



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΥΠΟΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ  
ΤΕΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

# 3D Fiducial Marker Configurations for Optimal Pose Estimation

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΒΟΥΤΥΡΑΚΟΥ ΔΙΑΛΕΚΤΗΣ ΑΘΗΝΑΣ

**Επιβλέπων:** Δημήτριος Κουτσούρης  
Καθηγητής Ε.Μ.Π  
**Συνεπιβλέπουσα:** Ουρανία Πετροπούλου  
Ειδικό Τεχνικό και Εργαστηριακό Προσωπικό

Αθήνα, Αύγουστος 2020





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟ-  
ΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕ-  
ΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

## 3D Fiducial Marker Configurations for Optimal Pose Estimation

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

της

ΒΟΥΤΥΡΑΚΟΥ ΔΙΑΛΕΚΤΗΣ ΑΘΗΝΑΣ

**Επιβλέπων:** Δημήτριος Κουτσούρης  
Καθηγητής Ε.Μ.Π  
**Συνεπιβλέπουσα:** Ουρανία Πετροπούλου  
Ειδικό Τεχνικό και Εργαστηριακό Προσωπικό

Εγκρίθηκε από την τριμελή εξεταστική επιτροπή την 25<sup>η</sup> Αυγούστου 2020.

(Υπογραφή)

(Υπογραφή)

(Υπογραφή)

.....  
Δημήτριος Κουτσούρης  
Καθηγητής Ε.Μ.Π

.....  
Γεώργιος Ματσόπουλος  
Καθηγητής Ε.Μ.Π

.....  
Παναγιώτης Τσανάκας  
Καθηγητής Ε.Μ.Π

Αθήνα, Αύγουστος 2020





ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟ-  
ΛΟΓΙΣΤΩΝ  
ΤΟΜΕΑΣ ΣΥΣΤΗΜΑΤΩΝ ΜΕΤΑΔΟΣΗΣ ΠΛΗΡΟΦΟΡΙΑΣ ΚΑΙ ΤΕ-  
ΧΝΟΛΟΓΙΑΣ ΥΛΙΚΩΝ

.....

Βουτυράκου Διαλεκτή Αθηνά

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

**Copyright © Βουτυράκου Διαλεκτή Αθηνά, 2020.**

**Με την επιφύλαξη παντός δικαιώματος. All rights reserved.**

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Ερωτήματα που αφορούν τη χρήση της εργασίας για κερδοσκοπικό σκοπό πρέπει να απευθύνονται προς τον συγγραφέα.

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



# Περίληψη

Η παρούσα διατριβή αντιμετωπίζει το πρόβλημα της παρακολούθησης ενός αντικειμένου (π.χ. ενός ιατρικού οργάνου) σε πραγματικό χρόνο, καθώς και τους περιορισμούς που υπάρχουν για την εκτίμηση της θέσης του στον τρισδιάστατο χώρο.

Τα κύρια θέματα με τα οποία ασχολούμαστε στη συγκεκριμένη διπλωματική είναι τα ακόλουθα τρία: 1.Πρώτον, εξετάζουμε την ύπαρξη βελτιστοποιημένων συντεταγμένων στον τρισδιάστατο χώρο, οι οποίες μπορούν να αυξήσουν την ακρίβεια και την ευρωστία των μεθόδων εκτομής χώρου (localization) ενός αντικειμένου.

2.Δεύτερον, δοκιμάζουμε την εξάρτηση των βελτιστοποιημένων συντεταγμένων σε πληθώρα διαφορετικών θέσεων της κάμερας ανάγνωσης των σημείων.

3.Τέλος, αναλύουμε την εξάρτηση της θέσης της κάμερας (απόσταση από το δέκτη και γωνία προβολής) με την ακρίβεια του υπολογισμού τρισδιάστατων συντεταγμένων.

Στη συνέχεια, προσομοιώνεται μια περίπτωση Λαπαροσκοπικής Χειρουργικής Επέμβασης και τα παραπάνω ζητήματα αντιμετωπίζονται ξανά για αυτήν τη συγκεκριμένη περίπτωση. Με γνώμονα τα συμπεράσματα αυτά, προτείνεται ένας τρισδιάστατος γεωμετρικός συνδυασμός βασικών δεικτών, ο οποίος μπορεί να τοποθετηθεί σε ένα λαπαροσκόπιο για να βελτιωθεί η ακρίβεια στη διαδικασία οπτικής παρακολούθησης κατά τη διάρκεια των λαπαροσκοπικών χειρουργικών επεμβάσεων.

## Λέξεις Κλειδιά

Επαυξημένη Πραγματικότητα, Λαπαροσκοπικές Επεμβάσεις, Βελτιστοποίηση, Βασικοί Δείκτες, Εντοπισμός στο χώρο, Ρομποτική





# Abstract

The present thesis addresses the problem of tracking an object (e.g. a medical instrument) in real-time and estimating its pose. We mainly work on three different issues. First, we examine the existence of optimized 3D configurations, which can increase the accuracy and robustness of space resection methods; second, we test the dependence of optimized 3 Dimensions configurations on the camera poses. Finally, we analyze the dependence of the camera pose (distance from the marker and angle-view) on the accuracy. Afterwards, a Laparoscopic Surgery case is simulated and the above issues are addressed again for this specific case. Then, a 3D Fiducial Marker is proposed, to be placed on a Laparoscope to improve accuracy in the optical tracking process during Laparoscopic Surgeries.

## Keywords

Augmented Reality, Laparoscopic Surgeries, Optimization, Fiducial Markers, Localization, Robotics



# Ευχαριστίες

I would like to express my very profound gratitude to:

- Prof. Dimitrios Koutsouris, for being my supervisor from my home-university and for guiding and supporting me all these months.
- Prof. Dr.-Ing. Jürgen Adamy, for accepting my request to do my Master Thesis at TU Darmstadt, as an exchange student, at the Control Methods and Robotics Lab.
- Dr.-Ing. Volker Willert, for supervising me and for his remarks and engagement through the learning process of this master thesis.
- Dr. Ourania Petropoulou, for guiding and helping me through my thesis.
- M.Sc Raúl Acuña Godoy, for more than 8 months of mentoring and supervision and for always being available to answer my questions and advice me.
- My parents, Konstantinos and Dafni, my grandmother, Athina, and my brothers, Aris and Nikos, for providing me with unfailing support and continuous encouragement throughout my years of study and through the process of researching and writing this thesis. Thank you for inspiring me to follow my dreams.
- My friend, Nikos, for his encouragement when the times got rough and his endless support.
- My co-partner, Pavlos, for taking the responsibility to run on his own our organization back in Greece, while I was abroad doing my master thesis.
- My friends -the old and the new ones-, as this journey would not have been possible without them.



# Περιεχόμενα

Περίληψη	i
Abstract	iii
Ευχαριστίες	v
Περιεχόμενα	ix
Κατάλογος Σχημάτων	xiii
Κατάλογος Πινάκων	xv
<b>1 Extended Summary-Εκτεταμένη Περίληψη</b>	<b>1</b>
1.1 Εισαγωγή . . . . .	1
1.2 Κίνητρο . . . . .	1
1.3 Περιβάλλον Προγραμματισμού για προσομοίωση Πειραμάτων . . . . .	5
1.4 Δομή της Διπλωματικής . . . . .	6
1.5 Μαθηματικό Υπόβαθρο και Βασικές Αρχές . . . . .	7
1.6 Το μοντέλο της κάμερας . . . . .	8
1.7 Ο Αλγόριθμος Direct Linear Transformation . . . . .	12
1.8 Υπολογίζοντας τα Σφάλματα στην εκτίμηση θέσης . . . . .	14
1.9 Αλγόριθμοι Βελτιστοποίησης . . . . .	16
1.10 Γωνίες του Euler . . . . .	17
1.11 Αναλύοντας το πρόβλημα εκτίμησης θέσης σε σφαιρικές συντεταγμένες . . . . .	18
1.12 Οι τρισδιάστατες διατάξεις βασικών δεικτών με τη μεγαλύτερη ακρίβεια στον εντοπισμό θέσης . . . . .	24
1.13 Σχεδιάζοντας ένα βασικό δείκτη για Λαπαροσκοπικές Επεμβάσεις . . . . .	33
1.14 Συμπεράσματα . . . . .	40
<b>2 Introduction</b>	<b>41</b>
2.1 Motivation . . . . .	41
2.2 Structure . . . . .	43

<b>3</b>	<b>The Fundamentals</b>	<b>45</b>
3.1	Fiducial Markers . . . . .	45
3.2	Camera Model . . . . .	47
3.3	Calibrating Camera . . . . .	49
3.4	A Pinhole Camera . . . . .	49
3.5	Camera Pose Estimation . . . . .	50
3.6	Perspective-n-Point . . . . .	50
3.7	Direct Linear Transformation . . . . .	51
3.7.1	Extracting the Rotation $\hat{\mathbf{R}}$ . . . . .	52
3.7.2	Extracting the Translation $\hat{\mathbf{t}}$ . . . . .	52
3.8	Pose Estimation Errors . . . . .	52
3.8.1	Rotational Error . . . . .	52
3.8.2	Translation Error . . . . .	53
3.8.3	Root Mean Square Error . . . . .	53
3.8.4	Reprojection Error . . . . .	53
3.8.5	Condition Number . . . . .	53
3.9	Covariance Matrix . . . . .	54
3.10	Error Covariance Ellipsoids . . . . .	54
3.11	Back Propagation . . . . .	55
3.11.1	SuperSAB Algorithm . . . . .	55
3.11.2	Gradient Descent . . . . .	55
3.11.3	Adam Optimization . . . . .	57
3.12	Euler Angles . . . . .	57
3.13	Gaussian (Normal) Distribution . . . . .	58
3.14	Convert Cartesian to Spherical Coordinates . . . . .	58
<b>4</b>	<b>Simulation Setup</b>	<b>59</b>
4.1	Camera Model . . . . .	59
4.2	Sphere Real-World Object . . . . .	60
4.3	Using Direct Linear Transformation to the Perspective-n-Point Problem . . . . .	61
4.4	Pose Estimation Errors . . . . .	61
<b>5</b>	<b>Analyzing PnP in Spherical Coordinates</b>	<b>63</b>
5.1	Preliminaries . . . . .	63
5.2	Covariance matrix for detecting the marker . . . . .	64
5.3	Relation between Accuracy and the Distance . . . . .	67
5.4	Relation between Accuracy and the Angle View . . . . .	68
5.5	Draw Ellipsoids to represent errors of $\Sigma_{\alpha,\beta,r}$ . . . . .	68
5.6	Concluding Remarks . . . . .	71

---

<b>6</b>	<b>Most accurate 3D Points Configurations for Pose Estimation</b>	<b>73</b>
6.1	Preliminaries . . . . .	73
6.2	Random Selection of Points . . . . .	74
6.3	Gradient Descent . . . . .	77
6.4	Adam Optimizer . . . . .	80
6.5	Estimate accurate 3D Points Configurations, using spherical coordinates . .	84
6.5.1	Comparing Errors for the 2 different positions of the Camera . . . .	85
6.6	Concluding Remarks . . . . .	88
<b>7</b>	<b>Current Applications of Fiducial Markers in Laparoscopic Surgeries</b>	<b>91</b>
7.1	Surgical Augmented Reality . . . . .	91
7.2	Optical Tracking for medical devices localization . . . . .	91
7.3	Electromagnetic Tracking Systems . . . . .	92
7.4	Comparing Optical and Electromagnetic Tracking Systems . . . . .	94
7.5	Fiducial Markers for Optical Tracking in Medical Applications . . . . .	94
7.5.1	2D Fiducial Marker for Endoscope Tracking . . . . .	95
7.5.2	Stereoscopic augmented reality for laparoscopic surgery . . . . .	96
7.5.3	Near Infrared Optical Tracking System . . . . .	97
7.5.4	Augmented reality navigation system for laparoscopic splenectomy in children . . . . .	97
<b>8</b>	<b>Applying optimized 3D Points Configurations for Pose Estimation on Laparoscopic Surgeries</b>	<b>99</b>
8.1	Designing our Fiducials . . . . .	99
8.2	Re-validate the Fiducials Configuration using Gradient Descent Optimiza- tion Algorithm . . . . .	102
8.3	Comparing Errors between well- and ill-conditioned Points Configurations .	105
8.3.1	Root Mean Square Errors . . . . .	105
8.3.2	Translation Errors and Rotational Errors . . . . .	105
8.3.3	Reprojection Errors . . . . .	107
8.3.4	Condition Number and Condition Number of Covariance Matrix . .	108
8.4	Concluding Remarks . . . . .	109
<b>9</b>	<b>Conclusion and Future Works</b>	<b>111</b>
9.1	Conclusion . . . . .	111
9.2	Outlook . . . . .	112





# Κατάλογος Σχημάτων

1.1	Παράδειγμα occlusion στη χρήση επαυξημένης πραγματικότητας στην ιατρική [14]. . . . .	2
1.2	Παραδείγματα βασικών δεικτών. . . . .	8
1.3	Το μοντέλο μίας κάμερας με οπές. . . . .	11
1.4	Εκτίμησης θέσης της κάμερας. . . . .	12
1.5	Ο Αλγόριθμος Βελτιστοποίησης Gradient Descent. . . . .	16
1.6	Η πλοήγηση της κάμερας σε σφαιρικές συντεταγμένες. . . . .	20
1.7	Η σχέση μεταξύ του Condition number και της απόστασης της κάμερας από το βασικό δείκτη ( $X \sim \mathcal{N}(0, 4)$ ). . . . .	22
1.8	Η εξάρτηση της ακρίβειας της γωνίας $\alpha$ της κάμερας και του βασικού δείκτη ( $X \sim \mathcal{N}(0, 4)$ ). . . . .	23
1.9	Ελλειψοειδή Σφάλματος του Πίνακα Συνδιακύμανσης $\Sigma_{\alpha, \beta, r}$ . . . . .	24
1.10	Ακρίβεια της θέσης της κάμερας σε σχέση με το βασικό δείκτη για διάφορες γωνίες $\alpha$ [6]. . . . .	25
1.11	Οι διατάξεις των σημείων στην πρώτη εκτέλεση του Gradient Descent. . . . .	26
1.12	Οι διατάξεις των σημείων στην 50 επανάληψη του Gradient Descent. . . . .	27
1.13	Η τελική διάταξη των σημείων που έδειξε η εκτέλεση του Gradient Descent. . . . .	27
1.14	Συγκρίνοντας τις διατάξεις σημείων της πρώτης και της εκατοστής επανάληψης του αλγορίθμου Gradient Descent. . . . .	28
1.15	Σφάλματα κατά την εκτέλεση του Gradient Descent ( $X \sim \mathcal{N}(2, 8)$ ). . . . .	29
1.16	Οι γεωμετρικές διατάξεις των 6 σημείων με τη χρήση του αλγορίθμου Adam. . . . .	30
1.17	Σύνολο σημείων της πρώτης και της τελευταίας επανάληψης στην εκτέλεση του Adam Optimizer. . . . .	31
1.18	Σφάλματα σε κάθε επανάληψη του Adam Optimization ( $X \sim \mathcal{N}(2, 8)$ ). . . . .	32
1.19	Οι διατάξεις τριγωνικών πρισμάτων. . . . .	32
1.20	Η προτεινόμενη τρισδιάστατη διάταξη σημείων για να προσαρτηθούν σε ένα Λαπαροσκόπιο. . . . .	34
1.21	Η γεωμετρία των βασικών δεικτών. . . . .	35
1.22	Το τελικό πρίσμα εγγεγραμμένο στη σφαίρα ακτίνας 0.07071 μέτρων. . . . .	35
1.23	Συγκρίνοντας τα Σφάλματα Μετάφρασης. . . . .	37
1.24	Συγκρίνοντας τα Σφάλματα Περιστροφής. . . . .	37

1.25	Συγκρίνοντας τα Σφάλματα Επαναπροβολής. . . . .	38
1.26	Συγκρίνοντας τα Condition Number του $H_{\text{DLT}}$ . . . . .	39
1.27	Συγκρίνοντας τα Condition Number των πινάκων συνδιακύμανσης του $H_{\text{DLT}}$ . . . . .	39
1.28	Η τρισδιάστατη διάταξη σημείων που προτείνουμε στο βασικό δείκτη προσαρμοσμένη σε ένα Λαπαροσκόπιο. . . . .	40
3.1	Examples of fiducial markers. . . . .	46
3.2	Camera model. . . . .	48
3.3	The Pinhole Camera Model. . . . .	49
3.4	Camera Pose Estimation. . . . .	50
3.5	Gradient Descent Optimization. . . . .	56
4.1	Plot 1000 random Sphere Points using Mayavi. . . . .	61
5.1	Navigation of camera on spherical coordinates. . . . .	65
5.2	Relation between Condition number and Distance $r$ ( $X \sim \mathcal{N}(0, 4)$ ). . . . .	68
5.3	Accuracy Depending on the $\alpha$ Viewing Angle ( $X \sim \mathcal{N}(0, 4)$ ). . . . .	69
5.4	Error Covariance Ellipsoid. . . . .	70
5.5	Accuracy of the pose of the camera in marker coordinates [6]. . . . .	70
6.1	Six-Points Configuration with the minimum condition number after $N=500$ tests. . . . .	76
6.2	Six-Points Configuration with minimum condition number after $N = 10.000$ tests. . . . .	76
6.3	Points Configuration at the first iteration of Gradient Descent. . . . .	78
6.4	Points Configuration at the $50^{\text{th}}$ iteration of Gradient Descent. . . . .	79
6.5	Final Points Configuration using Gradient Descent. . . . .	79
6.6	Errors in Gradient Descent ( $X \sim \mathcal{N}(2, 8)$ ). . . . .	81
6.7	Comparing Sets of Points from first and last iteration of Gradient Descent. . . . .	82
6.8	Six-Points Configurations using Adam. . . . .	82
6.9	Set of points from the first and the last iteration of Adam Optimizer. . . . .	83
6.10	Errors in each iteration of Adam Optimization ( $X \sim \mathcal{N}(2, 8)$ ). . . . .	84
6.11	Points Configuration at the first iteration of Gradient Descent. . . . .	85
6.12	Points Configuration of Gradient Descent ( $N = 50$ ). . . . .	86
6.13	Final Points Configuration of Gradient Descent ( $N = 100$ ). . . . .	86
6.14	Comparing Sets of Points from first and last iteration of Gradient Descent. . . . .	87
6.15	Errors in each iteration of Gradient Descent at the new position of the Camera ( $X \sim \mathcal{N}(2, 8)$ ). . . . .	88
6.16	Rotational Error in both cases: First position of the camera and the final one. . . . .	89
6.17	Error in Translation in both cases: First position of the camera and the final one. . . . .	89

6.18	Observed Triangular Prisms. . . . .	90
7.1	The basic principles of AR. [38] . . . . .	92
7.2	A fixed markers pattern placed on the laparoscope for optical tracking and the detection cameras [7]. . . . .	93
7.3	Principle of intra-operative electromagnetic (EM) tracking [16]. . . . .	93
7.4	Already developed laparoscope marker body configurations [13]. . . . .	95
7.5	The proposed fiducial marker [17]. . . . .	96
7.6	Custom-designed fixtures, with reference frames attached, for the 3D laparoscope [24]. . . . .	97
7.7	Schematic of surgical tool and the coordinate relationship between the tip of surgical tool and markers [43]. . . . .	98
7.8	System configuration of AR navigation for laparoscopic splenectomy [22]. . . . .	98
8.1	Our Proposed 3D Point Configurations to be attached on the Laparoscope. . . . .	100
8.2	Fiducial's Geometry. . . . .	101
8.3	Final Points Configuration Inscribed in a Sphere. . . . .	102
8.4	"Ill-conditioned" random initial points ( $N = 1$ ). . . . .	103
8.5	Points Configuration at $N = 50$ . . . . .	103
8.6	The final Points Configuration at $N = 100$ . . . . .	104
8.7	Errors in Gradient Descent Optimization $X \sim \mathcal{N}(2, 8)$ . . . . .	105
8.8	Comparing Errors in Translation. . . . .	106
8.10	Comparing Reprojection Errors. . . . .	107
8.9	Comparing Rotational Errors. . . . .	107
8.11	Comparing Condition Numbers of $H_{\text{DLT}}$ . . . . .	108
8.12	Comparing Condition Numbers of Covariance Matrices of $H_{\text{DLT}}$ . . . . .	109
8.13	Our Proposed Fiducial Marker attached on a Laparoscope. . . . .	110



# Κατάλογος Πινάκων

1.1	Σχέση μεταξύ $sc$ και επιπέδου εμπιστοσύνης σε 1,2 και 3 διαστάσεις. . . . .	15
1.2	Διαθέσιμα μεγέθη για παθητικούς σφαιρικούς δείκτες στην αγορά. . . . .	34
1.3	Μέσα Σφάλματα των διατάξεων Τυχαίων Σημείων Σφαίρας και των Σημείων Τριγωνικού Πρίσματος. . . . .	38
3.1	Relation between scale and confidence level in different dimensions. . . . .	55
7.1	Comparison Between Optical and Electromagnetic Tracking Systems [19]. . . . .	94
8.1	Available Sizes for Passive Spherical Markers in the Market. . . . .	100
8.2	Mean Errors of Sphere- and Prism- Points Configurations. . . . .	108



# Κεφάλαιο 1

## Extended Summary-Εκτεταμένη Περίληψη

### 1.1 Εισαγωγή

Σε αυτό το εισαγωγικό κεφάλαιο αναλύονται, αρχικά το κίνητρο το οποίο οδήγησε στην επιλογή του συγκεκριμένου θέματος διπλωματικής εργασίας και έπειτα παρατίθεται η μορφολογία των κεφαλαίων που θα ακολουθήσουν.

### 1.2 Κίνητρο

Η εκτίμηση της θέσης μίας κάμερας είναι μία σημαντική πρόκληση σε πληθώρα εφαρμογών που απαιτούν τον ακριβή εντοπισμό αντικειμένων στο περιβάλλον. Για παράδειγμα, σε εφαρμογές επαυξημένης πραγματικότητας ή σε εφαρμογές στον τομέα της ρομποτικής, χρειάζεται η εκτίμηση αυτής της θέσης να έχει όσο μικρότερη απόκλιση από την πραγματική θέση γίνεται. Επιπλέον, ιδιαίτερα σε εφαρμογές επαυξημένης πραγματικότητας, μπορεί να εμφανίζονται και προβλήματα έμφραξης, στις περιπτώσεις που το πραγματικό αντικείμενο εμφανίζεται να κάνει οσζλυδε την εικονική/ψηφιακή σκηνή. Μία πιθανή περίπτωση occlusion σε εφαρμογές επαυξημένης πραγματικότητας φαίνεται στην εικόνα 1.1.

Ένας ακόμα σημαντικός περιορισμός είναι ότι όλες οι γνωστές μέθοδοι διατομής χώρου είναι εγγενώς μη γραμμικές και πολύ ευαίσθητες στον θόρυβο, όταν εκφράζονται σε συντεταγμένες της εικόνας που προβάλλεται από την κάμερα σε διδιάστατη απεικόνιση. Επομένως, η ακρίβεια της διαδικασίας εξαρτάται και από τη διαμόρφωση του marker που χρησιμοποιείται, η οποία δεν είναι προβλέψιμη σε κάθε εφαρμογή και ιδιαίτερα σε εφαρμογές όπου ένας βασικός δείκτης παρακολουθείται στο χώρο.

Για να ξεπεραστούν αυτές οι προκλήσεις, πρέπει να υπολογιστούν με όσο το δυνατόν μεγαλύτερη ακρίβεια οι αντιστοιχίες μεταξύ γνωστών σημείων στο περιβάλλον (στον τρισδιάστατο χώρο) και των σημείων που προβάλλονται από την κάμερα (στο διδιάστατο πεδίο απεικόνισης) [42].

Για να επιτευχθεί αυτό, χρησιμοποιούνται βασικοί δείκτες (fiducial markers), δηλαδή αντι-



(a) Occlusion handling for multiple tools



(b) Cheek-bone detail

Σχήμα 1.1: Παράδειγμα occlusion στη χρήση επαυξημένης πραγματικότητας στην ιατρική [14].



κείμενα με προκαθορισμένο μέγεθος, τα οποία εμφανίζονται στην καταγεγραμμένη εικόνα και χρησιμοποιούνται ως σημεία αναφοράς ή ως μέτρο ενός μεγέθους [34]. Τα fiducial markers αυτά επιλέγονται συνήθως σε εφαρμογές λόγω της ευκολίας που παρουσιάζουν στη χρήση, καθώς και την ακρίβεια που παρέχουν στη διαδικασία εντοπισμού τους.

Σε εφαρμογές, όπως η χρήση των βασικών δεικτών αυτών στην ιατρική για την παρακολούθηση ενός ενδοσκοπίου ή ενός λαπαροσκόπιου κατά τη διάρκεια χειρουργικών επεμβάσεων, η διάταξη όλων των επιμέρους στοιχείων που συνθέτουν το βασικό δείκτη είναι ζωτικής σημασίας, διότι όσο τα σφάλματα (για παράδειγμα κατά τη μετάφραση, περιστροφή και επαναπροβολή) και ο το condition number μειώνονται, η ακρίβεια της διαδικασίας αυξάνεται, καθιστώντας τα χαρακτηριστικά αυτά αντιστρόφως ανάλογα. Εν συνεχεία, σε άλλες εφαρμογές, όπως το τυπωμένο ηλεκτρικό κύκλωμα για αυτοματοποιημένο εξοπλισμό, τα fiducials βοηθούν τις μηχανές να αναγνωρίσουν τον τύπο και τον προσανατολισμό της εκάστοτε πλακέτας.

Είναι ακόμα σημαντικό πως κάθε τεχνική και μέθοδος για την δημιουργία περιβάλλοντων επαυξημένης πραγματικότητας, πρέπει να μπορεί να εγγυηθεί ότι θα εναρμονίζει την ψηφιακή εικόνα με το πραγματικό περιβάλλον. Διότι η επαυξημένη πραγματικότητα βασίζεται στην απόλυτη ευθυγράμμιση των μοντέλων ψηφιακών εικόνων και του πραγματικού αντικειμένου, το οποίο πραγματικό αντικείμενο ποικίλει ανάλογα με την εφαρμογή και μπορεί να είναι ένας ασθενής, ένα τυπωμένο κύκλωμα, ένα ρομπότ, ένα ανθρώπινο όργανο και άλλα. Έτσι, η ελαχιστοποίηση των σφαλμάτων στον προσανατολισμό και την αναπροσαρμογή είναι ένα σημαντικό βήμα για τη βελτίωση στη δημιουργία περιβάλλοντων επαυξημένης πραγματικότητας. Συνεπώς, τα fiducial markers μπορούν να είναι πραγματικά χρήσιμα όσον αφορά την αναγνώριση της θέσης ενός αντικειμένου στον τρισδιάστατο χώρο.

Ωστόσο, η ακρίβεια κάθε fiducial marker μπορεί να βελτιωθεί σημαντικά, οπότε απαιτείται να εξεταστεί εάν υπάρχουν συγκεκριμένες διαμορφώσεις ή γεωμετρικές διατάξεις σημείων, που είναι σε θέση να ελαχιστοποιήσουν τα σφάλματα προσανατολισμού.

Επιπρόσθετα είναι σημαντικό να τονισθεί πως η ακριβής εκτίμηση θέσης είναι μία από τις βασικές επιδιώξεις στην όραση των υπολογιστών. Από την παρακολούθηση στόχων έως τη συναρμολόγηση μηχανημάτων, απαιτείται ο ακριβής εντοπισμός της κάμερας και των βασικών δεικτών. Παρόλο που, υπάρχουν πολλές προηγούμενες έρευνες σχετικά με αλγόριθμους που μπορούν να λύσουν αυτό το πρόβλημα, το σχήμα, το μέγεθος, οι διαμορφώσεις των fiducial markers και πως τα γεωμετρικά τους χαρακτηριστικά μπορούν και επιδρούν στην ακρίβεια της εκτίμησης της θέσης του πραγματικού αντικειμένου προς μελέτη, δεν έχουν διερευνηθεί περαιτέρω.

Έτσι, σε αυτή τη διπλωματική εργασία, αποφασίσαμε να αναλύσουμε και να δοκιμάσουμε συστήματα παρακολούθησης, τα οποία συνήθως βασίζονται σε fiducial markers και, να εξετάσουν όλες οι γεωμετρικές παράμετροι των σημείων των βασικών δεικτών προκειμένου να διερευνηθεί η επίδραση τους στα σφάλματα προσανατολισμού.

Ο κύριος στόχος αυτής της διπλωματικής είναι ο προσδιορισμός των γεωμετρικών διατάξεων τρισδιάστατων βασικών δεικτών, οι οποίες μπορούν να αυξήσουν την ακρίβεια της διαδικασίας παρακολούθησης σε έναν δεδομένο τρισδιάστατο όγκο προς μελέτη. Οι τρισδιάστατες διατάξεις αυτές δύνανται να έχουν πληθώρα εφαρμογών όπως είναι ο εντοπισμός και

η μετακίνηση ενός ρομπότ.

Μέσα στην παρούσα διπλωματική εργασία εξετάζονται επίσης και η σχέση των γωνιών της κάμερας λήψης με το πραγματικό προς μελέτη αντικείμενο και οι οπτικές πλευρές της κάμερας που παρέχουν μεγαλύτερη ακρίβεια στον εντοπισμό αντικειμένων. Αυτές οι γωνίες ή οι οπτικές πλευρές της κάμερας μπορούν να χρησιμοποιηθούν για παράδειγμα για την καλύτερη τοποθέτηση δύο ρομπότ μεταξύ τους, προκειμένου να βελτιωθεί η ακρίβεια της στάσης τους τόσο του ενός ως προς το άλλο όσο και σε σχέση με έναν σταθερό σημείο στο περιβάλλον.

Εκτός αυτών, στη διπλωματική αυτή μελετώνται και ξεχωριστά ιατρικές εφαρμογές με τη χρήση βασικών δεικτών και όλα τα ανωτέρω συμπεράσματα εφαρμόζονται σε αυτή την περίπτωση. Έτσι αυτές οι γεωμετρικές διατάξεις σημείων μπορούν να χρησιμοποιηθούν για παράδειγμα για τον εντοπισμό και την εκτίμηση της θέσης των ιατρικών συσκευών όπως ένα λαπαροσκόπιο κατά τη διάρκεια μιας ιατρικής διαδικασίας. Επιπλέον, οι γωνίες της κάμερας για καλύτερη εκτίμηση συντεταγμένων ενός σημείου, μπορούν να αυξήσουν την ακρίβεια μίας ελάχιστα επεμβατικής διαδικασίας π.χ. μιας επέμβασης.

Μία ιμαγε-γυιδεδ επέμβαση, και συγκεκριμένα η χρήση επαυξημένης πραγματικότητας σε λαπαροσκοπικές επεμβάσεις, έχει ήδη αποδειχθεί ως ένα σημαντικό εργαλείο καθοδήγησης του χειρουργού. Συγκεκριμένα η επαυξημένη πραγματικότητα στις λαπαροσκοπικές επεμβάσεις παρέχει ενδοεγχειρητική καθοδήγηση με τον προσδιορισμό των υποεπιφανειακών στόχων και των κρίσιμων δομών και εμφανίσεων κοπτικών τροχιών, οι οποίες εικονικοποιούνται εκ των προτέρων. Βοηθά επίσης τον χειρουργό να αποφύγει περιττές πληροφορίες από διαφορετικές πηγές στη σκηνή του χειρουργείου. Για αυτό το λόγω οι τεχνικές επαυξημένης πραγματικότητας με αντίστοιχες εφαρμογές θεωρούνται εφαρμογές θεωρούνται λίγο πολύ επιτυχημένες, ωστόσο, υπάρχουν προκλήσεις και περιορισμοί, οι οποίοι πρέπει να ξεπεραστούν.

Πιο συγκεκριμένα η τεχνική της παρακολούθησης αντικειμένων είναι ένα από τα εμπόδια για την επαυξημένη πραγματικότητα εν γένει. Ειδικά στις λαπαροσκοπικές χειρουργικές επεμβάσεις, απαιτείται να προσδιοριστεί η στάση όλων των ενδοχειρουργικά χρησιμοποιούμενων συσκευών απεικόνισης με ιδιαίτερα μικρό σφάλμα, διαφορετικά η θέση τους θα είναι αβέβαιη ή και άγνωστη και προβλήματα ος ελυσισιον δύναται να εμφανιστούν.

Συνεπώς αυτή η διπλωματική εμπεριέχει τα αποτελέσματα τις ερευνάς μας στα ανωτέρω προβλήματα και συνοψίζονται ως εξής:

1. Εύρεση των γεωμετρικών διατάξεων τρισδιάστατων σημείων, τα οποία μπορούν να οδηγήσουν στην ελαχιστοποίηση σφαλμάτων σε μετάφραση, περιστροφή και επαναπροβολή, όπως επίσης και το condition number του  $\mathbf{H}_{DLT}$  πίνακα που μελετάται.
2. Παρακολούθηση των παραπάνω σφαλμάτων και εύρεση της σχέσης μεταξύ της όξυνσης της ακρίβειας στον εντοπισμό ενός αντικειμένου και της θέσης της κάμερας στο χώρο.
3. Εφαρμογή των παραπάνω αποτελεσμάτων και εκτιμήσεων, σε μία προσομοίωση οπτικής παρακολούθησης, βασισμένη στη χρήση fiducial markers για παρακολούθηση ενός ενδοσκοπίου σε μια Λαπαροσκοπική Χειρουργική Επέμβαση.

## 1.3 Περιβάλλον Προγραμματισμού για προσομοίωση Πειραμάτων

Για τους σκοπούς της παρούσας διπλωματικής αναπτύχθηκαν κατάλληλες προσομοιώσεις με τη χρήση της γλώσσας προγραμματισμού Python. Όλος ο κώδικας που αναπτύχθηκε είναι διαθέσιμος σε αυτόν τον σύνδεσμο:

<https://github.com/D1vt/DLT3D>.

Πιο συγκεκριμένα οι κλάσεις, οι συναρτήσεις και τα υποπρογράμματα που παρουσιάζονται αναλύονται ως εξής:

- **Sphere:** Μία κλάση που δημιουργεί μία σφαίρα της ακτίνας και των χαρακτηριστικών που επιλέγονται από τον χρήστη. Μπορεί να βρεθεί στο ακόλουθο path:  
`/D1vt/DLT3D/Python/sphere.py`
- **DLT and errors:** Ο Αλγόριθμος Direct Linear Transformation που χρησιμοποιήθηκε για την εκτέλεση των πειραμάτων της διπλωματικής. Ο αλγόριθμος αυτός δέχεται σημεία του τρισδιάστατου χώρου και εξάγει την εκτιμώμενη περιστροφή και μετάφραση τους καθώς και τη θέση του κέντρου της κάμερας κατά την απεικόνιση τους σε δύο διαστάσεις σε μορφή εικόνας. Με αυτό τον κώδικα υπολογίζονται επίσης τα σφάλματα κατά την μετάφραση και περιστροφή των πραγματικών σημείων του χώρου σε διδιάστατα σημεία εικόνας, το μέσο τετραγωνικό σφάλμα και το σφάλμα επαναπροβολής. Τα παραπάνω μπορούν να βρεθούν στον ακόλουθο σύνδεσμο: `/D1vt/DLT3D/Python/dltanderrors.py`  
Στον κώδικα διατίθενται δύο εναλλακτικές επιλογές εκτέλεσης του αλγορίθμου. Η πρώτη `def DLT3D` δέχεται ως είσοδο ακριβώς έξι(6) σημεία του πραγματικού τρισδιάστατου χώρου και τα αντίστοιχα διδιάστατα σημεία στο πλάνο της εικόνας. Για να λειτουργήσει ο Αλγόριθμος σε τρισδιάστατο χώρο απαιτεί είσοδο από τουλάχιστον 6 σημεία ως είσοδο κατά την εφαρμογή του. Επιπλέον, η δεύτερη επιλογή είναι η συνάρτηση `def HDLTwithfor`, που δέχεται οποιοδήποτε  $N$  αριθμό τρισδιάστατων σημείων ως είσοδο  $N \geq 6$ . Ο λόγος που υπάρχουν δύο τέτοιες συναρτήσεις είναι πως η πρώτη είναι γρογορότερη σε θέμα εκτελεστικού χρόνου και όταν στα πειράματα μας χρησιμοποιούσαμε συντεταγμένες 6 σημείων χρησιμοποιήθηκε αυτή η συνάρτηση.
- **$R_{\text{Euler}}$ :** Μία συνάρτηση που δημιουργεί τον πίνακα της περιστροφής, που θα αναλυθεί και στη συνέχεια στην εξίσωση (3.29) βασισμένο στις γωνίες του Euler. Η συνάρτηση είναι διαθέσιμη εδώ: `/D1vt/DLT3D/Python/RFromEulerAngles.py`
- **3D Plotting:** Ο κώδικας για την δημιουργία τρισδιάστατων γραφικών παραστάσεων χρησιμοποιώντας το εργαλείο `mayavi` είναι διαθέσιμος εδώ:  
`D1vt/DLT3D/Python/3DplotSphere.py`
- **Gradient Descent:** Η τεχνική βελτιστοποίησης με τη χρήση της κλίσης κατάβασης (Gradient Descent Optimization), για 100 επαναλήψεις, όπως επίσης και ο αλγόριθμος `superSAB` βρίσκονται εδώ: `D1vt/DLT3D/Python/gradient3dpoints.py`

- Adam optimization: Για τα πειράματα μας χρησιμοποιήθηκε επιπλέον και ο αλγόριθμος βελτιστοποίησης Adam optimizer με τις κατάλληλες τροποποιήσεις σε παραμέτρους για τη χρήση του για τα πειράματα της παρούσας διπλωματικής. Ο κώδικας βρίσκεται εδώ: `D1vt/DLT3D/Python/adam3dpoints.py`
- Πίνακες περιστροφής και μετάφρασης σε σφαιρικές συντεταγμένες: Άλλες οι μαθηματικές εξισώσεις που παρουσιάζονται στο κεφάλαιο 5, σχετικά με τα  $R_{\text{spherical}}$ ,  $t_{\text{spherical}}$ , τον Ιακωβιανό πίνακα  $J_p$  και τον πίνακα συνδιακύμανσης  $\Sigma_{\alpha,\beta,r}$  αναπτύσσονται εδώ: `D1vt/DLT3D/Python/Rtsphericalcoordinates.py`
- Αλγόριθμος επιλογής τυχαίων σημείων: Ο αλγόριθμος αυτός επιλέγει τυχαίους συνδυασμούς 6 σημείων μετά από έναν  $N$  αριθμό επαναλήψεων που του ορίζει ο χρήστης τα οποία είχαν το μικρότερο condition number σε σχέση με τις άλλες τυχαίες διατάξεις σημείων που δοκιμάστηκαν και φαίνεται εδώ: `D1vt/DLT3D/Python/randompointsconfiguration.py`
- Αλγόριθμος για γραφική αναπαράσταση των Error Covariance Ellipsoids: Ο αλγόριθμος δέχεται ως είσοδο το επίπεδο εμπιστοσύνης και τον πίνακα συνδιακύμανσης και παράγει τα αντίστοιχα ελλειψοειδή. Ο αλγόριθμος χρειάζεται επιπλέον ως είσοδο τα  $\alpha$ ,  $\beta$  και  $r$  όπως φαίνεται εδώ: `D1vt/DLT3D/Python/plotellipsoid.py`
- Κύρια Συνάρτηση: Όλα τα πειράματα για τις παραπάνω συναρτήσεις βρίσκονται εδώ και επιπλέον εδώ ορίζονται οι παράμετροι για την κλάση της σφαίρας, τον τυχαίο θόρυβο που προστίθεται καθώς και τις ιδιότητες της απεικόνισης των σημείων: `D1vt/DLT3D/Python/main.py`

## 1.4 Δομή της Διπλωματικής

Η διπλωματική αυτή έχει αναλυθεί στα εξής κεφάλαια:

- Στο Κεφάλαιο 2 γίνεται μία εισαγωγή όλων των βασικών πειραμάτων και συμπερασμάτων της διπλωματικής εργασίας,
- Στο Κεφάλαιο 3 αναφέρονται οι βασικοί ορισμοί και οι μαθηματικές σχέσεις που χρησιμοποιήθηκαν.
- Στο Κεφάλαιο 4, αναλύεται το στήσιμο και η υλοποίηση των προσομοιώσεων και των πειραμάτων στα πλαίσια της διπλωματικής.
- Στο Κεφάλαιο 5, φαίνονται όλα τα συμπεράσματα στα οποία καταλήξαμε αναλύοντας το πρόβλημα εκτίμησης θέσης σε σφαιρικές συντεταγμένες. Παράλληλα παρατίθενται και τα ελληψοειδή σφάλματος συνδιακύμανσης.
- Στο Κεφάλαιο 6, περιλαμβάνονται οι γεωμετρικές διατάξεις που βρέθηκαν να οδηγούν σε μικρότερα σφάλματα και προτείνονται να χρησιμοποιούνται ως διατάξεις κατασκευής βασικών δεικτών.

- Στο Κεφάλαιο 7, παρουσιάζονται περιληπτικά οι διατάξεις fiducial markers που χρησιμοποιούνται σήμερα σε Λαπαροσκοπικές Επεμβάσεις.
- Στο Κεφάλαιο 8, εμπεριέχονται όλα τα πειράματα και τα αποτελέσματα με την εφαρμογή της γεωμετρικής διάταξης από fiducial markers που προτείνουμε.
- Στο Κεφάλαιο 9, παραθέτουμε τα συμπεράσματα στα οποία καταλήξαμε και προτείνουμε ενδεικτικές μελλοντικές ιδέες και εφαρμογές έρευνας στο πεδίο της διπλωματικής αυτής.

Στην αρχή κάθε κεφαλαίου, δίνεται μία σύντομη περίληψη αντίστοιχων μελετών και εργασιών, που χρησιμοποιήθηκαν ως αφετηρία στην έρευνά μας.

## 1.5 Μαθηματικό Υπόβαθρο και Βασικές Αρχές

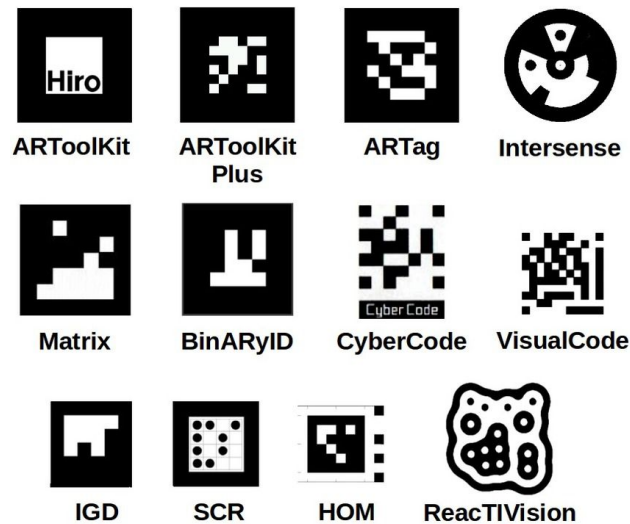
Στην παρούσα ενότητα θα ανεφερθούν όλες οι βασικές θεωρητικές και μαθηματικές έννοιες που απαιτούνται για τη μελέτη και κατανόηση της συγκεκριμένης διπλωματικής εργασίας.

Αρχικά με τον όρο βασικοί δείκτες ή fiducial markers αναφερόμαστε σε ένα αντικείμενο που μπορεί να κατασκευαστεί από χρυσό, άνθρακα ή πολυμερές και το μέγεθός του όπως και η γεωμετρική του διάταξη διαφέρει ανάλογα με την εφαρμογή για την οποία χρησιμοποιείται [34]. Ο βασικός δείκτης τοποθετείται είτε εσωτερικά είτε εξωτερικά από το αντικείμενο μελέτης που ενδιαφερόμαστε να απεικονίσουμε. Τα κριτήρια για την ανάπτυξη ενός βασικού δείκτη είναι τα εξής [18]:

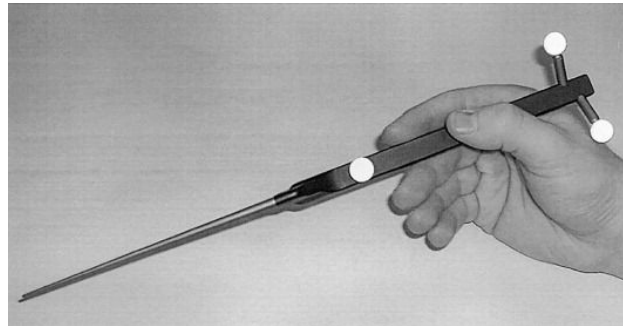
1. Ανθεκτικότητα και ανιχνευσιμότητα: Οι βασικοί δείκτες πρέπει να είναι ανθεκτικοί σε πληθώρα συνθηκών (π.χ. θερμοκρασία) και ανιχνεύσιμοι, κυρίως υπό διαφορετικές συνθήκες φωτισμού.
2. Ακρίβεια: Θα πρέπει ο βασικός δείκτης να έχει κατασκευαστεί με τρόπο που να επιτρέπεται η ακριβής θέση του μέσα σε μια παρατηρούμενη σκηνή.
3. Διακριτική ικανότητα: Η ικανότητα μιας μεθόδου παρακολούθησης να κλιμακωθεί ανάλογα με τις απαιτήσεις που επιβάλλονται από μία συγκεκριμένη εφαρμογή ή σενάριο.
4. Οικονομική σκοπιμότητα: Η ανάγκη για φθηνούς και εύκολα παραγωγικούς δείκτες.

Το σχήμα και οι διαστάσεις ενός βασικού δείκτη είναι γνωστά πριν από κάθε εφαρμογή και χρησιμοποιείται συχνά για τον καθορισμό μίας θέσης αναφοράς σε παγκόσμιες συντεταγμένες που μπορούν εύκολα να ανιχνευθούν σε εικόνες κάμερας [28].

Στον τομέα της επαυξημένης πραγματικότητας, οι βασικοί δείκτες αποτελούν ένα χρήσιμο εργαλείο για την ορθή ευθυγράμμιση και εν συνεχεία ενοποίηση της πραγματικής και της ψηφιακής/σύνθετης πληροφορίας. Συνήθως, ένα σύστημα βασικών δεικτών αποτελείται από ένα σύνολο έγχρωμων markers και ενός αλγορίθμου για την ανίχνευσή τους, ενώ μπορεί να χρησιμοποιηθεί σε πολλές εφαρμογές που σχετίζονται με την όραση του υπολογιστή, οι οποίες απαιτούν παρακολούθηση και εντοπισμό (π.χ. οπτικό εντοπισμό, αναγνώριση αντικειμένων, βαθμονόμηση κάμερας και παθητική πλοήγηση).



(α) Διδιάστατοι βασικοί δείκτες [18].



(β') Τρισδιάστατο σύστημα βασικών δεικτών [15].

Σχήμα 1.2: Παραδείγματα βασικών δεικτών.

Εστιάζοντας σε βιοϊατρικές εφαρμογές και πιο συγκεκριμένα στη χρήση της επανυξημένης πραγματικότητας σε λαπαροσκοπικές επεμβάσεις, οι βασικοί δείκτες είναι πραγματικά χρήσιμοι για την επίλυση προβλημάτων παρακολούθησης, καθώς υπάρχει ανάγκη για πιο προηγμένες οπτικοποιήσεις που μπορούν να βελτιώσουν την προβολή ώστε να περιλαμβάνουν πληροφορίες κάτω από την επιφάνεια των οργάνων, δηλαδή αγγεία της αορτής, όγκοι και όργανα κατά τη διάρκεια της χειρουργικής επέμβασης.

Στο Σχήμα 1.2 υπάρχουν παραδείγματα fiducial markers, τα οποία χρησιμοποιούνται για εκτίμηση της θέσης κάμερας σε πολλές εφαρμογές. Συγκεκριμένα, το Σχήμα 1.2α' δείχνει διαφορετικούς τύπους επίπεδων βασικών δεικτών (διδιάστατων), ενώ στο Σχήμα 1.2β', παρουσιάζεται μια εφαρμογή των τρισδιάστατων βασικών δεικτών στη Λαπαροσκόπηση.

## 1.6 Το μοντέλο της κάμερας

Σε όλες τις μετρήσεις της διπλωματικής χρησιμοποιήθηκε η προσομοίωση μίας κάμερας, καθώς και όλες οι εξωγενείς και ενδογενείς παράμετροί της. Οι εξωγενείς παράμετροι καθορίζουν τη θέση και τον προσανατολισμό της κάμερας (δηλαδή την περιστροφή  $\mathbf{R}$  και την

μετάφραση  $\mathbf{t}$ ), ενώ οι ενδογενείς ( $\mathbf{K}$ ) επιτρέπουν τη χαρτογράφηση μεταξύ συντεταγμένων κάμερας και συντεταγμένων pixel στο επίπεδο εικόνας. Το μοντέλο της κάμερας γενικά είναι μια χαρτογράφηση από τον πραγματικό κόσμο σε συντεταγμένες εικόνας.

Επιπλέον στην όραση υπολογιστών είναι σημαντικό να χρησιμοποιείται το σωστό πλαίσιο αναφοράς, δεδομένου ότι οι μετρήσεις γίνονται σε ένα συγκεκριμένο σύστημα συντεταγμένων. Δεδομένου ότι υπάρχουν πολλά συστήματα συντεταγμένων, όσον αφορά την ανάλυση ενός πραγματικού αντικειμένου, απαιτούνται κατάλληλοι μετασχηματισμοί για τη μετατροπή κάθε συστήματος σε άλλο. Συγκεκριμένα, τα συστήματα είναι τα ακόλουθα [11]:

1. Ομοιογενές σύστημα συντεταγμένων  $(u, v)$ : Μετρά τις θέσεις των pixel στο επίπεδο εικόνας.
2. Σύστημα συντεταγμένων επιπέδου εικόνας  $(x, y)$ : Ένα επίπεδο στο οποίο σχηματίζεται η εικόνα.
3. Σύστημα συντεταγμένων κάμερας  $(X_c, Y_c, Z_c)$ : Ένα σύστημα που χρησιμοποιεί το κέντρο της κάμερας ως κέντρο των αξόνων του.
4. Παγκόσμιο σύστημα συντεταγμένων  $(X_w, Y_w, Z_w)$ : Ένα σταθερό σύστημα συντεταγμένων για την αναπαράσταση αντικειμένων στον κόσμο.

Τα σημεία σε μία εικόνα μπορούν να αναπαρασταθούν τόσο με τη χρήση του συστήματος συντεταγμένων επιπέδου εικόνας όσο και με χρήση του συστήματος ομοιογενών συντεταγμένων. Για τους σκοπούς της παρούσας διπλωματικής επιλέχθηκε το δεύτερο σύστημα συντεταγμένων. Έτσι τα σημεία που ανήκουν στο επίπεδο της εικόνας θα συμβολίζονται ως [20]:

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \quad (1.1)$$

Από την άλλη, τα σημεία που αναφέρονται στον πραγματικό τρισδιάστατο χώρο θα αναφέρονται ως [20]:

$$\mathbf{X} = \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}. \quad (1.2)$$

Από την άλλη, οι ενδογενείς παράμετροι της κάμερας και συγκεκριμένα ο πίνακας  $\mathbf{K}$  [20], που είναι ένας άνω τριγωνικός πίνακας με 3 διαφορετικές παραμέτρους (το εστιακό μήκος, τη λοξότητα και τη μετατόπιση του κύριου σημείου) υπολογίζεται μετά την βαθμονόμηση της κάμερας, συμβολίζεται όπως φαίνεται στην παρακάτω εξίσωση:

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (1.3)$$

Πιο αναλυτικά με  $s$  συμβολίζεται η λοξότητα,  $u_0$  και  $v_0$  είναι οι συντεταγμένες της μετατόπισης του κύριου σημείου, δηλαδή της τοποθεσίας του κύριου σημείου σε σχέση με την προέλευση

του στο πραγματικό σύστημα συντεταγμένων, ενώ  $f_x$  και  $f_y$  είναι το εστιακό μήκος, δηλαδή η απόσταση μεταξύ της οπής και του επιπέδου εικόνας [26]. Για τις ανάγκες των πειραμάτων που εκτελέστηκαν στα πλαίσια της παρούσας διπλωματικής χρησιμοποιήθηκε ο  $\mathbf{K}$  πίνακας με τις τιμές που φαίνονται παρακάτω:

$$\mathbf{K} = \begin{bmatrix} 800 & 0 & 640 \\ 0 & 800 & 480 \\ 0 & 0 & 1 \end{bmatrix},$$

και αναφέρεται σε μία ορθά βαθμονομημένη κάμερα με οπές (pinhole camera). Το  $u_0$  είναι 640 επειδή το πλάτος του επιπέδου εικόνας που χρησιμοποιήσαμε είναι 1280 pixels και το  $v_0$  είναι 480 επειδή το ύψος του επιπέδου εικόνας το ορίσαμε ως 960 pixels.

Παράλληλα οι εξωγενείς παράμετροι της κάμερας εκφράζονται από τον πίνακα  $R_{3 \times 3}$ , που αναφέρεται στην περιστροφή μεταξύ του πλάνου της εικόνας και του πραγματικού χώρου και από τον πίνακα  $t_{3 \times 1}$ , που αναφέρεται στη μετάφραση δηλαδή στις σχετικές θέσεις του πραγματικού πλάνου και του πλάνου εικόνας. Επιπλέον αξιοποιώντας τους δύο παραπάνω πίνακες της περιστροφής και της μετάφρασης, μπορούμε να ορίσουμε την ακόλουθη εξίσωση για τον υπολογισμό του κέντρου της κάμερας ( $\mathbf{C}$ ), εκφραζόμενο στο πραγματικό σύστημα συντεταγμένων [20]:

$$\mathbf{C} = -\mathbf{R}^{-1}\mathbf{t} = -\mathbf{R}^T\mathbf{t}. \quad (1.4)$$

Οι πίνακες της περιστροφής και της μετάφρασης που χρησιμοποιήθηκαν στα πειράματά μας φαίνονται παρακάτω:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -2.3512749 \times 10^{-16} \\ 0 & 2.35127499 \times 10^{-16} & -1 \end{bmatrix},$$

$$\mathbf{t} = \begin{bmatrix} 0 \\ 1.1756375 \times 10^{-16} \\ 0.5 \end{bmatrix}.$$

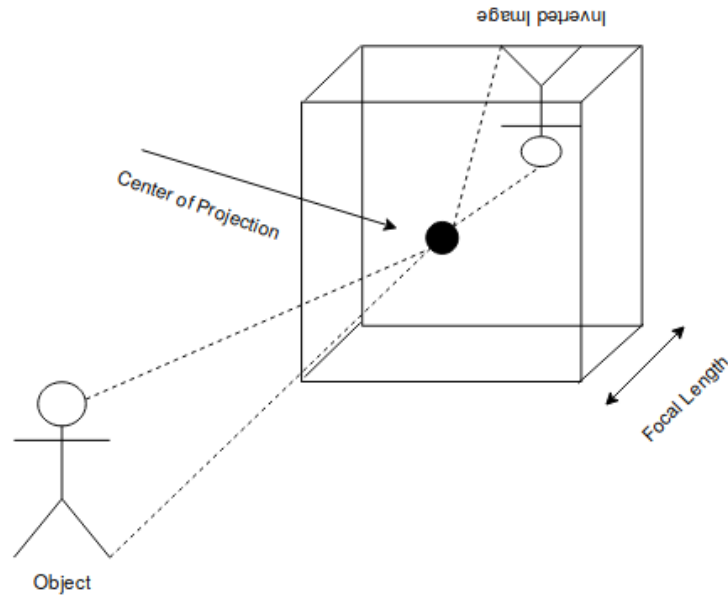
Γνωρίζοντας λοιπόν τις πραγματικές συντεταγμένες ενός αντικειμένου, καθώς και τις εξωγενείς και ενδογενείς παραμέτρους της βαθμονομημένης κάμερας που χρησιμοποιείται σε μία εφαρμογή, είναι δυνατή η έρευνα των σημείων απεικόνισης στο πλάνο της εικόνας εκφραζόμενα σε pixel. Για το σκοπό αυτό ορίζεται ένας νέος πίνακας, ο πίνακας προβολής

$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$  [20] και άρα τα σημεία της εικόνας υπολογίζονται από τη σχέση:

$$\mathbf{x} = \mathbf{P} \mathbf{X}. \quad (1.5)$$

Όλα τα σημεία του πραγματικού συστήματος συντεταγμένων που χρησιμοποιήθηκαν στα πλαίσια της διπλωματικής, επιλέχθηκαν τυχαία μέσα από μία σαφώς ορισμένη σφαίρα με ακτίνα 0.3 μέτρα. Η ακτίνα αυτή επιλέχθηκε, ώστε να διασφαλισθεί ότι κάθε πιθανώς επιλεγόμενο σημείο θα μπορεί να προβληθεί στο πεδίο απεικόνισης που έχει οριοθετημένες διαστάσεις με μήκος 1280 pixel και ύψος 960 pixel.





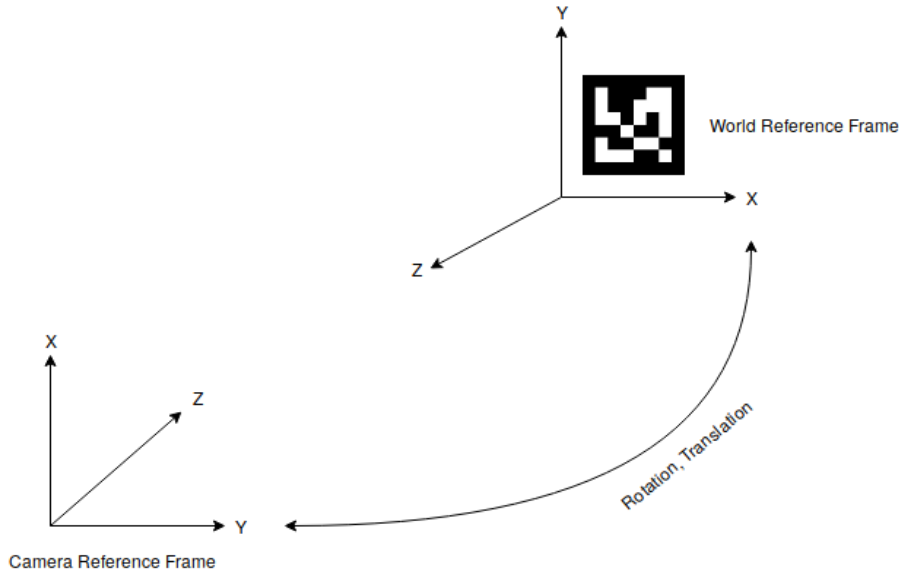
Σχήμα 1.3: Το μοντέλο μίας κάμερας με οπές.

Είναι σημαντικό να αναφερθεί πως η κάμερα που χρησιμοποιήθηκε είναι μία κάμερα με οπές, που έχει σχήμα κλειστού κουτιού με ένα μόνο μικρό άνοιγμα (τρύπα) και λόγω της ευθύγραμμης διάδοσης του φωτός, δημιουργείται μια αντίστροφη εικόνα του εξωτερικού χώρου στην αντίθετη πλευρά του κουτιού. Η διάμετρος της οπής εξαρτάται από το εστιακό μήκος και το μήκος κύματος του φωτός.

Η κάμερα αυτή είναι βαθμονομημένη, δηλαδή έχει περάσει μία γεωμετρική διεργασία που υπολογίζει τις παραμέτρους του φακού και του αισθητήρα εικόνας μιας κάμερας, ώστε να επιτευχθεί η ελαχιστοποίηση της παραμόρφωσης του φακού και ο προσδιορισμός της θέσης της κάμερας στο χώρο [12]. Η λειτουργία μίας κάμερας με οπές προσομοιώνεται στο Σχήμα 1.3. Έχοντας αναφέρει όλες τις αναγκαίες παραμέτρους της κάμερας, είναι σημαντικό να γίνει αναφορά σε μία σημαντική πρόκληση τόσο στην όραση υπολογιστών όσο και στη ρομποτική, που αποτελεί και το κύριο αντικείμενο μελέτης της διπλωματικής αυτής.

Η πρόκληση αυτή είναι ο εντοπισμός αντικειμένων σε μια εικόνα σε σχέση με τον προσανατολισμό τους στον πραγματικό κόσμο (μετατροπή από συντεταγμένες εικόνας σε συντεταγμένες πραγματικού κόσμου). Όσον αφορά την εκτίμηση της θέσης, τόσο η ακριβής θέση όσο και ο προσανατολισμός ενός αντικειμένου πρέπει να προσδιοριστούν, καθώς αυτές οι πληροφορίες χρησιμοποιούνται για την πλοήγηση ενός ρομπότ, όπως είναι για παράδειγμα ένα χειρουργικό ρομπότ [11].

Παράλληλα τα δεδομένα μίας εικόνας μπορούν να αντλούνται είτε από μια μεμονωμένη εικόνα, είτε από ένα ζεύγος ή μια ακολουθία εικόνων. Σε αυτή τη διπλωματική, αποφασίσαμε να χρησιμοποιήσουμε μια μόνο εικόνα που τραβάται από μία μόνο κάμερα. Επιπλέον, τα αντικείμενα του πραγματικού κόσμου μπορεί να αφορούν οτιδήποτε, από έναν ολόκληρο άνθρωπο



Σχήμα 1.4: Εκτίμησης θέσης της κάμερας.

μέχρι απλώς σε ένα χέρι ή ακόμα και σε ένα δάχτυλο.

Ακόμη η θέση της κάμερας στο χώρο μετρείται με τη χρήση δύο διαφορετικών πινάκων, αυτών της περιστροφής και της μετάφρασης που αναφέρθηκαν και παραπάνω και φαίνονται και στο Σχήμα 1.4. Για την εκτίμηση της θέσης της κάμερας μπορούν να χρησιμοποιηθούν πολλοί αλγόριθμοι. Στα πλαίσια αυτής της διπλωματικής επιλέξαμε να χρησιμοποιήσουμε τον Άμεσο Γραμμικό Μετασχηματισμό Direct Linear Transformation.

## 1.7 Ο Αλγόριθμος Direct Linear Transformation

Ο Direct Linear Transformation είναι ένας αλγόριθμος που καθορίζει τον προσανατολισμό (περιστροφή και μετάφραση) μιας μεμονωμένης εικόνας χωρίς να γνωρίζει εκ των προτέρων τις παραμέτρους της κάμερας. Για πειράματα που αφορούν στις 2 διαστάσεις, ο αλγόριθμος απαιτεί τουλάχιστον 4 σημεία για να λειτουργήσει, ενώ σε τρισδιάστατες εφαρμογές απαιτεί τουλάχιστον 6 σημεία, όπως σαφώς καθορίζεται από τον αλγόριθμο "Gold Standard", που ανέπτυξαν οι Hartley και Zisserman [20].

Στην περίπτωση μας αναφερόμαστε σε τρισδιάστατο χώρο οπότε καταλήγουμε στα εξής 6 ζευγάρια εξισώσεων:

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \quad (1.6)$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \quad (1.7)$$

Οι εξισώσεις (1.6) και (1.7) μας οδηγούν στο ακόλουθο ομοιογενές σύνολο εξισώσεων:

$$\underbrace{\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_i X_i & -u_i Y_i & -u_i Z_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_i X_i & -v_i Y_i & -v_i Z_i & -v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{10} \\ m_{11} \end{bmatrix}}_{\mathbf{m}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \quad (1.8)$$

Για να λύσουμε αυτό το σύστημα εξισώσεων εκμεταλλευόμαστε την Αποσύνθεση Μοναδικής Αξίας (Singular Value Decomposition) [2] του πίνακα  $\mathbf{A}$  και βρίσκουμε τον πίνακα  $\mathbf{m}$  (12x1), ως την τελευταία στήλη του διανύσματος  $\mathbf{V}$  της αποσύνθεσης. Έπειτα τροποποιούμε καταλλήλως το μέγεθος του  $\mathbf{m}$ , για να βρούμε τον τελικό  $H_{\text{DLT}}$ , από τον οποίο και θα εξάγουμε τελικά τους εκτιμώμενους πίνακες για την περιστροφή και τη μετάφραση, οι οποίοι και θα συμβολίζονται ως  $\hat{\mathbf{R}}$  και  $\hat{\mathbf{t}}$ .

Είναι σημαντικό πως στα σημεία της εικόνας προστέθηκε και τυχαίος θόρυβος για την εξαγωγή πιο ρεαλιστικών συμπερασμάτων, για αυτό και απαιτείται να προηγηθεί κανονικοποίηση στα δεδομένα πριν την εφαρμογή του Direct Linear Transformation. Σύμφωνα με το [20], τα τρισεδιάστατα σημεία κανονικοποιούνται ώστε το κέντρο τους να βρίσκεται στην αρχή μέτρησης των αξόνων και η τετραγωνική ρίζα της διακύμανσής τους να είναι ίση με  $\sqrt{3}$ .

Συνεπώς ο πίνακας  $\hat{\mathbf{P}}$  μπορεί να βρεθεί από τις 3 πρώτες στήλες του τελικού  $H_{\text{DLT}}$  [27]. Έπειτα γνωρίζοντας εκ των προτέρων τον πίνακα  $\mathbf{K}$  από την βαθμονόμηση της κάμερας, με τη χρήση της παραγοντοποίησης QR [39] καταλήγουμε την εκτίμηση του πίνακα της περιστροφής. Έτσι, ο πίνακας  $H_{\text{DLT}}$  αναλύεται ως:

$$H_{\text{DLT},1:3} = \mathbf{K}\hat{\mathbf{R}} \implies H_{\text{DLT},1:3}^T = (\mathbf{K}\hat{\mathbf{R}})^T = \hat{\mathbf{R}}^T \mathbf{K}^T$$

Ακόμη, η αποσύνθεση QR υλοποιείται ως εξής:  $\hat{\mathbf{R}}^T \mathbf{K}^T = \mathbf{Q}\mathbf{B}$ , όπου  $\mathbf{Q}$  είναι ο ανάστροφος πίνακας της εκτιμώμενης περιστροφής, που μας ενδιαφέρει να εξάγουμε. Συνεπώς, ο τελικός πίνακας της εκτιμώμενης περιστροφής δίνεται από την εξίσωση:

$$\hat{\mathbf{R}} = \mathbf{Q}^T. \quad (1.9)$$

Έπειτα μας ενδιαφέρει η εκτίμηση του πίνακα περιστροφής, που για να υπολογιστεί σωστά απαιτείται η εύρεση του κέντρου της κάμερας. Σύμφωνα με το [8], η σχέση για να υπολογιστεί το κέντρο, προκύπτει αν αναλύσουμε τον πίνακα που λάβαμε από τον Direct Linear Transformation Algorithm, ως εξής  $H_{\text{DLT}} = [\mathbf{Q} \mid \mathbf{b}]$ , και άρα το κέντρο της κάμερας προκύπτει όπως φαίνεται παρακάτω:

$$\mathbf{C} = \begin{bmatrix} -\mathbf{Q}^{-1}\mathbf{b} \\ 1 \end{bmatrix}. \quad (1.10)$$

Έπειτα αξιοποιώντας την εξίσωση (1.4) και τον εκτιμώμενο πίνακα περιστροφής που υπολογίσαμε παραπάνω, προκύπτει ο εκτιμώμενος πίνακας μετάφρασης  $\hat{\mathbf{t}}$ .

## 1.8 Υπολογίζοντας τα Σφάλματα στην εκτίμηση θέσης

Η εφαρμογή του Direct Linear Transformation αλγορίθμου, οδήγησε στην εξαγωγή δύο πινάκων για την εκτιμώμενη περιστροφή και μετάφραση. Έτσι συγκρίνοντας αυτούς τους πίνακες με τους πραγματικούς μπορούμε να υπολογίσουμε τα σχετικά σφάλματα που προκύπτουν. Αρχικά ορίζεται ως Σφάλμα Περιστροφής (RE) [21], η ελάχιστη περιστροφή που απαιτείται για την ευθυγράμμιση του εκτιμώμενου και του πραγματικού πίνακα περιστροφής, που εκφράζεται σε μοίρες, όπως προκύπτει από τη σχέση:

$$RE = \hat{\mathbf{R}}\mathbf{R}^{-1}, \quad (1.11)$$

και η τελική γωνία με τον άξονα σε μοίρες δίνεται από τη σχέση:

$$\text{axis-angle} = \arccos\left(\frac{\text{trace}(RE) - 1}{2}\right). \quad (1.12)$$

Στη συνέχεια ορίζεται ως Σφάλμα Μετάφρασης (TE) το επί τοις εκατό σφάλμα μεταξύ εκτιμώμενου και πραγματικού πίνακα μετάφρασης όπως δίνεται από τη σχέση:

$$TE = \|\hat{\mathbf{t}} - \mathbf{t}\|_2 / \|\mathbf{t}\|_2 \cdot 100\% \quad (1.13)$$

Άλλα Σφάλματα που χρειάστηκε να υπολογιστούν για τους σκοπούς αυτής της διπλωματικής αναλύονται στη συνέχεια. Πρώτον, το μέσο τετραγωνικό σφάλμα (RMS), είναι ένα απόλυτο μέτρο σφάλματος και υπολογίζει το σφάλμα σε σχέση με μια γνωστή τιμή ground truth [6]. Η σχέση από την οποία εξάγεται είναι η παρακάτω:

$$e_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|^2}, \quad (1.14)$$

όπου ως  $y_i$  εκφράζεται η παρατηρούμενη τιμή και ως  $\hat{y}_i$  η εκτιμώμενη τιμή σε κάθε  $i$  παρατήρηση, ενώ το Σφάλμα μετριέται στις ίδιες μονάδες και κλίμακα με τις τιμές αυτές.

Δεύτερον το Σφάλμα Επαναπροβολής είναι ένα γεωμετρικό σφάλμα που αντιστοιχεί στην απόσταση της εικόνας μεταξύ ενός εκτιμώμενου προβαλλόμενου σημείου και των συντεταγμένων της πραγματικής προβολής του [20]. Υπολογίζεται ως η Ευκλείδεια απόσταση μεταξύ των πραγματικών συντεταγμένων  $\mathbf{x}$  και  $\mathbf{y}$  της εικόνας και των εκτιμώμενων συντεταγμένων  $\hat{\mathbf{x}}$  και  $\hat{\mathbf{y}}$ . Οι πραγματικές συντεταγμένες της εικόνας βρίσκονται από την εξίσωση (1.1), ενώ αντίστοιχα οι εκτιμώμενες συντεταγμένες βρίσκονται από την ίδια εξίσωση αλλά θεωρώντας ως πίνακα προβολής τον:  $\hat{\mathbf{P}} = \mathbf{K}[\hat{\mathbf{R}}|\hat{\mathbf{t}}]$ . Έπειτα πρέπει να τροποποιηθεί κατάλληλα ώστε να δίνει ως αποτέλεσμα μέτρα αντί για pixel. Έτσι για κάθε ένα από τα σημεία που χρησιμοποιήθηκαν στον αλγόριθμο, υπολογίζεται η Ευκλείδεια απόσταση με τον εξής τύπο:

$$\text{Distance} = \sum_{i=1}^n \sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2}, \quad (1.15)$$

όπου  $n$  είναι το σύνολο των σημείων, ενώ αν δεν προστεθεί σε αυτά θόρυβος το σφάλμα αυτό είναι οριακά μηδενικό.

Ένα ακόμα πολύ σημαντικό μέγεθος στην εκτίμηση σφάλματος είναι ο αριθμός συνθήκης πίνακα (condition number), καθώς υπολογίζει πόσο ευαίσθητη είναι η έξοδος στις διαταραχές που εφαρμόζονται στην είσοδο [31]. Έτσι ο αριθμός αυτός μας δείχνει πόσο ευαίσθητος είναι ένας πίνακας σε μικρές αλλαγές στην είσοδό του. Ο αριθμός αυτός προκύπτει για έναν πίνακα  $\mathbf{A}$  ως:

$$\text{condition number} = \|A^{-1}\| * \|\mathbf{A}\| \quad (1.16)$$

Η συνάρτηση ακρίβειας εξαρτάται γενικά από τις εξωγενείς και ενδογενείς παραμέτρους της κάμερας, τον ίδιο το βασικό δείκτη, τη στάση της κάμερας και επίσης τον αριθμό συνθήκης, όπως φαίνεται παρακάτω:

Accuracy = f(K, R, t, marker size, condition number).

Τέλος η μέτρηση της συνδιακύμανσης είναι μία ακόμα σημαντική παράμετρος που μας δείχνει την κατεύθυνση της γραμμικής σχέσης μεταξύ μεταβλητών. Ο πίνακας συνδιακύμανσης είναι ένας τετραγωνικός πίνακας, που χρησιμοποιείται συχνά στη θεωρία και στα στατιστικά στοιχεία πιθανότητας, που δίνεται από την εξίσωση:

$$\Sigma_B = \frac{1}{n-1} \sum_{i=1}^n (B_i - \bar{B}_i)(B_i - \bar{B}_i)^T, \quad (1.17)$$

όπου ο  $B_i$  είναι μία ομάδα δεδομένων που μελετάμε και  $\bar{B}_i$  είναι ο μέσος όρος αυτού του συνόλου δεδομένων [23]. Ο πίνακας αυτός χρησιμοποιείται και για την εκτίμηση των ελλειψοειδών του σφάλματος συνδιακύμανσης. Ένα ελλειψοειδές εμπιστοσύνης επιτρέπει την απεικόνιση ενός επιλεγμένου επιπέδου εμπιστοσύνης. Γενικά ένα ελλειψοειδές ορίζεται από την ακόλουθη εξίσωση:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \quad (1.18)$$

που οδηγεί στην εξίσωση του ελλειψοειδούς συνδιακύμανσης:

$$\frac{x^2}{\lambda_x} + \frac{y^2}{\lambda_y} + \frac{z^2}{\lambda_z} = sc^2 \implies \frac{x^2}{sc\sqrt{\lambda_x}} + \frac{y^2}{sc\sqrt{\lambda_y}} + \frac{z^2}{sc\sqrt{\lambda_z}} = 1, \quad (1.19)$$

με  $\lambda_x$ ,  $\lambda_y$  ανδ  $\lambda_z$  να εκφράζουν τις ιδιοτιμές του πίνακα συνδιακύμανσης και  $sc$  να δηλώνει την κλίμακα του επιπέδου εμπιστοσύνης [6], και παρουσιάζεται στον Πίνακα 1.1.

Confidence Level						Dimensions
25 %	50%	75%	95%	97 %	99%	
0.10153	0.45494	1.3233	3.84146	4.70929	6.6349	n = 1
0.57536	1.38629	2.77259	5.99146	7.01312	9.2103	n = 2
1.21253	2.36597	4.10834	7.81473	8.94729	11.3449	n = 3

Πίνακας 1.1: Σχέση μεταξύ  $sc$  και επιπέδου εμπιστοσύνης σε 1,2 και 3 διαστάσεις.

Τέλος οι βασικοί άξονες του ελλειψοειδούς έχουν μήκος, το οποίο δίνεται από τη σχέση:

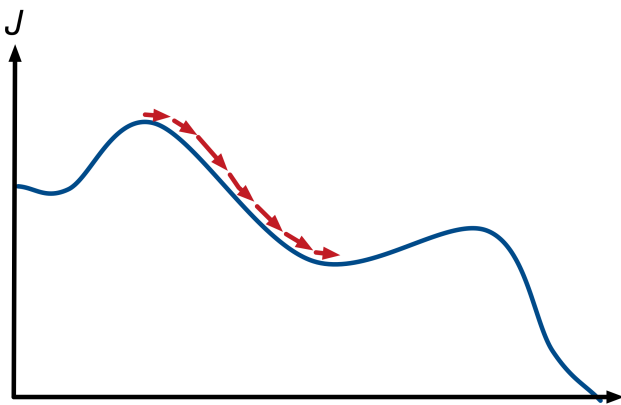
$$l_i = sc\sqrt{\lambda_i}, \quad \forall i = 1, 2, 3 \quad (1.20)$$

## 1.9 Αλγόριθμοι Βελτιστοποίησης

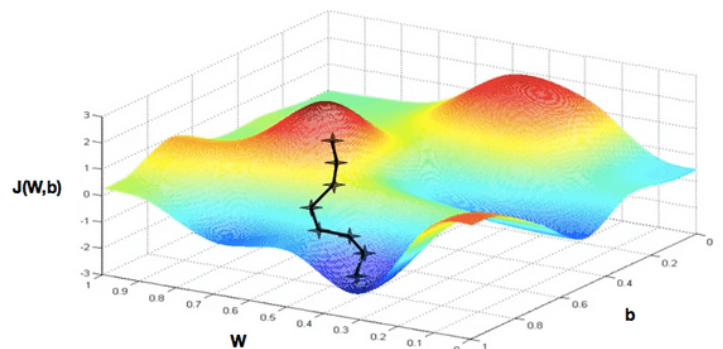
Για την εύρεση των διατάξεων των βασικών δεικτών που οδηγούν στα μικρότερα Σφάλματα, όπως ορίστηκαν παραπάνω, χρησιμοποιήθηκαν 2 διαφορετικοί Αλγόριθμοι Βελτιστοποίησης. Ο πρώτος λέγεται Gradient Descent και είναι ένας αλγόριθμος επαναληπτικής βελτιστοποίησης πρώτης τάξης, ο οποίος επιτρέπει τον υπολογισμό των τοπικών ελάχιστων σημείων μιας συνάρτησης [36]. Αρχικά επιλέγεται ένα τυχαίο σημείο της συνάρτησης και η αξιολόγηση των διαβαθμίσεων αρχίζει ακολουθώντας το αρνητικό της διαβάθμισης της συνάρτησης έως ότου επιτευχθεί ένα τοπικό ή ένα ολικό ελάχιστο.

Σε κάθε αξιολόγηση (επανάληψη), υπάρχει ένα συγκεκριμένο ποσοστό, που ονομάζεται ρυθμός εκμάθησης, το οποίο ορίζεται ως το μέγεθος κάθε βήματος. Εάν αυτός ο ρυθμός είναι μικρός, προχωράμε προς την κατεύθυνση της αρνητικής κλίσης. Επιπλέον, ορίζεται και μια συνάρτηση κόστους, η οποία δείχνει την ακρίβεια της συνάρτησης στις προβλέψεις. Χρησιμοποιείται συχνά στη μηχανική μάθηση για την ενημέρωση των παραμέτρων ενός μοντέλου [10].

Στο Σχήμα 1.5α' υπάρχει ένα παράδειγμα της εφαρμογής του αλγορίθμου Gradient Descent σε ένα διδιάστατο πρόβλημα. Σε αυτή την περίπτωση ο ρυθμός εκμάθησης εξαρτάται από τους άξονες  $x$  και  $y$  και το επιλεγμένο σημείο αλλάζει σε κάθε επανάληψη έως ότου δεν προκληθούν αλλαγές, πράγμα που σημαίνει ότι υπάρχει τοπικό ή ολικό ελάχιστο. Επιπλέον όπως μπορούμε να δούμε από το μήκος του διανύσματος βημάτων, αλλάζει ανάλογα με το ρυθμό εκμάθησης. Από την άλλη, όπως μπορούμε να δούμε στο Σχήμα 1.5β', που αναφέρεται στην εφαρμογή του αλγορίθμου σε τριδιάστατο πρόβλημα, όπως αυτά που εξετάζονται στην παρούσα διπλωματική ο ρυθμός εκμάθησης, εξαρτάται από τους άξονες  $x$ ,  $y$  και  $z$ .



(α') Ο Αλγόριθμος Gradient Descent σε 2 διαστάσεις [4].



(β') Ο Αλγόριθμος Gradient Descent σε 3 διαστάσεις [10].

Σχήμα 1.5: Ο Αλγόριθμος Βελτιστοποίησης Gradient Descent.

Ο δεύτερος αλγόριθμος βελτιστοποίησης που χρησιμοποιήθηκε είναι ο Adam Optimizer, που αποτελεί επέκταση της στοχαστικής καθόδου διαβάθμισης, και μπορεί να χρησιμοποιηθεί αντί της κλασικής διαδικασίας στοχαστικής καθόδου κλίσης για την ενημέρωση επαναληπτικών βαρών δικτύου με βάση τα εκπαιδευτικά δεδομένα, για παράδειγμα ρυθμός εκπαίδευσης [25].

Ο αλγόριθμος Adam έχει τέσσερις παραμέτρους διαμόρφωσης:

- alpha: Ρυθμός εκμάθησης ή μέγεθος βήματος
- beta1: Εκθετικός ρυθμός αποσύνθεσης για τις εκτιμήσεις της πρώτης στιγμής
- beta2: Εκθετικός ρυθμός αποσύνθεσης για τις εκτιμήσεις της δεύτερης στιγμής
- epsilon: Μικρή τιμή που αποτρέπει τη διαίρεση με μηδέν

Αξίζει να σημειωθεί ότι στον αλγόριθμο Adam, απαιτούνται τόσο το beta1 όσο και το beta2, καθώς ο Adam, αντί να προσαρμόζει τα ποσοστά εκμάθησης των παραμέτρων με βάση το μέσο όρο της πρώτης στιγμής, κάνει χρήση του μέσου όρου των δευτέρων στιγμών των διαβαθμίσεων [9]. Ακόμη ο ρυθμός εκμάθησης χρησιμοποιείται επίσης συχνά στον Adam, ο οποίος διατηρείται για κάθε παράμετρο και προσαρμόζεται ξεχωριστά καθώς η διαδικασία εκπαίδευσης διεκπεραιώνεται.

## 1.10 Γωνίες του Euler

Οι γωνίες Euler είναι τρεις γωνίες, οι οποίες περιγράφουν τον προσανατολισμό ενός άκαμπτου σώματος σε σχέση με ένα σταθερό σύστημα συντεταγμένων. Χρησιμοποιούνται επίσης για να αντιπροσωπεύσουν τον προσανατολισμό μιας γενικής βάσης στη γραμμική άλγεβρα.

Αυτές οι τρεις γωνίες εντοπίζονται από τον αρχικό πίνακα της περιστροφής  $\mathbf{R}$ , όπως ορίστηκε παραπάνω, ως εξής [6]:

$$a = \arctan \frac{R_{3,2}}{R_{3,3}}, \quad (1.21)$$

$$b = -\arcsin R_{3,1}, \quad (1.22)$$

$$c = \arctan \frac{R_{2,1}}{R_{1,1}}. \quad (1.23)$$

Αυτές οι γωνίες χρησιμοποιούνται για την εύρεση του πίνακα περιστροφής που εκφράζεται από τις γωνίες Euler όπως φαίνεται παρακάτω:

$$\mathbf{R}_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{bmatrix}, \quad (1.24)$$

$$\mathbf{R}_b = \begin{bmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{bmatrix}, \quad (1.25)$$

$$\mathbf{R}_c = \begin{bmatrix} \cos c & -\sin c & 0 \\ \sin c & \cos c & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1.26)$$

Χρησιμοποιώντας τους πίνακες  $\mathbf{R}_a$ ,  $\mathbf{R}_b$ , και  $\mathbf{R}_c$ , καταλήγουμε στον Euler Rotation πίνακα:

$$\mathbf{R}_{\text{Euler}} = \mathbf{R}_c \mathbf{R}_b \mathbf{R}_a. \quad (1.27)$$

## 1.11 Αναλύοντας το πρόβλημα εκτίμησης θέσης σε σφαιρικές συντεταγμένες

Αρχικά οι καρτεσιανές συντεταγμένες που αναφέρονται στους  $x$ ,  $y$ ,  $z$  άξονες μπορούν να εκφραστούν με σφαιρικές συντεταγμένες αξιοποιώντας τις παρακάτω σχέσεις:

$$\alpha = \arccos\left(\frac{z}{r}\right), \quad (1.28)$$

$$\beta = \arctan\left(\frac{y}{x}\right), \quad (1.29)$$

$$r^2 = x^2 + y^2 + z^2, \quad (1.30)$$

όπου το  $r$  αντιπροσωπεύει την ακτίνα της σφαίρας ή αλλιώς λέγεται ακτινική απόσταση,  $\alpha$  ονομάζεται κλίση ή πολική γωνία και περιορίζεται από  $-90$  έως  $90$  μοίρες. Τέλος,  $\beta$  είναι το αζιμούθιο και περιορίζεται από  $0$  έως  $360$  μοίρες. Έτσι λοιπόν σε αυτή την ενότητα θα αναλύσουμε τους πίνακες περιστροφής  $\mathbf{R}$  και μετάφρασης  $\mathbf{t}$  σε σφαιρικές συντεταγμένες. Ο στόχος είναι να μελετηθεί η θέση ή οι θέσεις της κάμερας που οδηγούν στα μικρότερα σφάλματα, καθώς και το πως η απόσταση της κάμερας από το βασικό δείκτη μπορεί να επιδράσει στην ακρίβεια εκτίμησης θέσης.

Για να βρούμε τον πίνακα συνδιακύμανσης, για την κάμερα που περιγράφηκε παραπάνω, υποθέτουμε ότι όλες οι εξωγενείς και ενδογενείς παράμετροι της κάμερας είναι γνωστές και καταγράφηκαν στην αντίστοιχη ενότητα και ορίζονται στην τάξη μας χωρίς σφάλματα και ο βασικός δείκτης που μελετάμε είναι τοποθετημένος σε σταθερό σημείο, ενώ η κάμερα κινείται σε μια σφαίρα με συγκεκριμένη ακτίνα  $r$ ,  $0.3$  μέτρα.

Τώρα ορίζουμε 12 μη γραμμικές εξισώσεις κάμερας με βάση την εξίσωση ( $;$ ), καθώς ο αλγόριθμος Direct Linear Transformation χρειάζεται τουλάχιστον 6 σημεία για να λειτουργήσει σωστά [6]:

$$q : \begin{aligned} u_i &= \frac{1}{\rho_i} \mathbf{P}x_i \\ &\vdots \\ v_i &= \frac{1}{\rho_i} \mathbf{P}y_i \quad \forall i = 1, 2, \dots, 6 \\ &\vdots \end{aligned} \quad (1.31)$$

Στη συνέχεια, προστίθεται τυχαίος Γκαουσιανός θόρυβος, με τυπική απόκλιση 4 και μέση τιμή 0, σε κάθε συντεταγμένη  $x_i$ ,  $y_i$ , σε αυτές τις μη γραμμικές εξισώσεις.

Έτσι, σύμφωνα με το [6], χωρίς απώλεια γενικότητας παραμετροποιούμε τις εξωγενείς παραμέτρους της κάμερας χρησιμοποιώντας τρεις παραμέτρους  $\alpha$ ,  $\beta$  και  $r$ , αντιπροσωπεύοντας τη στάση της κάμερας που σχετίζεται με το βασικό δείκτη σε σφαιρικές συντεταγμένες, όπως φαίνεται στο σχήμα  $;$ :

Στη συνέχεια, τα  $\alpha$ ,  $\beta$  και  $r$  υπολογίζονται μετατρέποντας την Καρτεσιανή σε σφαιρικές συντεταγμένες, χρησιμοποιώντας τις εξισώσεις (1.30), όπου  $x$ ,  $y$ ,  $z$  καθορίζονται από την θέση της κάμερας στο πραγματικό σύστημα συντεταγμένων.

Έπειτα, χρησιμοποιούμε τη μήτρα κάμερας  $\mathbf{K}$ , και οι πίνακες  $\mathbf{R}$ ,  $\mathbf{t}$ , μετατρέπονται σε σφαιρικές συντεταγμένες. Στη συνέχεια, υπολογίζονται οι παράγωγοι των  $\alpha$ ,  $\beta$ ,  $r$ , για να



χρησιμοποιηθούν για τη μέτρηση των πινάκων περιστροφής και μετάφρασης για να βρούμε τον Ιακωβιανό πίνακα του  $q$ , όπως ορίζεται στις εξισώσεις (1.31). Για να βρούμε τα συγκεκριμένα  $\alpha_0, \beta_0, r_0$ , όπου θα αξιολογηθεί ο Ιακωβιανός πίνακας, πρέπει να υπολογίσουμε τα  $x_0, y_0, z_0$  εκτιμώντας τη θέση της κάμερας σε πραγματικές συντεταγμένες.

Ως εκ τούτου, πρέπει να βρούμε το γινόμενο του  $\mathbf{Rt}$  και του διανύσματος  $[0, 0, 0, 1]^T$ . Στη συνέχεια,  $x_0, y_0, z_0$  είναι οι σειρές του γινομένου και χρησιμοποιώντας την εξίσωση (1.30) τα  $\alpha_0, \beta_0, r_0$ , μπορούν να υπολογιστούν.

Στη συνέχεια, ο Ιακωβιανός πίνακας του  $q$  στα  $\alpha_0, \beta_0, r_0$  εκτιμάται ως:

$$\mathbf{J}_q = \left. \frac{\partial q}{\partial(\alpha, \beta, r)} \right|_{\alpha=\alpha_0, \beta=\beta_0, r=r_0} \in \mathbb{R}^{2n \times 3}, \quad (1.32)$$

όπου  $n$  είναι ο αριθμός των επιλεγμένων σημείων και άρα στην περίπτωση του Ιακωβιανού πίνακα που μελετάμε θα έχει διαστάσεις  $6n \times 3$ . Για να βρούμε τα παράγωγα του  $q$ , πρέπει να υπολογίσουμε ξεχωριστά καθεμία από τις 12 μη γραμμικές εξισώσεις. Σε κάθε εξίσωση μόνο ο πίνακας  $\mathbf{P}$  επηρεάζεται από τα  $\alpha, \beta$  και  $r$ , οπότε είναι ο μόνος πίνακας που πρέπει να αναλύσουμε και να βρούμε τα παράγωγά του.

Το  $\mathbf{P}$  αναλύεται ως:  $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ . Ωστόσο, το ματριξ τεξτβφ  $\mathbf{K}$  δεν εξαρτάται από τις παραμέτρους  $\alpha, \beta$  και  $r$ . Έτσι, για να βρούμε τον Ιακωβιανό του  $q$ , απαιτείται μόνο ο υπολογισμός των παραγώγων του πίνακα περιστροφής και του πίνακα μετάφρασης.

Έτσι θέτουμε  $[(\mathbf{R})|(\mathbf{t})] = (\mathbf{Rt})$ , και η εξίσωση (1.31) αναλύεται τώρα ως:

$$u_i = \frac{1}{\rho_i} \mathbf{K}(\mathbf{Rt})x_i, \quad \forall i = 1, 2, \dots, 6, \quad (1.33)$$

$$v_i = \frac{1}{\rho_i} \mathbf{K}(\mathbf{Rt})y_i, \quad \forall i = 1, 2, \dots, 6. \quad (1.34)$$

Στη συνέχεια, για να βρούμε τον τελικό Ιακωβιανό πίνακα πρέπει να υπολογίσουμε τις παραγώγους μόνο για το  $(\mathbf{Rt})$  των εξισώσεων (1.33) και (1.34), όπως φαίνεται παρακάτω:

$$\frac{\partial u_i}{\partial(\alpha, \beta, r)} = \frac{1}{\rho_i} \mathbf{K} \frac{\partial(\mathbf{Rt})_{\text{spherical}}}{\partial(\alpha, \beta, r)} x_i, \quad (1.35)$$

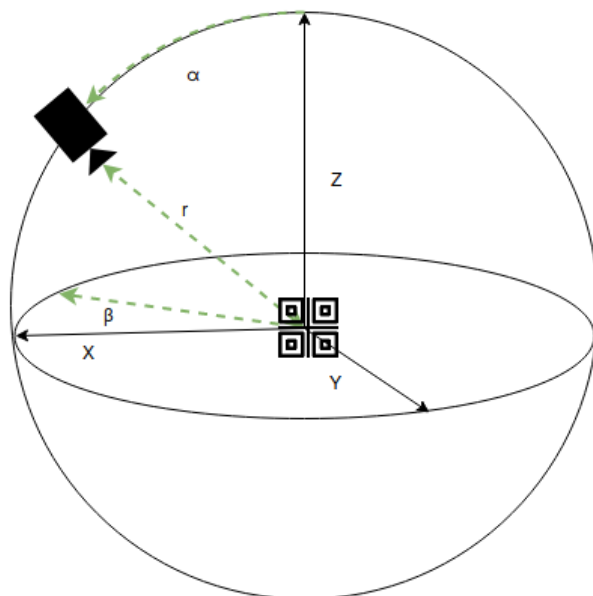
$$\frac{\partial v_i}{\partial(\alpha, \beta, r)} = \frac{1}{\rho_i} \mathbf{K} \frac{\partial(\mathbf{Rt})}{\partial(\alpha, \beta, r)} y_i. \quad (1.36)$$

Ο τελικός πίνακας  $(\mathbf{Rt})$  θα έχει την ακόλουθη μορφή:

$$(\mathbf{Rt}) = \left[ \underline{(\mathbf{R})}_{3 \times 3} \mid \underline{(\mathbf{t})}_{3 \times 1} \right]$$

Έπειτα το πρόβλημα μπορεί να προσεγγιστεί εκτιμώντας μόνο τον  $(\mathbf{R})$  και τον  $(\mathbf{t})$ . Αρχικά ο  $(\mathbf{R})$  είναι το γινόμενο του  $R_{\text{Euler}}$ , που ορίστηκε στην εξίσωση (1.27), και του πίνακα περιστροφής  $\mathbf{R}_s$ , όπως φαίνεται παρακάτω:

$$\mathbf{R}_s = \begin{bmatrix} \cos \beta \cos \alpha & -\sin \beta & \cos \beta \sin \alpha \\ \sin \beta \cos \alpha & \cos \beta & \sin \beta \sin \alpha \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (1.37)$$



Σχήμα 1.6: Η πλοήγηση της κάμερας σε σφαιρικές συντεταγμένες.

και οι ζητούμενες παράγωγοι είναι:

$$\frac{\partial \mathbf{R}_s}{\partial \alpha} = \begin{bmatrix} -\cos \beta \sin \alpha & 0 & \cos \beta \cos \alpha \\ -\sin \beta \sin \alpha & 0 & \sin \beta \cos \alpha \\ \cos \alpha & 0 & -\sin \alpha \end{bmatrix}, \quad (1.38)$$

$$\frac{\partial \mathbf{R}_s}{\partial \beta} = \begin{bmatrix} -\sin \beta \cos \alpha & -\cos \beta & -\sin \beta \sin \alpha \\ \cos \beta \cos \alpha & -\sin \beta & \cos \beta \sin \alpha \\ 0 & 0 & 0 \end{bmatrix}, \quad (1.39)$$

$$\frac{\partial \mathbf{R}_s}{\partial r} = 0 \quad (1.40)$$

Οπότε και ο πίνακας περιστροφής σφαιρικών συντεταγμένων  $\mathbf{t}_{\text{spherical}}$ , δίνεται από την ακόλουθη εξίσωση:

$$(\mathbf{t}) = \mathbf{R}_s \left( - \begin{bmatrix} r \sin \alpha \cos \beta \\ r \sin \beta \sin \alpha \\ r \cos \alpha \end{bmatrix} \right) \quad (1.41)$$

Οι αναγκαίες παράγωγοι της (1.41), βρίσκονται χρησιμοποιώντας τις (1.38), (1.39), (1.40)

ως εξής:

$$\frac{\partial(\underline{\mathbf{t}})}{\partial\alpha} = \frac{\partial\mathbf{R}_s}{\partial\alpha} \begin{pmatrix} -r \sin\alpha \cos\beta \\ -r \sin\beta \sin\alpha \\ r \cos\alpha \end{pmatrix} + \mathbf{R}_s \begin{pmatrix} r \cos\alpha \cos\beta \\ -r \sin\beta \cos\alpha \\ -r \sin\alpha \end{pmatrix} \quad (1.42)$$

$$\frac{\partial(\underline{\mathbf{t}})}{\partial\beta} = \frac{\partial\mathbf{R}_s}{\partial\beta} \begin{pmatrix} -r \sin\alpha \cos\beta \\ -r \sin\beta \sin\alpha \\ r \cos\alpha \end{pmatrix} + \mathbf{R}_s \begin{pmatrix} -r \sin\alpha \sin\beta \\ r \cos\beta \sin\alpha \\ 0 \end{pmatrix} \quad (1.43)$$

$$\frac{\partial(\underline{\mathbf{t}})}{\partial r} = -\mathbf{R}_s \begin{pmatrix} \sin\alpha \cos\beta \\ \sin\beta \sin\alpha \\ \cos\alpha \end{pmatrix}. \quad (1.44)$$

Βρίσκουμε έτσι τον τελικό Ιακωβιανό πίνακα  $\mathbf{J}_q$ , τον οποίο και χρησιμοποιούμε για να βρούμε τον πίνακα συνδιακύμανσης για τον εντοπισμό του βασικού δείκτη. Εν συνεχεία, χρησιμοποιώντας Backward propagation καταλήγουμε στην παρακάτω σχέση:

$$\Sigma = (\mathbf{J}_q^T \Sigma_q^{-1} \mathbf{J}_q)^{-1} \quad (1.45)$$

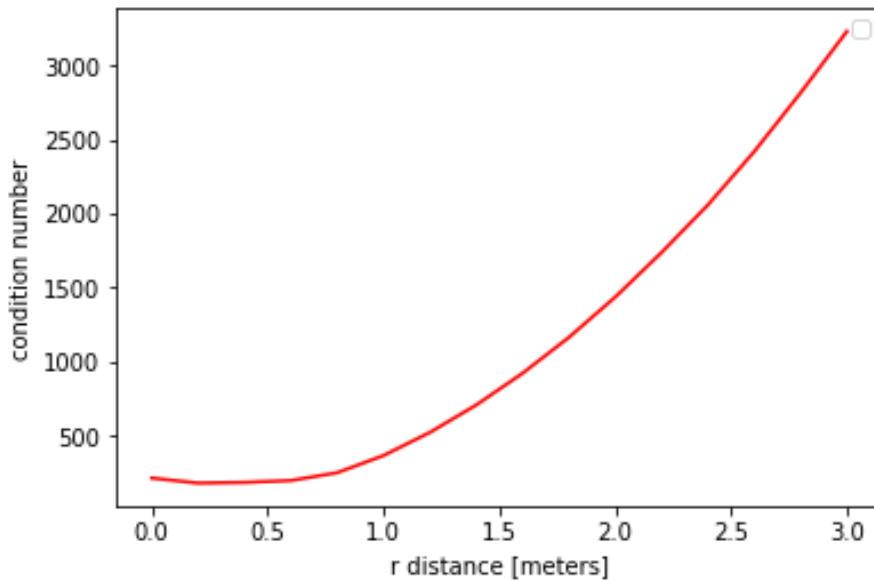
Που στα δικά μας πειράματα κατέληξε στην εξής μορφή:

$$\begin{bmatrix} \Sigma v_1 & & & & & \\ & \Sigma v_2 & & & & \\ & & \Sigma v_3 & & & \\ & & & \Sigma v_4 & & \\ & & & & \Sigma v_5 & \\ & & & & & \Sigma v_6 \end{bmatrix}^{-1} \approx \begin{bmatrix} 0.062 & & & & & \\ & 0.063 & & & & \\ & & 0.061 & & & \\ & & & 0.062 & & \\ & & & & 0.062 & \\ & & & & & 0.061 & \\ & & & & & & 0.063 & \\ & & & & & & & 0.062 & \\ & & & & & & & & 0.064 & \\ & & & & & & & & & 0.064 & \\ & & & & & & & & & & 0.064 & \\ & & & & & & & & & & & 0.061 \end{bmatrix}$$

Ως εκ τούτου, καθώς οι διαγώνιοι όροι του  $\Sigma v_i$  είχαν σχεδόν τις ίδιες τιμές σε κάθε δοκιμή, ο τελικός πίνακας συνδιακύμανσης  $\Sigma_{\alpha, \beta, r}$ , εξαρτάται κυρίως από τους Ιακωβιανούς πίνακες, που σχετίζονται με τη θέση της κάμερας σε σχέση με το βασικό δείκτη.

Με γνώμονα τα ανωτέρω συμπεράσματα, κάναμε δύο διαφορετικά πειράματα. Ένα για τη σχέση της ακρίβειας με την απόσταση της κάμερας από το βασικό δείκτη και ένα για την εξάρτηση της ακρίβειας από τη γωνία  $\alpha$  που σχηματίζει η κάμερα με το βασικό δείκτη.

Έτσι τοποθετώντας μία τυχαία διάταξη βασικού δείκτη σε σταθερό σημείο στο χώρο, αλλάζαμε διαδοχικά την απόσταση της κάμερας ( $r$ ) ξεκινώντας από 0 μέτρα και αυξάνοντας με βήμα 0.2 μέτρα κάθε φορά. Το γράφημα που προέκυψε φαίνεται στο Σχήμα 1.7 και δείχνει πως τα σφάλματα αυξάνονται όσο μεγαλώνει η απόσταση της κάμερας από το βασικό δείκτη. Το metric που χρησιμοποιήσαμε για να αξιολογήσουμε τη σχέση των σφαλμάτων και της απόστασης ήταν το condition number του πίνακα συνδιακύμανσης  $\Sigma_{\alpha, \beta, r}$ .

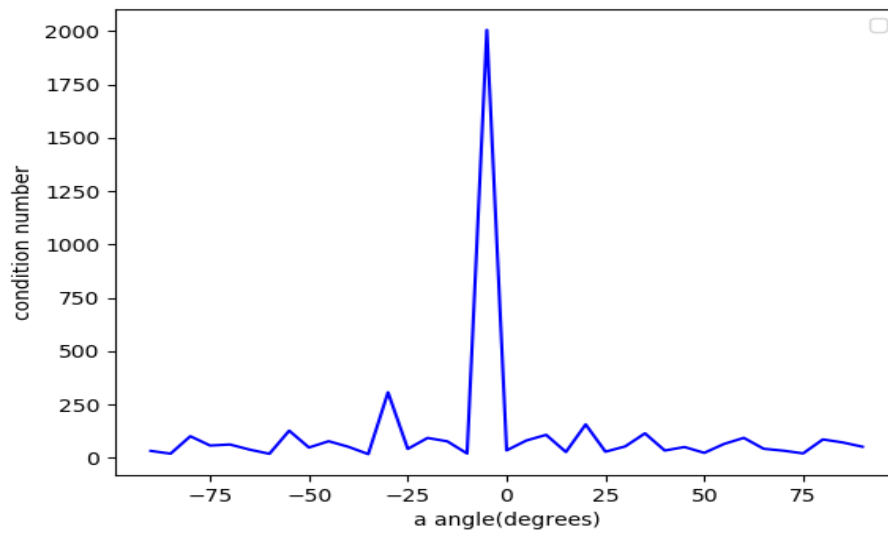


Σχήμα 1.7: Η σχέση μεταξύ του Condition number και της απόστασης της κάμερας από το βασικό δείκτη ( $X \sim \mathcal{N}(0, 4)$ ).

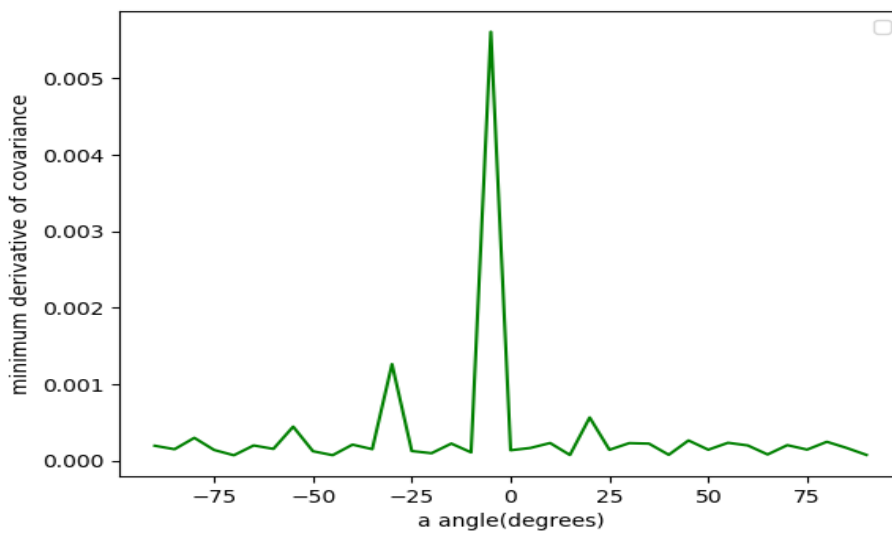
Έπειτα τοποθετήσαμε και πάλι μία τυχαία διάταξη βασικού δείκτη σε ένα σταθερό σημείο στο χώρο, και κρατώντας την απόσταση  $r$  σταθερή μεταβάλαμε την γωνία  $\alpha$  της κάμερας με το βασικό δείκτη από  $-\frac{\pi}{2}$  έως  $\frac{\pi}{2}$  με βήμα 5 μοίρες κάθε φορά. Στο πείραμα αυτό δημιουργήσαμε δύο διαφορετικά γραφήματα. Το πρώτο αφορά πάλι στο condition number του πίνακα συνδιακύμανσης  $\Sigma_{\alpha, \beta, r}$  και το δεύτερο στην εύρεση των σημείων αυτών που ελαχιστοποιούν την παράγωγο του πίνακα συνδιακύμανσης  $\Sigma_{\alpha, \beta, r}$ , κρατώντας και στις δύο περιπτώσεις σταθερή τη μεταβολή του βήματος σε 5 μοίρες κάθε φορά. Στα Σχήματα 1.8α' και 1.8β' φαίνονται τα αποτελέσματα των πειραμάτων αυτών, αντιστοίχως.

Όπως φαίνεται, τα δύο πειράματα κατέληξαν σε παρεμφερή αποτελέσματα, αφού η γωνία  $\alpha$  της κάμερας που προκαλεί τα λιγότερα σφάλματα, είναι η γωνία 90 και των -90 μοιρών. Αντίθετα το σημείο που εμφανίζονται τα υψηλότερα σφάλματα είναι όταν τοποθετούμε την κάμερα σε γωνία  $\alpha$  γύρω από τις 0 μοίρες, όπου εμφανίζεται και το ολικό μέγιστο στο Σχήμα 1.8β'.

Στη συνέχεια κατασκευάσαμε και τα ελλειψοειδή εκτίμησης του σφάλματος του  $\Sigma_{\alpha, \beta, r}$ ,

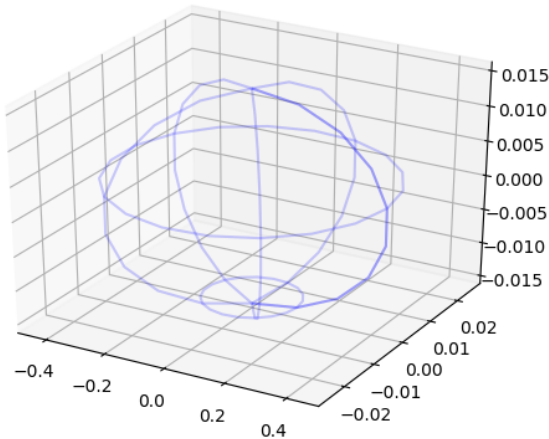
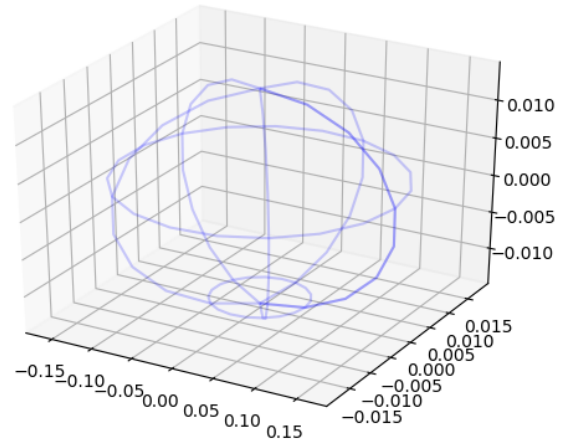


(α) Η σχέση του Condition Number και της γωνίας  $\alpha$  της κάμερας με το βασικό δείκτη.



(β) Η σχέση μεταξύ του ελαχίστου της παραγώγου του  $\Sigma_{\alpha, \beta, r}$  και της γωνίας  $\alpha$  της κάμερας με το βασικό δείκτη.

Σχήμα 1.8: Η εξάρτηση της ακρίβειας της γωνίας  $\alpha$  της κάμερας και του βασικού δείκτη ( $X \sim \mathcal{N}(0, 4)$ ).

(α') Γωνία  $\alpha = 0$ .(β') Γωνία  $|\alpha| = \frac{\pi}{2}$ .Σχήμα 1.9: Ελλειψοειδή Σφάλματος του Πίνακα Συνδιακύμανσης  $\Sigma_{\alpha,\beta,r}$ .

αξιοποιώντας το μαθηματικό υπόβαθρο που αναλήφθηκε παραπάνω. Για την γραφική αναπαράσταση των ελλειψοειδών θεωρήσαμε επίπεδο εμπιστοσύνης 95 % και παρακάτω φαίνονται στο Σχήμα 1.9α' το ελλειψοειδές σε γωνία  $\alpha$  0 μοιρών και στο Σχήμα 5.4β το ελλειψοειδές σε γωνία  $\alpha$  90 μοιρών. Τέλος, τα ελλειψοειδή του πίνακα συνδιακύμανσης για πληθώρα διαφορετικών τιμών της γωνίας  $\alpha$ , φαίνονται στο Σχήμα 1.10.

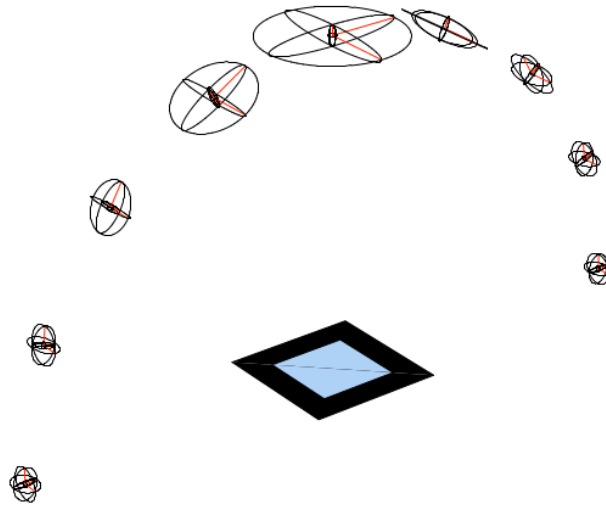
Γίνεται έτσι σαφές, ότι η βέλτιστη θέση τοποθέτησης της κάμερας σε σχέση με έναν βασικό δείκτη είναι σε γωνία  $-\frac{\pi}{2}$  ή  $\frac{\pi}{2}$  μοιρών από αυτόν και σε όσο το δυνατόν μικρότερη απόσταση  $r$ .

Συνεπώς, στην πλειονότητα των πειραμάτων που θα περιγραφούν παρακάτω τοποθετούσαμε την κάμερα στη βέλτιστη θέση που βρήκαμε σε σχέση με το βασικό δείκτη.

## 1.12 Οι τρισδιάστατες διατάξεις βασικών δεικτών με τη μεγαλύτερη ακρίβεια στον εντοπισμό θέσης

Το δεύτερο σημαντικό αντικείμενο μελέτης της παρούσας διπλωματικής ήταν η μελέτη του αν υπάρχουν συγκεκριμένες γεωμετρικές διατάξεις των βασικών δεικτών που να ελαχιστοποιούν τα σφάλματα στον προσανατολισμό, την επαναπροβολή, τη μετάφραση κτλ. Οι πρώτες μας παρατηρήσεις βασίστηκαν στην επιλογή τυχαίων σημείων. Επιλέγαμε λοιπόν 6 τυχαία σημεία που να ανήκουν στη σφαίρα ακτίνας 0.3 μ που μελετάμε, τους προσθέταμε τυχαίο θόρυβο, με τυπική απόκλιση 8 και μέση τιμή 2, και βρήκαμε τα αντίστοιχα σημεία προβολής στο πλάνο εικόνας.

Έπειτα κάνοντας χρήση του Direct Linear Transformation αλγορίθμου βρήκαμε τελικά τα σφάλματα περιστροφής, μετάφρασης και condition number για κάθε τυχαίο σύνολο ση-



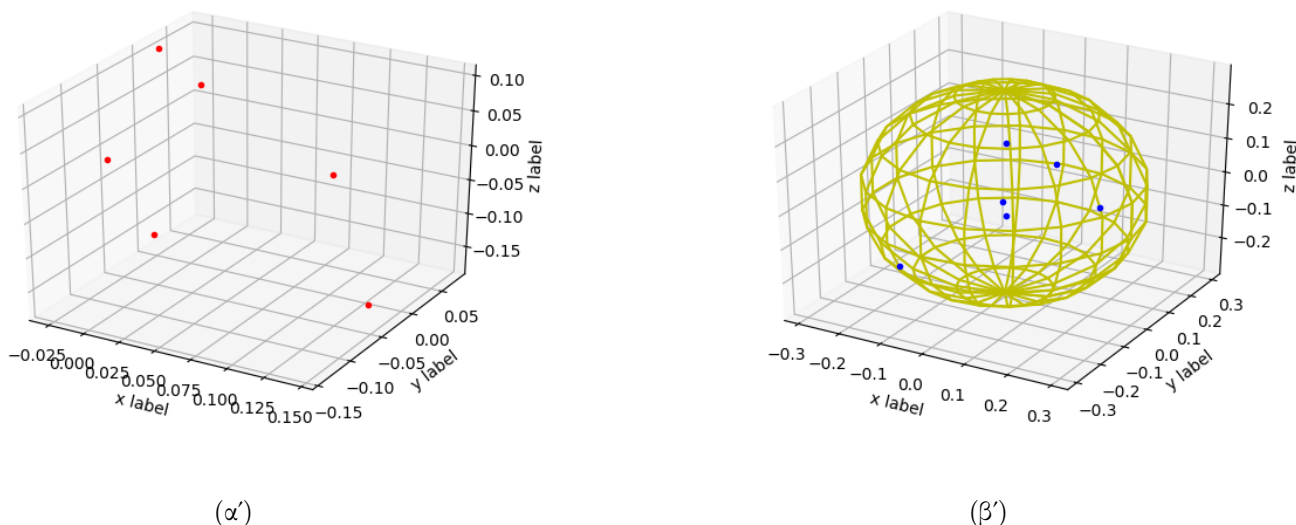
Σχήμα 1.10: Ακρίβεια της θέσης της κάμερας σε σχέση με το βασικό δείκτη για διάφορες γωνίες  $\alpha$  [6].

μείων που επιλεγόταν. Αρχικά πραγματοποιούσαμε 500 επαναλήψεις στις οποίες επιλέγαμε 6 τυχαία σημεία και στο τέλος των επαναλήψεων κρατούσαμε το σύνολο των σημείων αυτών που παρουσίασαν τα μικρότερα σφάλματα. Έπειτα αυξήσαμε τις επαναλήψεις σε 10.000. Αυτό που παρατηρήσαμε σε όλα τα test που πραγματοποιήσαμε ήταν πως οι διατάξεις σημείων που οδηγούσαν στα μικρότερα σφάλματα, πλησίαζαν αρκετά την περιφέρεια της σφαίρας μέσα από την οποία επιλέγαμε τα σημεία. Έτσι δηλαδή τα πιο απομακρυσμένα μεταξύ τους σημεία να οδηγούν σε μικρότερα σφάλματα.

Με γνώμονα αυτά τα πρωτόλεια συμπεράσματα αυτό το επόμενο μας βήμα ήταν να εφαρμόσουμε διαδοχικά τους δύο αλγόριθμους βελτιστοποίησης που αναλύσαμε παραπάνω: τον Gradient Descent και τον Adam Optimizer.

Ξεκινώντας με τον αλγόριθμο Gradient Descent προσπαθήσαμε να εντοπίσουμε το μοτίβο των τρισδιάστατων διατάξεων βασικών δεικτών που εμφανίζουν τα μικρότερα σφάλματα στον εντοπισμό τους. Έτσι επειδή αναλύουμε ένα τρισδιάστατο πρόβλημα, ο ρυθμός εκπαίδευσης αξιολογείται σε κάθε βήμα συναρτήσει 3 διαφορετικών παραμέτρων. Το metric που επιλέχθηκε προς βελτιστοποίηση μέσα από την εφαρμογή του συγκεκριμένου αλγόριθμου, είναι το condition number, ενώ για την αξιολόγηση του βήματος χρησιμοποιήθηκε ένας αλγόριθμος που ονομάζεται SuperSAB που αποτελεί μια στρατηγική επιτάχυνστης για εκμάθηση error back propagation [37].

Εκτελέσαμε λοιπόν τον αλγόριθμο βελτιστοποίησης Gradient Descent για επαναλήψεις  $N = 100$ . Ως το πρώτο αρχικό σύνολο σημείων, χρησιμοποιούσαμε ένα τυχαία επιλεγμένο σύνολο έξι σημείων που ανήκουν στο ίδιο επίπεδο σφαίρας με πριν, με ακτίνα δηλαδή 0.3 μέτρα. Τα



Σχήμα 1.11: Οι διατάξεις των σημείων στην πρώτη εκτέλεση του Gradient Descent.

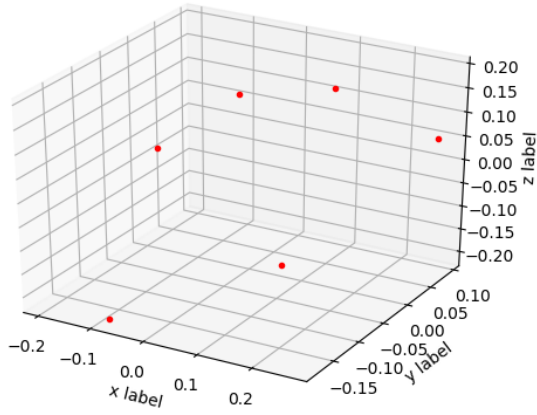
προβαλλόμενα σημεία εικόνας υπολογίζονται ξανά και προστίθεται τυχαίος θόρυβος Γαους σε κάθε ένα από τα pixel της εικόνας. Κάθε δείγμα θορύβου είχε μέση τιμή 2 και τυπική απόκλιση 8. Για τη βελτίωση της ποιότητας των εκτιμήσεών μας προσθέταμε 1000 τυχαία δείγματα θορύβου Gauss σε κάθε pixel και υπολογίζαμε τη μέση τιμή καθενός από αυτά.

Τα αποτελέσματα αυτών των δοκιμών παρουσιάζονται παρακάτω. Συγκεκριμένα, στο Σχήμα 1.11 υπάρχει το σύνολο των έξι τυχαίων σημείων που επιλέχθηκαν από τη σφαίρα στην πρώτη επανάληψη. Στη συνέχεια, το Σχήμα 1.12 δείχνει τις θέσεις των 6 σημείων στην επανάληψη 50 και στο Σχήμα 1.13 υπάρχει η τελική θέση του συνόλου των 6 σημείων μετά την επανάληψη 100. Αυτό το τελικό σύνολο σημείων έχει το ελάχιστο condition number, αλλά και το ελάχιστο σφάλμα περιστροφής, μετάφρασης και επαναπροβολής.

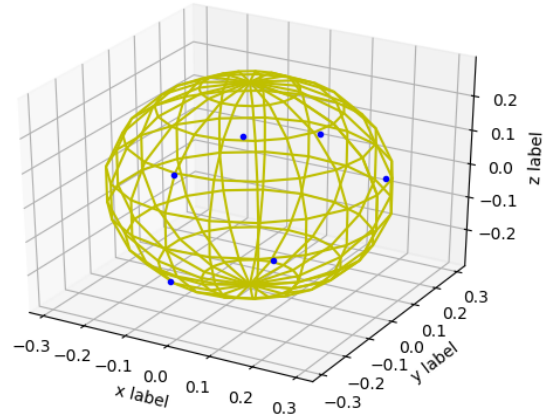
Μετά από μερικές δοκιμές του αλγορίθμου, συμπεράναμε για άλλη μια φορά, ότι το σύνολο των σημείων που προκαλούν τα μικρότερα σφάλματα και το ελάχιστο condition number, είναι σημεία που ανήκουν στην περιφέρεια της σφαίρας, ή τουλάχιστον είναι πολύ κοντά σε αυτή. Επιπλέον μετά από περίπου 80 επαναλήψεις του αλγορίθμου, τα σημεία έτειναν να σχηματίζουν ένα τριγωνικό πρίσμα. Παράλληλα με την εκτέλεση του αλγορίθμου για βελτιστοποίηση του condition number, που φαίνεται στο Σχήμα 1.15, μελετούσαμε επίσης πως εξελίσσονται τα σφάλματα επαναπροβολής, περιστροφής και μετάφρασης.

Έτσι στο ίδιο Σχήμα γίνεται σαφές ότι μικρές μεταβολές στην τιμή του condition number προκαλούν επίσης μεταβολές στα σφάλματα περιστροφής και μετάφρασης. Επίσης κάθε τοπικό μέγιστο στη γραφική παράσταση του σφάλματος περιστροφής, αντιστοιχεί σε ένα τοπικό μέγιστο στη γραφική παράσταση του σφάλματος μετάφρασης, πράγμα που ισχύει και για τα τοπικά ελάχιστα. Ως εκ τούτου, αυτά τα σύνολα σημείων, τα οποία αντιστοιχούν στο τοπικό ελάχιστο στη γραφική παράσταση του σφάλματος μετάφρασης, μπορούν να χρησιμοποιηθούν ως είσοδος στον αλγόριθμο Direct Linear Transformation και το αποτέλεσμα που θα λάβουμε



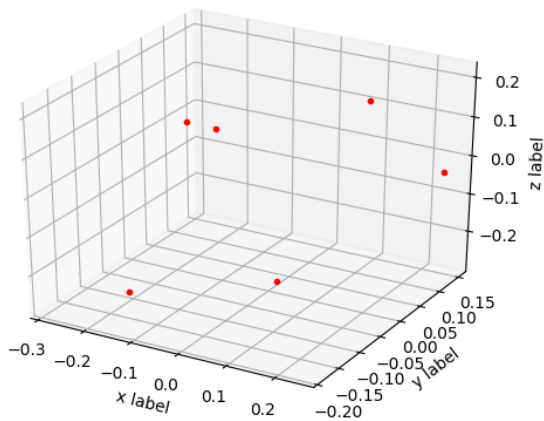


(α')

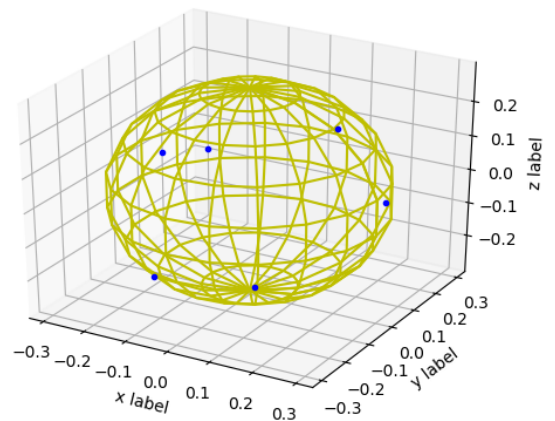


(β')

Σχήμα 1.12: Οι διατάξεις των σημείων στην 50 επανάληψη του Gradient Descent.

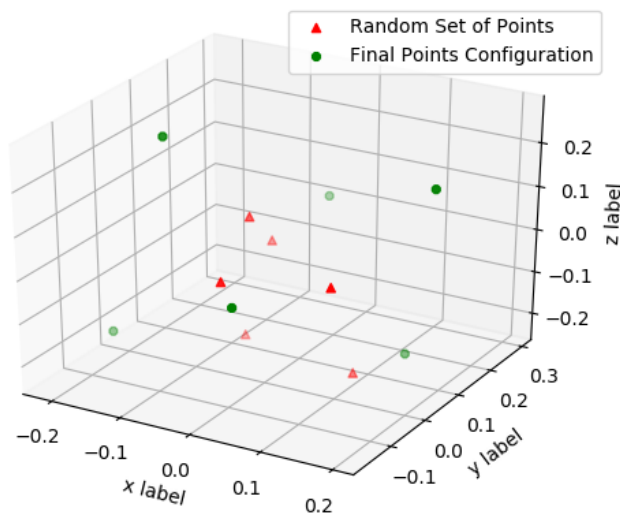


(α'')



(β'')

Σχήμα 1.13: Η τελική διάταξη των σημείων που έδειξε η εκτέλεση του Gradient Descent.



Σχήμα 1.14: Συγκρίνοντας τις διατάξεις σημείων της πρώτης και της εκατοστής επανάληψης του αλγορίθμου Gradient Descent.

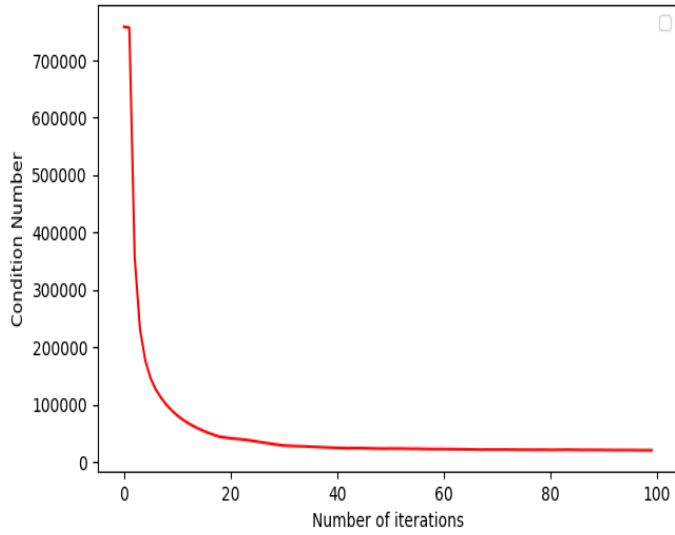
θα οδηγήσει στην εκτίμηση του  $\hat{R}$  με τιμή όσο γίνεται πιο κοντά στην πραγματική του.

Παράλληλα, όπως γίνεται αντιληπτό, η τελική γεωμετρική, τρισδιάστατη διάταξη των σημείων (στην τελευταία επανάληψη  $N = 100$ ), αντιστοιχεί στο ολικό ελάχιστο σε κάθε μία από τις γραφικές παραστάσεις, πράγμα που σημαίνει ότι αυτή είναι η διάταξη που οδηγεί συνολικά σε ελάχιστα σφάλματα.

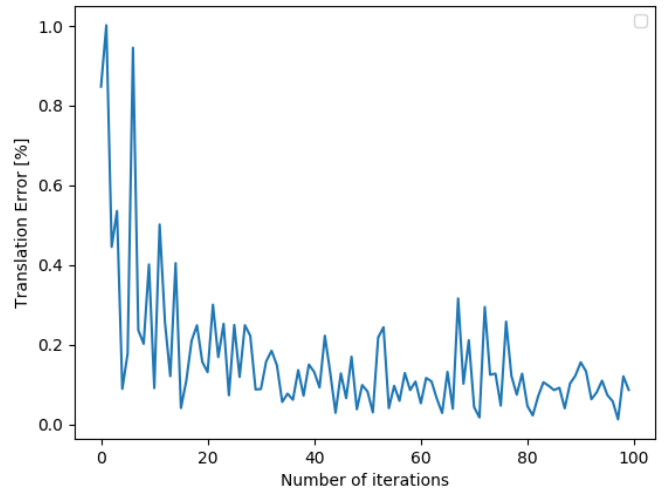
Επιπλέον το Σχήμα 1.14 απεικονίζει το σύνολο σημείων της πρώτης επανάληψης στην εκτέλεση του αλγορίθμου ( $N = 1$ ), και την τελική διάταξη σημείων στην τελευταία επανάληψη κατά την εκτέλεση του αλγορίθμου ( $N = 100$ ). Συνεπώς, όπως φαίνεται τα σημεία με την πάροδο των επαναλήψεων τείνουν να απομακρύνονται το ένα από το άλλο και να πλησιάζουν όλο και πιο κοντά στην περιφέρεια της σφαίρας.

Θα μπορούσαμε να πούμε πως αλγόριθμος βελτιστοποίησης Gradient Descent, επιβεβαιώνει τις παρατηρήσεις μας μέσω της τυχαίας επιλογής σημείων σφαίρας που έγινε πειραματικά στο πρώτο στάδιο της διπλωματικής αυτής. Ωστόσο, αποφασίσαμε να χρησιμοποιήσουμε έναν ακόμη αλγόριθμο όπως αναφέρθηκε και παραπάνω για να δούμε αν θα επικυρώσει και τα αποτελέσματά μας.

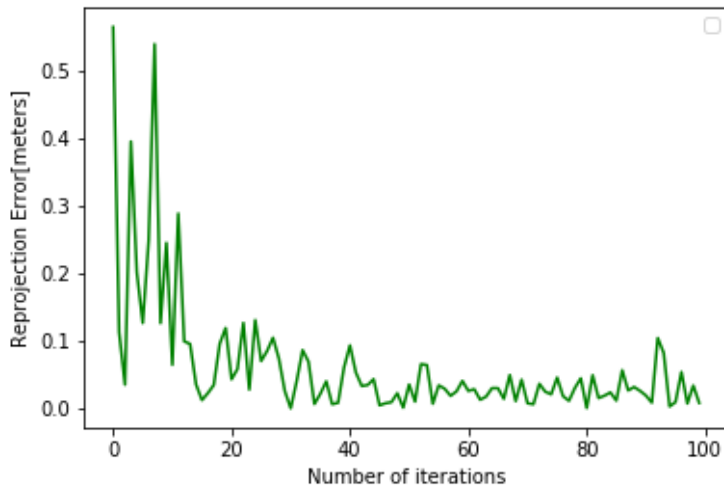
Ο αλγόριθμος Adam Optimizer χρησιμοποιήθηκε προκειμένου να διαπιστώσουμε αν θα επιβεβαιωνθούν ξανά οι παρατηρήσεις μας. Χρησιμοποιήσαμε λοιπόν και πάλι την ίδια σφαίρα με ακτίνα 0.3 μέτρα και τον έτοιμο αλγόριθμο όπως παρέχεται από την Python [30]. Αυτή τη φορά εκτελούσαμε τον αλγόριθμο για επαναλήψεις  $N = 200$ , αντί για  $N = 100$  που επιλέξαμε για τον αλγόριθμο Gradient Descent προηγουμένως. Και πάλι 1000 τυχαία δείγματα θορύβου Gauss προστέθηκαν στα pixel εικόνας με μέση τιμή 2 και τυπική απόκλιση 8 και εκτιμήσαμε τη μέση τιμή καθενός.



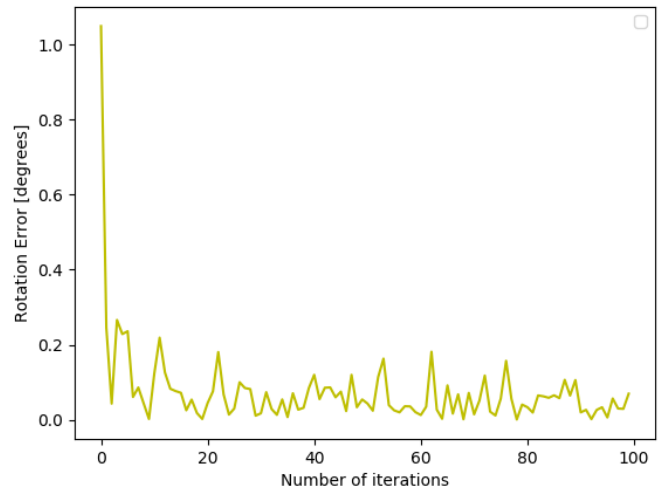
(α') Ο Αριθμός Συνθήκης (Condition Number).



(β') Τα σφάλματα μετάφρασης.

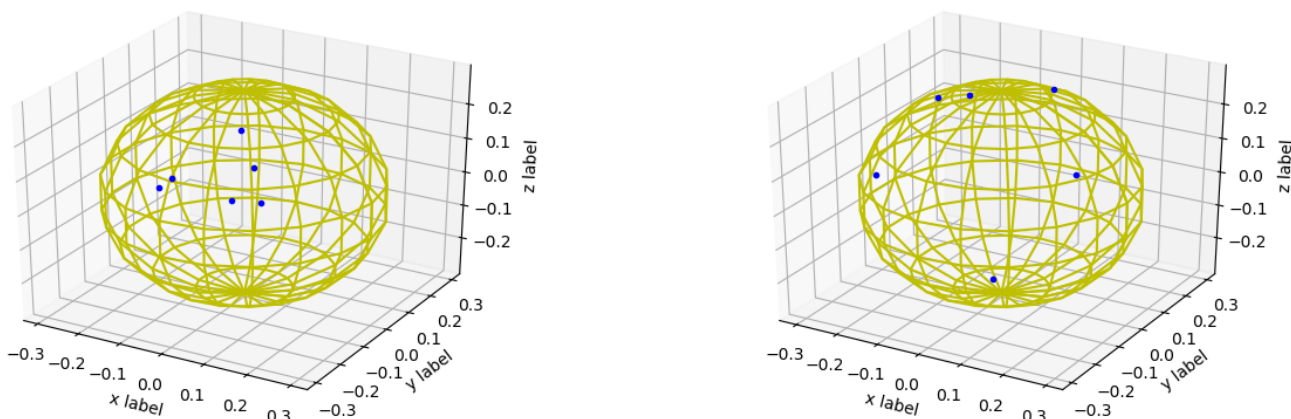


(γ') Τα σφάλματα επαναπροβολής.



(δ') Τα σφάλματα περιστροφής.

Σχήμα 1.15: Σφάλματα κατά την εκτέλεση του Gradient Descent ( $X \sim \mathcal{N}(2, 8)$ ).

(α)  $N = 1$ .(β)  $N = 200$ .

Σχήμα 1.16: Οι γεωμετρικές διατάξεις των 6 σημείων με τη χρήση του αλγορίθμου Adam.

Κάθε φορά εφαρμόζαμε τον αλγόριθμο Direct Linear Transformation στα 6 σημεία που προέκυπταν από τον αλγόριθμο Adam και βρίσκαμε τα  $\hat{R}$ ,  $\hat{t}$  τους. Για τη βελτιστοποίηση χρησιμοποιήσαμε και πάλι ως metric το condition number του πίνακα  $H_{DLT}$ , και για κάθε σύνολο έξι σημείων που μελετούσαμε προσπαθούσαμε να βρούμε τη διάταξη που ελαχιστοποιεί την τιμή του.

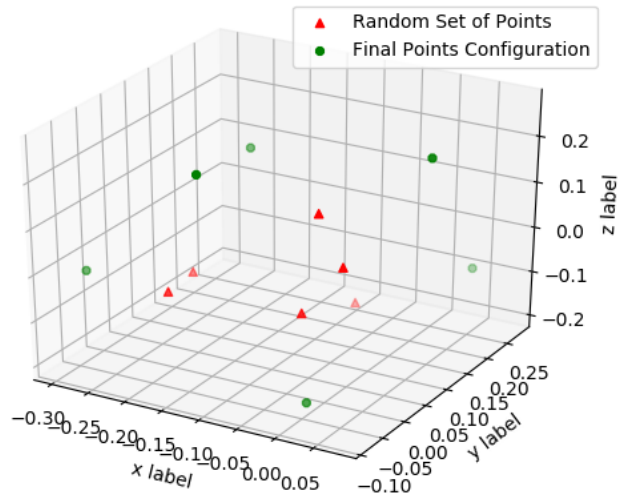
Στο Σχήμα 1.16α εμφανίζεται το πρώτο τυχαίο επιλεγμένο σύνολο σημείων, το οποίο περιορίζεται από το καθορισμένο επίπεδο σφαίρας. Επιπλέον, το Σχήμα 1.16β απεικονίζει την τελική διάταξη των 6 τρισδιάστατων σημείων.

Παρατηρήσαμε επίσης πως η διάταξη των 6 σημείων μετά από έναν αριθμό επαναλήψεων ( $\approx 100$ ) υιοθετούσαν μία τελική θέση και έπειτα παρατηρούνταν μόνο μικρές αποκλίσεις γύρω από αυτή τη θέση.

Στο Σχήμα ;; παρουσιάζονται τα σφάλματα, καθώς και το condition number όπως μεταβάλλονται στις 200 επαναλήψεις του αλγορίθμου. Καθώς ο αριθμός  $N$  των επαναλήψεων αυξάνεται, τα σφάλματα μειώνονται μέχρι που αποκτούν μια σχεδόν σταθερή τιμή, καθώς το σύνολο των 6 σημείων οριστικοποιείται σε μια συγκεκριμένη θέση, σχηματίζοντας την τελική του γεωμετρική διάταξη.

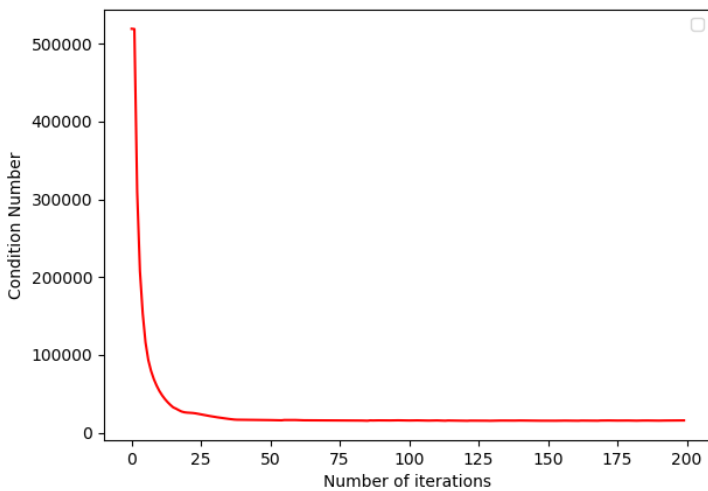
Ακόμη, παρατηρούμε ξανά ότι καθώς τα σημεία απομακρύνονται το ένα από το άλλο και πλησιάζουν στην περιφέρεια της σφαίρας, τα σφάλματα μειώνονται. Επιπλέον, μόλις τα σημεία δημιουργούν ένα τριγωνικό πρίσμα (γεωμετρική διάταξη), το οποίο είναι το βέλτιστο σχήμα για τις δοκιμές 6 σημείων, με βάση τις προσομοιώσεις μας, ο αλγόριθμος βελτιστοποίησης βρίσκει το ολικό ελάχιστο σε κάθε γράφημα των σφαλμάτων που μελετάμε.

Τέλος, φαίνεται στο σχήμα 1.17 η σχέση μεταξύ του συνόλου των σημείων που επιλέχθηκαν τυχαία στην πρώτη επανάληψη ( $N = 1$ ) και της τελικής γεωμετρικής διάταξης των σημείων ( $N = 200$ ). Τα αποτελέσματα είναι παρόμοια με αυτά που είχαμε με τις άλλες μεθόδους που

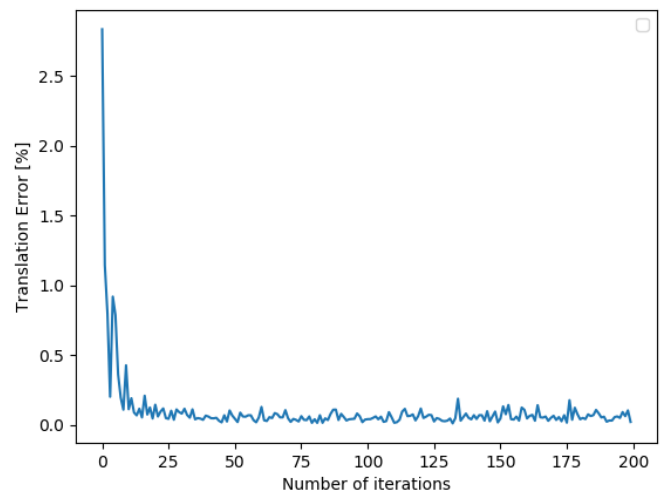


Σχήμα 1.17: Σύνολο σημείων της πρώτης και της τελευταίας επανάληψης στην εκτέλεση του Adam Optimizer.

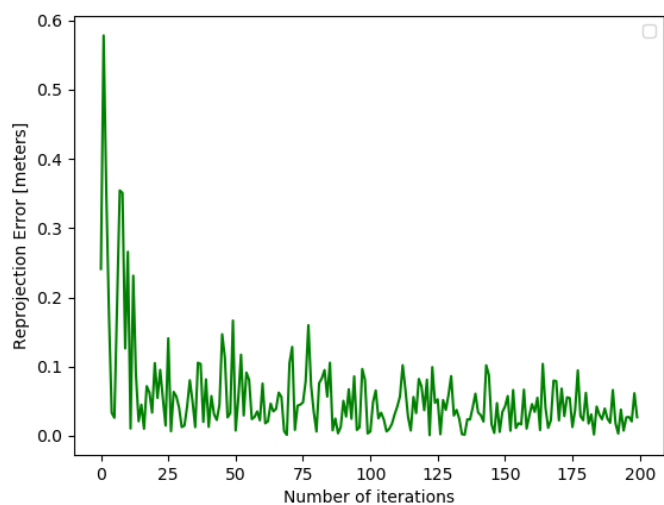
δοκιμάσαμε, οπότε επιβεβαιώνουμε ότι οι πιο ακριβείς τρισδιάστατες διατάξεις σημείων είναι θυμίζουν τριγωνικά πρίσματα.



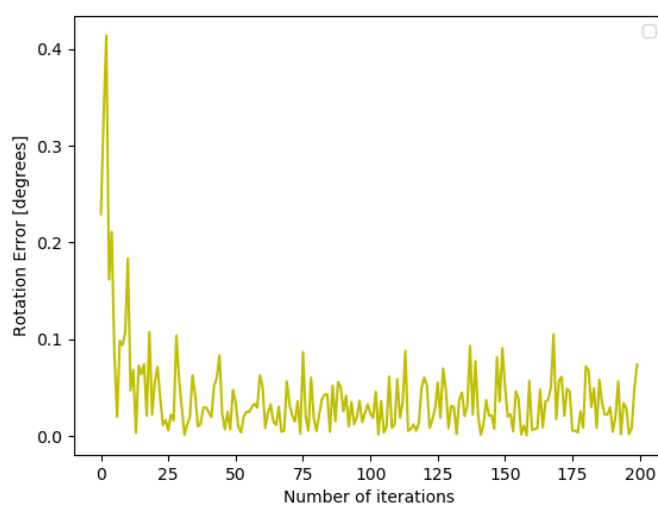
(α') Ο αριθμός Συνθήκης (Condition Number).



(β') Σφάλματα Μετάφρασης.



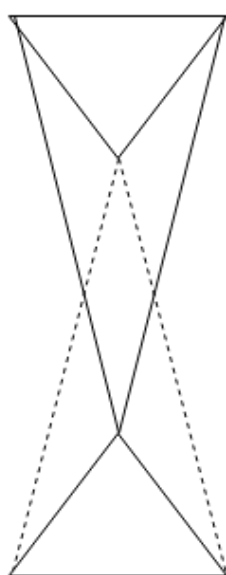
(γ') Σφάλματα Επαναπροβολής.



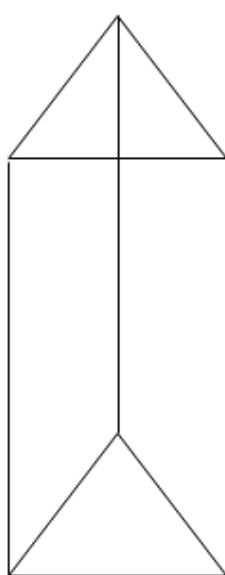
(δ') Σφάλματα Περιστροφής.

Σχήμα 1.18: Σφάλματα σε κάθε επανάληψη του Adam Optimization ( $X \sim \mathcal{N}(2, 8)$ ).

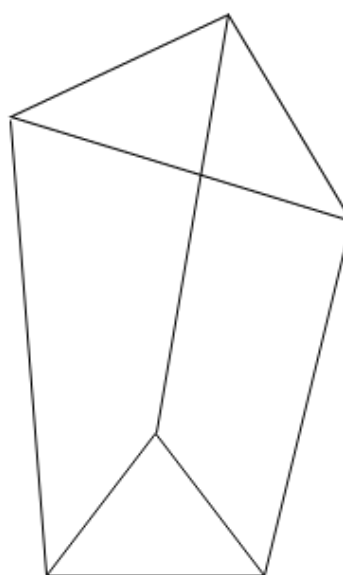
Κλείνοντας, ολοκληρώνοντας τα 3 αυτά βασικά πειράματα της διπλωματικής μας, καταλήξαμε στις πιθανές τρισδιάστατες διατάξεις σημείων βασικών δεικτών, που μπορούν να οδηγήσουν σε μειωμένα σφάλματα. Οι διατάξεις αυτές παρουσιάζονται στο Σχήμα 1.19. Τα αποτελέσματα αυτά θα χρησιμοποιηθούν στη συνέχεια για τη συγκεκριμένη περίπτωση της λαπαροσκοπικής χειρουργικής επέμβασης.



Twisted Triangular Prism



Normal Triangular Prism



Truncated Triangular Prism

Σχήμα 1.19: Οι διατάξεις τριγωνικών πρισμάτων.

Στην παρούσα διπλωματική πραγματοποιήθηκαν και πειράματα αλλάζοντας τη θέση της κάμερας, ωστόσο για όσα θα αναφερθούν στην εκτεταμένη περίληψη, υποθέτουμε την κάμερα πάντοτε σε γωνία 90 μοιρών από τον τρισδιάστατο βασικό δείκτη, τη διάταξη του οποίου αναζητούμε.

### 1.13 Σχεδιάζοντας ένα βασικό δείκτη για Λαπαροσκοπικές Επεμβάσεις

Η οπτική παρακολούθηση χρησιμοποιείται κυρίως για την παρακολούθηση ιατρικών οργάνων σε χειρουργία (π.χ. παρακολούθηση ενδοσκοπίου σε λαπαροσκοπικές χειρουργικές επεμβάσεις). Έτσι οι βασικοί δείκτες, που έχουν προκαθορισμένα μοτίβα προσαρτώνται στο ιατρικό όργανο (λαπαροσκόπιο, ενδοσκόπιο), το οποίο χρειάζεται να ξέρουμε σε ποια θέση στο χώρο βρίσκεται κάθε στιγμή, σε πραγματικό χρόνο.

Η οπτική παρακολούθηση προσδιορίζει τη στάση ενός αντικειμένου που παρακολουθείται μετρώντας το φως που μεταδίδεται ή αντανακλάται από αυτό το αντικείμενο. Έτσι απώτερος στόχος αυτής της διπλωματικής ήταν ο σχεδιασμός ενός βασικού δείκτη που θα επέτρεπε την οπτική παρακολούθηση με τα λιγότερα δυνατά σφάλματα.

Δεδομένου λοιπόν, ότι η ακρίβεια είναι ένα κρίσιμο ζήτημα στη χειρουργική και πιο συγκεκριμένη λαπαροσκοπική επέμβαση με τη χρήση επαυξημένης πραγματικότητας, εξετάζουμε εάν οι τρισδιάστατες γεωμετρικές διατάξεις σημείων βασικών δεικτών 'καλά προσαρμοσμένης' (μετά τη βαθμίδα βελτίωσης κατάβασης), που υπολογίσαμε προηγουμένως, θα οδηγήσουν σε ελάχιστα σφάλματα σε μια προσομοιωμένη χειρουργική λαπαροσκοπίου σε σχέση με το να επιλέγαμε μία τυχαία γεωμετρική διάταξη. Αρχικά, σύμφωνα με το [6], περισσότερα fiducials μπορούν να βελτιώσουν την ακρίβεια, καθώς μειώνουν το σφάλμα συνδιακύμανσης κατά:

$$\frac{1}{\sqrt{n}}, \text{ where } n = \text{number of fiducials.} \quad (1.46)$$

Από την άλλη πλευρά, καθώς ο αριθμός των fiducials αυξάνεται, αυξάνεται επίσης η εμφάνιση occlusions μεταξύ τους. Έτσι, αποφασίζουμε να χρησιμοποιήσουμε 6 δείκτες (αντί για τέσσερις που χρησιμοποιούνται συνήθως), καθώς έξι είναι ο ελάχιστος αριθμός σημείων, που απαιτείται να εφαρμοστούν στο και στον αλγόριθμο Direct Linear Transformation, για την επίλυση του προβλήματος εκτίμησης θέσης.

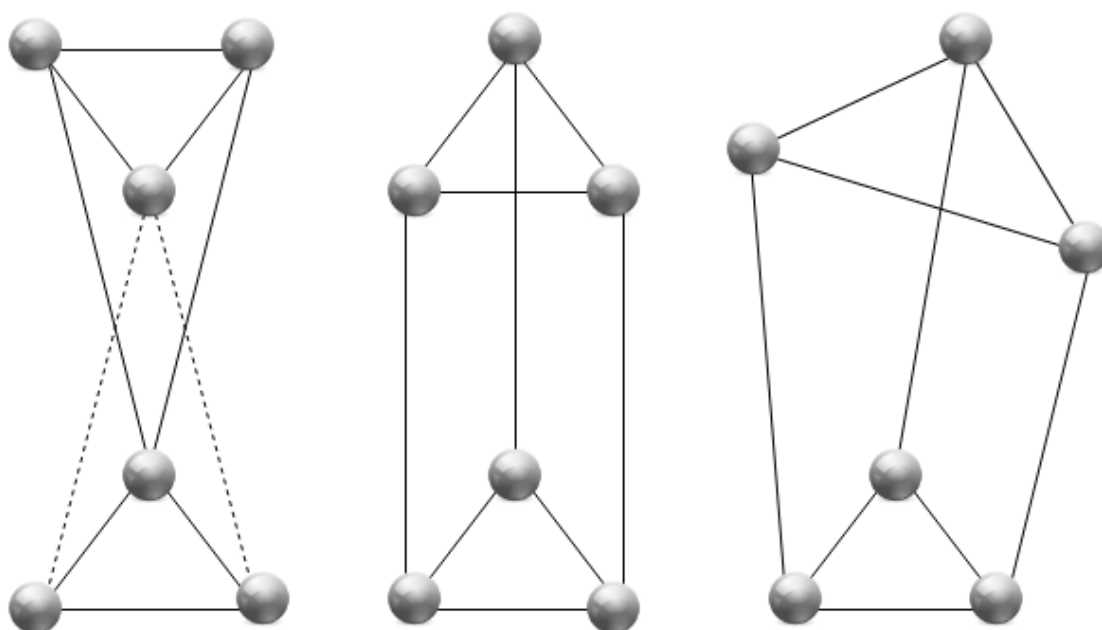
Οι δείκτες που προτείνουμε είναι παθητικοί - καθώς αυτό το είδος δεικτών χρησιμοποιείται πιο συχνά στα συστήματα παρακολούθησης-, που σημαίνει ότι αντανακλούν την εισερχόμενη ακτινοβολία προς την κατεύθυνση του εισερχόμενου φωτός. Έχουν σφαιρικό σχήμα και καλύπτονται με ανακλαστικά φύλλα. Για να αποφασίσουμε την ακτίνα των σφαιρικών δεικτών μας, αναζητήσαμε τα διαθέσιμα μεγέθη σφαιρικών δεικτών στην αγορά σήμερα.

Όλα τα διαθέσιμα μεγέθη παθητικών σφαιρικών δεικτών παρουσιάζονται στον Πίνακα 1.2. Ωστόσο, δείκτες με επικαλυμμένη, στιβαρή επιφάνεια διατίθεται μόνο στα ακόλουθα μεγέθη: 0.014, 0.016 και 0.02 μέτρα. Έτσι, επιλέγουμε να έχουμε 6 σφαιρικούς δείκτες με διάμετρο 1.6 εκατοστά.

Μέγεθος [mm]	Μέγεθος [1/64 in]
12	30.2
14	35.3
16	40.3
20	50.4
30	75.6

Πίνακας 1.2: Διαθέσιμα μεγέθη για παθητικούς σφαιρικούς δείκτες στην αγορά.

Αυτοί οι παθητικοί 6 δείκτες θα προσαρτηθούν σε ένα λαπαροσκόπιο με μία από τις πιθανές διατάξεις που φαίνονται στο Σχήμα 1.20, προκειμένου να παρακολουθούν το εν λόγω ιατρικό όργανο σε πραγματικό χρόνο. Συγκεκριμένα, θα έχουν το βέλτιστο σχήμα ενός περικομμένου, φυσιολογικού ή ανάποδου τριγωνικού πρίσματος.



Σχήμα 1.20: Η προτεινόμενη τρισδιάστατη διάταξη σημείων για να προσαρτηθούν σε ένα Λαπαροσκόπιο.

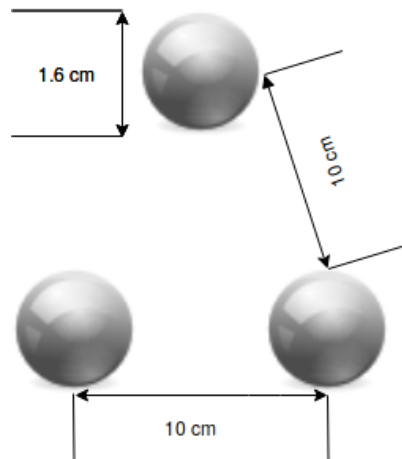
Για να υπολογίσουμε τη βάση του πρίσματος, πρέπει να γνωρίζουμε εκ των προτέρων την απόσταση μεταξύ καθενός από τους έξι δείκτες. Ως εκ τούτου, ψάχνουμε το μήκος ενός λαπαροσκοπίου, καθώς ο νέος δείκτης που κατασκευάζουμε θα πρέπει να προσαρμοστεί σε αυτό. Υπάρχουν πολλά λαπαροσκόπια στην αγορά, αλλά αποφασίσαμε να χρησιμοποιήσουμε το Olympus Laparoscopes HD και διαβάζοντας τις λεπτομέρειες του προϊόντος, διαπιστώνουμε ότι ο ελεύθερος χώρος για την τοποθέτηση των δεικτών είναι 10 εκατοστά.

Έτσι, χρησιμοποιώντας το Πυθαγόρειο θεώρημα βρίσκουμε την ακτίνα της σφαίρας στην οποία είναι προσαρτημένοι όλοι οι 6 δείκτες. Εάν ορίσουμε το  $\alpha = 10$  εκατοστά στο Σχήμα 1.21β', τότε θα έχουμε την ακόλουθη εξίσωση:

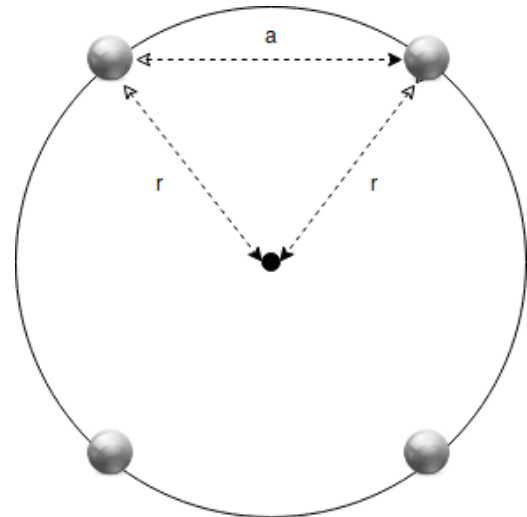


$$a^2 = r^2 + r^2 \implies 10^2 = 2r^2 \implies r = 7.071\text{cm} = 0.07071\text{m}. \quad (1.47)$$

Οπότε τα 6 σημεία θα ανήκουν σε μία σφαίρα με ακτίνα 0.07071 μέτρα, όπως παρουσιάζεται στο Σχήμα 1.22.

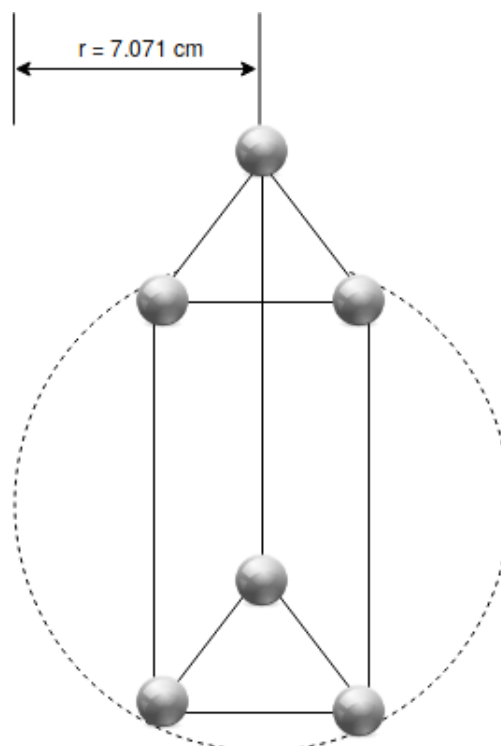


(α') Γεωμετρικά χαρακτηριστικά για την κατασκευή του δείκτη.



(β') Απόσταση ανάμεσα σε κάθε παθητική σφαίρα στη βάση του πρίσματος.

Σχήμα 1.21: Η γεωμετρία των βασικών δεικτών.



Σχήμα 1.22: Το τελικό πρίσμα εγγεγραμμένο στη σφαίρα ακτίνας 0.07071 μέτρων.

Έπειτα αφού επιβεβαιώσαμε ότι στη νέα μας σαφώς ορισμένη σφαίρα με τη χρήση των αλγορίθμων βελτιστοποίησης Gradient Descent, Adam Optimizer, καθορίζει πάλι πως τα σημεία που οδηγούν στην ελαχιστοποίηση των σφαλμάτων ανήκουν σε τριγωνικά πρίσματα, προχωρήσαμε στα τελευταία μας πειράματα όπου δημιουργήσαμε 2 νέους αλγορίθμους.

Ο πρώτος επέλεγε 6 τυχαία σημεία της σφαίρας που σχημάτιζαν τριγωνικά πρίσματα στην επιφάνειά της, ενώ ο δεύτερος επέλεγε 6 εντελώς τυχαία σημεία της σφαίρας. Έτσι λοιπόν δημιουργήσαμε 1000 δοκιμές με τυχαίες επιλογές σημείων από τους δύο αλγορίθμους και συγκρίναμε το τελικό Root Mean Square σφάλμα που έδιναν. Σε κάθε επανάληψη του πειράματος αυτού καταλήγαμε και πάλι πως το Root Mean Square Σφάλμα των σημείων του τριγωνικού πρίσματος ήταν μικρότερο από αυτό της τυχαίας επιλογής σημείων από τη σφαίρα. Η διαφορά των σφαλμάτων σε pixel και σε μέτρα φαίνεται παρακάτω:

$$e_{RMS,prism} \approx 1674.07611847 px \implies e_{RMS,prism} \approx 0.442933 m,$$

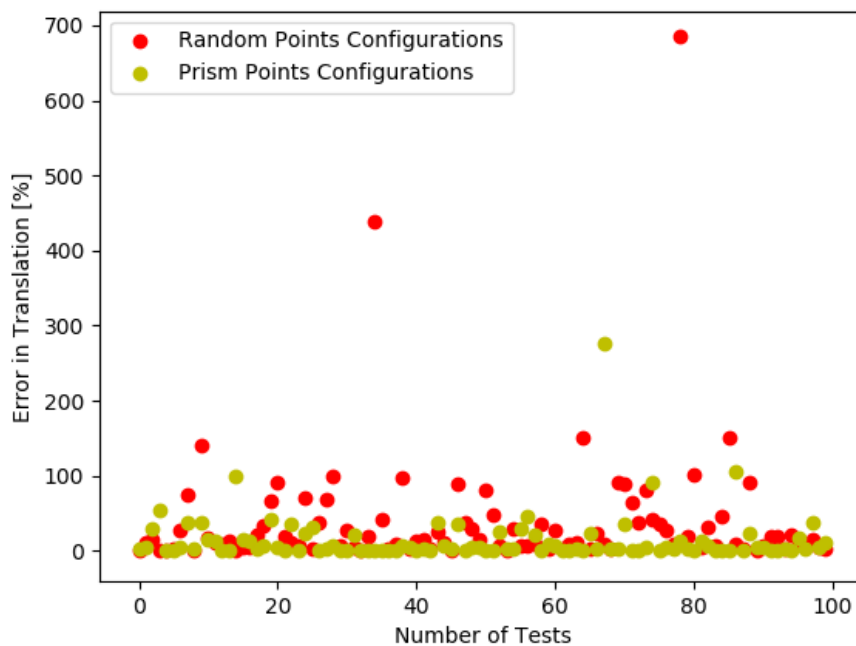
$$e_{RMS,sphere} \approx 1721.83215184 px \implies e_{RMS,sphere} \approx 0.455568 m.$$

Ταυτόχρονα πραγματοποιήσαμε και 100 δοκιμές ανά πείραμα όπου συγκρίναμε 6 τυχαία σημεία που ανήκουν στη σφαίρα και σχηματίζουν τριγωνικά πρίσματα με 6 τυχαία εντελώς σημεία της σφαίρας κάθε φορά. Για κάθε ένα από τα 100 σύνολα σημείων που διαλέγαμε συγκρίναμε τα σφάλματα περιστροφής και μετάφρασης των τυχαίων σημείων της σφαίρας και των σημείων του τριγωνικού πρίσματος. Για να λάβουμε τα σφάλματα αυτά πάλι προσθέταμε 1000 δείγματα Γκαουσιανού θορύβου με τυπική απόκλιση 4 και μέση τιμή 2 στα pixel της εικόνας και λαμβάναμε το μέσο όρο κάθε φορά, για υψηλότερη ακρίβεια στους υπολογισμούς μας.

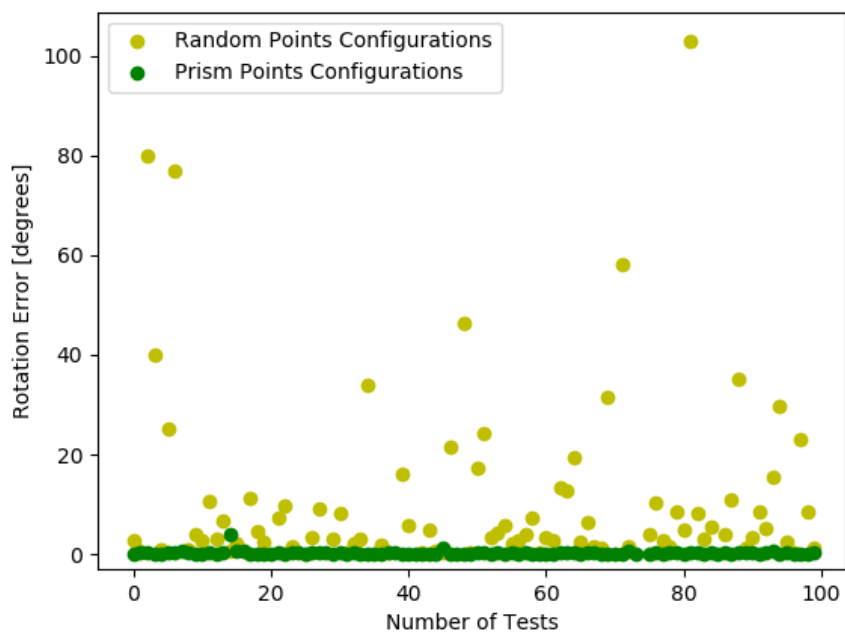
Τα αποτελέσματά μας φαίνονται για τη σύγκριση των σφαλμάτων μετάφρασης στο Σχήμα 1.23 και για τη σύγκριση των σφαλμάτων περιστροφής στο Σχήμα 1.24. Είναι ιδιαίτερα σημαντικό πως στο 93 % των περιπτώσεων τα τυχαία σημεία είχαν υψηλότερο σφάλμα μετάφρασης, ενώ στο 97 % των περιπτώσεων τα τυχαία σημεία είχαν και υψηλότερο σφάλμα περιστροφής. Συγκρίνοντας λοιπόν, καταλήγουμε πως η επιλογή σημείων τριγωνικών πρισμάτων δίνει στη συντριπτική πλειονότητα των περιπτώσεων μικρότερα σφάλματα.

Στη συνέχεια πραγματοποιήσαμε τα ίδια πειράματα με 100 δείγματα για να βρούμε τις σχέσεις των σημείων τριγωνικών πρισμάτων και των τυχαίων σημείων με το σφάλμα επαναπροβολής. Όπως είναι ορατό από το Σχήμα 1.25, οι διατάξεις σημείων τριγωνικού πρίσματος οδηγούν στο 90 % των δοκιμών στο ελάχιστο Σφάλμα επαναπροβολής σε σχέση με τις τυχαίες παραγόμενες διατάξεις σημείων σφαίρας.

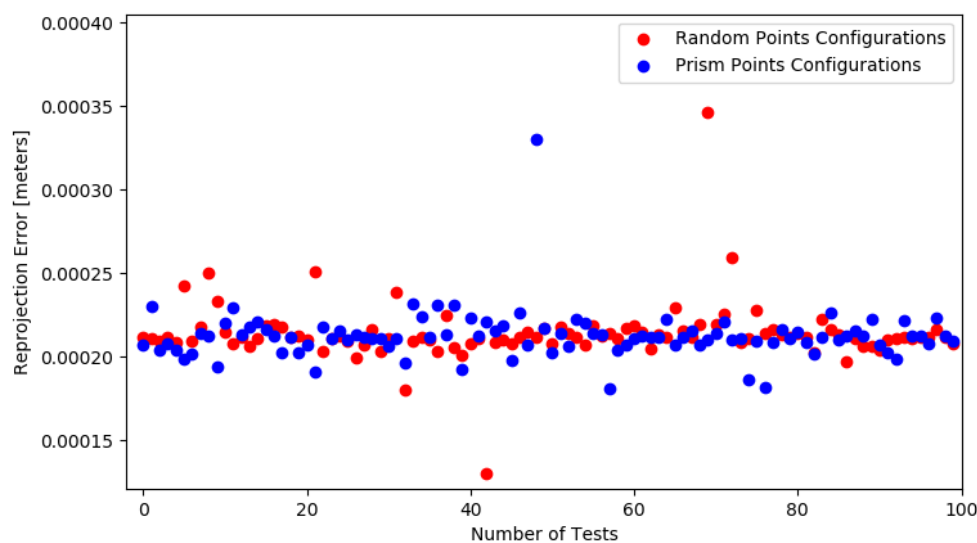
Στη συνέχεια, υπολογίζουμε τα μέσα σφάλματα στη μετάφραση, την περιστροφή και την επαναπροβολή, για αυτά τα 100 σύνολα σημείων σφαίρας και 100 σύνολα πρισμάτων. Σε κάθε προσομοίωση που εκτελέστηκε, τα μετρούμενα μέσα σφάλματα των διατάξεων σημείων σφαίρας, ήταν υψηλότερα από τα μέσα σφάλματα των διατάξεων σημείων τριγωνικού πρίσματος, παρατήρηση η οποία επικυρώθηκε επίσης μέσω των παραπάνω σχημάτων. Τα μέσα σφάλματα που μετρήθηκαν παρουσιάζονται στον Πίνακα 1.3.



Σχήμα 1.23: Συγκρίνοντας τα Σφάλματα Μετάφρασης.



Σχήμα 1.24: Συγκρίνοντας τα Σφάλματα Περιστροφής.

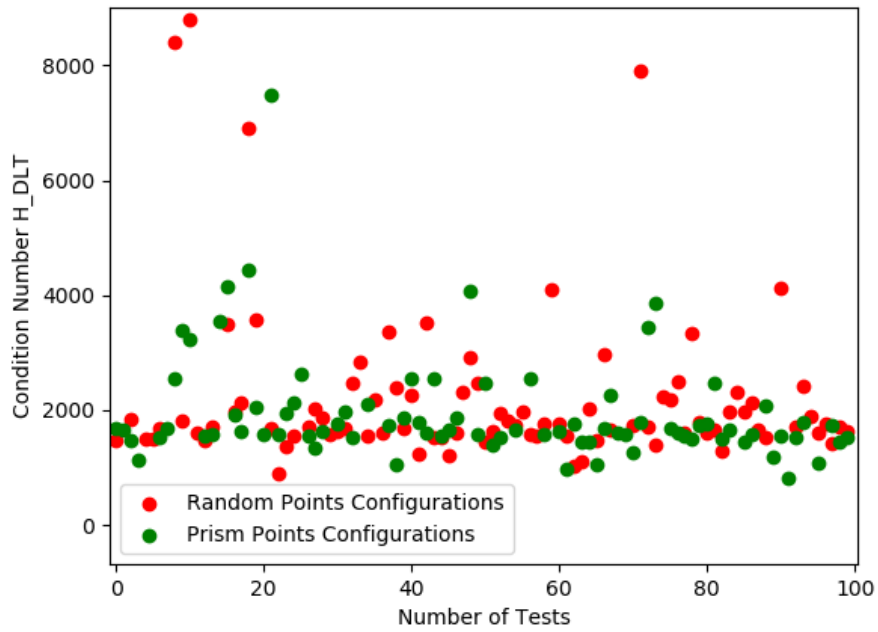
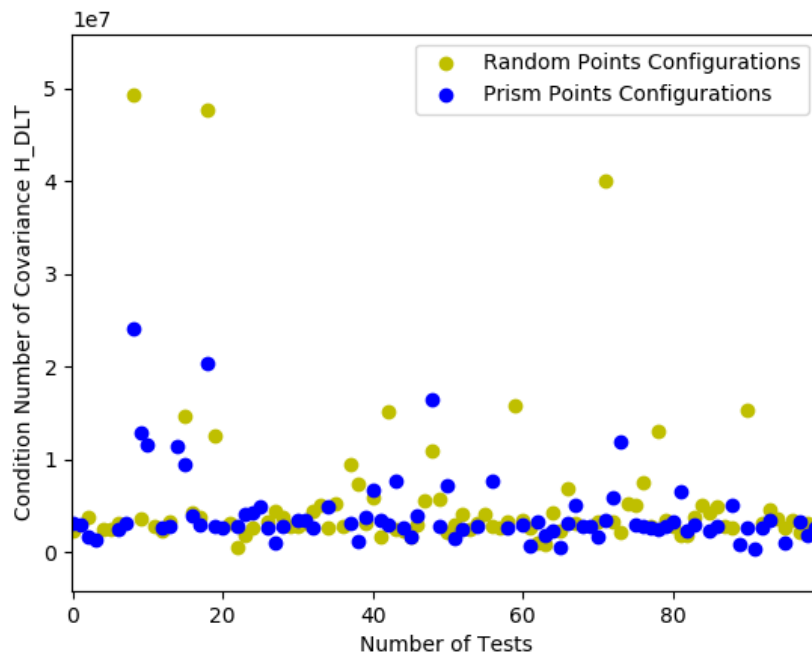


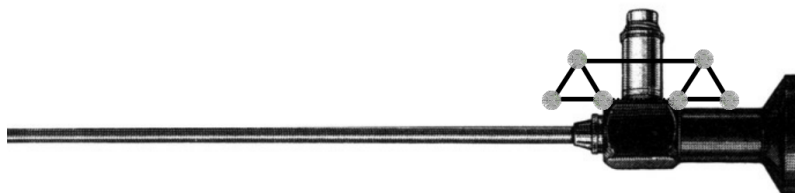
Σχήμα 1.25: Συγκρίνοντας τα Σφάλματα Επαναπροβολής.

Measured Mean Errors	Sphere Points Configurations	Prism Points Configurations
Mean Rotational Error [°]	37.8125	9.2809
Mean Error in Translation [%]	79.0060	14.1995
Mean Reprojection Error [cm]	0.02143	0.02116

Πίνακας 1.3: Μέσα Σφάλματα των διατάξεων Τυχαίων Σημείων Σφαίρας και των Σημείων Τριγωνικού Πρίσματος.

Τέλος μας έμειναν οι τελευταίες συγκρίσεις, δηλαδή του condition number και του πίνακα συνδιακύμανσης του  $H_{DLT}$ . Όπως φαίνεται αρχικά στο Σχήμα 1.26, στο 97 % των συγκρίσεων τα τυχαία σημεία της σφαίρας οδήγησαν σε μεγαλύτερο condition number από αυτό των σημείων τριγωνικού πρίσματος. Αντίστοιχα για τον πίνακα συνδιακύμανσης τα αποτελέσματα μας φαίνονται στο Σχήμα 1.27, όπου πάλι προέκυψε πως στο 95 % των συγκρίσεων τα σημεία που επιλέγονται τυχαία από τη σφαίρα οδηγούν σε μεγαλύτερες τιμές του, από αυτές των σημείων τριγωνικών πρισμάτων εγγεγραμμένων στην ίδια σφαίρα.

Σχήμα 1.26: Συγκρίνοντας τα Condition Number του  $H_{DLT}$ .Σχήμα 1.27: Συγκρίνοντας τα Condition Number των πινάκων συνδιακύμανσης του  $H_{DLT}$ .



Σχήμα 1.28: Η τρισδιάστατη διάταξη σημείων που προτείνουμε στο βασικό δείκτη προσαρμοσμένη σε ένα Λαπαροσκόπιο.

## 1.14 Συμπεράσματα

Στην παρούσα διπλωματική εξετάσαμε διάφορα σημαντικά ζητήματα που αφορούν στον οπτικό εντοπισμό αντικειμένων στο χώρο, στα οποία έχει τοποθετηθεί ένας βασικός δείκτης. Αρχικά παρουσιάσαμε τη βέλτιστη θέση της κάμερας σε σχέση με το βασικό δείκτη και καταλήξαμε πως πρέπει να σηματοδοτεί με αυτόν γωνία 90 μοιρών και να είναι σε κοντινή του απόσταση.

Στη συνέχεια αναζητήσαμε τις διατάξεις σημείων βασικών δεικτών που περιορίζουν σημαντικά όλα τα σφάλματα εντοπισμού, δηλαδή της μετάφρασης, της περιστροφής, της επαναπροβολής και τον αριθμό συνθήκης. Τα σημεία αυτά είχαν τη διάταξη ενός τριγωνικού πρίσματος.

Τέλος, εφαρμόσαμε τα αποτελέσματα αυτά σε μία Λαπαροσκοπική Χειρουργική Επέμβαση για τον εντοπισμό του Λαπαροσκόπειου και καταλήξαμε στην τελική διάταξη των σημείων που προτείνουμε στα πλαίσια αυτής της διπλωματικής και φαίνεται στο Σχήμα 1.28.

# Chapter 2

## Introduction

In this introductory chapter, the motivation that led to this particular topic is explained and afterwards the structure of the thesis is analyzed.

### 2.1 Motivation

Camera pose estimation is a critical challenge in many applications requiring a precise localization in the environment, such as Augmented Reality (AR) applications, robotics etc. Moreover, occlusion may be a problem, when a real object appears occluding the virtual scene. To overcome these challenges, the correspondences between known points in the environment and their camera projected points need to be accurately calculated [42]. To achieve that, fiducial markers<sup>1</sup> are mainly used, which are easily and precisely detected.

Furthermore, all known space resectioning methods are inherently non-linear and highly susceptible to noise, when they are expressed in image coordinates. Thus, the accuracy of the process depends on the configuration of the marker, which is non-predictable in applications where a fiducial marker is tracked in the 3 Dimensions (3D) space.

In AR applications, such as the use of fiducial markers in medicine for tracking an endoscope during surgical interventions, the configuration of the marker is crucial, because as the errors (*e.g.*, translation, rotation, reprojection) and condition number are decreased, the accuracy of the process is increased. In other applications, such as Printed Circuit Board (PCB) for automated equipment, fiducials help the machines to recognize the type and the orientation of each board.

Every AR method is required to guarantee that it will always augment the scene and not diminish it; *e.g.*, as AR is based on the absolute alignment of the virtual images models and the real object (*e.g.*, patient, PCB, robot, instrument). So, minimizing errors in orientation and reprojection is an important step to improve the AR. Thus, fiducials can be really helpful in terms of recognizing the position of an object in space.

---

<sup>1</sup>An object with pre-defined size, that appears in the captured image, and is used as a point of reference or a measure [34]

However, the accuracy of each fiducial can be improved, so it is required to examine whether there are specific points configurations for the fiducial marker in the 3D space, that are able to minimize orientation errors.

Pose estimation is one of the fundamentals in computer vision. From target tracking to parts assembly, the exact localization of the camera and the markers is required. Even though, there are many previous research regarding algorithms that can solve this problem, the shape, size and configurations of fiducials and their influence on the accuracy have not investigated further.

Thus in this thesis, we decide to analyze and test tracking systems, which are usually fiducial based, and thus, the 3D points configurations of fiducial markers chosen to be examined.

The main goal of this thesis is the identification of fiducial 3D marker configurations, which are able to increase the accuracy of the tracking process in a given 3D volume. The 3D point configurations that are better for pose estimation, can be used for localizing and moving robots.

We also examine the estimation of camera angles or points of view, which are better for point estimation in a given 3D space. These camera angles or points of view can be used to better place the robots relative to each other in order to improve the pose accuracy or relative to a fixed marker in the environment.

Moreover, medical applications, were also examined in this thesis, where these points configurations can be used for example to detect and estimate the position of medical devices (*e.g.*, a Laparoscope) during a medical procedure. Furthermore, the camera angles for better points estimation, can increase the accuracy of the Minimally Invasive Surgery (MIS).

Image-guided surgery, and specifically Augmented Reality (AR) in Laparoscopic Surgeries, are trying to overcome the above challenges, and reports have already shown advantages using AR as a guidance tool. Specifically, laparoscopic AR provides intraoperative guidance with the identification of subsurface targets and critical structures and displays cutting trajectories, which are virtualized beforehand. It also helps the surgeon to avoid mentally match information from different sources to the scene [7]. Laparoscopic Augmented Reality applications are considered more or less successful, however, there are challenges and limitations, which are needed to be overcome.

Tracking technology is one of the bottlenecks for augmented reality in general. In Laparoscopic Surgeries specifically, it is required to determine the pose of all intraoperatively utilized imaging devices (*i.e.*, endoscope), otherwise their position becomes unknown and the augmentation inaccurate [7].

This manuscript presents our contributions on the field of Pose Estimation, which are the following:

1. Finding 3D Points Configurations, which minimize errors in translation, rotation and reprojection, as well as, the condition number of the  $\mathbf{H}_{DLT}$  matrix.



2. Tracking the above errors and estimating the relation between accuracy and the position of the camera.
3. Applying the above results and estimations, simulating an optical tracking, fiducial-based system, to track an endoscope in a Laparoscopic Surgery.

## 2.2 Structure

This thesis is organized as follows:

- In Chapter 3, basic definitions and principles of this thesis are presented, as required fundamentals.
- In Chapter 4, the setup for this thesis simulations is explained.
- In Chapter 5, we analyze our results, solving the Perspective-n-Point problem in spherical coordinates. Moreover the error covariance ellipsoids for different camera poses are calculated and explained.
- Chapter 6 consists of our experiments and results, in the identification of 3D point configurations, which are minimizing errors in translation, rotation and reprojection.
- In Chapter 7, current applications of fiducial markers in Laparoscopy are presented.
- In Chapter 8, we apply our previous results for 3D Points Configurations simulating a laparoscopic surgery. Specifically, we use those results to improve the accuracy in optical tracking in MIS.
- Chapter 9 concludes this thesis and presents future outlook.

At the beginning of each Chapter, a summary of related works is analyzed, as a preliminary to our research.



## Chapter 3

# The Fundamentals

In this Chapter, we discuss some necessary preliminaries which will be used throughout this thesis. Starting from the definition of fiducial markers, a Camera Model and the calibration technique, then, the DLT, which is the main algorithm used in the thesis is analyzed. Finally, the estimated errors in orientation and the condition number are introduced.

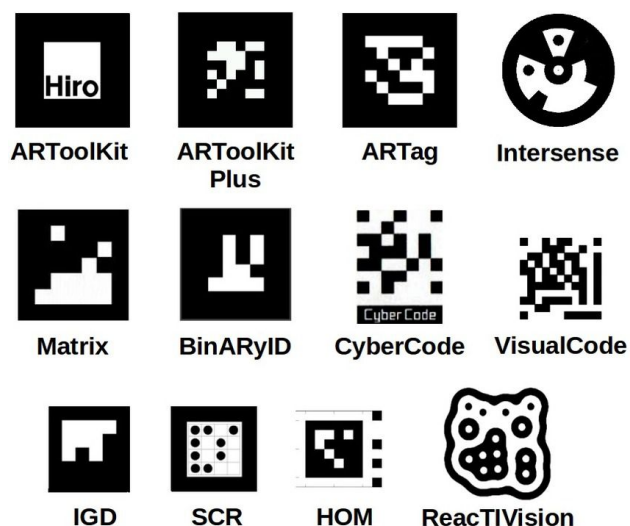
### 3.1 Fiducial Markers

A fiducial marker is an object that can be made of gold, carbon or polymer and its size may varies depending on the application [34]. It is placed either into or on the subject we captured during the image production and can be one or more marks in the reticle of an optical instrument. The main criteria to develop a fiducial marker [18] are:

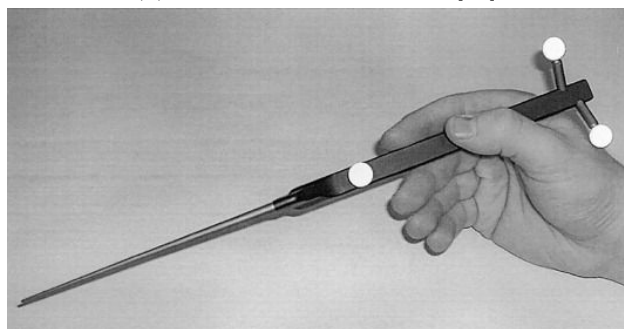
1. Robustness: Markers must be robustly detectable (*i.e.*, under varying lighting conditions).
2. Precision: The precise position of a marked object within an observed scene is provided.
3. Distinguish-ability: The ability of a tracking method to be scaled accordingly to the requirements imposed by the specific application or scenario.
4. Economic Feasibility: The need for cheap and easily producible markers.

Its shape and dimensions are well-known before its application and it is often used to define references in world coordinates which can be easily detected in camera images. Afterwards, using algorithms to solve the pose estimation problem we can find the camera position (translation and rotation) in world coordinates [28].

In AR applications, fiducial markers provide a useful tool to the integration between the real world view and the synthetic images that augment it. Generally, a fiducial marker system is composed by a set of valid markers and an algorithm for their detection, and possibly correction, in images. Such a system can be used in many applications related



(a) Planar fiducial markers [18].



(b) 3D fiducial marker system [15].

Figure 3.1: Examples of fiducial markers.

to computer vision, that require tracking and localization (*e.g.*, visual inspection, object recognition, camera calibration and passive navigation).

Focusing on biomedical applications and more specifically in AR in laparoscopic surgeries, fiducial markers are really helpful to solve tracking problems, as there is a need for more advanced visualizations that can enhance the view to include information below the surface of the organs (*i.e.*, palpate vessels, tumors, and organs during surgery) for planning of the procedure and for control and guidance during treatment.

In Figure 3.1 there are examples of fiducial markers, which are used for camera pose estimation in several applications. Specifically, Figure 3.1a shows different types of planar fiducial markers, while in Figure 3.1b, an application of 3D fiducials in Laparoscopy is presented.

## 3.2 Camera Model

In order to perform metric measurements with a camera, its extrinsic<sup>1</sup> ( $\mathbf{R}$ ,  $\mathbf{t}$ ) and intrinsic<sup>2</sup> parameters ( $\mathbf{K}$ ) need to be known. Extrinsic parameters define the location and orientation of the camera, while intrinsic parameters allow a mapping between camera coordinates and pixel coordinates in the image plane. Camera model in general is a mapping from world to image coordinates.

In computer vision, it is important to use the right frame of reference, *i.e.*, that measurements are made with respect to a particular coordinate system. As there are plenty of coordinate systems, in terms of a real object analysis, appropriate transformations are needed to convert each system to another. Specifically, the systems are the following [11]:

1. Homogeneous coordinate system<sup>3</sup> ( $u$ ,  $v$ ); measured in px.
2. Image plane coordinate system<sup>4</sup> ( $x$ ,  $y$ ); measured in m.
3. Camera coordinate system<sup>5</sup> ( $X_c$ ,  $Y_c$ ,  $Z_c$ ); measured in m.
4. World coordinate system<sup>6</sup> ( $X_w$ ,  $Y_w$ ,  $Z_w$ ); measured in m.

Image points are expressed either in image plane coordinates ( $x$ ,  $y$ ) or in homogeneous coordinates ( $u$ ,  $v$ ), which were chosen in this thesis. Thus, image points are written as follows [20]:

$$\mathbf{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}. \quad (3.1)$$

Moreover real world 3D points are expressed as follows [20]:

$$\mathbf{X} = \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}. \quad (3.2)$$

A camera model is presented in Figure 3.2. The image points are projected in the focal plane, and they are discretized in pixels in the image plane.

$\mathbf{K}$  is an upper triangular matrix, contains 3 different parameters (focal length, image sensor format, and principal point) and can be calculated through camera calibration [20]. The equation for the  $\mathbf{K}$  is:

$$\mathbf{K} = \begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}, \quad (3.3)$$

---

<sup>1</sup>Camera's pose on the world.

<sup>2</sup>The internal parameters of a camera.

<sup>3</sup>A coordinate system that measures pixel locations in the image plane.

<sup>4</sup>A plane on which the image is formed.

<sup>5</sup>A coordinate system that uses the camera center as its origin.

<sup>6</sup>A fixed coordinate system for representing objects in the world.

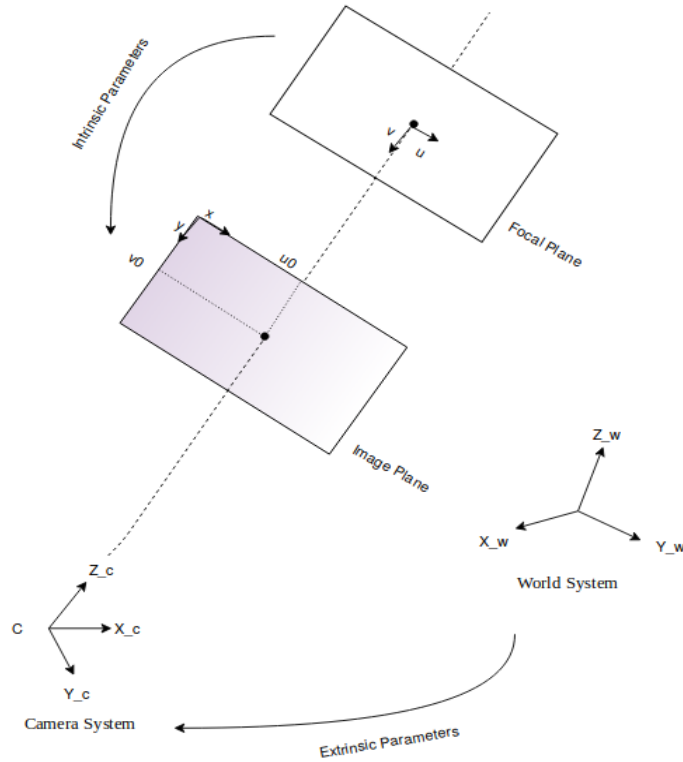


Figure 3.2: Camera model.

where  $s$  is the skew<sup>7</sup>,  $u_0$  and  $v_0$  are the coordinates of the principal point offset<sup>8</sup>, while  $f_x$  and  $f_y$  represent focal length<sup>9</sup> in terms of px.

Additionally, the extrinsic camera parameters are expressed using:  $R_{3 \times 3}$ , which represents the rotation (*i.e.*, rotates corresponding axes of each frame into each other) and  $t_{3 \times 1}$ , which represents the translation (*i.e.*, defines relative positions of each frame).

Using these matrices the position of the camera center ( $C$ ) expressed in world coordinates can be estimated as [20]:

$$C = -R^{-1}t = -R^T t. \quad (3.4)$$

Knowing the world coordinates of an object, as well as, the extrinsic and intrinsic parameters of the used camera, we can find the pixel coordinates. To achieve that a projection matrix  $P = K[R \mid t]$ , is used [20], with  $R$  and  $t$  being the Rotation and translation matrices. Then image points can now be found as:

$$x = P X. \quad (3.5)$$

<sup>7</sup>Axis skew causes shear distortion in the projected image [26].

<sup>8</sup>The "principal point offset" is the location of the principal point relative to the film's origin [26].

<sup>9</sup>Focal Length is the distance between the pinhole and the image plane [26].

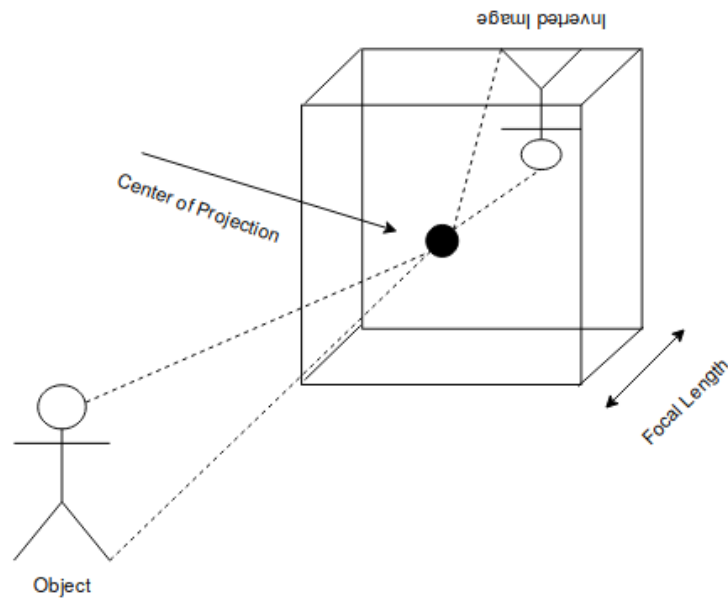


Figure 3.3: The Pinhole Camera Model.

### 3.3 Calibrating Camera

Camera calibration is a geometric process [12], that estimates the parameters of lens and image sensor of an image or video camera (*e.g.*, focal length, skew). It is also called camera resectioning. All the parameters, which are calculated through the calibration, are used to minimize lens distortion or determine the location of the camera in the scene.

Camera calibration is required for solving the pose estimation problem, but also in 3D scenes reconstruction. Through this geometric process all the parameters, which are needed for the final  $\mathbf{K}$  matrix of the camera are estimated. In this thesis, we assume that the used camera is calibrated.

### 3.4 A Pinhole Camera

A real pinhole camera [5], has a shape of a closed box with a single small opening (pinhole) and due to the rectilinear propagation of light, a reversed image of the outside space on the opposite side of the box is created. The pinhole diameter is depending on the focal length and the wavelength of light.

In a pinhole camera both  $f_x$  and  $f_y$  from the equation (3.3) have the same value, the principal point is defined as the centre of the image plane and the skew is considered zero. A pinhole camera model, is shown in Figure 3.3.

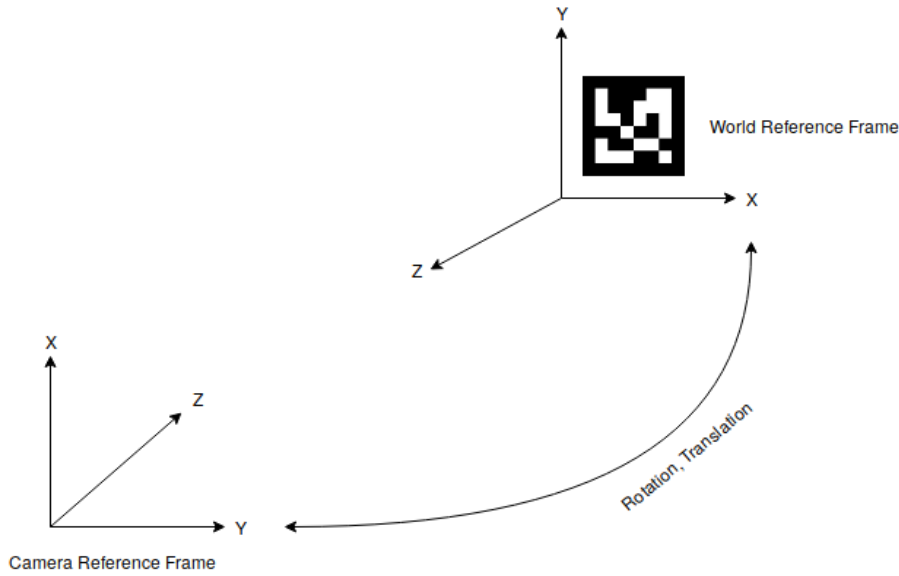


Figure 3.4: Camera Pose Estimation.

### 3.5 Camera Pose Estimation

”Pose estimation is a problem which found its origin in photogrammetry, where it is known as space resection” [29].

Both in computer vision and robotics a crucial challenge is to identify objects in an image in relation to their orientation in the real world, *i.e.*, from image coordinates to real world coordinates. In terms of pose estimation, both position and orientation must be identified, as this information is being used to navigate a robot, such as a surgical robot [11].

Image data can either be a single image, a pair or sequence of images. In this thesis, we decide to use a single image that is captured by a single camera. Moreover, the real world objects may be anything, *e.g.*, a whole human being, just a hand or even a finger.

The pose of the camera is measured, by using two different matrices. The first one describes the rotation and the other the translation as shown in Figure 3.4.

Camera’s pose can be estimated with several algorithms, such as: Efficient Perspective-n-Point (EPnP); EPnP Iterative. In this thesis, we decide to use the Direct Linear Transformation (DLT) algorithm.

### 3.6 Perspective-n-Point

PnP is the problem of estimating the pose of a calibrated camera given a set of 3D world points and their projected 2 Dimensions (2D) image points [41]. The analysis of the equation (3.5) for the perspective project model for cameras, in addition with the



equations (3.1), (3.2) and (3.3) gives:

$$\rho \underbrace{\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}}_{\mathbf{x}} = \underbrace{\begin{bmatrix} f_x & s & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix}}_{\mathbf{K}} \underbrace{\begin{bmatrix} R_{1,1} & R_{1,2} & R_{1,3} & | & t_1 \\ R_{2,1} & R_{2,2} & R_{2,3} & | & t_2 \\ R_{3,1} & R_{3,2} & R_{3,3} & | & t_3 \end{bmatrix}}_{\mathbf{R}, \mathbf{t}} \underbrace{\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}}_{\mathbf{X}}, \quad (3.6)$$

which is called the PnP problem model.

The camera pose consists of 6 Degrees-Of-Freedom (DOF) depending on Rotation and translation of the camera with respect to the world coordinate system.

The problem stems from camera calibration and has a variety of applications in robotics as well as in computer vision (*e.g.*, 3D pose estimation, AR).

### 3.7 Direct Linear Transformation

DLT is an algorithm that determines the orientation (rotation and translation) of a single image without knowing the camera parameters *a priori*. DLT requires at least 4 points to operate in 2D cases and at least 6 points for 3D cases, as the ‘‘Gold Standard’’ algorithm developed by Hartley and Zisserman determines [20].

In case of applying 3D world-points on the DLT, 6 different pairs of equations are created as shown below [8]:

$$u_i = \frac{m_{00}X_i + m_{01}Y_i + m_{02}Z_i + m_{03}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \quad (3.7)$$

$$v_i = \frac{m_{10}X_i + m_{11}Y_i + m_{12}Z_i + m_{13}}{m_{20}X_i + m_{21}Y_i + m_{22}Z_i + m_{23}} \quad (3.8)$$

The equations (3.7) and (3.8) lead us to the following homogeneous set of equations:

$$\underbrace{\begin{bmatrix} X_i & Y_i & Z_i & 1 & 0 & 0 & 0 & 0 & -u_i X_i & -u_i Y_i & -u_i Z_i & -u_i \\ 0 & 0 & 0 & 0 & X_i & Y_i & Z_i & 1 & -v_i X_i & -v_i Y_i & -v_i Z_i & -v_i \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_n & Y_n & Z_n & 1 & 0 & 0 & 0 & 0 & -u_n X_n & -u_n Y_n & -u_n Z_n & -u_n \\ 0 & 0 & 0 & 0 & X_n & Y_n & Z_n & 1 & -v_n X_n & -v_n Y_n & -v_n Z_n & -v_n \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} m_0 \\ m_1 \\ \vdots \\ m_{10} \\ m_{11} \end{bmatrix}}_{\mathbf{m}} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 0 \end{bmatrix}. \quad (3.9)$$

To solve that system of equations we take advantage of the Singular Value Decomposition (SVD) [2] of the  $\mathbf{A}$  and we find the  $\mathbf{m}$  (12x1), as the last column of the  $\mathbf{V}$  vector of the decomposition. Then we resize  $\mathbf{m}$  to find the final  $H_{DLT}$ , from which we extract afterwards the  $\hat{\mathbf{R}}$  and  $\hat{\mathbf{t}}$ .

If noise is added to the image pixels, a normalization is required before applying these points on the DLT. According to [20], 3D reference points should be normalized so that their centroid will be at the origin and their Root Mean Square (RMS) deviation from

origin will be  $\sqrt{3}$ . For 2D imaged points the normalization is the same but the RMS deviation is  $\sqrt{2}$ .

DLT is a method, which is often used to solve the PnP problem.

### 3.7.1 Extracting the Rotation $\hat{\mathbf{R}}$

$\hat{\mathbf{R}}$  is extracted from the first 3 columns of the final  $H_{\text{DLT}}$  matrix [27]. Knowing already the  $\mathbf{K}$  matrix from the calibration of the camera, a QR factorization [39] is needed to estimate the Rotation matrix. So,  $H_{\text{DLT}}$  matrix is analyzed as:

$$H_{\text{DLT},1:3} = \mathbf{K}\hat{\mathbf{R}} \implies H_{\text{DLT},1:3}^T = (\mathbf{K}\hat{\mathbf{R}})^T = \hat{\mathbf{R}}^T \mathbf{K}^T$$

The QR decomposition is implemented so that  $\hat{\mathbf{R}}^T \mathbf{K}^T = \mathbf{Q}\mathbf{B}$ , where  $\mathbf{Q}$  is the transpose of the Rotation matrix we wanted to extract. Thus, the final Rotation 3x3  $\hat{\mathbf{R}}$  is given by this equation:

$$\hat{\mathbf{R}} = \mathbf{Q}^T. \quad (3.10)$$

### 3.7.2 Extracting the Translation $\hat{\mathbf{t}}$

In order to find the estimated translation, the calculation of the camera center is required.  $H_{\text{DLT}}$  matrix encodes all the necessary camera parameters. According [8] there is a formula to find the camera center. If we analyze the final matrix from DLT as  $H_{\text{DLT}} = [\mathbf{Q} \mid \mathbf{b}]$  the camera center is:

$$\mathbf{C} = \begin{bmatrix} -\mathbf{Q}^{-1}\mathbf{b} \\ 1 \end{bmatrix}. \quad (3.11)$$

Then, the equation (3.4) is used and after finding the  $\hat{\mathbf{R}}^T$  and the camera center  $\mathbf{C}$ , the  $\hat{\mathbf{t}}$  is calculated.

## 3.8 Pose Estimation Errors

After applying the world- and image-points on the DLT the estimated rotation and translation matrices are extracted. Thus, the errors between the real values and the estimated ones can be measured.

### 3.8.1 Rotational Error

The Rotational Error (RE) is expressed in degrees and is defined as the minimal Rotation needed to align  $\hat{\mathbf{R}}$  to  $\mathbf{R}$ . It is obtained from the axis-angle representation of the  $\hat{\mathbf{R}}^T \mathbf{R}$  [21]. The RE is estimated as:

$$\text{RE} = \hat{\mathbf{R}}\mathbf{R}^{-1}, \quad (3.12)$$

and to measure the final axis-angle, expressed in degrees, the following equation is used:

$$\text{axis-angle} = \arccos\left(\frac{\text{trace}(RE) - 1}{2}\right). \quad (3.13)$$

### 3.8.2 Translation Error

The relative Translation Error (TE) is expressed as a percentage (%) and it is given by this equation [21]:

$$\text{TE} = \|\hat{\mathbf{t}} - \mathbf{t}\|_2 / \|\mathbf{t}\|_2 \cdot 100\% \quad (3.14)$$

### 3.8.3 Root Mean Square Error

RMS error, is defined as the root of the mean of the squared absolute errors. It is an absolute error measure and it calculates the error relative to a known ground truth value [6]. The RMS error is given by the following equation:

$$e_{RMS} = \sqrt{\frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|^2}, \quad (3.15)$$

where  $y_i$  is the observed value for the  $i^{\text{th}}$  observation and  $\hat{y}_i$  is the predicted value. The RMS error is measured on the same scale, with the same units as  $y$ .

### 3.8.4 Reprojection Error

The Reprojection error is a geometric error corresponding to the image distance between a projected point and a measured one. It is used to quantify how closely an estimate of a 3D point recreates the point's true projection [20].

It can be measured by denoting the Euclidean distance between the  $\mathbf{x}$  and  $\hat{\mathbf{x}}$  of the image, as defined in (3.1).  $\hat{\mathbf{x}}$  is calculated, using  $\hat{\mathbf{P}} = \mathbf{K}[\hat{\mathbf{R}}|\hat{\mathbf{t}}]$  instead of the real extrinsic parameters and as it is expressed in px we need to convert it to find the final error in m.

Euclidean distance for each point is measured and then, the Reprojection error is the sum of those distances. If no noise is added to the image points the Reprojection error will be almost zero. To find each distance the following equation is used :

$$\text{Distance} = \sum_{i=1}^n \sqrt{(u_i - \hat{u}_i)^2 + (v_i - \hat{v}_i)^2}, \quad (3.16)$$

where  $n$  is the number of image points.

The final Reprojection error is measured in px and thus if we need to represent it in meters we should first converting px into m.

### 3.8.5 Condition Number

The condition number measures how sensitive the answer is to perturbations in the input data and to roundoff errors made during the solution process [31].

As some matrices are very sensitive to small changes in input data, the extent of this sensitivity is measured by the condition number. Hence, condition number of one random  $\mathbf{A}$  matrix is:

$$\text{cond}(\mathbf{A}) = \max \left( \frac{\text{Relative Error of Output}}{\text{Relative Error of Inputs}} \right), \quad (3.17)$$

or it can be also expressed as:

$$\text{condition number} = \|A^{-1}\| * \|\mathbf{A}\| \quad (3.18)$$

The accuracy function is depending in general on extrinsic and intrinsic camera parameters, planar marker, camera pose and also the condition number, as showing below:

$$\text{Accuracy} = f(\mathbf{K}, \mathbf{R}, \mathbf{t}, \text{marker size}, \text{condition number}).$$

### 3.9 Covariance Matrix

A covariance matrix give us the direction of the linear relationship between variables. It is a square matrix, often used in probability theory and statistics, given by the equation:

$$\Sigma_B = \frac{1}{n-1} \sum_{i=1}^n (B_i - \bar{B}_i)(B_i - \bar{B}_i)^T, \quad (3.19)$$

where  $B_i$  is a set of data that we examine and  $\bar{B}_i$  is the mean of this dataset [23].

The covariance matrix can be used to estimate the error covariance ellipsoids.

### 3.10 Error Covariance Ellipsoids

A confidence ellipsoid allows the visualization of a selected confidence level. In general an ellipsoid is defined by the following equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1, \quad (3.20)$$

which leads to the covariance ellipsoid equation:

$$\frac{x^2}{\lambda_x} + \frac{y^2}{\lambda_y} + \frac{z^2}{\lambda_z} = sc^2 \implies \frac{x^2}{sc\sqrt{\lambda_x}} + \frac{y^2}{sc\sqrt{\lambda_y}} + \frac{z^2}{sc\sqrt{\lambda_z}} = 1, \quad (3.21)$$

where  $\lambda_x$ ,  $\lambda_y$  and  $\lambda_z$ , are the eigenvalues of the covariance matrix and  $sc$  is the scale related to the confidence level of the ellipsoid.

The relation between confidence level and the scale  $sc$  in the different dimensions, according to [6], is presented in the Table 3.1.

Confidence Level						Dimensions
25 %	50%	75%	95%	97 %	99%	
0.10153	0.45494	1.3233	3.84146	4.70929	6.6349	n = 1
0.57536	1.38629	2.77259	5.99146	7.01312	9.2103	n = 2
1.21253	2.36597	4.10834	7.81473	8.94729	11.3449	n = 3

Table 3.1: Relation between scale and confidence level in different dimensions.

Furthermore, the principle axes of the ellipsoid have length, which is given by:

$$l_i = sc\sqrt{\lambda_i}, \quad \forall i = 1, 2, 3 \quad (3.22)$$

## 3.11 Back Propagation

Back propagation algorithms are used to train networks (*e.g.*, neural networks), following a gradient descent approach. Back propagation is showing whether a prediction was mistaken or not.

Back propagation algorithms update the the weights to improve the network until it is able to perform the task for which it is being trained. At the beginning, weights are randomly initialized and by a default loss function, which measures the discrepancy between the expected output and the actual one, the weights are updated and the network is trained [35].

In other words, Back propagation takes the error associated with a wrong guess by a neural network, and uses that error to adjust the neural network's parameters in the direction of less error [32].

In this thesis Back Propagation techniques are used to solve optimization problems, with the condition number being the optimization objective.

### 3.11.1 SuperSAB Algorithm

The SuperSAB algorithm is an acceleration strategy for error back propagation learning [37]. It works as shown below in algorithm 1, to estimate the factors and weights in order for the network to be well trained.

The step of the SuperSAB is changing only when the derivative of a weight has changed its sign. Then the step is decreased as far as it does not cause any change at the weight derivative's sign.

Where,  $step_{start}$  is the starting step,  $n_{start}$  is the starting factor,  $n_{decrease}$  is the decrease factor,  $w_{i,j}$  is each weight.

### 3.11.2 Gradient Descent

Gradient Descent is a first-order iterative optimization algorithm, which enables to calculate the local minimum points of a function [36]. At first a random point of the

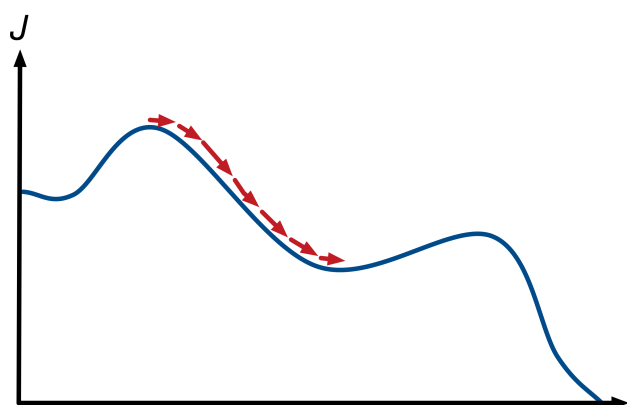
**Algorithm 1** SuperSAB**Input** :  $step_{start}$ ,  $w_{i,j}$ ,  $n_{decrease}$ ,  $n_{start}$ **Output**: Estimated  $n$  factors

1. Initialize all values for  $n_{i,j} \leftarrow n_{start}$
2. Do Back Propagation with  $step_{start}$
3. **while**  $dw_{i,j} \dot{=} 0$  **do**
  - 4.  $n_{i,j}^{k+1} \leftarrow n_{decrease} + n_{i,j}^k$
- end**
5. Undo the previous weight update
6.  $n_{i,j}^{k+1} \leftarrow n_{decrease} - n_{i,j}^k$

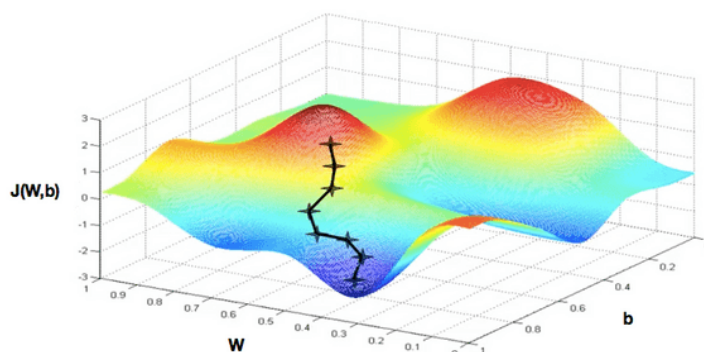
function is chosen and the evaluation of the gradients begins by following the negative of the gradient of the function until a local or a global minimum is reached.

In each evaluation(iteration), there is a specific rate, called learning rate, which is set as the size of every step. If this rate is small we move along to the direction of the negative gradient. Moreover, a cost function is estimated, which shows the accuracy of the function to predictions. It is often used in machine learning in order to update the parameters of a model [10].

In Figure 3.5a there is an example of the Gradient Descent in a 2D case problem. In the 2D case the learning rate is evaluated in dependence of  $x$  and  $y$  axis and the selected point is changing in every iteration until no changes are caused, meaning there is a local or a global minimum. As we can see from the length of the step vector, it is changing according to the learning rate. Moreover, in the 3D case, which is shown in Figure 3.5b, the learning rate depends on  $x$ ,  $y$  and  $z$  axis.



(a) Gradient Descent in 2D Case [4].



(b) Gradient Descent in 3D Case [10].

Figure 3.5: Gradient Descent Optimization.

### 3.11.3 Adam Optimization

Adam optimizer, which is an optimization algorithm, extension to stochastic gradient descent, that can be used instead of the classical stochastic gradient descent procedure to update network weights iterative based in training data [25].

Adam has four configuration parameters:

- alpha: Learning rate or step size
- beta1: Exponential decay rate for the first moment estimates
- beta2: Exponential decay rate for the second moment estimates
- epsilon: Small value preventing division by zero

Both beta1 and beta2 are required as Adam, instead of adapting the parameter learning rates based on the mean of the first moment, makes use of the mean of the second moments of the gradients [9]. A learning rate is also often used in Adam, which is maintained for each parameter and separately adapted as learning unfolds.

## 3.12 Euler Angles

The Euler angles are three angles, which describe the orientation of a rigid body with respect to a fixed coordinate system. They are also used to represent the orientation of a general basis in 3D linear algebra.

These three angles can be found from the original Rotation matrix  $\mathbf{R}$ , that was defined in the Camera-class, as follows [6]:

$$a = \arctan \frac{R_{3,2}}{R_{3,3}}, \quad (3.23)$$

$$b = -\arcsin R_{3,1}, \quad (3.24)$$

$$c = \arctan \frac{R_{2,1}}{R_{1,1}}. \quad (3.25)$$

These angles are used to find the Rotation matrix expressed by Euler angles as shown below:

$$\mathbf{R}_a = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos a & -\sin a \\ 0 & \sin a & \cos a \end{bmatrix}, \quad (3.26)$$

$$\mathbf{R}_b = \begin{bmatrix} \cos b & 0 & \sin b \\ 0 & 1 & 0 \\ -\sin b & 0 & \cos b \end{bmatrix}, \quad (3.27)$$

$$\mathbf{R}_c = \begin{bmatrix} \cos c & -\sin c & 0 \\ \sin c & \cos c & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (3.28)$$

Using the matrices  $\mathbf{R}_a$ ,  $\mathbf{R}_b$ , and  $\mathbf{R}_c$ , we obtain the Euler Rotation matrix

$$\mathbf{R}_{\text{Euler}} = \mathbf{R}_c \mathbf{R}_b \mathbf{R}_a. \quad (3.29)$$

### 3.13 Gaussian (Normal) Distribution

The Gaussian distribution is used in many engineering and optimization applications and is defined by applying the following values [3]:

- Mean ( $\mu$ ): the square root of the variance,
- Standard deviation ( $\sigma$ ): the average deviation,

on the equation:

$$f_X(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-(x-\mu)^2/2\sigma^2}, \quad \forall x \in (-\infty, \infty) \quad (3.30)$$

Base on this equation, the Gaussian Noise is defined, as a statistical noise equal to the Gaussian distribution. The Gaussian noise is represented as:  $X \sim \mathcal{N}(\mu, \sigma)$ .

### 3.14 Convert Cartesian to Spherical Coordinates

The Cartesian Coordinates ( $x, y, z$ ) can be converted to Spherical ( $r, \alpha, \beta$ ), using the following equations:

$$\alpha = \arccos\left(\frac{z}{r}\right), \quad (3.31)$$

$$\beta = \arctan\left(\frac{y}{x}\right), \quad (3.32)$$

$$r^2 = x^2 + y^2 + z^2, \quad (3.33)$$

where  $r$  is the radius of the sphere or radial distance,  $\alpha$  is called inclination or polar angle and it is limited from -90 to 90 degrees;  $\beta$  is the azimuth or azimuthal angle and it is limited from 0 to 360 degrees.



## Chapter 4

# Simulation Setup

In this Chapter, the fundamentals that were analyzed before are applied to our specific simulations, executed for this thesis. Starting by setting values of the matrices for the camera model, then the real-world object is defined. Thereafter, the DLT algorithm, that is used for the 3D case is analyzed and the measured pose estimation errors are explained. All of the algorithms and function, which are developed for this thesis are available in Github following this path: `/D1vt/DLT3D/Python`.

### 4.1 Camera Model

In this thesis, we use a Camera-class, by defining a pinhole camera model along with its extrinsic ( $\mathbf{R}$ ,  $\mathbf{t}$ ) and intrinsic ( $\mathbf{K}$ ) parameters. However, our pinhole camera model, does not include, geometric distortions or blurring of unfocused objects.

Starting with defining  $\mathbf{K}$  matrix:

$$\mathbf{K} = \begin{bmatrix} 800 & 0 & 640 \\ 0 & 800 & 480 \\ 0 & 0 & 1 \end{bmatrix},$$

where  $u_0$  is 640, because the image plane's width is defined as 1280 px, while  $v_0$  is 480, as the image plane's height is set 960 px. The above  $\mathbf{K}$  matrix is used in every simulation that will be presented from now on in this thesis.

Afterwards, the rotation and translation matrices are defined as follows:

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & -2.3512749 \times 10^{-16} \\ 0 & 2.35127499 \times 10^{-16} & -1 \end{bmatrix},$$

$$\mathbf{t} = \begin{bmatrix} 0 \\ 1.1756375 \times 10^{-16} \\ 0.5 \end{bmatrix}.$$

The camera model, which is expressed from the above matrices, is placed on the top of the marker.

## 4.2 Sphere Real-World Object

All real-world points in this thesis, belong to a specific sphere with a radius of 0.3 m. This radius was chosen as it is needed to be guaranteed that every possibly selected world-point, will be able to be projected in the specific image plane, with a width of 1280 px and a height of 960 px.

A class is developed that creates a sphere, and has many different function (*e.g.*, to rotate it, change its color or its origin). Randomly points can also be selected from this sphere, using the algorithm, that takes as arguments the number of random points we want to generate (the default number is 6), the radius of the sphere, the minimum distance that we want the points to have from each other. There is also a Boolean variable argument, which is called *periphery* and it determines whether the points will be randomly selected from the sphere or only from the periphery of the sphere. The algorithm that creates the real-world points is shown below 2:

---

### Algorithm 2 Sphere Points Selection

---

**Input** : points, periphery, min\_distance, radius

**Output**:  $\mathbf{X} = [X_i, Y_i, Z_i, 1]^T \forall i = 0, 1, 2, \dots, \text{points}$

1. **if** *periphery* = *True* **then**

    2. Do  $r \leftarrow \text{radius}$

**end**

3. **else**

    4. Do  $r \leftarrow \text{random number} \in (0, \text{radius})$

**end**

5. **while**  $i \leq \text{points}$  **do**

    6. Select random  $\alpha \in (-\frac{\pi}{2}, \frac{\pi}{2})$

    7. Select random  $\beta \in (0, 2\pi)$

    8. Find  $\mathbf{X}$  using the equations (3.33)

    9.  $i \leftarrow i + 1$

    10. Find Euclidean Distances between this  $\mathbf{X}$  and each previous  $\mathbf{X}$  using the equation (3.16)

    11. **if** *Euclidean Distance*  $\leq \text{min\_distance}$  **then**

        12. Remove this  $\mathbf{X}$

        13.  $i \leftarrow i - 1$

**end**

**end**

---

An example of 1000 randomly selected sphere points, which are generated from the function mentioned above is shown in Figure 4.1. The pose of the camera, which is looking straight down into the sphere model is also presented in the same Figure. To create this plot, the mayavi python tool was used [33].

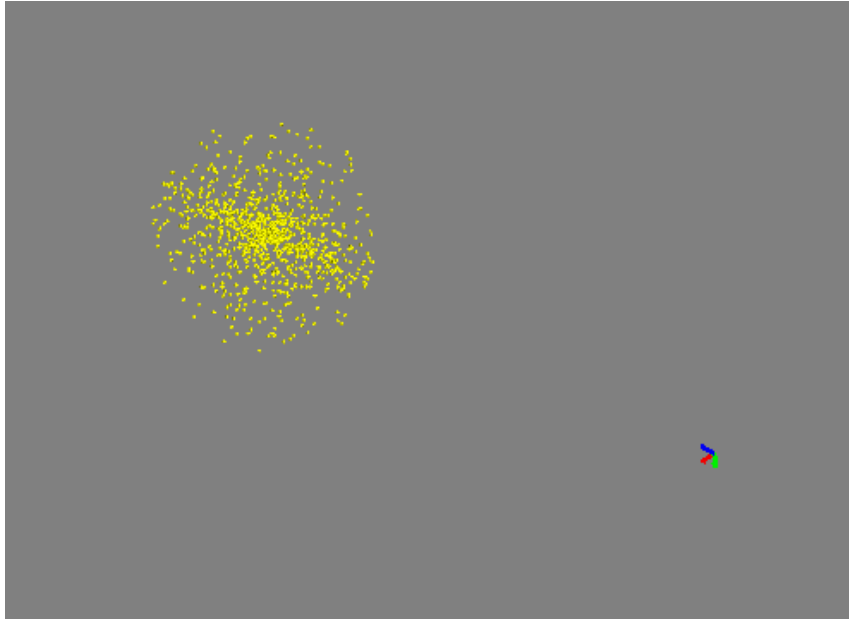


Figure 4.1: Plot 1000 random Sphere Points using Mayavi.

### 4.3 Using Direct Linear Transformation to the Perspective-n-Point Problem

In this thesis, we are working on the 3D space, so we use 6 randomly selected real world points for our experiments. All these points belong to the same sphere plane with a radius of 0.3 m. Thus, from the previous defined sphere, these points are selected in each one of our tests.

After the points selection, we calculate their projected image points expressed in homogeneous coordinates  $(u, v)$ . Then, these points are applied on the algorithm, which is shown below in 3 and the final  $\mathbf{H}_{\text{DLT}}$  is calculated. As in many cases a normalization is required before applying the points on the DLT, a cross-validation of whether our points are normalized or not is needed. If they are normalized, a denormalization is required to the  $H_{\text{DLT}}$  matrix in order to find the final one, that will be used to extract the  $\hat{\mathbf{R}}$  and  $\hat{\mathbf{t}}$  matrices.

### 4.4 Pose Estimation Errors

We use the pinhole camera-class and we randomly select 3D world points, belonging to the sphere plane mentioned before. Using these points we estimate their projected image points and we apply them on the DLT algorithm to calculate the  $\hat{\mathbf{R}}$ ,  $\hat{\mathbf{t}}$ .

Afterwards, we measure the translation and Rotation errors between  $\hat{\mathbf{R}}$ ,  $\hat{\mathbf{t}}$ , estimated from DLT and the real ones. The Reprojection error is also measured, as it is required for the image resection, as well as the condition number of the covariance matrix of the  $H_{\text{DLT}}$ . The error covariance ellipsoids for different positions of the camera are also measured.

---

**Algorithm 3** Direct Linear Transformation
 

---

**Input** :  $\mathbf{X} = [X_i, Y_i, Z_i, 1]^T$ ,  $\mathbf{x} = [u_i, v_i, 1]^T$ 
**Output:**  $H_{\text{DLT}}$ 

1. **if** *Normalization is needed* **then**
    - | 2. Do normalize  $\mathbf{X}$ ,  $\mathbf{x}$
  - end**
  3. **for**  $i = M$  **do**
    - | 4. Create  $\mathbf{A}$  applying each  $\mathbf{X}$ ,  $\mathbf{x}$  in the equation (3.9)
  - end**
  5. Decompose  $\mathbf{A}$ , using SVD, into  $\mathbf{UDV}^T$
  6. Minimize  $\|\mathbf{Am}\|$  by setting  $\mathbf{m} \leftarrow$  last column of  $\mathbf{V}$
  7. Reshape  $\mathbf{m}$  from 12x1 into the 3x4  $H_{\text{DLT}}$  final matrix
  8. **if** *Normalization was required at the beginning* **then**
    - | 9. Do denormalize to find the final  $H_{\text{DLT}}$
  - end**
- 

Moreover, as this thesis goal is to find the configurations of points that lead to the minimum orientation errors, the condition number of  $H_{\text{DLT}}$  matrix is measured, as the “well-conditioned” matrices lead to decreased errors.

## Chapter 5

# Analyzing PnP in Spherical Coordinates

In this Chapter we express the  $\mathbf{R}$  and  $\mathbf{t}$  camera matrices using spherical coordinates, in order to find the covariance matrix for detecting the marker. Then, we examine the relation between accuracy and camera's distance from the marker, as well as, the relation between accuracy and the angle view of the camera. We also calculate the covariance error ellipsoids for different positions of the camera.

### 5.1 Preliminaries

In [6], the accuracy of monocular tracking has been examined. Even though, there are not any tests for the marker detection, which is the main part of this thesis, it analyzes the geometric setup of the marker and the camera and uses the error covariance matrix to estimate the accuracy.

Using a pinhole camera model, always centered towards the origin where the marker is placed and assuming that the camera parameters are all known without error, the error covariance matrix is calculated.

As concluded, the markers should be positioned in such a way that the camera can look at the marker at a certain angle view and not frontal. Those results were also verified using Monte-Carlo simulations.

The dependence of accuracy on the camera's distance from the marker was also analyzed. It was found, that the positional error of a monocular tracking system increases more than linearly as the distance increased.

Hence, we also test all the above, in our simulated case, where the marker points are belonging to the sphere class we defined. By expressing the camera extrinsic parameters in spherical coordinates, we find the covariance matrix, which is used for the error covariance ellipsoids. Additionally, the relation between the condition number and the angle view, as well as, the dependence of the distance on the condition number are also examined.

## 5.2 Covariance matrix for detecting the marker

Using the same pinhole camera model as described before, we assume that all the camera parameters are known and defined in our class without errors, and the target is fixed, while the camera is moving on a sphere with a specific radius  $r$ . We are now defining 12 nonlinear camera equations based on the equation (3.5), since the DLT algorithm needs at least 6 points to work properly [6]:

$$q : \begin{array}{l} u_i = \frac{1}{\rho_i} \mathbf{P} x_i \\ \vdots \\ v_i = \frac{1}{\rho_i} \mathbf{P} y_i \\ \vdots \end{array} \quad \forall i = 1, 2, \dots, 6 \quad (5.1)$$

Then, random Gaussian noise, with a standard deviation of 4 and a mean value of 0, is added to each of the  $x_i$ ,  $y_i$ , pixel coordinates in those nonlinear equations.

So according to [6], without loss of generality we parameterize the extrinsic camera parameters using three parameters  $\alpha$ ,  $\beta$  and  $r$ , representing the pose of the camera related to the marker in spherical coordinates, as shown in Figure 5.1. Thereafter,  $\alpha$ ,  $\beta$  and  $r$  are calculated by converting Cartesian to spherical coordinates, using the equations (3.33), where  $x$ ,  $y$ ,  $z$  are defined by the world position of the camera.

Afterwards, we use the  $\mathbf{K}$  camera matrix as defined in 3.4, and the  $\mathbf{R}$ ,  $\mathbf{t}$  matrices, converted in spherical coordinates. Then, the derivatives of  $\alpha$ ,  $\beta$ ,  $r$  are calculated, to be used for measuring the Rotation and translation matrices to find the Jacobian matrix of the  $q$ , as defined in equations (5.1). To find the specific  $\alpha_0$ ,  $\beta_0$ ,  $r_0$ , where the Jacobian matrix will be evaluated, we need to calculate the  $x_0$ ,  $y_0$ ,  $z_0$  by estimating the global position of the camera.

Hence, we have to find the dot product of  $\mathbf{R}\mathbf{t}$  and the vector  $[0, 0, 0, 1]^T$ . Then,  $x_0$ ,  $y_0$ ,  $z_0$  are the rows of the dot product and using the equation (3.33) the  $\alpha_0$ ,  $\beta_0$ ,  $r_0$  are calculated.

Thereafter, the Jacobian matrix of the  $q$  evaluated at  $\alpha_0$ ,  $\beta_0$ ,  $r_0$  is estimated as:

$$\mathbf{J}_q = \left. \frac{\partial q}{\partial(\alpha, \beta, r)} \right|_{\alpha=\alpha_0, \beta=\beta_0, r=r_0} \in \mathfrak{R}^{2n \times 3}, \quad (5.2)$$

where  $n$  is the number of selected points, and thus, our final Jacobian matrix  $\in \mathfrak{R}^{12 \times 3}$ . To find the derivatives of  $q$ , we need to calculate separately each of the 12 non-linear equations. In each equation only the  $\mathbf{P}$  matrix is affected by the  $\alpha$ ,  $\beta$  and  $r$ , so it is the only matrix that we need to analyze and find its derivatives.

$\mathbf{P}$  is analyzed as:  $\mathbf{P} = \mathbf{K}[\mathbf{R}|\mathbf{t}]$ .

However,  $\mathbf{K}$  matrix does not depend on the parameters  $\alpha$ ,  $\beta$ ,  $r$ . Thus, to find the Jacobian of  $q$ , it is only required to calculate the derivatives of the Rotation and the translation matrices.

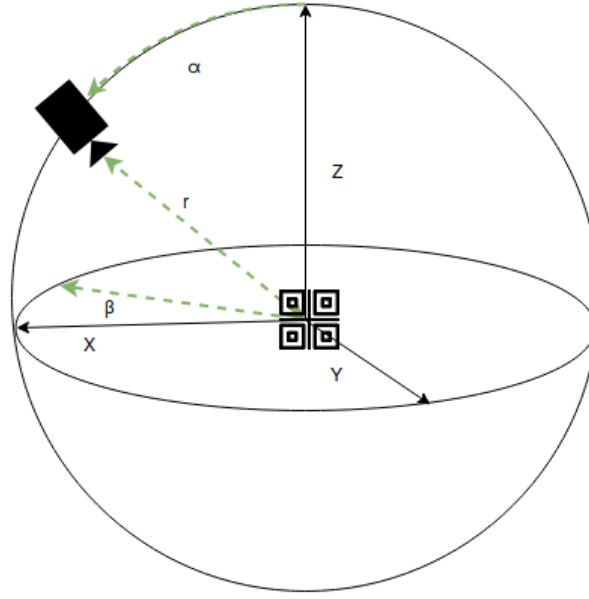


Figure 5.1: Navigation of camera on spherical coordinates.

By setting  $[(\mathbf{R})|(\mathbf{t})] = (\mathbf{Rt})$ , the equations in (5.1) are now analyzed as:

$$u_i = \frac{1}{\rho_i} \mathbf{K}(\mathbf{Rt})x_i, \quad \forall i = 1, 2, \dots, 6, \quad (5.3)$$

$$v_i = \frac{1}{\rho_i} \mathbf{K}(\mathbf{Rt})y_i, \quad \forall i = 1, 2, \dots, 6. \quad (5.4)$$

Thereafter, to find the final Jacobian matrix we need to calculate the derivatives only for the  $(\mathbf{Rt})$  of the equations (5.3) and (5.4), as shown below:

$$\frac{\partial u_i}{\partial(\alpha, \beta, r)} = \frac{1}{\rho_i} \mathbf{K} \frac{\partial(\mathbf{Rt})_{spherical}}{\partial(\alpha, \beta, r)} x_i, \quad (5.5)$$

$$\frac{\partial v_i}{\partial(\alpha, \beta, r)} = \frac{1}{\rho_i} \mathbf{K} \frac{\partial(\mathbf{Rt})}{\partial(\alpha, \beta, r)} y_i. \quad (5.6)$$

The final  $(\mathbf{Rt})$  matrix will have the following format:

$$(\mathbf{Rt}) = \left[ \begin{array}{c|c} (\mathbf{R})_{3 \times 3} & (\mathbf{t})_{3 \times 1} \end{array} \right]$$

Therefore, the problem can be addressed by estimating only the  $(\mathbf{R})$  and the  $(\mathbf{t})$ . The  $(\mathbf{R})$  is the dot product of the  $R_{Euler}$ , as defined in equation (3.29), and the rotation  $\mathbf{R}_s$ , as shown below:

$$\mathbf{R}_s = \begin{bmatrix} \cos \beta \cos \alpha & -\sin \beta & \cos \beta \sin \alpha \\ \sin \beta \cos \alpha & \cos \beta & \sin \beta \sin \alpha \\ \sin \alpha & 0 & \cos \alpha \end{bmatrix} \quad (5.7)$$

and the required derivatives are:

$$\frac{\partial \mathbf{R}_s}{\partial \alpha} = \begin{bmatrix} -\cos \beta \sin \alpha & 0 & \cos \beta \cos \alpha \\ -\sin \beta \sin \alpha & 0 & \sin \beta \cos \alpha \\ \cos \alpha & 0 & -\sin \alpha \end{bmatrix}, \quad (5.8)$$

$$\frac{\partial \mathbf{R}_s}{\partial \beta} = \begin{bmatrix} -\sin \beta \cos \alpha & -\cos \beta & -\sin \beta \sin \alpha \\ \cos \beta \cos \alpha & -\sin \beta & \cos \beta \sin \alpha \\ 0 & 0 & 0 \end{bmatrix}, \quad (5.9)$$

$$\frac{\partial \mathbf{R}_s}{\partial r} = 0 \quad (5.10)$$

The translation matrix  $\mathbf{t}_{\text{spherical}}$  is given by the following equation:

$$\mathbf{t} = \mathbf{R}_s \left( - \begin{bmatrix} r \sin \alpha \cos \beta \\ r \sin \beta \sin \alpha \\ r \cos \alpha \end{bmatrix} \right) \quad (5.11)$$

Thus, the necessary derivatives of (5.11), can be found using the equations (5.8), (5.9), (5.10) as follows:

$$\frac{\partial \mathbf{t}}{\partial \alpha} = \frac{\partial \mathbf{R}_s}{\partial \alpha} \left( - \begin{bmatrix} r \sin \alpha \cos \beta \\ r \sin \beta \sin \alpha \\ r \cos \alpha \end{bmatrix} \right) + \mathbf{R}_s \left( - \begin{bmatrix} r \cos \alpha \cos \beta \\ r \sin \beta \cos \alpha \\ -r \sin \alpha \end{bmatrix} \right) \quad (5.12)$$

$$\frac{\partial \mathbf{t}}{\partial \beta} = \frac{\partial \mathbf{R}_s}{\partial \beta} \left( - \begin{bmatrix} r \sin \alpha \cos \beta \\ r \sin \beta \sin \alpha \\ r \cos \alpha \end{bmatrix} \right) + \mathbf{R}_s \left( - \begin{bmatrix} -r \sin \alpha \sin \beta \\ r \cos \beta \sin \alpha \\ 0 \end{bmatrix} \right) \quad (5.13)$$

$$\frac{\partial \mathbf{t}}{\partial r} = - \mathbf{R}_s \begin{bmatrix} \sin \alpha \cos \beta \\ \sin \beta \sin \alpha \\ \cos \alpha \end{bmatrix}. \quad (5.14)$$

Calculating the above, we find the final Jacobian matrix  $\mathbf{J}_q$ , and we use it to find the covariance matrix for detecting the marker. Consequently, we compute the Backward propagation as:

$$\Sigma = (\mathbf{J}_q^T \Sigma_q^{-1} \mathbf{J}_q)^{-1} \quad (5.15)$$

which in our case is:

$$\Sigma_{\alpha, \beta, r} = \left( \mathbf{J}_q^T \begin{bmatrix} \Sigma v_i & & \\ & \ddots & \\ & & \Sigma v_n \end{bmatrix}^{-1} \mathbf{J}_q \right)^{-1} \in \mathfrak{R}^{3 \times 3} \quad (5.16)$$



The diagonal matrix was almost similar in every test we did:

$$\begin{bmatrix} \Sigma v_1 & & & & & & \\ & \Sigma v_2 & & & & & \\ & & \Sigma v_3 & & & & \\ & & & \Sigma v_4 & & & \\ & & & & \Sigma v_5 & & \\ & & & & & \Sigma v_6 & \\ & & & & & & \Sigma v_6 \end{bmatrix}^{-1} \approx \begin{bmatrix} 0.062 & & & & & & \\ & 0.063 & & & & & \\ & & 0.061 & & & & \\ & & & 0.062 & & & \\ & & & & 0.062 & & \\ & & & & & 0.061 & \\ & & & & & & 0.063 \\ & & & & & & & 0.062 \\ & & & & & & & & 0.064 \\ & & & & & & & & & 0.064 \\ & & & & & & & & & & 0.064 \\ & & & & & & & & & & & 0.061 \end{bmatrix}$$

Hence, as the diagonal terms of  $\Sigma v_i$  have almost the same values in every test we run, the final covariance matrix  $\Sigma_{\alpha, \beta, r}$ , mostly depends on the Jacobian matrices, that are related to the position of the camera or of the target.

### 5.3 Relation between Accuracy and the Distance

We also examine the dependence of distance on accuracy. Thus, we use the covariance matrix  $\Sigma_{\alpha, \beta, r}$ . Keeping the values of  $\alpha$ ,  $\beta$  stable, we only change the distance  $r$  with a step of 0.2 m each time. Distance  $r$  is limited between 0 and 3, as it represents the distance in m, where a marker is possible to be "seen" by the camera. Therefore, a marker which is placed more than 3 m away of the camera, can not be captured. Then, we find the different covariance matrices  $\Sigma_{\alpha, \beta, r}$  for each of those  $r$  distances.

Afterwards, we calculate the condition number of these covariance matrices and we plot their relationship to the distance as shown in Figure 5.2. Based on the results presented in the Figure, we concluded that as we increase the distance  $r$  between the camera and the marker, the condition number values are increased as well.

Thus, the covariance matrix is becoming "ill-conditioned" and the orientation errors in the measurements increased too.

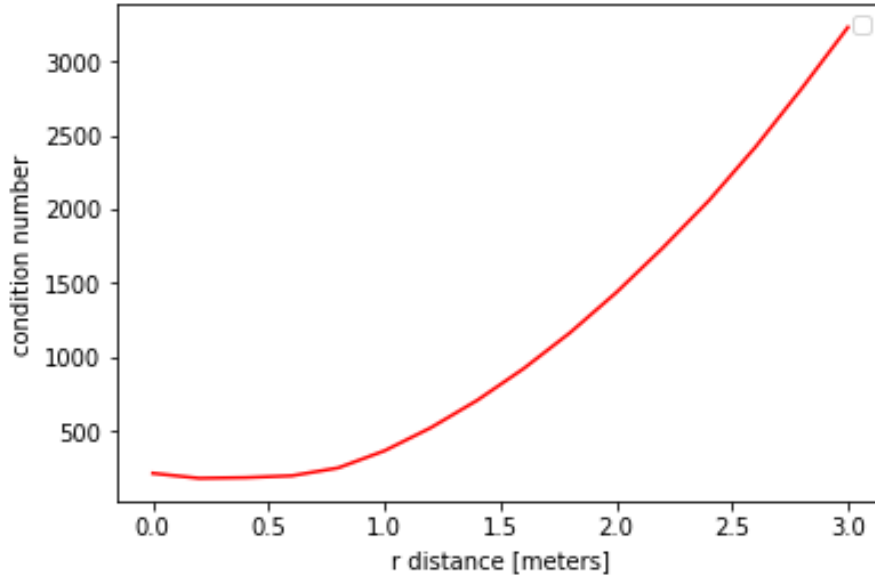


Figure 5.2: Relation between Condition number and Distance  $r$  ( $X \sim \mathcal{N}(0, 4)$ ).

## 5.4 Relation between Accuracy and the Angle View

Thereafter, we use the covariance matrix  $\Sigma_{\alpha, \beta, r}$  to find the relation between accuracy of the pose of the camera in marker coordinates and the  $\alpha$  angle view. The  $\alpha$  angle is limited between  $-\frac{\pi}{2}$  and  $\frac{\pi}{2}$ , as is the inclination in every system expressed in spherical coordinates. The step is set 5 degrees and two different plots were produced.

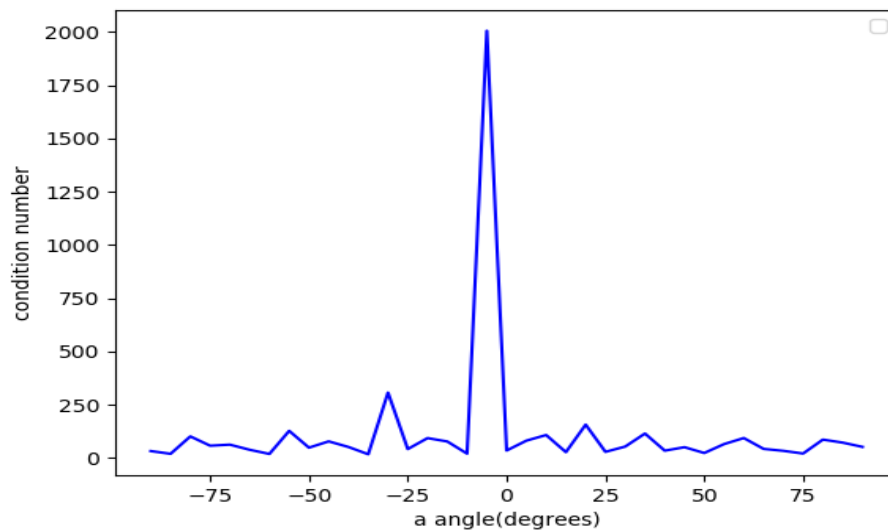
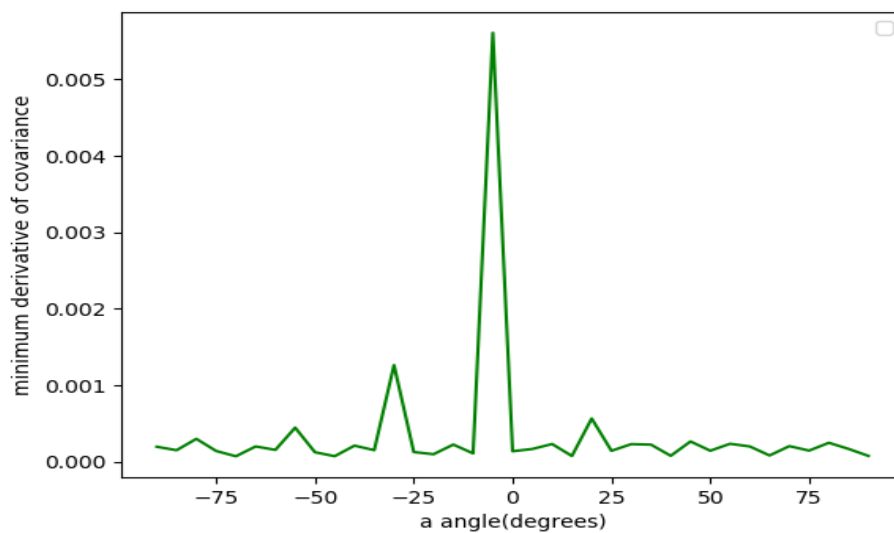
The first one, which is presented in Figure 5.3a shows the relation between the condition number and the angle  $\alpha$ , while the  $\beta$  and  $r$  are keeping the same values as before, that are extracted from the world position of the camera. The second one is the Figure 5.3b, that presents how the derivative of  $\Sigma_{\alpha, \beta, r}$  is affected, due to changes in  $\alpha$  angle view.

As shown, the  $\alpha$  angle view of the camera that causes the minimum condition number and orientation errors is the 90 degrees angle. The same results can be extracted from the Figure 5.3b, as local minimum are spotted in 90 and -90 degrees angle view, while the global maximum is observed around the 0 degrees angle view.

## 5.5 Draw Ellipsoids to represent errors of $\Sigma_{\alpha, \beta, r}$

The covariance error ellipsoid for an angle view  $\alpha$  equals to zero is presented in Figure 5.4a. In addition the covariance error ellipsoid for an angle view  $\alpha$  equals to  $-\frac{\pi}{2}$  or  $\frac{\pi}{2}$  is shown in Figure 5.4b. To plot those ellipsoids we consider confidence level as 95 % .

Additionally, the covariance error ellipsoids for a variety of  $\alpha$  angle views are presented in Figure 5.5.

(a) Relation between Condition Number and the  $\alpha$  Angle View(b) Relation between minimal derivative  $\Sigma_{\alpha, \beta, r}$  of and the  $\alpha$  Angle ViewFigure 5.3: Accuracy Depending on the  $\alpha$  Viewing Angle ( $X \sim \mathcal{N}(0, 4)$ ).

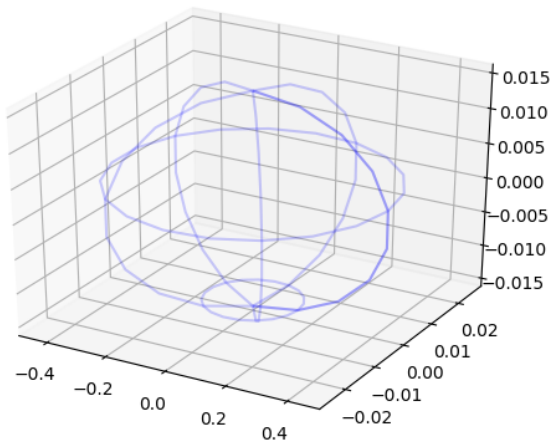
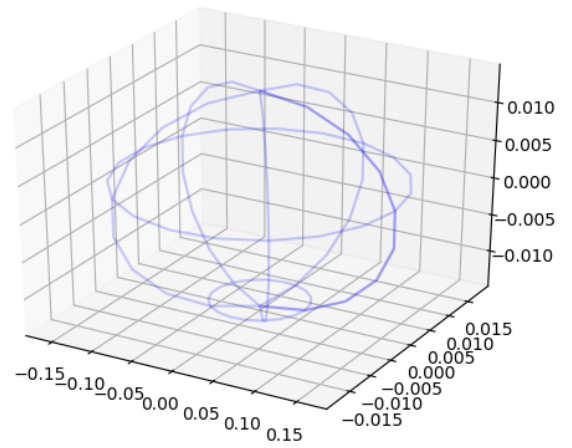
(a) Angle view  $\alpha = 0$ .(b) Angle view  $|\alpha| = \frac{\pi}{2}$ .

Figure 5.4: Error Covariance Ellipsoid.

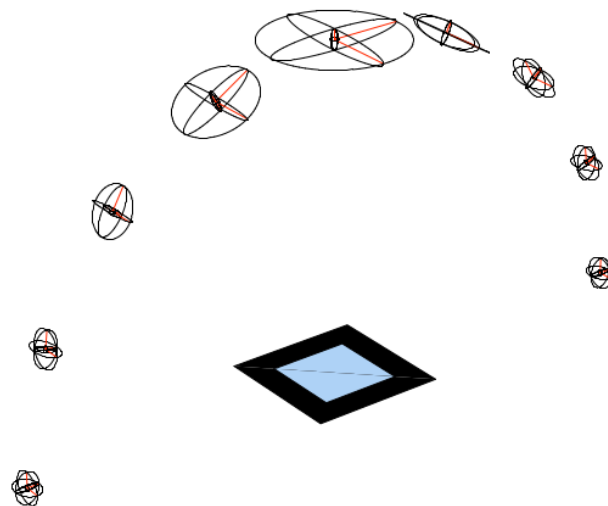


Figure 5.5: Accuracy of the pose of the camera in marker coordinates [6].

## 5.6 Concluding Remarks

Analyzing the above Figures, which are showing the dependence of the distance  $r$  on accuracy and the relation between  $\alpha$  angle view and accuracy, we cross-confirm the results of [6]. Specifically, we find that the worst position for the camera is to be placed on the top of the marker, with an  $\alpha$  angle of zero degrees. The accuracy is also decreased as we move the camera further from the marker.

Moreover, calculating and plotting the error covariance ellipsoids, shows that the pose error tends to be larger when the camera is pointing frontally to the marker. Thus, to be minimized, larger tilt angles are required.



## Chapter 6

# Most accurate 3D Points Configurations for Pose Estimation

Our goal is the estimation of configurations of points in the 3D space, that minimize the Condition Number and errors (RE, TE, Reprojection). The equations for all these errors were mentioned in section 3.8. In such a way, three different methods are evaluated:

- i Randomly selected set of points,
- ii Gradient Descent,
- iii Adam Optimizer.

In this chapter these methods are analyzed and the results and conclusions are explained implementing each of them separately.

Afterwards, we use the Gradient Descent Algorithm, to see the relation between these configurations of points in the 3D space and world position of the camera. Specifically, we evaluate whether changing the position of the camera will affect these configuration of points (*e.g.*, orientation).

### 6.1 Preliminaries

Points Configurations can influence the accuracy and robustness of pose estimation. They also affect the local minimum in the PnP problem.

Thus, in [1], questions are raised, regarding the existence of certain point configurations, that may be able to increase the accuracy by minimizing reprojection and orientation errors; and whether there is a specific configuration of points, which is optimal for every camera pose or not.

In [1], 2D Points Configurations were examined, using the standard linear algorithm for homography estimation, the DLT, as explained in Chapters 2 and 3. To extract the

final matrix, normalization was required, although it is mentioned that, the noise affection is independent for each image pixel.

For the estimation of optimal control points configurations, the minimizing of the condition number of matrix  $\mathbf{A}$  of the DLT, in terms of all  $n$  control points, was selected as optimization criterion.

World-points were selected from a circular plane with a radius of 0.15 m and then their projected image points were measured. For the optimization the gradient descent algorithm was selected, with the condition number being the optimization objective. Moreover, in every image pixel 1000 random Gaussian noise samples were added and the average value of each one of them was calculated.

The authors of [1] concluded, that some 2D points configurations increase the accuracy of homography and pose estimation methods. Specifically optimized points configurations are driven to distribute in space and the points tend to increase the distance from each other, while the most common optimal shape for the 4-points 2D configurations was a square. They also mentioned, that the optimized points configurations do not really depend on the pose of the camera.

In our research, the fiducial markers and thus, the configurations we examine, are belonging to the 3D space. Hence, as mentioned, the DLT in the 3D case needs at least 6 world-points to run correctly. Therefore, we decided to expand the above tests in the 3D case and examine the existence of optimal shape(s) of 6- 3D points. We also compare our results with the ones extracted from the 2D case.

Finally, as the condition number was proven to be a good optimization objective, we decide to use it in our optimization algorithms too.

## 6.2 Random Selection of Points

A Python algorithm is developed as explained in algorithm 4, which gets a  $N$  number as an input, that shows the number of tests, which will be executed. In each test  $M$  random world 3D points are selected, from a sphere with a radius  $r$ .

In our simulations, we set  $M$  number of world points as 6, in order to run the DLT for the 3D case. In addition, we use the same sphere plane as in previous tests, with a radius of 0.3 m.

Then, the projected image points of those 6 real world points are estimated and random Gaussian noise, with a standard deviation of 8 and a mean value of 2, is added. To improve the accuracy of our results, in each image point, 2000 random Gaussian noise samples are added and we keep the average measurement of each pixel.

After normalizing the image and world points, the DLT algorithm is applied on them, and the Reprojection error, as well as the Condition Number, the RE and TE are estimated. Hence, for every test we run, we compare the condition numbers and we keep the points configuration, that lead to the minimum one.

At first we set  $N$  as 500 tests of randomly selected set of 6-points and we examine



**Algorithm 4** Random Points Selection**Input** :  $\mathbf{X} = [X_i, Y_i, Z_i, 1]^T$ ,  $\mathbf{x} = [u_i, v_i, 1]^T$ ,  $N, M, r$ **Output**: Final M-Sphere Points Configuration

1. Set Best M-Points 3D Configuration := None
  2. Set minimum condition number  $cond\_min := \inf$
  3. **for**  $N$  iterations **do**
    4. Get a M random Set of  $\mathbf{X}$
    5. Find their Projected  $\mathbf{x}$
    6. Add Gaussian noise to  $\mathbf{x}$
    7. Run DLT as explained in algorithm 3, and find  $H_{DLT}$
    8. Find the Condition Number of  $H_{DLT}$
    9. **if**  $cond(H_{DLT}) < cond\_min$  **then**
      10.  $cond\_min$  gets  $cond(H_{DLT})$
      11. Update M-Points Configuration := This random Set of  $\mathbf{X}$
- end**

whether there is a specific pattern in the 6-points configurations. As we observe, every time we run these tests, the points that have the minimum errors and Condition Number tend to be closer to the periphery of the sphere.

In Figure 6.1, there is the set of 6-points, which compared to other 499 6-points configurations, had the minimum condition number.

In the second round of tests the  $N$  is increased from 500 to 10.000 tests, in order to validate whether the best set of points tend to have a constant geometric location, a specific optimal shape. Once again, the points with the minimum errors tend to be closer to the perimeter of the sphere.

Additionally, it is also noticed, that in almost every test, the best configuration of points looks similar to a triangular prism (truncated<sup>1</sup>, twisted<sup>2</sup>, normal<sup>3</sup>).

In parallel to the above, Figure 6.2a shows the points configuration, which lead to the minimum Condition Number after the 10.000 random tests. In Figure 6.2b, it is shown also the final set of points and the sphere plane, where all the world points are belonging.

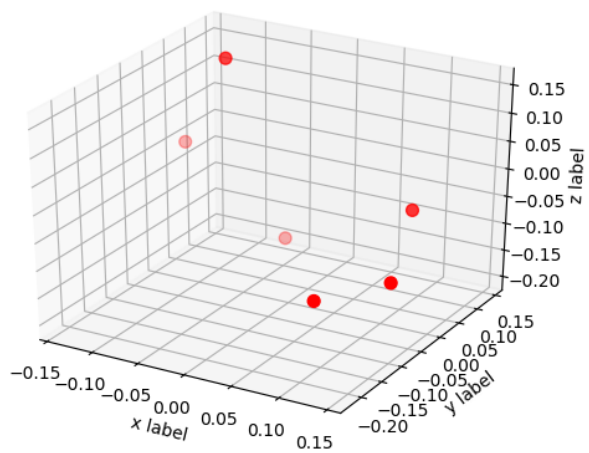
However, as the  $N$  number of tests is increased, the time needed to execute the simulation tests in Python is also highly increased. Thus, the maximum number of tests we run was 11.000, which lead us to similar results with the presented one, but it requires disproportionately more time to be executed.

Even though, the results are pretty close to the 2D case of a circular plane, where the set of points causes the minimum condition number tend to be close to the perimeter of the circle [1], and the points increase their distance from each other, all of our results are

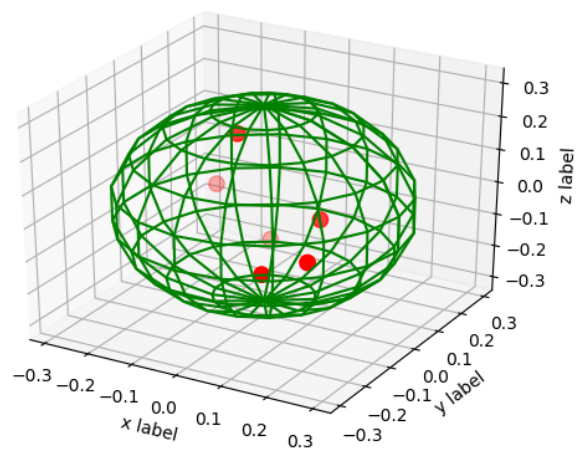
<sup>1</sup>A prism with non parallel top and bottom faces [40].

<sup>2</sup>A prism with one of its triangular base twisted compared to the other by a twist angle [40].

<sup>3</sup>A polyhedron made of a triangular base, a translated copy, and 3 faces joining corresponding sides [40].

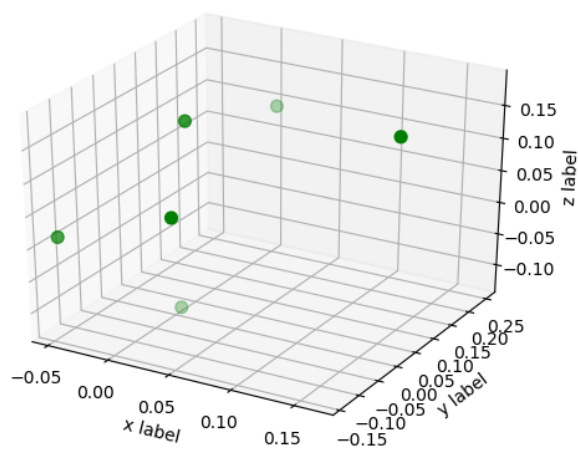


(a) Six world-points.

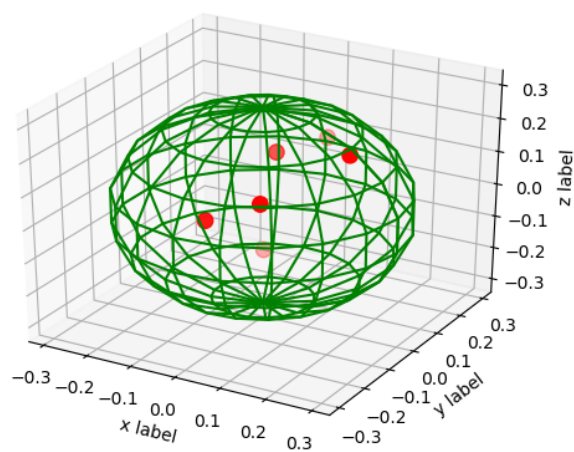


(b) Six world-points and the sphere plane.

Figure 6.1: Six-Points Configuration with the minimum condition number after  $N=500$  tests.



(a) Six world-points configuration.



(b) Six world-points configuration inside the sphere plane.

Figure 6.2: Six-Points Configuration with minimum condition number after  $N = 10.000$  tests.

so far observations from random points selection tests.

Hence, to find out whether the points near periphery are better (in terms of causing minimum errors) to be applied on the DLT, as well as, to examine if the optimal shape for 6 points, which lead the minimum errors (TE, RE, Reprojection) and Condition Number is similar to a triangular prism, an alternative way than randomly selection of points is required. Therefore, other two methods are decided to be tested.

## 6.3 Gradient Descent

First, the Gradient Descent Algorithm that was mentioned in Chapter 2, is used to solve a 3D case problem of points configurations. As we work in the 3D case, a learning rate of 3 parameters is estimated each time.

We decide to use the Gradient Descent optimization algorithm, as the second computational method to estimate 3D points configurations, which are causing the minimum Condition Number end errors. The condition number is selected as the optimization objective and we develop an algorithm in Python, explained in the algorithm 5.

To evaluate the n step(learning rate) in each iteration, the SuperSAB algorithm is used, as it is explained in section 3.11.1.

---

### Algorithm 5 Gradient Descent

---

**Input** :  $\mathbf{X} = [X_i, Y_i, Z_i, 1]^T$ ,  $\mathbf{x} = [u_i, v_i, 1]^T$ , r

**Output**: Final M-Sphere Points Configuration

1. Initialize parameters  $d_{X_i}$ ,  $d_{Y_i}$ ,  $d_{Z_i}$ ,  $d_{X_i, \text{evaluation}}$ ,  $d_{Y_i, \text{evaluation}}$ ,  $d_{Z_i, \text{evaluation}} := 0$
  2. Initialize starting gradient descent step n and  $n_{X_i}$ ,  $n_{Y_i}$ ,  $n_{Z_i}$  as none
  3.  $n_{X_i}$ ,  $n_{Y_i}$ ,  $n_{Z_i} \leftarrow$  n starting step
  4. Define  $n_{\text{positive}}$ ,  $n_{\text{negative}}$
  5. **for** 100 iterations **do**
    6. Add Random Gaussian Noise to  $\mathbf{x}$
    7. Execute DLT algorithm 3, find  $H_{\text{DLT}}$
    8. Find the Condition Number of  $H_{\text{DLT}}$
    9. Run SuperSAB algorithm to find  $n_{X_i}$ ,  $n_{Y_i}$ ,  $n_{Z_i}$
    10. Find  $d_{X_i, \text{evaluation}}$ ,  $d_{Y_i, \text{evaluation}}$ ,  $d_{Z_i, \text{evaluation}}$  by multiplying their current values with  $n_{X_i}$ ,  $n_{Y_i}$ ,  $n_{Z_i}$
    11. Update  $\mathbf{X}$  adding the opposite value of the gradient  $d_{X_i, \text{evaluation}}$ ,  $d_{Y_i, \text{evaluation}}$ ,  $d_{Z_i, \text{evaluation}}$  to each one of the  $\mathbf{X}$
    12. **if** New point doesn't belong to the Sphere **then**
      13.  $\mathbf{x} \leftarrow$  Sphere Radius / Point's RMS Deviation
  - end**
  14. Find new  $\mathbf{x}$  as the projection of the updated  $\mathbf{X}$
- end**
- 

We execute the Gradient Descent optimization algorithm for  $N = 100$  iterations. As

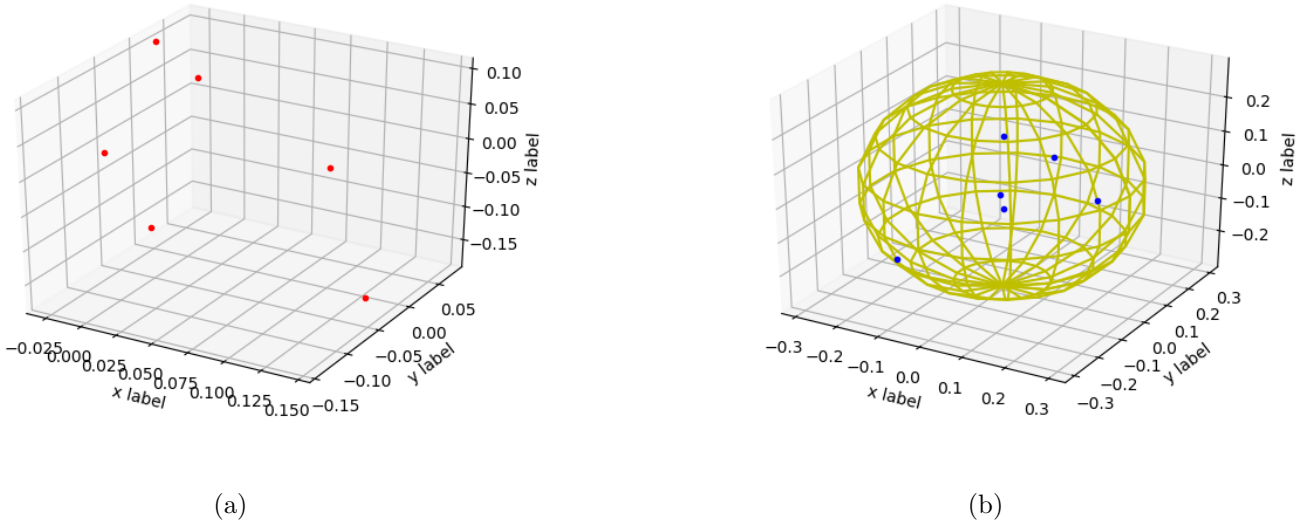


Figure 6.3: Points Configuration at the first iteration of Gradient Descent.

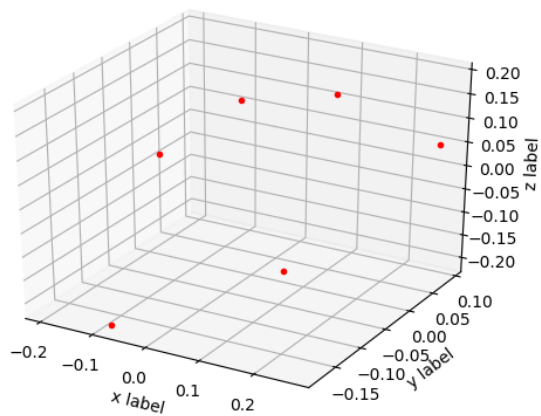
first initial set of points, we use a randomly selected set of six points belonging to the same sphere plane as before, with a radius of 0.3 m. Their projected image points are calculated again and random Gaussian noise is added to each one of the image pixels. Each noise sample had a mean value of 2 and a standard deviation of 8. To improve the quality of our estimations we add 1000 random Gaussian noise samples to each pixel and we calculate the average value of each one of them.

The results of those tests are presented below. Specifically, in Figure 6.3 there is the set of six points in the first iteration, where the selection of points is totally random. Then, Figure 6.4 shows the positions of the 6-points in the 50<sup>th</sup> iteration, and in Figure 6.4 there is the final position of the set of 6-points after the 100<sup>th</sup> iteration. This final set of points has the minimum condition number, but also the minimum RE, TE and reprojection error.

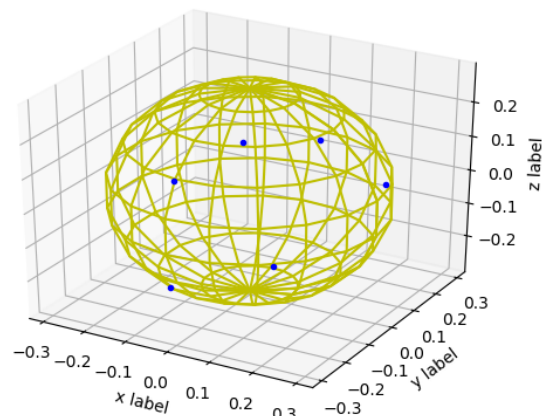
After a few tests, we conclude once again, that the set of points that causes minimum errors and Condition Number, is the one where the points are belonging to the sphere periphery, or at least are very close to it. Moreover, after a while ( $N \approx 80$  iterations), the points tend to create a triangular prism, which was happening also through the random selection of points tests.

We also calculate the RE, TE, Reprojection error and Condition Number of these sets of points in every iteration and they are presented in the Figure 6.6. As shown, the final set of points at the 100<sup>th</sup> iteration, -which from Figure 6.5, we can see that they are belonging to the sphere periphery and have a triangular prism as an optimal shape,- cause the minimum errors.

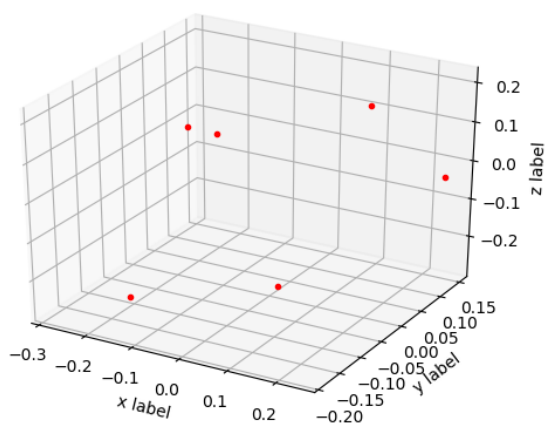
Furthermore, the relation between the Condition Number and the TE and RE, is also shown in Figure 6.6, as small changes in the Condition Number are causing also changes



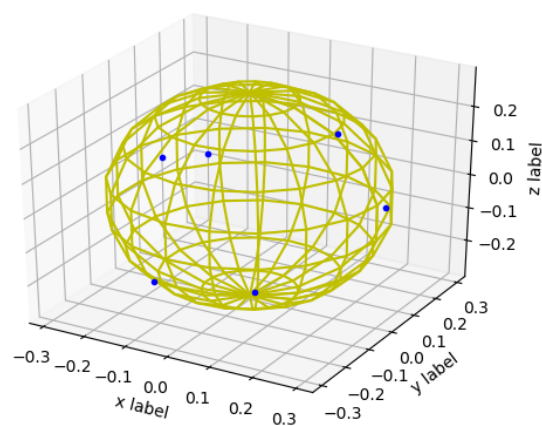
(a)



(b)

Figure 6.4: Points Configuration at the 50<sup>th</sup> iteration of Gradient Descent.

(a)



(b)

Figure 6.5: Final Points Configuration using Gradient Descent.

in those errors. Each local maximum in the RE corresponds to a local maximum in TE and the same is happening with each local minimum. Hence, these sets of points, which correspond to local minimum in TE can be used in DLT and they will also result in the estimation of a  $\hat{R}$ , that has a value closer to the real one.

Additionally, we see that the final configuration of points (at the last iteration  $N = 100$ ), corresponds to the global minimum in every one of the plots, which means that this is the configuration that causes the minimum errors.

Furthermore, Figure 6.7 represents both the first set of points ( $N = 1$ ), which is randomly selected from the sphere class, and the final one ( $N = 100$ ). Thus, as it is shown, the set of points after the final iteration is sparsely arranged and closer to the limits of the sphere comparing with the points, that were randomly selected in the first iteration. The Gradient Descent optimization algorithm cross-confirms our observations through the random selection of sphere points. However, we decide to use one more algorithm to see whether it will also validate our results.

## 6.4 Adam Optimizer

After the above tests, we try one last method, the Adam Optimizer, as explained in Chapter 2. An Adam optimizer tool is used, which is already provided in Python [30], to examine whether this optimization method will lead to the same results as the others or not. We use the same sphere plane with a radius of 0.3 m to select our world points. This time we run the algorithm for  $N = 200$  iterations, instead of  $N = 100$  that we chose for the Gradient Descent algorithm. Again 1000 random Gaussian noise samples were added to image pixels with a mean value of 2 and a standard deviation of 8 and we estimate the average value of each one.

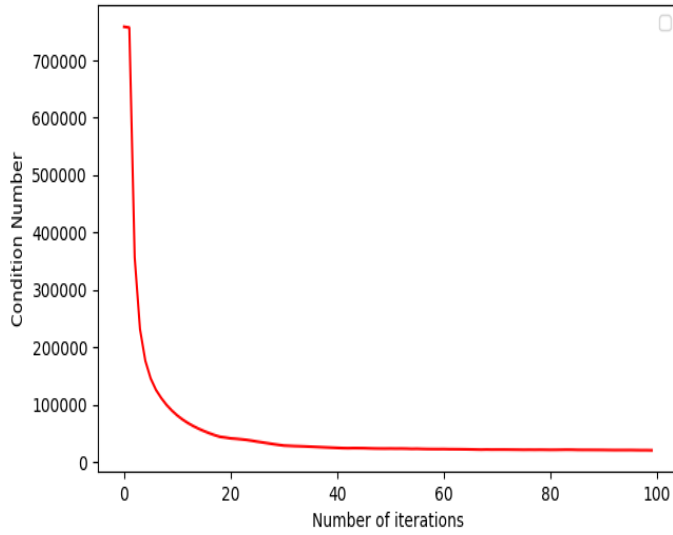
Each time we apply the DLT on the sets of 6-points extracted from Adam, and we find the  $\hat{R}$ ,  $\hat{t}$  for each set. Then the errors are calculated. For the optimization we are comparing the Condition Number of  $H_{DLT}$ , for each set of six points and we try to find the set that minimizes its value.

In Figure 6.8a it is shown the first random selected set of points, which is limited by the defined sphere plane. Moreover, Figure 6.8 illustrates the final 3D Points Configuration.

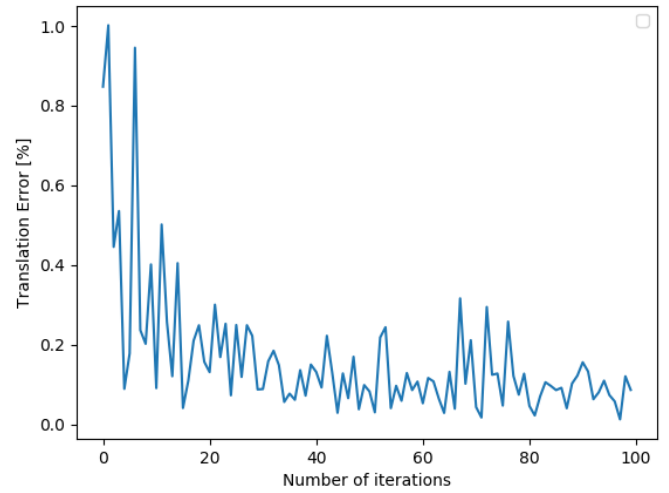
As we see the points after a number of iterations ( $\approx 100$ ), adapt their final position, with possible small tilts around it.

In Figure 6.10 the errors are presented, as well as, the Condition Number for the 200 iterations of the algorithm. As the number  $N$  increased Reprojection Error, Condition Number, RE and TE are getting an almost constant value, as the set of points is finalized in a specific 3D configuration.

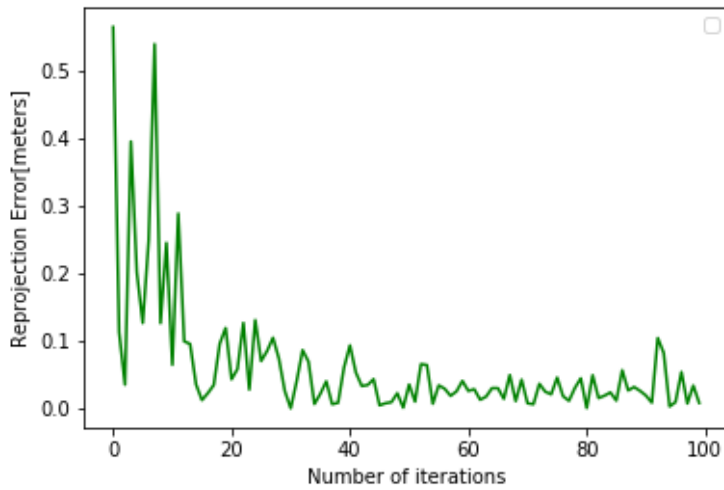
Additionally, we see that as the points move away from each other and close to the periphery of the sphere, the errors are decreased. Moreover, when the points are creating a triangular prism, which are the optimal shape for the 6-points tests, based on our simulations, the optimization algorithm finds the global minimum in every plot. Finally,



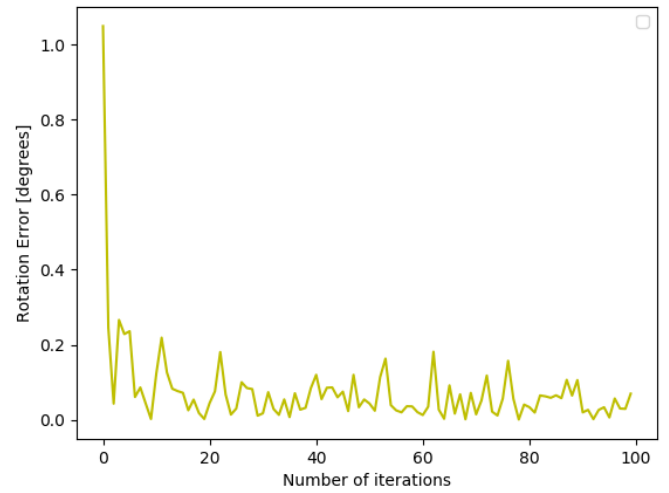
(a) Condition Number.



(b) Error in Translation.



(c) Reprojection Error.



(d) Rotational Error.

Figure 6.6: Errors in Gradient Descent ( $X \sim \mathcal{N}(2, 8)$ ).

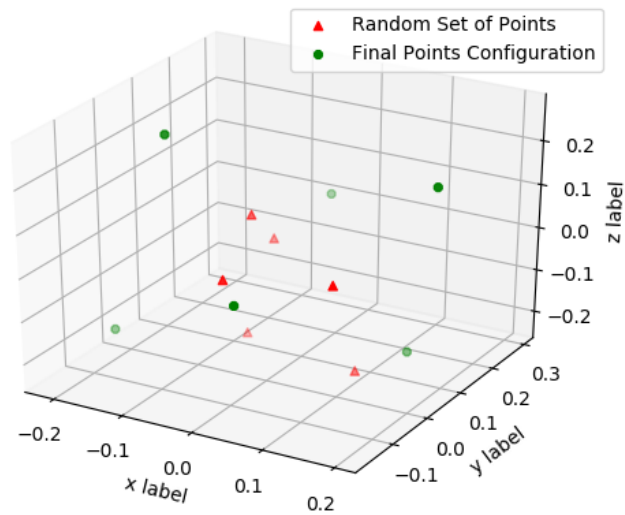
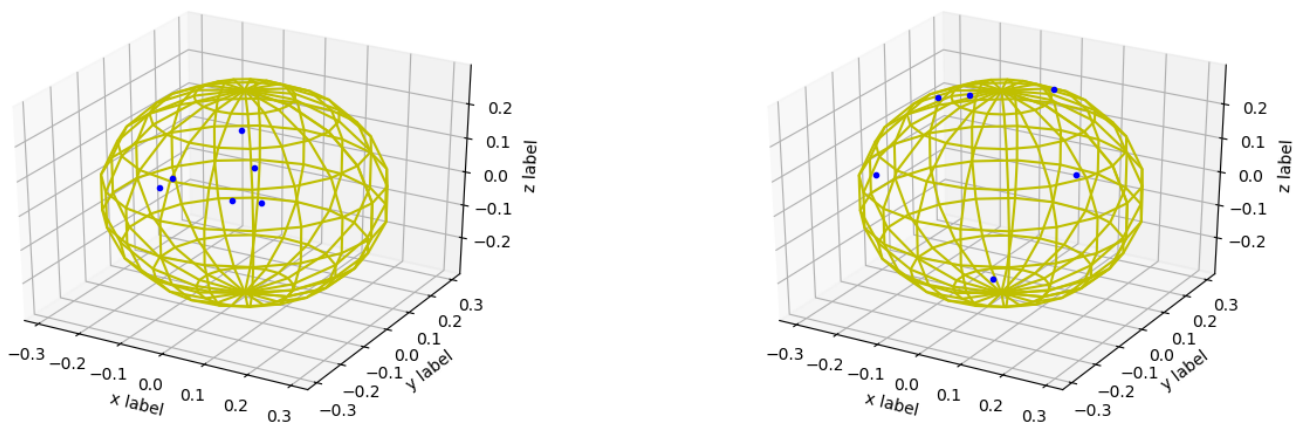


Figure 6.7: Comparing Sets of Points from first and last iteration of Gradient Descent.



(a)  $N = 1$ .

(b)  $N = 200$ .

Figure 6.8: Six-Points Configurations using Adam.



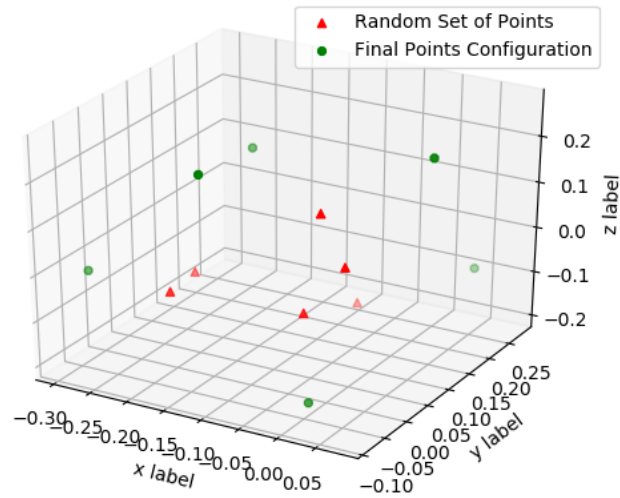
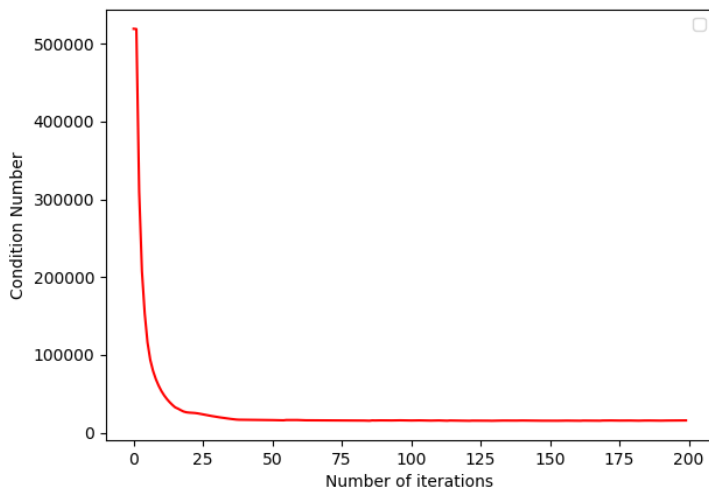
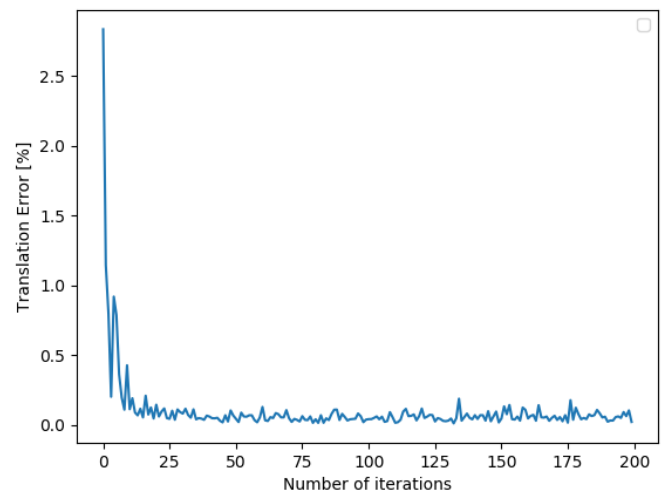


Figure 6.9: Set of points from the first and the last iteration of Adam Optimizer.

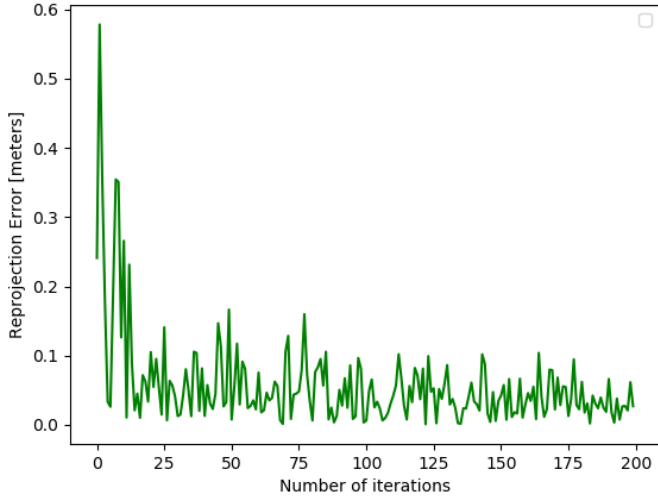
Figure 6.9 is shown the relation between the set of points from the first iteration ( $N = 1$ ) and the final points configuration ( $N = 200$ ). The results are similar to the ones we had with the other methods we tried, so we cross-confirm that the most accurate 3D points configurations are similar to triangular prisms.



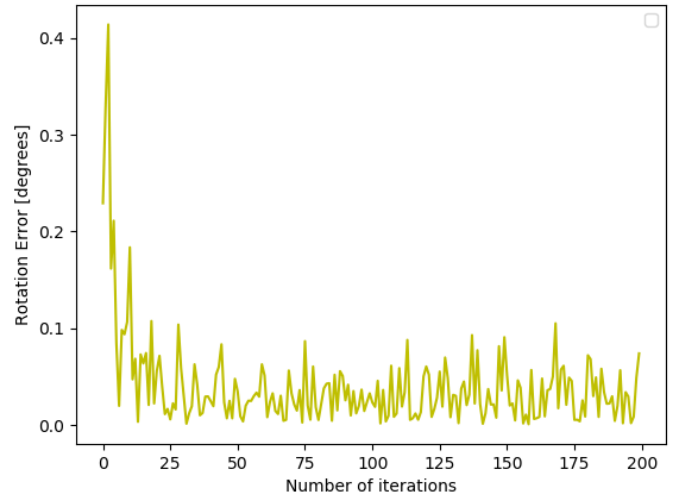
(a) Condition Number in Adam Optimization.



(b) Error in Translation in Adam Optimization.



(c) Reprojection Error in Adam Optimization.



(d) Rotational Error in Adam Optimization.

Figure 6.10: Errors in each iteration of Adam Optimization ( $X \sim \mathcal{N}(2, 8)$ ).

## 6.5 Estimate accurate 3D Points Configurations, using spherical coordinates

In Chapter 4, we concluded that the view angle and the distance between the camera and the marker can affect the accuracy of our measurements. More specific a small distance  $r$  (less than a meter, as then the condition number are non-linear, highly increased) and an  $\alpha$  angle view equals to  $\frac{\pi}{2}$ , was proved that give the most accurate results. Hence, we decide to change the world position of the camera, to calculate a new  $(Rt)$  matrix with the same  $\beta$  angle, but we set  $\alpha$  angle as  $\frac{\pi}{2}$  and  $r$  distance as 0.8 m.

Using the above  $(Rt)$  matrix, we find the projection matrix expressed in spherical coordinates  $(P)$  and we apply this matrix on the equation (3.5), to find the image projected points of the real world 3D points. Once again, the world points were limited inside a sphere with a radius of 0.3 m. The  $\mathbf{K}$  matrix is still the same as defined in Chapter 3. Then, we apply this points on the DLT and we calculate the TE, RE and the Condition Number of the  $H_{DLT}$  matrix, using this time for the error estimations, the spherical matrices  $((R), (t))$ .

To find the 3D points configuration based on the new camera pose, we use the Gradient Descent Optimization Algorithm, as defined before, and we run 100 iterations of it. Once again, 1000 samples of random Gaussian noise, with a mean of 2 and a standard deviation of 8 are added to the image pixels and the average value is calculated each time. All the errors are measured again after applying these new world-points on the DLT algorithm. Moreover, the optimization objective is selected to be once again the condition number.

The results for the first iteration, where the points were randomly selected, are pre-

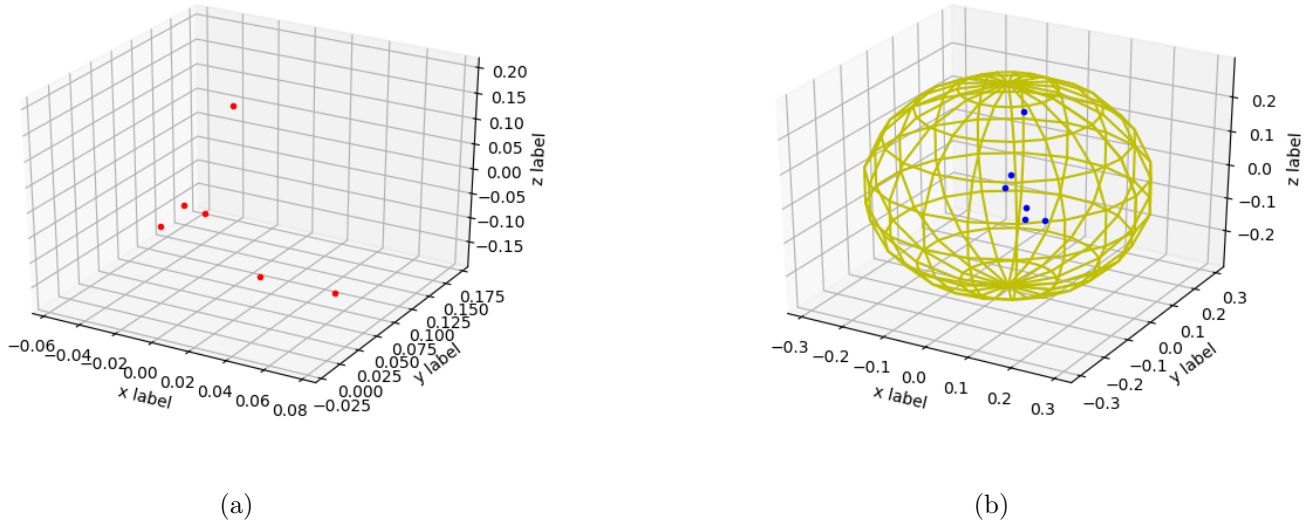


Figure 6.11: Points Configuration at the first iteration of Gradient Descent.

sented in Figure 6.11. Afterwards, Figure 6.12 is shown the 6-points configuration at the 50<sup>th</sup> iteration and Figure 6.13 presents the final Points Configuration(N=100). Thereafter, Figure 6.14 is shown both sets of 6 Points from the first (N = 1) and the last (N = 100) iteration of the optimization algorithm. Again, we can see how the points are moving near to the periphery as the number of iterations is increased.

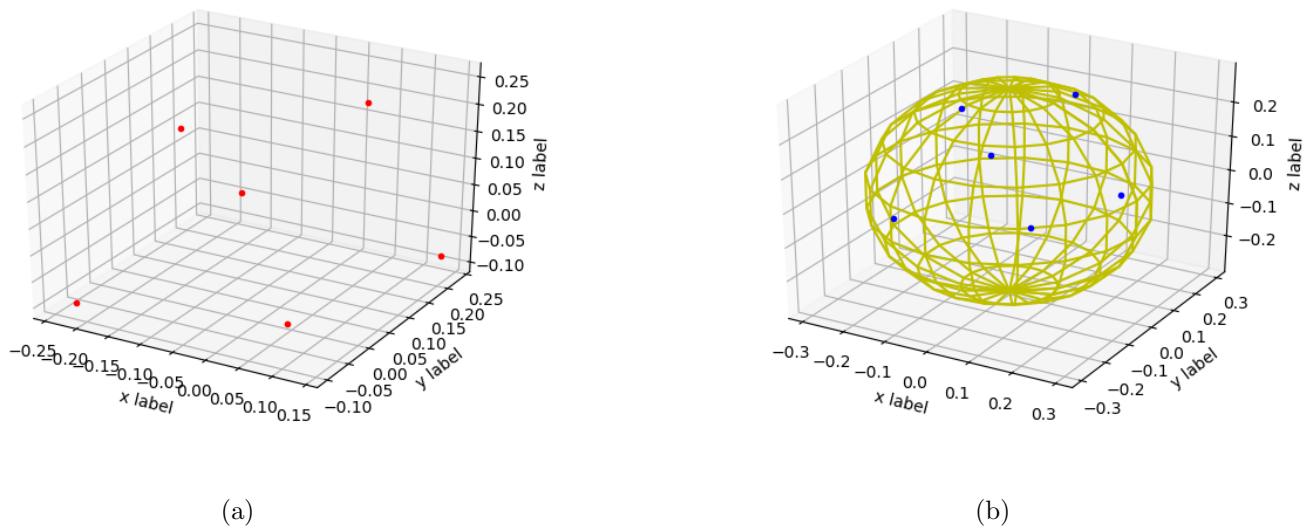
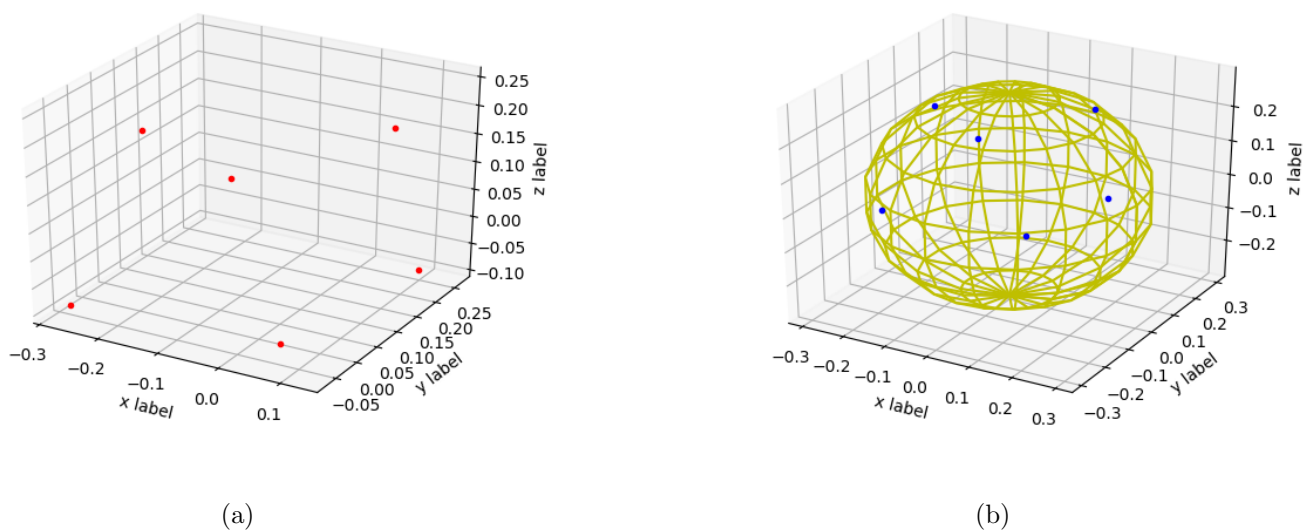
As we can see, changing the pose of the camera, doesn't affect neither the fact that the points tend to move closer to the sphere's periphery in each iteration, nor the optimal shape of the points configuration, which is still a triangular prism. However, the points were finalizing their position faster than before, as triangular prisms can be observed even from the 50<sup>th</sup> iteration. Thus, the orientation of the optimal is considered to be more or less the same (*i.e.*, not depending on the camera pose).

The RE, TE and Condition Number are also calculated in each iteration and the results are presented in Figure 6.15. As shown, each local maximum in the Condition Number, corresponds to a local maximum in both RE and TE, as was happening in all the other test for different positions of the camera.

### 6.5.1 Comparing Errors for the 2 different positions of the Camera

We compare the RE, TE in these two different positions of the Camera. In the first one the camera is placed looking straight down into the plane model<sup>4</sup>, while in the second case the camera is placed with an  $\alpha$  angle-view of  $\frac{\pi}{2}$  with the marker. In both cases, we use the same set of world-points, that is randomly selected from our sphere plane. We also add the same Gaussian noise with a mean of 2 and a standard deviation of 8, and we

<sup>4</sup>This position is also the one we used in all the above simulations, as defined in Chapter 4.

Figure 6.12: Points Configuration of Gradient Descent ( $N = 50$ ).Figure 6.13: Final Points Configuration of Gradient Descent ( $N = 100$ ).

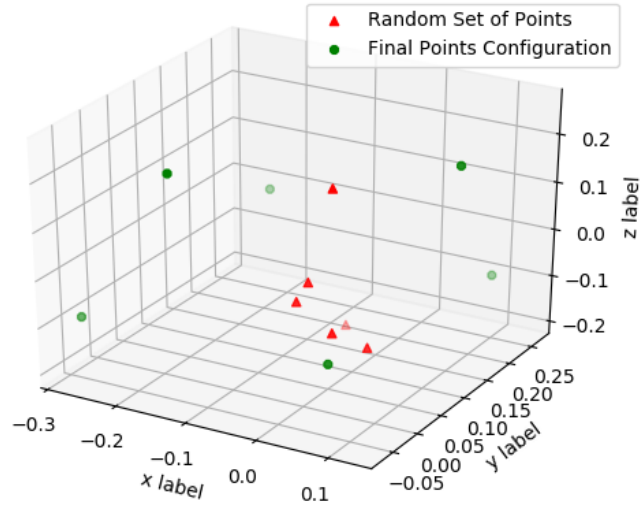
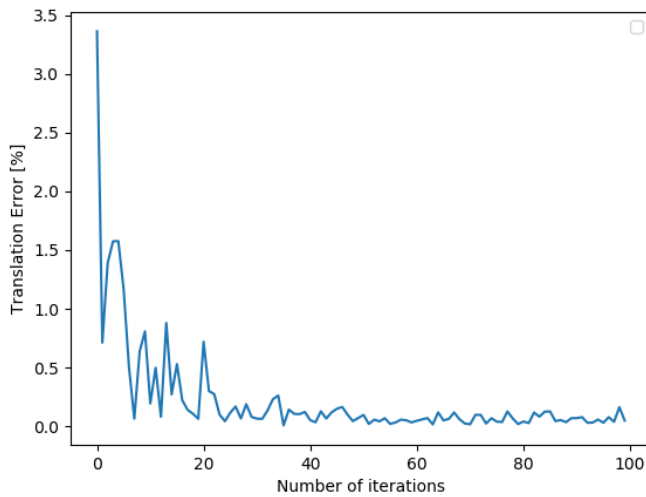
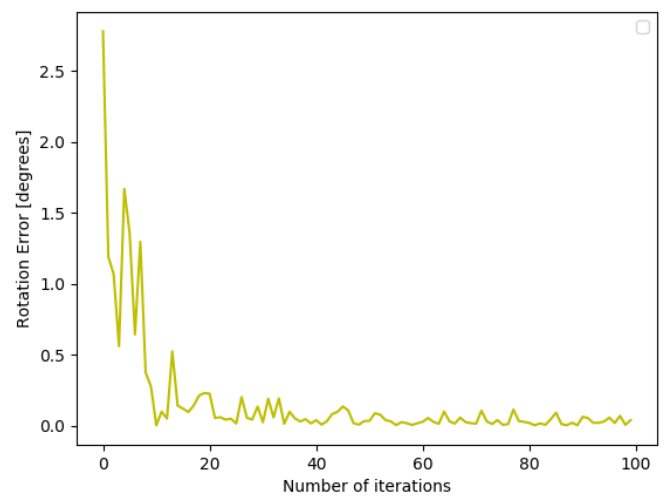


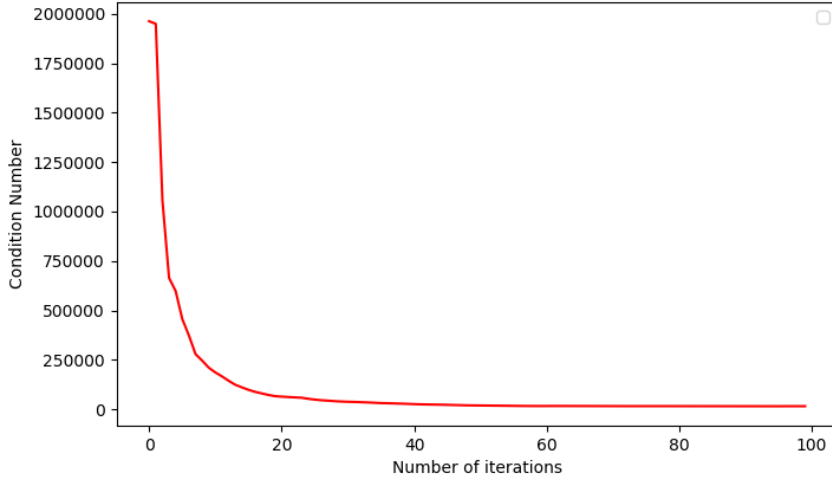
Figure 6.14: Comparing Sets of Points from first and last iteration of Gradient Descent.



(a) Error in Translation.



(b) Rotational Error.



(c) Condition Number.

Figure 6.15: Errors in each iteration of Gradient Descent at the new position of the Camera ( $X \sim \mathcal{N}(2, 8)$ ).

run the Gradient Descent optimization algorithm for 100 iterations. Afterwards, we plot the calculated errors for these two camera positions. The RE is presented in Figure 6.16 for both positions of the camera, while the TE is shown in Figure 6.17.

Comparing the results between these two camera positions we can see the following:

1. The position of the Camera with an angle of  $\frac{\pi}{2}$  towards the markers, leads to minimum error values in both rotation (RE, TE).
2. The maximum value of RE for the same set of 6-points is bigger in the first camera pose, where the camera is placed on the top of the marker. Specifically, in the presented example of the simulation, the maximum RE value for the first camera position is almost 3 degrees, while in the final position is less than 0.5 degree.
3. The maximum value of TE for the same world-points is higher in the first camera pose than in the final one. In our example, the maximum TE for the first selected camera position is almost 5 (%), while in the final position is less than 1.5 (%).

## 6.6 Concluding Remarks

In this Chapter we have introduced two optimization algorithms, in order to validate our observations through random selection of points process. Both algorithms lead to similar results in terms of the 3D points configurations, which are causing the minimum errors and Condition Number between the estimated camera parameters, extracted from DLT and the real ones.

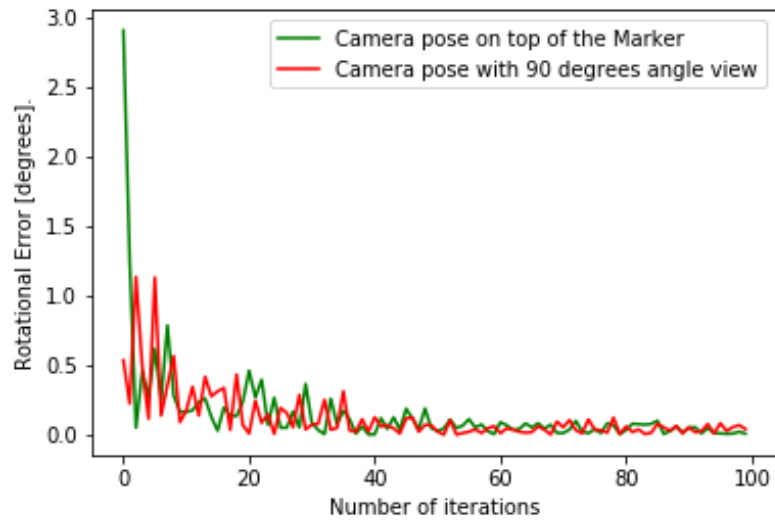


Figure 6.16: Rotational Error in both cases: First position of the camera and the final one.

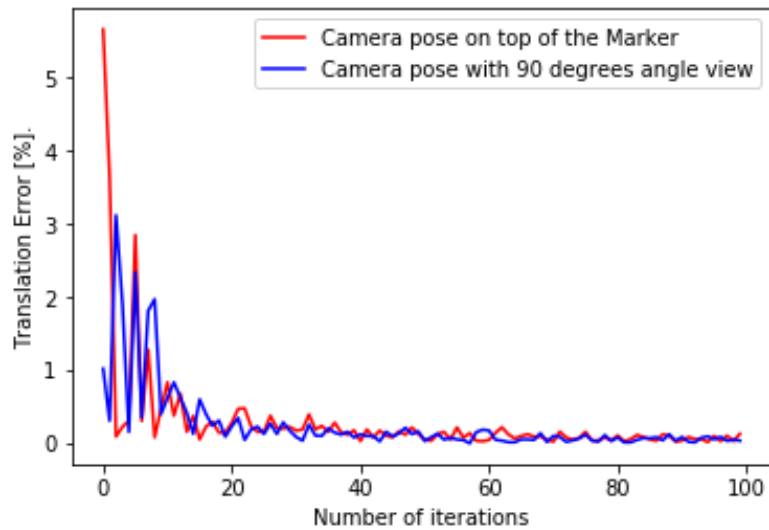


Figure 6.17: Error in Translation in both cases: First position of the camera and the final one.

The points that behave better when they are applied on the DLT algorithm, are close to the periphery of the sphere, which in our case was 0.3 m. Moreover, the optimal shape of these points tend to be a triangular prism. The possible triangular prism shapes that we observe throughout our tests are presented in Figure 6.18.

Furthermore, we also see that changing the position of the camera does not affect neither the optimal shape of the points nor the fact that as the points move closer to the sphere limits, the errors and the Condition Number are decreasing. However, the position of the camera may not affect optimal shape's orientation, but it affects the value of the errors (RE, TE), as a Camera close to the marker and with a 90 degrees angle-view causes smaller errors than placed in any other world position.

Comparing our results with the [1], we can see a few similarities between 2D and 3D optimized object points:

1. The final optimized points configurations are located near to the object plane limits.
2. There is an optimal shape in both cases.
3. The pose of the camera does not affect the optimal shape of the final points configurations.

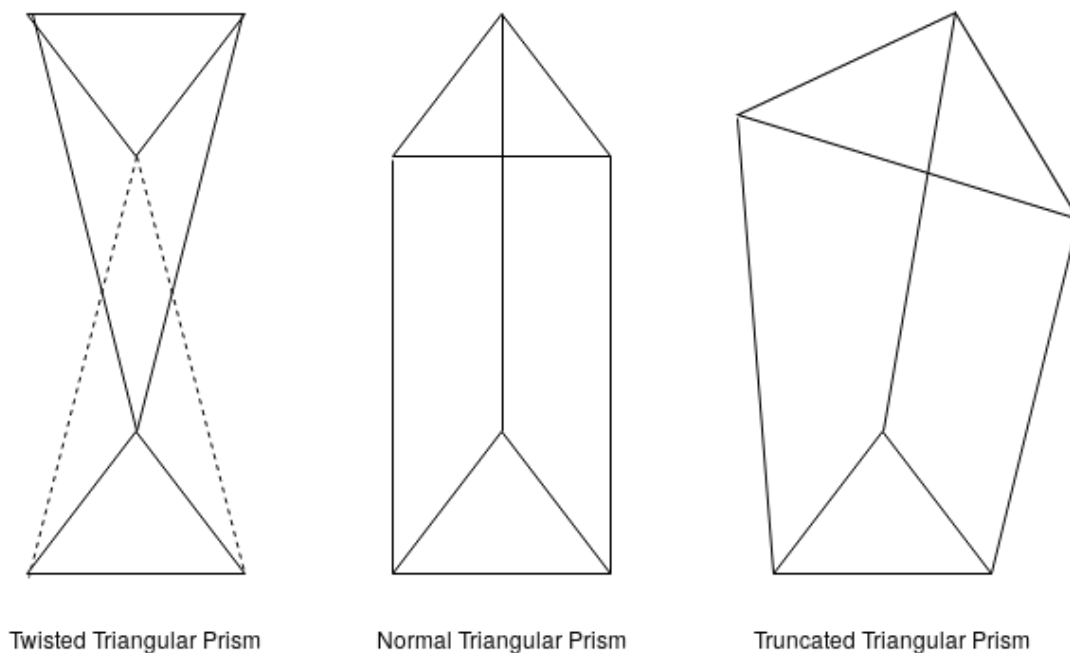


Figure 6.18: Observed Triangular Prisms.



## Chapter 7

# Current Applications of Fiducial Markers in Laparoscopic Surgeries

In this Chapter, a current narrative review on the field of fiducials in Laparoscopy is presented, as the results from Chapter 5 will be applied afterwards on this specific case.

### 7.1 Surgical Augmented Reality

Although, the first experiments using medical images are dated back to 1895, the real-time access to 2D or 3D reconstructed images during an ongoing surgery using AR is a new concept in medicine.

An augmented reality system produces computer-processed imaging data in real-time. To achieve that, a set of devices is required as presented in Figure 7.1. A computer generated image covers a real-world image captured by a camera and presented in a device (*e.g.*, computer).

Usually the objects that are targeted are changing during the surgical procedure. Thus, it is important to monitor the operative field and making real-time changes to the displayed 3D model. Another important aspect is the exact alignment of a real environment and computer-generated images. That can be achieved either by manually aligning both images or by using a set of trackers, to determine the position of the camera and the patient's body. These trackers also provide real-time information regarding the position of certain medical instruments. They are tracking markers, that provide the system with points of reference [38].

### 7.2 Optical Tracking for medical devices localization

Optical tracking is mainly used for tracking medical instruments in surgical AR, *e.g.*, endoscope tracking in Laparoscopic Surgeries. Fiducial markers, that have pre-defined patterns are attached to the medical instrument (*e.g.*, laparoscope, endoscope), which is needed to be tracked, and at least a camera detects them, and their 3D location in the

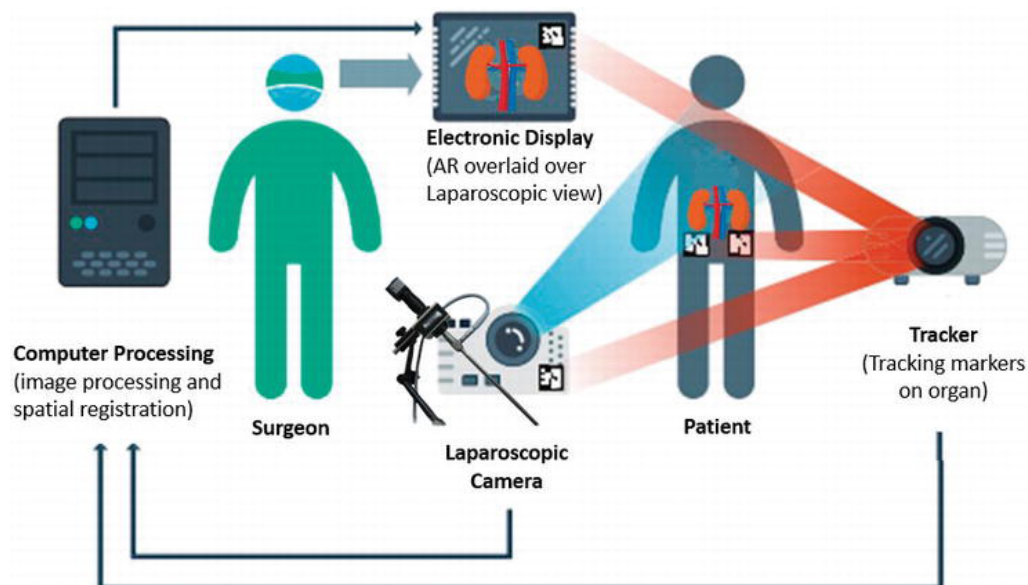


Figure 7.1: The basic principles of AR. [38]

scene is calculated in real-time. The markers are attached to the tracked object in such a place, where the disturbance for the surgeon is minimized and the visibility by the camera is constant.

Optical tracking identifies the pose of a tracked object by measuring light that is either transmitted or reflected by this object. An example of attached fiducial markers on a laparoscope, with a specific pattern, and the two cameras that detect them is shown in Figure 7.2.

### 7.3 Electromagnetic Tracking Systems

Electromagnetic tracking systems represent another class of commonly used tracking devices. This technology consists of tracking some small wired sensors inside a magnetic field -that has a known geometry- generated by a device close to the surgical site. The magnetic field inducing a current in the sensors at various voltages which are measured. These measurements are then used to determine the position of the sensors [13].

Since the sensors are much smaller than the optical ones and given that the magnetic field traverses tissues, the markers can be attached at the exact location that requires tracking, even inside the patient. The principle of intra-operative electromagnetic tracking is presented in Figure 7.3.



Figure 7.2: A fixed markers pattern placed on the laparoscope for optical tracking and the detection cameras [7].

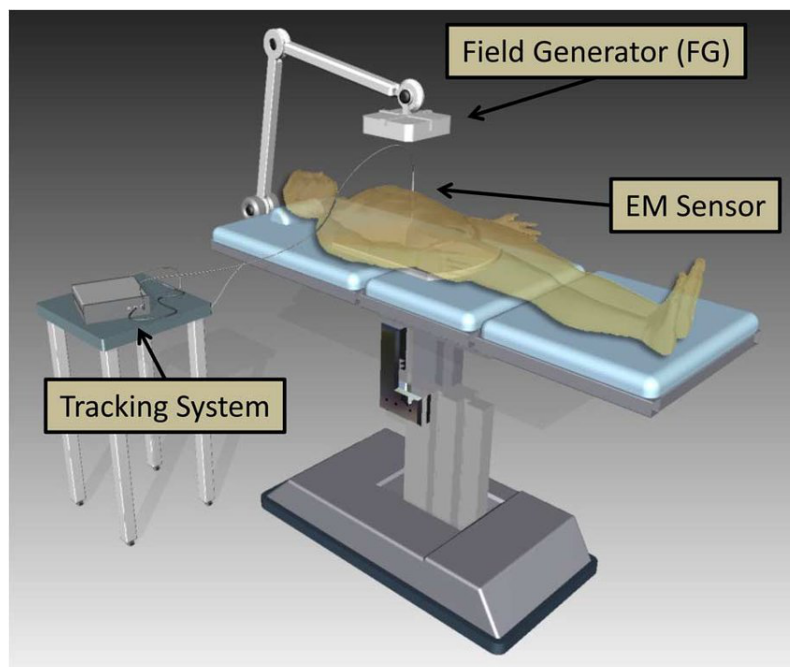


Figure 7.3: Principle of intra-operative electromagnetic (EM) tracking [16].

	Optical	Electromagnetic
Accuracy	superior/good	acceptable
Workflow	acceptable	acceptable
Line of sight	inferior/poor	acceptable
Working volume	superior/good	inferior/poor
Influence of metal	acceptable	inferior/poor
Instrument size	inferior/poor	superior/good
Sampling speed	acceptable	acceptable
Number of targets	superior/good	acceptable

Table 7.1: Comparison Between Optical and Electromagnetic Tracking Systems [19].

## 7.4 Comparing Optical and Electromagnetic Tracking Systems

Optical and electromagnetic systems are widely used commercially for tracking, in computer-assisted surgery. The tracking system is placed near the patient in the procedure room. The tracker is attached to an instrument. Both passive (wireless) and active (wired) trackers exist in the market [19].

Optical trackers are considered to have two main disadvantages. Firstly, as the accuracy is related to the size of each tracker, the trackers are large related to the electromagnetic systems. Optical tracking systems consist of at least three spherical markers, that have to be always visualized by the camera system. Secondly, they are line-of-sight devices, requiring a direct optical path between the position sensor system and the markers, which are being tracked.

Electromagnetic systems, are able to track without requiring a direct line of sight to the instrumentation, but they have smaller tracking volumes, which is considered as insufficient in a few cases (*e.g.*, trauma). Furthermore, the main disadvantage of these systems is the influence of metal, which can distort measurements and influence accuracy, while they also requiring the use of tethering cables connected to the position sensor [19].

In Table 7.1 the comparison Between Optical and Electromagnetic Tracking Systems is analyzed. Thus, the optical tracking system is decided to be examined and used for designing our proposed markers in this thesis.

## 7.5 Fiducial Markers for Optical Tracking in Medical Applications

Fiducial markers have been a basic tool in developing AR applications, as they allow both target identification and camera localization [29]. These fiducials are mainly passive, spherical markers, coated with an Infrared (IR) light retroreflecting material.

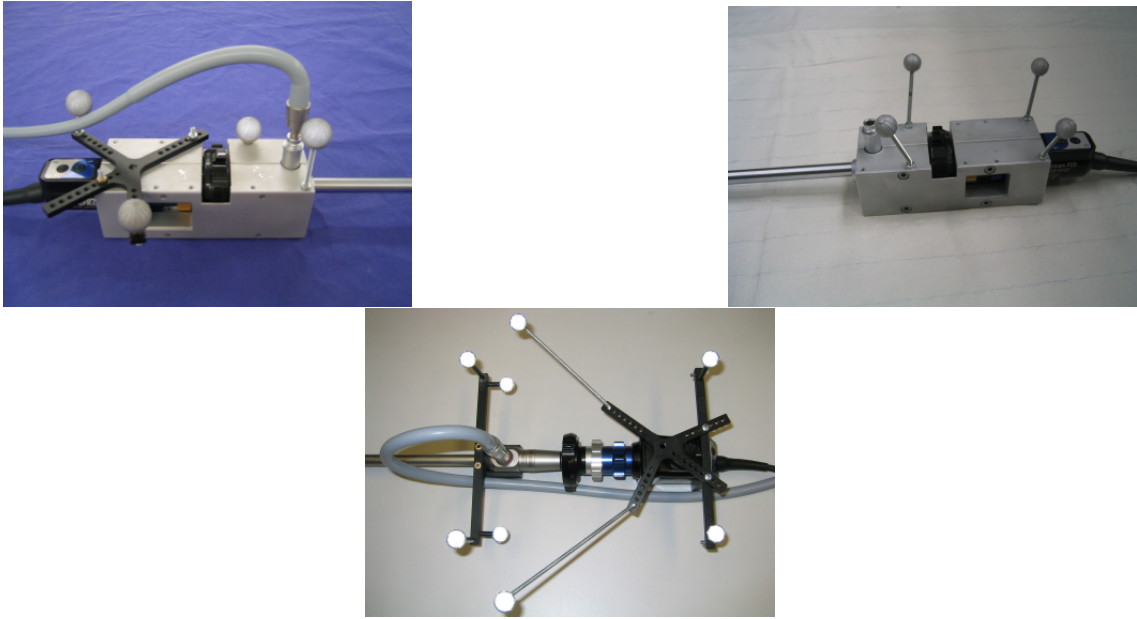


Figure 7.4: Already developed laparoscope marker body configurations [13].

This light is transmitted from an IR light source at the optical tracking system, reflected by the markers, and finally measured by cameras that are positioned at a defined distance from each other. In Figure 7.4 3 different developed laparoscope marker body with 4 fiducials configurations are presented, which according to the surgeons they are not disadvantageous during the laparoscopic surgeries [13].

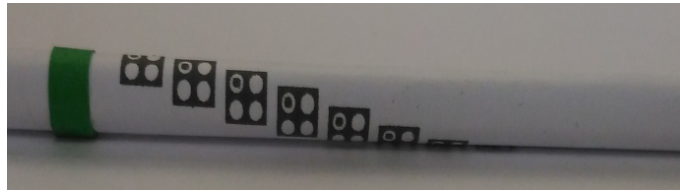
A few examples of proposed markers, which can be used in medical applications for optical tracking, are presented in the following subsections. Usually, 4 spherical 3D markers are used for optical tracking, however some other cases are also analyzed.

### 7.5.1 2D Fiducial Marker for Endoscope Tracking

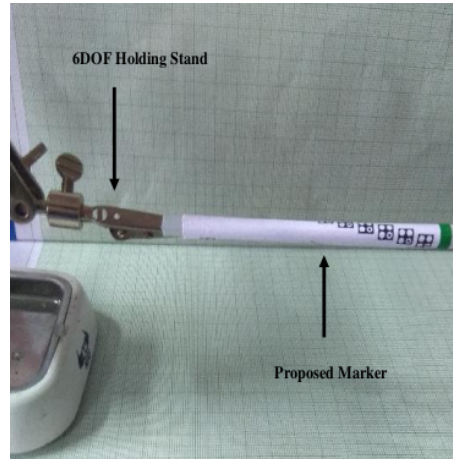
In [17] a marker design is proposed, consists of green band and squares with five circles inscribed in it, structured in a spiral fashion. A single square with four white and one black circle inscribed on it forms a single module, and 12 modules in total are placed. The distance from the green band to a square module helps, so that the exact square module that is detected, will be known.

This marker is presented in Figure 7.5a. This printable and wrap-able marker is made to estimate the tip pose for cylindrical surgical devices. Regarding the detection of the marker, the center of the circles in each square are used to estimate the pose of the tool.

The authors of [17], concluded that these markers are low-cost markers for the existing endoscopic instruments and they provide additional assistance and monitoring instead of just relying on the experiences of the surgeons or on costly additional measurement instruments. Figure 7.5b presents a front view of the proposed marker with camera fixed on a 6 DOF holding stand.



(a) Endoscopic view of marker after wrapping on the tool.



(b) Real Experimental setup: A front view of a proposed marker with camera fixed.

Figure 7.5: The proposed fiducial marker [17].

### 7.5.2 Stereoscopic augmented reality for laparoscopic surgery

A stereoscopic AR system, composed of a stereoscopic vision system and an optical tracker has been designed in [24]. The system uses two imaging devices and an ultrasound scanner. The authors mention that, the optical tracker they used, is rated to have submillimeter tracking error. The tracker is presented in Figure 7.6.

As mentioned, a series of experiments was conducted using the developed stereoscopic AR system. It is highlighted, that low system latency and high image-to-video registration accuracy are critical for a successful clinical AR system. The system prototype has been evaluated in phantoms and animals, having acceptable image-to-video registration accuracy.

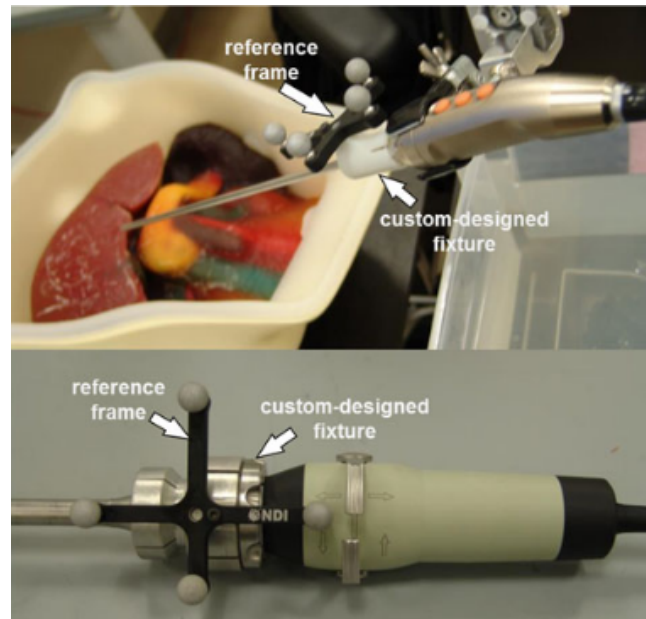


Figure 7.6: Custom-designed fixtures, with reference frames attached, for the 3D laparoscope [24].

### 7.5.3 Near Infrared Optical Tracking System

In [43] a near infrared optical tracking system, to track surgical tools is designed with its corresponding algorithms. Two IR cameras and a calibration model is used. Each near IR camera be is calibrated to obtain the intrinsic and extrinsic parameters.

At first markers that are attached on surgical tools are captured by the two IR cameras. As the authors mentioned, matching the subpixel points before computing the 3D coordinates of markers, is a challenging problem. However, these values are needed to be estimated in high accuracy, as positions and orientations of surgical tools are obtained using them, in the global coordinate frame.

The performance of the above systems is characterized prominent [43], although it is highlighted that as accuracy of calibration directly determines the tracking results and thus, calibrated beforehand remains a critical aspect of optical tracking.

The designed system, which capture near IR images, using the two IR cameras and send them to a computer via Universal Serial Bus (USB) is presented in Figure 7.7.

### 7.5.4 Augmented reality navigation system for laparoscopic splenectomy in children

In [22], AR visualization that superimposed preoperative multidetector low computed tomography images onto captured laparoscopic live images, was constructed and presented in Figure 7.8. Optical markers were attached to the camera head for measuring its six DOF pose parameters using an optical tracker. The rotary encoder was geared to the scopecylinder for measuring the rotation parameters. Afterwards, directions of laparo-

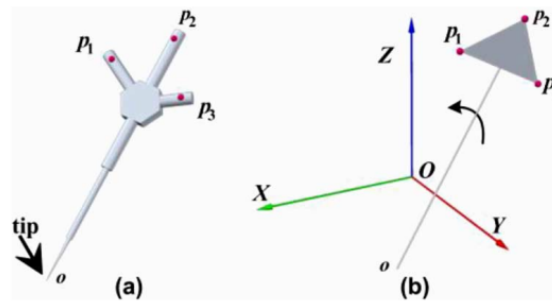


Figure 7.7: Schematic of surgical tool and the coordinate relationship between the tip of surgical tool and markers [43].

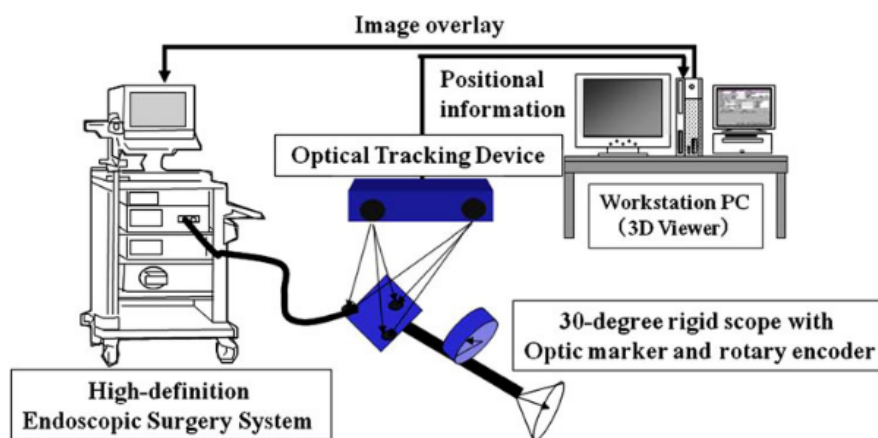


Figure 7.8: System configuration of AR navigation for laparoscopic splenectomy [22].

scopic views were calculated. By using the measurement of the rotation parameters, the navigation images were able to correspond to changes of the field of view.

The presented navigation system, as mentioned in [22] provides real-time anatomical information, that cannot be otherwise visualized without navigation. The registration accuracy seems to be in an acceptable level in clinical laparoscopic operation. However, the authors highlighted, that it is required to increase the system's accuracy.



## Chapter 8

# Applying optimized 3D Points Configurations for Pose Estimation on Laparoscopic Surgeries

In this Chapter, we present fiducial markers, which are used or proposed in medical applications in general. Then we analyze specifically the marker configurations applied on Laparoscopic surgeries for optical tracking. Thereafter, we describe our proposed marker configuration for tracking medical devices, *e.g.*, endoscope, in real-time. As it is said: "The most important criterion in surgical AR is accuracy" [7].

In Chapter 4, we concluded, that specific 3D points configurations can increase the accuracy of pose estimation. Hence, we decide to use our previous results, to create a simulation of a fiducial marker, that can be used in a Laparoscopic Surgery to increase the accuracy of the optimal tracking.

### 8.1 Designing our Fiducials

As the accuracy is a critical issue in surgical and most specific laparoscopic AR, we examine whether our "well-conditioned" (after gradient descent optimization) 3D points configurations, that we calculate in Chapter 4, will lead to minimum errors in a simulated laparoscope surgery than the ill-conditioned (random initial points).

To begin with, according to [6], more fiducials can improve the accuracy, as they are decreasing the covariance error by:

$$\frac{1}{\sqrt{n}}, \text{ where } n = \text{number of fiducials.} \quad (8.1)$$

On the other hand, as the number of fiducials is increased, the occurrence of occlusions between the fiducials is also increased. Thus, we decide to use 6 markers (instead of four

Size [mm]	Size [1/64 in]
12	30.2
14	35.3
16	40.3
20	50.4
30	75.6

Table 8.1: Available Sizes for Passive Spherical Markers in the Market.

that are mainly used), as six is the minimum number of points, that required to be applied on the DLT, to solve the PnP problem.

The markers we suggest are passive -as that kind of markers is most commonly used in tracking systems-, which means they reflect incoming IR radiation back in the direction of the incoming light. They have a spherical shape and they are covered with retroreflecting foils. To decide the radius of our spherical markers, we search the available sizes of spherical trackers in the market nowadays. All the available passive spherical markers' sizes are presented in the Table 8.1.

However, a coated version of the markers with a robust surface is available only in the following sizes: 0.014 m, 0.016 m, 0.02 m.

Thus, we choose to have 6 spherical markers with a diameter of 1.6 cm.

These passive 6 markers will be attached to a laparoscope having one of the shapes that are shown in Figure 8.1, in order to track the endoscope in real-time. Specifically, they will have an optimal shape of a truncated, normal or twisted triangular prism.

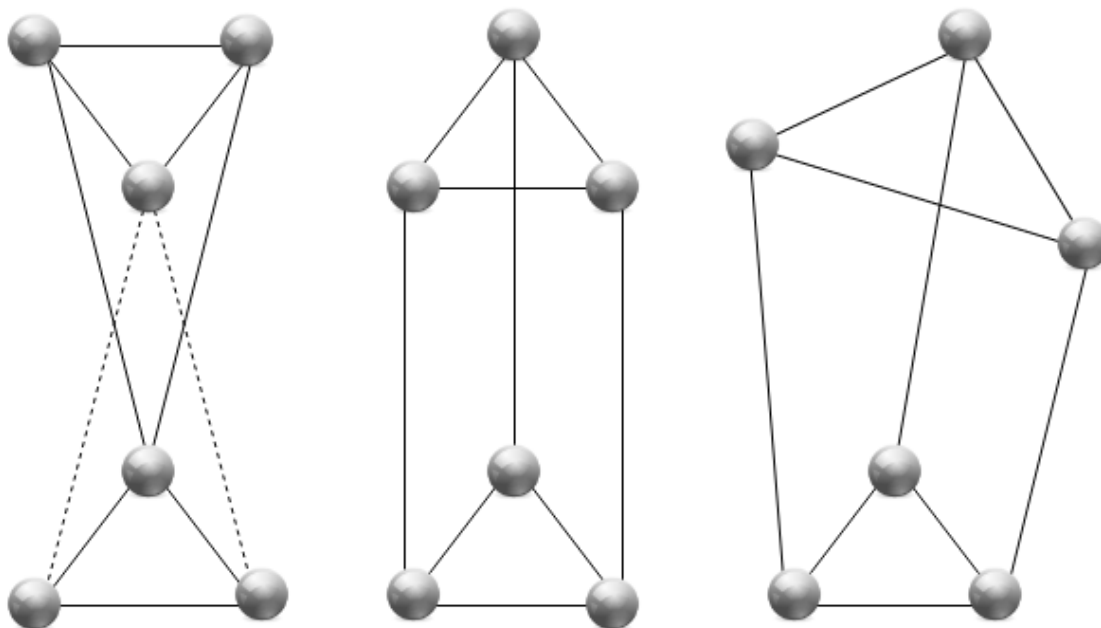


Figure 8.1: Our Proposed 3D Point Configurations to be attached on the Laparoscope.

To calculate the base of the prism, we need to know beforehand the distance between each one of the six markers. Therefore, we search the length of a laparoscope, as our fiducials will be attached on that. There are many laparoscope in the market, but we decide to use the Olympus Laparoscopes HD and by reading the product details, we find that the free space for the markers placement is 10 cm.

Thus, using the Pythagorean theorem we find the radius of the sphere in which all the markers are inscribed. If we set  $a = 10$  cm in Figure 8.2b, then we will have the following equation:

$$a^2 = r^2 + r^2 \implies 10^2 = 2r^2 \implies r = 7.071\text{cm} = 0.07071\text{m}. \quad (8.2)$$

So the 6 points will belong to a sphere with a radius of 0.070 71 m, as presented in Figure 8.3.

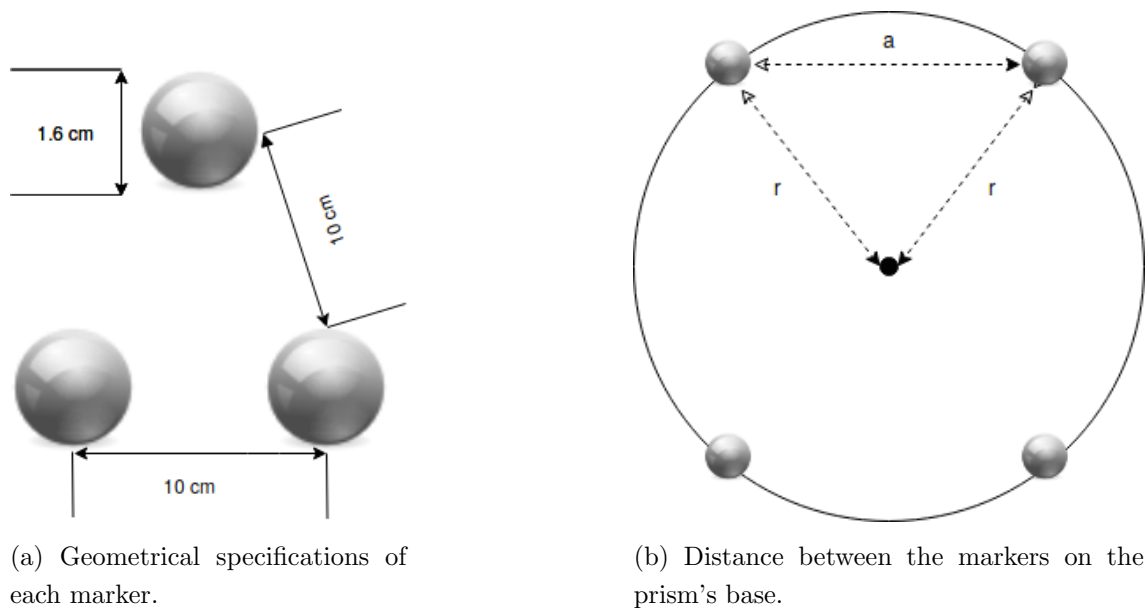


Figure 8.2: Fiducial's Geometry.

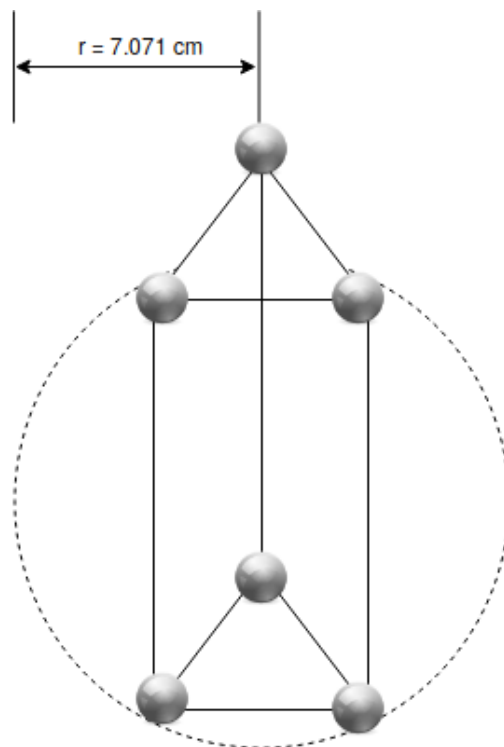


Figure 8.3: Final Points Configuration Inscribed in a Sphere.

## 8.2 Re-validate the Fiducials Configuration using Gradient Descent Optimization Algorithm

Radius  $r$  is set as 0.070 71 m and the minimum separate distance between the selected points equals to the diameter of each spherical marker which is 0.016 m. Afterwards, as the radius is now ten times smaller than in our previous tests, we decrease ten times the step in the gradient descent algorithm and the decreasing factor in the superSAB algorithm.

Then, we run the gradient descent optimization algorithm for  $N = 100$  iterations, using the condition number as the optimization’s objective, in order to cross-confirm that the optimal shape is still a prism near the sphere’s periphery.

The camera pose is set with an angle of 90 degrees, as this position was found to improves the accuracy. The camera is chosen to be placed 0.5 m away from the markers. Again, we add the same 1000 samples of random Gaussian noise with a mean equals to 2 and a standard deviation of 8 to every image pixel, and we find the average value of each one of them. The results from our tests are presented below.

Specifically, Figure 8.4 shows the first iteration of the algorithm ( $N = 1$ ), where the points are randomly selected. Then, in Figure 8.5 the points after 50 iterations of the gradient descent optimization algorithm are presented. Finally, Figure 8.6 shows the final points configuration, which once again is a triangular prism, near the perimeter of the sphere.

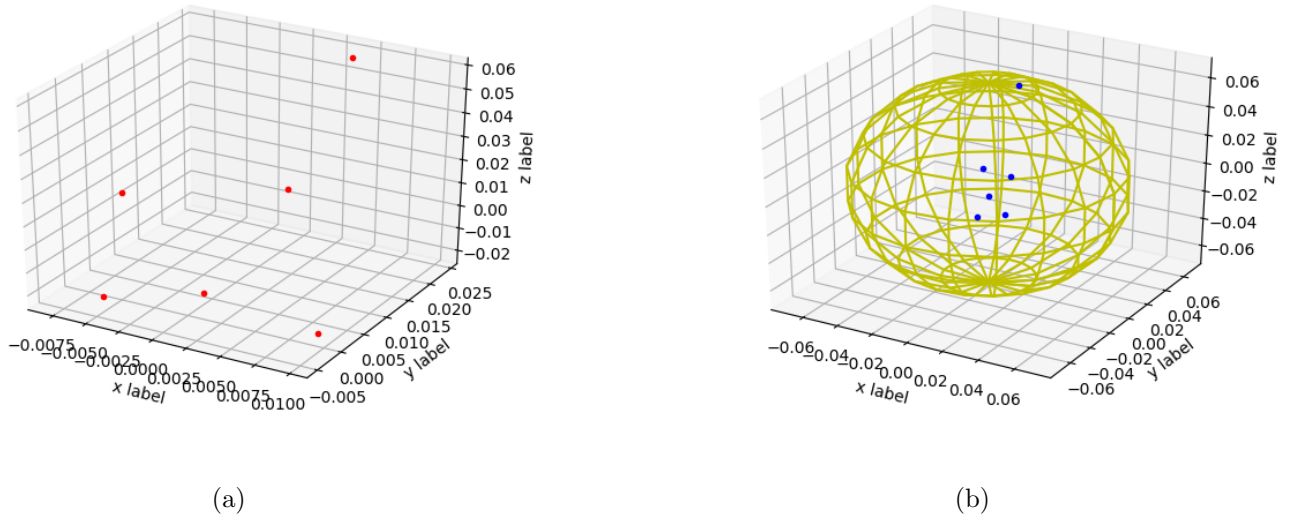


Figure 8.4: "Ill-conditioned" random initial points ( $N = 1$ ).

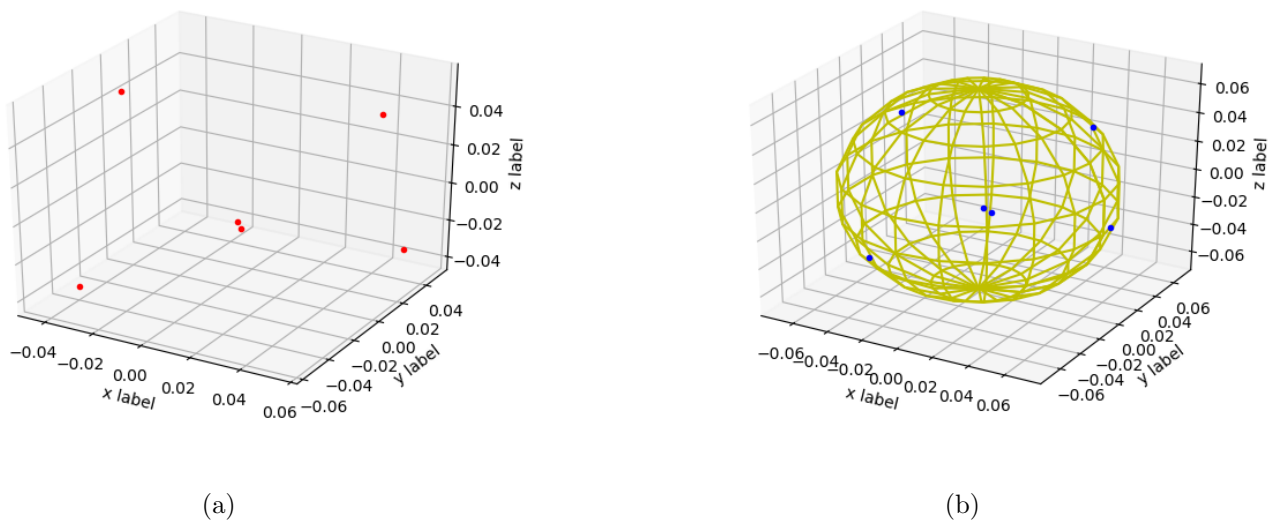


Figure 8.5: Points Configuration at  $N = 50$ .

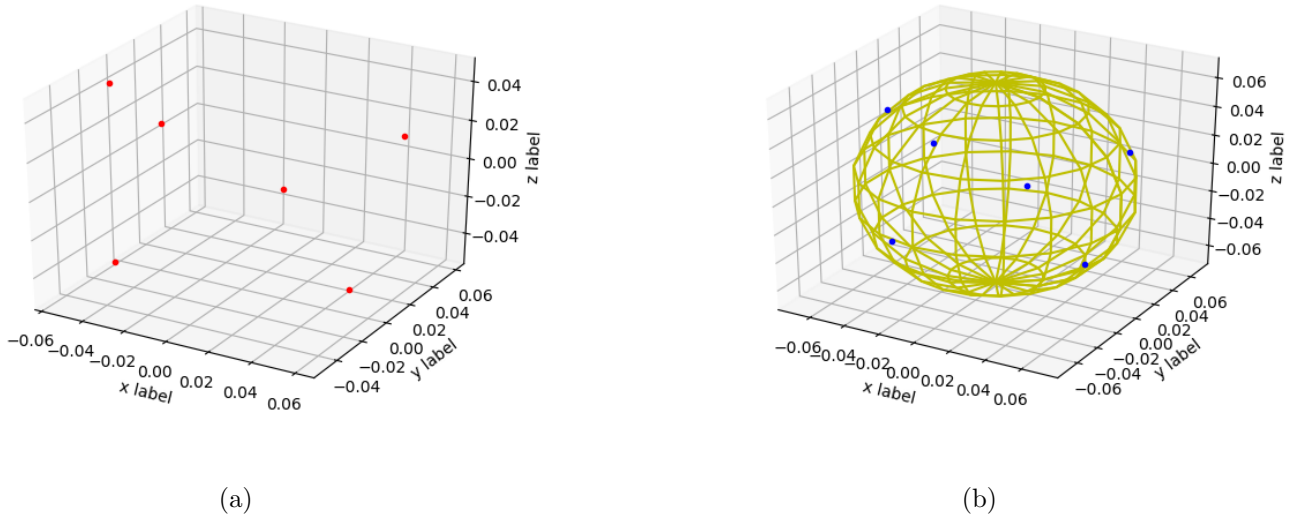
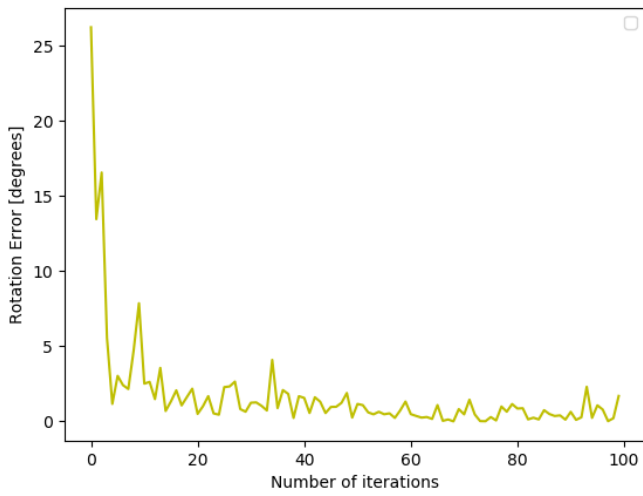
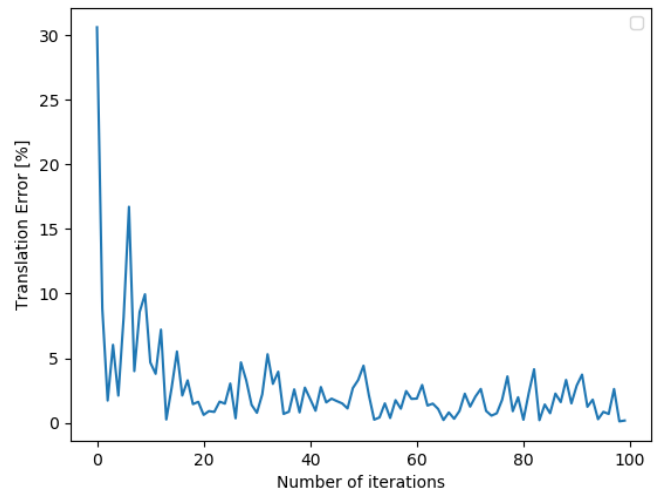


Figure 8.6: The final Points Configuration at  $N = 100$ .

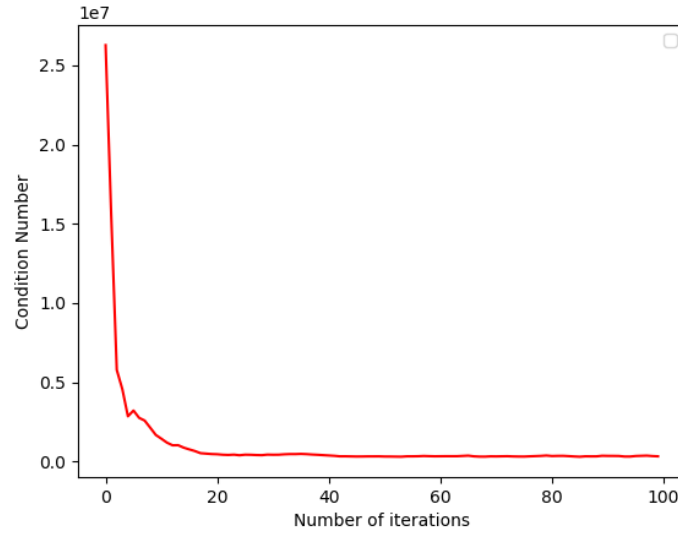
The errors in translation, rotation and the condition number are also measured for these 100 iterations, using the new defined sphere and the new position of the camera and we present them below. As the number of iterations is increased, the algorithm measures the minimum value of condition number, which also lead to the minimum values in RE and TE. These results are available in Figure 8.6.



(a) Rotational Error.



(b) Error in Translation.



(c) Condition Number.

Figure 8.7: Errors in Gradient Descent Optimization  $X \sim \mathcal{N}(2, 8)$ .

## 8.3 Comparing Errors between well- and ill-conditioned Points Configurations

As the optimal shape was proved to be once again a triangular prism, a function is created, that selects 6 random points, which create a triangular prism inscribed each time to the defined sphere.

### 8.3.1 Root Mean Square Errors

First, we create 1000 sets of random prism points configurations (well-conditioned) and 1000 totally random sets of points (ill-conditioned) and we compare their Root Mean Square Error based on the equation (3.15).

In every test we cross-confirm that the RMS Error of the prism points was smaller than the same error of the random sphere points. More specific in our test we estimate that:

$$e_{RMS,prism} \approx 1674.07611847 px \implies e_{RMS,prism} \approx 0.442933 m,$$

$$e_{RMS,sphere} \approx 1721.83215184 px \implies e_{RMS,sphere} \approx 0.455568 m.$$

### 8.3.2 Translation Errors and Rotational Errors

We randomly select 100 set of 6-points, which belong to our sphere class with a radius of 0.07071m and generate the same number of sets of 6-points from our prism-points generator function. We add again, 1000 samples of random Gaussian noise with a standard

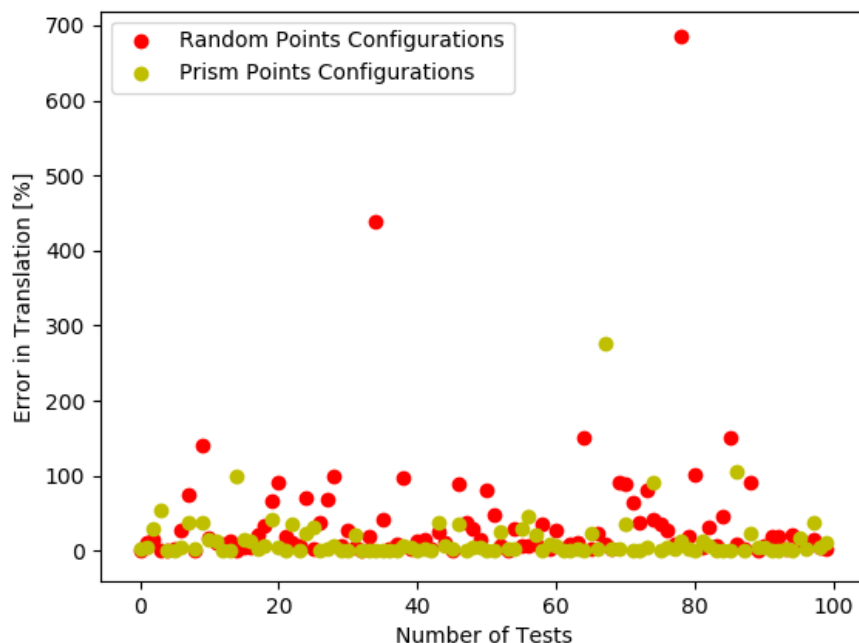


Figure 8.8: Comparing Errors in Translation.

deviation equals 4 and mean equals 2 to each image pixel and we measure their average value. Then, we apply these points on the DLT and we estimate the TE and RE.

Afterwards, we compare each sphere set's TE with the same prism set's error.

Our results are presented in Figure 8.8. Hence, we can see that in more than 95 % of the comparisons, the Sphere Point Configurations are causing higher TE. Moreover, we can see that the Prism Point Configuration does not cause any TE bigger than 93 %, while at least once of the Sphere Point Configurations cause TE higher than 100 %.

In addition, our results for the RE are shown in Figure 8.9. Here we can see even bigger differences between the two points configurations. Specifically in more than 97 % of the comparisons the sphere point configurations lead to higher errors in rotation.



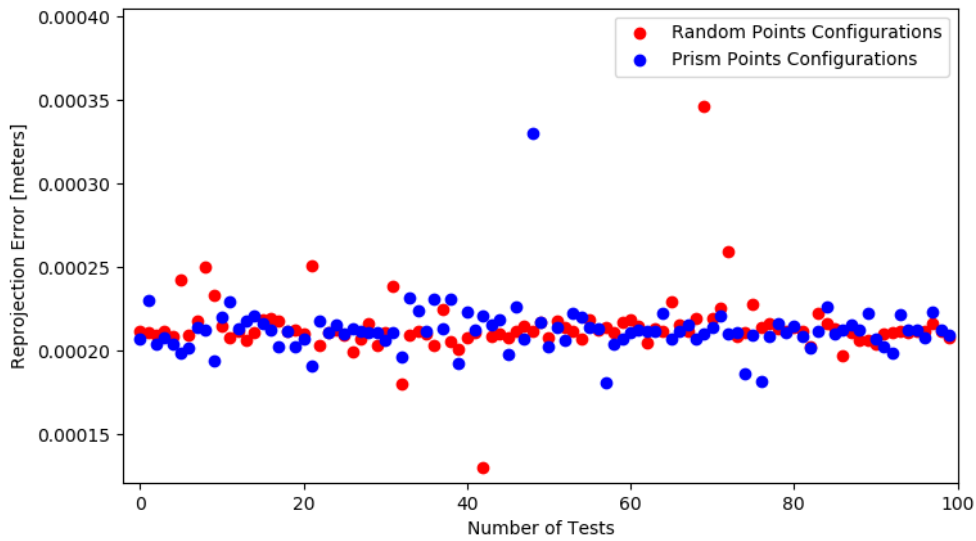


Figure 8.10: Comparing Reprojection Errors.

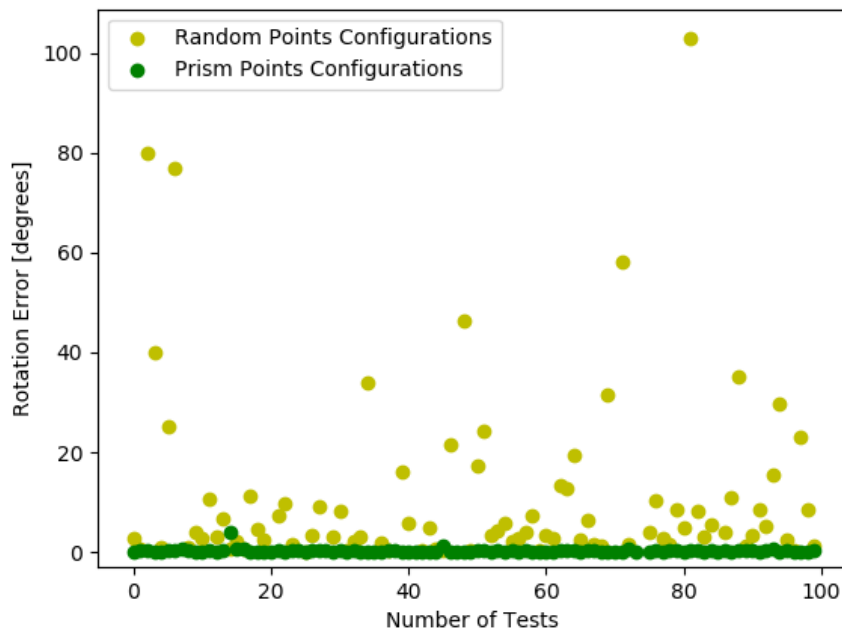
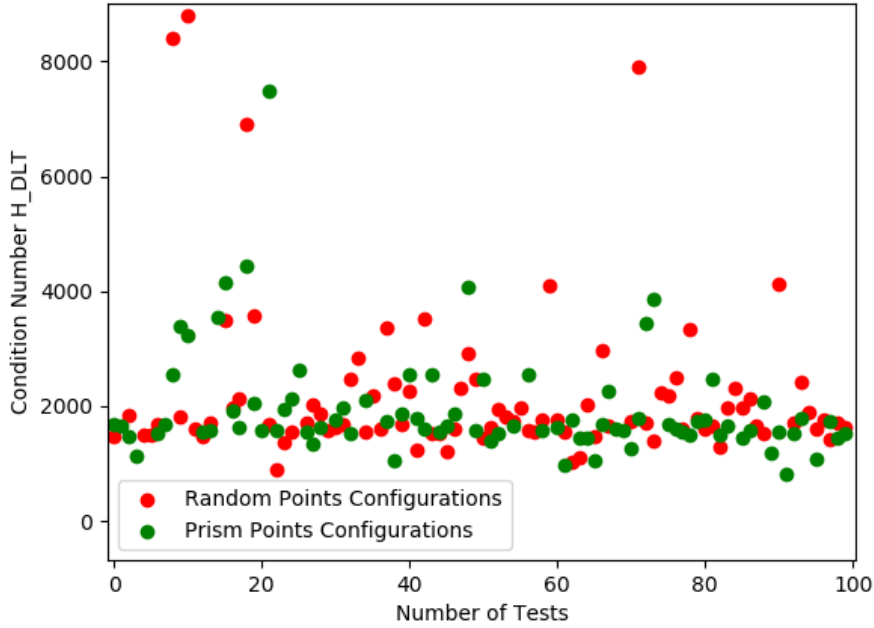


Figure 8.9: Comparing Rotational Errors.

### 8.3.3 Reprojection Errors

Using the above sets of points and the methodology, that was explained in the section 2.6.4, we estimate the Reprojection Error of each set of points in px. Then we convert px to m by multiplying each with pixel with 0.0002645833. In Figure 8.10, the Reprojec-

Figure 8.11: Comparing Condition Numbers of  $H_{DLT}$ .

tection Errors for the two different points configurations are presented. The prism points configurations lead at the 90 % of the tests in minimum Reprojection Error than the random generated sphere points configurations.

Thereafter, we calculate the mean errors in translation, rotation and reprojection, for these 100 sets of sphere points and 100 sets of prism points. In every simulation which was executed, the measured mean errors of the sphere points configurations, were higher than the mean errors of the prism points configurations, which was also validate through the above Figures. The mean errors are presented in the Table 8.2.

Measured Mean Errors	Sphere Points Configurations	Prism Points Configurations
Mean Rotational Error [°]	37.8125	9.2809
Mean Error in Translation [%]	79.0060	14.1995
Mean Reprojection Error [cm]	0.02143	0.02116

Table 8.2: Mean Errors of Sphere- and Prism- Points Configurations.

### 8.3.4 Condition Number and Condition Number of Covariance Matrix

After applying the above random sets of points on the DLT, we extract the  $H_{DLT}$  matrix. Then, we calculate for each set the condition number of this matrix, as well as, the condition number of the covariance matrix of  $H_{DLT}$ .

In Figure 8.11<sup>1</sup>, we present the condition numbers of  $H_{\text{DLT}}$  of all the sphere- and prism-points configurations. In more than 97 % of the comparisons, sphere points configurations lead to higher condition numbers, proving that these set of points are "ill-conditioned" compared to the prism points configurations.

Moreover, Figure 8.12<sup>2</sup> shows the condition numbers of the covariance matrices of each  $H_{\text{DLT}}$ , which was extracted by applying the different set of points to DLT. The sphere points configurations lead once again in higher values in the condition number in more than 95 % of the comparisons.

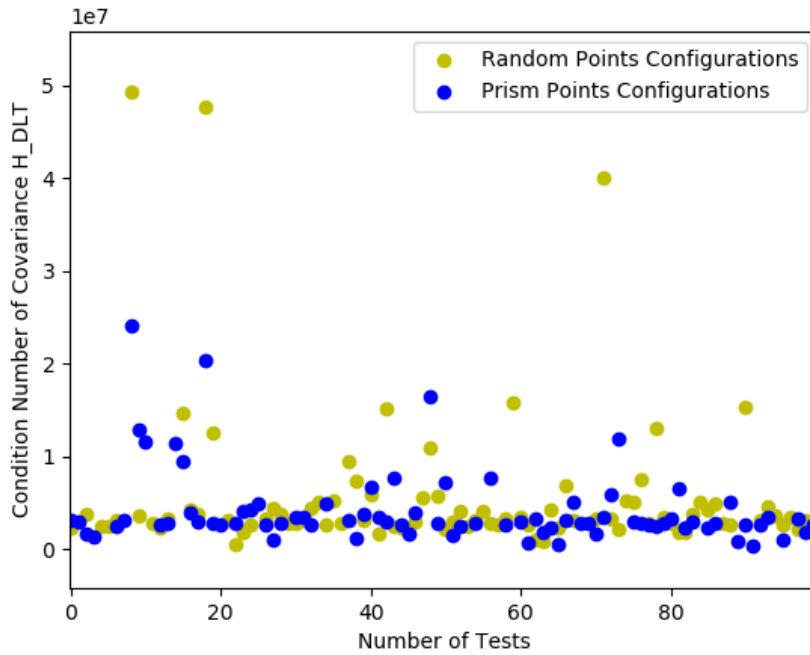


Figure 8.12: Comparing Condition Numbers of Covariance Matrices of  $H_{\text{DLT}}$ .

## 8.4 Concluding Remarks

In this section we applied the 3D points configurations, which were estimated in Chapter 4, specifically in the laparoscopic case. First, we adjust all the parameters (*e.g.*, sphere radius, separate distance between points), in order to simulate a laparoscopic surgery case.

Afterwards, we validate that triangular prism is the optimal shape for the 6-sphere points and we design our proposed marker. The marker will be attached on the laparoscope to provide a real-time optical tracking. It consists of six IR spherical fiducials as presented in Figure 8.13.

We also compared the orientation errors caused by the marker with the errors caused by random points configurations, which were belonging to the circumscribed sphere in the

<sup>1</sup>The maximum values of random points Condition Number are not included.

<sup>2</sup>The maximum values of random points Covariance Matrices are not included.



Figure 8.13: Our Proposed Fiducial Marker attached on a Laparoscope.

triangular prism of the marker. Thus, we found out that the orientation errors, which were caused by using the proposed marker had always lower values.

## Chapter 9

# Conclusion and Future Works

In this final, concluding chapter, we present an overview of this thesis, as well as our results and consequences. Additionally, we denote a number of aspects which have not been considered in the thesis, and we suggest to be addressed in follow-up works.

### 9.1 Conclusion

Improving the accuracy in surgical AR remains a critical challenge, as the real-time localization (*i.e.*, position and orientation) of medical instruments should be guaranteed. Thus, in this work we have addressed the above challenges regarding pose estimation and accuracy in optical tracking.

Specifically, 3D points configurations, which can improve the accuracy in pose estimation were examined. These configurations were applied on a simulated laparoscopic surgery case to contribute to the field of optical tracking in surgical AR.

To achieve this goal, the existence of optimized six points configurations, which can increase the accuracy and robustness of space resection methods were examined, as well as the relation between the pose of the camera and accuracy.

At first, a pinhole camera model and a sphere were defined in Python and we selected randomly points, which were belonged to this sphere. Then we found the image projected points of those world-points and we added random Gaussian noise and applied those points on the DLT. After that, we measured the orientation and reprojection errors, and the condition number of the final  $\mathbf{H}_{DLT}$  and of its covariance matrix.

Then, we changed the position of the camera (*i.e.*, distance from the marker(s) and angle-view), and we were keep tracking the effect of these changes in the measured errors. Hence, we ended up, finding the pose of the camera that causes the minimum errors, which is to be placed less than a meter away from the marker(s) and with a 90 degrees angle-view. Moreover, placing the camera on the top of the marker, with a zero angle-view was estimated to be the worst position as it was proven to increase the measured errors.

Secondly, we used three optimization methods to find optimized six 3D points configurations, that can decrease the above errors and, thus, improve the accuracy. To begin with,

we tested many random points selection from our defined sphere and we observed that the points near sphere periphery were having lower errors, than the other points. Thus, we used Gradient Descent Optimization and Adam Optimization algorithms to verify those observations.

Both algorithms showed that optimized points configurations were close to the sphere periphery. In addition, it was also shown that the optimal shape for 6 3D points configurations was a triangular prism(normal, truncated or twisted). Furthermore, the triangular prism configuration was proved to be optimal for every pose of the camera.

Afterwards, we applied our results in a simulated laparoscopic surgery case. Specifically, we proposed a triangular prism marker to be used for optical tracking in real-time medical instruments localization. The marker is passive and consists of 6 spherical IR trackers, which will be attached on instruments (*e.g.*, laparoscope). The sphere that was circumscribed to that triangular prism was also calculated. Then, all the measured errors(rotation, translation, reprojection) and the condition number were measured for this marker configuration, and they were compared with random points configurations that were belonging to the same circumscribed sphere in the triangular prism.

As shown, our proposed marker was leading to minimum errors in every test. In addition the mean errors(in rotation, translation and reprojection) were measured and the triangular prisms configurations had always lower mean error values than the random configurations of circumscribed sphere.

## 9.2 Outlook

We would also like to highlight some considerations on the field of 3D Point Configurations in general and specifically in laparoscopy, which should be examined in follow-up works.

- We suggest to create a 3D printed model of the proposed fiducial marker, in order to validate our results in a lab-simulated situation.
- In our simulations we use one single camera. However, in Laparoscopic Surgeries, two cameras are usually used. Thus, we suggest an experimental evaluation of our markers in Laparoscopic Surgeries utilizing two cameras. a
- Currently, 4-points configurations are mainly used for tracking the medical instruments. Thus, a cost-effectiveness research for our proposed 6-points configuration marker is required.
- Even though our simulations validate that our proposed Fiducial Marker can improve the accuracy of optical tracking, the evaluation of accuracy in a clinical set-up is required.
- Increasing the number of control points increases the accuracy of the results in presence of noise. Thus, all of our simulations can be executed using 8-points con-

figurations instead of 6. Then, new questions are raised, regarding the existence of an optimal shape for 8-points and whether the pose of the camera affect the well-conditioned 8-points configurations.

- In our work, we select the condition number to be the optimization's objective. However, other metrics such as the reprojection error or the covariance matrix can be selected as optimization objectives.





# Bibliography

- [1] Raul Acuna and Volker Willert. “Robustness of control point configurations for homography and planar pose estimation”. In: *CoRR* abs/1803.03025 (2019).
- [2] Omar Alter, Patrick Brown, and David Botstein. “Singular Value Decomposition for Genome-Wide Expression Data Processing and Modeling”. In: *Proceedings of the National Academy of Sciences of the United States of America* 97 (Sept. 2000), pp. 10101–6. DOI: 10.1073/pnas.97.18.10101.
- [3] Jasbir Singh Arora. *Introduction to Optimum Design*. 3rd ed. Academic Press, 2011. ISBN: 9780123813756.
- [4] Deep Ideas Authors. *Deep Learning From Scratch IV: Gradient Descent and Backpropagation*. URL: <http://www.deepideas.net/deep-learning-from-scratch-iv-gradient-descent-and-backpropagation/>. (accessed: 24.05.2019).
- [5] David Balihar. *WHAT IS A PINHOLE CAMERA?* URL: <https://www.pinhole.cz/en/pinholecameras/whatis.html>. (accessed: 24.05.2019).
- [6] Martin A. Bauer. “Tracking Errors in Augmented Reality”. PhD thesis. Technische Universität München, Institut für Informatik, 2007.
- [7] Sylvain Bernhardt et al. “The status of augmented reality in laparoscopic surgery as of 2016”. In: *Medical Image Analysis* 37 (Jan. 2017). DOI: 10.1016/j.media.2017.01.007.
- [8] Aaron Bobick. *Calibration and Projective Geometry*.
- [9] Jason Brownlee. *Gentle Introduction to the Adam Optimization Algorithm for Deep Learning*. URL: <https://machinelearningmastery.com/adam-optimization-algorithm-for-deep-learning/>. (accessed: 24.05.2019).
- [10] Daniel Burkhardt Cerigo. *On Why Gradient Descent is Even Needed*. URL: <https://medium.com/@DBCerigo/on-why-gradient-descent-is-even-needed-25160197a635>. (accessed: 23.05.2019).
- [11] Robert Collins. *Lecture 12: Camera Projection, CSE486, Penn State*.
- [12] MathWorks contributors. *What Is Camera Calibration?* URL: <https://de.mathworks.com/help/vision/ug/camera-calibration.html>. (accessed: 24.05.2019).

- [13] Marco Feuerstein. “Augmented Reality in Laparoscopic Surgery. New Concepts for Intraoperative Multimodal Imaging”. PhD thesis. Technische Universität München, Institut für Informatik, 2007.
- [14] Jan Fischer, Dirk Bartz, and Wolfgang Straßer. “Occlusion handling for medical augmented reality using a volumetric phantom model”. In: Jan. 2004, pp. 174–177. DOI: 10.1145/1077534.1077570.
- [15] Chris Flask et al. “A method for fast 3D tracking using tuned fiducial markers and a limited projection reconstruction FISP (LPR-FISP) sequence”. In: *Journal of Magnetic Resonance Imaging* 14.5 (2001), pp. 617–627. DOI: 10.1002/jmri.1227.
- [16] Alfred Franz et al. “Electromagnetic Tracking in Medicine-A Review of Technology, Validation, and Applications”. In: *IEEE Transactions on Medical Imaging* (May 2014). DOI: 10.1109/TMI.2014.2321777.
- [17] A. Gadwe and H. Ren. “Real-Time 6DOF Pose Estimation of Endoscopic Instruments Using Printable Markers”. In: *IEEE Sensors Journal* 19.6 (Mar. 2019), pp. 2338–2346. ISSN: 1530-437X. DOI: 10.1109/JSEN.2018.2886418.
- [18] Sergio Garrido-Jurado et al. “Automatic generation and detection of highly reliable fiducial markers under occlusion”. In: *Pattern Recognition* 47 (June 2014), pp. 2280–2292. DOI: 10.1016/j.patcog.2014.01.005.
- [19] Neil Glossop. “Advantages of Optical Compared with Electromagnetic Tracking”. In: *The Journal of bone and joint surgery. American volume* 91 Suppl 1 (Mar. 2009), pp. 23–8. DOI: 10.2106/JBJS.H.01362.
- [20] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. New York, NY, USA: Cambridge University Press, 2003. ISBN: 0521540518.
- [21] William Hoff. *Computer Vision*.
- [22] Satoshi Ieiri et al. “Augmented reality navigation system for laparoscopic splenectomy in children based on preoperative CT image using optical tracking device”. In: *Pediatric surgery international* 28 (Dec. 2011), pp. 341–6. DOI: 10.1007/s00383-011-3034-x.
- [23] Nikolai Janakiev. *Understanding the Covariance Matrix*. URL: <https://datascienceplus.com/understanding-the-covariance-matrix/>. (accessed: 24.05.2019).
- [24] Xin Kang et al. “Stereoscopic augmented reality for laparoscopic surgery”. In: *Surgical endoscopy* 28 (Feb. 2014). DOI: 10.1007/s00464-014-3433-x.
- [25] Diederik Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).
- [26] ksimek. *Dissecting the Camera Matrix, Part 3: The Intrinsic Matrix*. URL: <http://ksimek.github.io/2013/08/13/intrinsic/>. (accessed: 23.05.2019).

- [27] *Lecture 3: Camera Calibration, DLT, SVD*. Jan. 2013.
- [28] Peter Lightbody, Tomáš Krajiník, and Marc Hanheide. “An efficient visual fiducial localisation system”. In: *ACM SIGAPP Applied Computing Review* 17 (Nov. 2017), pp. 28–37. DOI: 10.1145/3161534.3161537.
- [29] E. Marchand, H. Uchiyama, and F. Spindler. “Pose Estimation for Augmented Reality: A Hands-On Survey”. In: *IEEE Transactions on Visualization and Computer Graphics* 22.12 (Dec. 2016), pp. 2633–2651. ISSN: 1077-2626. DOI: 10.1109/TVCG.2015.2513408.
- [30] mattjj. *autograd/misc*. 2017. URL: <https://github.com/HIPS/autograd/tree/master/autograd/misc>.
- [31] Cleve Moler. *What is the Condition Number of a Matrix?* URL: <https://blogs.mathworks.com/cleve/2017/07/17/what-is-the-condition-number-of-a-matrix/>. (accessed: 24.05.2019).
- [32] Michael A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.
- [33] ETS Project. *Mayavi: 3D scientific data visualization and plotting in Python*. 2008. URL: <https://docs.enthought.com/mayavi/mayavi/>.
- [34] J. Sattar et al. “Fourier tags: Smoothly degradable fiducial markers for use in human-robot interaction”. In: *Fourth Canadian Conference on Computer and Robot Vision (CRV '07)*. May 2007, pp. 165–174. DOI: 10.1109/CRV.2007.34.
- [35] Jürgen Schmidhuber. “Deep learning in neural networks: An overview”. In: *Neural Networks* 61 (2015), pp. 85–117. ISSN: 0893-6080. DOI: <https://doi.org/10.1016/j.neunet.2014.09.003>.
- [36] Jan A. Snyman. *Practical Mathematical Optimization*. 2nd ed. University of Pretoria, Pretoria, South Africa, 2005. ISBN: 9780387243498. DOI: 10.1007/B105200.
- [37] Tom Tollenaere. “SuperSAB: Fast adaptive back propagation with good scaling properties”. In: *Neural Networks* 3.5 (1990), pp. 561–573. ISSN: 0893-6080. DOI: 10.1016/0893-6080(90)90006-7.
- [38] P Vávra et al. “Recent Development of Augmented Reality in Surgery: A Review”. In: *Journal of Healthcare Engineering* 2017 (Aug. 2017), pp. 1–9. DOI: 10.1155/2017/4574172.
- [39] Eric W. Weisstein. *QR decomposition*. URL: <http://mathworld.wolfram.com/QRDecomposition.html>. (accessed: 23.05.2019).
- [40] Eric W. Weisstein. *Triangular prism*. URL: <http://mathworld.wolfram.com/TriangularPrism.html>. (accessed: 11.06.2019).
- [41] Xiao Xin Lu. “A Review of Solutions for Perspective-n-Point Problem in Camera Pose Estimation”. In: *Journal of Physics: Conference Series* 1087 (Sept. 2018), p. 052009. DOI: 10.1088/1742-6596/1087/5/052009.

- 
- [42] Xu Wenli and Zhang Lihua. “Pose estimation problem in computer vision”. In: *Proceedings of TENCON '93. IEEE Region 10 International Conference on Computers, Communications and Automation*. Vol. 2. Oct. 1993, 1138–1141 vol.2. DOI: 10.1109/TENCON.1993.320206.
- [43] R. Yang et al. “Design of an Accurate Near Infrared Optical Tracking System in Surgical Navigation”. In: *Journal of Lightwave Technology* 31.2 (Jan. 2013), pp. 223–231. ISSN: 0733-8724. DOI: 10.1109/JLT.2012.2227943.

# Απόδοση ξενόγλωσσων όρων

## Απόδοση

Εντοπισμός στο χώρο  
Απόφραξη  
Αριθμός Συνθήκης ενός Πίνακα  
Βασικοί Δείκτες  
Επαυξημένη Πραγματικότητα  
Κάμερα με Οπές  
Εξωγενείς Παράμετροι  
Ενδογενείς Παράμετροι  
Εστιακό Μήκος  
Λοξότητα  
Ομοιογενείς Συντεταγμένες  
Άμεσος Γραμμικός Μετασχηματισμός  
Αποσύνθεση Μοναδικής Τιμής

## Ξενόγλωσσος όρος

Localization  
Occlusion  
condition number  
Fiducial Markers  
Augmented Reality  
Pinhole Camera  
Extrinsic Parameters  
Intrinsic Parameters  
Focal Length  
Skew  
Homogeneous Coordinates  
Direct Linear Transformation  
Singular Value Decomposition



# Acronyms

**2D** 2 Dimensions. 50–52, 56, 73–75, 90, 91, 95

**3D** 3 Dimensions. 41–43, 46, 47, 49–51, 53, 56, 57, 59, 61, 73–78, 80, 82–84, 86, 88, 90, 91, 95, 99, 100, 109, 111, 112

**AR** Augmented Reality. 41, 42, 45, 46, 51, 91, 92, 94, 96–99, 111

**DLT** Direct Linear Transformation. 45, 50–52, 59, 61, 64, 73–75, 77, 80, 84, 88, 90, 100, 106, 108, 109, 111

**DOF** Degrees-Of-Freedom. 51, 95, 97

**EPnP** Efficient Perspective-n-Point. 50

**IR** Infrared. 94, 95, 97, 100, 112

**MIS** Minimally Invasive Surgery. 42, 43

**PCB** Printed Circuit Board. 41

**PnP** Perspective-n-Point. 43, 50–52, 61, 63, 64, 66, 68, 70, 73, 100

**RE** Rotational Error. 52, 73, 74, 77, 78, 80, 84, 85, 88, 90, 104–106

**RMS** Root Mean Square. 14, 36, 51–53, 77, 105

**SVD** Singular Value Decomposition. 51, 62

**TE** Translation Error. 53, 73, 74, 77, 78, 80, 84, 85, 88, 90, 104–106

**USB** Universal Serial Bus. 97