3.2	Παράσταση παραμέτρων Denavit–Hartenberg . . . . .	46
4.1	Χωρισμός του προβλήματος σε μικρότερα υποπροβλήματα . . . . .	47
4.2	Προβολή των κύκλων ως ελλείψεις σε ένα επίπεδο . . . . .	50
4.3	Γραφική παράσταση του χάρτη του πρώτου μοντέλου . . . . .	50
4.4	Γεωμετρικά στοιχεία εύρεσης χάρτη. . . . .	53
4.5	Γωνίες ορατότητας στόχων. . . . .	54
4.6	Γωνίες οριακής θέασης . . . . .	56
4.7	Γραφική παράσταση του χάρτη του δεύτερου μοντέλου. . . . .	57

4.8	Γεωμετρική αναπαράσταση του τρίτου μοντέλου. . . . .	57
4.9	Γραφική παράσταση του χάρτη του τρίτου μοντέλου. . . . .	58
4.10	Ανίχνευση χρωμάτων με ταυτόχρονη όραση από δύο κάμερες. . . . .	59
4.11	Πραγματικός χάρτης επίλυσης του ρομπότ. . . . .	59
5.1	Πειραματική διάταξη . . . . .	66
5.2	Οι κάμερες της διάταξης όπως φαίνονται από την άρθρωση του ρομπότ. . . . .	66
5.3	Ανίχνευση χρωμάτων από τις δυο κάμερες . . . . .	67
5.4	Επεξεργασία ανιχνευμένων χρωμάτων από τις δυο κάμερες . . . . .	68
6.1	Το ρομπότ Staübli Rx90 . . . . .	71
6.2	Η Logitech C525HD . . . . .	71
6.3	Μαρκάρισμα ρομπότ . . . . .	72
6.4	Σημάδι από μετρητικό έλασμα . . . . .	72
6.5	Επανατοποθέτηση του ελάσματος στο σημάδι του . . . . .	73
6.6	Τοποθέτηση δεύτερης κάμερας . . . . .	73
6.7	Ορατά pixels από την αριστερή κάμερα χωρίς συνθήκη θέασης. . . . .	75
6.8	Ορατά pixels από την δεξιά κάμερα χωρίς συνθήκη θέασης. . . . .	75
6.9	Ορατά pixels από την αριστερή κάμερα με συνθήκη θέασης. . . . .	76
6.10	Ορατά pixels από την δεξιά κάμερα με συνθήκη θέασης. . . . .	76
6.11	Ορατό εμβαδόν από την αριστερή κάμερα. . . . .	77
6.12	Ορατό εμβαδόν από την δεξιά κάμερα. . . . .	77
6.13	Λύσεις γωνίας αριστερής κάμερας και ο μέσος όρος τους. . . . .	78
6.14	Λύσεις γωνίας δεξιάς κάμερας και ο μέσος όρος τους. . . . .	78
6.15	Λύσεις των δυο καμερών . . . . .	79
6.16	Επίδραση των μετατοπίσεων $dx, dy, dz$ στον χάρτη επίλυσης. . . . .	81
6.17	Πειραματικά αποτελέσματα της μέτρησης του σφάλματος. . . . .	82
A'.1	Διάγραμμα ροής της επίλυσης . . . . .	87

# Κεφάλαιο 1

## Εισαγωγή

Η ανάπτυξη της τεχνολογίας κατά τον 20<sup>ο</sup> αιώνα έφερε την επανάσταση στον τομέα των μελετών και των κατασκευών. Ειδικότερα η ανάπτυξη της ηλεκτρονικής αλλά και της επιστήμης του αυτομάτου ελέγχου κατέστησε δυνατή την μαζική αυτοματοποίηση γραμμών παραγωγής με χρήση έξυπνων συστημάτων υψηλής ακρίβειας και ταχύτητας. Μια βασική συνιστώσα των συστημάτων αυτών είναι τα ρομπότ. Ως ρομπότ μπορούν να θεωρηθούν βάσει ενός γενικού ορισμού, οι μηχανές αυτές οι οποίες είναι σε θέση να πραγματοποιούν αυτόματα ή με καθοδήγηση μια ακολουθία εργασιών. Βέβαια είναι προφανές ότι ο παραπάνω ορισμός δεν είναι δυνατόν να καλύψει όλα τα είδη ρομπότ. Ωστόσο από την μεριά του μηχανολόγου μηχανικού ως ρομπότ θα θεωρηθεί κατ'ουσίαν ένα φυσικό ηλεκτρομηχανικό σύστημα το οποίο οδηγείται μέσω προγραμματισμού από έναν ηλεκτρονικό υπολογιστή.

Ένα από τα πιο συχνά χρησιμοποιούμενα είδη ρομπότ είναι οι ρομποτικοί βραχίονες που είναι συστήματα τα οποία μοιάζουν με ανθρώπινα χέρια. Φυσικά είναι προφανές ότι οι αρθρώσεις σε ρομποτικούς βραχίονες δεν είναι απαραίτητα τρεις όπως στο ανθρώπινο χέρι αλλά και δεν έχουν το ίδιο εύρος κινήσεων. Κάθε ρομπότ εξοπλίζεται με ένα εργαλείο στον ακροδέκτη του, ανάλογα με την εργασία που πρέπει να επιτελέσει. Ένα βασικό πρόβλημα προς επίλυση για την σωστή λειτουργία ενός ρομπότ είναι η κινηματική του λειτουργία. Με τον όρο αυτό εννοούμε την δυνατότητα εξεύρεσης του τελικού σημείου δράσης για δεδομένες γωνίες αρθρώσεων (ευθεία κινηματική), ή την εύρεση των γωνιών των αρθρώσεων για δεδομένο σημείο δράσης (αντίστροφη κινηματική). Τα παραπάνω προβλήματα είναι λυμένα και πλήρως τεκμηριωμένα και οι κινηματικές εξισώσεις του ρομπότ οι οποίες είναι απαραίτητες για την λειτουργία του είναι ενσωματωμένες στους αντίστοιχους βιομηχανικούς ελεγκτές.

### 1.1 Διατύπωση του προβλήματος

Όπως αναφέρθηκε, για την επίλυση του προβλήματος της ευθείας κινηματικής πρέπει απαραίτητα να είναι γνωστές οι γωνίες των αρθρώσεων. Αυτό ισχύει κατά την λειτουργία των ρομπότ αφού συχνά αυτά εμπεριέχουν κωδικοποιητές στις αρθρώσεις τους οι οποίοι τους επιτρέπουν την μέτρηση των γωνιών τους. Ωστόσο, αυτή η πληροφορία δεν είναι εύκολα διαθέσιμη εκτός του ελεγκτή ή εκτός του περιβάλλοντος προγραμματισμού του ρομπότ. Σκοπός της παρούσας διπλωματι-



κής είναι η συνεισφορά προς την κατεύθυνση της ανάπτυξης και υλοποίησης ενός συστήματος τηλεπαρακολούθησης το οποίο θα μπορεί να μετρά τις θέσεις των αρθρώσεων ενός ρομπότ κατά την λειτουργία του χρησιμοποιώντας τεχνικές τεχνητής όρασης. Συγκεκριμένα γίνεται μια μαθηματική διερεύνηση και υλοποίηση του συστήματος για την πρώτη άρθρωση της κινηματικής αλυσίδας η οποία βρίσκεται μετά την βάση του ρομπότ.

## 1.2 Σημασία επίλυσης και πιθανές εφαρμογές

Ένα σύστημα τηλεπαρακολούθησης δίνει την δυνατότητα εξαγωγής πληροφορίας σχετικά με την κινηματική κατάσταση του ρομπότ χωρίς να διατηρεί διεπαφή με τον ελεγκτή του ρομπότ. Η ανεξαρτησία του συστήματος από από τα ηλεκτρονικά και μηχανικά κομμάτια του ρομπότ επιτρέπει την διάγνωση βλαβών ή αποκλίσεων στους αισθητήρες του ρομπότ οι οποίες δεν θα ήταν εύκολο να εντοπισθούν αφού οι αισθητήρες θα έδιναν λάθος πληροφορία. Επιπρόσθετα η εύκολη διάθεση πληροφορίας καθιστά ευκολότερη την ενσωμάτωση του ρομπότ σε κύτταρα παραγωγής συνεργαζόμενων μηχανών. Τέλος, η ύπαρξη του οπτικού συστήματος δίνει τη δυνατότητα παρακολούθησης όχι μόνο του ρομπότ αλλά και του περιβάλλοντος χώρου εργασίας επιτρέποντας έτσι στο ρομπότ να λειτουργεί πλέον όχι σε στατικό αλλά σε δυναμικό περιβάλλον εργασίας. Μερικά παραδείγματα εφαρμογών αποτελούν:

- Ταυτόχρονη παρακολούθηση πολλών ρομπότ από το ίδιο οπτικό πεδίο.
- Ένταξη ρομπότ σε έξυπνο κύτταρο παραγωγής.
- Αντίδραση σε δυναμικό χώρο εργασίας(π.χ ανίχνευση παρουσίας ατόμου στην τροχιά του ρομπότ και αυτόματη διακοπή λειτουργίας).

## 1.3 Προηγούμενη Σχετική Έρευνα

Οι τομείς της ρομποτικής, της επιστήμης υπολογιστών και της τεχνητής όρασης είναι αρκετά συναφείς. Συνέπεια αυτού είναι η ύπαρξη πολλών ερευνητών οι οποίοι δραστηριοποιούνται σε ερευνητικά έργα τα οποία συνδυάζουν τους παραπάνω τομείς. Τα πεδία που καλύπτονται είναι αυτά του οπτικού εντοπισμού(Visual Tracking), έλεγχος οπτικής ανάδρασης(Visual Servoing, Visual Feedback, Visual Control) αλλά και της τηλεπαρακολούθησης(Telemonitoring). Είναι προφανές πως η βιβλιογραφία γύρω από αυτά τα θέματα είναι εκτενής. Παρακάτω παρουσιάζονται εργασίες οι οποίες είναι συγγενείς με την παρούσα εργασία.

Οι Chi-Yi Tsai και Kai-Tai Song[1] παρουσίασαν έναν τρόπο ελέγχου ενός κινητού ρομπότ μέσω στιβαρού οπτικού εντοπισμού. Ο έλεγχος οπτικής ανάδρασης γίνεται για ένα κινητό ρομπότ το οποίο είναι μη ολονομικό. Πάνω στο ρομπότ έχει τοποθετηθεί μια κάμερα η οποία έχει τη δυνατότητα κλίσης. Ο σχεδιασμός αυτός έχει στόχο να επιτρέπει στο ρομπότ να κρατά στο οπτικό του πεδίο έναν δυναμικά κινούμενο στόχο. Ακόμη και εάν αυτός είναι κρυμμένος για λίγο. Για την λειτουργία του, το σύστημα χρησιμοποιεί ένα σχήμα ελέγχου με οπτικού εντοπισμό αλλά και έναν εκτιμητή κατάστασης βασισμένος και αυτός σε οπτικό εντοπισμό. Για τον οπτικό εντοπισμό χρησιμοποιείται ένα μοντέλο οπτικής αλληλεπίδρασης.

Ο εκτιμητής βρίσκει την βέλτιστη κατάσταση του στόχου και την ταχύτητα στον χώρο εικόνας. Στην συνέχεια ο ελεγκτής υπολογίζει τις ταχύτητες που χρειάζεται το ρομπότ. Η χρήση εκτιμητή δίνει στιβαρότητα στο σύστημα και λύνει το πρόβλημα του μη ορατού στόχου.

Οι T.P.Sim, G.S.Hong και K.B.Lim[2] προτείνουν ένα σύστημα τρισδιάστατου οπτικού ελέγχου το οποίο βασίζεται στην θέση της κάμερας. Η κάμερα βρίσκεται στον ακροδέκτη του ρομπότ και κινείται μαζί με αυτόν. Το μοντέλο τους εισάγει έναν προσαρμοσμένο εκτιμητή τύπου Smith. Η διαφορά του σε σχέση με άλλες παρόμοιες τεχνικές είναι η εισαγωγή του εκτιμητή για την αντιμετώπιση της αργής δειγματοληψίας της εικόνας.

Η εργασία των Oliver Lang και Axel Graser[3] αναπτύσσει ένα στιβαρό σύστημα οπτικού ελέγχου για έναν βραχίονα 6 βαθμών ελευθερίας. Ένα από τα συχνά προβλήματα που εμφανίζονται σε συστήματα οπτικού ελέγχου είναι η αλλαγή μεγέθους των αντικειμένων τα οποία χρησιμοποιούνται ως σημεία αναφοράς. Αυτό έχει ως αποτέλεσμα υψηλές απαιτήσεις για αλγόριθμους αναγνώρισης. Επιπρόσθετα η μείωση του μεγέθους των αντικειμένων οδηγεί σε μεγαλύτερα επίπεδα θορύβου. Στην εργασία τους οι Oliver Lang και Axel Graser προσεγγίζουν το πρόβλημα χρησιμοποιώντας μια κάμερα με δυνατότητα zoom έτσι ώστε το zoom να αντισταθμίζει τις αλλαγές μεγέθους των σημείων αναφοράς που βλέπει η κάμερα κατά την κίνηση του ακροδέκτη του ρομπότ.

Οι Hesheng Wang, Yun-Hui Liu και Weidong Chen[4] ανέπτυξαν ένα σύστημα οπτικής παρακολούθησης ρομπότ σε μη ρυθμισμένα περιβάλλοντα. Αναπτύχθηκε ένας προσαρμοστικός ελεγκτής για την παρακολούθηση ενός ρομποτικού βραχίονα με χρήση μιας σταθερής κάμερας με μεταβλητές εσωτερικές παραμέτρους. Πέρα από τις παραμέτρους της κάμερας άγνωστες θεωρούνται και οι θέσεις των σημείων της εικόνας που παρακολουθούνται. Οι ερευνητές βασίστηκαν στο γεγονός ότι άγνωστες παράμετροι εμφανίζονται γραμμικά στην δυναμική κλειστού βρόγχου του συστήματος, εάν το μητρώο αλληλεπίδρασης (ανεξαρτήτου βάνθους) προσαρμοσθεί έτσι ώστε να δείχνει τα σφάλματα εικόνας στον χώρο αρθρώσεων του ρομπότ. Έτσι αναπτύχθηκε ένας αλγόριθμος on-line εκτίμησης των αγνώστων παραμέτρων, λαμβάνοντας υπόψη του την δυναμική του ρομποτικού βραχίονα.

Οι Tyrrell, J.A. et al[5] παρουσιάζουν τρόπους για αποδοτική μεταφορά πολύπλοκων off-line συστημάτων όρασης σε υλοποιήσεις πραγματικού χρόνου χρησιμοποιώντας κοινά υπολογιστικά συστήματα. Αυτό επιτυγχάνεται εισάγοντας ειδικά modules στον πυρήνα λειτουργικών συστημάτων όπως το Linux, τα αποτελέσματα ήταν ικανοποιητικά όταν χρησιμοποιούνται προβλέψιμοι αλγόριθμοι όρασης.

Οι Healey, G. και Binford, T.O.[6] παρουσιάζουν έναν τρόπο ποσοτικοποίησης χρωματικών χώρων. Η μεγάλη του διαφορά σε σχέση με άλλους τρόπους είναι ότι μπορεί να εφαρμοσθεί ευθέως σε έγχρωμες εικόνες που λαμβάνονται από σύστημα όρασης. Αυτό γίνεται μέσω χρωματικών φίλτρων. Η βαθμονόμηση ορίζεται σε όρους συχνοτικών χαρακτηριστικών των φίλτρων και της κάμερας. Η χρωματική απόσταση ορίζεται ως η ευκλείδεια απόσταση μεταξύ κανονικοποιημένων κατανομών ισχύος.

Οι Fung, J. και Mann, S.[7] παρουσιάζουν μεθόδους χρήσης κοινών υποσυστημάτων γραφικών ως υπολογιστές γενικής χρήσης. Ως παράδειγμα χρήσης προσαρμόζουν και υλοποιούν κοινούς αλγόριθμους όρασης οι οποίοι τρέχουν πάνω στην κατανεμημένη πολυεπεξεργαστική πλατφόρμα. Η παραπάνω εργασία είναι συγγενική με αυτή των Fung, J. et al[8] η οποία διαφοροποιείται ως προς την εφαρμογή η οποία είναι η δημιουργία ενός επαυξημένου περιβάλλοντος που η βάση του είναι οι προσλαμβανόμενες εικόνες από κάμερες τοποθετημένες σε ρούχα.

Ενδιαφέρουσα είναι επίσης η έρευνα των Shutao Zhao et al[9] στην οποία υλοποιείται ένα γενικό σύστημα μέτρησης. Το σύστημα αυτό κάνει μια οπτική διεπαφή με αναλογικούς και ψηφιακούς μετρητές διαβάζοντας την ένδειξη τους. Για τον σκοπό αυτό γίνεται οπτική αναγνώριση χαρακτήρων.

Οι Hyoung-Seok et al[10] παρουσιάζουν ένα σύστημα επιθεώρησης πραγματικού χρόνου. Το παρακολουθούμενο μέγεθος είναι το σφάλμα μορφοποίησης για μικρούς σωλήνες. Στην υλοποίηση του χρησιμοποιούνται προσαρμοσμένοι μετασχηματισμοί Hough για την εύρεση της γεωμετρίας των σωλήνων αλλά και των σφαλμάτων.

Αντίστοιχα οι Jia Liu και Tong Cui[11] ανέπτυξαν μια μέθοδο μέτρησης του σφάλματος κυκλικότητας το οποίο βασίζεται σε τεχνητή όραση. Η μέτρηση γίνεται σε κατασκευές πολύ μικρού μεγέθους. Οι φωτογραφίες λαμβάνονται με την βοήθεια μικροσκοπίου και οι ακμές ανιχνεύονται με προσαρμοσμένο φίλτρο canny.

Οι Wen-Ge Wang et al[12] δημιούργησαν ένα σύστημα όρασης το οποίο μετράει τρισδιάστατα αντικείμενα. Αυτό γίνεται με την χρήση στόχων οι οποίοι είναι προσκολλημένοι στο μετρούμενο αντικείμενο. Στην συνέχεια τα νέφη σημείων που έχουν προκύψει συναρμολογούνται στις αντίστοιχες επιφάνειες στο σύστημα συντεταγμένων αναφοράς.

Οι Laghari, M.S. και Ahmed, F.[13] παρουσιάζουν ένα σύστημα όρασης για την παρακολούθηση της φθοράς των μηχανών η οποία οφείλεται σε σωματίδια. Τα οπτικά χαρακτηριστικά των σωματιδίων συνδυάζονται με μοντέλα τεχνητής νοημοσύνης και φθοράς για εξαγωγή συμπερασμάτων για το είδος της φθοράς που προκαλούν τα σωματίδια στην μηχανή.

Στο άρθρο τους Computer vision applied to super resolution οι Capel, D. και Zisserman, A.[14] παρουσιάζουν μια τεχνική ενίσχυσης εικόνας. Η τεχνική αυτή συνθέτει μια εικόνα μεγάλης ανάλυσης από πολλές εικόνες μικρότερης ανάλυσης.

Οι Moon, C.W. και McVey, E.S.[15] παρουσιάζουν τις τεχνικές μέτρησης ακριβείας σε σύγκριση με της τεχνικές laser. Επιπρόσθετα επισημαίνουν τις δυσκολίες που συναντώνται σε αυτά τα συστήματα και τέλος κάνουν μια εισαγωγή στα σφάλματα μετρήσεων για δισδιάστατα και τρισδιάστατα αντικείμενα.

Ο E.S. Gadelmawla[16] παρουσιάζει ένα οπτικό σύστημα ακριβούς μέτρησης οδοντωτών τροχών. Το σύστημα αυτό έχει πλεονέκτημα σε σχέση με τα συμβατικά καθώς μετρά όλες τις παραμέτρους του τροχού χωρίς επαφή και σημαντικά μειωμένο χρόνο μετρήσεων.

Οι Chao-Ton Su et al[17] πραγματεύονται το θέμα της εισαγωγής νευρωνικών δικτύων για μετρήσεις ακριβείας σε συστήματα όρασης. Και περιγράφει διαδικασίες για την διόρθωση σφαλμάτων από οπτικές και άλλου είδους παραμορφώσεις. Αρχικά στην κατάστρωση μοντέλων για διόρθωση σφαλμάτων και στην συνέχεια στην χρήση αυτών των μοντέλων για την διόρθωση των μετρήσεων.

Ο T.D. Doiron[18] παρουσιάζει ένα κέντρο κατεργασιών το οποίο χρησιμοποιεί σύστημα όρασης το οποίο δίνει πληροφορίες για την θέση και τον τύπο του εργαλείου που χρησιμοποιείται. Το σύστημα δοκιμάζεται ως προς την επαναληψιμότητα και την ακρίβεια του δίνοντας καλά αποτελέσματα.

## Κεφάλαιο 2

# Στοιχεία Τεχνητής Όρασης

Η τεχνητή όραση αποτελεί το επιστημονικό πεδίο που συμπεριλαμβάνει και μελετά μεθόδους για την λήψη, επεξεργασία και ανάλυση δεδομένων τα οποία προέρχονται από ψηφιακές εικόνες. Σκοπός είναι, είτε ο μετασχηματισμός των δεδομένων της εικόνας σε άλλα πιο εύκολα αξιοποιήσιμα είτε, η λήψη κάποιας απόφασης η οποία δικαιολογείται από το μοτίβο των δεδομένων που υπάρχουν στην εικόνα.

### 2.1 Βασικές έννοιες

Για την εισαγωγή στις μεθόδους της τεχνητής όρασης, θα πρέπει πρώτα να θεμελιωθεί ένα βασικό υπόβαθρο. Μερικές από τις έννοιες ίσως είναι ήδη γνωστές στον αναγνώστη, ωστόσο αναφέρονται για λόγους πληρότητας.

#### 2.1.1 Αναπαράσταση εικόνας και εικονοστοιχεία

Στην ψηφιακή αναπαράσταση της μια εικόνα δεν μπορεί να παρασταθεί από ένα σύνολο απειροστών στοιχείων καθώς όπως είναι γνωστό ένας υπολογιστής μπορεί να παραστήσει αριθμητικές τιμές με πεπερασμένη ακρίβεια. Για τον λόγο αυτό το οπτικό πεδίο της κάμερας χωρίζεται σε μικρά αλλά διακριτά στοιχεία στα οποία ονομάζονται εικονοστοιχεία (από το Αγγλικό PICTure ELeMent ή Pixel). Κάθε εικονοστοιχείο λαμβάνει μια ή περισσότερες τιμές ανάλογα με τον αισθητήρα λήψης που χρησιμοποιείται (έγχρωμη εικόνα ή όχι) και αποθηκεύεται στον υπολογιστή υπό την μορφή μητρώου. Το γινόμενο του αριθμού των γραμμών επί τον αριθμό των στηλών ονομάζεται ανάλυση και είναι ένα μέτρο ποιότητας της εικόνας. Προφανώς όσο μεγαλύτερη η ανάλυση τόσο καλύτερη η ποιότητα της εικόνας.

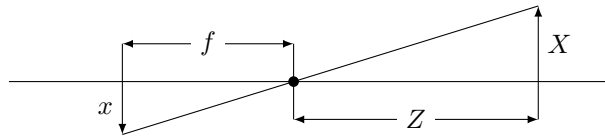
Στην απλούστερη μορφή της (όταν κάθε pixel έχει μόνο μια τιμή), μια εικόνα αποθηκεύεται ως ένα διδιάστατο μητρώο. Από εκεί και πέρα, για κάθε επιπλέον τιμή που αντιστοιχεί στο εκάστοτε pixel προστίθεται ένα επιπλέον 2D επίπεδο που περιέχει την επιπλέον τιμή του εκάστοτε pixel. Το επίπεδο αυτό ονομάζεται και κανάλι. Ο αριθμός των καναλιών που μπορούν να έχουν οι εικόνες, αλλά και το τι τιμές λαμβάνει το κάθε pixel αναλύεται εκτενέστερα στην ενότητα για την αναπαράσταση των χρωμάτων.

### 2.1.2 Κάμερες

Η κάμερα είναι μια διάταξη η οποία επιτρέπει την λήψη φωτογραφιών. Οι σύγχρονες κάμερες είναι ψηφιακές, δηλαδή επιτρέπουν την απευθείας αποθήκευση των φωτογραφιών σε ηλεκτρονικό υπολογιστή. Αποτελούνται από δυο βασικά μέρη, τον φακό και τον αισθητήρα. Ο φακός επιτρέπει την συγκέντρωση του φωτός ενώ ο αισθητήρας μετατρέπει το συγκεντρωμένο φως σε σήμα το οποίο διαβάζεται και μετά από κάποια επεξεργασία αποθηκεύεται ως εικόνα είτε στην ίδια την κάμερα είτε στον ηλεκτρονικό υπολογιστή. Στην παρούσα μελέτη η κάμερα μοντελοποιείται με το μοντέλο pinhole το οποίο είναι ένα σχετικά απλό μοντέλο το οποίο περιγράφει μαθηματικά την κάμερα.

### 2.1.3 Μοντέλο Pinhole

Έστω μια κάμερα με σημειακή οπή από την οποία διέρχονται ακτίνες φωτός. Αριστερά της οπής βρίσκεται το επίπεδο αποτύπωσης της φωτογραφίας και δεξιά της το απεικονιζόμενο αντικείμενο ή επίπεδο. Όταν η ακτίνα φωτός φεύγει από το αντικείμενο και προσπίπτει στο επίπεδο αποτύπωσης διαμέσου της οπής τότε το σημείο αυτό προβάλλεται στο επίπεδο αποτύπωσης. Αυτό έχει ως αποτέλεσμα η εικόνα να είναι πάντοτε εστιασμένη και το μέγεθος της ως προς το εικονιζόμενο αντικείμενο να δίνεται από μια και μόνο παράμετρο, το εστιακό μήκος  $f$ .



Σχήμα 2.1: Ακτίνα φωτός από το εικονιζόμενο αντικείμενο στο επίπεδο αποτύπωσης διαμέσου της οπής

Εάν  $Z$  είναι η απόσταση του αντικειμένου από την κάμερα,  $X$  η κατακόρυφη απόσταση από το οριζόντιο επίπεδο και  $x$  η κατακόρυφη απόσταση από το οριζόντιο επίπεδο στο επίπεδο αποτύπωσης, τότε χρησιμοποιώντας όμοια τρίγωνα προκύπτει:

$$-x = f \frac{X}{Z} \quad (2.1)$$

Για λόγους ευκολίας μεταφέρεται το επίπεδο αποτύπωσης μπροστά από την οπή. Στην διάταξη αυτή τόσο το επίπεδο αποτύπωσης όσο και το αντικείμενο βρίσκονται πλέον δεξιά της οπής η οποία μετονομάζεται σε κέντρο της προβολής. Χρησιμοποιώντας πάλι όμοια τρίγωνα προκύπτει:

$$x = f \frac{X}{Z} \quad (2.2)$$

Όπου το  $x$  δεν είναι αρνητικό αφού η εικόνα εμφανίζεται πλέον ορθά και όχι ανάποδα. Ωστόσο στις πραγματικές κάμερες το κέντρο της προβολής δεν βρίσκεται στο κέντρο του αισθητήρα και αυτό διότι μια πλήρης μηχανική ευθυγράμμιση είναι αδύνατον να επιτευχθεί. Έτσι για αντιστάθμιση αυτής της μετατόπισης εισάγονται οι παράμετροι  $c_x$  και  $c_y$ . Ένα άλλο σημείο το οποίο πρέπει να τονιστεί είναι πως οι τυπικές κάμερες δεν έχουν τετραγωνική μορφή Pixels αλλά ορθογωνική, έτσι

υπάρχουν δυο εστιακά μήκη. Τελικά με βάση τα παραπάνω ένα σημείο στο χώρο  $Q(X, Y, Z)$  αντιστοιχείται σε σημείο  $(x, y)$  πάνω στο επίπεδο προβολής με τις σχέσεις:

$$x = f_x \frac{X}{Z} + c_x \quad y = f_y \frac{Y}{Z} + c_y \quad (2.3)$$

Τα  $f_x$  και  $f_y$  συνδέονται με το μέγεθος του αισθητήρα με τις σχέσεις:

$$f_x = F s_x \quad f_y = F s_y \quad (2.4)$$

Με το  $s_i$  να είναι ο αριθμός pixel/mm σε κάθε κατεύθυνση και  $F$  να είναι σε mm. Δηλαδή τα  $f_i$  είναι σε μονάδες pixel. Αξίζει να σημειωθεί πως ενώ οι ποσότητες  $f_i$  είναι δυνατόν να εξαχθούν, τα  $F$  και  $s$  δεν είναι μετρήσιμα εάν δεν γίνει αποσυναρμολόγηση της κάμερας ώστε να μετρηθούν απευθείας.

### Ομογενείς Συντεταγμένες

Η σχέση μεταξύ των σημείων  $Q_i(X_i, Y_i, Z_i)$  και των σημείων  $(x_i, y_i)$  είναι ένας προβολικός μετασχηματισμός. Στον μετασχηματισμό αυτό χρησιμοποιούνται οι ομογενείς συντεταγμένες όπου ένα σημείο σε  $n$ -διάστατο χώρο γράφεται ως διάνυσμα  $n+1$  διαστάσεων με τον επιπλέον περιορισμό πως δυο σημεία των οποίων οι τιμές είναι ανάλογες είναι ίσα<sup>1</sup>. Χρησιμοποιώντας αυτόν τον περιορισμό είναι δυνατή η εξαγωγή των συντεταγμένων του pixel που τους αντιστοιχεί. Με τον τρόπο αυτό οι παράμετροι που ορίζουν την κάμερα μπορούν να γραφούν ως ένα μητρώο  $3 \times 3$  που ονομάζεται χαρακτηριστικός πίνακας. Έτσι οι συντεταγμένες  $q$  στο επίπεδο αποτύπωσης υπολογίζονται σύμφωνα με την επόμενη σχέση:

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.5)$$

Το Pinhole μοντέλο αποδίδει μια χρήσιμη αναπαράσταση της τρισδιάστατης γεωμετρίας της τεχνητής όρασης. Φυσικά στην πράξη πολύ λίγο φως περνά από την οπή και έτσι η φωτογράφιση θα ήταν εξαιρετικά αργή καθώς θα χρειαζόταν πολύ ώρα για συσσωρευθεί αρκετό φως για να γίνει η αποτύπωση. Για τον λόγο αυτό χρησιμοποιούνται φακοί για την συγκέντρωση φωτός αλλά με κόστος την εισαγωγή οπτικών παραμορφώσεων.

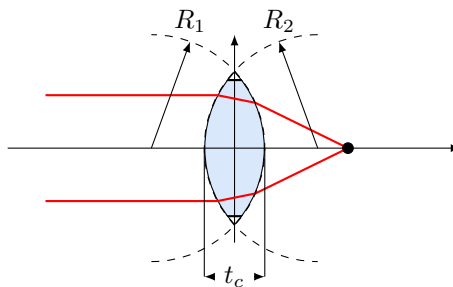
## 2.2 Οπτική

Η οπτική είναι ο τομέας αυτός της φυσικής ο οποίος μελετά τη διάδοση του φωτός στο κενό, τον αέρα αλλά και άλλα οπτικά μέσα όπως το γυαλί ή πολυμερή. Επιπρόσθετα, στο αντικείμενο της οπτικής περιλαμβάνεται η κατασκευή διατάξεων οι οποίες μεταδίδουν οπτικά σήματα καθώς και φακών για τη συγκέντρωση αλλά και την διαχείριση οπτικών δεσμών. Αυτό επιτυγχάνεται χρησιμοποιώντας έναν ή περισσότερους φακούς σε συνδυασμό.

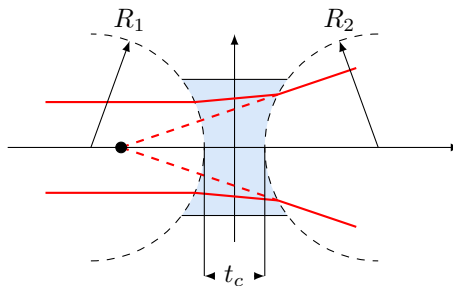
<sup>1</sup> Δηλαδή όταν τα σημεία αυτά βρίσκονται στην τροχιά ίδιας ακτίνας φωτός

### 2.2.1 Φακοί - Τύποι φακών

Ως φακός μπορεί να χαρακτηριστεί μια οπτική συσκευή η οποία διαθλά μια οπτική δέσμη με σκοπό την σύγκλιση ή απόκλιση της. Βάσει αυτής της ιδιότητας οι φακοί κατηγοριοποιούνται σε θετικούς ή συγκλίνοντες που είναι οι φακοί που προκαλούν σύγκλιση σε μια παράλληλη εισερχόμενη δέσμη φωτός. Τέτοιοι φακοί είναι οι επιπεδόκυρτοι, αμφίκυρτοι και οι μηνίσκοι. Αντίστοιχα αρνητικοί ή αποκλίνοντες είναι οι φακοί που προκαλούν απόκλιση σε μια παράλληλη εισερχόμενη δέσμη. Τέτοιοι φακοί είναι οι επιπεδόκοιλοι, αμφίκοιλοι και οι μηνίσκοι.



Σχήμα 2.2: Συγκέντρωση φωτεινών δεσμών στο εστιακό σημείο από θετικό φακό



Σχήμα 2.3: Απόκλιση φωτεινών δεσμών από αρνητικό φακό

Κάθε φακός έχει ένα εστιακό σημείο, που είναι το σημείο όπου θεωρητικά περνούν όλες οι ακτίνες φωτός που διατρέχουν τον φακό. Η θέση του σημείου αυτού δίνεται από την σχέση:

$$\frac{1}{f} = (n - 1) \left[ \frac{1}{R_1} - \frac{1}{R_2} + \frac{t_c(n - 1)}{nR_1R_2} \right] \quad (2.6)$$

Όπου  $n$  ο δείκτης διάθλασης του υλικού του φακού,  $t_c$  το πάχος του φακού και  $R_i$  οι ακτίνες καμπυλότητας των πλευρών του φακού. Από τα παραπάνω προκύπτει πως για αρνητικούς φακούς το εστιακό σημείο είναι αριστερά του φακού και εμφανίζεται στην τομή της νοητής επέκτασης των ακτίνων που αποκλίνουν.

Στις περισσότερες κάμερες χρησιμοποιούνται αμφίκυρτοι φακοί ώστε να υπάρχει συγκέντρωση φωτός στον αισθητήρα της. Ο φακός αυτός είναι προσαρμοσμένος με ένα περικύχλιο στο σώμα της κάμερας επιτρέποντας την σχετική κίνηση του φακού ώστε να επιτευχθεί εστίαση σε διαφορετικά βάθη.

Σε ακριβότερες βιομηχανικές ή επαγγελματικές κάμερες χρησιμοποιείται μια συστοιχία θετικών και αρνητικών φακών. Αυτό συμβαίνει γιατί οι κάμερες αυτές έχουν μεγαλύτερο αισθητήρα. Ο μεγαλύτερος αισθητήρας απαιτεί την χρήση μεγαλύτερου φακού για την σωστή αποτύπωση της εικόνας κάτι που εισάγει μεγαλύτερες οπτικές παραμορφώσεις. Για την διόρθωση αυτών των παραμορφώσεων αλλά και την σωστή εστίαση χρησιμοποιούνται οι υπόλοιποι φακοί της συστοιχίας.

### 2.2.2 Σφάλματα και παραμορφώσεις φακών

Παρά την σαφή θεωρητική αναλυτική μοντελοποίηση των φακών υπάρχουν σφάλματα στην πρακτική τους χρήση. Αυτό συμβαίνει κυρίως για κατασκευαστικούς λόγους καθώς είναι πιο εύκολη η κατασκευή ενός σφαιρικού σχήματος από ένα ιδανικό παραβολικό σχήμα. Επιπρόσθετα, η πλήρης μηχανική ευθυγράμμιση είναι αδύνατη. Συνεπώς τα σφάλματα των φακών μπορεί να ποικίλουν. Στα προβλήματα τεχνητής όρασης ωστόσο δύο είδη παραμορφώσεων είναι σημαντικά. Η ακτινική παραμόρφωση και η εφαιπτομενική παραμόρφωση.

#### Ακτινική Παραμόρφωση

Η ακτινική παραμόρφωση οφείλεται στην καμπυλότητα του φακού, την απόσταση του από το επίπεδο αποτύπωσης αλλά και την κυματική φύση του φωτός και εμφανίζεται με δυο μορφές: την βαρελοειδή μορφή και την ρομβοειδή μορφή. Κατά την βαρελοειδή μορφή τα στοιχεία της εικόνας που βρίσκονται στο κέντρο του φακού μεγεθύνονται περισσότερο από εκείνα που βρίσκονται στην άκρη αυτού, ενώ στην ρομβοειδή μορφή συμβαίνει το αντίθετο.

Η μαθηματική μοντελοποίηση της ακτινικής παραμόρφωσης γίνεται χρησιμοποιώντας το ανάπτυγμα Taylor στην γειτονιά του κέντρου του φακού. Έτσι η συνάρτηση παραμόρφωσης δίνεται από την σχέση:

$$f(r) = \alpha_0 + \alpha_1 r + \alpha_2 r^2 + \dots + \alpha_n r^n \quad (2.7)$$

Για την εύρεση των συντελεστών χρησιμοποιούνται οι περιορισμοί ότι  $f(0) = 0$  άρα  $\alpha_0 = 0$  και πως υπάρχει συμμετρία στο πεδίο παραμορφώσεων. Συνεπώς όλοι οι περιττοί συντελεστές είναι μηδενικοί. Η χρήση συντελεστών έως και τρίτου βαθμού δίνει την επιθυμητή ακρίβεια και η συνάρτηση παραμόρφωσης που προκύπτει είναι η εξής:

$$f(r) = \alpha_2 r^2 + \alpha_4 r^4 + \alpha_6 r^6 \quad (2.8)$$

Εφόσον είναι πλέον γνωστή η έκφραση της παραμόρφωσης είναι δυνατή η διόρθωση της στην εικόνα. Έτσι οι διορθωμένες συντεταγμένες είναι:

$$x_{cor} = x(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) \quad (2.9)$$

$$y_{cor} = y(1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) \quad (2.10)$$

#### Εφαιπτομενική Παραμόρφωση

Η παραμόρφωση αυτή οφείλεται σε σφάλματα μηχανικής ευθυγράμμισης των φακών. Η εξαγωγή της μαθηματικής της έκφρασης είναι πολύπλοκη αλλά οι διορθωμένες θέσεις των συντεταγμένων τελικά μπορούν να παρασταθούν ικανοποιητικά με τις παρακάτω σχέσεις:



$$x_{cor} = x + [2p_1y + p_2(r^2 + 2x^2)] \quad (2.11)$$

$$y_{cor} = y + [p_1(r^2 + 2y) + 2p_2x] \quad (2.12)$$

Για μια κάμερα που έχει και τα δυο είδη παραμορφώσεων υπάρχει η δυνατότητα αντιστάθμισης χρησιμοποιώντας τις παρακάτω σχέσεις:

$$\begin{bmatrix} x_{cor} \\ y_{cor} \end{bmatrix} = (1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2p_1xy + p_2(r^2 + 2x^2) \\ p_1(r^2 + 2y) + 2p_2xy \end{bmatrix} \quad (2.13)$$

Η εύρεση των 5 αναγκαίων παραμέτρων θα αναλυθεί στην ενότητα της αποπαραμόρφωσης εικόνας.

## 2.3 Αισθητήρες φωτογραφίας

Οι αισθητήρες φωτογραφίας είναι τα κομμάτια αυτά που μετατρέπουν το φως ή καλύτερα την ηλεκτρομαγνητική ακτινοβολία σε μια τάση η οποία μετράται και η τιμή της αποθηκεύεται σε μέσα αποθήκευσης. Στις κάμερες παλαιότερης τεχνολογίας το μέσο το οποίο αντιλαμβάνονταν το φως και ταυτόχρονα το αποθήκευε ήταν το φιλμ. Στις σύγχρονες κάμερες η δουλειά αυτή γίνεται από τους αισθητήρες CCD και CMOS. Οι παραπάνω αισθητήρες απλά μετατρέπουν το φως σε ηλεκτρικό σήμα ενώ η διαδικασία αποθήκευσης γίνεται από άλλες διατάξεις.

### 2.3.1 Αισθητήρες CCD

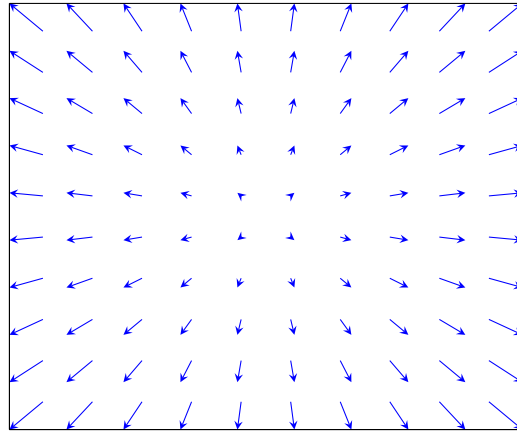
Οι αισθητήρες αυτοί αποτελούνται από μια φωτοευαίσθητη επιφάνεια και από μια περιοχή αναμετάδοσης φορτίου. Όταν το φως προσπίπτει στην φωτοευαίσθητη επιφάνεια τότε αυτή συσσωρεύει φορτίο ανάλογο της έντασης της ακτινοβολίας. Όταν η εικόνα έχει αποτυπωθεί το κύκλωμα ελέγχου μεταφέρει το φορτίο από κάθε pixel στο διπλανό του μέχρι αυτό να φτάσει σε έναν ενισχυτή φορτίου, ο οποίος το μετατρέπει σε τάση. Επαναλαμβάνοντας αυτή τη διαδικασία για κάθε pixel η εικόνα μετατρέπεται σε μια ακολουθία τάσεων. Οι τάσεις αυτές ψηφιοποιούνται και στην συνέχεια αποθηκεύονται.

### 2.3.2 Αισθητήρες CMOS

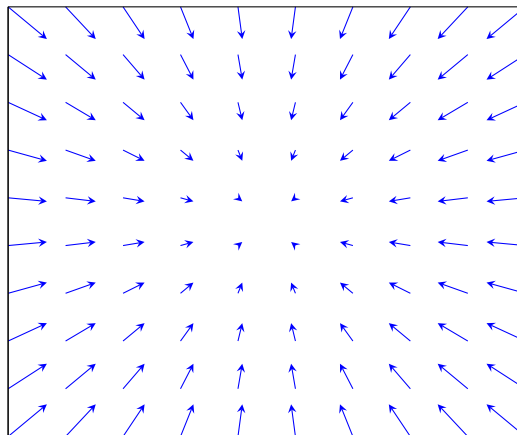
Αποτελούνται από συστοιχία φωτοδίοδων αντί για ένα συνεχές φωτοευαίσθητο μέσο. Αυτό έχει το πλεονέκτημα ότι όταν ο αισθητήρας εκτίθεται σε φως υψηλής έντασης δεν υπάρχει διαρροή φορτίου σε γειτονικές περιοχές. Επιπρόσθετα η κατασκευή των αισθητήρων αυτών είναι οικονομικότερη. Ωστόσο οι αισθητήρες αυτοί είναι πιο αργοί και παράγουν παραμορφωμένες εικόνες όταν απεικονίζονται γρήγορα κινούμενα αντικείμενα.

## 2.4 Αναπαράσταση Χρωμάτων - Colorspaces

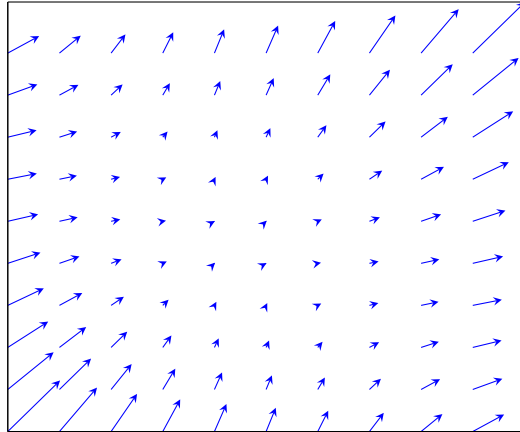
Τα μοντέλα χρώματος είναι τρόπος μαθηματικής περιγραφής των χρωμάτων χρησιμοποιώντας σύνολα αριθμών. Συχνά τα σύνολα αυτά αποτελούνται από τρεις ή τέσσερις αριθμούς. Ένα ακόμη σημαντικό στοιχείο στην αναπαράσταση των



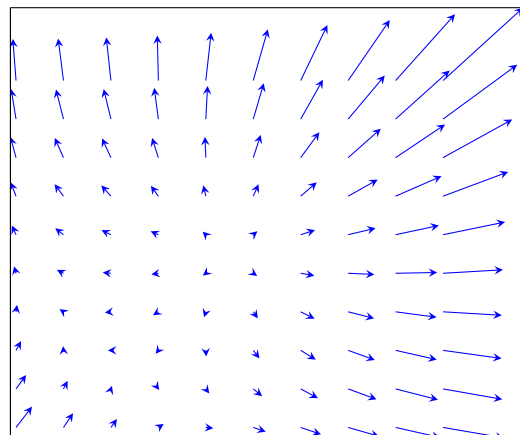
Σχήμα 2.4: Μετατόπιση των σημείων από την κανονική τους θέση κατά την βαρελοειδή παραμόρφωση



Σχήμα 2.5: Μετατόπιση των σημείων από την κανονική τους θέση κατά την ρομβοειδή παραμόρφωση



Σχήμα 2.6: Μετατόπιση των σημείων από την κανονική τους θέση κατά την εφαπτομενική παραμόρφωση



Σχήμα 2.7: Συνολική παραμόρφωση υπό την επίδραση βαρελοειδούς και εφαπτομενικής παραμόρφωσης



Σχήμα 2.8: Φωτογραφία με χρήση ευρυγώνιου φακού(μεγάλης ακτίνας καμπυλότητας). Οι παραμορφώσεις είναι εμφανείς από το σχήμα των ευθειών

χρωμάτων είναι η ανάλυση, δηλαδή πόσα διαφορετικά χρώματα μπορούν να παρασταθούν από έναν ηλεκτρονικό υπολογιστή. Στην συνέχεια αναλύονται οι διάφορες αναλύσεις αλλά και τα πιο συχνά χρησιμοποιούμενα μοντέλα χρωμάτων.

#### 2.4.1 Ανάλυση χρώματος - Ακρίβεια εικόνας

Έστω δυο χρώματα το μαύρο και το άσπρο τα οποία μπορούν να παρασταθούν από έναν υπολογιστή. Όπου το μηδέν αντιστοιχεί σε πλήρη απουσία χρώματος (μαύρο) και το ένα σε πλήρη παρουσία χρώματος (άσπρο). Ζητούμενο είναι η αναπαράσταση μιας αυθαίρετης απόχρωσης του γκριζου. Είναι γνωστό πως το γκριζο αποτελεί μια ανάμειξη ενός ποσοστού μαύρου και ενός ποσοστού άσπρου. Εάν  $x$  είναι αυτό το ποσοστό, τότε:

$$V_g = xV_b + (1 - x)V_w \quad (2.14)$$

Εν προκειμένω επειδή η τιμή του μαύρου είναι μηδέν όλες οι γκριζες αποχρώσεις ανάγονται σε έναν δεκαδικό αριθμό επί την τιμή του άσπρου χρώματος. Η ανάλυση του χρώματος προκύπτει από την δυνατότητα του υπολογιστή να παραστήσει σωστά αυτόν τον αριθμό. Στην πράξη δεν χρησιμοποιούνται δεκαδικοί αριθμοί αλλά ακέραιοι οι οποίοι λαμβάνουν τιμές από μηδέν έως μια μέγιστη τιμή που εξαρτάται από το μήκος ψηφιολέξης του υπολογιστή. Σημειώνεται εδώ ότι αυτή η τιμή δεν είναι ίση με τον μέγιστο ακέραιο που μπορεί να παραστήσει ο υπολογιστής.

Θεωρώντας τώρα ότι τα bits που είναι διαθέσιμα για αποχρώσεις του μαύρου είναι  $n$  τότε όλες οι αποχρώσεις που είναι δυνατόν να παρασταθούν με αυτήν την ακρίβεια είναι  $2^n$ . Μια συνήθης ακρίβεια διακριτοποίησης για ένα χρώμα είναι η ακρίβεια 8-bit η οποία αποδίδει  $2^8 = 256$  αποχρώσεις του χρώματος αυτού με 0 να είναι η πλήρης απουσία χρώματος δηλαδή το μαύρο χρώμα και 255 να είναι καθαρό χρώμα.

Εφόσον μια ανάλυση χρώματος της τάξεως των 8-bit είναι επαρκής και όλα τα χρώματα μπορούν να παρασταθούν με έναν συνδυασμό κάποιων βασικών χρωμάτων, τότε κάθε στοιχείο με την σειρά του μπορεί να παρασταθεί με έναν συνδυασμό κάποιων τιμών. Κάθε διαφορετική τιμή ονομάζεται κανάλι. Σε μια τυπική ψηφιακή εικόνα τα κανάλια είναι τρία δηλαδή τα χρώματα που μπορούν να παρασταθούν είναι  $256^3 = 16.777.216$ . Στο σημείο αυτό τονίζεται ότι το ανθρώπινο μάτι μπορεί να διακρίνει "μόλις" 10 εκατομμύρια χρώματα.



(α') Ανάλυση γκριζων αποχρώσεων με  $n = 2$



(β') Ανάλυση γκριζων αποχρώσεων με  $n = 4$



(γ') Ανάλυση γκριζων αποχρώσεων με  $n = 6$



(δ') Ανάλυση γκριζων αποχρώσεων με  $n = 8$

Σχήμα 2.9: Ανάλυση αποχρώσεων του γκριζου για διάφορα μήκη ψηφιολέξης.

## 2.4.2 Το μοντέλο RGB

Το μοντέλο αυτό είναι ένα προσθετικό μοντέλο χρώματος στο οποίο όλα τα χρώματα παράγονται από έναν συνδυασμό των χρωμάτων του κόκκινου, του πράσινου και του μπλε. Συνεπώς κάθε χρώμα μπορεί να παρασταθεί από ένα σύνολο τριών αριθμών που παραστούν τις τιμές των βασικών χρωμάτων που φτιάχνουν το τελικό χρώμα ως εξής:  $(r, g, b)$ . Οι τιμές των  $r, g, b$  μπορεί να δίνονται είτε μέσα στο εύρος τιμών της ψηφιολέξης δηλαδή από 0-255 είτε με την μορφή ποσοστών, δηλαδή από 0-1 για κάθε χρώμα.

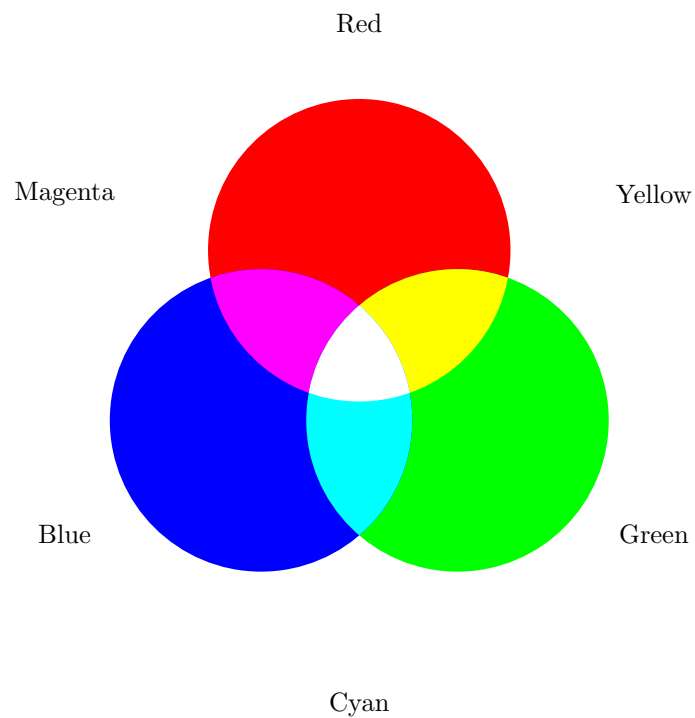
Όταν υπάρχει πλήρης ανάμιξη μόνο των δυο βασικών χρωμάτων, τότε τα χρώματα που προκύπτουν ονομάζονται δευτερεύοντα και αυτά είναι το μωβ, το κίτρινο και το κυανό. Ο χρωματικός χώρος του μοντέλου RGB είναι ένας κύβος όπου κάθε άξονας είναι και ένα βασικό χρώμα. Στο σημείο αυτό σημειώνεται πως το μοντέλο RGB δεν είναι απόλυτο αλλά εξαρτάται από εκάστοτε συσκευή απεικόνισης, δηλαδή διαφορετικές συσκευές απεικονίζουν διαφορετικά το ίδιο χρώμα.

## 2.4.3 Τα μοντέλα HSV και HSL

Αυτά τα μοντέλα χρώματος μεταμορφώνουν τον χρωματικό χώρο του μοντέλου RGB από κύβο σε κύλινδρο δηλαδή από καρτεσιανές συντεταγμένες σε κυλινδρικές. Στο μοντέλο HSV οι τιμές που ορίζουν ένα χρώμα είναι η χροιά (Hue), ο κορεσμός



Σχήμα 2.10: Ανάλυση χρωματικών αποχρώσεων στο μοντέλο χρώματος RGB.



Σχήμα 2.11: Ανάμιξη βασικών χρωμάτων στο μοντέλο RGB.



(α') Αρχική εικόνα



(β') Κόκκινο κανάλι



(γ') Πράσινο κανάλι



(δ') Μπλε κανάλι

Σχήμα 2.12: Έγχρωμη φωτογραφία και τα RGB κανάλια της.

(Saturation), και η τιμή (Value). Ενώ στο μοντέλο HSL η τιμή αντικαθίσταται με την φωτεινότητα (Lightness).

Για κάθε κύλινδρο η γωνία γύρω από τον άξονα του κυλίνδρου αντιστοιχεί στην χροιά, η ακτίνα του κυλίνδρου αντιστοιχεί στον κορεσμό του χρώματος ενώ η απόσταση κατά τον άξονα του κυλίνδρου αντιστοιχεί στην τιμή ή στην φωτεινότητα ανάλογα με το μοντέλο που περιγράφεται.

Στο σημείο αυτό τονίζεται πως ενώ η χροιά αναφέρεται στο ίδιο χαρακτηριστικό, ο ορισμός του κορεσμού είναι διαφορετικός, διότι τα μοντέλα HSV και HSL είναι μετασχηματισμοί των αντίστοιχων μη απόλυτων μοντέλων RGB. Συνεπώς κάθε συσκευή που χρησιμοποιεί το μοντέλο RGB έχει τα δικά της συνοδευτικά μοντέλα HSV και HSL. Από τα παραπάνω προκύπτει πως μια αριθμητική τιμή δεν αναπαριστά το ίδιο χρώμα σε δυο διαφορετικά μοντέλα HSV ή HSL.

Τα βασικά χρώματα ξεκινούν από τις  $0^\circ$  με το κόκκινο. Συνεχίζουν με τις  $120^\circ$  όπου βρίσκεται το πράσινο ενώ το μπλε είναι στις  $240^\circ$ . Στον άξονα του κυλίνδρου υπάρχουν ουδέτερα ή γκριζα χρώματα τα οποία είναι το μαύρο για τιμή ή φωτεινότητα ίση με μηδέν και το άσπρο για τιμή ή φωτεινότητα ίση με ένα. Τα βασικά και τα δευτερεύοντα χρώματα καθώς και οι αναμίξεις τους βρίσκονται στην περιφέρεια του κυλίνδρου. Όμως η βασική διαφορά των δυο μοντέλων είναι ότι αυτό συμβαίνει για την τιμή ένα στο μοντέλο HSV ενώ για το μοντέλο HSL για τιμή φωτεινότητας  $\frac{1}{2}$ .

Άμεση συνέπεια των παραπάνω είναι ότι στο μοντέλο HSL οι κάτω και πάνω επιφάνειες είναι χρώματος μαύρο και άσπρο αντίστοιχα, ενώ στο μοντέλο HSV η κάτω επιφάνεια έχει μαύρο χρώμα αλλά η πάνω επιφάνεια έχει άσπρο κέντρο με τα βασικά χρώματα και τις αναμίξεις τους στην περιφέρεια. Επιπρόσθετα σημαίνει πως για το μοντέλο HSV οι αναμίξεις των χρωμάτων με το άσπρο γίνονται ακτινικά στο άνω επίπεδο ενώ για το μοντέλο HSL οι αναμίξεις γίνονται περιφερειακά κατά μήκος του άξονα αλλά μόνο για το άνω μισό μήκος αυτού. Το ίδιο ισχύει και για τις αναμίξεις του μαύρου και στα δύο μοντέλα αλλά μόνο για το κάτω μισό του κυλίνδρου. Η μετατροπή του μοντέλου RGB στα μοντέλα HSV και HSL γίνεται ως εξής. Για την χροιά:

$$H = \arctan(\beta, \alpha) \quad (2.15)$$

όπου:

$$\alpha = \frac{1}{2}(2r - g - b) \quad \beta = \frac{\sqrt{3}}{2}(g - b) \quad (2.16)$$

Η φωτεινότητα είναι:

$$L = \frac{1}{2}(\max(r, g, b) + \min(r, g, b)) \quad (2.17)$$

Και η τιμή είναι:

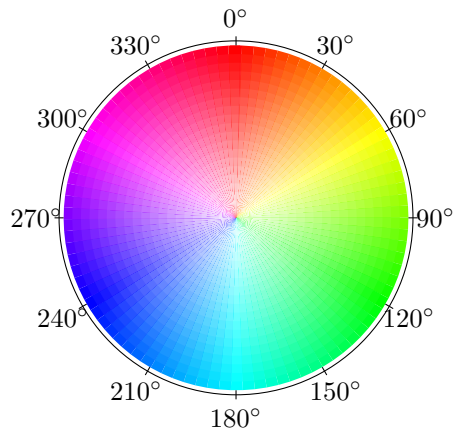
$$V = \max(r, g, b) \quad (2.18)$$

Τέλος ο κορεσμός είναι:

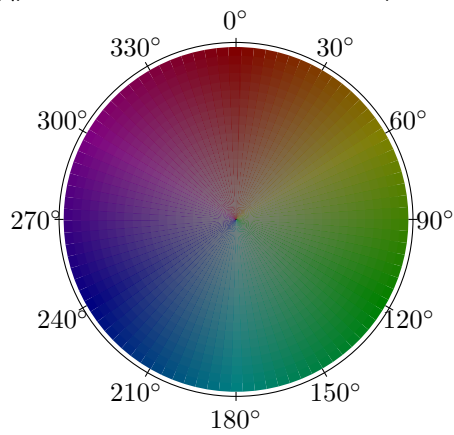
$$S_{HSV} = \begin{cases} 0 & \text{εάν } C = 0 \\ \frac{C}{V} & \text{εάν } C \neq 0 \end{cases} \quad S_{HSL} = \begin{cases} 0 & \text{εάν } C = 0 \\ \frac{C}{1-|2L-1|} & \text{εάν } C \neq 0 \end{cases} \quad (2.19)$$

όπου  $C = \max(r, g, b) - \min(r, g, b)$

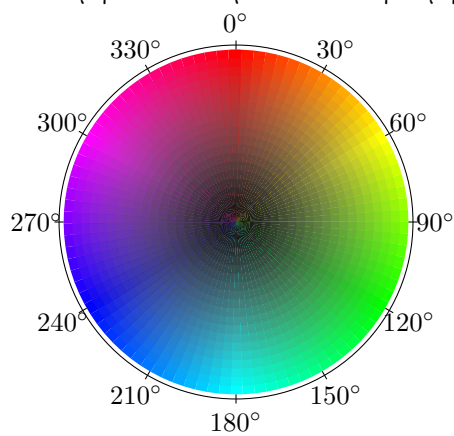




Σχήμα 2.13: Άνω επίπεδο του κυλίνδρου HSV.



Σχήμα 2.14: Τομή του κυλίνδρου HSV στην τιμή  $V = 0.5$ .



Σχήμα 2.15: Τομή του κυλίνδρου HSL στην τιμή  $L = 0.5$ .



(α') Αρχική εικόνα



(β') Κανάλι H



(γ') Κανάλι S



(δ') Κανάλι V

Σχήμα 2.16: Έγχρωμη φωτογραφία και τα HSV κανάλια της.



(α') Αρχική εικόνα



(β') Κανάλι H



(γ') Κανάλι S



(δ') Κανάλι L

Σχήμα 2.17: Έγχρωμη φωτογραφία και τα HSL κανάλια της.

## 2.5 Επεξεργασία φωτογραφίας και φίλτρα

Επόμενο κομμάτι μετά την λήψη των δεδομένων είναι η επεξεργασία τους. Η επεξεργασία αυτή αποτελείται από τρία στάδια, την προεπεξεργασία, επεξεργασία και μεταεπεξεργασία. Τα όρια μεταξύ τους είναι θολά αλλά γενικά μπορούν να χωριστούν σε κομμάτια όπου η εικόνα ετοιμάζεται για επεξεργασία όπως π.χ αφαίρεση θορύβου, κυρίως επεξεργασία όπου θα μπορούσαν να ανιχνεύονται χρώματα ή μοτίβα και τέλος στο στάδιο της μεταεπεξεργασίας όπου αυτά παρουσιάζονται στον υπολογιστή.

### 2.5.1 Αποπαραμόρφωση εικόνας

Σε προηγούμενες ενότητες παρουσιάστηκε ο τρόπος με τον οποίο αποτυπώνονται φυσικά αντικείμενα στον αισθητήρα όπου έγινε εισαγωγή του χαρακτηριστικού μητρώου της κάμερας, αλλά και το πως εισάγονται οπτικές παραμορφώσεις από τους φακούς. Ένα σημαντικό κομμάτι το οποίο ωστόσο δεν είναι πάντα απαραίτητο είναι το κομμάτι της αποπαραμόρφωσης. Η αποπαραμόρφωση είναι απαραίτητη όταν οι παραμορφώσεις είναι σημαντικές, δηλαδή όταν χρησιμοποιούνται ευρυγώνιοι φακοί.

Βασικό ζήτημα στο πρόβλημα της αποπαραμόρφωσης είναι η μη γνώση του χαρακτηριστικού μητρώου της κάμερας. Αυτό επιλύεται βάζοντας την κάμερα να απεικονίσει ένα γνωστό επίπεδο το οποίο έχει ένα μοτίβο. Συνήθως το επίπεδο αυτό είναι μια σκακιέρα. Ο τρόπος με τον οποίο γίνεται η αποπαραμόρφωση βασίζεται σε μια μικρή μαθηματική θεμελίωση η οποία χρησιμοποιείται όχι μόνο στον τομέα της τεχνητής όρασης, αλλά συναντάται και στα θέματα της ρομποτικής τα οποία αναπτύσσονται σε επόμενες ενότητες.

#### Περιστροφή και μεταφορά διανύσματος

Όπως είναι γνωστό ένα διάνυσμα περιγράφεται από τρεις συντεταγμένες και γράφεται σε μητρική μορφή ως ένας πίνακας στήλης:

$$r = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (2.20)$$

Εύκολα αποδεικνύεται πως η περιστροφή του διανύσματος γύρω από τους άξονες  $x, y, z$  μπορεί να περιγραφεί ως ένα γινόμενο του διανύσματος αυτού με έναν πίνακα του οποίου τα στοιχεία εξαρτώνται από τον άξονα ως προς τον οποίο περιστρέφεται το διάνυσμα. Για περιστροφή του διανύσματος γύρω από τους  $x, y, z$  κατά  $\psi, \phi, \theta$  αντίστοιχα οι πίνακες περιστροφής είναι:

$$R_x(\psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix} \quad (2.21)$$

$$R_y(\phi) = \begin{bmatrix} \cos(\phi) & 0 & -\sin(\phi) \\ 0 & 1 & 0 \\ \sin(\phi) & 0 & \cos(\phi) \end{bmatrix} \quad (2.22)$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.23)$$

Αντίστοιχα η μεταφορά διανύσματος η οποία αναπαριστά την αλλαγή του συστήματος συντεταγμένων σε ένα άλλο το οποίο δεν έχει περιστραφεί αλλά έχει μετατοπισθεί μπορεί να παρασταθεί απλά ως το διάνυσμα θέσης του νέου συστήματος συντεταγμένων στο παλιό σύστημα και συμβολίζεται με  $b$ . Έτσι οι συντεταγμένες ενός καινούριου σημείου  $P_n$  το οποίο έχει περιστραφεί και μετατοπισθεί σε σχέση με το  $P_o$  δίνονται από την σχέση:

$$P_n = R(P_o - b) \quad (2.24)$$

### Επίλυση σχέσεων και αποπαραμόρφωση

Συνδυάζοντας την σχέση 2.24 με τις σχέσεις μοντελοποίησης του οπτικού σφάλματος της κάμερας διαμορφώνεται ένα βασικό σύστημα εξισώσεων το οποίο επιλύεται ώστε να είναι γνωστές όλες οι παράμετροι και η εικόνα να μπορεί να αποπαραμορφωθεί.

Η περιστροφή περιγράφεται από τρεις γωνίες και η μεταφορά από τρεις αριθμούς. Επιπρόσθετα το χαρακτηριστικό μητρώο της κάμερας έχει τέσσερις αγνώστους έχοντας έτσι ένα σύστημα  $10 \times 10$ . Ωστόσο εάν η εστίαση της κάμερας δεν αλλάζει, το χαρακτηριστικό μητρώο της κάμερας παραμένει σταθερό μεταξύ διαδοχικών φωτογραφήσεων. Κάθε φωτογράφιση του επίπεδου μοτίβου δίνει οκτώ παραμέτρους, έτσι εισάγεται ο περιορισμός ότι πρέπει να γίνει φωτογράφιση τουλάχιστον δυο φορές υπό διαφορετική γωνία και απόσταση.

Ένα εργαλείο που βοηθά στην επίλυση του συστήματος είναι η επιφανειακή ομογραφία η οποία είναι ένας προβολικός μετασχηματισμός από ένα επίπεδο σε ένα άλλο. Εν προκειμένω τα επίπεδα αυτά είναι το επίπεδο του μοτίβου και το επίπεδο αποτύπωσης της κάμερας. Για λόγους ευκολίας ο μετασχηματισμός γράφεται σε μητρωική μορφή με χρήση ομογενών συντεταγμένων. Για  $\vec{q}$  σημείο στο επίπεδο αποτύπωσης και  $\vec{Q}$  σημείο στο επίπεδο του μοτίβου. Ισχύει:

$$\vec{q} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \vec{Q} = \begin{bmatrix} Z \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.25)$$

Και ο μετασχηματισμός περιγράφεται ως:

$$\vec{q} = sH\vec{Q} \quad (2.26)$$

Όπου  $s$  ένας αυθαίρετος συντελεστής μεγέθυνσης. Μια σημαντική ιδιότητα του μητρώου  $H$  είναι ότι αποτελείται από δυο κομμάτια. Το κομμάτι που είναι το άθροισμα περιστροφής και μετακίνησης που τοποθετεί το μοτίβο στο επίπεδο αποτύπωσης και το κομμάτι της προβολής το οποίο εισάγει το χαρακτηριστικό μητρώο της κάμερας. Έτσι ο μετασχηματισμός γράφεται:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \quad (2.27)$$

Στο σημείο αυτό τονίζεται ότι δεν είναι σημαντικό το σημείο στον χώρο  $Q$  αλλά το σημείο  $Q'$  που βρίσκεται πάνω στο επίπεδο μοτίβο. Για λόγους ευκολίας το

επίπεδο αυτό ορίζεται ότι βρίσκεται στο  $Z = 0$ . Και ο μετασχηματισμός έρχεται στην μορφή:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = s \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix} \begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} \quad (2.28)$$

με βάση τις παραπάνω σχέσεις είναι δυνατόν να υπολογιστεί το μητρώο ομογραφίας που είναι το γινόμενο:

$$H = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ r_{31} & r_{32} & t_z \end{bmatrix}$$

Για τον σκοπό αυτό χρησιμοποιούνται πολλές φωτογραφήσεις υπό διαφορετικές γωνίες για τον υπολογισμό των περιστροφών, μετατοπίσεων και των στοιχείων του χαρακτηριστικού μητρώου (τα οποία παραμένουν σταθερά). Όπως έχει προαναφερθεί κάθε θέση του μοτίβου περιγράφεται από 6 παραμέτρους δηλαδή 6 αγνώστους για κάθε διαφορετική θέαση. Ωστόσο ένα γνωστό επίπεδο αντικείμενο όπως το μοτίβο σακατιέρας, δίνει 8 εξισώσεις αφού ο μετασχηματισμός τετραγώνου σε τετράπλευρο περιγράφεται με 4 σημεία  $(x, y)$ . Έτσι κάθε θέαση δίνει 8 νέες εξισώσεις και 6 νέους αγνώστους. Με αυτόν τον τρόπο είναι δυνατός ο υπολογισμός όλων των αρχικών αγνώστων.

Πρακτικά η επίλυση γίνεται θεωρώντας πως η κάμερα δεν εισάγει οπτικές παραμορφώσεις. Για κάθε θέαση υπολογίζεται το μητρώο  $H$ . Το  $H$  μπορεί να θεωρηθεί ότι αποτελείται από 3 μητρώα στήλες ως εξής:

$$H = [h_1 \ h_2 \ h_3] \quad (2.29)$$

Όμως:

$$H = sM[r_1 \ r_2 \ t] \quad (2.30)$$

Εξισώνοντας και λύνοντας ως προς  $r_1, r_2, t$  προκύπτουν οι εκφράσεις:

$$r_1 = \frac{M^{-1}h_1}{s} \quad (2.31)$$

$$r_2 = \frac{M^{-1}h_2}{s} \quad (2.32)$$

$$t = \frac{M^{-1}h_3}{s} \quad (2.33)$$

τα  $r_1, r_2$  είναι μοναδιαία διανύσματα σε διαφορετικούς άξονες από το οποίο προκύπτει:

$$r_1^T r_2 = 0 \quad (2.34)$$

όμως γενικά ισχύει πως  $(ab)^T = b^T a^T$  και έτσι προκύπτει ο πρώτος περιορισμός.

$$h_1^T (M^{-1})^T M^{-1} h_2 = 0 \quad (2.35)$$

Επιπρόσθετα τα μέτρα τους είναι ίσα δηλαδή:

$$|r_1| = |r_2| \rightarrow r_1^T r_1 = r_2^T r_2 \quad (2.36)$$

και με αντικατάσταση προκύπτει ο δεύτερος περιορισμός:

$$h_1^T (M^{-1})^T M^{-1} h_1 = h_2^T (M^{-1})^T M^{-1} h_2 \quad (2.37)$$

αφού οι εκφράσεις (και όχι οι τιμές) των στοιχείων του  $M$  είναι γνωστές, τότε η έκφραση μπορεί να υπολογισθεί και είναι ίση με:

$$(M^{-1})^T M^{-1} = B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{12} & B_{22} & B_{23} \\ B_{13} & B_{23} & B_{33} \end{bmatrix} = \begin{bmatrix} \frac{1}{f_x^2} & 0 & \frac{-c_x}{f_x^2} \\ 0 & \frac{1}{f_y^2} & \frac{-c_y}{f_y^2} \\ \frac{-c_x}{f_x^2} & \frac{-c_y}{f_y^2} & \frac{c_x^2}{f_x^2} + \frac{c_y^2}{f_y^2} + 1 \end{bmatrix}$$

Οι περιορισμοί είναι της μορφής  $h_i^T B h_j$  και κάνοντας τις πράξεις προκύπτει πως

$$h_i^T B h_j = v_{ij}^T b = \begin{bmatrix} h_{i1} h_{j1} \\ h_{i1} h_{j2} + h_{i2} h_{j1} \\ h_{i2} h_{j2} \\ h_{i3} h_{j1} + h_{i1} h_{j3} \\ h_{i3} h_{j2} + h_{i2} h_{j3} \\ h_{i3} h_{j3} \end{bmatrix}^T \begin{bmatrix} B_{11} \\ B_{12} \\ B_{22} \\ B_{13} \\ B_{23} \\ B_{33} \end{bmatrix}^T \quad (2.38)$$

έτσι οι δυο περιορισμοί μπορούν να γραφούν στην μορφή

$$\begin{bmatrix} v_{12}^T \\ (v_{11} - v_{22})^T \end{bmatrix} b = 0 \quad (2.39)$$

εάν τώρα ληφθούν  $n$  εικόνες του μοτίβου, τότε είναι δυνατόν γράφουν όλες οι εξισώσεις μαζί στην μορφή:

$$V b = 0 \quad (2.40)$$

Όπου το  $V$  είναι ένα μητρώο  $2n \times 6$  και για  $n \geq 2$  η σχέση μπορεί να λυθεί ως προς  $b$ . Με γνωστό το μητρώο  $b$  είναι δυνατή η εύρεση των παραμέτρων της κάμερας

$$f_x = \sqrt{\frac{\lambda}{B_{11}}} \quad (2.41)$$

$$f_y = \sqrt{\frac{\lambda B_{11}}{(B_{11} B_{22} - B_{12}^2)}} \quad (2.42)$$

$$c_x = -B_{13} \frac{f_x^2}{\lambda} \quad (2.43)$$

$$c_y = \frac{B_{12} B_{13} - B_{11} B_{23}}{B_{11} B_{12} - B_{11}^2} \quad (2.44)$$

όπου  $\lambda$ :

$$\lambda = B_{33} - \frac{B_{13}^2 + c_y (B_{12} B_{13} - B_{11} B_{23})}{B_{11}} \quad (2.45)$$

από εκεί και πέρα προκύπτουν και τα διανύσματα περιστροφής και μεταφοράς

$$r_1 = \lambda M^{-1} h_1 \quad (2.46)$$

$$r_2 = \lambda M^{-1} h_2 \quad (2.47)$$

$$r_3 = r_1 \times r_2 \quad (2.48)$$

$$t = \lambda M^{-1} h_3 \quad (2.49)$$

εδώ το  $\lambda$  υπολογίζεται από την σχέση:

$$\lambda = \frac{1}{|M^{-1}h_1|} \quad (2.50)$$

Παρόλα αυτά, δεν έχουν ληφθεί υπόψιν ακόμα οι οπτικές παραμορφώσεις που εισάγουν οι φακοί. Έτσι εάν το σημείο για τέλεια κάμερα είναι  $(x_p, y_p)$  και για παραμορφώσεις μετατοπίζονται στις θέσεις  $(x_d, y_d)$  τότε ισχύει:

$$\begin{bmatrix} x_p \\ y_p \end{bmatrix} = (1 + \kappa_1 r^2 + \kappa_2 r^4 + \kappa_3 r^6) \begin{bmatrix} x_d \\ y_d \end{bmatrix} + \begin{bmatrix} 2p_1 x_d y_d + p_2 (r^2 + 2x_d^2) \\ p_1 (r^2 + 2y_d^2) + 2p_2 x_d y_d \end{bmatrix}$$

Στο σχήμα 2.18 φαίνεται η αρχική εικόνα αλλά και το αποτέλεσμα της αποπαραμόρφωσης.

### 2.5.2 Συνέλιξη

Στην επεξεργασία εικόνας η συνέλιξη είναι στην ουσία ένας σταθμισμένος μέσος όρος. Τα βάρη του μέσου όρου δίνονται σε έναν πίνακα τιμών συχνά πολύ μικρότερων διαστάσεων από την επεξεργαζόμενη εικόνα. Κάθε διαφορετικό σετ τιμών αποτελεί και ένα διαφορετικό φίλτρο επεξεργασίας. Στην συνέχεια ο πίνακας αυτός παρατίθεται πάνω στην εικόνα και υπολογίζεται ο σταθμισμένος μέσος όρος. Το αποτέλεσμα αποθηκεύεται σε ένα σημείο το οποίο ονομάζεται και σημείο αγκίστρωσης. Ο πίνακας των συντελεστών αλλά και η θέση του σημείου αγκίστρωσης θεωρούνται ως ένας τελεστής που ονομάζεται πυρήνας. Έτσι για έναν πυρήνα  $K(i, j)$  μεγέθους  $m \times n$  ο οποίος παρατίθεται σε μια εικόνα  $F(x, y)$ , η συνέλιξη τους, δημιουργεί μια νέα εικόνα  $R(x, y)$  η οποία έχει τιμές:

$$R(x, y) = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} F(x+i-a_i, y+j-a_j) K(i, j) \quad (2.51)$$

όπου  $(a_i, a_j)$  το σημείο αγκίστρωσης σε συντεταγμένες του πυρήνα. Μια τυπική μορφή πυρήνα φαίνεται στο σχήμα 2.19 ενώ η παράθεση του πάνω σε μια εικόνα φαίνεται στο σχήμα 2.20.

### 2.5.3 Θόλωση εικόνας - Blurring

Η θόλωση εικόνας είναι μια συχνά χρησιμοποιούμενη διαδικασία επεξεργασίας. Υπάρχουν πολλοί λόγοι για τους οποίους χρησιμοποιείται η θόλωση εικόνας, όπως για παράδειγμα η μείωση θορύβου δηλαδή η εξισορρόπηση μεγάλων διαφορών μεταξύ γειτονικών pixels, αλλά και διεργασίες όπως είναι η αλλαγή μεγέθους (scaling, resize) εικόνας. Στην συνέχεια παρουσιάζονται οι δύο από τις πιο συχνά συναντώμενες μορφές θόλωσης.

#### Θόλωση με χρήση μέσου όρου

Η θόλωση μέσου όρου με ένα πυρήνα μεγέθους  $m \times n$  δίνεται από την σχέση:

$$R(x, y) = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} F(x+i-a_i, y+j-a_j) \quad (2.52)$$





(α) Αρχική παραμορφωμένη εικόνα

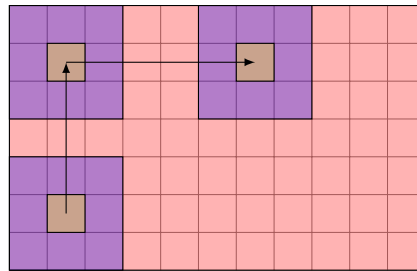


(β) Αρχική επεξεργασμένη εικόνα μετά από αποπαραμόρφωση

Σχήμα 2.18: Φωτογραφία πριν ρύθμιση της κάμερας και στην συνέχεια αποπαραμορφωμένη με γνωστές παραμέτρους κάμερας

$w_0$	$w_1$	$w_2$
$w_3$	$w_4$	$w_5$
$w_6$	$w_7$	$w_8$

Σχήμα 2.19: Πυρήνας συνέλιξης και τα αντίστοιχα βάρη του. Το κίτρινο κελί αποτελεί το σημείο αγκίστρωσης.



Σχήμα 2.20: Παράθεση πυρήνα πάνω στην επεξεργαζόμενη εικόνα

Η παραπάνω σχέση εκφράζει τη συνέλιξη με έναν πυρήνα του οποίου τα στοιχεία είναι ίσα με ένα. Αυτού του τύπου η θόλωση δεν αποτελεί ικανοποιητικό μοντέλο καθώς το αποτέλεσμα δεν μοιάζει με φωτογραφία κάμερας με κακή εστίαση.

Αυτό μπορεί να εξηγηθεί και αλλιώς. Έστω μια εικόνα η οποία έχει μηδενικά στοιχεία δηλαδή μαύρα pixels παντού εκτός από ένα στο κέντρο το οποίο είναι άσπρου χρώματος δηλαδή έχει μοναδιαία τιμή<sup>2</sup>. Εάν γίνει θόλωση η εικόνα θα έχει στο κέντρο ένα pixel όχι άσπρο, αλλά μιας απόχρωσης γκριζου. Ωστόσο αυτό δεν είναι το αποτέλεσμα μιας μη εστιασμένης κάμερας. Το σωστό αποτέλεσμα θα ήταν ένα κυκλικό σχήμα περισσότερο άσπρο στο κέντρο από τα άκρα αλλά και ομαλή μετάβαση από την άσπρη στην μαύρη περιοχή.

### Θόλωση Gauss

Η θόλωση Gauss πραγματοποιείται όπως η κανονική θόλωση, όμως σε αυτή την περίπτωση τα βάρη δίνονται από την συνάρτηση ή κατανομή Gauss η οποία δίνεται από την σχέση:

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.53)$$

όπου  $\sigma$  είναι η τυπική απόκλιση της κατανομής και οι μονάδες μέτρησης είναι pixels. Το όνομα της θόλωσης προέρχεται από την κατανομή η οποία είναι κανονική μίας 2D μεταβλητής. Από την μορφή του πυρήνα προκύπτει πως το κεντρικό pixel λαμβάνεται περισσότερο υπόψιν και άρα επιτυγχάνεται το επιθυμητό αποτέλεσμα που περιγράφηκε στην θόλωση μέσου όρου. Το σημείο που πρέπει να προσεχθεί είναι πως εάν το  $\sigma$  είναι πολύ μικρό, τότε η θόλωση έχει πολύ μικρό αντίκτυπο, ενώ για μεγαλύτερες τιμές η θόλωση τείνει να ευνοήσει περισσότερο τα γειτονικά pixels

<sup>2</sup>Τιμές σε μορφή ποσοστού και όχι σε διάστημα 0-255

με αποτέλεσμα ο θόρυβος να εξαφανίζεται με κόστος κάποια απώλεια ευκρίνειας. Ωστόσο, εάν ο πυρήνας έχει πολύ μεγάλη τυπική απόκλιση και μεν ο θόρυβος εξαφανίζεται, αλλά με ταυτόχρονη απόρριψη μιας μεγάλης ποσότητας χρήσιμης πληροφορίας. Στο σχήμα 2.21 φαίνονται οι τιμές του πυρήνα για διάφορα  $\sigma$ .

#### 2.5.4 Μορφολογία εικόνας

Οι μορφολογικοί μετασχηματισμοί χρησιμοποιούνται σε διάφορα πλαίσια επεξεργασίας όπως είναι η αφαίρεση θορύβου, η απομόνωση στοιχείων ή η ένωση απομονωμένων στοιχείων. Οι δύο βασικές μορφές είναι η διαστολή και η διάβρωση. Και οι δύο είναι συνελίξεις με έναν πυρήνα ο οποίος παρατίθεται σταδιακά σε όλη την εικόνα. Στην διαστολή το pixel στο σημείο αγκίστρωσης αντικαθίσταται από την μέγιστη τιμή που υπάρχει στο πεδίο της εικόνας που καλύπτει ο πυρήνας. Αντίστροφα στην διάβρωση αντικαθίσταται από την ελάχιστη τιμή. Δηλαδή η διαστολή και η διάβρωση είναι αντίστοιχα  $\max$  και  $\min$  τελεστές.

Συχνά οι παραπάνω μορφολογικές μετατροπές χρησιμοποιούνται σε δυαδικές εικόνες. Οι δυαδικές εικόνες είναι εικόνες οι οποίες έχουν ένα κανάλι και οι τιμές που λαμβάνουν τα pixels τους είναι μόνο δυο. Τέτοιες εικόνες προκύπτουν από την διαδικασία του διαχωρισμού η οποία αναφέρεται στην συνέχεια. Η χρήση λοιπόν της διαστολής και της διάβρωσης μπορεί να μειώσει τον θόρυβο σε αυτές τις δυαδικές εικόνες σε περιπτώσεις όπου υπάρχουν μεμονωμένα pixels.

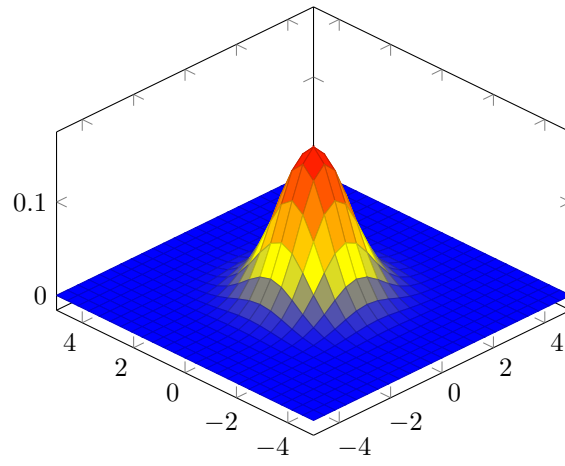
#### 2.5.5 Διαχωρισμός - Threshold

Ο διαχωρισμός είναι μια διαδικασία κατά την οποία τα pixel τοποθετούνται σε δύο κατηγορίες τιμών με κριτήριο μια ορισμένη ανώτατη τιμή χρώματος. Αυτό έχει ως αποτέλεσμα τις περισσότερες φορές την παραγωγή δυαδικών εικόνων ανάλογα με τον τύπο του διαχωρισμού που χρησιμοποιείται.

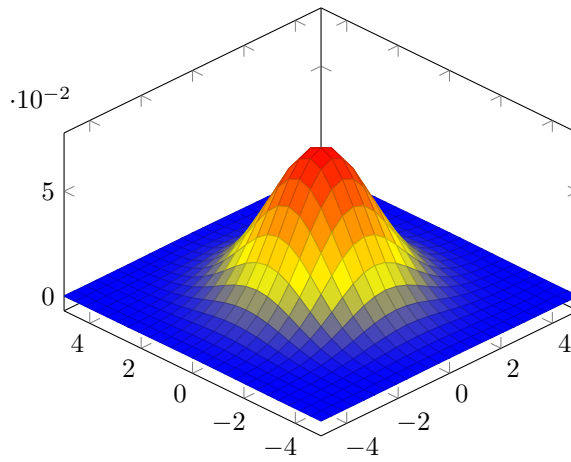
Η βασική μορφή είναι αυτή του δυαδικού διαχωρισμού όπου, όποιες τιμές βρίσκονται πάνω από την τιμή διαχωρισμού λαμβάνουν μια αυθαίρετη αλλά ίδια για όλες τιμή. Ενώ όσες είναι μικρότερες μηδενίζονται. Στον *ανάστροφο* δυαδικό διαχωρισμό μηδενίζονται όσες τιμές βρίσκονται πάνω από την τιμή διαχωρισμού. Στον *διαχωρισμό αποκοπής* όσες τιμές βρίσκονται πάνω από την τιμή διαχωρισμού, εξισώνονται με αυτήν ενώ οι υπόλοιπες αφήνονται αμετάβλητες. Στον *μηδενικό διαχωρισμό αποκοπής* όσες τιμές βρίσκονται κάτω από την τιμή διαχωρισμού μηδενίζονται και οι υπόλοιπες δεν επηρεάζονται. Ενώ τέλος στον *ανάστροφο μηδενικό διαχωρισμό αποκοπής* όσες τιμές βρίσκονται πάνω από την τιμή διαχωρισμού μηδενίζονται και οι υπόλοιπες αφήνονται αμετάβλητες. Στο σχήμα 2.23 φαίνεται η τροποποίηση ενός αρχικού σήματος για τα διάφορα είδη διαχωρισμού.

#### 2.5.6 Παράγωγοι εικόνας - Εύρεση ακμών

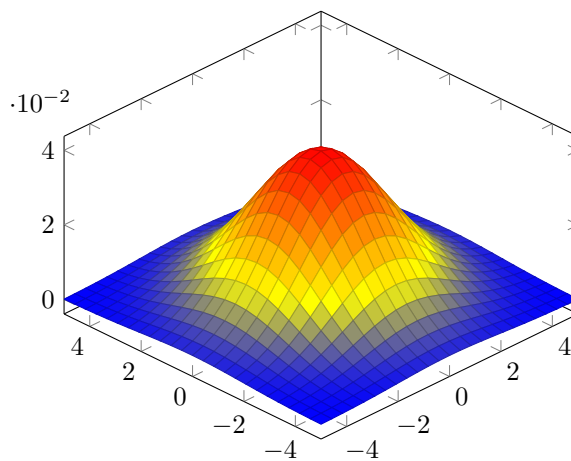
Ένα από τα βασικότερα και σημαντικότερα προβλήματα της τεχνητής όρασης είναι αυτό της εύρεσης παραγώγων. Και εδώ για την λύση του προβλήματος χρησιμοποιείται η συνέλιξη. Επιπρόσθετα η παραγωγή χρησιμοποιείται για την εύρεση ακμών καθώς μια ακμή σηματοδοτείται από μια απότομη αλλαγή χρώματος δηλαδή μεγάλη τιμή της παραγώγου.



(α')  $\sigma=1$

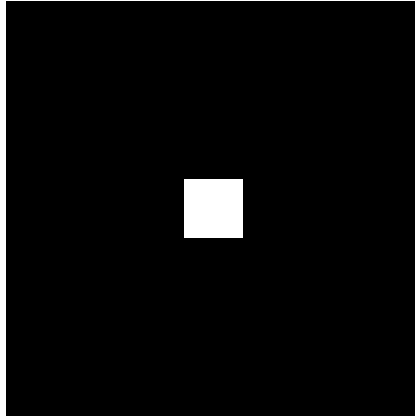


(β')  $\sigma=1.5$

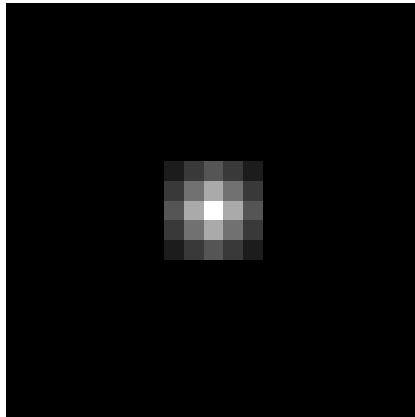


(γ')  $\sigma=2$

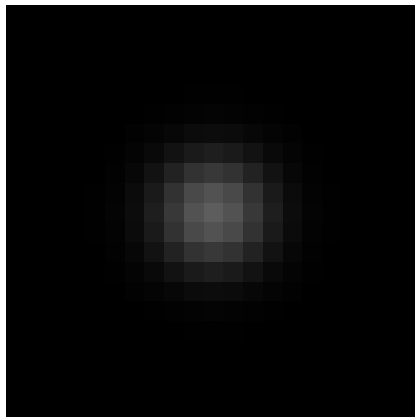
Σχήμα 2.21: Κατανομές Gauss για διάφορες τιμές τυπικής απόκλισης



(α') Αρχική εικόνα

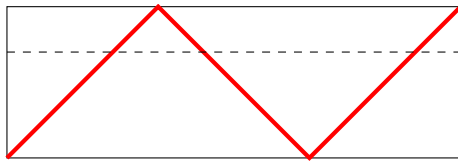


(β') Θόλωση μέσου όρου



(γ') Θόλωση Gauss

Σχήμα 2.22: Σύγκριση θόλωσης μέσου όρου και θόλωσης Gauss



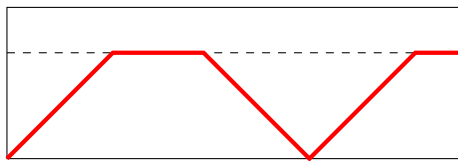
(α) Τιμές pixel και τιμή αποκοπής



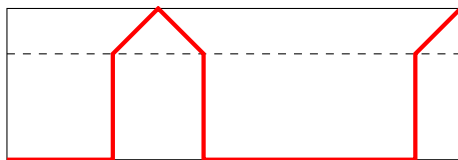
(β) Δυαδικός διαχωρισμός



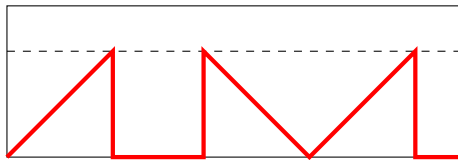
(γ') Ανάστροφος δυαδικός διαχωρισμός



(δ) Διαχωρισμός αποκοπής



(ε') Μηδενικός διαχωρισμός αποκοπής



(στ') Ανάστροφος μηδενικός διαχωρισμός αποκοπής

Σχήμα 2.23: Τα διάφορα είδη διαχωρισμού. Η τιμή αποκοπής ορίζεται από την διακοπτόμενη γραμμή



(α) Πράσινο κανάλι



(β) Τμή αποκοπής 100



(γ) Τμή αποκοπής 127



(δ) Τμή αποκοπής 180

Σχήμα 2.24: Δυαδικός διαχωρισμός για διάφορες τιμές αποκοπής.

### Τελεστής Sobel

Ένας από τους πιο γνωστούς τελεστές για παραγωγή εικόνας είναι ο τελεστής *sobel*. Ο τελεστής *sobel* χωρίζεται σε δύο ξεχωριστούς τελεστές, ο ένας για την οριζόντια διεύθυνση και ο δεύτερος για την κατακόρυφη διεύθυνση και παρατείνονται διαδοχικά στην εικόνα. Οι τελεστές αυτοί δίνονται από τις παρακάτω σχέσεις:

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2.54)$$

### Τελεστής Laplace

Ο τελεστής Laplace ορίζεται ως το άθροισμα των παραγώγων δεύτερης τάξης:

$$\mathcal{L}(f) = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (2.55)$$

Ένας συνήθης τρόπος υλοποίησης του τελεστή Laplace είναι με χρήση δύο διαδοχικών τελεστών *sobel* ανά κατεύθυνση.

Ο τελεστής Laplace έχει διάφορα σενάρια χρήσης. Μια τυπική εφαρμογή είναι η ανίχνευση περιοχών. Καθώς οι μικρές περιοχές που περιβάλλονται από μεγαλύτερες τιμές τείνουν να μεγιστοποιήσουν την συνάρτηση. Ενώ στην αντίστροφη περίπτωση να την ελαχιστοποιήσουν. Με αντίστοιχο τρόπο γίνεται και η ανίχνευση ακμών.

### Τελεστής Canny

Ο τελεστής Canny είναι ένας τελεστής ο οποίος βρίσκει ακμές. Η διαφορά με τον τελεστή Laplace βρίσκεται στο γεγονός ότι στον συντελεστή Canny υπολογίζονται οι παράγωγοι κατά  $x$  και  $y$  και στην συνέχεια συνδυάζονται για τον υπολογισμό τεσσάρων παραγώγων κατά κατεύθυνση. Τα σημεία όπου αυτές οι παράγωγοι κατά κατεύθυνση είναι μέγιστες μπορεί να ανήκουν σε ακμές.

Στην συνέχεια συναρμολογούνται οι ακμές εφαρμόζοντας ένα όριο υστέρησης. Αυτό σημαίνει πως υπάρχει ένα άνω και ένα κάτω όριο. Εάν η κλίση σε ένα pixel είναι κάτω από το ορισμένο διάστημα απορρίπτεται. Εάν είναι μέσα στο διάστημα λαμβάνεται υπόψιν μόνο όταν συνορεύει με pixel του οποίου η κλίση είναι μεγαλύτερη του άνω ορίου.

### 2.5.7 Μετασχηματισμοί Hough

Οι μετασχηματισμοί *hough* χρησιμοποιούνται για την εύρεση καμπυλών σε μια εικόνα. Συχνά η εικόνα προεπεξεργάζεται με κάποιον τελεστή εύρεσης ακμών. Για την καλύτερη κατανόηση του αλγορίθμου παρουσιάζεται το απλούστερο παράδειγμα της εύρεσης ευθειών.

Είναι γνωστό ότι μια ευθεία δίνεται από την σχέση:

$$y = ax + b \quad (2.56)$$

Συνεπώς αφού μια γραμμή καθορίζεται από τις παραμέτρους  $(a, b)$ , μπορεί να παρασταθεί ως ένα σημείο σε ένα χώρο παραμέτρων. Ωστόσο για κάθετες γραμμές



οι παράμετροι δεν έχουν πεπερασμένες τιμές. Συνεπώς είναι πιο βολικό η καμπύλη να παρασταθεί παραμετρικά με την σχέση:

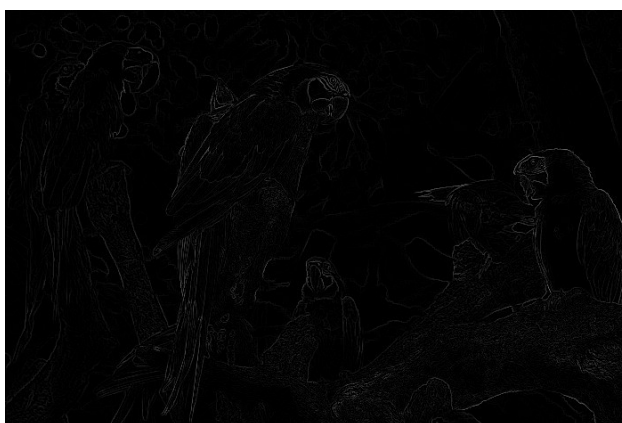
$$\rho = x \cos(\theta) + y \sin(\theta) \quad (2.57)$$

Όπου  $\rho$  είναι η απόσταση και  $\theta$  είναι η γωνία που σχηματίζει η γραμμή της απόστασης με τον άξονα των  $x$ . Με την παραπάνω περιγραφή είναι δυνατή η εύρεση του ζεύγους των παραμέτρων της γραμμής στην οποία ανήκει κάθε μη μηδενικό σημείο. Τα ζεύγη των παραμέτρων τα οποία υποδεικνύονται περισσότερες φορές στον χώρο των παραμέτρων ανήκουν σε ευθείες.

Η αναζήτηση των καμπυλών γίνεται στο πεδίο των παραμέτρων, όπου για ευθείες είναι δισδιάστατο. Όμως για άλλες καμπύλες το πεδίο μπορεί να είναι πολυδιάστατο. Για παράδειγμα για τον κύκλο, οι παράμετροι είναι το κέντρο και η ακτίνα και ο χώρος είναι τρισδιάστατος. Γενικά οποιαδήποτε καμπύλη που μπορεί να παρασταθεί παραμετρικά μπορεί να εντοπισθεί με τους μετασχηματισμούς Hough, ωστόσο για πολυπαραμετρικές καμπύλες ο μετασχηματισμός είναι υπολογιστικά ασύμφορος.



(α') Αρχικό κανάλι

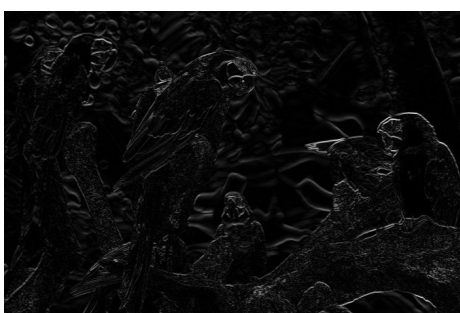


(β') Τελεστής Laplace

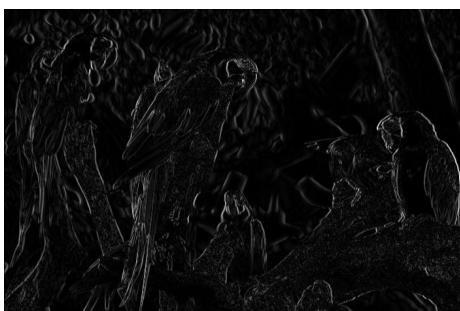
Σχήμα 2.25: Εφαρμογή τελεστή Laplace.



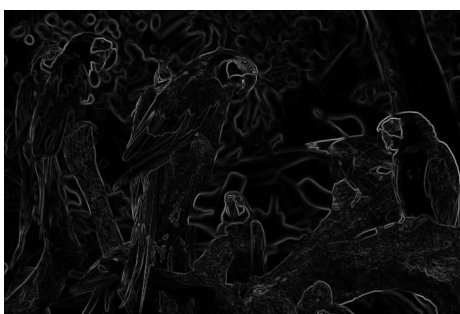
(α') Αρχικό κανάλι



(β') Τελεστής Sobel στην οριζόντια διεύθυνση

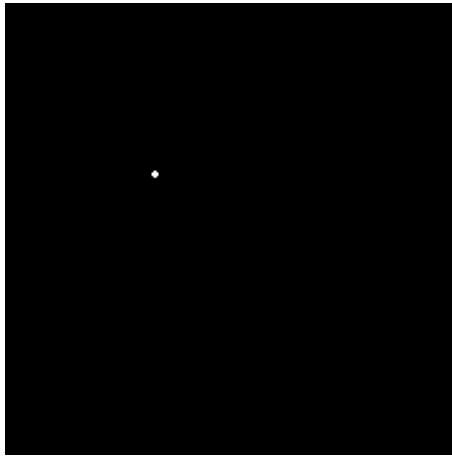


(γ') Τελεστής Sobel στην κατακόρυφη διεύθυνση

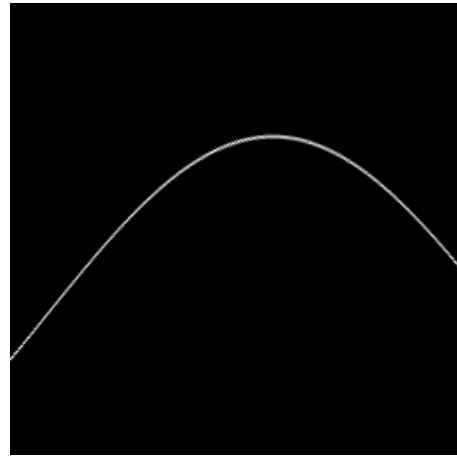


(δ') Άθροισμα οριζόντιας και κατακόρυφης διεύθυνσης

Σχήμα 2.26: Εφαρμογή τελεστών Sobel.



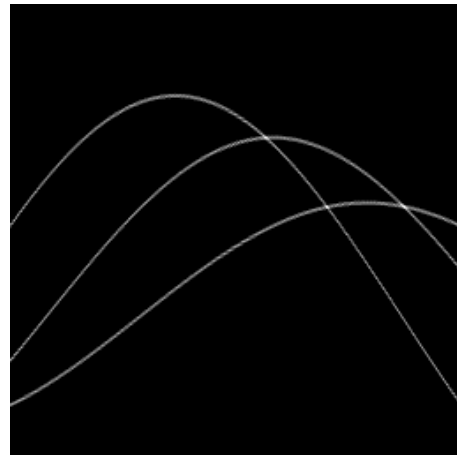
(α') Ένα σημείο



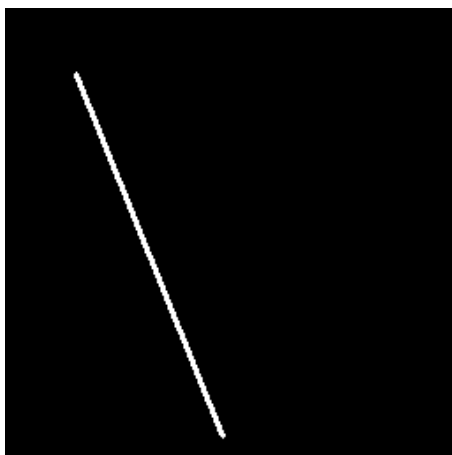
(β') Παραμετρικός χώρος



(γ') Τρία σημεία



(δ') Παραμετρικός χώρος

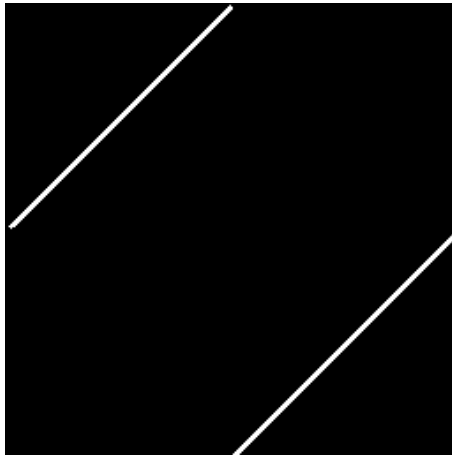


(ε') Ευθεία



(στ') Παραμετρικός χώρος

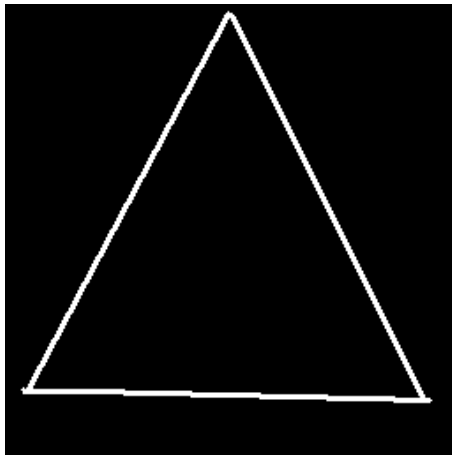
Σχήμα 2.27: Απλές εφαρμογές Hough.



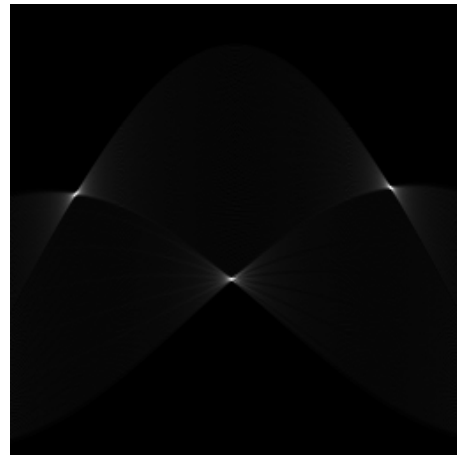
(α') Δύο ευθείες



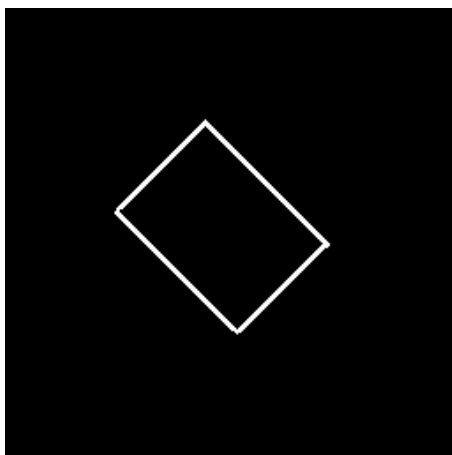
(β') Παραμετρικός χώρος



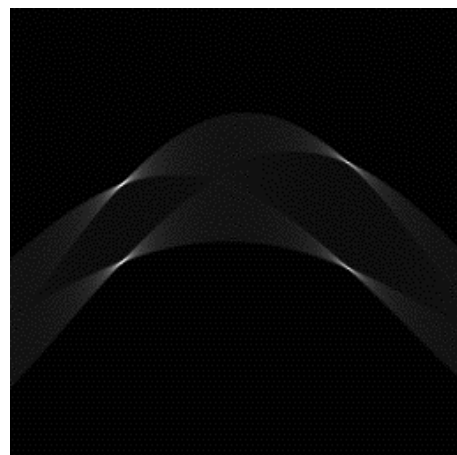
(γ') Τρίγωνο



(δ') Παραμετρικός χώρος



(ε') Ορθογώνιο παραλληλόγραμμο



(στ') Παραμετρικός χώρος

Σχήμα 2.28: Απλές εφαρμογές Hough(συνέχεια).

## Κεφάλαιο 3

# Στοιχεία Ρομποτικής

Στο παρόν κεφάλαιο αναπτύσσεται το πρόβλημα της ευθείας κινηματικής. Η ευθεία κινηματική είναι το πρόβλημα εύρεσης της τελικής θέσης και προσανατολισμού του ακροδέκτη του ρομπότ δεδομένης της γεωμετρίας του<sup>1</sup>. Η αντίστροφη κινηματική αν και συμπληρωματικό πρόβλημα δεν αναπτύσσεται καθώς δεν έχει άμεση σχέση με το αναπτυσσόμενο μετρητικό σύστημα.

### 3.1 Τύποι Αρθρώσεων

Η γεωμετρία ενός ρομπότ καθορίζεται από τους συνδέσμους του και τις αρθρώσεις του. Τα μήκη των συνδέσμων για δεδομένο ρομπότ παραμένουν σταθερά και έτσι η αλλαγή της γεωμετρίας του ρομπότ μπορεί να γίνει μέσω των αρθρώσεων. Υπάρχουν έξι τύποι αρθρώσεων, αυτοί είναι:

1. Περιστροφική
2. Πρισματική
3. Κυλινδρική
4. Επίπεδη
5. Κοχλιωτή
6. Σφαιρική

Οι τέσσερις τελευταίοι τύποι έχουν πάνω από έναν βαθμό ελευθερίας και σπάνια συναντώνται. Ωστόσο όταν εμπλέκονται σε μηχανισμούς μοντελοποιούνται με διαδοχικές αρθρώσεις ενός βαθμού ελευθερίας χωρίς απώλεια γενικότητας. Για τις περιστροφικές αρθρώσεις, η παράμετρος που τις περιγράφει είναι η γωνία στροφής τους ενώ για τις πρισματικές είναι μετατόπιση τους.

### 3.2 Ομογενείς μετασχηματισμοί

Στο κεφάλαιο της όρασης έγινε αναφορά στα μητρώα μεταφοράς και περιστροφής. Υπενθυμίζεται εδώ πως το μητρώο μεταφοράς είναι το διάνυσμα που συνδέει

<sup>1</sup> Δηλαδή τις γωνίες των αρθρώσεων του

το αρχικό με το τελικό σημείο της μεταφοράς. Ενώ τα μητρώα περιστροφής δίνονται από τις σχέσεις 2.21-2.23. Τα μητρώα αυτά μπορούν να εφαρμοσθούν και σε συστήματα συντεταγμένων και η ιδιότητα αυτή χρησιμοποιείται στην μελέτη της κινηματικής των ρομπότ.

Ωστόσο είναι πιο εύκολο να περιγράφεται η περιστροφή και η μεταφορά ενός συστήματος συντεταγμένων με έναν μόνο μετασχηματισμό. Ο μετασχηματισμός αυτός λέγεται ομογενής.

Για την καλύτερη περιγραφή του ομογενή μετασχηματισμού θεωρείται ένα γνωστό διάνυσμα το οποίο αναπαριστάται ως προς ένα πλαίσιο  $B$  και αναζητείται η περιγραφή του σε ένα πλαίσιο  $A$ . Το πλαίσιο  $B$  έχει υποστεί μια περιστροφή και μια μεταφορά ως προς το  $A$ . Το διάνυσμα που εκφράζει την μεταφορά είναι το  ${}^A P_o$ . Εδώ ο αριστερός επάνω δείκτης συμβολίζει το σύστημα συντεταγμένων ως προς το οποίο μετράται το διάνυσμα. Η λύση του προβλήματος δίνεται από την σχέση:

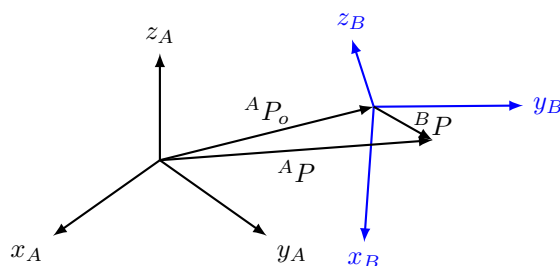
$${}^A P = {}^A_B R {}^B P + {}^A P_o \quad (3.1)$$

η οποία πρώτα εκφράζει το διάνυσμα από το πλαίσιο  $B$  στο  $A$  και στην συνέχεια με διανυσματική πρόσθεση βρίσκει το διάνυσμα  ${}^A P$ . Ο αρχικός μετασχηματισμός είναι απαραίτητος διότι δεν ορίζονται πράξεις μεταξύ διανυσμάτων τα οποία εκφράζονται σε διαφορετικά συστήματα συντεταγμένων. Η σχέση 3.1 εκφράζει τον γενικό μετασχηματισμό μεταξύ δύο πλαισίων. Ωστόσο για λόγους ευκολίας η σχέση 3.1 γράφεται στην μορφή:

$${}^A P = {}^A_B T {}^B P \quad (3.2)$$

όπου ο τελεστής  ${}^A_B T$  μετασχηματίζει ή καλύτερα αναπαριστά το πλαίσιο  $B$  ως προς το πλαίσιο  $A$  και είναι ένας πίνακας  $4 \times 4$ . Λόγω των διαστάσεων του τελεστή προστίθεται ένα ακόμη στοιχείο στα διανύσματα  ${}^A P$  και  ${}^B P$  και η σχέση 3.2 γράφεται στην μορφή:

$$\begin{bmatrix} {}^A P \\ 1 \end{bmatrix} = \begin{bmatrix} {}^A_B R & {}^A P_o \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} {}^B P \\ 1 \end{bmatrix} \quad (3.3)$$



Σχήμα 3.1: Γενική περίπτωση απεικόνισης πλαισίων που έχουν υποστεί περιστροφή και μεταφορά.

Οι τελεστές των ομογενών μετασχηματισμών μπορούν να περιγράψουν μόνο μεταφορά όταν είναι στην μορφή:

$$T = \begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

όπου  $x, y, z$  οι συντεταγμένες του διανύσματος μεταφοράς. Ενώ μόνο για περιστροφή ο τελεστής δίνεται από την σχέση:

$$T = \begin{bmatrix} R_i & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.5)$$

Όπου  $R_i$  το αντίστοιχο μητρώο περιστροφής για τον άξονα γύρω από τον οποίο γίνεται η περιστροφή.

### 3.2.1 Συμβάσεις στροφών

#### Στροφή ως προς σταθερούς άξονες

Έστω δυο πλαίσια  $A, B$  τα οποία ταυτίζονται. Εάν το πλαίσιο  $B$  στραφεί κατά γωνία  $\gamma$  γύρω από τον άξονα  $x_A$ , γωνία  $\beta$  γύρω από τον άξονα  $y_A$  και γωνία  $\alpha$  γύρω από τον άξονα  $z_A$  τότε ο μετασχηματισμός του πλαισίου δίνεται από το μητρώο:

$${}^A R_{xyz}(\alpha, \beta, \gamma) = R_z(\alpha) R_y(\beta) R_x(\gamma) \quad (3.6)$$

σημειώνεται εδώ ότι η αλληλουχία των στροφών παίζει ρόλο. Και ότι το παραπάνω μητρώο είναι σωστό μόνο για στροφές που γίνονται με την παρακάτω σειρά  $x \rightarrow y \rightarrow z$ .

#### Γωνίες Euler $z - y - x$

Έστω δυο πλαίσια  $A, B$  τα οποία ταυτίζονται. Εάν το πλαίσιο  $B$  στραφεί κατά γωνία  $\alpha$  γύρω από τον άξονα  $z_B$ , γωνία  $\beta$  γύρω από τον άξονα  $y_B$  και γωνία  $\gamma$  γύρω από τον άξονα  $x_B$  τότε ο μετασχηματισμός του πλαισίου δίνεται από το μητρώο:

$${}^A R(\alpha, \beta, \gamma) = {}^A R_{B'} R_{B''} R_B = R_z(\alpha) R_y(\beta) R_x(\gamma) \quad (3.7)$$

Οι σχέσεις 3.6 και 3.7 δίνουν το ίδιο αποτέλεσμα υποδηλώνοντας πως τρεις περιστροφές ως προς σταθερούς άξονες παράγουν ίδιο αποτέλεσμα με τρεις περιστροφές ως προς άξονες κινούμενου πλαισίου.

Συνολικά υπάρχουν 24 περιπτώσεις διαφορετικής αλληλουχίας στροφών. 12 για στροφές γύρω από σταθερούς άξονες και 12 για γωνίες Euler. Κάθε περίπτωση στροφής γωνιών Euler είναι ισοδύναμη με μια περίπτωση στροφής με γωνίες γύρω από απόλυτο πλαίσιο.

## 3.3 Παράμετροι Denavit–Hartenberg

Για την αντιμετώπιση της γεωμετρικής πολυπλοκότητας, προσαρτώνται πλαίσια σε διάφορα τμήματα του μηχανισμού και περιγράφεται η μεταξύ τους σχέση. Έτσι πρώτα αριθμούνται τα μέλη του βραχίονα. Στην ακίνητη βάση του αποδίδεται ο αριθμός 0, στο πρώτο κινούμενο μέλος ο αριθμός 1 κ.ο.κ μέχρι το τέλος του βραχίονα όπου αποδίδεται ο αριθμός  $n$ . Για την πλήρη τοποθέτηση του ακροδέκτη του ρομπότ χρειάζονται τουλάχιστον έξι αρθρώσεις<sup>2</sup>.

<sup>2</sup>Αφού η πλήρης περιγραφή της θέσης και προσανατολισμού χρειάζεται τρεις παραμέτρους για την θέση και τρεις για τον προσανατολισμό

Τα πλαίσια τοποθετούνται έτσι ώστε ο άξονας  $Z$  να βρίσκεται πάνω στον άξονα  $i$  του μελετούμενου μέρους. Εδώ ως άξονας ορίζεται η ευθεία γύρω από την οποία στρέφεται το μέλος  $i$  ως προς το  $i - 1$ . Έτσι με τον τρόπο αυτό κάθε μέλος μπορεί να περιγραφεί από τις δύο παραμέτρους που καθορίζουν την σχετική θέση μεταξύ δύο αξόνων.

Για κάθε ζεύγος η απόσταση των αξόνων είναι το ευθύγραμμο τμήμα το οποίο είναι κάθετο και στους δύο άξονες, συμβολίζεται με  $a_i$  και αποτελεί την πρώτη παράμετρο. Έτσι η απόσταση  $a_{i-1}$  είναι η απόσταση μεταξύ των αξόνων  $i - 1$  και  $i$ .

Η δεύτερη παράμετρος ονομάζεται στρέψη, συμβολίζεται με  $\alpha_i$  και είναι η γωνία που σχηματίζεται μεταξύ των αξόνων εάν τους προβάλλουμε σε ένα επίπεδο κάθετο στην κοινή τους κάθετη. Η γωνία αυτή μετράται δεξιόστροφα σε σχέση με την  $a_i$ <sup>3</sup>. Σε περίπτωση που οι άξονες τέμνονται, η γωνία μετράται πάνω στο επίπεδο που τους περιέχει και η φορά μπορεί να ορισθεί αυθαίρετα.

### 3.3.1 Ενδιάμεσα και ακραία μέλη αλυσίδας

Η σύνδεση μελών έχει ως αποτέλεσμα να υπάρχει μια κοινή άρθρωση και άρα ένας άξονας. Ωστόσο η σύνδεση αυτή εισάγει και μια μετατόπιση πάνω σε αυτόν τον κοινό άξονα η οποία ονομάζεται μετάθεση μέλους και συμβολίζεται με  $d_i$  όπου  $i$  ο άξονας πάνω στον οποίο ανήκει.

Επιπρόσθετα τα μέλη αυτά μπορούν να έχουν σχετική περιστροφή η οποία ονομάζεται γωνία άρθρωσης και συμβολίζεται με  $\theta_i$ . Στο σημείο αυτό σημειώνεται πως η  $d_i$  είναι μεταβλητή όταν η άρθρωση είναι πρισματική ενώ αντίστοιχα η  $\theta_i$  μεταβάλλεται όταν η άρθρωση είναι περιστροφική.

Τέλος για τα ακραία μέλη της αλυσίδας ισχύει πως  $a_0 = a_n = 0$  και  $\alpha_0 = \alpha_n = 0$  ενώ στις περιπτώσεις όπου η άρθρωση 1 είναι περιστροφική η μηδενική θέση για την γωνία  $\theta_1$  επιλέγεται αυθαίρετα και  $d_1 = 0$ . Ενώ όταν η άρθρωση είναι πρισματική ισχύει  $\theta_1 = 0$  και αυθαίρετη επιλογή της  $d_1$ . Στο σχήμα 3.2 φαίνονται οι παράμετροι D-H για και η τοποθέτηση τους για έναν ρομποτικό βραχίονα.

### 3.3.2 Μετασχηματισμός Μέλους

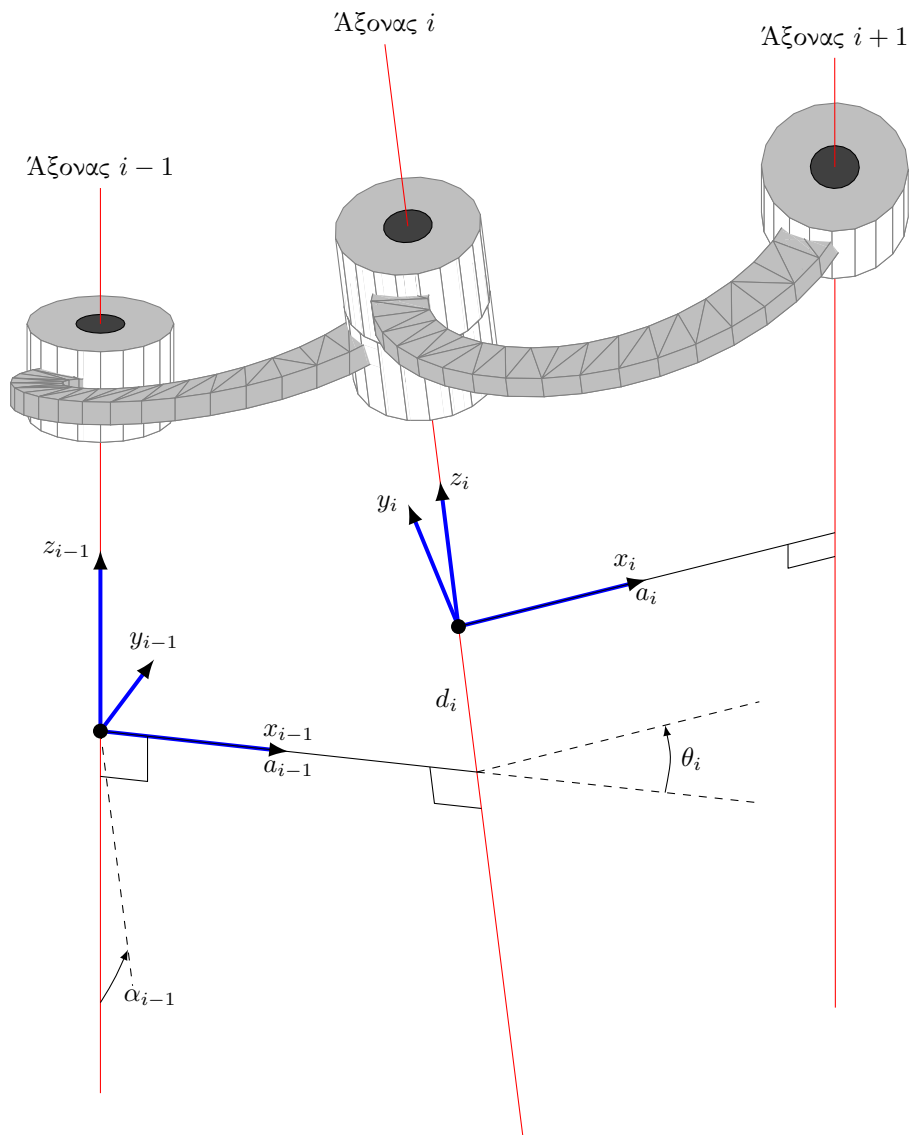
Ο μετασχηματισμός μέλους είναι αυτός που περιγράφει το πλαίσιο  $i$  ως προς το πλαίσιο  $i - 1$ . Για δεδομένο βραχίονα όλες οι παράμετροι πέρα από την μεταβλητή της άρθρωσης είναι καθορισμένες από την μηχανολογική σχεδίαση του βραχίονα. Η εύρεση του μετασχηματισμού δίνεται χρησιμοποιώντας πολλά διαδοχικά πλαίσια που συνδέονται μεταξύ τους με ομογενείς μετασχηματισμούς.

Το αρχικό πλαίσιο  $i - 1$  περιστρέφεται κατά  $\alpha_{i-1}$  στην συνέχεια μετατοπίζεται κατά  $a_{i-i}$  περιστρέφεται κατά  $\theta_i$  και μετατοπίζεται κατά  $d_i$ . Το πλαίσιο που παράγεται είναι το πλαίσιο  $i$ . Με τον ίδιο τρόπο προκύπτουν και οι μετασχηματισμοί των άλλων μελών του ρομπότ. Γενικά όμως είναι:

$${}^i{}_{i-1}T = Rot_z(\alpha_{i-1}) Trans(a_{i-i}) Rot_z(\theta_i) Trans(d_i) \quad (3.8)$$

<sup>3</sup> Δηλαδή με διεύθυνση από τον  $i - 1$  προς τον  $i$





Σχήμα 3.2: Παράσταση παραμέτρων Denavit-Hartenberg

## Κεφάλαιο 4

# Διατύπωση και επίλυση του προβλήματος

Μέχρι στιγμής έχει γίνει μια βασική εισαγωγή στην θεωρία της τεχνητής όρασης αλλά και στην ρομποτική. Στο παρόν κεφάλαιο αναπτύσσεται το πρόβλημα το οποίο πρέπει να επιλυθεί, οι διάφοροι τρόποι επίλυσης του και πως αυτοί δοκιμάστηκαν. Τέλος παρουσιάζονται και ερμηνεύονται τα αποτελέσματα που προκύπτουν και γίνεται και μια ανάλυση σφάλματος για την μέθοδο επίλυσης.

### 4.1 Εξέταση του προβλήματος

Το πρόβλημα που ορίζεται προς επίλυση είναι η εύρεση των θέσεων των αρθρώσεων του ρομπότ με χρήση μεθόδων τεχνητής όρασης. Τεχνικά το πρόβλημα μπορεί να χωριστεί σε μικρότερα υποπροβλήματα όπως φαίνεται στο επόμενο σχήμα.



Σχήμα 4.1: Χωρισμός του προβλήματος σε μικρότερα υποπροβλήματα

Όλα αυτά τα υποπροβλήματα δεν έχουν αυστηρά μοναδική λύση. Για παράδειγμα θα μπορούσαν να τοποθετηθούν πολλές κάμερες υπό διάφορες γωνίες ώστε να υπάρχει πλήρης κάλυψη του χώρου εργασίας του ρομπότ. Ωστόσο αυτό θα ήταν οικονομικά και υπολογιστικά ασύμφορο και λειτουργικά μη πρακτικό. Επιπρόσθετα περιορισμοί στον χώρο εργασίας του ρομπότ μπορεί να μην επιτρέπουν

την τοποθέτηση καμερών σε όλα τα σημεία και έτσι να επιβάλλουν επιπρόσθετους περιορισμούς.

Τέλος η επιλογή των χαρακτηριστικών τα οποία ανιχνεύονται παίζει σημαντικό ρόλο στην ακρίβεια αλλά και στην υπολογιστική αποτελεσματικότητα του συστήματος. Είναι γενικά αναγκαίο να ανιχνεύονται χαρακτηριστικά της εικόνας τα οποία δεν υπόκεινται σε σημαντικές μεταβολές θορύβου, αλλά είναι και υπολογιστικά εφικτό να ανιχνεύονται και να παρακολουθούνται σε πραγματικό χρόνο.

## 4.2 Προδιαγραφές

Ένα βασικό κομμάτι το οποίο οριοθετεί κατά κάποιον τρόπο τις μεθόδους αλλά και τα εργαλεία που μπορούν να χρησιμοποιηθούν είναι οι προδιαγραφές. Οι βασικές προδιαγραφές που τέθηκαν αρχικά ήταν η δυνατότητα του συστήματος να παρέχει δεδομένα σε πραγματικό χρόνο, να είναι χαμηλού κόστους αλλά και μια αυθαίρετη ακρίβεια της τάξεως των  $\pm 5^\circ$  η οποία θα μπορούσε να βελτιωθεί στην συνέχεια.

## 4.3 Γενικός τρόπος επίλυσης

Η επίλυση του προβλήματος έγινε σε βήματα. Αυτά ήταν τα παρακάτω. Αρχικά τοποθετούνται στόχοι και στην συνέχεια γίνεται λήψη βίντεο από τις κάμερες. Από την κάθε εικόνα που λαμβάνεται<sup>1</sup> ανιχνεύονται οι στόχοι και μετράται η παραμόρφωση τους. Τέλος με βάση την παραμόρφωση των στόχων επιλύεται αντίστροφα ένας χάρτης που δείχνει το εμβαδόν το οποίο είναι ορατό στην κάμερα σε σχέση με την γωνία της άρθρωσης και έτσι βρίσκεται η γωνία.

Η μέτρηση της παραμόρφωσης μετράται με βάση τα ορατά pixels. Αρχικά ο στόχος τοποθετείται στην άρθρωση έτσι ώστε να είναι πλήρως ορατός. Στην συνέχεια ανιχνεύεται και μετρώνται τα pixel του. Όταν ο στόχος παραμορφώνεται τα ορατά pixels μειώνονται σε ένα ποσοστό του αρχικού αριθμού. Κατά το ίδιο ποσοστό μειώνεται και το ορατό εμβαδόν.

## 4.4 Πρώτο μοντέλο

Έστω ότι υπάρχουν κυκλικοί στόχοι πάνω στην περιστρεφόμενη άρθρωση του ρομπότ. Καθώς η άρθρωση περιστρέφεται οι κύκλοι των στόχων μετασχηματίζονται σε ελλείψεις. Εάν οι ελλείψεις αυτές προβάλλονται σε ένα επίπεδο όπως φαίνεται στο σχήμα 4.2 και μετρηθεί το εμβαδόν τους τότε η μεταβολή του συνολικού εμβαδού σε σχέση με την γωνία περιστροφής της άρθρωσης αποτελεί τον χάρτη επίλυσης για την αντίστροφη διαδικασία.

### 4.4.1 Εύρεση χάρτη

Το συνολικό εμβαδόν που είναι ορατό στην κάμερα αποτελείται από δύο ξεχωριστά εμβαδά, ένα για κάθε κύκλο. Για γωνία στροφής  $\varphi$  ισχύει για τις ακτίνες των δύο κύκλων:

$$R_{1p} = R_c \cos(\varphi) \quad R_{2p} = R_c \cos\left(\frac{\pi}{2} - \varphi\right)$$

<sup>1</sup>Υπενθυμίζεται πως ένα βίντεο είναι στην ουσία μια αλληλουχία από εικόνες

Όμως εφόσον δεν υπάρχει περιστροφή γύρω από τους δυο οριζόντιους άξονες η προβολή της κατακόρυφης ακτίνας δεν μεταβάλλεται και είναι ίση με  $R_c$ . Έτσι το εμβαδόν κάθε ορατής έλλειψης είναι:

$$E_1 = |\pi R_c R_{1p}| \quad E_2 = |\pi R_c R_{2p}|$$

έτσι όλο το εμβαδόν που είναι ορατό στην κάμερα είναι:

$$E_{sum} = \pi R_c \left( |\cos(\varphi)| + \left| \cos\left(\frac{\pi}{2} - \varphi\right) \right| \right) \quad (4.1)$$

Ο χάρτης φαίνεται στο σχήμα 4.3 και υποδεικνύει πως για μια τιμή εμβαδού δεν υπάρχει μοναδική λύση. Υπάρχουν δυο στρατηγικές για την αντιμετώπιση αυτού του προβλήματος. Η μία είναι η εύρεση όλων των λύσεων και επιλογή της μίας και η δεύτερη είναι η λύση μόνο σε μια περιοχή της καμπύλης. Ωστόσο και τις δύο φορές πρέπει να υπάρχει ένα κριτήριο που είτε να υποδεικνύει την περιοχή της λύσης είτε να υποδεικνύει ποια λύση να κρατηθεί. Όμως στην πράξη ένα τέτοιο κριτήριο δεν είναι εύκολο να βρεθεί.

Η στρατηγική που επιλέχθηκε για λόγους υπολογιστικής οικονομίας ήταν η επίλυση μόνο σε μια περιοχή της καμπύλης. Αυτό θα επιτυγχάνεται με προϋπόθεση να είναι γνωστή η γωνία εκκίνησης και η φορά περιστροφής. Η παραδοχή αυτή όμως, επιβάλλει έναν πρόσθετο περιορισμό και αυτός είναι η μη αλλαγή της φοράς περιστροφής του βραχίονα στις κορυφές της καμπύλης.

#### 4.4.2 Αποτυχία μοντέλου

Όταν το μοντέλο υλοποιήθηκε, τα αποτελέσματα που προέκυψαν δεν ήταν τα επιθυμητά. Τα προβλήματα που προέκυψαν ήταν ότι η γωνία που επιλύθηκε δεν έφτασε ποτέ τις 45 μοίρες αλλά και δεν έπεσε ποτέ κάτω από το όριο των 10 μοιρών. Τα λάθη αυτά αποδόθηκαν στην μη χρήση του μοντέλου pinhole αλλά και στην μεταβολή του χρώματος του στόχου σε μεγάλες γωνίες στροφής σε σχέση με την κάμερα που οφείλονταν στην αντανάκλαση περισσότερου φωτός.

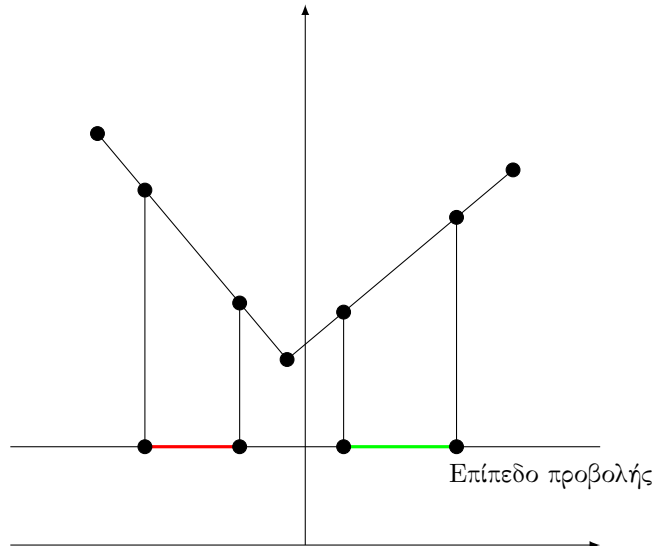
### 4.5 Δεύτερο Μοντέλο

Το δεύτερο θεωρητικό μοντέλο βασίζεται στο μοντέλο κάμερας pinhole. Οι στόχοι είναι πάλι μαύροι κύκλοι οι οποίοι τοποθετούνται στις πλευρές της άρθρωσης. Αλλά τώρα προβάλλονται όχι σε ένα επίπεδο παράλληλο στην κάμερα, αλλά στο επίπεδο αποτύπωσης της κάμερας, διαμέσου της οπής. Η παρακολούθηση γίνεται πάλι με χρήση μοναδικής κάμερας.

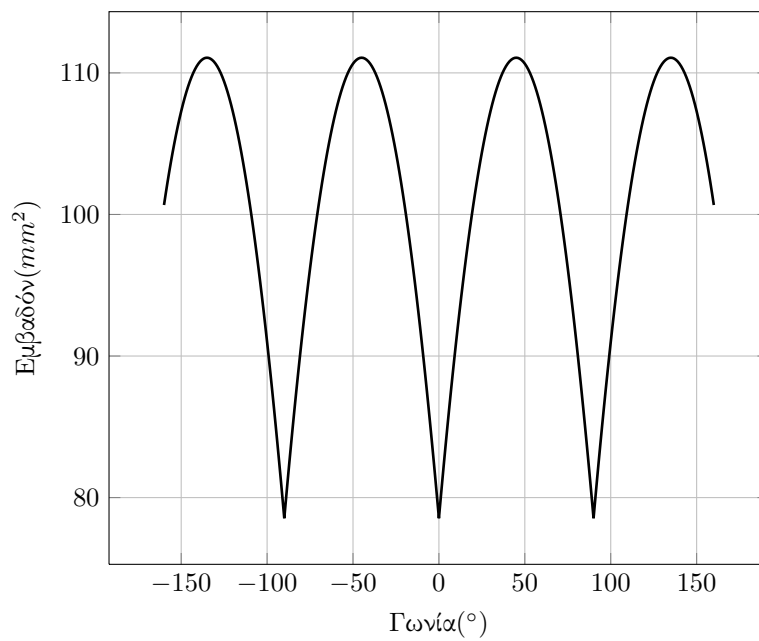
#### 4.5.1 Εύρεση χάρτη

Έστω δυο διανύσματα  $u$  και  $v$  όπως φαίνεται στο σχήμα 4.4 τα οποία ξεκινάνε από το κέντρο περιστροφής της άρθρωσης και καταλήγουν στα αντιδιαμετρικά σημεία του παρακολουθούμενου κύκλου και θεωρείται πως βρίσκονται στο επίπεδο  $z = 0$ . Καθώς ο κύκλος περιστρέφεται κατά γωνία  $\theta$  τότε το μητρώο περιστροφής είναι:

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



Σχήμα 4.2: Προβολή των κύκλων ως ελλείψεις σε ένα επίπεδο



Σχήμα 4.3: Γραφική παράσταση του χάρτη του πρώτου μοντέλου

Για την υλοποίηση θεωρείται πως η περιστροφή γίνεται πάνω σε ένα επίπεδο. Συνεπώς το μητρώο μεταπίπτει στην παρακάτω απλοποιημένη μορφή:

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (4.2)$$

Απλοποιώντας την μοντελοποίηση της κάμερας θεωρείται ότι δεν έχουμε σφάλματα φακού και σημαντικά κατασκευαστικά σφάλματα τα οποία να εισάγουν μετατοπίσεις στο εστιακό σημείο και άρα αλλαγές στις εστιακές αποστάσεις. Αυτό ισχύει διότι στην χρησιμοποιούμενη κάμερα δεν χρησιμοποιούνται ευρυγώνιοι φακοί. Ωστόσο σε περίπτωση που κάτι τέτοιο δεν ίσχυε θα έπρεπε πρώτα να γίνει αποπαραμόρφωση εικόνας και στην συνέχεια να γίνει η εφαρμογή του μοντέλου του χάρτη.

Για υλοποίηση σε υπολογιστή τα διανύσματα γράφονται σε μητρωική μορφή:

$$u = \begin{bmatrix} x_u \\ y_u \end{bmatrix} \quad v = \begin{bmatrix} x_v \\ y_v \end{bmatrix} \quad (4.3)$$

Τα αντίστοιχα περιστραμμένα διανύσματα δίνονται από τις σχέσεις:

$$u' = \begin{bmatrix} x_u \cos(\theta) - y_u \sin(\theta) \\ x_u \sin(\theta) + y_u \cos(\theta) \end{bmatrix} \quad v' = \begin{bmatrix} x_v \cos(\theta) - y_v \sin(\theta) \\ x_v \sin(\theta) + y_v \cos(\theta) \end{bmatrix} \quad (4.4)$$

Έτσι με γνωστά το κέντρο περιστροφής και τα περιστραμμένα διανύσματα είναι δυνατή η εύρεση των θέσεων των αντιδιαμετρικών σημείων των περιστραμμένων κύκλων, σύμφωνα με τις παρακάτω σχέσεις.

$$P_{ux} = x_u \cos(\theta) - y_u \sin(\theta) + c_x \quad P_{uy} = x_u \sin(\theta) + y_u \cos(\theta) + c_y \quad (4.5)$$

και:

$$P_{vx} = x_v \cos(\theta) - y_v \sin(\theta) + c_x \quad P_{vy} = x_v \sin(\theta) + y_v \cos(\theta) + c_y \quad (4.6)$$

όπου  $c$  το κέντρο περιστροφής. Θεωρώντας το εστιακό σημείο ως το κέντρο του συστήματος συντεταγμένων είναι δυνατή η εύρεση των γωνιών που σχηματίζουν οι ευθείες, που διέρχονται από τα σημεία  $P$  και το εστιακό σημείο, με τον οριζόντιο άξονα. Που είναι:

$$\psi_i = \arctan\left(\frac{P_{iy}}{P_{ix}}\right) \quad (4.7)$$

Οι οριζόντιες αποστάσεις από το εστιακό σημείο, πάνω στο επίπεδο αποτύπωσης είναι:

$$d_i = \tan\left(\frac{\pi}{2} - \psi_i\right) f \quad (4.8)$$

Έτσι για κάθε κύκλο η ακτίνα στο επίπεδο αποτύπωσης είναι:

$$r_c = \frac{\sqrt{(d_1 - d_2)^2}}{2} \quad (4.9)$$

και το εμβαδόν της σχηματιζόμενης έλλειψης είναι:

$$A = \pi r_c r_{const} \quad (4.10)$$

όπου  $r_{const}$  είναι η ακτίνα στον κατακόρυφο άξονα, η οποία για δεδομένη απόσταση του παρατηρούμενου κύκλου από το κέντρο περιστροφής παραμένει σταθερή. Ενώ  $r_c$  είναι η ακτίνα που αλλάζει βάσει της γωνίας περιστροφής  $\theta$ . Επιπλέον ισχύει πως  $r_{const} = r_c(0)$ .

Το συνολικό εμβαδόν που είναι ορατό στην κάμερα είναι το άθροισμα των εμβαδών των ορατών ελλείψεων οι οποίες μπορεί να είναι το πολύ δύο. Ωστόσο αυτό εισάγει το βασικό ερώτημα του πότε είναι ορατή μια έλλειψη.

### Γωνίες μη ορατότητας ελλείψεων

Έστω ένας κύβος ο οποίος παρακολουθείται από κάμερα και έχει αρχική θέση τέτοια ώστε το διάνυσμα της κάμερας να διέρχεται κάθετα στο κέντρο της πλευράς του. Εάν σχεδιαστούν ευθείες οι οποίες διέρχονται από την οπή της κάμερας προς τα "άκρα" των κύκλων έτσι όπως αυτοί είναι ορατοί στο επίπεδο  $xy$ , τότε είναι εμφανές πως δεν αποτυπώνονται όλες οι ελλείψεις από την κάμερα αφού κάποιες από τις ευθείες διέρχονται μέσα από το σώμα του κύβου. Έτσι το πρόβλημα μπορεί να διατυπωθεί ως εξής:

"Για ποια γωνία περιστροφής του τετραγώνου αρχίζει να είναι ορατός ο παρακολουθούμενος κύκλος;"

Το πρόβλημα αυτό λύνεται εύκολα εάν είναι γνωστή η περιγραφή του πολυγώνου. Καθώς είναι δυνατή η εύρεση της τομής της ευθείας με τις πλευρές του πολυγώνου. Αυτό όμως προϋποθέτει την γνώση της περιγραφής του πολυγώνου. Αλλά και πολλές επιλύσεις στις διάφορες πλευρές αυτού. Όμως το πρόβλημα είναι δυνατόν να λυθεί πιο εύκολα, καθώς είναι ισοδύναμο με μια ισότητα ευθειών. Διότι ο κύκλος είναι οριακά ορατός όταν οι ευθείες που διέρχονται από την οπή και τα δύο ακραία σημεία του κύκλου είναι ίσες. Η απόδειξη αυτή επεκτείνεται σε όλα τα ευκλείδεια κλειστά πολύγωνα. Δεδομένου ότι οι ευθείες περνάνε από το κέντρο του συστήματος συντεταγμένων τότε αυτές δίνονται από την σχέση:

$$y = ax$$

έτσι η ισότητα ευθειών συνεπάγεται εξίσωση συντελεστών διεύθυνσης. Η οποία με την σειρά της συνεπάγεται την παρακάτω εξίσωση:

$$\frac{P_{uy}}{P_{ux}} = \frac{P_{vy}}{P_{vx}} \quad (4.11)$$

και με αντικατάσταση των εκφράσεων των  $P$  προκύπτει:

$$\frac{x_u \sin(\theta) + y_u \cos(\theta) + c_y}{x_u \cos(\theta) - y_u \sin(\theta) + c_x} = \frac{x_v \sin(\theta) + y_v \cos(\theta) + c_y}{x_v \cos(\theta) - y_v \sin(\theta) + c_x} \quad (4.12)$$

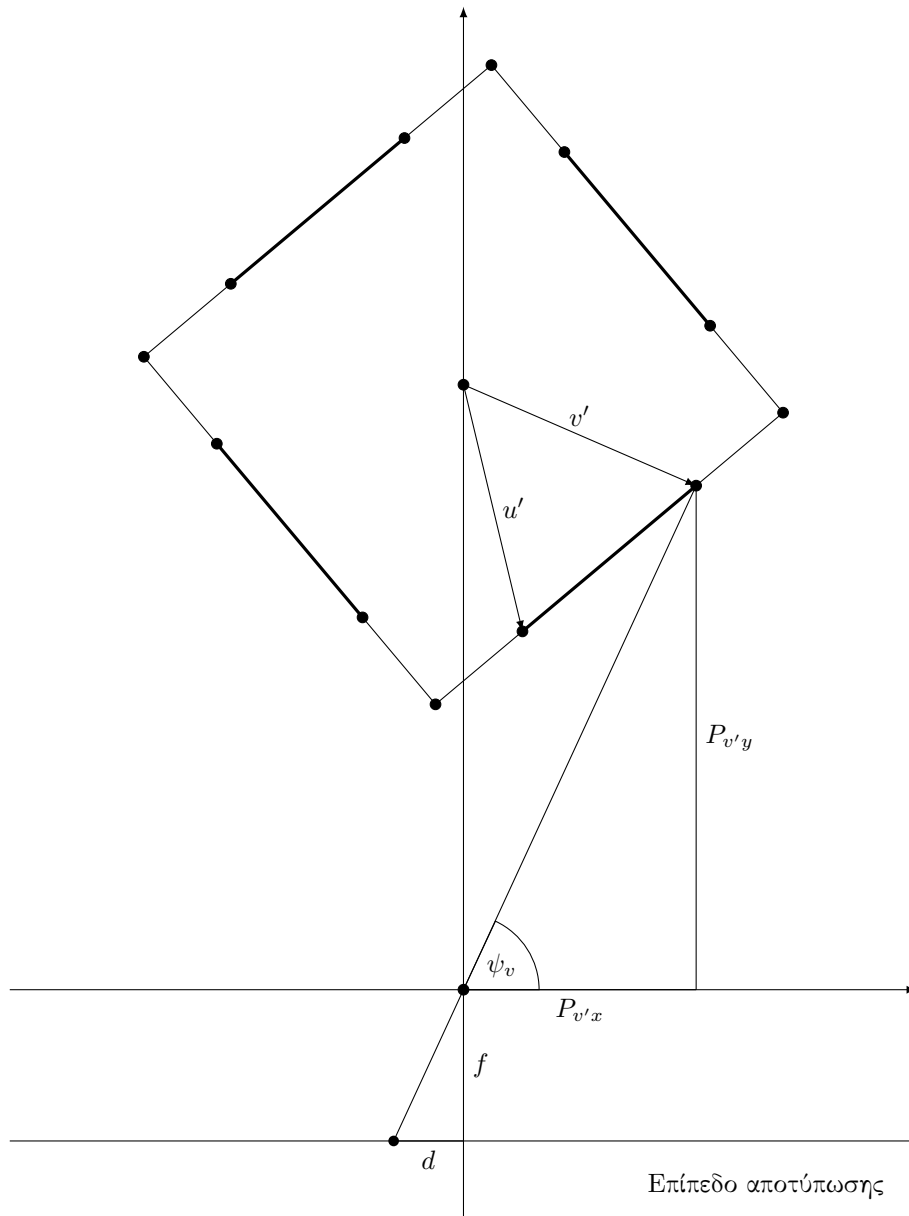
η λύση της παραπάνω σχέσης δίνει τις τιμές της γωνίας για τις οποίες υπάρχει έναρξη και λήξη της ορατότητας του κύκλου.

Συγκεκριμένα για τετράγωνο όμως ισχύει πως:

$$x_u = x_v \quad y_u = -y_v \quad (4.13)$$

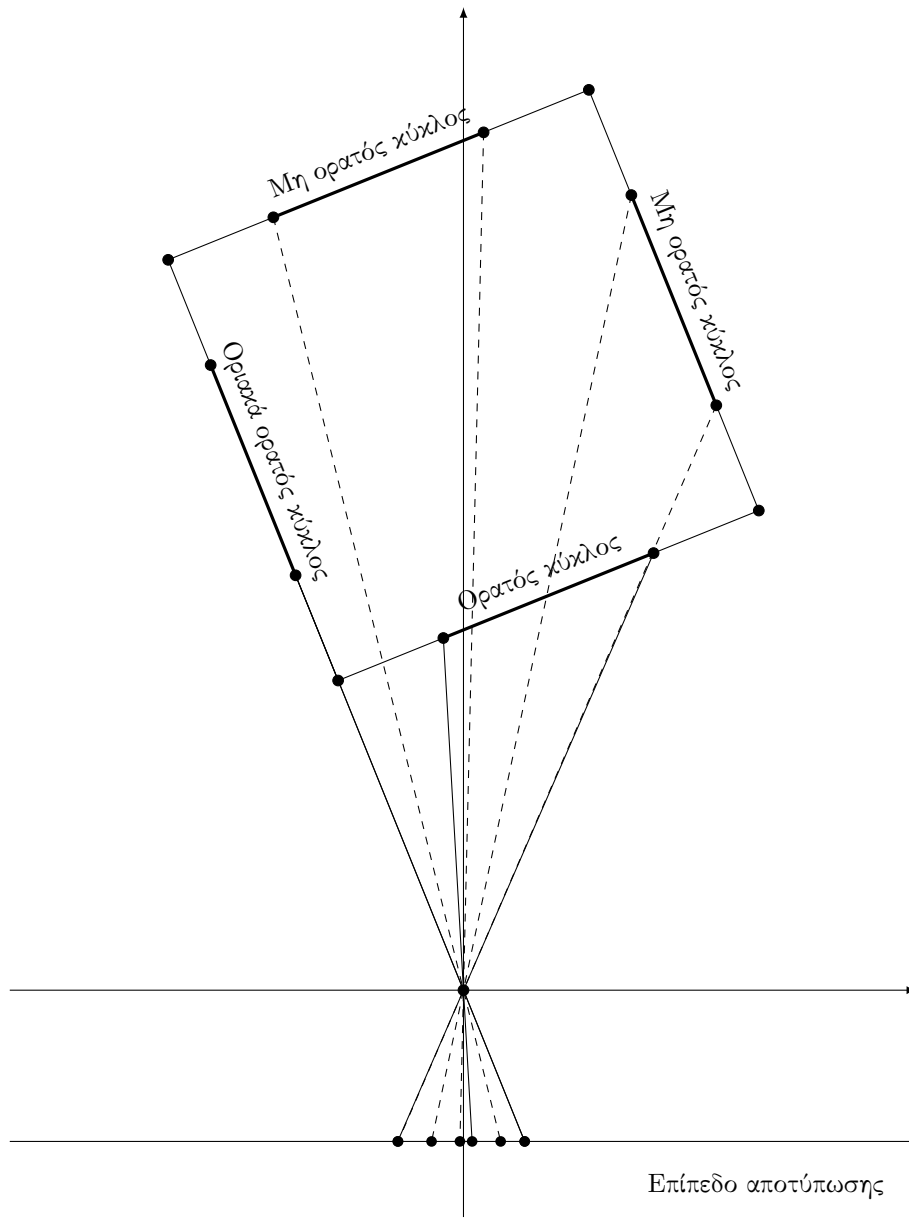
και με αντικατάσταση προκύπτει η σχέση:

$$2x_u y_u \sin^2(\theta) + 2x_u y_u \cos^2(\theta) + 2y_u \sin(\theta)c_y + 2y_u \cos(\theta)c_x = 0 \quad (4.14)$$



Σχήμα 4.4: Γεωμετρικά στοιχεία εύρεσης χάρτη.





Σχήμα 4.5: Γωνίες ορατότητας στόχων.

Είναι προφανές ότι οι τιμές της τυφλής γωνίας εξαρτώνται από την απόσταση της κάμερας από το κέντρο περιστροφής. Όσο πιο κοντά βρίσκεται η κάμερα τόσο μεγαλύτερη είναι η τυφλή γωνία. Στο σχήμα 4.6(α') φαίνεται η λύση της σχέσης 4.14 για χρήση των δυο διανυσμάτων ενός κύκλου. Είναι προφανές ότι αυτά δίνουν την ίδια λύση. Στο σχήμα 4.6(β') φαίνεται ο υπολογισμός των τυφλών σημείων όλων των κύκλων. Για τον υπολογισμό τους γίνεται μια πρόσθετη διαφορά φάσης ανάλογα με την πλευρά που ανήκουν, με θετική φορά την αντισωρολογιακή.

Πλέον με γνωστές τις τυφλές γωνίες των κύκλων είναι δυνατός ο υπολογισμός του χάρτη εμβαδού ο οποίος φαίνεται στο σχήμα 4.7. Ο χάρτης που προκύπτει είναι αρκετά πιο πολύπλοκος από τον χάρτη του πρώτου μοντέλου. Επιπρόσθετα ο αριθμός των ακμών είναι ίσος με τον αριθμό των τυφλών γωνιών. Αυτό δίνει και μια καλή ερμηνεία του χάρτη.

Η αρχική θέση του κύβου είναι τέτοια ώστε η κάμερα να βλέπει τέλειο κύκλο. Όταν ο κύβος αρχίζει και περιστρέφεται τότε η κάμερα συνεχίζει να βλέπει μόνο έναν κύκλο και το συνολικό μετρούμενο εμβαδόν μειώνεται. Όταν η περιστροφή ξεπεράσει την τυφλή γωνία του επόμενου κύκλου τότε το εμβαδόν αρχίζει να αυξάνεται μέχρι την μέγιστη τιμή του στις  $45^\circ$  και η διαδικασία επαναλαμβάνεται. Τοπικά μέγιστα είναι τα σημεία που η κάμερα βλέπει τέλειους κύκλους, δηλαδή οι γωνίες  $0^\circ, 90^\circ$  κ.ο.κ και ολικά μέγιστα οι γωνίες  $45^\circ, 135^\circ$  κ.ο.κ.

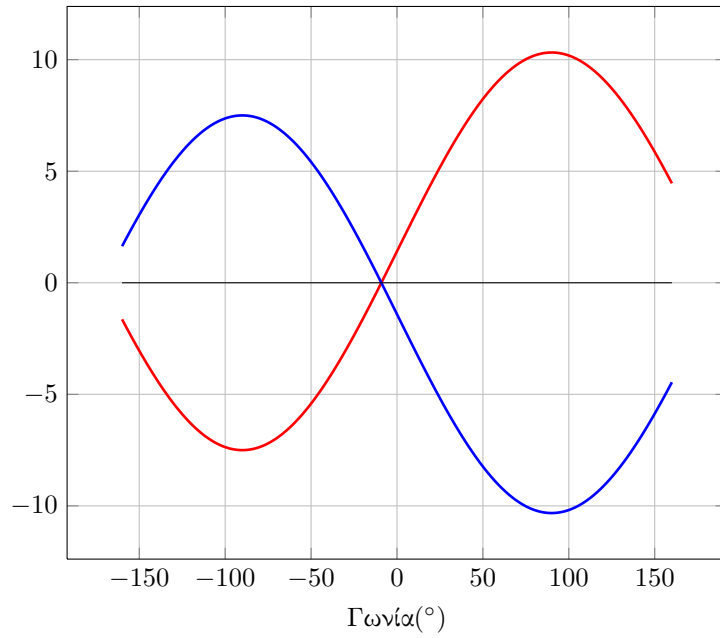
#### 4.5.2 Αποτυχία δεύτερου μοντέλου

Η αποτυχία του δεύτερου μοντέλου έγκειται στο γεγονός ότι πλέον τα κομμάτια που πρέπει να κοπεί η καμπύλη έτσι ώστε να αυτή να είναι 1-1 είναι πλέον αρκετά, καθιστώντας ασύμφορο τον υπολογισμό και τη διαχείριση των αποτελεσμάτων. Επιπρόσθετα υπάρχει ο περιορισμός της μη αλλαγής φοράς κίνησης κοντά στα σημεία αλλαγής κομματιού καμπύλης. Αυτό σε συνδυασμό πλέον με τα πολλά κομμάτια του χάρτη εισάγει έναν τεράστιο προγραμματιστικό περιορισμό στο ρομπότ. Τέλος όταν υλοποιήθηκε το μοντέλο η μετάβαση μεταξύ των κομματιών του χάρτη δεν ήταν αξιόπιστη. Αυτό ήταν απόρροια του θορύβου καθώς μικρές διακυμάνσεις στον αριθμό των ορατών pixel οδηγούσαν σε διακυμάνσεις του μετρούμενου εμβαδού και συνεπώς διακυμάνσεις και στην ταχύτητα η οποία προέκυπτε από παραγωγή της μετρούμενης γωνίας. Ωστόσο η διακύμανση στην ταχύτητα παραβίαζε την παραδοχή της σταθερής φοράς και κατά συνέπεια την αξιοπιστία του μοντέλου.

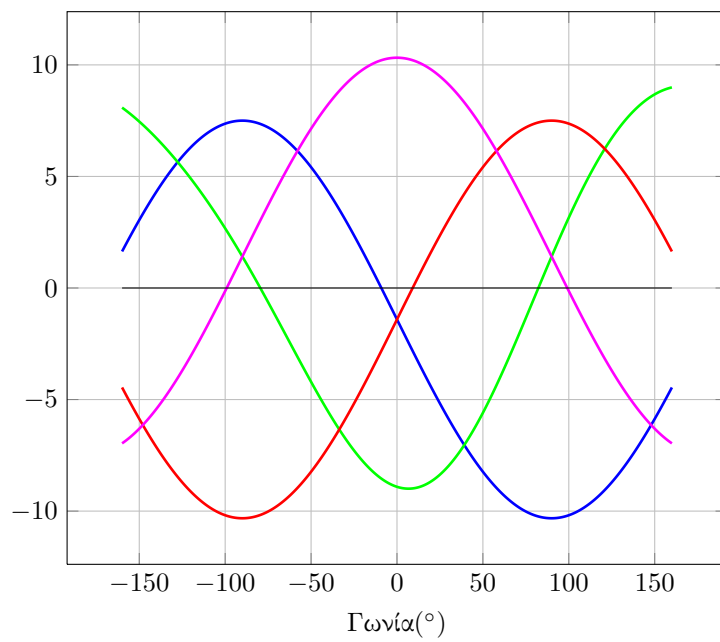
### 4.6 Τρίτο Μοντέλο

Μετά το δεύτερο μοντέλο είναι πλέον προφανές πως η παρακολούθηση του ρομπότ χρησιμοποιώντας μόνο μαύρους στόχους είναι μη πρακτική καθώς δεν είναι εύκολη η εύρεση της περιοχής της καμπύλης γωνίας-εμβαδού η οποία πρέπει να επιλυθεί.

Για την πιο αξιόπιστη εύρεση της περιοχής της καμπύλης προς επίλυση γίνεται εισαγωγή χρωμάτων σε κάθε παρακολουθούμενο στόχο. Έτσι ανάλογα με τα χρώματα που είναι ορατά κάθε φορά αυτόματα είναι γνωστή και η περιοχή της καμπύλης που πρέπει να επιλυθεί. Η διαφορά πλέον βρίσκεται ότι η λύση της καμπύλης δεν πρέπει να γίνεται στο συνολικό εμβαδόν, αλλά στις συνιστώσες του συνολικού εμβαδού.

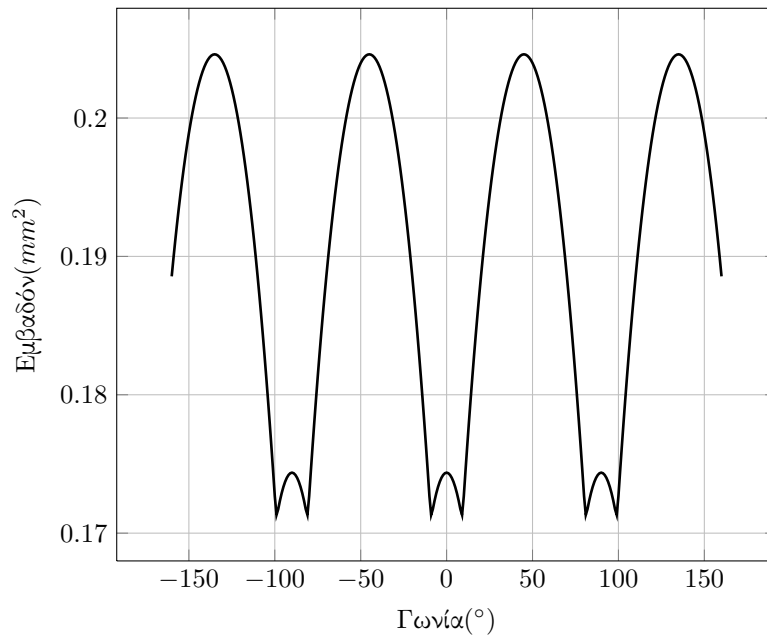


(α') Τυφλές γωνίες για έναν κύκλο.

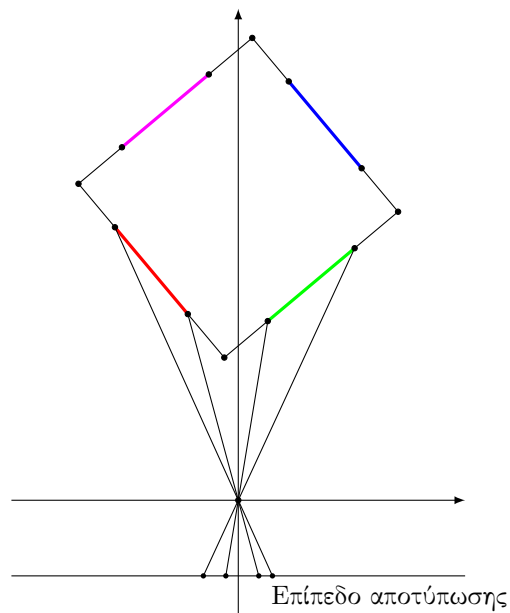


(β') Τυφλές γωνίες για όλους του κύκλους.

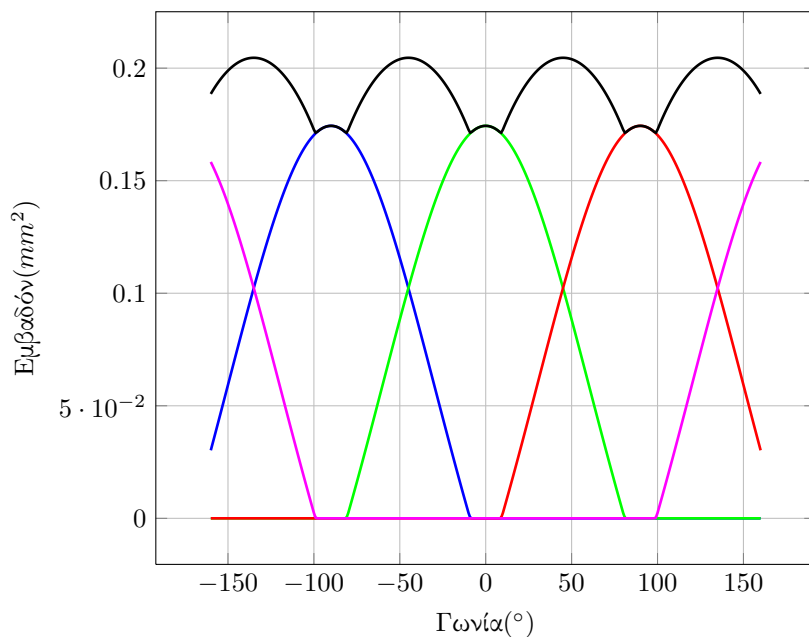
Σχήμα 4.6: Τιμές του αριστερού μέλους της σχέσης 4.14. Οι τιμές στον άξονα  $y$  είναι λογιστικό μέγεθος το οποίο δεν έχει φυσική σημασία και άρα μονάδες.



Σχήμα 4.7: Γραφική παράσταση του χάρτη του δεύτερου μοντέλου.



Σχήμα 4.8: Γεωμετρική αναπαράσταση του τρίτου μοντέλου.



Σχήμα 4.9: Γραφική παράσταση του χάρτη του τρίτου μοντέλου.

Ωστόσο το πρόβλημα που προκύπτει εδώ είναι η ύπαρξη σημείων όπου υπάρχει ορατότητα μόνο ενός χρώματος και αυτό διότι ο πλήρης ορισμός της περιοχής επίλυσης γίνεται μόνο όταν είναι ορατά δύο χρώματα. Η μόνιμη όραση χρωμάτων δεν μπορεί να επιτευχθεί όμως με μία μόνο κάμερα και για τον λόγο αυτό εισάγεται μια δεύτερη κάμερα σε γωνία μικρότερη των  $90^\circ$  από την αρχική κάμερα. Η συνθήκη της γωνίας να είναι μικρότερη από  $90^\circ$  εξασφαλίζει την ορατότητα πάντα δύο χρωμάτων. Φυσικά η τεχνική αυτή μπορεί να επεκταθεί και σε περισσότερες από μία κάμερες. Η γεωμετρική αναπαράσταση του τρίτου μοντέλου φαίνεται στο σχήμα 4.8.

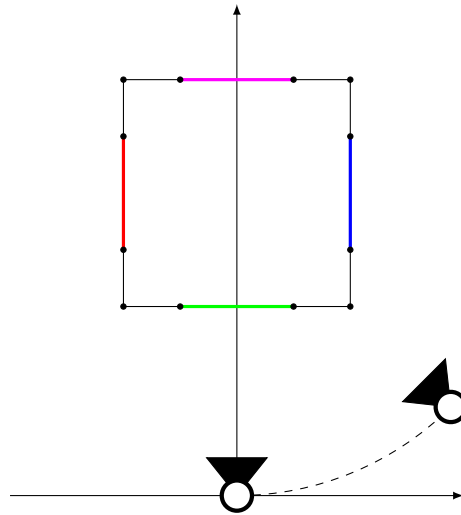
Για τον υπολογισμό των εμβαδών χρησιμοποιείται η θεωρία που αναπτύχθηκε για το δεύτερο μοντέλο. Η γραφική παράσταση του χάρτη φαίνεται στο σχήμα 4.9. Η απόσταση μεταξύ των χρωματιστών καμπυλών στον άξονα  $x = 0$  φανερώνει τα τυφλά σημεία της κάμερας. Κατά την υλοποίηση το ζεύγος των ορατών χρωμάτων ορίζει και το διάστημα επίλυσης.

## 4.7 Πραγματικός Χάρτης

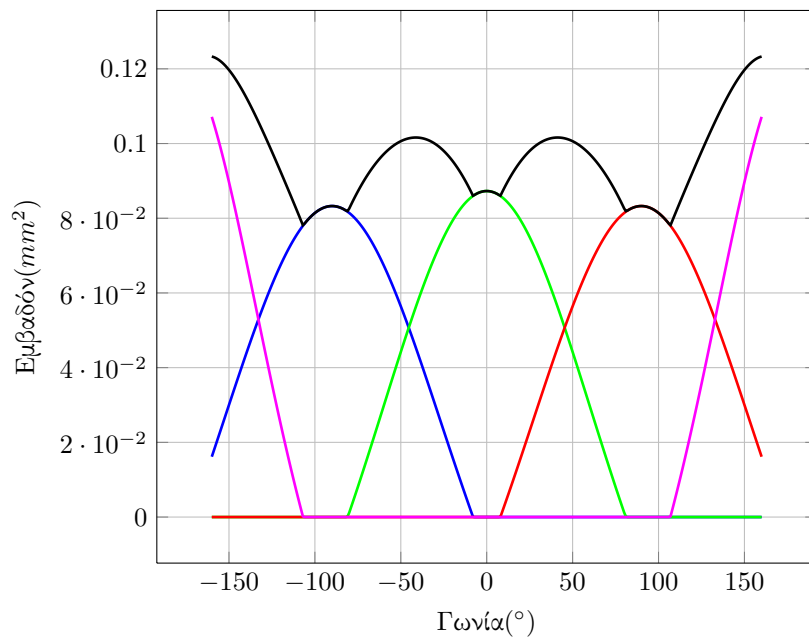
Για την μέτρηση της περιστροφής του ρομπότ εφαρμόσθηκε το μοντέλο που περιγράφεται στην παράγραφο 4.6. Ένα σημείο άξιο προσοχής είναι ότι η διατομή του ρομπότ δεν είναι τετράγωνη αλλά παραλληλόγραμμη. Οι κύκλοι τοποθετούνται σε σχέση με τις κάμερες όπως φαίνεται στο σχήμα 4.10. Η απόσταση της κάμερας από το κέντρο περιστροφής καθορίστηκε με απευθείας μέτρηση και βρέθηκε περίπου ίση με  $94.8 \text{ cm}$ . Οι κύκλοι που χρησιμοποιήθηκαν είχαν ακτίνα ίση με  $4.7 \text{ cm}$ . Τα διανύσματα που χρησιμοποιούνται στο μοντέλο δίνονται από τον πίνακα 4.1 και ο χάρτης που προκύπτει από το σχήμα 4.11.

Πίνακας 4.1: Αρχικές τιμές διανυσμάτων πραγματικού μοντέλου σε *cm*

Τιμές	$r_1$	$r_2$	$g_1$	$g_2$	$b_1$	$b_2$	$m_1$	$m_2$
x	-13.1	-13.1	-4.7	4.7	13.1	13.1	4.7	-4.7
y	4.7	-4.7	-15	-15	-4.7	4.7	27.55	27.55



Σχήμα 4.10: Ανίχνευση χρωμάτων με ταυτόχρονη όραση από δύο κάμερες.



Σχήμα 4.11: Πραγματικός χάρτης επίλυσης του ρομπότ.

#### 4.7.1 Ερμηνεία χάρτη

Είναι εμφανές ότι ο παραγόμενος χάρτης είναι διαφορετικός από τον αναμενόμενο. Πιο συγκεκριμένα ενώ ο χάρτης είναι συμμετρικός με άξονα συμμετρίας την ευθεία  $x = 0$ , η μαύρη καμπύλη που αποτελεί και το άθροισμα των επιμέρους καμπυλών δεν έχει ομοιόμορφη μορφή. Αυτό οφείλεται σε δυο παράγοντες, ο πρώτος είναι η γεωμετρία της άρθρωσης η οποία είναι μορφής ορθογώνιου παραλληλόγραμμου και όχι τετραγώνου και ο δεύτερος είναι η έκκεντρη θέση του άξονα περιστροφής της.

Αυτά τα δεδομένα σε συνδυασμό έχουν ως αποτέλεσμα η πλευρά με τον μωβ κύκλο να έρχεται πιο κοντά στην κάμερα από την πλευρά με πράσινο κύκλο. Η κοντινότερη απόσταση κάνει τον μωβ κύκλο να έχει στην εικόνα μεγαλύτερο εμβαδόν και άρα υψηλότερη καμπύλη από τον πράσινο.

Με τον ίδιο μηχανισμό θέασης η πράσινη καμπύλη είναι ψηλότερη από την κόκκινη και μπλε καμπύλη αλλά εδώ η διαφορά οφείλεται στο γεγονός ότι ο πράσινος κύκλος βρίσκεται σε διαφορετική επιφάνεια από τον κόκκινο και μπλε.

## Κεφάλαιο 5

# Υλοποίηση

Στα προηγούμενα κεφάλαια αναπτύχθηκαν οι αρχές και οι μέθοδοι της τεχνητής όρασης, οι αρχές της ρομποτικής αλλά και η μαθηματική μοντελοποίηση μέσω της οποίας έγινε η επίλυση του προβλήματος. Ωστόσο είναι σημαντικό να παρουσιαστεί λεπτομερώς ο τρόπος με τον οποίο όλες αυτές οι γνώσεις συνδυάστηκαν για την δημιουργία μιας πλατφόρμας η οποία εφαρμόζει όλη την προηγούμενη θεωρία.

### 5.1 Επιλογή εργαλείων

Τα κριτήρια βάσει των οποίων έγινε η επιλογή των εργαλείων ήταν τα εξής:

- Χαμηλό κόστος
- Δυνατότητα παραμετροποίησης
- Υψηλή απόδοση

Το χαμηλό κόστος συνάδει με τον συνολικό στόχο για την δημιουργία ενός οικονομικού συστήματος. Η δυνατότητα παραμετροποίησης ήταν αναγκαία καθώς αρχικά δεν ήταν γνωστές οι προκλήσεις οι οποίες θα έπρεπε να αντιμετωπιστούν και άρα οι ανάγκες σε αλγορίθμους όρασης, ενώ η υψηλή απόδοση θα παρείχε την δυνατότητα εκτέλεσης μεγάλου υπολογιστικού φόρτου με μικρότερες απαιτήσεις σε hardware.

Με βάση τα παραπάνω, απορρίφθηκε η λύση της χρησιμοποίησης του πακέτου Matlab καθώς το κόστος του και η απόδοση του ήταν απαγορευτικά. Μεταξύ των πολλών διαθέσιμων πακέτων σχετικά με την όραση επιλέχθηκε η λύση του OpenCV. Το OpenCV αποτελεί μια multiplatform βιβλιοθήκη τεχνητής όρασης, που σημαίνει πως παρέχει βασικά προγραμματιστικά εργαλεία για την ανάπτυξη εφαρμογών όρασης ενώ ταυτόχρονα ο κώδικας της βιβλιοθήκης διανέμεται υπό την άδεια BSD κάνοντας το OpenCV ιδανικό.

Το επόμενο βήμα ήταν η επιλογή γλώσσας προγραμματισμού. Το OpenCV παρέχει interfaces με πολλές γλώσσες προγραμματισμού. Τα περισσότερο χρησιμοποιούμενα είναι αυτά της C,C++ και python περιορίζοντας τις επιλογές σε μια από της παραπάνω γλώσσες. Η γλώσσα που επιλέχθηκε ήταν η C++ για τους λόγους ότι αποτελεί μια σύγχρονη γλώσσα η οποία προσφέρει ένα σχετικά εύκολο συντακτικό, object oriented προγραμματισμό και καλή απόδοση. Ως compiler και build system χρησιμοποιήθηκε το GNU toolchain(GCC compiler και GNU make)



το οποίο είναι ευρέως γνωστό και χρησιμοποιούμενο. Η ανάπτυξη έγινε σε λειτουργικό σύστημα Windows 7.

Πέρα από τα κοινά εργαλεία που χρησιμοποιήθηκαν, έγινε και η ανάπτυξη ενός απλού εργαλείου βασισμένο σε OpenCV το οποίο καταγράφει συγχρονισμένα video από δυο κάμερες. Η λειτουργία του εργαλείου είναι πολύ απλή και παρουσιάζεται λεπτομερώς στην συνέχεια του κεφαλαίου.

## 5.2 Αρχιτεκτονική της επίλυσης

Το πρόγραμμα της επίλυσης χωρίζεται σε τρία κομμάτια. Την κύρια ρουτίνα, τις κοινές υπορουτίνες, και το κομμάτι του χάρτη. Η κύρια ρουτίνα είναι αυτή η οποία φροντίζει τα παρακάτω:

- Εισαγωγή εικόνων στο πρόγραμμα.
- Προεπεξεργασία δεδομένων.
- Εύρεση χρωμάτων.
- Μεταεπεξεργασία δεδομένων.
- Παρουσίαση αποτελεσμάτων στην οθόνη.
- Αποθήκευση των αποτελεσμάτων στον δίσκο.

Καθώς η κύρια ρουτίνα πραγματοποιεί τα παραπάνω χρησιμοποιεί τις κοινές υπορουτίνες αλλά και τον χάρτη.

Οι κοινές υπορουτίνες περιέχουν τα δεδομένα των διανυσμάτων για κάθε κύκλο, καθώς και συναρτήσεις αρχικοποίησης μεταβλητών. Επιπλέον περιέχει τις υλοποιήσεις των συναρτήσεων οι οποίες δίνουν την επιφάνεια ανά χρώμα, την μέγιστη επιφάνεια ανά χρώμα και την ακτίνα ανά χρώμα. Σημειώνεται εδώ πως πέρα από την μέγιστη επιφάνεια όλες οι άλλες συναρτήσεις εξαρτώνται από την γωνία περιστροφής της άρθρωσης.

Οι ρουτίνες του χάρτη ήταν αρχικά ήταν μια μεγάλη κλάση η οποία είχε υλοποιηθεί για τα αρχικά θεωρητικά μοντέλα και είχε αρκετές λειτουργίες οι οποίες εγκαταλείφθηκαν στην συνέχεια. Βέβαια η ανάγκη για την ύπαρξη της κλάσης συνεχίζει να υπάρχει καθώς έτσι είναι δυνατόν να καταστρωθούν ταυτόχρονα πολλοί διαφορετικοί χάρτες και να επιτελούνται λειτουργίες πάνω σε αυτούς. Οι λειτουργίες που περιέχονται στην κλάση είναι η επιλογή της περιοχής λύσης του χάρτη αλλά και η λύση του χάρτη για συγκεκριμένο ορατό εμβαδόν.

### 5.2.1 Κύρια ρουτίνα

#### Εισαγωγή εικόνων στο πρόγραμμα

Επειδή η επεξεργασία των μετρήσεων γίνεται offline στην ρουτίνα δηλώνονται τα ονόματα των αρχείων βίντεο που περιέχονται οι μετρήσεις. Στην συνέχεια ξεκινά ένας βρόγχος τύπου for ο οποίος ξεκινά από μηδέν και συνεχίζει μέχρι να καλυφθούν όλα τα στιγμιότυπα των βίντεο. Μέσα σε αυτόν τον βρόγχο γίνονται όλοι οι αναγκαίοι υπολογισμοί. Πριν την εκκίνηση του βρόγχου αρχικοποιούνται και τα αρχεία δεδομένων στα οποία θα αποθηκευτούν τα αποτελέσματα και εγγράφονται σε αυτά οι επικεφαλίδες. Προφανώς τα αρχεία αυτά αλλά και τα αρχεία βίντεο κλείνουν μετά το πέρας του βρόγχου.

## Προεπεξεργασία εικόνων

Σκοπός της προεπεξεργασίας εικόνας είναι μια αρχική μορφοποίηση των δεδομένων έτσι ώστε να είναι πιο εύκολη η κυρίως επεξεργασία. Η προεπεξεργασία γίνεται σε δύο βήματα. Το πρώτο είναι ο καθορισμός μιας περιοχής ενδιαφέροντος και το δεύτερο είναι η εφαρμογή μιας θόλωσης Gauss. Η περιοχή ενδιαφέροντος χρησιμοποιείται έτσι ώστε η κύρια επεξεργασία να γίνεται σε ένα μικρότερο μέρος της εικόνας. Η τεχνική αυτή χρησιμοποιείται συχνά στις εφαρμογές της τεχνητής όρασης και εξοικονομεί αρκετούς υπολογιστικούς πόρους.

Στο σημείο αυτό σημειώνεται πως οι συντεταγμένες της περιοχής ορίζονται με τέσσερις παραμέτρους εκ των οποίων οι δύο είναι οι συντεταγμένες  $(x, y)$  της γωνίας του ορθογωνίου πάνω στην εικόνα και οι άλλες δύο είναι το μήκος και πλάτος του ορθογωνίου. Οι παράμετροι αυτές δεν είναι γνωστές αλλά είναι δυνατόν να επιλεγθούν κάνοντας δοκιμές. Το κριτήριο για την επιλογή τους είναι να είναι ορατή η τροχιά των κύκλων στην ελάχιστη δυνατή επιφάνεια. Προφανώς κάθε φορά που γίνεται διαφορετική μέτρηση και άρα διαφορετική τοποθέτηση της κάμερας οι παράμετροι αλλάζουν.

Η εφαρμογή της θόλωσης Gauss εφαρμόζεται για την μείωση του θορύβου αλλά και για να υπάρχουν πιο ομαλά χρώματα μέσα στο σώμα του κύκλου. Οι παράμετροι της θόλωσης είναι τέτοιες ώστε αυτή να μην είναι πολύ έντονη και να μην αλλοιώνεται η λαμβανόμενη πληροφορία.

## Εύρεση χρωμάτων

Η εύρεση χρωμάτων είναι μια σημαντική υπορουτίνα. Σκοπός της όπως υποδηλώνει είναι η ανίχνευση χρωμάτων. Η ανίχνευση βασίζεται στις τιμές των καναλιών των pixel. Οι τιμές αυτές δίνονται στο `rgb colorspace`<sup>1</sup>. Το εύρος τιμών για τις οποίες γίνεται η αναζήτηση βρίσκεται και αυτό με δοκιμή. Οι τιμές επιλέγονται με τέτοιο τρόπο ώστε να γίνεται σωστή εύρεση των χρωμάτων. Στο σημείο αυτό παίζει ρόλο η σωστή επιλογή της περιοχής ενδιαφέροντος ώστε σε αυτήν να μην εισάγονται περιοχές συγγενούς χρώματος με αυτό που αναζητάται και να μην υπάρχουν εσφαλμένες ανιχνεύσεις.

Τεχνικά η ανίχνευση γίνεται περνώντας τις τιμές κάθε pixel από κάποιο λογικό φίλτρο. Με την έννοια λογικό φίλτρο εννοείται μια σειρά λογικών προτάσεων οι οποίες συνδυάζονται με τον λογικό τελεστή AND. Εάν οι προτάσεις ισχύουν τότε το pixel μαρκάρεται με μια ακραία τιμή χρώματος για εύκολη διαλογή στην συνέχεια.

Η υλοποίηση της εύρεσης χρωμάτων γίνεται για τα βασικά χρώματα με απαιτήσεις να βρίσκονται οι τιμές των *rgb* μέσα σε κάποια όρια και να ισχύει  $g > b$  για το πράσινο,  $b > g$  για το μπλε. Με αντίστοιχες προτάσεις ανιχνεύεται και το κόκκινο. Για τα χρώματα που δεν είναι βασικά όπως το κυανό, μωβ και κίτρινο η εύρεση είναι δυνατή χρησιμοποιώντας τις απόλυτες τιμές των διαφορών των βασικών χρωμάτων που τα απαρτίζουν. Ωστόσο στην παρούσα εφαρμογή υλοποιείται μόνο η εύρεση του πράσινου και του μπλε.

Η ανίχνευση χρωμάτων είναι γενικά ευαίσθητη στον φωτισμό συνεπώς σε διαφορετικά περιβάλλοντα φωτισμού αλλάζουν και οι τιμές των ορίων με βάση τα οποία πρέπει να γίνονται οι συγχρίσεις.

<sup>1</sup>Είναι δυνατή η μετατροπή της εικόνας σε κάποιο άλλο colorspace και αναζήτηση σε εκείνο το colorspace

## Μεταεπεξεργασία εικόνων

Στην μεταεπεξεργασία εικόνων η εικόνα στην οποία πλέον έχει γίνει η ανίχνευση χρωμάτων χωρίζεται στα κανάλια που την απαρτίζουν. Στο κάθε κανάλι που προκύπτει εφαρμόζεται η τεχνική του thresholding έτσι ώστε να απομονωθούν τα εντοπισμένα pixels. Για την απομάκρυνση όποιων μεμονωμένων λάθος ανιχνύσεων εφαρμόζεται η διαδικασία της διάβρωσης. Για μια ακόμα φορά οι παράμετροι της διάβρωσης προκύπτουν με δοκιμές φροντίζοντας για την απομάκρυνση του θορύβου ελαχιστοποιώντας την απώλεια πληροφορίας.

Ένα σημείο το οποίο πρέπει να προσεχθεί είναι πως το OpenCV δημιουργεί εικόνες με τρία κανάλια. Αυτό δεν είναι αρκετό όταν πρέπει να ανιχνευθούν πάνω από τρία χρώματα. Το πρόβλημα αντιμετωπίζεται εάν κατά την προεπεξεργασία γίνει μεταφορά των δεδομένων σε μια εικόνα η οποία έχει αριθμό καναλιών ίσο με τον αριθμό χρωμάτων που πρέπει να εντοπισθούν.

### Εύρεση εμβαδού

Μετά την μεταεπεξεργασία είναι δυνατή η εύρεση εμβαδού αφού πλέον είναι γνωστός ο αριθμός των pixels που αντιστοιχούν σε κάθε χρώμα. Αυτό γίνεται με βάση την παρακάτω σχέση:

$$A = \frac{A_{max} \cdot P}{P_{max}} \quad (5.1)$$

όπου  $A_{max}$  είναι η μέγιστη επιφάνεια που μπορεί να εμφανισθεί στον αισθητήρα της κάμερας και είναι διαφορετική για κάθε χρώμα και  $P$  ο τρέχων αριθμός pixels του παρακολουθούμενου χρώματος και  $P_{max}$  ο μέγιστος αριθμός pixels που εμφανίζονται για το παρακολουθούμενο χρώμα. Ο αριθμός αυτός βρίσκεται κατά την διαδικασία της δοκιμαστικής βιντεοσκόπησης.

### 5.2.2 Κοινές υπορουτίνες

Οι κοινές υπορουτίνες χρησιμοποιούνται τόσο στην επίλυση του χάρτη αλλά και στην κύρια ρουτίνα. Σε αυτές περιέχονται συναρτήσεις οι οποίες κρίθηκαν ότι μπορούν να χρησιμοποιηθούν σε διάφορα μέρη του κώδικα του προγράμματος όταν αυτό θα επεκταθεί με επιπλέον μεθόδους.

### Δήλωση μεταβλητών

Οι μεταβλητές πρωτοτυποποιούνται με τον χαρακτηρισμό external έτσι ώστε η αρχικοποίηση τους να γίνεται αφενός σε διαφορετικό σημείο στον κώδικα αφετέρου να επιτυγχάνεται η χρήση τους ως global μεταβλητές. Οι μεταβλητές που αρχικοποιούνται είναι τα διανύσματα των κύκλων.

### Συναρτήσεις "Εργαλεία"

Είναι κάποιες βασικές συναρτήσεις όπως οι μετατροπές των μοιρών σε ακτίνια για την χρήση τους σε τριγωνομετρικές συναρτήσεις αλλά και η μετατροπή των ακτινίων σε μοίρες για παρουσίαση των αποτελεσμάτων.

## Γεωμετρικοί υπολογισμοί

Είναι μια οικογένεια συναρτήσεων οι οποίες υλοποιούν την μαθηματική θεωρία η οποία αναπτύχθηκε στο κεφάλαιο 4. Οι συναρτήσεις που υπάρχουν είναι αυτές οι οποίες υπολογίζουν την ορατή ακτίνα, το ορατό εμβαδόν αλλά και το μέγιστο εμβαδόν ανά χρώμα. Το τελευταίο δεν είναι συνάρτηση της γωνίας καθώς το μέγιστο εμβαδόν για κάθε χρώμα εμφανίζεται σε μια συγκεκριμένη γωνία περιστροφής ανάλογα με το χρώμα που εξετάζεται.

### 5.2.3 Κλάση Χάρτη

Η κλάση του χάρτη επιτρέπει την δυνατότητα συνύπαρξης πολλών διαφορετικών χαρτών. Οι συναρτήσεις της είναι τρεις. Η μια είναι ο constructor ο οποίος ορίζει και την περιοχή επίλυσης της. Η δεύτερη είναι μια συνάρτηση η οποία αλλάζει την περιοχή επίλυσης και η τρίτη είναι μια συνάρτηση η οποία λύνει τον χάρτη. Προφανώς κατά την επίλυση χρησιμοποιούνται οι κοινές υπορουτίνες.

Η μέθοδος που επιλέχθηκε ήταν η μέθοδος της διχοτόμησης. Ο λόγος που επιλέχθηκε έναντι της newton-raphson, η οποία είναι γρηγορότερη, είναι επειδή αρχικά είχε υλοποιηθεί για χάρτη του οποίου η παράγωγος δεν ήταν συνεχής και έτσι η newton-raphson δεν εγγυάται πάντα σύγκλιση.

## 5.3 Διάταξη

Η πειραματική διάταξη στην οποία πραγματοποιήθηκε το πείραμα αποτελούνταν από το τον ρομποτικό βραχίονα και τις δύο κάμερες. Η μια κάμερα στηρίζεται στον τοίχο απέναντι από τον βραχίονα και η δεύτερη στηρίζεται στην ίδια απόσταση αλλά με διαφορετική γωνία. Οι δύο κάμερες στηρίζονται διαφορετικά. Η κάμερα του τοίχου στηρίζεται με χρήση μεταλλικού ελάσματος το οποίο σχηματίζει πρόβολο. Στην άκρη του προβόλου προσαρμόζεται η κάμερα με χρήση του ενσωματωμένου πλαστικού της clip.

Η κάμερα του τοίχου στηρίζεται με την βοήθεια ενός φωτογραφικού τρίποδα. Η στερέωση της γίνεται στην κοιλότητα που υπάρχει στο πάνω μέρος του τρίποδα. Επειδή η στερέωση δεν είναι στιβαρή, ασφαλίζεται επιπλέον με την χρήση κολλητικής ταινίας. Οι πολλοί βαθμοί ελευθερίας του τρίποδα επιτρέπουν την εξασφάλιση ικανοποιητικού προσανατολισμού για την θέαση του βραχίονα.

Ένα σημείο το οποίο αξίζει να σημειωθεί είναι πως υπάρχουν κάποια πρακτικά όρια θέασης του βραχίονα για τα οποία ο κώδικας παράγει αποτελέσματα. Για παράδειγμα η τοποθέτηση της δεύτερης κάμερας σε απόσταση  $20^\circ$  από την πρώτη δεν αποφέρει οφέλη καθώς ακόμα και σε αυτές τις μοίρες δεν υπάρχει μεγάλη διαφορά στην θέαση. Η εμπειρία έδειξε πως η βέλτιστη γωνία απόστασης μεταξύ των καμερών είναι οι  $40^\circ - 45^\circ$ .

Η στήριξη των στόχων αν και απλή αποδείχθηκε αρκετά επίπονη. Σχεδόν όλες οι κολλητικές ταινίες οι οποίες δοκιμάστηκαν απέτυχαν να κολλήσουν τους στόχους. Έτσι αρχικά χρησιμοποιήθηκε κολλητική πάστα blue tac η οποία όμως εισήγαγε μεγάλες παραμορφώσεις στον στόχο. Τελικά και μετά από πολύ έρευνα αλλά και τύχη χρησιμοποιήθηκε μια κολλητική ταινία διπλής όψευς η οποία προσκολλούσε τους στόχους χωρίς παραμόρφωση για αρκετό χρονικό διάστημα ώστε να γίνει τέλεση των πειραμάτων.



Σχήμα 5.1: Η πειραματική διάταξη. Οι κάμερες απέχουν  $20^\circ$ .



Σχήμα 5.2: Οι κάμερες της διάταξης όπως φαίνονται από την άρθρωση του ρομπότ.



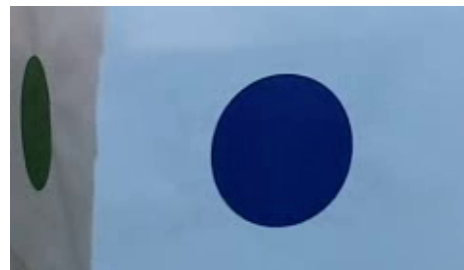
(α') Αριστερή αρχική εικόνα



(β') Δεξιά αρχική εικόνα



(γ') Αριστερό ROI



(δ') Δεξί ROI



(ε') Αριστερή προεπεξεργασία



(στ') Δεξιά προεπεξεργασία



(ζ') Αριστερή ανίχνευση



(η') Δεξιά ανίχνευση

Σχήμα 5.3: Ανίχνευση χρωμάτων από τις δυο κάμερες



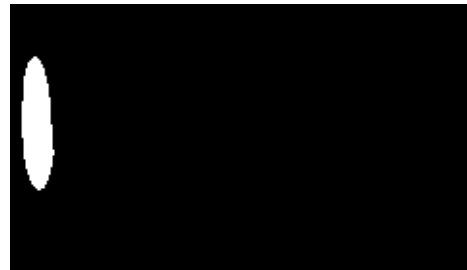
(α) Αριστερό πράσινο χωρίς διάβρωση



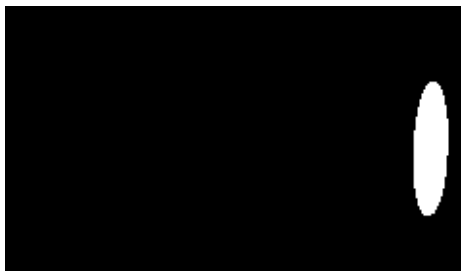
(β') Δεξί πράσινο χωρίς διάβρωση



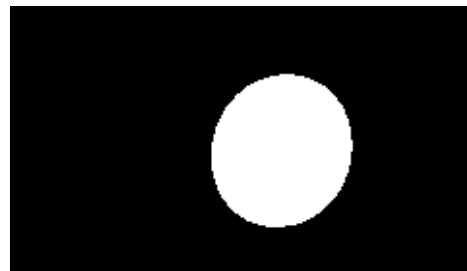
(γ') Αριστερό πράσινο με διάβρωση



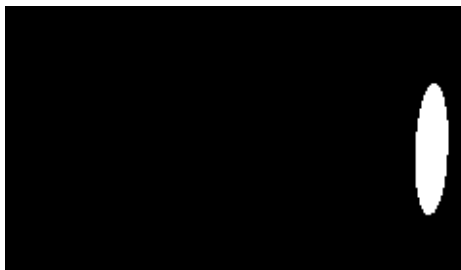
(δ') Δεξί πράσινο με διάβρωση



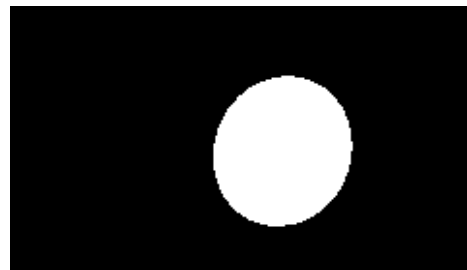
(ε') Αριστερό μπλε χωρίς διάβρωση



(στ') Δεξί μπλε χωρίς διάβρωση



(ζ') Αριστερό μπλε με διάβρωση



(η') Δεξί μπλε με διάβρωση

Σχήμα 5.4: Επεξεργασία ανιχνευμένων χρωμάτων από τις δυο κάμερες

## 5.4 Λοιπές Λεπτομέρειες Υλοποίησης

Μια από τις βασικότερες δυσκολίες υλοποίησης ήταν ο φωτισμός στον χώρο εργασίας του ρομπότ καθώς η υπάρχουσα εγκατάσταση του δημιουργούσε πολλές σκιές πάνω στους στόχους αλλάζοντας τις τιμές των παρακολουθούμενων pixel. Το πρόβλημα αυτό περιορίστηκε κάπως με την χρήση πρόσθετης πηγής φωτισμού σε διαφορετικό σημείο του χώρου εργασίας.

Ένα δεύτερο πρόβλημα που υπήρξε με τον φωτισμό ήταν συχνότητα λειτουργίας του. Επειδή η συχνότητα δειγματοληψίας της κάμερας βρίσκεται πολύ κοντά στην συχνότητα του δικτύου η οποία είναι  $50Hz$ , εμφανίζεται κατά την λήψη βίντεο το φαινόμενο του flickering. Το flickering είναι η παρουσίαση κινούμενων φωτεινότερων και σκοτεινότερων περιοχών στο βίντεο οι οποίες μετακινούνται στην κατακόρυφη διεύθυνση κατά την διάρκεια αυτού. Η επίλυση του προβλήματος γίνεται αλλάζοντας την συχνότητα δειγματοληψίας της κάμερας είτε ευθέως, είτε με την στρατηγική απόρριψη συγκεκριμένων εικόνων της ακολουθίας. Εν προκειμένω η επίλυση γίνεται αυτόματα με την χρήση της δεύτερης μεθόδου από το πρόγραμμα οδήγησης της χρησιμοποιούμενης κάμερας.

Ένα τρίτο σημείο δυσκολιών με τον φωτισμό αλλά και τον συγκεκριμένο εξοπλισμό ήταν η ύπαρξη κάποιων χαρακτηριστικών των καμερών με τις ονομασίες Rightlight και Autofocus. Τα χαρακτηριστικά αυτά βελτιώνουν αυτόματα την εικόνα σε κοινές χρήσεις όπως είναι η βιντεοκλήση μέσω δικτύου, αλλά δυσκολεύουν τα πράγματα για τις εφαρμογές της όρασης καθώς αλλάζουν τις παραμέτρους των μετρήσεων. Το χαρακτηριστικό autofocus απενεργοποιήθηκε για την σταθεροποίηση της εστιακής απόστασης αλλά το rightlight παρέμεινε ενεργό για την διόρθωση του φωτισμού καθώς αυτός κρίθηκε ανεπερκής.

Επιπρόσθετα για την αποφυγή διαχείρισης πολλών χαρτών, έναν για κάθε κάμερα, γίνεται η αντίστοιχη μετατόπιση των λύσεων που προκύπτουν από την δεύτερη κάμερα. Για τις κάμερες οι οποίες είναι μετατοπισμένες κατά  $n^\circ$  η μετατόπιση τους αφαιρείται από το αποτέλεσμα της γωνίας.

## 5.5 Εργαλείο λήψης μετρήσεων

Ο κώδικας λήψης μετρήσεων είναι απλός. Αυτό που κάνει είναι να λαμβάνει ταυτόχρονα μετρήσεις και από τις δυο κάμερες. Το εργαλείο λειτουργεί σε δυο διαφορετικές μορφές. Η πρώτη είναι γενική βιντεοσκόπηση η οποία σταματά όταν πατηθούν τα πλήκτρα esc ή enter. Η δεύτερη μορφή που χρησιμοποιείται είναι η μορφή αποσπασματικών μη συνεχών μετρήσεων. Στην μορφή αυτή γίνεται ένας αριθμός μετρήσεων με λίγα frames ο καθένας. Η μορφή αυτή χρησιμοποιήθηκε για την πειραματική μέτρηση των σφαλμάτων.



# Κεφάλαιο 6

## Αποτελέσματα

### 6.1 Μετρήσεις

Ως απόδειξη της λειτουργίας της μεθόδου, οι μετρήσεις γίνονται σε ένα τεταρτημόριο της ολικής τροχιάς του ρομπότ και όχι σε όλη την περιστροφή της. Για την πραγματοποίηση του χρησιμοποιήθηκαν δύο κάμερες οι οποίες απέχουν μεταξύ τους κατά  $40^\circ$  μοίρες.

#### 6.1.1 Εξοπλισμός μετρήσεων

Το ρομπότ που χρησιμοποιήθηκε είναι το Staübli RX90 που είναι ένας βραχίονας 6 περιστροφικών αρθρώσεων και συνεπώς ίδιου αριθμού βαθμών ελευθερίας. Για τις ανάγκες των μετρήσεων χρησιμοποιήθηκε μόνο η πρώτη άρθρωση η οποία βρίσκεται στην βάση του ρομπότ.

Οι κάμερες που χρησιμοποιήθηκαν είναι δυο Logitech C525HD οι οποίες είναι τυπικές mid range κάμερες που πωλούνται στα καταστήματα ηλεκτρονικών ειδών. Γνωρίσματα τους είναι η ύπαρξη φακού ο οποίος εισάγει πολύ μικρές οπτικές παραμορφώσεις, η λήψη βίντεο υψηλής ευκρίνειας και η ύπαρξη μικροφώνου. Το εστιακό τους μήκος είναι ίσο με 2.83 mm.

#### 6.1.2 Διαδικασία μετρήσεων

Για την τοποθέτηση των καμερών το ρομπότ έρχεται στην θέση η οποία έχει οριστεί ως μηδέν από το θεωρητικό μοντέλο και αποτελεί την θέση εκκίνησης. Η πρώτη κάμερα τοποθετείται απέναντι από την επιφάνεια του πράσινου κύκλου. Ιδανικά το διάνυσμα της κάμερας θα είναι στην ίδια διεύθυνση με το διάνυσμα της επιφάνειας. Φυσικά αυτό δεν δυνατόν και οι συνέπειες θα αναλυθούν στην ενότητα των σφαλμάτων.

Για την τοποθέτηση της δεύτερης κάμερας, το ρομπότ στρέφεται κατά  $n^\circ$ . Όπου  $n^\circ$  μοίρες είναι η γωνιακή απόσταση η οποία είναι επιθυμητή. Στην συνέχεια με βοήθεια μετρητικού ελάσματος σημειώνεται η διεύθυνση η οποία ενώνει το κέντρο περιστροφής της άρθρωσης με το κέντρο του φακού της κάμερας. Στην συνέχεια επιστρέφεται το ρομπότ στην αρχική του θέση. Πλέον η σημειωμένη διεύθυνση δηλώνει την διεύθυνση στην οποία πρέπει να τοποθετηθεί η δεύτερη κάμερα. Προφανώς η απόσταση από το κέντρο πρέπει να είναι ίση με αυτήν της πρώτης κάμερας.



Σχήμα 6.1: Το ρομπότ Staubli Rx90



Σχήμα 6.2: Η Logitech C525HD



Σχήμα 6.3: Μαρκάρισμα ρομπότ σε θέση στραμμένη ως προς την πρώτη κάμερα



Σχήμα 6.4: Σημάδι από μετρητικό έλασμα





Σχήμα 6.5: Επανατοποθέτηση του ελάσματος στο σημάδι του



Σχήμα 6.6: Τοποθέτηση της δεύτερης κάμερας στο ίδιο ύψος

Μετά την τοποθέτηση των καμερών γίνεται μια αρχική βιντεοσκόπηση της κίνησης της άρθρωσης. Σκοπός είναι η εύρεση του μέγιστου αριθμού pixels ανά παρακολουθούμενο χρώμα. Ο αριθμός αυτός χρησιμοποιείται για την εύρεση του εμβαδού των κύκλων κατά την διάρκεια της παρακολούθησης.

Οι μετρήσεις δεν λαμβάνονται online αλλά προσομοιώνονται. Δηλαδή δεν λαμβάνονται σε πραγματικό χρόνο, αλλά υπο μορφή συγχρονισμένων video τα οποία στην συνέχεια επεξεργάζονται από το πρόγραμμα μέτρησης. Αυτό έχει πολλαπλά οφέλη. Μεταξύ αυτών είναι η μικρή συχνότητα στησίματος του εξοπλισμού, οι εύκολες αλλαγές στο θεωρητικό μοντέλο χωρίς την ανάγκη λήψης νέων μετρήσεων και τέλος η εύκολη σύγκριση μεταξύ μετρήσεων από διαφορετικές κάμερες.

Για την λήψη των δεδομένων αναπτύχθηκε κώδικας ο οποίος λαμβάνει βίντεο ταυτόχρονα και από τις δυο κάμερες. Εν προκειμένω η απαίτηση που υπάρχει εδώ είναι τα δύο βίντεο να αναπαριστούν την ίδια θέση του ρομπότ από διαφορετικές οπτικές γωνίες, δηλαδή να είναι συγχρονισμένα.

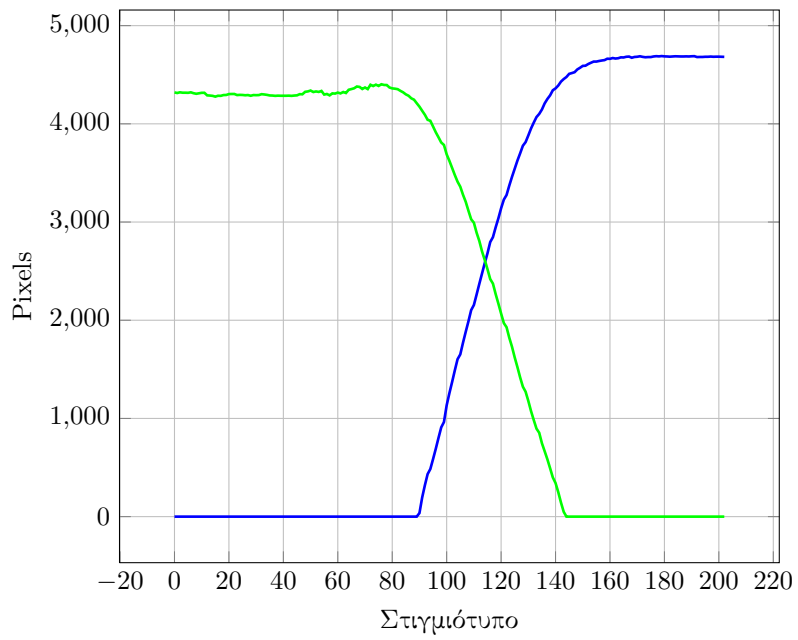
## 6.2 Αποτελέσματα

Οι πειραματικές μετρήσεις γίνονται για ένα εύρος κίνησης ίσο με  $90^\circ$ . Επιπρόσθετα η φορά του ρομπότ επιλέχθηκε αντιωρολογιακή και έτσι αναμένεται μέτρηση αρνητικής γωνίας. Το βίντεο ξεκινά με το ρομπότ ακίνητο στην συνέχεια το απεικονίζει κατά την περιστροφή του ενώ στο τέλος παραμένει ακίνητο. Αρχικά δεν έχουν και οι δύο κάμερες θέαση, αλλά μόνο η δεξιά. Στην συνέχεια και καθώς η άρθρωση περιστρέφεται, αποκτά θέαση και η αριστερή κάμερα. Οι μετρήσεις καταχωρούνται όταν ικανοποιείται η συνθήκη θέασης. Αυτή είναι να είναι ορατά τουλάχιστον 500 pixels από κάθε παρακολουθούμενο χρώμα. Ο αριθμός αυτός επιλέχθηκε αυθαίρετα, ωστόσο λαμβάνεται υπόψη να μην είναι πολύ μικρός ώστε να είναι ευαίσθητος στον θόρυβο, αλλά και όχι αρκετά μεγάλος ώστε να χάνεται έγκυρη πληροφορία. Στα σχήματα 6.7 έως 6.15 παρουσιάζεται ο αριθμός των pixels όπως αυτά καταχωρούνται για όλο το βίντεο χωρίς συνθήκες θέασης. Τα υπόλοιπα σχήματα είναι δεδομένα τα οποία καταχωρούνται υπό συνθήκες θέασης.

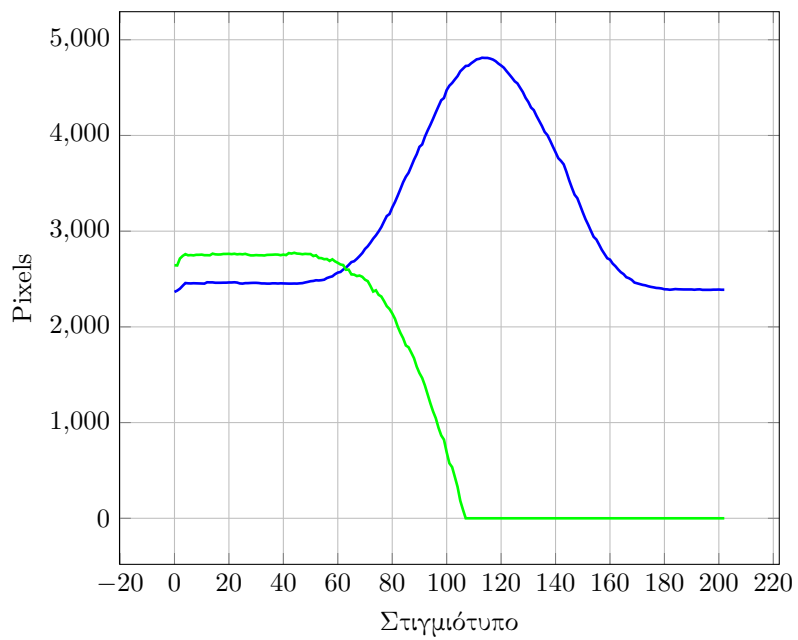
### Ερμηνεία αποτελεσμάτων

Ξεκινώντας από τα σχήματα 6.7 και 6.8 είναι εμφανές ότι ακόμα και όταν το ρομπότ παραμένει ακίνητο, ο αριθμός των pixels τα οποία εμπίπτουν στα χρωματικά πλαίσια του πρασίνου και μπλε είναι μεταβλητά. Αυτό φαίνεται κυρίως από την καμπύλη του πράσινου χρώματος. Η παρατήρηση αυτή οδηγεί στο συμπέρασμα πως όχι μόνο υπάρχουν μεταβλητότητες, αλλά αυτές είναι και διαφορετικές μεταξύ διαφορετικών χρωμάτων.

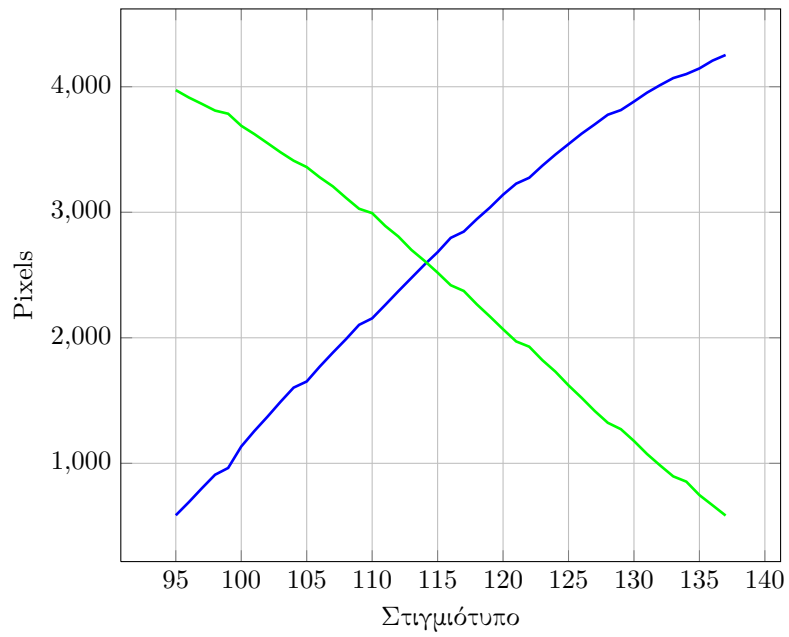
Οι μεταβλητότητες αυτές έχουν ως αποτέλεσμα διαφορετικές καμπύλες μετρούμενης γωνίας. Επιπρόσθετα είναι εμφανές από το σχήμα 6.15 πως οι καμπύλες μετρήσεων της δεξιάς και της αριστερής κάμερας έχουν μια κοινή περιοχή από μετρήσεις οι οποίες ιδανικά θα έπρεπε να ταυτίζονται για το ίδιο στιγμιότυπο. Στην πράξη όμως αυτό δεν συμβαίνει καθώς δεν έχουν οι κάμερες το ίδιο σφάλμα. Βέβαια οι μετρήσεις, στην περιοχή όπου υπάρχει ορατότητα και από τις δυο κάμερες έχουν την ίδια κλίση δείχνοντας την ορθότητα του μοντέλου.



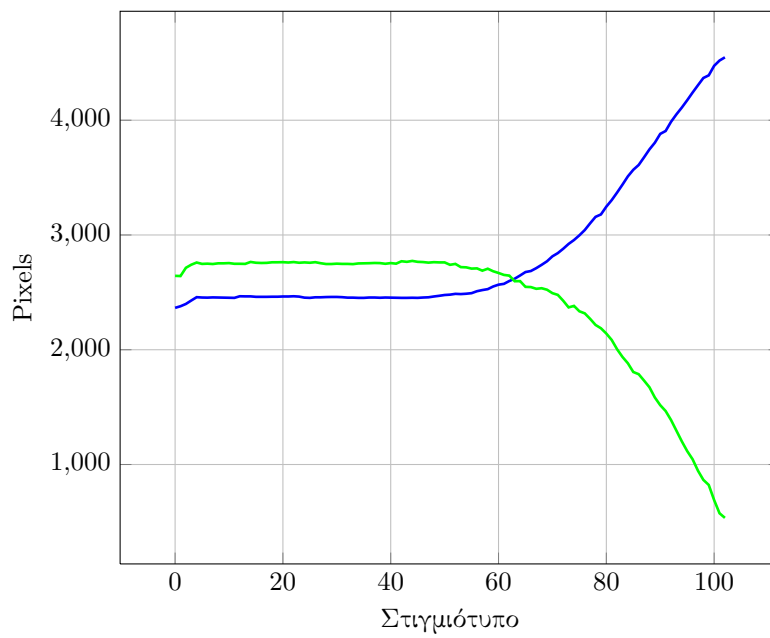
Σχήμα 6.7: Ορατά pixels από την αριστερή κάμερα χωρίς συνθήκη θέασης.



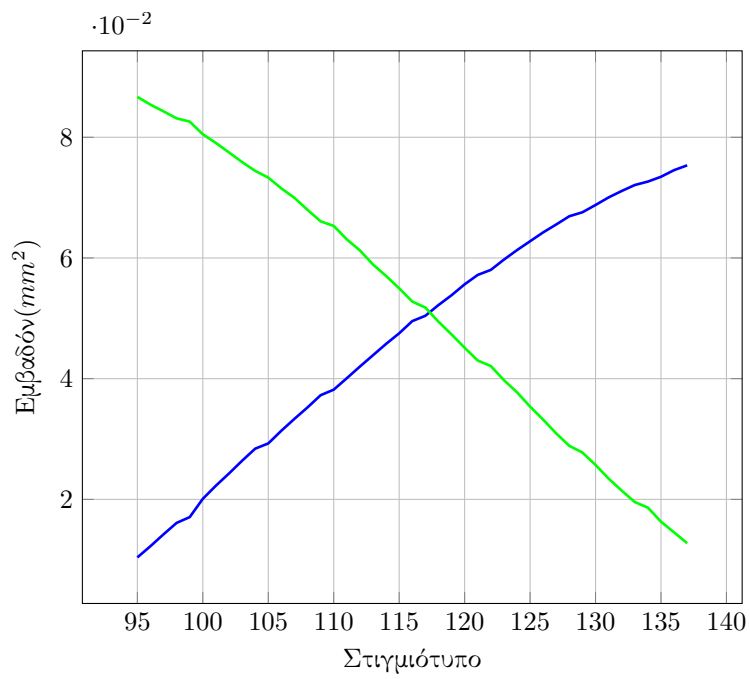
Σχήμα 6.8: Ορατά pixels από την δεξιά κάμερα χωρίς συνθήκη θέασης.



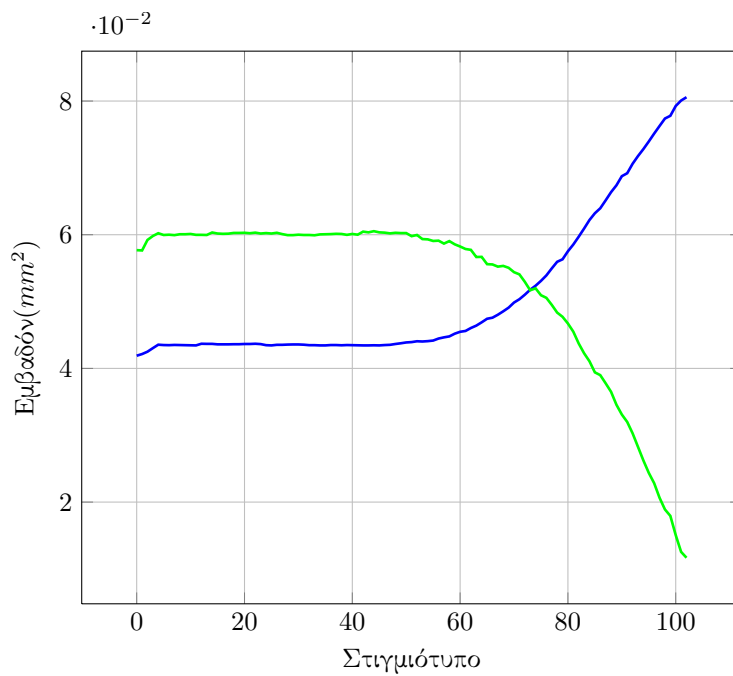
Σχήμα 6.9: Ορατά pixels από την αριστερή κάμερα με συνθήκη θέασης.



Σχήμα 6.10: Ορατά pixels από την δεξιά κάμερα με συνθήκη θέασης.

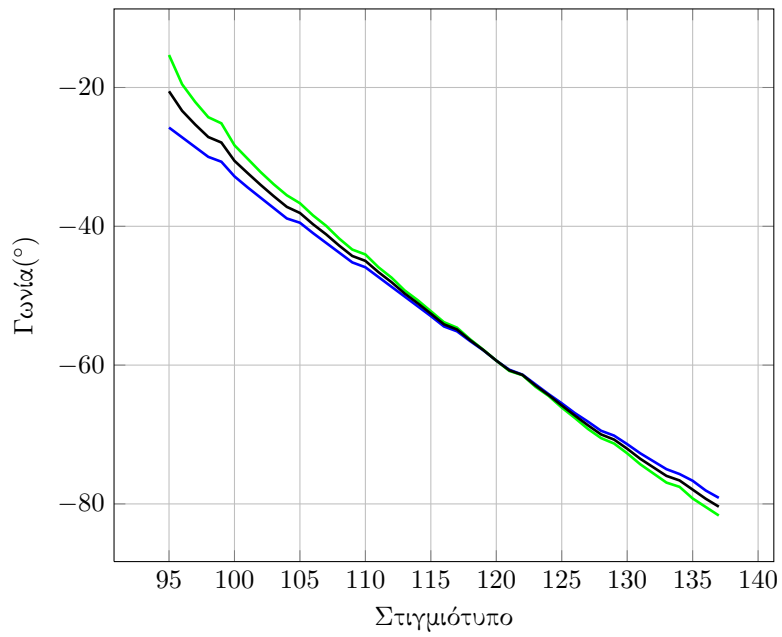


Σχήμα 6.11: Ορατό εμβαδόν από την αριστερή κάμερα.

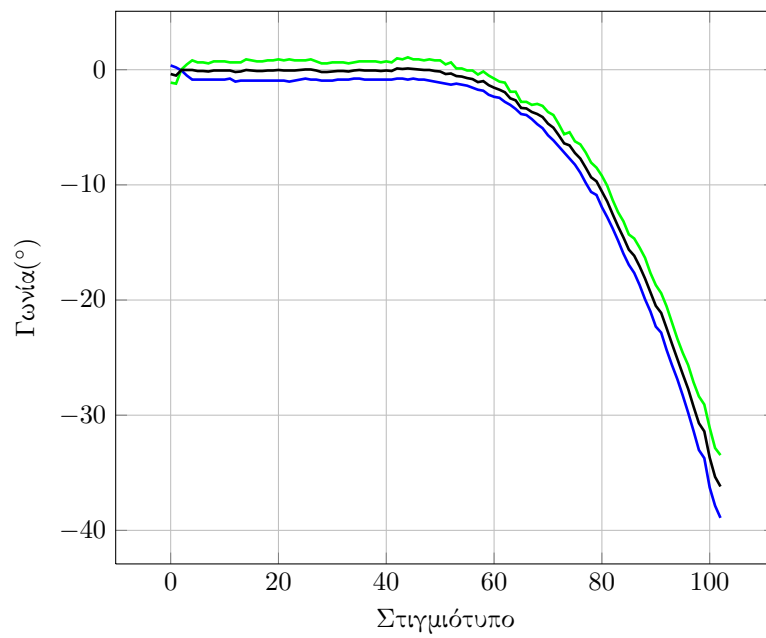


Σχήμα 6.12: Ορατό εμβαδόν από την δεξιά κάμερα.

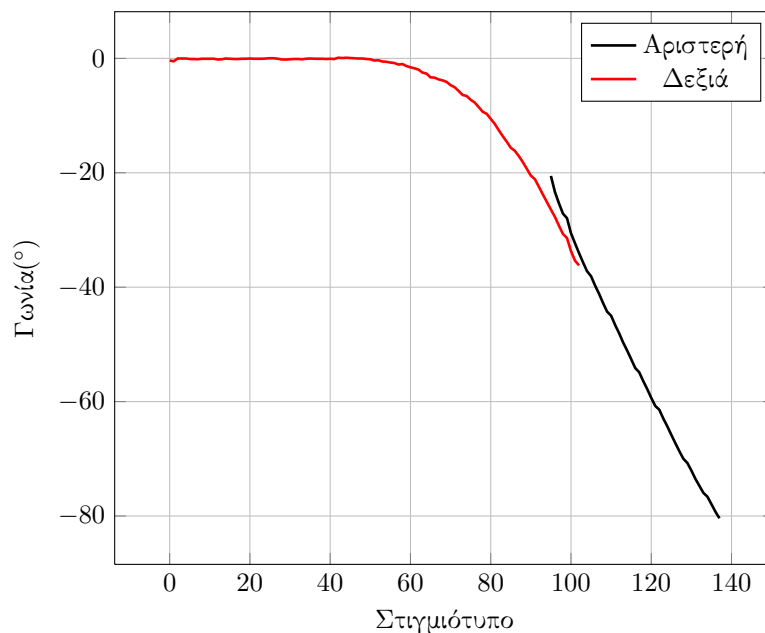




Σχήμα 6.13: Λύσεις γωνίας αριστερής κάμερας και ο μέσος όρος τους.



Σχήμα 6.14: Λύσεις γωνίας δεξιάς κάμερας και ο μέσος όρος τους.



Σχήμα 6.15: Λύσεις των δυο καμερών

## 6.3 Σφάλματα

Είναι προφανές πως όπως σε κάθε μετρητική διάταξη, έτσι και εδώ υπάρχουν σφάλματα. Εν προκειμένω δεν υπάρχει μοναδική πηγή σφάλματος, αλλά και δεν είναι όλες οι πηγές ίδιες ως προς την συμπεριφορά τους. Στην παρούσα ενότητα γίνεται μια παρουσίαση των διάφορων σφαλμάτων και προτείνεται τρόπος επιδιόρθωσής τους.

### 6.3.1 Φωτισμός

Ένα βασικό πρόβλημα στις εφαρμογές όρασης είναι ο φωτισμός. Συχνά σε βιομηχανικές εφαρμογές εφαρμόζεται ειδικός φωτισμός ο οποίος ενισχύει τα χαρακτηριστικά τα οποία πρέπει να ανιχνευθούν και να επεξεργαστούν. Στην εξεταζόμενη εφαρμογή, δεν υπήρχαν πολλές επιλογές φωτισμού και αυτό διότι ο χώρος στον οποίο έχει εγκατασταθεί το ρομπότ είχε συγκεκριμένη τοποθέτηση λαμπτήρων. Η παρούσα εγκατάσταση δημιούργησε μεγάλες σκιάσεις αλλά και φαινόμενα glaring τα οποίες δυσχέραιναν την ανίχνευση χρώματος.

Δυστυχώς τα φαινόμενα του φωτισμού είναι αρκετά χαστικά για να μελετηθούν. Έτσι αντιμετωπίζονται εμπειρικά δοκιμάζοντας διάφορους τύπους φωτισμού. Εν προκειμένω η καλύτερη ανίχνευση χρωμάτων απαιτεί ομοιόμορφο φωτισμό που προκύπτει από ειδικούς ανακλαστήρες και όχι συγκεκριμένες φωτεινές πηγές όπως είναι οι λαμπτήρες. Πρακτικά αυτό μεταφράζεται σε πολλούς λαμπτήρες μέσα στον χώρο εργασίας του ρομπότ.

### 6.3.2 Παραμορφώσεις στόχου

Ένα ακόμα φαινόμενο το οποίο είναι χαοτικό είναι οι παραμορφώσεις του στόχου. Με τον όρο αυτό χαρακτηρίζονται αφενός οι παραμορφώσεις που προκύπτουν από την μηχανική παραμόρφωση του στόχου κατά την τοποθέτηση του και αφετέρου οι χρωματικές παραμορφώσεις που προκύπτουν κατά την εκτύπωση των στόχων. Οι μηχανικές παραμορφώσεις δημιουργούνται κατά την επικόλληση του στόχου πάνω στο σώμα του ρομπότ ενώ οι χρωματικές παραμορφώσεις οφείλονται σε σφάλματα εκτύπωσης του στόχου. Οι χρωματικές παραμορφώσεις που συναντήθηκαν είναι η μη ομοιομορφία του χρώματος σε όλο τον στόχο αλλά και αλλαγή χρώματος από πράσινο σε μπλε σε μερικά σημεία στα άκρα του στόχου.

Ο προτεινόμενος τρόπος διόρθωσης είναι η αποτύπωση του στόχου με μονοχρωματική ματ βαφή στο κέντρο της κάθε πλευράς του ρομπότ. Με τον τρόπο αυτό καταπολεμάται το ζήτημα του glaring που προκαλείται από τον φωτισμό και στην ουσία εκμηδενίζονται τα σφάλματα παραμόρφωσης στόχου.

### 6.3.3 Απόκλιση θέσης

Η επίλυση του θεωρητικού μοντέλου βασίζεται στην σωστή τοποθέτηση των καμερών σε σχέση με τους στόχους. Φυσικά η απόλυτη τοποθέτηση είναι αδύνατη. Ωστόσο είναι δυνατή η τοποθέτηση καμερών με τρόπους τέτοιους ώστε τα σφάλματα απόκλισης θέσης να είναι αμελητέα. Εν προκειμένω προτείνονται δυο τρόποι. Και οι δύο προϋποθέτουν μια βάση η οποία στηρίζεται σε τρεις κοχλίες, ένας για κάθε βαθμό ελευθερίας.

Ο πρώτος τρόπος είναι η αρχική προσαρμογή ενός πλαισίου πάνω στις πλευρές του ρομπότ. Το πλαίσιο αυτό έχει ένα σχήμα τύπου σταυρού. Στην συνέχεια μέσω κοχλιών η κάμερα μετατοπίζεται έτσι ώστε να κεντραριστεί ο σταυρός στο οπτικό της πεδίο. Η σωστή τοποθέτηση μπορεί να γίνει ακριβέστερη μέσω λογισμικού όρασης το οποίο μπορεί να βρρίσκει την απόσταση της τομής των ευθειών από το κέντρο της εικόνας της κάμερας.

Ο δεύτερος τρόπος είναι παρόμοιος αλλά χρησιμοποιεί lasers. Αυτή τη φορά υπάρχουν δυο πλαίσια. Αυτό που προσαρμόζεται στο ρομπότ έχει τέσσερα σημεία στις άκρες του και αυτό που προσαρμόζεται στην κάμερα φέρει τα lasers. Με την βοήθεια των κοχλιών μετακινείται η κάμερα έτσι ώστε τα σημεία πρόσπτωσης των lasers να συμπίπτουν με τα προσημειωμένα σημεία στο πλαίσιο.

### 6.3.4 Τοποθέτηση καμερών και ευαισθησία

Πέρα όμως από την δυνατότητα σωστής τοποθέτησης της κάμερας είναι καλό να υπάρχει και ένα μέτρο του σφάλματος το οποίο μπορεί να προκύψει σε περιπτώσεις που η σωστή τοποθέτηση δεν μπορεί να επιτευχθεί. Έστω ότι η κάμερα απέχει από την θεωρητικά ιδανική της θέση κατά  $dx$ ,  $dy$ ,  $dz$ . Τα διανύσματα  $u$ ,  $v$  δεν έχουν διαφορετικές συντεταγμένες σε σχέση με την περίπτωση όπου  $dx = dy = dz = 0$  και αυτό διότι τα διανύσματα αυτά αποτελούν ελεύθερα διανύσματα και όχι διανύσματα θέσης. Ωστόσο επηρεάζονται τα σημεία πέρατος των διανυσμάτων τα οποία δίνονται από τις σχέσεις 4.5 και 4.6 οι οποίες επαναλαμβάνονται παρακάτω.

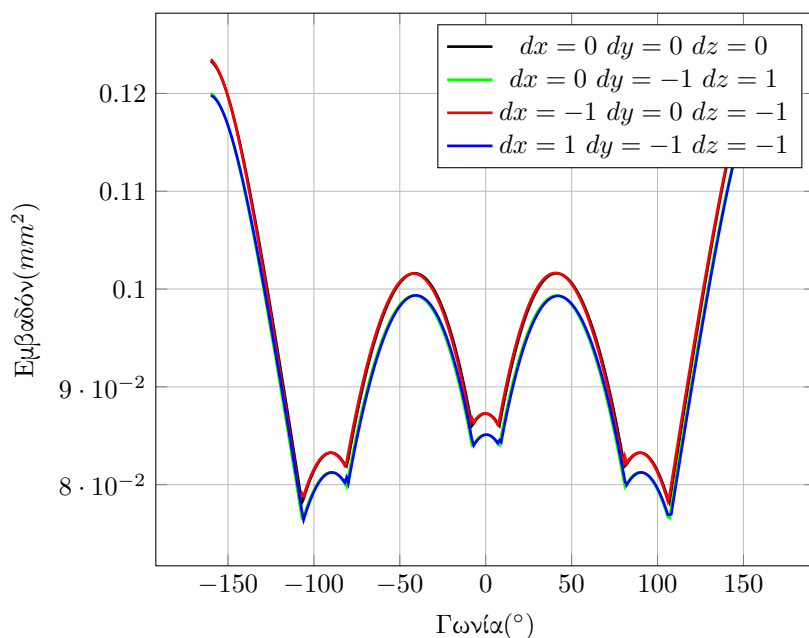
$$P_{ux} = x_u \cos(\theta) - y_u \sin(\theta) + c_x \quad P_{uy} = x_u \sin(\theta) + y_u \cos(\theta) + c_y$$

$$P_{vx} = x_v \cos(\theta) - y_v \sin(\theta) + c_x \quad P_{vy} = x_v \sin(\theta) + y_v \cos(\theta) + c_y$$

Οι όροι του κέντρου περιστροφής αλλάζουν κατά την μεταφορά του συστήματος και με την σειρά τους αλλάζουν και τις τιμές των σημείων ( $P_{ix}, P_{iy}$ ). Ιδανικά είναι αναγκαία η μελέτη των σημείων  $P_{ix}$  και των απεικονίσεων τους στον αισθητήρα της κάμερας, ωστόσο η χρήση του σχήματος του κύκλου για την οπτική παρακολούθηση έχει το πλεονέκτημα πως η περιστροφή κατά την ίδια γωνία γύρω από οποιονδήποτε άξονα δίνει ίδια προβολή. Στο παρόν μοντέλο η γωνία γύρω από τον άξονα  $z$  προκύπτει εύκολα γεωμετρικά και είναι ίση με:

$$\phi_z = \arctan\left(\frac{dz}{c_y - dy}\right) \quad (6.1)$$

Με βάση τα παραπάνω είναι πλέον εύκολη η υπολογιστική διερεύνηση των μετατοπίσεων της κάμερας στον χάρτη επίλυσης.



Σχήμα 6.16: Επίδραση των μετατοπίσεων  $dx, dy, dz$  στον χάρτη επίλυσης.

### Ερμηνεία σφαλμάτων

Από την ανάλυση σφάλματος που έγινε προκύπτει ότι η κύρια παράμετρος η οποία επηρεάζει κάπως τον χάρτη επίλυσης είναι η μετατόπιση κατά τον άξονα  $y$ . Ωστόσο πολύ σημαντικές είναι οι παρακάτω παρατηρήσεις. Η πρώτη είναι ότι οι περιοχές μεγιστοποίησης σφάλματος χωρίζονται σε δυο κατηγορίες. Η μια κατηγορία δεν χρησιμοποιείται από τις κάμερες καθώς δεν υπάρχει ορατότητα δύο χρωμάτων. Οι περιοχές αυτές είναι τα μικρά τοπικά μέγιστα που εμφανίζονται μεταξύ των ακμών της καμπύλης.

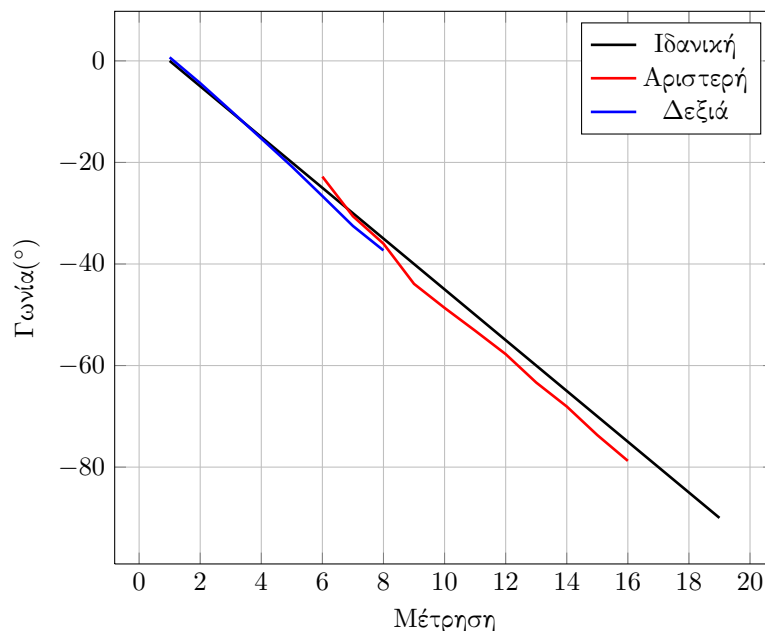
Η δεύτερη κατηγορία είναι οι τιμές εμβαδού οι οποίες βρίσκονται γύρω από τις τιμές γωνιών  $\pm 45^\circ$  και  $\pm 160^\circ$  γεγονός το οποίο αφενός εισάγει σφάλμα της τάξεως μόνο του του 4%, αφετέρου μπορεί να αντιμετωπιστεί με έναν χάρτη ο οποίος λαμβάνει υπόψη του την εσφαλμένη τοποθέτηση της κάμερας.

Ένα ακόμα ενδιαφέρον στοιχείο που προκύπτει από την ανάλυση σφάλματος είναι πως το κομμάτι του κύκλου που περιστρέφεται πιο έκκεντρα και έρχεται πιο κοντά στον φακό της κάμερας έχει αρκετά μειωμένο σφάλμα κατά την διάρκεια της διαδρομής. Αυτό συμβαίνει διότι οι μεταβολές των συνολικών ορατών pixels αποτελούν συνολικά μικρότερο ποσοστό μεταβολής επί του συνόλου όταν υπάρχει μικρή μεταβολή του  $dy$  για έναν φακό που είναι ήδη κοντά στον παρακολουθούμενο κύκλο.

## 6.4 Πειραματικός προσδιορισμός σφαλμάτων

Μια ποσοτικοποίηση σφαλμάτων πραγματοποιήθηκε και πειραματικά. Αρχικά αναζητήθηκε τρόπος εξαγωγής δεδομένων από τον ελεγκτή του ρομπότ, αλλά αυτό δεν ήταν δυνατόν. Έτσι οι μετρήσεις λήφθηκαν χειροκίνητα. Το ρομπότ τοποθετήθηκε σε γνωστές θέσεις και όντας ακίνητο λήφθηκαν 100 στιγμιότυπα από κάθε θέση έτσι ώστε να εμφανισθεί ο τυπικός θόρυβος που συναντάται κατά την πραγματική λειτουργία. Κατά την επεξεργασία των μετρήσεων ως μέτρηση λαμβάνεται ο μέσος όρος των 100 στιγμιότυπων. Οι μετρήσεις έγιναν ανά  $5^\circ$  ξεκινώντας από τις  $0^\circ$  και φτάνοντας μέχρι τις  $90^\circ$

Από τα αποτελέσματα που προκύπτουν φαίνεται πως η ακρίβεια των μετρήσεων είναι αρκετά ικανοποιητική. Με μέσο όρο αποκλίσεων από τις πραγματικές τιμές ίσο με  $1.5^\circ$  για την δεξιά κάμερα και  $3.07^\circ$  για την αριστερή κάμερα. Η απόσταση αυτή που είναι της τάξεως των περίπου  $4^\circ$  είναι και η κατακόρυφη απόσταση που εμφανίζεται στο σχήμα 6.15 οδηγώντας σε συμφωνία των λύσεων μεταξύ των δυο καμερών.



Σχήμα 6.17: Πειραματικά αποτελέσματα της μέτρησης του σφάλματος.

Ένα ακόμα σημείο που πρέπει να τονισθεί είναι πως η απόκλιση μεταξύ των

μετρήσεων δεν είναι ίδια υποδηλώνοντας πως οι δύο κάμερες δεν μετράνε με το ίδιο σφάλμα αλλά και πως η αριστερή κάμερα απέχει περισσότερο από το ιδανικό μοντέλο σε σχέση με την δεξιά κάμερα. Αυτό μπορεί να οφείλεται στον ελαφρώς διαφορετικό φωτισμό που υπάρχει μεταξύ των δυο θέσεων αλλά και στην ανακριβή μέτρηση της θέσης της που οδηγεί σε λανθασμένο μοντέλο επίλυσης.

Ένας τρόπος για την διαχείριση του ζητήματος αυτού είναι αφενός η λήψη πυκνότερων μετρήσεων (π.χ ανά  $2^\circ$  και όχι  $5^\circ$ ) εμβαδού και όχι γωνίας ώστε να υπάρχει μια πιο λεπτομερής εικόνα για την συμπεριφορά του σφάλματος και στην συνέχεια διαδοχικές επιλύσεις με χάρτες διαφορετικών παραμέτρων  $dx$ ,  $dy$ ,  $dz$  έως την ελαχιστοποίηση της γεωμετρικής απόστασης μεταξύ της πειραματικής και ιδανικής καμπύλης.

## Κεφάλαιο 7

# Συμπεράσματα

Στο κεφάλαιο αυτό γίνεται μια ανασκόπηση του προβλήματος αλλά και της μεθοδολογίας η οποία ακολουθήθηκε για την επίλυση του. Τέλος διατυπώνονται προτάσεις για μελλοντική έρευνα οι οποίες σκοπό έχουν αφενός την περαιτέρω ανάπτυξη τεχνογνωσίας γύρω από τους τομείς της όρασης και τις εφαρμογές τους στα συστήματα παραγωγής, αφετέρου την βελτίωση της απόδοσης και της ακριβείας του συστήματος ανίχνευσης το οποίο αναπτύχθηκε.

### 7.1 Συμπεράσματα

Στην παρούσα διπλωματική παρουσιάστηκε ένα σύστημα οπτικής μέτρησης θέσης περιστρεφόμενων αξόνων ρομποτικών βραχιόνων. Οι λόγοι που επιβάλλουν την δημιουργία του συστήματος είναι η εύκολη εξαγωγή και χρήση δεδομένων μέτρησης θέσης για μετρήσεις ακριβείας (όταν η ακρίβεια του συστήματος το επιτρέπει) όπως ο έλεγχος των encoders του ρομπότ για σφάλματα, ή η έξυπνη συνεργασία του ρομπότ με άλλα υποσυστήματα παραγωγής.

Η φυσική αρχή που χρησιμοποιήθηκε για την επίλυση του προβλήματος ήταν η οπτική παραμόρφωση των αντικειμένων όταν αυτά μετακινούνται στο οπτικό πεδίο της κάμερας. Έτσι τοποθετήθηκαν στόχοι πάνω στον βραχίονα οι οποίοι παρακολουθούνται. Από την παραμόρφωση των στόχων και με χρήση του αντίστοιχου μαθηματικού μοντέλου προκύπτει η τιμή της γωνίας κατά την οποία έχει περιστραφεί η άρθρωση. Ο μοναδικός περιορισμός του μοντέλου είναι η παραλληλία του άξονα περιστροφής της άρθρωσης με το επίπεδο προβολής της κάμερας.

Στην συνέχεια γίνεται μια μελέτη ευαισθησίας σφάλματος συναρτήσει της απόκλισης της θέσης της κάμερας από το επιθυμητό σημείο από το οποίο προκύπτει πως η ανίχνευση δεν είναι το ίδιο ευαίσθητη για κάθε μεταβολή καθώς μόνο η μεταβολή της θέσης στον οπτικό άξονα επηρέαζε κατά τρόπο μη αμελητέο τα αποτελέσματα του μαθηματικού μοντέλου.

Τέλος έγινε μια πειραματική μέτρηση του σφάλματος η οποία έδειξε πως τα επίπεδα του σφάλματος είναι εντός των αρχικών προδιαγραφών και λαμβάνοντας υπόψη τον χρησιμοποιούμενο εξοπλισμό και τις συνθήκες φωτισμού φαίνεται πως υπάρχουν σημαντικά περιθώρια βελτίωσης.

## 7.2 Μελλοντική έρευνα

Σε αυτή την παράγραφο προτείνονται κατευθύνσεις έρευνας. Η μια κατεύθυνση σχετίζεται με την ανάπτυξη της τεχνολογίας του τομέα κατεργασιών πάνω στους τομείς της τεχνητής όρασης. Και η δεύτερη πάνω στην εφαρμογή της τεχνολογίας αυτής για την βελτίωση του συστήματος ανίχνευσης που έχει ήδη αναπτυχθεί.

### 7.2.1 Τεχνητή όραση

Παρά την ραγδαία ανάπτυξη του τομέα της τεχνητής όρασης αλλά και των εφαρμογών του, οι μαθηματικές μέθοδοι δεν είναι αρκετά ώριμες αναδεικνύοντας την εμπειρία ως ένα σημαντικό εφόδιο των ερευνητών. Οι επόμενες προτάσεις έχουν ως σκοπό την απόκτηση εμπειρίας αλλά και των μεθόδων οι οποίες μπορούν να βοηθήσουν στην επίλυση προβλημάτων όρασης.

Ένα σημείο ενδιαφέροντος είναι η διερεύνηση της επίδρασης του φωτισμού στις εφαρμογές της όρασης, αλλά και η ανάπτυξη μοντέλων για την αντιστάθμιση των επιδράσεων του. Ένας τρόπος για να γίνει αυτό είναι η χρήση χρωμάτων αναφοράς από φωτεινές ενδείξεις όπως π.χ οθόνες και Leds(οι οποίες δεν μεταβάλλονται εύκολα) και η σύγκρισή τους με τις μεταβαλλόμενες τιμές των pixel στην εικόνα. Στο ίδιο πεδίο ανήκει και η διερεύνηση των παραμέτρων του εξοπλισμού όπως η φωτεινότητα και η αντίθεση.

Τέλος μια ακόμα εφαρμογή η οποία έχει πρακτικό ενδιαφέρον είναι η έξυπνη αναγνώριση σχημάτων. Με τον χαρακτηρισμό έξυπνη εννοείται η ανάπτυξη μεθόδων οι οποίες θα μπορούσαν όχι μόνο να αναγνωρίζουν απλά γεωμετρικά σχήματα αλλά και συνδυασμούς τέτοιων σχημάτων. Αυτό μπορεί να επιτευχθεί με χρήση γεωμετρικών μοντέλων και εύρεση των ομογενών μετασχηματισμών για την εύρεση του τελικού προσανατολισμού. Η πρόκληση εδώ έγκειται στην σωστή εξαγωγή του σχήματος προς επίλυση.

### 7.2.2 Βελτίωση συστήματος ανίχνευσης

Για την βελτίωση του συστήματος ανίχνευσης οι κατευθύνσεις είναι η αύξηση της ακρίβειας, η αύξηση της απόδοσης και η προσθήκη νέων χαρακτηριστικών. Ένα από τα βασικά χαρακτηριστικά του συστήματος είναι η εύρεση της θέσης των αρθρώσεων στον χώρο. Ο τρόπος που θα μπορούσε να επιτευχθεί αυτό είναι η εύρεση των γωνιών των συνδέσμων και στην συνέχεια η επίλυση της ευθείας κινηματικής του βραχίονα. Οι σύνδεσμοι θα μπορούσαν να ανιχνευθούν με επικάλυψη ορατών ευθύγραμμων τμημάτων και εφαρμογή μετασχηματισμών hough.

Ένα δεύτερο σημείο βελτιώσεως είναι η χρήση συνολικά 8 καμερών μια ανά 45° και υλοποίηση του συστήματος σε πλήρη κλίμακα. Στην συνέχεια στο σύστημα αυτό θα είχε ενδιαφέρον να αναπτυχθεί μια μέθοδος διαχείρισης των μετρήσεων η οποία να διαλέγει με βέλτιστο τρόπο την κάμερα της οποίας την μέτρηση θα λάβει υπόψη. Προφανώς το κριτήριο αυτό θα βασίζεται στο σφάλμα της μέτρησης.

Η ανάπτυξη ενός μοντέλου εκτίμησης αλλά και διόρθωσης σφάλματος είναι ένα χαρακτηριστικό μεγάλης σημασίας. Ανάλογα με το ορατό εμβαδόν αλλά και την θέση της κάμερας θα μπορούσαν να δημιουργηθούν καμπύλες αντιστάθμισης οι οποίες θα διόρθωναν την μέτρηση. Πρόσθετο ενδιαφέρον θα είχε η ανάπτυξη μιας κλειστής υπολογιστικής μεθόδου η οποία θα έβρισκε τις μετατοπίσεις  $dx$ ,  $dy$ ,  $dz$  της κάμερας για δεδομένες γωνίες και μετρήσεις.



Ταυτόχρονα για την μεγαλύτερη αυτοματοποίηση της διάταξης κρίνεται αναγκαία η ανάπτυξη αλγορίθμων εύρεσης των παραμέτρων όπως τις διαστάσεις της περιοχής ενδιαφέροντος, τα μέγιστα pixels ανά χρώμα αλλά και το εύρος των συγκρινόμενων χρωμάτων που χρησιμοποιούνται για την ανίχνευση.

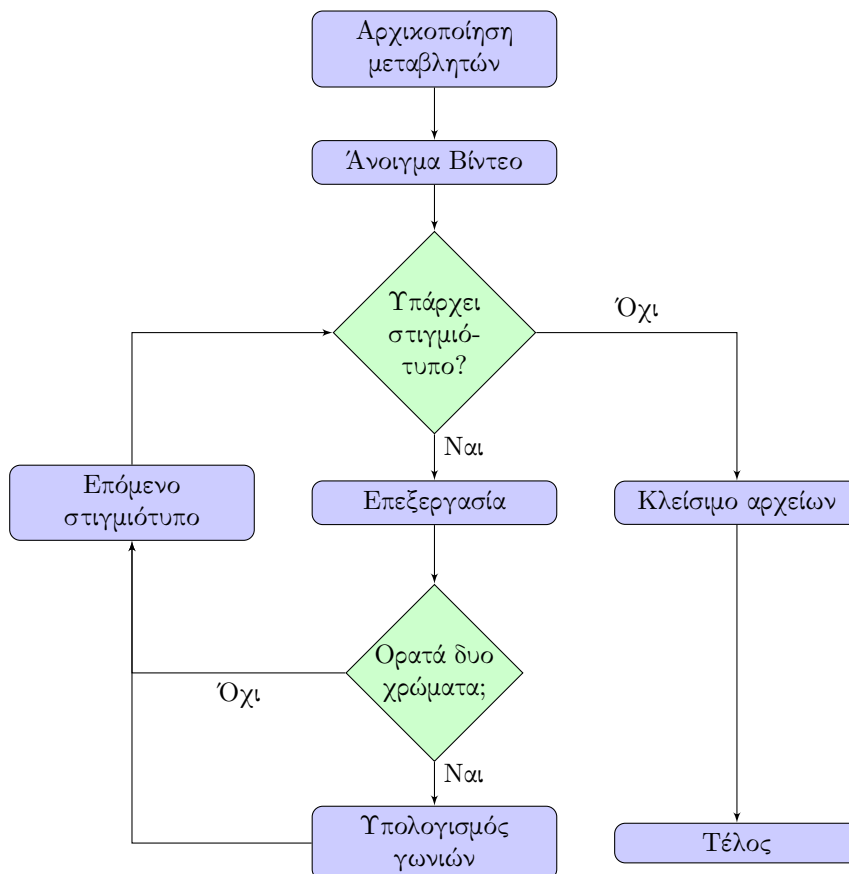
Επιπρόσθετα ενδιαφέρον θα είχε και η ανάπτυξη διαδικασιών αλλά και εργαλείων όπως για παράδειγμα αυτορυθμιζόμενες βάσεις για τις κάμερες οι οποίες θα επέτρεπαν την σωστή τοποθέτηση των καμερών χωρίς την ύπαρξη σοβαρού σφάλματος θέσης.

Τέλος ένα βήμα το οποίο είναι συγγενικό με την πρώτη πρόταση είναι η ανίχνευση όλων των αρθρώσεων στον χώρο όχι μέσω μετασχηματισμών Hough, αλλά με απευθείας μέτρηση των αρθρώσεων. Η μεθοδολογία εύρεσης θα μπορούσε να είναι η ίδια που ακολουθήθηκε στην παρούσα διπλωματική με την διαφορά ότι η επιλογή της περιοχής των καμπυλών γωνίας-εμβαδού θα γίνεται με χρήση τριών χρωμάτων και όχι δυο και με τη διαφορά πως ο χάρτης γωνίας-εμβαδού παράγεται πλέον με περιστροφές γύρω από δυο άξονες και όχι από έναν.

# Παράρτημα Α΄

## Κώδικες

Στο παρακάτω παράρτημα δίνεται το διάγραμμα ροής της επίλυσης που έγινε καθώς και οι αντίστοιχοι κώδικες. Για την μεταγλώττση τους είναι απαραίτητο να έχει εγκατασταθεί πρώτα η βιβλιοθήκη του OpenCV.



Σχήμα Α΄.1: Διάγραμμα ροής της επίλυσης

Κώδικας Α'1: Η επικεφαλίδα common.h

---

```
1 #ifndef COMMON_H
2 #define COMMON_H
3
4 #include<stdlib.h>
5
6 extern double vectors_r[2][2];
7 extern double vectors_g[2][2];
8 extern double vectors_b[2][2];
9 extern double vectors_c[2][2];
10
11 double degrees2Radians(double degrees);
12 double radians2degrees(double radians);
13
14 void initVectors();
15
16 double radius(int bgrc, double angle);
17 double color_area(int bgrc, double angle);
18 double max_area(int bgrc);
19 double area(double angle);
20
21 #endif // COMMON_H
```

---

Κώδικας Α'2: Η υλοποίηση common.cpp

---

```
1 #include "common.h"
2 #include <cmath>
3
4 double vectors_r[2][2] = {{0}};
5 double vectors_g[2][2] = {{0}};
6 double vectors_b[2][2] = {{0}};
7 double vectors_c[2][2] = {{0}};
8
9
10 double degrees2Radians(double degrees){
11     return degrees*M_PI/180.0;
12 }
13 double radians2degrees(double radians){
14     return radians*180.0/M_PI;
15 }
16
17 void initVectors(){
18
19     //Vectors of red circle leftmost one
20     vectors_r[0][0] = -13.1*10;
21     vectors_r[0][1] = 4.7*10;
22
23     vectors_r[1][0] = -13.1*10;
24     vectors_r[1][1] = -4.7*10;
25
26     //Vectors of green circle center one
27     vectors_g[0][0] = -4.7*10;
28     vectors_g[0][1] = -15*10;
29 }
```

```

30     vectors_g[1][0] = 4.7*10;
31     vectors_g[1][1] = -15*10;
32
33     //Vectors of blue circle rightmost one
34     vectors_b[0][0] = 13.1*10;
35     vectors_b[0][1] = -4.7*10;
36
37     vectors_b[1][0] = 13.1*10;
38     vectors_b[1][1] = 4.7*10;
39
40     //Vectors of blue cyan upper one
41     vectors_c[0][0] = 4.7*10;
42     vectors_c[0][1] = 27.55*10;
43
44     vectors_c[1][0] = -4.7*10;
45     vectors_c[1][1] = 27.55*10;
46 }
47
48 //Returns the radius(theta) of a given color
49 //Angle:In Degrees.
50 //Positive is the counter clockwise direction
51 double radius(int bgrc, double angle){
52
53     double points_r[2][2] = {{0}};
54     double points_g[2][2] = {{0}};
55     double points_b[2][2] = {{0}};
56     double points_c[2][2] = {{0}};
57
58     double center_x = 0*10;
59     double center_y = 94.8*10;
60
61     //Convert angle for use in trigonometric functions
62     angle = degrees2Radians(angle);
63
64     //Rotation matrix with counter clockwise rotation
65     for(int i=0; i<2; i++){
66         points_r[i][0] = vectors_r[i][0]*cos(angle) - vectors_r[i][1]*sin(
        angle) + center_x;
67         points_r[i][1] = vectors_r[i][0]*sin(angle) + vectors_r[i][1]*cos(
        angle) + center_y;
68
69         points_g[i][0] = vectors_g[i][0]*cos(angle) - vectors_g[i][1]*sin(
        angle) + center_x;
70         points_g[i][1] = vectors_g[i][0]*sin(angle) + vectors_g[i][1]*cos(
        angle) + center_y;
71
72         points_b[i][0] = vectors_b[i][0]*cos(angle) - vectors_b[i][1]*sin(
        angle) + center_x;
73         points_b[i][1] = vectors_b[i][0]*sin(angle) + vectors_b[i][1]*cos(
        angle) + center_y;
74
75         points_c[i][0] = vectors_c[i][0]*cos(angle) - vectors_c[i][1]*sin(
        angle) + center_x;

```

```

76     points_c[i][1] = vectors_c[i][0]*sin(angle) + vectors_c[i][1]*cos(
77         angle) + center_y;
78 }
79 double angles_r[2] = {0};
80 double angles_g[2] = {0};
81 double angles_b[2] = {0};
82 double angles_c[2] = {0};
83
84 double focalDistances_r[2] = {0};
85 double focalDistances_g[2] = {0};
86 double focalDistances_b[2] = {0};
87 double focalDistances_c[2] = {0};
88
89 //Find distances and solve for sin(angle) for the focal point
90 for(int i=0; i<2; i++){
91     angles_r[i] = atan2(points_r[i][1],points_r[i][0]);
92     angles_g[i] = atan2(points_g[i][1],points_g[i][0]);
93     angles_b[i] = atan2(points_b[i][1],points_b[i][0]);
94     angles_c[i] = atan2(points_c[i][1],points_c[i][0]);
95
96     focalDistances_r[i] = tan(M_PI/2.0 - angles_r[i])*0.283*10;
97     focalDistances_g[i] = tan(M_PI/2.0 - angles_g[i])*0.283*10;
98     focalDistances_b[i] = tan(M_PI/2.0 - angles_b[i])*0.283*10;
99     focalDistances_c[i] = tan(M_PI/2.0 - angles_c[i])*0.283*10;
100 }
101
102 double radius_r = 0;
103 double radius_g = 0;
104 double radius_b = 0;
105 double radius_c = 0;
106
107 radius_r = sqrt(pow(focalDistances_r[1] - focalDistances_r[0],2))
108     /2.0;
109 radius_g = sqrt(pow(focalDistances_g[1] - focalDistances_g[0],2))
110     /2.0;
111 radius_b = sqrt(pow(focalDistances_b[1] - focalDistances_b[0],2))
112     /2.0;
113 radius_c = sqrt(pow(focalDistances_c[1] - focalDistances_c[0],2))
114     /2.0;
115
116 if(bgrc == 0)
117     return radius_b;
118 if(bgrc == 1)
119     return radius_g;
120 if(bgrc == 2)
121     return radius_r;
122 if(bgrc == 3)
123     return radius_c;
124 return -1;
125 }
126
127 double area(double angle){

```

```

124     return color_area(0,angle)+color_area(1,angle)+color_area(2,angle)+
        color_area(3,angle);
125 }
126
127 double color_area(int bgrc, double angle){
128     double area_b;
129     double area_g;
130     double area_r;
131     double area_c;
132
133     bool visible_b = false;
134     bool visible_g = false;
135     bool visible_r = false;
136     bool visible_c = false;
137
138     //Max radius for blue circle is for -90 degrees
139     area_b = M_PI*radius(0,angle)*radius(0,-90);
140
141     //Max radius for green circle is for 0 degrees
142     area_g = M_PI*radius(1,angle)*radius(1,0);
143
144     //Max radius for red circle is for 90 degrees
145     area_r = M_PI*radius(2,angle)*radius(2,90);
146
147     //Max radius for cyan circle is for 180 degrees
148     area_c = M_PI*radius(3,angle)*radius(3,180);
149
150     double checkAngle = angle;
151     //Convert angle to CounterClockwise for visibility check
152     if(angle < 0)
153         checkAngle = angle + 360;
154
155     //Do the visibility check
156     if(checkAngle >= 8 && checkAngle <= 172)
157         visible_r = true;
158     if((checkAngle >= 0 && checkAngle <= 81) || (checkAngle>=279 &&
        checkAngle<=360))
159         visible_g = true;
160     if(checkAngle >= 188 && checkAngle <= 352)
161         visible_b = true;
162     if(checkAngle >= 107 && checkAngle <= 253)
163         visible_c = true;
164
165     if(!visible_r)
166         area_r = 0;
167     if(!visible_g)
168         area_g = 0;
169     if(!visible_b)
170         area_b = 0;
171     if(!visible_c)
172         area_c = 0;
173
174     if(bgrc == 0)
175         return area_b;

```

```

176     if(bgrc == 1)
177         return area_g;
178     if(bgrc == 2)
179         return area_r;
180     if(bgrc == 3)
181         return area_c;
182 }
183
184 double max_area(int bgrc){
185     double area_b;
186     double area_g;
187     double area_r;
188     double area_c;
189
190     //Max radius for blue circle is for -90 degrees
191     area_b = M_PI*radius(0,-90)*radius(0,-90);
192
193     //Max radius for green circle is for 0 degrees
194     area_g = M_PI*radius(1,0)*radius(1,0);
195
196     //Max radius for red circle is for 90 degrees
197     area_r = M_PI*radius(2,90)*radius(2,90);
198
199     //Max radius for cyan circle is for 180 degrees
200     area_c = M_PI*radius(3,180)*radius(3,180);
201
202     if(bgrc == 0)
203         return area_b;
204     if(bgrc == 1)
205         return area_g;
206     if(bgrc == 2)
207         return area_r;
208     if(bgrc == 3)
209         return area_c;
210 }

```

---

Κώδικας Α'.3: Η επικεφαλίδα jointrotationmap.h

---

```

1  #ifndef JOINTROTATIONMAP_H
2  #define JOINTROTATIONMAP_H
3
4  #include <iostream>
5  #include <cmath>
6  #include "common.h"
7
8
9  class JointRotationMap
10 {
11 public:
12     JointRotationMap(double, double);
13     void changeSegment(double minAngle, double maxAngle);
14     double solveAngle(double area, int bgrc);
15 public:
16     double segMaxAngle;

```

```

17     double segMinAngle;
18     int currentSegment;
19 };
20
21 #endif // JOINTROTATIONMAP_H

```

---

Κώδικας Α'.4: Η υλοποίηση jointrotationmap.cpp

---

```

1 #include "jointrotationmap.h"
2
3 JointRotationMap::JointRotationMap(double minAngle,double maxAngle)
4 {
5     //Set the solution limits
6     changeSegment(minAngle,maxAngle);
7 }
8
9 void JointRotationMap::changeSegment(double minAngle,double maxAngle){
10     segMinAngle = minAngle;
11     segMaxAngle = maxAngle;
12     return;
13 }
14
15 //Implementation of Bisection method
16 double JointRotationMap::solveAngle(double area,int bgrc){
17
18     double root;
19     double funcUpperBound = degrees2Radians(segMaxAngle);
20     double funcLowerBound = degrees2Radians(segMinAngle);
21     //Starting value
22     double rootOld=funcLowerBound;
23     double term;
24
25     for(int i=0; i<100; i++){
26         root = (funcUpperBound + funcLowerBound)/2.0;
27         term = (color_area(bgrc,radians2degrees(root))-area)*(color_area(
28             bgrc,radians2degrees(funcLowerBound))-area);
29         if(fabs(root - rootOld) <= 0.001){
30             return radians2degrees(root);
31         }
32         if(term < 0){
33             rootOld = root;
34             funcUpperBound = root;
35             continue;
36         }
37         if(term > 0){
38             rootOld = root;
39             funcLowerBound = root;
40             continue;
41         }
42     }
43     return -1;
44 }

```

---



```
1 #include <iostream>
2 #include <fstream>
3 #include <vector>
4 #include <cmath>
5
6 #include <opencv2/opencv.hpp>
7
8 #include "common.h"
9 #include "jointrotationmap.h"
10
11 //Function prototyping
12 double lcalcArea(int,int);
13 double rcalcArea(int,int);
14 cv::Mat lframePreprocess(cv::Mat);
15 cv::Mat rframePreprocess(cv::Mat);
16 cv::Mat findColors(cv::Mat);
17 std::vector<cv::Mat> framePostprocess(cv::Mat);
18
19 using namespace std;
20
21 int main()
22 {
23     cout << "Staubli Robotics Rotation Detection with Computer Vision" <<
24         endl;
25     cout << "Author: Aris Karagiannidis" << endl;
26
27     initVectors();
28
29     cv::VideoCapture lcam,rcam;
30     cv::Mat lframe,rframe;
31
32     //Video files to open
33     lcam.open("lfull.avi");
34     rcam.open("rfull.avi");
35
36     //Check for errors in opening
37     if(!lcam.isOpened() || !rcam.isOpened()){
38         cout<< "Cannot open capture device" << endl;
39         cout<<"Press any key to exit" <<endl;
40         cin.ignore(1);
41         return 0;
42     }
43
44     //We assume here that both videos have the same frame count
45     int lframeCount = lcam.get(CV_CAP_PROP_FRAME_COUNT);
46
47     //Vector of decomposed channels
48     std::vector<cv::Mat> lbgr,rbgr;
49
50     //Datafiles for left and right camera
51     ofstream ldata;
52     ofstream rdata;
```

```

53 //Open datafiles and write the headers
54 ldata.open("ldata.dat");
55 ldata << "#Frame || BPixels || GPixels || BArea || GArea || BAngle ||
        GAngle || AngMean" <<endl;
56
57 rdata.open("rdata.dat");
58 rdata << "#Frame || BPixels || GPixels || BArea || GArea || BAngle ||
        GAngle || AngMean" <<endl;
59
60 //Declare the map and solution range
61 JointRotationMap map(-90,0);
62
63 //Start processing
64 for(int counter =0; counter<lframeCount; counter++){
65     lcam >> lframe;
66     rcam >> rframe;
67
68     lframe = lframePreprocess(lframe);
69     rframe = rframePreprocess(rframe);
70
71     lframe = findColors(lframe);
72     rframe = findColors(rframe);
73
74     //Display colored images
75     cv::imshow("Left Colors",lframe);
76     cv::imshow("Right Colors",rframe);
77
78     //Populate vector with slit images
79     lbgr = framePostprocess(lframe,0);
80     rbgr = framePostprocess(rframe,1);
81
82     //Display the images
83     cv::imshow("lBlue",lbgr.at(0));
84     cv::imshow("lGreen",lbgr.at(1));
85
86     cv::imshow("rBlue",rbgr.at(0));
87     cv::imshow("rGreen",rbgr.at(1));
88
89     //Count the colored pixels for each channel
90     int bpixel = cv::countNonZero(lbgr.at(0));
91     int gpixel = cv::countNonZero(lbgr.at(1));
92
93     //Define reusable variables for the
94     //blue and green areas and angles
95     double barea = 0;
96     double garea = 0;
97
98     double bangle = 0;
99     double gangle = 0;
100
101     //Left camera calculations if is visible
102     if(bpixel >=500 && gpixel >=500){
103         barea = lcalcArea(bpixel,0);
104         garea = lcalcArea(gpixel,1);

```

```

105
106     bangle = map.solveAngle(barea,0);
107     gangle = map.solveAngle(garea,1);
108
109     //offset the result, typical range -10 to -5
110     bangle = bangle -8.0;
111     gangle = gangle -8.0;
112
113     //Write the left data
114     //      1      2      3      4      5      6      7
115     //<< "#Frame || BPixels || GPixels || BArea || GArea || BAngle
        || GAngle"
116     ldata << counter << " " << bpixel << " " << gpixel << " " <<
        barea << " " << garea << " "
117         << bangle << " " << gangle << " " << (bangle+gangle)/2.0
        << std::endl;
118 }
119
120 //Perform the same calculations for the right camera
121 bpixel = cv::countNonZero(rbgr.at(0));
122 gpixel = cv::countNonZero(rbgr.at(1));
123
124 if(bpixel >=500 && gpixel >=500){
125     barea = rcalcArea(bpixel,0);
126     garea = rcalcArea(gpixel,1);
127
128     bangle = map.solveAngle(barea,0);
129     gangle = map.solveAngle(garea,1);
130
131     //offset the result, typical range +38 to +42
132     bangle = bangle +39.0;
133     gangle = gangle +39.0;
134
135     //Write the left data
136     //      1      2      3      4      5      6      7
137     //<< "#Frame || BPixels || GPixels || BArea || GArea || BAngle
        || GAngle"
138     rdata << counter << " " << bpixel << " " << gpixel << " " <<
        barea << " " << garea << " "
139         << bangle << " " << gangle << " " << (bangle+gangle)/2.0
        << std::endl;
140 }
141
142 //25 msec depending on the frame rate
143 char c = cv::waitKey(25); //50
144 if(c == 27){ //27 == esc key
145     cout <<"Esc or key pressed, returning" << endl;
146     break;
147 }
148 //Structure to pause on enter
149 if(c == 13){
150     //If the next key i press is not esc
151     //don't do anything.
152     char k = cv::waitKey();

```

```

153         if(k == 27){
154             cout <<"Esc or Enter key pressed, returning" << endl;
155             lcam.release();
156             rcam.release();
157             return 0;
158         }
159     }
160 }
161 ldata.close();
162 rdata.close();
163 return 0;
164 }
165
166 cv::Mat lframePreprocess(cv::Mat frame){
167     //Rect args(int x, int y, int width, int height)
168     cv::Rect roi(216,225,238,137);
169     cv::Mat roi_img = frame(roi);
170     cv::GaussianBlur(roi_img,roi_img,cv::Size(5,5),5,5);
171     return roi_img;
172 }
173
174 cv::Mat rframePreprocess(cv::Mat frame){
175     //Rect args(int x, int y, int width, int height)
176     cv::Rect roi(200,237,238,137);
177     cv::Mat roi_img = frame(roi);
178     cv::GaussianBlur(roi_img,roi_img,cv::Size(5,5),5,5);
179     return roi_img;
180 }
181
182 cv::Mat findColors(cv::Mat frame){
183     int i,j,b,g,r;
184     for(i=0; i<frame.rows; ++i){
185         for(j=0; j<frame.cols; ++j){
186             //Get pixel values
187             uchar* pixel = &((uchar*)(frame.data + frame.step*i))[j*3];
188             b = pixel[0];
189             g = pixel[1];
190             r = pixel[2];
191             if(g<121 && g>20 && r<80 && b<100 && g > b)
192                 pixel[1]=255;
193             //Refresh Pixel Values to rule out overlapping detections
194             b = pixel[0];
195             g = pixel[1];
196             r = pixel[2];
197             if(g<120 && r<60 && b<137 && b>38 && b > g)
198                 pixel[0]=255;
199         }
200     }
201     return frame;
202 }
203
204 std::vector<cv::Mat> framePostprocess(cv::Mat frame){
205     //Temporary vector
206     std::vector<cv::Mat> bgr;

```

```

207
208 //Split the channels
209 cv::split(frame,bgr);
210
211 //Threshold the channels
212 cv::threshold(bgr.at(0),bgr.at(0),245,255,CV_THRESH_BINARY);
213 cv::threshold(bgr.at(1),bgr.at(1),245,255,CV_THRESH_BINARY);
214
215 //Errode the channels the channels
216 cv::Mat element = cv::getStructuringElement(cv::MORPH_RECT,cv::Size
      (2,2),cv::Point(0,0));
217 cv::erode(bgr.at(0),bgr.at(0),element);
218 cv::erode(bgr.at(1),bgr.at(1),element);
219 return bgr;
220 }
221
222 double lcalcArea(int pixelNu,int bgrc){
223     double area;
224     if(bgrc == 0)
225         area = ((double)pixelNu * max_area(bgrc))/4700.0;
226     else
227         area = ((double)pixelNu * max_area(bgrc))/4000.0;
228     return area;
229 }
230
231 //Calls lcalcarea exists in case of fine tuning
232 double rcalcArea(int pixelNu,int bgrc){
233     return lcalcArea(pixelNu,bgrc);
234 }

```

---

Κώδικας Α'.6: Η υλοποίηση του εργαλείου videoCapture.cpp

---

```

1 #include <iostream>
2 #include <sstream>
3 #include<opencv2/opencv.hpp>
4
5
6 using namespace std;
7
8 int main()
9 {
10     cv::Mat frame1,frame2;
11
12     cv::VideoCapture camera1,camera2;
13     cv::VideoWriter recorder1,recorder2;
14
15     //1 Continuous mode recording
16     //2 Parts mode recording
17     int mode=0;
18
19     cout << "Welcome to video capture!" << endl;
20     cout << "Author: Aris Karagiannidis" << endl;
21
22     //Force correct input

```

```

23 while(1){
24     cout << "Press 1 for recording mode or 2 for measurement mode" <<
        endl;
25     cin >> mode;
26     if(mode==1 || mode==2)
27         break;
28     else
29         cout << "Invalid mode" <<endl;
30 }
31
32
33 //Camera index does not link with left
34 //Or right camera
35 //operating system changes indexes arbitrarily
36 camera1.open(1);
37 camera2.open(0);
38
39 cv::namedWindow("Video Cam 1");
40 cv::namedWindow("Video Cam 2");
41
42 //Initialize some frames just to
43 //be on the safe side
44 camera1 >> frame1;
45 camera1 >> frame1;
46
47 camera2 >> frame2;
48 camera2 >> frame2;
49
50 //Continuous Video recording is at 18 fps.
51 if(mode==1){
52     //Open files to write data
53     recorder1.open("cam1-out.avi",CV_FOURCC('D','I','V','3'),18,frame1
        .size(),true);
54     recorder2.open("cam2-out.avi",CV_FOURCC('D','I','V','3'),18,frame2
        .size(),true);
55
56     //Never stop until i tell you to
57     while(1){
58         camera1 >> frame1;
59         camera2 >> frame2;
60
61         recorder1 << frame1;
62         recorder2 << frame2;
63
64         cv::imshow("Video Cam 1",frame1);
65         cv::imshow("Video Cam 2",frame2);
66         char c = cv::waitKey(33); //18 FPS but works for 18
67         //Press enter or esc key
68         if(c==13 || c==27){
69             cout<<"Enter or esc key pressed. Recording stoped." <<endl;
70             //Cleanup before closing
71             camera1.release();
72             camera2.release();
73             recorder1.release();

```

```

74         recorder2.release();
75         return 0;
76     }
77 }
78 }
79
80 //Parts mode recording
81 if(mode==2){
82     int exitFlag=0;
83     cout<<"Begin measurement from 0 to 90 deg." <<endl;
84
85     //Recorder must be released before enter the mode
86     //Just to be safe
87     recorder1.release();
88     recorder2.release();
89
90     //Measure from degree 0 to 90 interval 5 deg
91     for(int i=0; i<=90; i+=5){
92         //Form a dynamic filename
93         stringstream filename1;
94         stringstream filename2;
95
96         filename1.str("");
97         filename2.str("");
98
99         filename1 <<i<<"cam1"<<".avi";
100        filename2 <<i<<"cam2"<<".avi";
101
102        //Open the newly formed filename
103        recorder1.open(filename1.str(),CV_FOURCC('D','I','V','3'),18,
104            frame1.size(),true);
105        recorder2.open(filename2.str(),CV_FOURCC('D','I','V','3'),18,
106            frame2.size(),true);
107
108        cout<<"Measuring for "<<i<<" degrees press c to continue"<<
109            endl;
110        //Dummy char to wait before we continue
111        char l;
112        cin >> lol;
113
114        //Actual measurement
115        for(int j=0; j<100; j++){
116            camera1 >> frame1;
117            camera2 >> frame2;
118            recorder1 << frame1;
119            recorder2 << frame2;
120            cv::imshow("Video Cam 1",frame1);
121            cv::imshow("Video Cam 2",frame2);
122            //Wait here to make the fps right
123            char c = cv::waitKey(33);
124        }
125        //Release recorder to make new measurement
126        recorder1.release();
127        recorder2.release();

```

```
125
126     //Have an option to stop midway
127     cout<<"Press 1 to continue or 0 to exit"<<endl;
128     cin>>exitFlag;
129     if(exitFlag==0){
130         cout<<"Stopping Measurements"<<endl;
131         camera1.release();
132         camera2.release();
133         recorder1.release();
134         recorder2.release();
135         return 0;
136     }
137 }
138
139 //Cleanup
140 cout<<"Measurements Ended"<<endl;
141 camera1.release();
142 camera2.release();
143 recorder1.release();
144 recorder2.release();
145 return 0;
146 }
147
148
149 return 0;
150 }
```

---



# Βιβλιογραφία

- [1] C.-Y. Tsai and K.-T. Song, "Dynamic visual tracking control of a mobile robot with image noise and occlusion robustness," *Image and Vision Computing*, vol. 27, no. 8, pp. 1007 -- 1022, 2009.
- [2] T. P. Sim, G. S. Hong, and K. B. Lim, "A pragmatic 3d visual servoing system," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation, ICRA 2002, May 11-15, 2002, Washington, DC, USA*, pp. 4185--4190, IEEE, 2002.
- [3] O. Lang and A. Graser, "Visual control of 6 dof robots with constant object size in the image by means of zoom camera," in *Industrial Electronics Society, 1999. IECON '99 Proceedings. The 25th Annual Conference of the IEEE*, vol. 3, pp. 1342 --1347 vol.3, 1999.
- [4] H. Wang, Y.-H. Liu, and W. Chen, "Visual tracking of robots in uncalibrated environments," *Mechatronics*, vol. 22, no. 4, pp. 390 -- 397, 2012.
- [5] J. Tyrrell, J. LaPre, C. Carothers, B. Roysam, and C. Stewart, "Efficient migration of complex off-line computer vision software to real-time system implementation on generic computer hardware," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, pp. 142 --153, june 2004.
- [6] G. Healey and T. Binford, "A color metric for computer vision," in *Computer Vision and Pattern Recognition, 1988. Proceedings CVPR '88., Computer Society Conference on*, pp. 10 --17, jun 1988.
- [7] J. Fung and S. Mann, "Using multiple graphics cards as a general purpose parallel computer: applications to computer vision," in *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, vol. 1, pp. 805 -- 808 Vol.1, aug. 2004.
- [8] J. Fung, F. Tang, and S. Mann, "Mediated reality using computer graphics hardware for computer vision," in *Wearable Computers, 2002. (ISWC 2002). Proceedings. Sixth International Symposium on*, pp. 83 -- 89, 2002.
- [9] S. Zhao, B. Li, J. Yuan, and G. Cui, "Research on remote meter automatic reading based on computer vision," in *Transmission and Distribution Conference and Exhibition: Asia and Pacific, 2005 IEEE/PES*, pp. 1 --4, 2005.

- [10] H.-S. Kim, E. Naranbaatar, S. Ahammad, Y.-H. Bae, and B.-R. Lee, "Real-time forming error inspection system using computer vision for small-sized tubes," in *ICCAS-SICE, 2009*, pp. 1950 --1955, aug. 2009.
- [11] J. Liu and T. Cui, "A measurement method on the roundness error based on computer vision system," in *Information Science and Engineering (ICISE), 2010 2nd International Conference on*, pp. 884 --887, dec. 2010.
- [12] W.-G. Wang, Z.-Y. Hu, and Z.-Y. Shun, "3d shape measurement based on computer vision," in *Machine Learning and Cybernetics, 2002. Proceedings. 2002 International Conference on*, vol. 2, pp. 910 -- 914 vol.2, 2002.
- [13] M. Laghari and F. Ahmed, "Computer vision in the field of wear particles," in *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, vol. 1, pp. 126 --129, dec. 2009.
- [14] D. Capel and A. Zisserman, "Computer vision applied to super resolution," *Signal Processing Magazine, IEEE*, vol. 20, pp. 75 -- 86, may 2003.
- [15] C. Moon and E. McVey, "Precision measurement techniques using computer vision," in *Industry Applications Society Annual Meeting, 1990., Conference Record of the 1990 IEEE*, pp. 1521 --1526 vol.2, oct. 1990.
- [16] E. Gadelmawla, "Computer vision algorithms for measurement and inspection of spur gears," *Measurement*, vol. 44, no. 9, pp. 1669 -- 1678, 2011.
- [17] C.-T. Su, C. Chang, and F.-C. Tien, "Neural networks for precise measurement in computer vision systems," *Computers in Industry*, vol. 27, no. 3, pp. 225 -- 236, 1995.
- [18] T. Doiron, "Computer vision based station for tool setting and tool form measurement," *Precision Engineering*, vol. 11, no. 4, pp. 231 -- 238, 1989.
- [19] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*. Prentice Hall Professional Technical Reference, 2002.
- [20] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2nd ed., 2001.
- [21] W. Burger and M. Burge, *Principles of Digital Image Processing: Fundamental Techniques*. Undergraduate Topics in Computer Science, Springer, 2009.
- [22] J. Russ, *The Image Processing Handbook*. Taylor & Francis, 2011.
- [23] B. Siciliano, L. Sciavicco, L. Villani, and G. Oriolo, *Robotics: Modelling, Planning and Control*. Advanced Textbooks in Control and Signal Processing, Springer, 2011.
- [24] J. Craig, *Introduction to robotics: mechanics and control*. Addison-Wesley series in electrical and computer engineering: control engineering, Pearson/Prentice Hall, 2005.

- [25] G. Bradski and A. Kaehler, *Learning OpenCV*. O'Reilly Media Inc., 2008.
- [26] H. Deitel and P. Deitel, *C+: How to Program*. How to program series, Pearson Education, 2008.
- [27] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery, *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, 2007.