



ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ  
ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ & ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ –  
ΜΗΧΑΝΙΚΩΝ ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ  
ΤΟΜΕΑΣ ΤΟΠΟΓΡΑΦΙΑΣ  
ΕΡΓΑΣΤΗΡΙΟ ΦΩΤΟΓΡΑΜΜΕΤΡΙΑΣ

**‘ΑΝΑΠΤΥΞΗ ΕΦΑΡΜΟΓΗΣ ΕΠΙΛΥΣΗΣ ΦΩΤΟΓΡΑΜΜΕΤΡΙΚΩΝ  
ΠΡΟΒΛΗΜΑΤΩΝ ΣΕ ΣΥΣΚΕΥΕΣ ANDROID’**

**‘APPLICATION DEVELOPMENT FOR SOLVING  
PHOTOGRAMMETRIC PROBLEMS IN ANDROID DEVICES’**

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΤΟΥ

**Στέφανου Θανόπουλου**

Επιβλέπων: Ανδρέας Γεωργόπουλος, Καθηγητής Ε.Μ.Π.

ΑΘΗΝΑ, ΙΟΥΛΙΟΣ 2021



## ΕΥΧΑΡΙΣΤΙΕΣ

Η παρούσα διπλωματική εργασία ανατέθηκε από το Εργαστήριο Φωτογραμμετρίας του Τομέα Τοπογραφίας της σχολής Αγρονόμων και Τοπογράφων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, με σκοπό τη μελέτη σχετικά με τον προσδιορισμό των παραμέτρων του εξωτερικού προσανατολισμού και των συντεταγμένων σημείων που φωτογραφίζονται με την φωτογραφική μηχανή ενός smartphone που είναι συμβατό με το λειτουργικό σύστημα ANDROID.

Ευχαριστώ θερμά:

Τον κ. Ανδρέα Γεωργόπουλο, Καθηγητή του Τομέα Τοπογραφίας της σχολής Αγρονόμων και Τοπογράφων Μηχανικών του Εθνικού Μετσόβιου Πολυτεχνείου, επιβλέποντα καθηγητή, για την ανάθεση του θέματος, τις πολύτιμες συμβουλές κατά τη διαμόρφωση της παρούσας εργασίας, τη συμπαράσταση, τη συνεχή καθοδήγηση και τη βοήθειά του καθ' όλη τη διάρκεια της εκπόνησης της παρούσας διπλωματικής εργασίας.

Τον καλό μου φίλο Σεβαστιανό για την υπέροχη καθοδήγηση στα 'μονοπάτια' του προγραμματισμού.

Τους φίλους μου για όλες τις αξέχαστες στιγμές που μου πρόσφεραν κατά τη διάρκεια των σπουδών μου.

Την αδερφή μου που πιστεύει σε μένα.

Τη Γιάννα που πιστεύει σε μένα και μου προσέφερε τη πολύτιμη βοήθειά της.

Τον Κωνσταντίνο που πάντα με στηρίζει αδιάκοπα.

Τη μητέρα μου, στην οποία οφείλω ό,τι έχω πετύχει στη μέχρι τώρα πορεία μου.



## ΠΕΡΙΛΗΨΗ

Η εντατικοποίηση των ρυθμών της καθημερινότητας των σύγχρονων ανθρώπων αλλά κι η ανάπτυξη της τεχνολογίας έχουν οδηγήσει στην εμφάνιση νέων αναγκών. Τις λύσεις στις νέες αυτές ανάγκες ήρθαν για να δώσουν τα 'έξυπνα' κινητά τηλέφωνα (smartphones), τα οποία μπορούν να παρέχουν πληθώρα εφαρμογών ώστε να διευκολύνουν και να καλυτερεύσουν την καθημερινότητα των χρηστών τους.

Το αντικείμενο της παρούσας διπλωματικής εργασίας είναι ο σχεδιασμός, η ανάπτυξη κι η υλοποίηση μίας εφαρμογής η οποία θα αποτελεί σημαντικό εργαλείο για τους μηχανικούς του 'χώρου' (Τοπογράφους, Πολιτικούς, Αρχιτέκτονες κτλ). Η εφαρμογή που δημιουργήθηκε ονομάστηκε 'Photogrammetry' και είναι συμβατή με κινητές συσκευές που χρησιμοποιούν λειτουργικό σύστημα Android.

Σκοπός αυτής της εφαρμογής (application) είναι η δημιουργία ενός αλγορίθμου, ο οποίος δεχόμενος τα κατάλληλα δεδομένα, να μπορεί να χρησιμοποιεί τη συνθήκη συγγραμμικότητας έτσι ώστε να επιλύει φωτογραμμετρική οπισθοτομία και εμπροσθοτομία. Με πιο απλά λόγια, στόχος της εφαρμογής είναι ο καλύτερος δυνατός εντοπισμός της θέσης και του προσανατολισμού του smartphone κατά τη στιγμή της λήψης κάθε φωτογραφίας καθώς και ο προσδιορισμός των 'πραγματικών' συντεταγμένων οποιουδήποτε σημείου θέλει ο χρήστης να βρει μέσω των προσανατολισμένων εικόνων.

Η υλοποίηση της εφαρμογής έγινε με τη βοήθεια ενός MAC σταθερού υπολογιστή κι ενός HP laptop. Επιπρόσθετα, χρησιμοποιήθηκε για προγραμματιστικό περιβάλλον το JAVA Eclipse, καθώς και το Android Studio (Android SDK). Οι δοκιμές έγιναν στο Κέντρο Μετρολογίας στο κτίριο Λαμπαδαρίου με τη βοήθεια ενός SAMSUNG GALAXY NOTE 10+ 'έξυπνου' κινητού τηλεφώνου.

## **ABSTRACT**

The intensification of the pace of modern people's everyday life, along with the development of technology, have led to the emergence of new needs. The solutions to these new needs are provided by 'smart' mobile phones (smartphones), which can offer a variety of applications to facilitate and improve the daily lives of their users.

The topic of this dissertation is the design, development and implementation of an application which will be an important tool for 'space' engineers (Surveyors, Civil Engineers, Architects, etc.). The application created was called 'PhotoTool' and it is compatible with mobile devices running the Android operating system.

The purpose of this application is to create an algorithm, which, receiving the appropriate data, can use the collinearity condition to solve photogrammetric resection and intersection. In simpler words, the goal of the application is to best locate the position and orientation of the smartphone at the time of capturing each photo, as well as to determine the 'real' coordinates of any point the user wants to find through the oriented images.

The implementation of the application was done with the help of a MAC desktop and an HP laptop. Additionally, JAVA Eclipse and Android Studio (Android SDK) were used as programming environments. The test was performed at the Metrology Center in the Lampadari building with the help of a SAMSUNG GALAXY NOTE 10+ smartphone.

## Περιεχόμενα

ΕΥΧΑΡΙΣΤΙΕΣ .....	3
ΠΕΡΙΛΗΨΗ .....	5
ABSTRACT .....	6
ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ .....	9
Προσδιορισμός 3D μετρήσεων με έξυπνες συσκευές– Ανασκόπηση βιβλιογραφίας .....	12
1.1 Έλεγχος GNSS και αισθητήρων προσδιορισμού γωνιών .....	12
1.2 Τρισδιάστατη απεικόνιση .....	14
1.3 Χειρουργική μέσω πλοήγησης .....	19
1.4 Συστήματα φωτογραμμετρίας UAV με χρήση smartphone και έξυπνες κάμερες .....	20
1.5 Μέτρηση στάθμης νερού από κάμερα smartphone .....	21
Προγραμματισμός συσκευών ANDROID .....	23
2.1 Τι είναι το Android ; .....	23
2.1.1 Τα λειτουργικά συστήματα κι οι εφαρμογές .....	25
2.1.2 Google Play Store .....	25
2.2 Βασικά στοιχεία προγραμματισμού .....	26
2.2.1 Μεταβλητές και τύποι δεδομένων .....	26
2.2.2 Δομές ελέγχου .....	27
2.2.3 Υπορουτίνες .....	27
2.3 Περιβάλλον εκτέλεσης κώδικα JAVA .....	27
2.3.1 Περιβάλλον και ανάπτυξη εφαρμογής .....	29
2.4 Αντικειμενοστραφής Προγραμματισμός .....	30
2.4.1 Αντικείμενα, κλάσεις, στιγμιότυπα .....	30
2.4.2 Συλλογή απορριμμάτων .....	31
2.5 Android Studio .....	31
2.5.1 Τα Android components και οι ‘Activities’ .....	32
Φωτογραμμετρία και Συνθήκη Συγγραμμικότητας .....	33
Ανάλυση .....	35
4.1 Η Εφαρμογή .....	35
4.2 Περιήγηση στο Android Studio .....	36

4.3 Μορφοποίηση της εφαρμογής .....	39
4.4 Κώδικας.....	44
4.4.1 Κλάσεις δεδομένων .....	44
4.4.2 Κλάση Οπισθοτομίας .....	46
4.4.3 Κλάση Εμπροσθοτομίας.....	52
4.4.4 Κλάση 'UserSession' .....	53
4.4.5 Κλάσεις Presenter .....	54
4.4.6 Κλάσεις Activities.....	55
4.5 Διαδικασία δοκιμής της εφαρμογής .....	58
4.6 Αποτελέσματα της εφαρμογής .....	61
Συμπερασματικά σχόλια .....	64
5.1 Αξιολόγηση και έλεγχοι .....	64
5.2 Μελλοντικές βελτιώσεις και προσθήκες .....	64
Βιβλιογραφία.....	65



## ΕΥΡΕΤΗΡΙΟ ΕΙΚΟΝΩΝ

Εικόνα 1-Τα αποτελέσματα της ακρίβειας εντοπισμού του αισθητήρα GNSS ενός smartphone- Bakula(2014).....	13
Εικόνα 2- 3D νέφος σημείων του αρχαιολογικού χώρου 'Σαράντα Κολώνες' (Κύπρος) το οποίο αποκτήθηκε συνεργατικά χρησιμοποιώντας 3 smartphones. Οι εικόνες έχουν χρωματική κωδικοποίηση σύμφωνα με το smartphone από το οποίο προήλθε - Roiesi et al. (2017) .....	14
Εικόνα 3- Παρεκκλήσι Αγίας Αικατερίνης - Kirchh et al. (2011).....	15
Εικόνα 4- Οχυρό πολυβόλων No 428.....	16
Εικόνα 5- Νέφος σημείων του οχυρού No 428 .....	16
Εικόνα 6- TIN του οχυρού No 428 .....	16
Εικόνα 7- Γεωμετρική Απεικόνιση του οχυρού No 428 .....	17
Εικόνα 8- 3D μοντέλο του οχυρού No 428 .....	17
Εικόνα 9- Οι φωτογραφίες των τριών τοίχων που επιλέχθηκαν.....	18
Εικόνα 10- Τα τελικά νέφη σημείων των τριών τοίχων.....	19
Εικόνα 11- Κεφαλή 2ου πειράματος (με βιδωτούς δείκτες) .....	20
Εικόνα 12- UAV (Μη επανδρωμένο αεροσκάφος) .....	20
Εικόνα 13- Οι κάμερες που χρησιμοποιήθηκαν.....	21
Εικόνα 14- Μέτρηση στάθμης νερού από κινητό, από ράβδο μέτρησης και κατασκευή μέτρησης πίεσης.....	22
Εικόνα 15- Εντοπισμός της παραλίας στον χάρτη και οι γύρω βράχοι .....	22
Εικόνα 16- Λογότυπο ANDROID .....	24
Εικόνα 17- Εκδόσεις ANDROID .....	25
Εικόνα 18- Εικονίδιο 'Google Play Store' .....	26
Εικόνα 19- Μεταγλώττιση και διερμηνεία προγραμμάτων JAVA.....	29
Εικόνα 20- Λογότυπο εργαλείου 'Android Studio'.....	32
Εικόνα 21- Σχηματική απεικόνιση στοιχείων εσωτερικού προσανατολισμού (Έλλη Πέτσα 2000).....	33
Εικόνα 22- Σχέση μεταξύ εικονοσυντεταγμένων και συντεταγμένων αντικειμένου (Kraus, σελ. 14) .....	34
Εικόνα 23- Σχηματική απεικόνιση εμπροσθοτομίας.....	35
Εικόνα 24- Φάκελοι των οντοτήτων που αποτελούν την εφαρμογή .....	36
Εικόνα 25- Κώδικας που αποτελεί το 'AndroidManifest' .....	37
Εικόνα 26- Περιεχόμενο φακέλου 'java'.....	37
Εικόνα 27- Περιεχόμενο φακέλου 'res'.....	38
Εικόνα 28- Κώδικας και εργαλεία στο Android Studio.....	39
Εικόνα 29- Αρχεία '.xml' .....	39
Εικόνα 30- Διαμόρφωση και οντότητες της 'MainActivity' .....	40

Εικόνα 31- Οθόνη 'MainActivity' .....	40
Εικόνα 32- Κώδικας για τη διαμόρφωση του φόντου της οθόνης και του κουμπιού 'Resection' .....	41
Εικόνα 33- 'Focal_length_dialog_layout.xml' .....	41
Εικόνα 34- 'Intersection_results.xml' .....	42
Εικόνα 35- 'Resection_results.xml' .....	42
Εικόνα 36- 'Point_dialog_layout.xml' .....	43
Εικόνα 37- 'Position_orientation_dialog.xml' .....	43
Εικόνα 38- Μέρος του κώδικα της κλάσης 'GroundControlPoint' .....	44
Εικόνα 39- Μέρος του κώδικα της κλάσης 'PixelPoint' .....	45
Εικόνα 40- Μέρος του κώδικα της κλάσης 'PositionAndOrientation' .....	46
Εικόνα 41- Δήλωση μεταβλητών στην κλάση 'SpaceResection' .....	46
Εικόνα 42- Δήλωση σταθερών στην κλάση 'SpaceResection' .....	47
Εικόνα 43- Μέρος του κώδικα της κλάσης 'SpaceResection' .....	47
Εικόνα 44- Μέρος του κώδικα της κλάσης 'SpaceResection' .....	48
Εικόνα 45- Υπολογισμός πίνακα στροφής της κλάσης 'SpaceResection' .....	48
Εικόνα 46- Υπολογισμός πίνακα 'f' της κλάσης 'SpaceResection' .....	49
Εικόνα 47- Υπολογισμός διανυσμάτων 'A1', 'A2', 'Π' της κλάσης 'SpaceResection' ..	49
Εικόνα 48- Υπολογισμός συνιστωσών 'α2', 'α3' της κλάσης 'SpaceResection' .....	50
Εικόνα 49- Υπολογισμός πινάκων 'dFx' και 'dFy' της κλάσης 'SpaceResection' .....	50
Εικόνα 50- Υπολογισμός πινάκων 'A', 'N', 'Fx', 'Fy' και 'L' της κλάσης 'SpaceResection' .....	51
Εικόνα 51- Έλεγχος συνθηκών και προσδιορισμός νέων ή τελικών τιμών της κλάσης 'SpaceResection' .....	51
Εικόνα 52- Υπολογισμός πινάκων στροφής για την αριστερή και δεξιά φωτογραφία της κλάσης 'SpaceIntersection' .....	52
Εικόνα 53- Υπολογισμός προσεγγιστικών τιμών των X,Y,Z του ίδιου σημείου για την αριστερή και δεξιά φωτογραφία της κλάσης 'SpaceIntersection' .....	52
Εικόνα 54- Υπολογισμός πίνακα 'A' της κλάσης 'SpaceIntersection' .....	53
Εικόνα 55- Υπολογισμός πινάκων 'dl', 'N', 'U' της κλάσης 'SpaceIntersection' .....	53
Εικόνα 56- Υπολογισμός πίνακα 'dx', συνθήκης επανάληψης και τελικών τιμών της κλάσης 'SpaceIntersection' .....	53
Εικόνα 57- Μέρος του κώδικα της κλάσης 'UserSession' .....	54
Εικόνα 58- Δηλώσεις της κλάσης 'ResectionActivityPresenter' .....	54
Εικόνα 59- Μέρος του κώδικα της κλάσης 'ResectionActivityPresenter' .....	55
Εικόνα 60- Μέρος του κώδικα της κλάσης 'ResectionActivityPresenter' .....	55
Εικόνα 61- 'Δηλώσεις' της κλάσης 'ResectionActivity' .....	56
Εικόνα 62- Απόσπασμα No1 από τον κώδικα του 'joystick' της κλάσης 'ResectionActivity' .....	57
Εικόνα 63- Απόσπασμα No2 από τον κώδικα του 'joystick' της κλάσης 'ResectionActivity' .....	57
Εικόνα 64- Μετατροπή εικονοσυντεταγμένων ως προς το κεντρικό pixel της κλάσης 'ResectionActivity' .....	58
Εικόνα 65- Μετατροπή εικονοσυντεταγμένων από pixel σε mm της κλάσης 'ResectionActivity' .....	58

Εικόνα 66- Λήψη φωτογραφιών.....	59
Εικόνα 67- Σημεία γνωστών φωτογραμμετρικών συντεταγμένων .....	59
Εικόνα 68- Ένα κομμάτι του σκαριφήματος.....	60
Εικόνα 69- 9 από τα 138 σημεία του αρχείου Word.....	60
Εικόνα 70- Αποτελέσματα 1ου σετ φωτοσταθερών του δεξιού βάθρου.....	61
Εικόνα 71- Αποτελέσματα 2ου σετ φωτοσταθερών του δεξιού βάθρου.....	62
Εικόνα 72- Αποτελέσματα 1ου σετ φωτοσταθερών του αριστερού βάθρου .....	62
Εικόνα 73- Αποτελέσματα 2ου σετ φωτοσταθερών του αριστερού βάθρου .....	63
Εικόνα 74- Αποτελέσματα εμπροσθοτομίας για την εύρεση του σημείου 136.....	63

# ΚΕΦΑΛΑΙΟ 1

## Προσδιορισμός 3D μετρήσεων με έξυπνες συσκευές– Ανασκόπηση βιβλιογραφίας

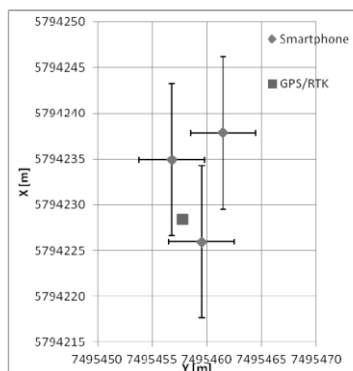
Το πλήθος των λειτουργιών των smartphones, η ολοένα αυξανόμενη δημοτικότητά τους και η ευρεία διαθεσιμότητά τους εγείρει το ερώτημα εάν είναι δυνατόν να χρησιμοποιηθεί ένα smartphone ως φορητό εργαλείο χαμηλού κόστους στη φωτογραμμετρία. Τα smartphones είναι συνήθως εξοπλισμένα με ψηφιακή φωτογραφική μηχανή, δέκτη GNSS, πυξίδα, συνδεσιμότητα στο διαδίκτυο και άλλα στοιχεία. Τα πιο σύγχρονα smartphones διαθέτουν επίσης γυροσκόπιο, επιταχυνσιόμετρο και μαγνητόμετρο. Είναι πολύ σημαντικό το γεγονός ότι μία μόνο συσκευή μπορεί να διαθέτει τόσο μεγάλο εξοπλισμό. Παρά τον ερασιτεχνικό σκοπό της ψηφιακής φωτογραφικής μηχανής και του δέκτη GNSS, με τα οποία είναι εξοπλισμένα τα smartphones, πολλές ερευνητικές ομάδες έχουν πραγματοποιήσει αξιολόγηση της καταλληλότητάς τους όσον αφορά την παροχή φωτογραμμετρικού προϊόντος χαμηλού κόστους και χαμηλής ακρίβειας.

### 1.1 Έλεγχος GNSS και αισθητήρων προσδιορισμού γωνιών

Σύμφωνα με μελέτη των Bakula και Flasiński (2014) για την αξιοπιστία των αποτελεσμάτων του δέκτη GNSS ενός smartphone υπολογίστηκε το μέσο τετραγωνικό σφάλμα της θέσης του αισθητήρα κοντά στα 4.3 μέτρα ενώ η ακρίβεια του υψομέτρου υπολογίστηκε πως είναι αρκετά χαμηλή. Στα πειράματα χρησιμοποιήθηκαν παρατηρήσεις GNSS / RTK ως δεδομένα αναφοράς. Όσον αφορά τις γωνίες  $\omega, \varphi, \kappa$  πραγματοποιήθηκαν δοκιμές σε δύο φάσεις: χωριστά για  $\omega, \kappa$  τα οποία μετρούνται χρησιμοποιώντας ένα σύνολο επιταχυνσιόμετρων και γυροσκοπίου και ξεχωριστά για το  $\varphi$  (αζιμούθιο), το οποίο μετρείται με το μαγνητόμετρο. Στην ανάλυση λαμβάνοντας υπόψη τις γωνίες  $\omega$  και  $\kappa$ , διεξήχθησαν αρκετές δοκιμές που περιλάμβαναν τόσο συνεχείς όσο και ασυνεχείς αλλαγές γωνίας. Σε αντίθεση με αυτήν την ανάλυση, το φάσμα της γωνίας  $\varphi$  εκτιμήθηκε πολύ μακριά από μεταλλικά αντικείμενα λόγω της ευαισθησίας του μαγνητικού αισθητήρα. Το σφάλμα γωνίας  $\varphi$  ήταν πολύ υψηλότερο από το σφάλμα των γωνιών  $\omega$  και  $\kappa$  που εκτιμήθηκε αρκετά χαμηλό. Το μαγνητόμετρο δεν έδειξε αρκετή ακρίβεια σε σύντομες αλλαγές γωνίας που οδηγούν στο συμπέρασμα ότι οι μετρήσεις του ήταν ασταθείς.

Η δυνατότητα χρήσης smartphone για τη δημιουργία τρισδιάστατων μοντέλων με γεωαναφορά μπορεί σίγουρα να αξιολογηθεί ως θετική. Αν και αποκλειστικά η χρήση

δεδομένων απευθείας από τη συσκευή δεν παρέχει ικανοποιητική ακρίβεια, η συμπερίληψη πρόσθετων δεδομένων σχετικά με τον εξωτερικό προσανατολισμό που προέρχονται από ένα smartphone στον υπολογισμό επιτρέπει τη σημαντική μείωση της απαιτούμενης εργασίας στο πεδίο που σχετίζεται με τον αριθμό των επαναλήψεων για τη σύγκλιση του αλγορίθμου. Πρόβλημα αποτελεί κι η μειωμένη δυνατότητα μέτρησης του υψομέτρου καθώς και της γωνίας .



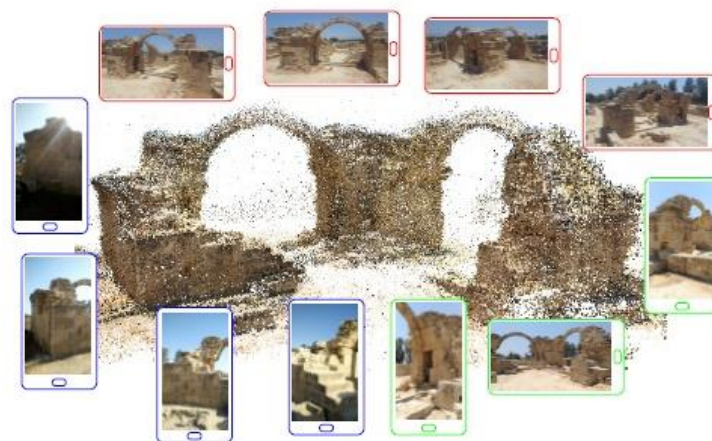
**Εικόνα 1-Τα αποτελέσματα της ακρίβειας εντοπισμού του αισθητήρα GNSS ενός smartphone-Bakula(2014)**

Η ακρίβεια των μεθόδων ανακατασκευής φωτογραμμετρίας μπορεί να αλλάξει ανάλογα με τις συνθήκες εργασίας (π.χ. αριθμός εικόνων, φωτισμός, αποστάσεις μεταξύ των ληφθεισών εικόνων...). Παρά την εξέλιξη της τεχνικής ανακατασκευής των εικόνων, η φωτογραμμετρία δε φτάνει στο ίδιο επίπεδο αξιοπιστίας των τεχνικών σάρωσης με χρήση laser scanner. Μεγάλη διαφορά όμως στις δύο τεχνικές είναι το κόστος των εργαλείων. Σε έρευνα που έγινε από την Fissore (2018) και την ομάδα της μελετήθηκε η πιθανή βελτίωση που μπορεί να επιτευχθεί στην ακρίβεια ανακατασκευής από φωτογραφίες smartphone εισάγοντας πληροφορίες από το σύστημα πλοήγησής του ή η λογική του συνδυασμού διαφορετικών αισθητήρων που είναι ενσωματωμένοι στο smartphone (επιταχυνσιόμετρο, γυροσκόπιο, μαγνητόμετρο) στην περίπτωση που το σήμα GNSS δεν είναι διαθέσιμο. Δεδομένου του αριθμού των φυσικών καταστροφών που σημειώθηκαν παγκοσμίως τις τελευταίες δεκαετίες, το ενδιαφέρον για ένα γρήγορο σύστημα απόκτησης χωρικών δεδομένων, εύκολο στη χρήση και διαθέσιμο σε χαμηλά επίπεδα κόστους αυξάνεται συνεχώς. Η διαθεσιμότητα τέτοιων δεδομένων μπορεί να είναι ενδιαφέρουσα για να βοηθήσει τη διαχείριση τέτοιων δύσκολων συνθηκών. Η συνδυασμένη χρήση αυτών των εκτιμήσεων επιτρέπει την ενημέρωση της θέσης της συσκευής σε σχέση με την προηγούμενη τιμή της.

Το σύστημα παρακολούθησης έχει σχεδιαστεί για να εκτιμά τις αλλαγές στάσης της συσκευής . Οι μετρήσεις βαρόμετρου χρησιμοποιούνται στην εκτίμηση υψομέτρου. Στην εργασία αυτή της Fissore (2018) έγιναν πειράματα από 3 εθελοντές σε ένα τριώροφο κτίριο , του οποίου τα χαρακτηριστικά σε κάθε σημείο ήταν γνωστά. Κάθε εθελοντής κινήθηκε στο κτίριο για 10 λεπτά. Αφού έγινε σύγκριση των τιμών υψομέτρων που έδειχνε το smartphone με αυτές που ήδη ήταν υπολογισμένες με βάση τα σημεία του κτιρίου , προέκυψε ότι η αβεβαιότητα ήταν μεγαλύτερη όταν ο κάθε εθελοντής βρισκόταν στις σκάλες, ότι το μέσο σφάλμα ύψους ήταν 0.2 m και ότι το σφάλμα μειώθηκε 20% σε σχέση με τα αποτελέσματα που προκύπτουν από το σύστημα πλοήγησής χωρίς τη διόρθωση από βαρόμετρο. Επομένως, αποδείχτηκε ότι η ενσωμάτωση των μετρήσεων βαρομέτρου στο σύστημα πλοήγησής είναι ένας βιώσιμος τρόπος για να παρέχεται εκτίμηση υψομέτρου και να μειώνονται τα σφάλματα σε εσωτερικούς χώρους.

## 1.2 Τρισδιάστατη απεικόνιση

Με τη χρήση φωτογραμμετρικών μεθόδων έχουν δημιουργηθεί αρκετές εφαρμογές είτε για smartphone είτε για Η/Υ οι οποίες αποσκοπούν στην αποτύπωση αντικειμένων. Πιο συγκεκριμένα, παρατηρώντας μια δημοσίευση από φοιτητές του Πανεπιστημίου του Τρέντο και της Ζυρίχης (Poiesi et al.- 2017) μπορεί να κατανοηθεί η λογική με την οποία σχεδίασαν και δημιούργησαν μια εφαρμογή τρισδιάστατης ανακατασκευής σε 'έξυπνο' κινητό τηλέφωνο. Αυτό το application λειτουργεί με βάση έναν κεντρικό server ο οποίος δέχεται τα δεδομένα των χρηστών, τα επεξεργάζεται και παράγει το τελικό αποτέλεσμα δηλαδή την τρισδιάστατη αναπαράσταση. Αυτός ο server αντλεί τα δεδομένα που χρειάζεται για να λειτουργήσει, από τα χαρακτηριστικά που παρουσιάζει ο κάθε 'κατασκευαστής' στα smartphones (εσωτερικός προσανατολισμός) και σε μορφή φωτογραφιών ή βίντεο. Πολλαπλοί χρήστες ταυτόχρονα μπορούν να ανεβάζουν τα βίντεο ή τις φωτογραφίες τους στην πλατφόρμα έτσι ώστε να δημιουργηθεί ένα νέφος σημείων που απεικονίζει όσο το δυνατόν καλύτερα την πραγματικότητα. Η εφαρμογή αυτή είναι προγραμματισμένη ώστε να δίνει και οδηγίες για το ποια δεδομένα λείπουν ώστε να ολοκληρωθεί η τρισδιάστατη αναπαράσταση. Όλα αυτά με σκοπό ένα 'οπτικό αποτέλεσμα', γεγονός που 'επισημαίνει' τη διαφορετικότητα αυτής της εφαρμογής που θα παρουσιάσουμε παρακάτω με τις ήδη υπάρχουσες .



**Εικόνα 2- 3D νέφος σημείων του αρχαιολογικού χώρου 'Σαράντα Κολώνες' (Κύπρος) το οποίο αποκτήθηκε συνεργατικά χρησιμοποιώντας 3 smartphones. Οι εικόνες έχουν χρωματική κωδικοποίηση σύμφωνα με το smartphone από το οποίο προήλθε - Poiesi et al. (2017)**

Η πολιτιστική κληρονομιά διαδραματίζει ζωτικό ρόλο στην εκπαίδευση για το παρελθόν, στη δημιουργία πολιτιστικής ή ατομικής ταυτότητας, ακόμη και στην παροχή οικονομικής υποστήριξης για τις τοπικές κοινότητες. Η πολιτιστική κληρονομιά βρίσκεται υπό διαρκή απειλή φθοράς ή ακόμη και καταστροφής. Απαιτούνται οικονομικά αποδοτικές και εύχρηστες μέθοδοι για την καταγραφή της πολιτιστικής κληρονομιάς. Η φωτογραμμετρία μικρής εμβέλειας (close range) έχει αποδειχθεί δυναμικό εργαλείο σε αυτόν τον τομέα. Εργασία των Kirchhoferl et al. (2011) από το πανεπιστήμιο του Loughborough, περιγράφει την ανάπτυξη ενός

συστήματος εγγραφής με βάση την εικόνα για καταγραφή πολιτιστικής κληρονομιάς. Η πρόθεση είναι να εκτιμηθεί η εφικτή ακρίβεια και πρακτικότητα του συστήματος. Προκειμένου να επιτευχθούν ακριβείς παράμετροι εξωτερικού προσανατολισμού της κάμερας και των αισθητήρων έπρεπε να γίνει βαθμονόμηση. Τα δεδομένα καταγράφηκαν από ποικίλο αριθμό σταθμών κάμερας. Οι τιμές βαθμονόμησης από τα δεδομένα που συλλέχθηκαν εφαρμόστηκαν στις μετρήσεις του ίδιου συνόλου δεδομένων. Οι μετρήσεις χρησιμοποιήθηκαν για την παροχή αρχικών παραμέτρων εξωτερικού προσανατολισμού. Το σύστημα αρχικά δοκιμάστηκε στο Πανεπιστήμιο Loughborough, όπου το αντικείμενο δοκιμής ήταν μια κάθετη κατασκευή με μικρή διάμετρο στο έδαφος με πρόσβαση από όλες τις πλευρές. Τα δεδομένα που συλλέχθηκαν υποβλήθηκαν σε επεξεργασία και αναλύθηκαν. Ένα παρεκκλήσι της Αγίας Αικατερίνης στο Isle of Wight του Ηνωμένου Βασιλείου επελέγη ως τόπος δοκιμής μελέτης περιπτώσεων.



Εικόνα 3- Παρεκκλήσι Αγίας Αικατερίνης - Kirchh et al. (2011)

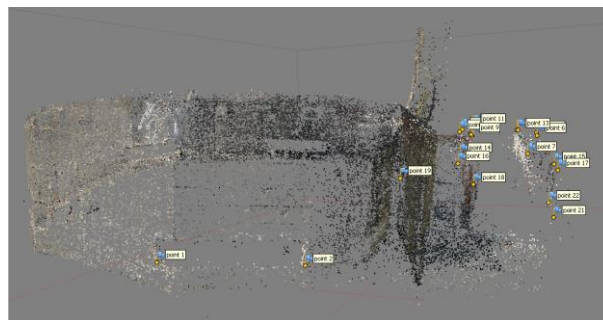
Κατά τη διάρκεια της μελέτης συλλέχθηκαν δύο σύνολα δεδομένων. Το πλαίσιο στήριξης του κινητού είχε κλίση έως  $21^\circ$  για να καλύψει όλο το ύψος του πύργου. Το σύνολο δεδομένων αποτελείται από δεδομένα που συλλέχθηκαν από 12 σταθμούς κάμερας που ήταν διατεταγμένοι σε τόξο γύρω από την ανατολική πλευρά του πύργου με κατά προσέγγιση απόσταση κάμερας-αντικειμένου τα 10 μέτρα. Η απόλυτη ακρίβεια ποσοτικοποιεί την ικανότητα των συστημάτων καταγραφής να παρέχουν δεδομένα για μετρήσεις που είναι ακριβείς σε σχέση με ένα εθνικό σύστημα αναφοράς συντεταγμένων. Η σχετική ή εσωτερική ακρίβεια ποσοτικοποιεί την ικανότητα του συστήματος εγγραφής να παρέχει δεδομένα για μετρήσεις που είναι ακριβείς μεταξύ τους. Το smartphone πέτυχε απόλυτη ακρίβεια 1,15m και σχετική ακρίβεια 0,68m. Όπως υπέδειξαν οι ερευνητές, smartphone με πιο ακριβείς αισθητήρες προσανατολισμού και θέσης, μπορούν να χρησιμοποιηθούν για αποτύπωση αντικειμένων πολιτιστικής κληρονομιάς χαμηλού κόστους από μη ειδικούς.

Σε άλλη εργασία από το Εθνικό πανεπιστήμιο κατασκευών κι αρχιτεκτονικής του Κιέβου (Shults-2017) εξετάστηκαν κάποια ζητήματα σχετικά με τη χαμηλού κόστους φωτογραμμετρία με βάση τις νέες δυνατότητες των smartphones. Η έρευνα είχε ως στόχο να γίνει γεωμετρική τεκμηρίωση μιας ιστορικής κατασκευής που προέκυψε από το Β' Παγκόσμιο Πόλεμο. Για την επίτευξη του ζητούμενου αποτελέσματος χρησιμοποιήθηκαν εικόνες από smartphone, εφαρμογές μέτρησης του smartphone, laser LeicaDisto (για να εκτελεστούν μετρήσεις ελέγχου), laptop (για να βρεθούν τα στοιχεία της κάμερας και να δημιουργηθεί το 3D μοντέλο), το smartphone MEIZU M3 Max και το πρόγραμμα Agisoft Photoscan για την 3D μοντελοποίηση. Η διαδικασία που ακολουθήθηκε ξεκίνησε με το καλιμπράρισμα της κάμερας με χρήση υπολογιστή

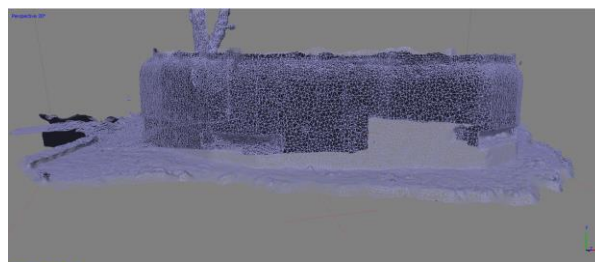
ώστε να προκύψουν τα στοιχεία του εσωτερικού προσανατολισμού. Τα αποτελέσματα όπως ήταν λογικό έδειξαν πως υπάρχει μεγάλη διαστροφή αλλά η ποιότητα της βαθμονόμησης ήταν καλή κι οι παράμετροι εν τέλει προσδιορίστηκαν με μεγάλη ακρίβεια. Το μετρητικό μοντέλο που δημιουργήθηκε έλαβε πληροφορίες από την εφαρμογή Android Smart Tools (μετρήσεις αποστάσεων, γωνιών, υψών, κλίσεων) και από μετρήσεις συντεταγμένων από γεωδαιτικό δέκτη GNSS. Με βάση το παραγόμενο αποτέλεσμα αποδείχθηκε ακόμη μια φορά πως η χαμηλού κόστους φωτογραμμετρία μπορεί να είναι αρκετά αποτελεσματική για την τεκμηρίωση της πολιτιστικής κληρονομιάς.



**Εικόνα 4- Οχυρό πολυβόλων No 428**

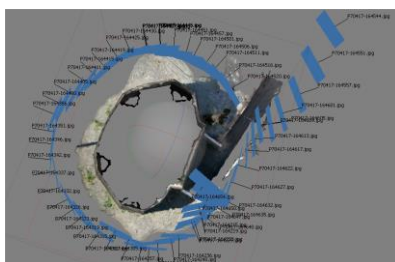


**Εικόνα 5- Νέφος σημείων του οχυρού No 428**



**Εικόνα 6- TIN του οχυρού No 428**





**Εικόνα 7- Γεωμετρική Απεικόνιση του οχυρού Νο 428**



**Εικόνα 8- 3D μοντέλο του οχυρού Νο 428**

Τα smartphone όμως δοκιμάστηκαν και σε σπήλαια. Πιο συγκεκριμένα, σε έρευνα των Dabove et al. (2019) από το Πολυτεχνείο του Τορίνο αναλύθηκε η απόδοση μιας συσκευής smartphone για μια προσέγγιση φωτογραμμετρίας μικρής εμβέλειας για την εξαγωγή ακριβών τρισδιάστατων πληροφοριών ενός υπόγειου σπηλαίου. Η προσέγγιση έχει δείξει μια καλή απόδοση όσον αφορά στη γεωμετρική ακρίβεια, τον υπολογιστικό χρόνο και την εφαρμοσιμότητα. Τα αποτελέσματα συγκρίθηκαν με μια επαγγελματική κάμερα και με έναν επίγειο σαρωτή λέιζερ. Η γεωμορφολογική και γεωλογική παρατήρηση σε υπόγεια περιβάλλοντα, όπως σπήλαια, μπορεί να αποτελεί πρόκληση. Η συνεχής εξέλιξη των τρισδιάστατων τεχνικών έρευνας και μοντελοποίησης αλλά και η βελτίωση των ηλεκτρονικών εξοπλισμών παρακολούθησης επιτρέπει σε κάποιον να αποκτήσει μια πλήρη περιγραφή του περιβάλλοντος. Οι σαρωτές λέιζερ είναι η πιο δημοφιλής λύση. Οι ενεργό αισθητήρες χρησιμοποιούνται πλέον ευρέως σε υπόγεια σπήλαια για την εξαγωγή πληροφοριών 3D, καθώς επιτρέπουν την άμεση λήψη της θέσης των σημείων που ανιχνεύονται, μερικές φορές σε συνδυασμό με τις πληροφορίες χρώματος. Ένα από τα σημαντικότερα πλεονεκτήματα της χρήσης ενός smartphone είναι η δυνατότητα άμεσης εκτέλεσης της επεξεργασίας δεδομένων με συγκεκριμένες εφαρμογές εγκατεστημένες στο κινητό. Στις πρώτες προσπάθειες αξιοποίησης της φωτογραμμετρικής τεχνικής μικρής εμβέλειας χρησιμοποιήθηκαν εικόνες smartphone, με σκοπό την ανακατασκευή του αντικειμένου στο συντομότερο δυνατό χρόνο. Ένα επιπλέον πλεονέκτημα προέρχεται από την εύκολη διαθεσιμότητα αυτών των εργαλείων, τα οποία τα καθιστούν κατάλληλα για χρήση σε καταστάσεις έκτακτης ανάγκης. Οι ερευνητές διερεύνησαν τις δυνατότητες απόκτησης τοπογραφικών δεδομένων υψηλής ανάλυσης και δεδομένων εδάφους χρησιμοποιώντας smartphone. Αυτές οι μελέτες υπογράμμισαν ότι μια προσέγγιση βασισμένη σε εικόνα που εκμεταλλεύεται συσκευές μαζικής αγοράς επιτρέπει την ανακατασκευή

σωστών γεωμετρικά μοντέλων με ακρίβεια εκατοστού. Η τεχνική μπορεί εύκολα να χρησιμοποιηθεί από μη ειδικούς χειριστές, με βάση τις αρχές της φωτογραμμετρίας. Πριν από την εφαρμογή αυτής της τεχνικής σε υπόγεια σπήλαια, το smartphone δοκιμάστηκε σε εργαστηριακό περιβάλλον. Η προτεινόμενη μεθοδολογία έχει σχεδιαστεί για να επιτρέπει την επίτευξη της διαδικασίας με εύκολα μεταφερόμενα όργανα (εμπορικές κάμερες, tablet και smartphone). Χρησιμοποιήθηκαν τεχνικές πυκνής συνταύτισης εικόνων για τη δημιουργία ενός πυκνού νέφους τρισδιάστατων σημείων, για να χρησιμοποιηθούν ως υποστήριξη γεωλογικών και γεωμορφολογικών ερευνών. Ελήφθησαν υπόψη μόνο τα δεδομένα από εικόνες, χωρίς καμία ενσωμάτωση στο μοντέλο άλλων ειδών δεδομένων. Η έρευνα αποσκοπούσε στην τρισδιάστατη ανακατασκευή ενός τμήματος του σπηλαιού της Βοσέας της Ιταλίας. Το σπήλαιο της Βοσέας είναι από τα πιο όμορφα και γεωλογικά ενδιαφέροντα σπήλαια στην Ευρώπη. Οι μετρήσεις του δείγματος προκάλεσαν ιδιαίτερο ενδιαφέρον για τη μελέτη της κυκλοφορίας του νερού σε ανθρακικά πετρώματα. Η δοκιμή πραγματοποιήθηκε σε βέλτιστες συνθήκες φωτισμού και περιβάλλοντος. Επιλέχθηκαν τρεις τοίχοι μέσα στο σπήλαιο. Οι τρεις επιλεγμένες επιφάνειες διακρίθηκαν από τις συνθήκες φωτισμού. Στη συνέχεια, το έργο είχε ως στόχο να τονίσει τη μεθοδολογία σε ένα πραγματικό υπόγειο σπήλαιο. Οι τρεις επιλεγμένες επιφάνειες χαρακτηρίζονταν από την τυπική ανωμαλία αυτών των περιβαλλόντων. Οι περισσότερες επιφάνειες υγραίνονται από ένα λεπτό στρώμα νερού, το οποίο αυξάνεται σημαντικά στις πιο υγρές περιόδους. Γενικά, πριν από τη χρήση εικόνων για ανακατασκευή 3D, είναι σημαντικό να αναλύονται τα εγγενή χαρακτηριστικά της ψηφιακής κάμερας. Σε αυτήν τη μελέτη έγινε βαθμονόμηση πεδίου ελέγχου. Η έρευνα πραγματοποιήθηκε χρησιμοποιώντας τη συσκευή smartphone OnePlus 5 στη λειτουργία βίντεο. Σε γενικές γραμμές, η γεωμετρία απόκτησης πρέπει να είναι κατασκευασμένη κατά τρόπο ώστε μεταξύ μιας εικόνας και της επόμενης να επικαλύπτεται τόσο κατά τη διαμήκη όσο και από την εγκάρσια διεύθυνση. Η συσκευή smartphone κρατήθηκε από τον χειριστή, ο οποίος κινήθηκε γύρω από το αντικείμενο, προσπαθώντας να έχει μια πλήρη περιγραφή. Η επικύρωση των προϊόντων που προέκυψαν πραγματοποιήθηκε μέσω της σύγκρισης κάθε σημείου του νέφους με το δημιουργημένο νέφος σημείου αναφοράς. Το τελικό μοντέλο αποκτήθηκε με σφάλμα περίπου 1 cm. Τα λαμβανόμενα μοντέλα έδειξαν ότι είναι δυνατόν να δημιουργηθούν αναπαραστάσεις σε κλίμακες αρκετά μεγάλες για τις κύριες γεωμορφολογικές αναλύσεις.



Εικόνα 9- Οι φωτογραφίες των τριών τοίχων που επιλέχθηκαν



Εικόνα 10- Τα τελικά νέφη σημείων των τριών τοίχων

### 1.3 Χειρουργική μέσω πλοήγησης

Ένα φορητό εργαλείο χαμηλού κόστους όπως το smartphone μπορεί να φανεί χρήσιμο και για την Ιατρική. Πιο συγκεκριμένα, στη χειρουργική μέσω πλοήγησης το σώμα του ασθενούς πρέπει να απεικονιστεί σε 3D δεδομένα τα οποία έχουν αποκτηθεί προχειρουργικά ώστε να καταστεί δυνατή η πλοήγηση του χειρουργικού οργάνου. Η επιφάνεια του σώματος μπορεί να προκύψει μέσω της φωτογραμμετρίας. Στην εργασία των Hellwich et al. (2016) επιλύεται με παραδείγματα η αντιστοιχία ανάμεσα σε 3D δεδομένα ανθρώπινων κεφαλών και των επιφανειών προσώπων με αποδεκτό σφάλμα μικρότερο του 1mm. Το εκάστοτε χειρουργικό όργανο είναι συμβατό με ηλεκτρομαγνητικό μηχάνημα το οποίο υπολογίζει τον εντοπισμό. Αυτό το μηχάνημα δεν παρέχει θέση όσον αφορά τη θέση της επιφάνειας του ασθενούς. Είναι πάρα πολύ σημαντικό να γίνει ταύτιση των συστημάτων συντεταγμένων των οργάνων και των προσώπων. Η φωτογραμμετρική ανακατασκευή επιτυγχάνεται με βάση αρκετές φωτογραφίες που έχουν ληφθεί από smartphone. Για τη συνταύτιση χρησιμοποιήθηκαν κοινά σημεία και για τα δύο συστήματα συντεταγμένων ώστε να μπορεί να καταστεί δυνατή η μετατροπή από το φωτογραμμετρικό σύστημα στο ηλεκτρομαγνητικό. Οι κάμερες που χρησιμοποιήθηκαν ήταν βαθμονομημένες για τις δεδομένες αποστάσεις. Για την εύρεση 7 παραμέτρων χρησιμοποιήθηκαν 8 δείκτες ώστε να υπάρχει πλεόνασμα. Οι 7 από τους 8 δείκτες ήταν μέσα στο επίπεδο πλαίσιο ενώ ο 8<sup>ος</sup> περίπου 5 εκατοστά πάνω. Οι δείκτες είχαν σχεδιαστεί να έχουν μηδενικό σφάλμα. Στη βασική διαδικασία για την ανακατασκευή προσώπων χρησιμοποιήθηκαν 3 εικόνες οι οποίες υπέστησαν επεξεργασία ως 2 ζευγάρια, των οποίων τα νέφη σημείων συνδυάστηκαν για τη τελική ανακατασκευή. Η απόσταση από το αντικείμενο ορίστηκε να είναι περίπου 50 cm και τα μήκη βάσης φωτογράφισης 4-10 cm. Για μια ρεαλιστική εκτίμηση έγιναν 2 πειράματα. Για το 1<sup>ο</sup> πείραμα τοποθετήθηκαν κεφάλια με ρεαλιστική υφή τα οποία είχαν πάνω τους σημειωμένα σημεία (δείκτες- σημεία ελέγχου). Προκειμένου να προσδιοριστούν οι συντεταγμένες των σημείων ελέγχου χρησιμοποιήθηκαν lego τουβλάκια(γνωστών διαστάσεων) . Για το 2<sup>ο</sup> πείραμα χρησιμοποιήθηκαν βιδωτοί δείκτες πάνω στην επιφάνεια του προσώπου. Οι αποκλίσεις που προέκυψαν από τα πειράματα ήταν εντός των απαραίτητων προδιαγραφών (  $\sigma_x = \pm 0.64$  mm ,  $\sigma_y = \pm 0.38$  mm ,  $\sigma_z = \pm 0.87$ mm). Τελικά , αποδείχτηκε ότι τριάδες εικόνων από I-Pad/smartphone των οποίων έχει επιλυθεί η γεωμετρία του φακού τους, επιτρέπουν την ανακατασκευή των επιφανειών σωμάτων με καλή ακρίβεια.



Εικόνα 11- Κεφαλή 2ου πειράματος (με βιδωτούς δείκτες)

#### 1.4 Συστήματα φωτογραμμετρίας UAV με χρήση smartphone και έξυπνες κάμερες

Οι Jeong et al. (2019) αξιολογούν συστήματα φωτογραμμετρίας UAV χρησιμοποιώντας smartphone και έξυπνες κάμερες.



Εικόνα 12- UAV (Μη επανδρωμένο αεροσκάφος)

Οι έξυπνες κάμερες είναι επαγγελματικές φωτογραφικές μηχανές προηγμένης τεχνολογίας που δίνουν τη δυνατότητα στον χρήστη να έχει ταχύτατη συνδεσιμότητα, προκειμένου να μπορεί να μοιράζεται ταχύτατα εικόνες υψηλής ποιότητας. Για την αξιολόγηση δημιουργήθηκαν ακριβείς εικόνες ορθοφωτομωσαϊκών και ψηφιακά μοντέλα επιφάνειας. Στην τρέχουσα κατάστασή τους, οι πλατφόρμες μη επανδρωμένων εναέριων οχημάτων (UAV) μπορούν να χρησιμοποιούν smartphone ως αισθητήρα και να καταγράφουν εικόνες με γεωγραφικό εντοπισμό. Τα smartphone μπορούν να λειτουργούν σε περιβάλλον δικτύου 3G και 4G ανά πάσα στιγμή ή τοποθεσία. Ένα ψηφιακό μοντέλο εδάφους μπορεί επίσης να δημιουργηθεί για σχετικά μικρές περιοχές χρησιμοποιώντας δορυφορικό σύστημα (GNSS). Ωστόσο, οι λεπτομέρειες που περιλαμβάνουν πλευρές κτιρίων, αιχμηρές γωνίες και άκρες ενδέχεται να μην αποδίδονται με ακρίβεια και τα αποτελέσματα της ανακατασκευής 3D ενδέχεται να μην κρίνονται με ακρίβεια. Η συγκεκριμένη εργασία επικεντρώθηκε στις δυνατότητες του smartphone Samsung Galaxy S7 και στην απόδοση της κάμερας του. Οι ερευνητές επικεντρώθηκαν στην επαλήθευση ποιότητας για τις ορθοφωτογραφίες που προέρχονται από την κάμερα και τα ψηφιακά μοντέλα επιφάνειας. Θεωρητικά, μπορούν να επιτευχθούν ακριβή αποτελέσματα τοποθετώντας φωτοσταθερά στην άκρη και στο εσωτερικό της περιοχής μελέτης. Δυστυχώς, σε αυτή τη μελέτη ένας μεγάλος αριθμός εγκατεστημένων φωτοσταθερών χάθηκε λόγω κατασκευαστικών εργασιών. Χρησιμοποιήθηκαν 2 είδη καμερών. Η κάμερα Samsung Galaxy S7 είναι εξοπλισμένη με αυτόματη εστίαση, η οποία είναι ένα κοινό χαρακτηριστικό στις κάμερες DSLR. Η κάμερα Galaxy NX διαθέτει φακό 16 mm και αυτόματη διόρθωση παραμόρφωσης φακού έξι στοιχείων κι επτά ρυθμίσεων. Συλλέχθηκαν δύο σύνολα



δεδομένων εγκαθιστώντας την αντιολισθητική βάση στο κέντρο του οχήματος και τοποθετώντας καθεμία από τις δύο κάμερες με τη σειρά . Λήφθηκαν περίπου 80 εικόνες με ρυθμίσεις αλληλεπικάλυψης κατά μήκος 80% και κατά πλάτος 60%. Ο καιρός ήταν καλός κατά τη διάρκεια των πτήσεων UAV, με ορατότητα περίπου 16 χιλιόμετρα . Το σφάλμα εξαρτήθηκε από την ποιότητα της βαθμονόμησης της κάμερας καθώς και από την ποιότητα των εικόνων. Οι κάμερες NX απέδωσαν πολύ καλά, με ακρίβεια εντός 1 ρixel σε σύγκριση με την άλλη κάμερα. Οι τιμές RMS της κάμερας S7 ήταν 2–3 ρixel. Αυτό πιθανότατα οφείλεται σε περιορισμούς στην απόδοση των σχετικά φθηνών φακών που χρησιμοποιούνται στο smartphone.. Η κατακόρυφη ακρίβεια των μοντέλων ψηφιακού εδάφους που προέρχονται από φωτογραμμετρία ήταν τουλάχιστον τόσο καλή όσο αυτή του αερομεταφερόμενου lidar. Όλες οι επιφάνειες είχαν εξαιρετική κατακόρυφη ακρίβεια. Καταληκτικά, σε αυτήν τη μελέτη, μετά από σύγκριση της NX με το S7 στο ορθοφωτομωσαϊκό και το ΨΜΕ, αποδείχτηκε ότι η ακρίβεια ήταν υψηλότερη για την κάμερα NX.



a) Samsung Galaxy S7



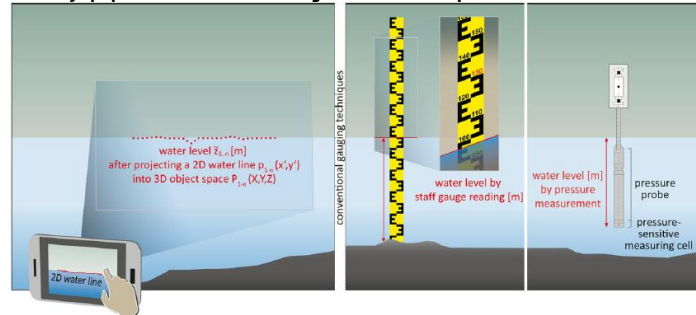
b) Samsung Galaxy NX

**Εικόνα 13- Οι κάμερες που χρησιμοποιήθηκαν**

## 1.5 Μέτρηση στάθμης νερού από κάμερα smartphone

Οι Elias et al. (2019) σε έρευνα που έκαναν εξέτασαν μια ευέλικτη μέθοδο μέτρησης της στάθμης του νερού από κάμερα smartphone. Την προηγούμενη δεκαετία υπήρξαν πολλές βροχοπτώσεις και πλημμύρες. Μέχρι και σήμερα, οι συμβατικοί σταθμοί μέτρησης παρέχουν πληροφορίες για τα επίπεδα νερού ανά σύντομα χρονικά διαστήματα και τοποθετούνται αραιά. Επομένως, χείμαρροι, ποτάμια, ρυάκια δεν ελέγχονται κατά τη διάρκεια πλημμυρών. Ως λύση σε αυτές τις ελλείψεις μπορεί να είναι η μέτρηση της στάθμης του νερού από κάμερα. Στο πείραμα που έγινε, τοποθετήθηκε μια ψηφιακή κάμερα σε μια πλευρά ενός τεχνητού ποταμού, με φορά στην αντίθετη όχθη. Σκοπός ήταν να παρατηρηθεί το επίπεδο του νερού. Στην παρατηρούμενη όχθη του ποταμού υπήρχαν σημεία με γνωστές συντεταγμένες με στόχο να προσδιοριστεί ο εξωτερικός προσανατολισμός της κάμερας. Αυτό το πείραμα έγινε το 2012 αλλά και το 2018. Κι οι δύο προσεγγίσεις ήταν περιορισμένες σε ήσυχα νερά κι οι κάμερες ήταν σταθερές και προβαθμονομημένες. Η διαφορά των δύο πειραμάτων ήταν ότι το 2018 χρησιμοποιήθηκε χαμηλού κόστους κάμερα από smartphone . Όπως αποδείχτηκε και στις δύο περιπτώσεις επετεύχθη ακρίβεια κοντά στο 1cm. Οι ακολουθίες εικόνων σε μικρό χρονικό διάστημα ενός τμήματος του ποταμού χρησιμοποιούνται για να διακριθεί μια στατική περιοχή ακτής από μια δυναμική περιοχή ποταμού η οποία παρέχει πληροφορίες για τη στάθμη του νερού . Προαπαιτούμενο για να λειτουργήσει όλο το σύστημα σωστά είναι τα δεδομένα 3D να είναι μετρημένα σε κλίμακα μέτρου και να έχουν μεγάλη ακρίβεια. Τα δεδομένα 3D χρησιμοποιούνται για να προσδιορισθεί ο εξωτερικός προσανατολισμός της εικόνας του smartphone και να τοποθετηθεί σε κλίμακα η 2D γραμμή του νερού. Είναι σημαντικό να έχουν ληφθεί εικόνες σε διαφορετική χρονική στιγμή αλλά να καλύπτουν την ίδια τοποθεσία ώστε να ληφθούν υπόψη οι χωροχρονικές αλλαγές σε ότι αφορά τη ραδιομετρία ή τις διαφορές φωτισμού. Επίσης τα δεδομένα 3D πρέπει να συμπεριλαμβάνουν RGB ή τιμές έντασης ώστε να μπορεί να είναι εφικτή η γεωμετρική εγγραφή της εικόνας. Τα περισσότερα smartphones διαθέτουν δέκτη GNSS και χαμηλού κόστους συστήματα αισθητήρων. Η τεχνική μπορεί να είναι ελκυστική για τις εθνικές αρχές επιτήρησης που μπορούν να λάβουν υπόψη τις προαναφερθείσες απαιτήσεις στη δημιουργία

κατάλληλων τρισδιάστατων μοντέλων πόλης. Περιορισμός επιβάλλεται και στην πυκνότητα των δεδομένων 3D στα οποία καταγράφονται τα δεδομένα εικόνας. Ένα αραιό τρισδιάστατο νέφος σημείου οδηγεί σε λιγότερα σημεία που μπορούν να χρησιμοποιηθούν για την εξαγωγή της στάθμης του νερού. Αυτό σημαίνει ότι η προσέγγιση είναι πιο ευαίσθητη σε ακραίες τιμές. Ο αλγόριθμος πρέπει να λειτουργεί σε διάφορες συσκευές χαμηλού κόστους. Τα κοινά smartphone πρέπει να χρησιμοποιούνται για τη δημιουργία δεδομένων σε περιπτώσεις ακραίων συμβάντων και χωρίς την ανάγκη επαγγελματικής εμπειρογνωμοσύνης. Οι μελλοντικές εξελίξεις στο υλικό και το λογισμικό των 'έξυπνων' κινητών σημαίνουν ότι είναι πιθανό οι εργασίες επεξεργασίας 'βαρέων βαρών' που έχουν ανατεθεί σήμερα να μπορούν να υποβληθούν σε επεξεργασία απευθείας στο smartphone.



**Εικόνα 14- Μέτρηση στάθμης νερού από κινητό, από ράβδο μέτρησης και κατασκευή μέτρησης πίεσης**

Εκτός από τα ποτάμια, σύμφωνα με έρευνα του πανεπιστημίου του Μπρεστ (Jaud et al. - 2019), με smartphone μπορούν να παρατηρηθούν και παράκτιες περιοχές. Η καλύτερη κατανόηση των μηχανισμών που οδηγούν στην αλλαγή των ακτών είναι απαραίτητη για τη βελτιωμένη διαχείριση των παράκτιων περιοχών. Μπορεί να πραγματοποιηθεί παρακολούθηση πολλαπλών πηγών, συμπεριλαμβανομένων διαφορετικών τεχνικών, από μεθόδους ευρείας κάλυψης, όπως δορυφορικές εικόνες έως μετρήσεις σημείων με έρευνες GPS. Τις τελευταίες δεκαετίες, διάφορες τεχνικές τηλεπισκόπησης υψηλής ανάλυσης έχουν αποδείξει τις δυνατότητές τους για παρακολούθηση των ακτών. Η παράκτια παρακολούθηση βίντεο παρέχει παρατηρήσεις υψηλής συχνότητας, συνεχείς και αυτόνομες. Αυτή η μελέτη προτείνει τη χρήση επίγειας φωτογραμμετρίας με φωτογραφίες smartphone για παραθαλάσσια παρακολούθηση. Ο τόπος μελέτης ήταν η παραλία Porstmilin, που βρίσκεται στην είσοδο του κόλπου του Μπρεστ στη Βρετάνη της Γαλλίας.



**Εικόνα 15- Εντοπισμός της παραλίας στον χάρτη και οι γύρω βράχοι**

Τρία μοντέλα smartphone δοκιμάστηκαν για να εκτιμηθεί εάν η ποιότητα της τοπογραφικής ανακατασκευής ποικίλλει από το ένα μοντέλο στο άλλο. Τα μοντέλα που χρησιμοποιήθηκαν σε αυτήν τη μελέτη είναι τα Samsung® Galaxy S7, το Nokia® Lumia 930 και το Apple® iPhone 8. Μια Panasonic Lumix FZ1000 δοκιμάζεται επίσης για να συγκρίνει τις ανακατασκευές από smartphone με τα αποτελέσματα που

λαμβάνονται με μια πιο «κλασική» κάμερα για επίγεια φωτογραμμετρία. Οι εικόνες δεν ήταν προεπιλεγμένες, αλλά ορισμένες από αυτές απορρίφθηκαν αυτόματα κατά τη διάρκεια της επεξεργασίας. Κατά τη διάρκεια της έρευνας, ο χειριστής φρόντισε να διατηρεί μια επικάλυψη μεταξύ των φωτογραφιών (περίπου 60 έως 80%). Χρησιμοποιήθηκαν 58 φωτογραφίες για το Galaxy S7, 55 για το Lumia 930, 38 για το iPhone 8 και 25 για τη φωτογραφική μηχανή Panasonic Fz1000. Όλα τα σύνολα δεδομένων αναφέρθηκαν γεωγραφικά στο RGF93-Lambert93 το οποίο είναι το επίσημο σύστημα συντεταγμένων στη Γαλλία. Για αυτήν τη δοκιμή 10 φωτοσταθερά κατανεμήθηκαν σε όλη την περιοχή της μελέτης. Στην παραλία Porssmilin, παρά τους πολλούς συνδυασμούς θέσεων λήψης, μόνο πέντε θέσεις στο δυτικό πρόσωπο επέτρεψαν τη λήψη ενός πυκνού νέφους σημείων. Ένα DEM δημιουργήθηκε από αυτό το σημείο νέφους με χωρική ανάλυση 7,9 cm. Διαπιστώθηκε ότι η ποιότητα ανακατασκευής δεν εξαρτάται από το μοντέλο smartphone που χρησιμοποιείται. Η φωτογραμμετρία από Smartphone έχει τη δυνατότητα να παρέχει μορφολογικές έρευνες με την ίδια ποιότητα (σε ανάλυση και ακρίβεια) με τις φωτογραμμετρικές έρευνες που χρησιμοποιούν "κλασικές" κάμερες. Τα διάφορα σφάλματα μπορεί να ποικίλουν χρονικά και χωρικά, ανάλογα με τη γωνία πρόσπτωσης του λέιζερ ή την τραχύτητα της επιφάνειας. Καθώς αυξάνεται η απόσταση, μειώνεται η ανάλυση της εικόνας, η οποία μπορεί να επηρεάσει την αναγνώριση σημείου πρόσδεσης. Για τη συμπλήρωση αυτής της μελέτης, η έρευνα για την παραλία δοκιμάστηκε σε άλλη παραλία. Συλλέχθηκαν τριάντα πέντε φωτογραφίες χρησιμοποιώντας το Lumia 930. Το αποτέλεσμα ήταν σίγουρα θετικό καθώς οι ερευνητές κατάφεραν να αναπαραγάγουν τοπογραφία πρώτης τάξης, αλλά και περίπλοκα μορφολογικά χαρακτηριστικά, όπως ακρογιαλιές παραλίας και βράχους στην άνω πλευρά της παραλίας. Η απόκτηση ικανοποιητικών αποτελεσμάτων ανεξάρτητα από το μοντέλο smartphone που χρησιμοποιείται και κατά την ανάμειξη φωτογραφιών από διαφορετικά smartphone είναι ιδιαίτερα ελπιδοφόρο εάν στοχεύουμε να εφαρμόσουμε αυτές τις μεθόδους σε παρατηρητήρια πολιτών. Τα Παρατηρητήρια Πολιτών μπορούν να οριστούν ως συστήματα περιβαλλοντικής παρακολούθησης και πληροφόρησης που βασίζονται στην κοινότητα, γενικά βασισμένα σε συσκευές των πολιτών. Οι φωτογραφίες που συλλέχθηκαν από διαφορετικά άτομα θα μπορούσαν να χρησιμοποιηθούν στην ίδια διαδικασία ανασυγκρότησης.

## ΚΕΦΑΛΑΙΟ 2

### Προγραμματισμός συσκευών ANDROID

#### 2.1 Τι είναι το Android ;

Το **Android** είναι λειτουργικό σύστημα για συσκευές κινητής τηλεφωνίας. Αρχικά αναπτύχθηκε από την Google και αργότερα από την Open Handset Alliance. Επιτρέπει στους κατασκευαστές λογισμικού να συνθέτουν κώδικα με την χρήση της γλώσσας προγραμματισμού JAVA, ελέγχοντας την συσκευή μέσω

βιβλιοθηκών λογισμικού ανεπτυγμένων από την Google. Το Android είναι κατά κύριο λόγο σχεδιασμένο για συσκευές με οθόνη αφής, όπως τα έξυπνα τηλέφωνα και τα τάμπλετ, με διαφορετικό περιβάλλον χρήσης για τηλεοράσεις (Android TV) αυτοκίνητα (Android Auto) και ρολόγια χειρός (Android Wear). Παρόλο που έχει αναπτυχθεί για συσκευές με οθόνη αφής, έχει χρησιμοποιηθεί σε κονσόλες παιχνιδιών, ψηφιακές φωτογραφικές μηχανές, συνηθισμένους Η/Υ και σε άλλες ηλεκτρονικές συσκευές. Το Android είναι το πιο ευρέως διαδεδομένο λογισμικό στον κόσμο. Η πρώτη παρουσίαση της πλατφόρμας Android έγινε στις 5 Νοεμβρίου 2007, παράλληλα με την ανακοίνωση της ίδρυσης του οργανισμού Open Handset Alliance, μιας κοινοπραξίας 48 τηλεπικοινωνιακών εταιριών, εταιριών λογισμικού καθώς και κατασκευής hardware, οι οποίες είναι αφιερωμένες στην ανάπτυξη και εξέλιξη ανοιχτών προτύπων στις συσκευές κινητής τηλεφωνίας. Η Google δημοσίευσε το μεγαλύτερο μέρος του κώδικα του Android υπό τους όρους της Apache License, μιας ελεύθερης άδειας λογισμικού. Από τον Απρίλιο του 2009, οι εκδόσεις του Android έχουν ονομασίες από την ζαχαροπλαστική στην κωδική ονομασία τους, και κυκλοφόρησαν σε αλφαβητική σειρά, εξαιρουμένων των εκδόσεων 1.0 και 1.1, που δεν τέθηκαν υπό συγκεκριμένα κωδικοποιημένα ονόματα. Η έκδοση που χρησιμοποιείται κατά την υλοποίηση αυτής της Διπλωματικής Εργασίας καλείται Android 11 και φέρνει σημαντικές αλλαγές σε σχέση με την προηγούμενη και φυσικά με την αρχική .



**Εικόνα 16- Λογότυπο ANDROID**



Κωδικό όνομα	Νούμερο έκδοσης	Ημερομηνία αρχικής κυκλοφορίας	Επίπεδο API
Alpha	1.0	23 Σεπτεμβρίου 2008	1
Beta	1.1	9 Φεβρουαρίου 2009	2
Cupcake	1.5	27 Απριλίου 2009	3
Donut	1.6	15 Σεπτεμβρίου 2009	4
Eclair	2.0 – 2.1	26 Οκτωβρίου 2009	5 – 7
Froyo	2.2 – 2.2.3	20 Μαΐου 2010	8
Gingerbread	2.3 – 2.3.7	6 Δεκεμβρίου 2010	9 – 10
Honeycomb	3.0 – 3.2.6	22 Φεβρουαρίου 2011	11 – 13
Ice Cream Sandwich	4.0 – 4.0.4	18 Οκτωβρίου 2011	14 – 15
Jelly Bean	4.1 – 4.3.1	9 Ιουλίου 2012	16 – 18
KitKat	4.4 – 4.4.4	31 Οκτωβρίου 2013	19 – 20
Lollipop	5.0 – 5.1.1	12 Νοεμβρίου 2014	21 – 22
Marshmallow	6.0 – 6.0.1	5 Οκτωβρίου 2015	23
Nougat	7.0 – 7.1.2	22 Αυγούστου 2016	24 – 25
Oreo	8.0 – 8.1	21 Αυγούστου 2017	26 – 27
Pie	9.0	6 Αυγούστου 2018	28
Q	10.0	3 Σεπτεμβρίου 2019	29
R	11.0	19 Φεβρουαρίου 2020	30

**Εικόνα 17- Εκδόσεις ANDROID**

### 2.1.1 Τα λειτουργικά συστήματα κι οι εφαρμογές

Οι 'έξυπνες' συσκευές λειτουργούν συνήθως χρησιμοποιώντας ένα από τα δύο δημοφιλέστερα λειτουργικά συστήματα, το Android (Google) και το iOS (Apple). Το Android χρησιμοποιείται από την ίδια τη Google και από τις SAMSUNG, HTC, MOTOROLA, LG, Xiaomi, Lenovo και πολλές άλλες. Η Apple χρησιμοποιεί αποκλειστικά η ίδια το iOS και η Huawei στα νέα της μοντέλα θα χρησιμοποιεί ένα νέο λειτουργικό σύστημα με όνομα HarmonyOS. Τα λειτουργικά συστήματα έχουν τη δυνατότητα να καλύψουν τις ανάγκες οποιουδήποτε χρήστη καθώς προσφέρουν μεγάλη ποικιλία εφαρμογών. Δημοφιλέστερες από αυτές είναι οι τηλεφωνικές εφαρμογές, οι εφαρμογές μηνυμάτων, ηλεκτρονικού ταχυδρομείου, φωτογραφιών, αριθμομηχανής, πλοήγησης, μέσων κοινωνικής δικτύωσης κ.τ.λ.

### 2.1.2 Google Play Store

Από άλλη οπτική όμως τα λειτουργικά συστήματα δίνουν την επιλογή στους εκάστοτε προγραμματιστές να δημιουργήσουν τη δική τους εφαρμογή που εξυπηρετεί τους ίδιους αλλά και πολλούς άλλους. Οι εφαρμογές που χρειάζεται και επιθυμεί να χρησιμοποιήσει ο κάθε χρήστης είναι διαθέσιμες στα ηλεκτρονικά καταστήματα εφαρμογών και μπορούν να τις αποκτήσουν είτε πληρώνοντας κάποιο χρηματικό ποσό είτε δωρεάν. Το ηλεκτρονικό κατάστημα της Google και συνεπώς του Android ονομάζεται 'Google Play Store' και διαθέτει τουλάχιστον 3.500.000

εφαρμογές διαθέσιμες για τους χρήστες. Ενδεικτικά, υπολογίζεται ότι ο συνολικός αριθμός λήψεων ξεπερνάει τα 70 δισεκατομμύρια. Γίνεται αντιληπτό λοιπόν, αυτός ο τομέας απασχολεί εκατομμύρια προγραμματιστές και τα smartphones έχουν γίνει αναπόσπαστο μέρος της καθημερινότητας της σύγχρονης κοινωνίας. Λαμβάνοντας όλα τα παραπάνω υπόψη δεν επιδέχεται αμφισβήτηση ότι αναλογικά με τους αυξανόμενους ρυθμούς ζωής αυξάνεται και η ζήτηση πιο αποδοτικών συσκευών αλλά και εφαρμογών οι οποίες θα ικανοποιούν όσο το δυνατόν περισσότερες ανάγκες των χρηστών όσο περίπλοκες κι αν είναι.



Εικόνα 18- Εικονίδιο 'Google Play Store'

## 2.2 Βασικά στοιχεία προγραμματισμού

Υπάρχουν δύο κύριες πτυχές κατά την ενασχόληση με τον προγραμματισμό: τα δεδομένα και οι εντολές. Προκειμένου να εργαστεί κανείς με δεδομένα σε οποιαδήποτε γλώσσα προγραμματισμού, πρέπει πρώτα να κατανοήσει την έννοια των μεταβλητών και τους τύπους. Σε ό,τι αφορά στις εντολές, είναι απαραίτητη η κατανόηση των δομών ελέγχου και των υπορουτίνων.

### 2.2.1 Μεταβλητές και τύποι δεδομένων

Μια μεταβλητή είναι απλά μια περιοχή της μνήμης του Η/Υ, στην οποία έχει αποδοθεί ένα όνομα, προκειμένου να διευκολύνεται η αναφορά σε αυτή και η χρήση της από ένα πρόγραμμα. Ο προγραμματιστής ασχολείται μονάχα με την απόδοση ονόματος, καθώς ο μεταγλωττιστής αναλαμβάνει τα υπόλοιπα. Είναι σημαντικό για τον προγραμματιστή να θυμάται, ότι το όνομα αυτό αναφέρεται σε ένα είδος «κουτιού» στη μνήμη, που μπορεί να φυλάσσει δεδομένα, ακόμα κι αν ο προγραμματιστής δε χρειάζεται να γνωρίζει πού ακριβώς βρίσκεται αυτό το κουτί. Στην Java, αλλά και σε άλλες γλώσσες προγραμματισμού, μια μεταβλητή έχει έναν τύπο, ο οποίος υποδηλώνει το είδος των δεδομένων που η μεταβλητή μπορεί να φυλάσσει. Χαρακτηριστικοί τύποι μεταβλητών, που συναντώνται στις περισσότερες γλώσσες προγραμματισμού, είναι αυτοί του ακέραιου αριθμού (π.χ. 10, 25, 400,

κτλ.), του δεκαδικού αριθμού (π.χ. 0.256, -2.05, 3.14, κτλ.), του χαρακτήρα (π.χ. „Α“, „β“, „2“, „#“, κτλ.), της συμβολοσειράς (π.χ. «Java», «μια συμβολοσειρά», κτλ.), της λογικής τιμής (true/false), κ.ά.

### 2.2.2 Δομές ελέγχου

Ένα πρόγραμμα είναι ουσιαστικά μια ακολουθία εντολών. Στην απλούστερη περίπτωση, ο υπολογιστής εκτελεί τις εντολές της ακολουθίας με τη σειρά που αυτές εμφανίζονται, τη μια μετά την άλλη. Κάτι τέτοιο, ωστόσο, είναι πολύ περιοριστικό και δεν επαρκεί για να καλύψει τις ανάγκες σύνθετων προγραμμάτων. Οι δομές ελέγχου αποτελούν μια ειδική μορφή εντολών, που μπορούν να αλλάξουν τη ροή ελέγχου σε ένα πρόγραμμα, και χωρίζονται σε δύο βασικές κατηγορίες: τις επαναλήψεις (loops), οι οποίες επιτρέπουν σε ένα σύνολο εντολών να εκτελείται κατ' επανάληψη, και τις διακλαδώσεις (branches), οι οποίες επιτρέπουν στον υπολογιστή να αποφασίζει την εκτέλεση ενός συνόλου εντολών μεταξύ πολλαπλών διαθέσιμων συνόλων-επιλογών, αξιολογώντας συνθήκες που προκύπτουν καθώς το πρόγραμμα εκτελείται.

### 2.2.3 Υπορουτίνες

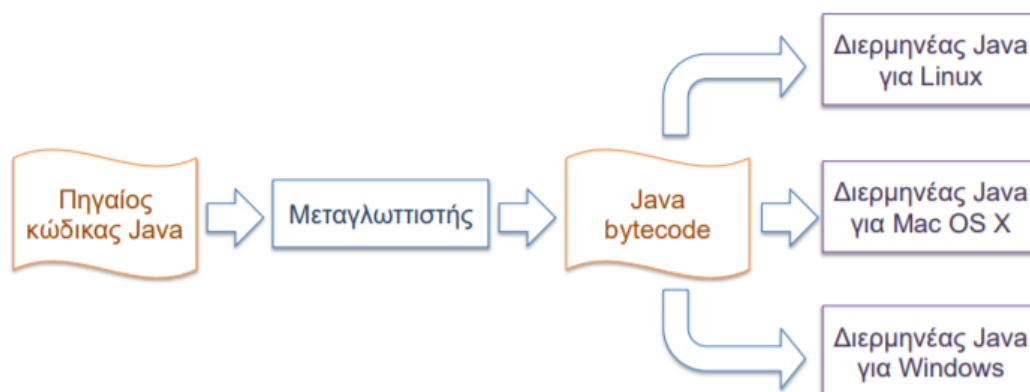
Τα προγράμματα λογισμικού που έχουν σχετικά μεγάλη έκταση είναι συνήθως τόσο πολύπλοκα, που θα ήταν αδύνατο να γραφτούν αν δεν υπήρχε κάποιος τρόπος να «σπάσουν» σε μικρότερα και πιο διαχειρίσιμα κομμάτια. Οι υπορουτίνες προσφέρουν ακριβώς έναν τρόπο για να επιτευχθεί κάτι τέτοιο. Μια υπορουτίνα αποτελείται από ένα σύνολο εντολών που απαιτούνται για την εκτέλεση μιας εργασίας, ομαδοποιημένες ως μια ενιαία μονάδα στην οποία επιπλέον αποδίδεται ένα όνομα. Έτσι, το όνομα της υπορουτίνας μπορεί πλέον να χρησιμοποιείται ως υποκατάστατο του συνόλου εντολών που η υπορουτίνα περιλαμβάνει.

## 2.3 Περιβάλλον εκτέλεσης κώδικα JAVA

Οι ηλεκτρονικοί υπολογιστές είναι σχεδιασμένοι να εκτελούν εντολές γραμμένες σε μια απλή μορφή, που είναι γνωστή ως γλώσσα μηχανής. Η γλώσσα αυτή αποτελείται από πολύ απλές εντολές, που μπορούν να εκτελεστούν απευθείας στην Κεντρική Μονάδα Επεξεργασίας (CPU) του υπολογιστή. Ωστόσο, σχεδόν όλα τα προγράμματα σήμερα δε γράφονται απευθείας σε γλώσσα μηχανής. Αντιθέτως, οι προγραμματιστές χρησιμοποιούν γλώσσες προγραμματισμού υψηλού επιπέδου, όπως είναι η Java, η C++, κ.ά., και επομένως ο παραγόμενος πηγαίος κώδικας δεν μπορεί να «τρέξει» απευθείας σε κανέναν υπολογιστή. Κάθε πρόγραμμα, συνεπώς, πρέπει πρώτα να μεταφραστεί σε γλώσσα μηχανής προκειμένου να καταστεί εκτελέσιμο. Η διαδικασία αυτή πραγματοποιείται από λογισμικό ειδικού σκοπού, το οποίο ονομάζεται μεταγλωττιστής (compiler). Αφού ολοκληρωθεί η διαδικασία της μεταγλώττισης, ο παραγόμενος κώδικας γλώσσας μηχανής μπορεί να εκτελεστεί οσσεδήποτε φορές, αλλά μόνο σε ένα συγκεκριμένο τύπο υπολογιστή (μιας και κάθε τύπος υπολογιστή υποστηρίζει μόνο τη δική του, ειδική γλώσσα μηχανής). Έτσι, αν χρειάζεται ένα πρόγραμμα να καταστεί εκτελέσιμο σε διαφορετικούς τύπους

υπολογιστή, πρέπει να επαναληφθεί η μεταγλώττιση χρησιμοποιώντας κάθε φορά τον κατάλληλο και συμβατό με τον εκάστοτε υπολογιστή μεταγλωττιστή. Μια εναλλακτική τεχνική μετάφρασης του υψηλού επιπέδου κώδικα σε γλώσσα μηχανής βασίζεται στη χρήση ενός ειδικού λογισμικού που ονομάζεται διερμηνέας (interpreter). Η βασική διαφορά του διερμηνέα από τον μεταγλωττιστή έγκειται στο ότι, ενώ ο μεταγλωττιστής πραγματοποιεί τη μετάφραση του κώδικα εξ ολοκλήρου, ο διερμηνέας μεταφράζει τον κώδικα εντολή-προς-εντολή, όποτε αυτό είναι απαραίτητο. Υποστηρίζοντας έναν τέτοιο κύκλο εκτέλεσης, ο διερμηνέας προσομοιάζει αρκετά τη λειτουργία της CPU. Προκειμένου να εκτελέσει ένα πρόγραμμα, ο διερμηνέας επαναληπτικά και διαδοχικά διαβάσει την επόμενη εντολή του προγράμματος, αποφασίζει ποιες εντολές γλώσσας μηχανής απαιτούνται για την εκτέλεσή της και στη συνέχεια τις εκτελεί καταλλήλως. Παρόλα αυτά, οι διερμηνείς έχουν έναν ακόμα σκοπό: επιτρέπουν την εκτέλεση ενός προγράμματος σε διαφορετικούς τύπους υπολογιστή, ανεξάρτητα από τη γλώσσα μηχανής στην οποία το πρόγραμμα έχει μεταφραστεί. Το χαρακτηριστικό αυτό βρίσκει εφαρμογή στην ανάπτυξη προγραμμάτων, που είναι γνωστά με την ονομασία εικονικός υπολογιστής (virtual computer). Πρέπει ωστόσο να αναφερθεί ότι ένας εικονικός υπολογιστής χρειάζεται να εκτελέσει αρκετές εντολές γλώσσας μηχανής προκειμένου να προσομοιώσει την κάθε μια εντολή γλώσσας μηχανής του προγράμματος. Έτσι, τα μεταγλωττισμένα προγράμματα είναι εγγενώς πιο γρήγορα από αυτά που εκτελούνται με τη χρήση διερμηνέα. Οι σχεδιαστές της γλώσσας Java επέλεξαν να χρησιμοποιήσουν ένα συνδυασμό των τεχνικών μεταγλώττισης και διερμηνείας που προαναφέρθηκαν. Συγκεκριμένα, τα προγράμματα Java μεταγλωττίζονται σε μια γλώσσα μηχανής, αλλά αυτή η γλώσσα μηχανής προορίζεται για έναν εικονικό υπολογιστή, γνωστό ως Εικονική Μηχανή Java (Java Virtual Machine), ή συντομογραφικά JVM. Με τη σειρά της, η γλώσσα μηχανής που προορίζεται για την JVM ονομάζεται χαρακτηριστικά Java bytecode. Στην πραγματικότητα, η γλώσσα αυτή θα μπορούσε να χρησιμοποιηθεί ως γλώσσα μηχανής και για κάποιον τύπο πραγματικού υπολογιστή. Ωστόσο, η επιλογή της JVM καθιστά εφικτό ένα από τα πιο σημαντικά, ανταγωνιστικά πλεονεκτήματα της Java: τα προγράμματά της μπορούν να εκτελεστούν σε οποιονδήποτε τύπο υπολογιστή. Το μόνο που απαιτείται είναι ο υπολογιστής να διαθέτει έναν διερμηνέα για τον κώδικα Java bytecode. Φυσικά, για κάθε τύπο υπολογιστή απαιτείται και ένας διαφορετικός διερμηνέας του Java bytecode. Από τη στιγμή όμως που ένας υπολογιστής διαθέτει το διερμηνέα αυτό, μπορεί να εκτελεί οποιοδήποτε πρόγραμμα Java. Αντίστροφα, το ίδιο πρόγραμμα Java μπορεί να εκτελείται σε οποιονδήποτε υπολογιστή διαθέτει ένα διερμηνέα Java bytecode. Η τεχνική αυτή συνδυάζει τα πλεονεκτήματα της ταχύτητας του μεταγλωττισμένου κώδικα και της ευελιξίας της διερμηνείας, ενώ μπορεί να βελτιώσει

τη γενικότερη ασφάλεια του περιβάλλοντος εκτέλεσης του κώδικα.



Εικόνα 19- Μεταγλώττιση και διερμηνεία προγραμμάτων JAVA

### 2.3.1 Περιβάλλον και ανάπτυξη εφαρμογής

Για τη γλώσσα Java, οι διαδικασίες δημιουργίας, μεταγλώττισης και επεξεργασίας προγραμμάτων Java διαφέρουν αρκετά, από το ένα προγραμματιστικό περιβάλλον στο άλλο. Σε γενικές γραμμές, υπάρχουν δύο προσεγγίσεις: το περιβάλλον γραμμής εντολών (command-line environment), όπου ο χρήστης πληκτρολογεί εντολές και ο υπολογιστής αποκρίνεται κατάλληλα, και το ενσωματωμένο περιβάλλον ανάπτυξης (integrated development environments, ή IDE), όπου ο χρήστης αλληλοεπιδρά με ένα γραφικό περιβάλλον προγραμματισμού χρησιμοποιώντας τόσο το πληκτρολόγιο όσο και το ποντίκι του υπολογιστή. Κι ενώ υπάρχει ένα μόνο κοινό περιβάλλον γραμμής εντολών για τον προγραμματισμό σε Java, τα τελευταία χρόνια έχουν εμφανιστεί αρκετά εναλλακτικά περιβάλλοντα IDE. Το Java Development Kit (JDK) το βασικό σύστημα ανάπτυξης για τη γλώσσα Java συχνά αναφέρεται ως JDK, ακρωνύμιο που προκύπτει από τα αρχικά των λέξεων Java Development Kit, και αποτελεί μέρος του πακέτου Java SE (Standard Edition). Επίσης, σημειώνεται ότι το πακέτο Java SE προσφέρεται σε δύο εκδόσεις, αυτήν του JDK και αυτή του JRE (Java Runtime Environment). Όπως και το JDK, η διανομή JRE μπορεί να χρησιμοποιηθεί για την εκτέλεση προγραμμάτων Java, ωστόσο, σε αντίθεση με το JDK, δεν υποστηρίζει τη μεταγλώττιση πηγαίου κώδικα Java σε Java bytecode. Η Java αναπτύχθηκε αρχικά από την εταιρεία Sun Microsystems Inc., αλλά εδώ και μερικά χρόνια ανήκει πλέον στην εταιρεία Oracle. Η Oracle προσφέρει δωρεάν τις διανομές της Java SE, καθώς και άλλες διανομές της γλώσσας, από την ιστοσελίδα της:

<https://www.oracle.com/java/technologies/>

Υπάρχουν πολλά διαφορετικά περιβάλλοντα IDE για προγραμματισμό με Java, από απλά περιβάλλοντα του JDK, ως εξαιρετικά πολύπλοκες εφαρμογές με πληθώρα χαρακτηριστικών. Καθώς η εκμάθηση της γλώσσας Java είναι από μόνη της αρκετά απαιτητική, συνήθως βοηθάει κατά τα πρώιμα στάδια εξοικείωσης με τη γλώσσα η χρήση κάποιου σχετικά απλού περιβάλλοντος IDE, ακόμα κι αν αυτό διαθέτει απλώς τα απολύτως απαραίτητα χαρακτηριστικά και λειτουργίες. Παρόλα αυτά,

περιβάλλοντα IDE όπως το Eclipse ή το NetBeans είναι εξαιρετικά δημοφιλή, προσφέρονται δωρεάν, ενώ είναι αρκετά εύκολα στην εκμάθησή τους.

## 2.4 Αντικειμενοστραφής Προγραμματισμός

Ο Αντικειμενοστραφής Προγραμματισμός (ΑΠ) αποβλέπει στη δημιουργία προγραμμάτων, που μοντελοποιούν πιο πιστά τον τρόπο με τον οποίο οι άνθρωποι αντιλαμβάνονται και αλληλοεπιδρούν με τον πραγματικό κόσμο. Σε παλιότερα συλ προγραμματισμού, ο προγραμματιστής, δεδομένου ενός προβλήματος, καλείται να αναγνωρίσει την υπολογιστική εργασία που απαιτείται για την επίλυσή του. Κατά τον τρόπο αυτό, ο προγραμματισμός αφορά στην εύρεση της ακολουθίας εντολών που διεκπεραιώνουν τη συγκεκριμένη εργασία. Αντίθετα, στην «καρδιά» του ΑΠ βρίσκονται τα αντικείμενα – οντότητες που διαθέτουν συμπεριφορά, διατηρούν πληροφορία και αλληλοεπιδρούν μεταξύ τους. Έτσι, στον ΑΠ, το πρόβλημα προσεγγίζεται καταρχάς με τη σχεδίαση του συνόλου των αντικειμένων που το μοντελοποιούν. Τα αντικείμενα αυτά μπορεί να αναπαριστούν τόσο πραγματικές οντότητες, όσο και αφηρημένες έννοιες του πεδίου εφαρμογής του προβλήματος. Κατά αυτόν τον τρόπο, ο σχεδιασμός του προγράμματος καθίσταται πιο φυσικός και ως εκ τούτου διευκολύνει την ορθή κατανόηση του προβλήματος. Υπό μία έννοια, ο ΑΠ αποτελεί απλά μια διαφορετική γωνία θέασης του ίδιου προβλήματος. Θα μπορούσε να περιγραφεί ένα αντικείμενο με έννοιες του παραδοσιακού προγραμματισμού, ως ένα σύνολο μεταβλητών και υπορουτίνων για το χειρισμό τους. Ουσιαστικά, μπορούν να χρησιμοποιηθούν αντικειμενοστραφείς τεχνικές σε οποιαδήποτε γλώσσα προγραμματισμού. Ωστόσο, υπάρχει μια σημαντική διαφορά μεταξύ μιας γλώσσας που απλώς καθιστά τον ΑΠ δυνατό και μιας γλώσσας που τον υποστηρίζει εγγενώς. Μια αμιγώς αντικειμενοστραφής γλώσσα προγραμματισμού, όπως η Java, περιλαμβάνει μια σειρά χαρακτηριστικών που τη διαφοροποιούν σημαντικά από άλλες, μη αντικειμενοστραφείς γλώσσες. Έτσι, προκειμένου ο προγραμματιστής να χρησιμοποιήσει αποτελεσματικά αυτά τα χαρακτηριστικά, είναι απαραίτητο να προσαρμόσει κατάλληλα τον τρόπο σκέψης του.

### 2.4.1 Αντικείμενα, κλάσεις, στιγμιότυπα

Τα αντικείμενα είναι στενά συνδεδεμένα με τις κλάσεις. Συγκεκριμένα, μια κλάση (class) περιγράφει τα κοινά χαρακτηριστικά και λειτουργίες μιας κατηγορίας αντικειμένων και χρησιμοποιείται για τη δημιουργία αντικειμένων αυτής της κατηγορίας. Με άλλα λόγια, μια κλάση μπορεί να θεωρηθεί ως ένα είδος εργοστασίου (factory) για την κατασκευή αντικειμένων. Τα αντικείμενα δημιουργούνται και καταστρέφονται καθώς το πρόγραμμα εκτελείται, ενώ πολλά αντικείμενα με την ίδια δομή μπορούν να συνυπάρχουν, αν δημιουργούνται με τη χρήση της ίδιας κλάσης. Ένα αντικείμενο που ανήκει σε μια κλάση αποτελεί στιγμιότυπο (instance) της κλάσης αυτής. Παρόλο που ο αντικειμενοστραφής προγραμματισμός εστιάζει στη σχεδίαση και υλοποίηση νέων κλάσεων, είναι σημαντικό να θυμόμαστε ότι η Java παρέχει ήδη ένα μεγάλο αριθμό επαναχρησιμοποιήσιμων κλάσεων. Μια κλάση αντιπροσωπεύει ένα σύνολο αντικειμένων που μοιράζονται την ίδια δομή και συμπεριφορά. Το βασικό νέο στοιχείο του αντικειμενοστραφούς προγραμματισμού, η

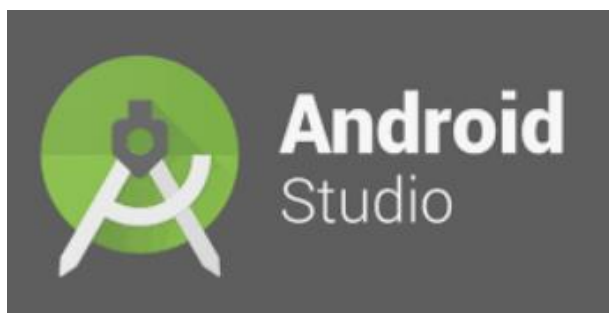
ειδοποιός διαφορά του από άλλα προγραμματιστικά μοντέλα, είναι ότι επιτρέπει στις κλάσεις να εκφράζουν τις ομοιότητες ανάμεσα σε αντικείμενα που μοιράζονται μέρος της δομής και συμπεριφοράς τους. Αυτές οι ομοιότητες εκφράζονται με τη χρήση της κληρονομικότητας (inheritance) και του πολυμορφισμού (polymorphism). Ο όρος κληρονομικότητα αναφέρεται στο γεγονός, ότι μια κλάση μπορεί να κληρονομή μέρος ή ολόκληρη τη δομή και τη συμπεριφορά κάποιας άλλης κλάσης. Με τον πολυμορφισμό, ο κώδικας ενός προγράμματος αποκτά τη δυνατότητα να πραγματοποιεί εργασίες που ο προγραμματιστής δεν είχε ενδεχομένως σκεφτεί κατά την αρχική φάση συγγραφής του. Οι μέθοδοι αποτελούν τον κώδικα μίας κλάσης και είναι μέλη αυτής. Γενικότερα, μια μέθοδος είναι πολυμορφική, όταν η εργασία που εκτελεί εξαρτάται από τον τύπο του αντικειμένου, του οποίου η μέθοδος αυτή καλείται. Μια καλά σχεδιασμένη κλάση μπορεί να επαναχρησιμοποιηθεί σε διαφορετικά προγράμματα, χωρίς την ανάγκη περαιτέρω επεξεργασίας του κώδικά της. Έτσι, όταν σχεδιαστεί μια κλάση για τις ανάγκες κάποιου προγράμματος, είναι σημαντικό αυτή να μοντελοποιεί κάποιο συγκεκριμένο τύπο αντικειμένου ή κάποια έννοια, και όχι κάποια πολύ συγκεκριμένη εργασία που χρειάζεται για το πρόγραμμα αυτό.

#### **2.4.2 Συλλογή απορριμμάτων**

Ένα ακόμα πλεονέκτημα της Java είναι ότι χρησιμοποιεί μια διαδικασία που ονομάζεται συλλογή απορριμμάτων (garbage collection), για την ανάκτηση της μνήμης που καταλαμβάνεται από αντικείμενα χωρίς αναφορές, που δεν είναι πλέον προσπελάσιμα από το πρόγραμμα. Ο έλεγχος της μνήμης και η αναγνώριση τέτοιων αντικειμένων πραγματοποιείται αυτόματα από το σύστημα, χωρίς να απαιτείται ειδική μέριμνα από τον προγραμματιστή. Αν ένα αντικείμενο χρησιμοποιηθεί από το πρόγραμμα μόνο για λίγο, ενδέχεται να υπάρχουν αρκετές αναφορές προς αυτό, αποθηκευμένες σε διαφορετικές μεταβλητές. Αν οι τιμές των μεταβλητών αυτών δεν αλλάξουν, το σύστημα θεωρεί ότι το συγκεκριμένο αντικείμενο εξακολουθεί να χρησιμοποιείται από το πρόγραμμα και επομένως δεν θα προσπαθήσει να το καταστρέψει στο πλαίσιο της διαδικασίας συλλογής απορριμμάτων. Σε πολλές γλώσσες προγραμματισμού, η συλλογή απορριμμάτων αποτελεί υποχρέωση του προγραμματιστή. Δυστυχώς, η διαχείριση της χρήσης μνήμης είναι δύσκολη με αποτέλεσμα συχνά να προκύπτουν σφάλματα, τα οποία με τη σειρά τους δημιουργούν προγραμματιστικά σφάλματα (bugs). Ο προγραμματιστής μπορεί κατά λάθος να καταστρέψει ένα αντικείμενο, ενώ αυτό ακόμα χρησιμοποιείται από το πρόγραμμα, ή, αντίθετα, να παραμελήσει την καταστροφή αντικειμένων που δε χρησιμοποιούνται πλέον από το πρόγραμμα.

### **2.5 Android Studio**

Για την ανάπτυξη εφαρμογών σε συσκευές Android, ο προγραμματιστής πρέπει να διαθέτει υπολογιστή στον οποίο να μπορεί να εγκατασταθεί το Android Studio. Το Android Studio είναι το επίσημο ολοκληρωμένο περιβάλλον ανάπτυξης για το λειτουργικό σύστημα Android, το οποίο παρέχει όλα τα εργαλεία που χρειάζονται για τη δημιουργία, διαχείριση, σχεδίαση και υλοποίηση εφαρμογών. Οι γλώσσες προγραμματισμού που χρησιμοποιεί αυτό το περιβάλλον είναι Java, Kotlin και C++.



Εικόνα 20- Λογότυπο εργαλείου 'Android Studio'

### 2.5.1 Τα Android components και οι 'Activities'

Τα συστατικά μέρη μιας εφαρμογής Android είναι τα βασικά δομικά της στοιχεία. Κάθε στοιχείο είναι ένα σημείο εισόδου, μέσω του οποίου το σύστημα ή κάποιος χρήστης μπορεί να μπει στην εφαρμογή. Υπάρχουν τέσσερις βασικοί τύποι τέτοιων στοιχείων :

- Activities
- Services
- Broadcast receivers
- Content providers

Το βασικό δομικό στοιχείο που έχει χρησιμοποιηθεί σε αυτή την εφαρμογή είναι οι 'Activities'. Μία δραστηριότητα (Activity) είναι το σημείο εισόδου αλληλεπίδρασης της εφαρμογής με το χρήστη.

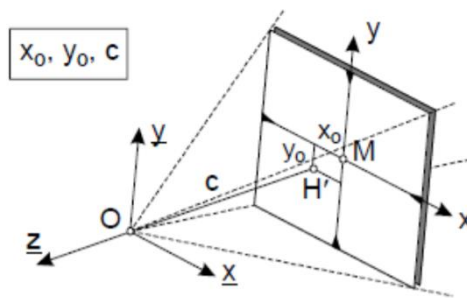
## ΚΕΦΑΛΑΙΟ 3



## Φωτογραμμετρία και Συνθήκη Συγγραμμικότητας

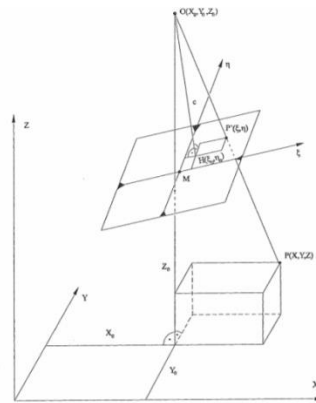
**Φωτογραμμετρία** ονομάζεται μια ιδιαίτερη μέθοδος - τεχνική προσδιορισμού θέσης, μορφής και διαστάσεων αντικειμένων με χρήση φωτογραφιών. Στη μέθοδο αυτή ακολουθείται ιδιαίτερος τρόπος φωτογράφισης. Με την εξέλιξη της ψηφιακής φωτογραφίας, η **ψηφιακή φωτογραμμετρία** είναι αυτή που κυριαρχεί σήμερα στο χώρο των επίγειων, αλλά και εναέριων αποτυπώσεων. Τα παρακάτω στοιχεία αποτελούν τα στοιχεία του εσωτερικού προσανατολισμού της φωτογραφικής μηχανής και καθορίζουν το μοντέλο εκείνο της Κεντρικής Προβολής, που περιγράφει καλύτερα τη συγκεκριμένη φωτογραφική μηχανή. (Διαφάνειες Διαλέξεων Φωτογραμμετρίας 1 Εθνικού Μετσόβιου Πολυτεχνείου - 2021)

- Η σταθερά της μηχανής ( $c$ )
- Οι εικονοσυντεταγμένες ( $x_0, y_0$ ) του πρωτεύοντος σημείου ( $H$ )
- Οι τιμές της ακτινικής διαστροφής ( $\Delta r$ )



Εικόνα 21- Σχηματική απεικόνιση στοιχείων εσωτερικού προσανατολισμού (Έλλη Πέτσα 2000)

Οι συντεταγμένες του κέντρου προβολής ( $X_0, Y_0, Z_0$ ) και ο προσανατολισμός του άξονα λήψης ( $\omega, \phi, \kappa$ ) αποτελούν τις έξι παραμέτρους (στοιχεία) του εξωτερικού προσανατολισμού και καθορίζουν τη θέση και τη στροφή της κάμερας στο χώρο (σύστημα αναφοράς  $,XYZ$ ). Ο εξωτερικός προσανατολισμός μιας φωτογραφίας είναι συνήθως άγνωστος και υπολογίζεται με την χρήση φωτοσταθερών. Η σχέση που χρησιμοποιείται για την επίλυση του εξωτερικού προσανατολισμού και την εύρεση των έξι αγνώστων  $X_0, Y_0, Z_0, \omega, \phi, \kappa$  είναι η συνθήκη συγγραμμικότητας.



Εικόνα 22- Σχέση μεταξύ εικονοσυντεταγμένων και συντεταγμένων αντικειμένου (Kraus, σελ. 14)

Η συνθήκη αυτή εκφράζει με μαθηματικό τρόπο το γεγονός ότι το σημείο M στο αντικείμενο, το προβολικό κέντρο O και η εικόνα μ του σημείου στη φωτογραφία βρίσκονται πάνω σε ευθεία. Οι εξισώσεις αυτές διατυπώνονται υπό μορφή πινάκων ως εξής:

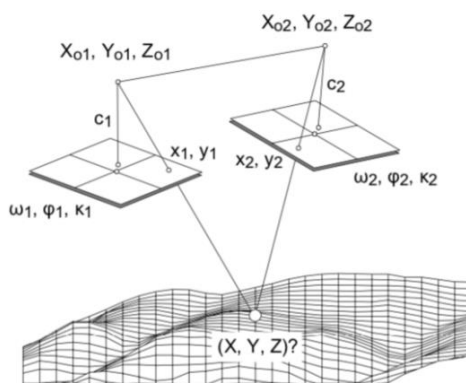
$$\begin{bmatrix} x \\ y \\ -c \end{bmatrix} = \lambda \cdot \mathfrak{R}_{\omega\varphi\kappa} \cdot \begin{bmatrix} X - X_0 \\ Y - Y_0 \\ Z - Z_0 \end{bmatrix}$$

Δεδομένου ότι η κλίμακα λ αφορά στο συγκεκριμένο σημείο και διαφοροποιείται από σημείο σε σημείο της λήψης, είναι επιθυμητή η εξάλειψή της. (Άσκηση 3 Φωτογραμμετρίας 1 Εθνικού Μετσόβιου Πολυτεχνείου)

$$\mathfrak{R}_{\omega\varphi\kappa} = \begin{pmatrix} \cos\varphi \cos\kappa & \sin\omega \sin\varphi \cos\kappa + \cos\omega \sin\kappa & -\cos\omega \sin\varphi \cos\kappa + \sin\omega \sin\kappa \\ -\cos\varphi \sin\kappa & -\sin\omega \sin\varphi \sin\kappa + \cos\omega \cos\kappa & \cos\omega \sin\varphi \sin\kappa + \sin\omega \cos\kappa \\ \sin\varphi & -\sin\omega \cos\varphi & \cos\omega \cos\varphi \end{pmatrix}$$

Κάποιες βασικές μέθοδοι που βασίζονται στο μαθηματικό μοντέλο της Συνθήκης Συγγραμμικότητας είναι η φωτογραμμετρική οπισθοτομία, η φωτογραμμετρική εμπροσθοτομία κι η βαθμονόμηση της μηχανής λήψης. Επισημαίνεται ότι, στα σύγχρονα όργανα, οι εικονοσυντεταγμένες μετρώνται με την ίδια ακρίβεια, τόσο για κάθε σημείο όσο και κατά x και y. Συνεπώς οι παρατηρήσεις στην προκειμένη περίπτωση είναι ισοβαρείς και έτσι δεν απαιτείται πίνακας βαρών. Η επίλυση ακολουθεί τη γνωστή διαδικασία των διαδοχικών προσεγγίσεων, δεδομένου ότι από κάθε επίλυση του συστήματος προκύπτουν τιμές για τις διαφορικές μεταβολές των προσωρινών τιμών, ώστε αυτές προστιθέμενες να πλησιάσουν την καλύτερη τιμή. Για την εμπροσθοτομία ισχύει ότι μετρώντας ομόλογα σημεία και γνωρίζοντας τον εξωτερικό προσανατολισμό δύο (ή και περισσότερων) εικόνων μπορούν να υπολογιστούν οι θέσεις σημείων στο χώρο (3D). Ο προσδιορισμός των γεωδαιτικών συντεταγμένων προκύπτει από την επίλυση της εμπροσθοτομίας, δηλαδή από τον υπολογισμό του σημείου τομής στο χώρο των δύο (τουλάχιστον) ομόλογων ακτίνων από τις δύο (τουλάχιστον) εικόνες. Δημιουργείται έτσι ένα σύστημα από 2n εξισώσεις

με τρεις αγνώστους , που λύνεται με τη βοήθεια της Μεθόδου Ελαχίστων Τετραγώνων(M.E.T.).



Εικόνα 23- Σχηματική απεικόνιση εμπροσθοτομίας

## ΚΕΦΑΛΑΙΟ 4

### Ανάλυση

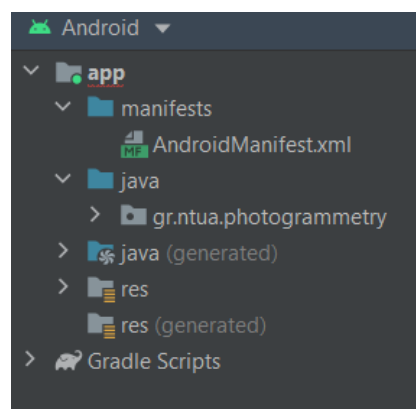
#### 4.1 Η Εφαρμογή

Η εφαρμογή της παρούσας διπλωματικής εργασίας ονομάστηκε 'Photogrammetry' και στοχεύει στην ικανοποίηση των βασικών παραμέτρων που αναφέρθηκαν προηγουμένως, στο πλαίσιο των απαιτήσεων ενός αλγορίθμου εκτέλεσης φωτογραμμετρικής οπισθοτομίας κι εμπροσθοτομίας. Στην εφαρμογή που παρουσιάζεται παρακάτω, ο χρήστης αποκτά αρκετές δυνατότητες αρκεί να ακολουθήσει σωστά τα βήματα που περιγράφονται. Αρχικά, κάθε κινητό τηλέφωνο διαθέτει φωτογραφική μηχανή. Αν ο εκάστοτε χρήστης τραβήξει μία φωτογραφία η οποία περιέχει τουλάχιστον τέσσερα φωτοσταθερά (σημεία γνωστών συντεταγμένων) τότε μπορεί με αρκετά μεγάλη ακρίβεια να εκτιμηθούν οι συντεταγμένες (X,Y,Z) κι οι γωνίες ( $\omega, \phi, \kappa$ ) του smartphone κατά τη στιγμή της λήψης της φωτογραφίας. Η εφαρμογή δίνει τη δυνατότητα στο χρήστη να εισέλθει στη συλλογή φωτογραφιών του και να επιλέξει τη φωτογραφία την οποία θέλει να επεξεργαστεί ώστε να παράγει τα ζητούμενα. Επόμενο βήμα είναι η εύρεση των εικονοσυντεταγμένων των φωτοσταθερών. Πιο συγκεκριμένα, ο χρήστης πρέπει να βρει τα pixels της φωτογραφίας, τα οποία αντιστοιχούν στα φωτοσταθερά που αναφέρθηκαν προηγουμένως. Αυτό γίνεται δυνατό επιλέγοντας απλώς, με ένα κλικ, πάνω στην οθόνη του κινητού το φωτοσταθερό και καταχωρώντας στο παράθυρο το οποίο αναδύεται τις συντεταγμένες X,Y,Z. Επαναλαμβάνοντας αυτή τη διαδικασία τουλάχιστον 4 φορές ο αλγόριθμος της οπισθοτομίας είναι έτοιμος να 'δουλέψει'. Πολύ σημαντικό είναι ο χρήστης να γνωρίζει ποια είναι η εστιακή απόσταση της κάμερας του κινητού κατά τη φωτογράφιση καθώς είναι αναγκαίο για τον αλγόριθμο

αφού αποτελεί παράμετρο για την επίλυση του εσωτερικού προσανατολισμού δηλαδή τη γεωμετρική αποκατάσταση της δέσμης των ακτίνων. Επόμενο βήμα και τελευταίο, είναι η επίλυση της εμπροσθοτομίας. Αυτό μπορεί να πραγματοποιηθεί εκτελώντας άλλη μια φωτογραμμετρική οπισθοτομία. Η φωτογραμμετρική εμπροσθοτομία είναι μια δεικνική διαδικασία και για να επιτευχθεί πρέπει να είναι γνωστοί και για τις δύο φωτογραφίες οι παράμετροι του εξωτερικού προσανατολισμού (X,Y,Z,ω,φ,κ). Επομένως, ο χρήστης πρέπει να ακολουθήσει την προηγούμενη διαδικασία άλλη μια φορά προσέχοντας η δεύτερη φωτογραφία να έχει επικαλυπτόμενο 'κομμάτι' με την πρώτη και φυσικά να διαθέτει τουλάχιστον τέσσερα φωτοσταθερά. Διαθέτοντας όλες αυτές τις παραμέτρους μπορεί να επιλυθεί η εμπροσθοτομία και πάλι με ένα μόνο κλικ στην οθόνη. Ο χρήστης μπορεί να επιλέξει όποιο εικονοστοιχείο επιθυμεί (που να ανήκει στο επικαλυπτόμενο 'κομμάτι') και η εφαρμογή θα δώσει τις συντεταγμένες του με αρκετά μεγάλη ακρίβεια όπως θα διαπιστώσουμε και παρακάτω.

## 4.2 Περιήγηση στο Android Studio

Το Android Studio είναι ένα ολοκληρωμένο προγραμματιστικό περιβάλλον για ανάπτυξη εφαρμογών στην πλατφόρμα Android. Υπάρχει μια σειρά βημάτων που μπορεί να ακολουθηθεί έτσι ώστε να δημιουργηθεί μια εφαρμογή. Σε όλες τις εφαρμογές Android κάθε προγραμματιστής καλείται να ασχοληθεί με το λειτουργικό και με το οπτικό κομμάτι. Στο αριστερό μέρος της πλατφόρμας εμφανίζεται ένας φάκελος ο οποίος ονομάζεται 'app'. Μέσα σε αυτό το φάκελο βρίσκονται αποθηκευμένοι άλλοι υποφάκελοι οι οποίοι περιέχουν ό,τι μπορεί να χρειαστεί για την υλοποίηση του εκάστοτε project. Ο φάκελος 'manifests' περιέχει πληροφορίες και άδειες για την εφαρμογή.



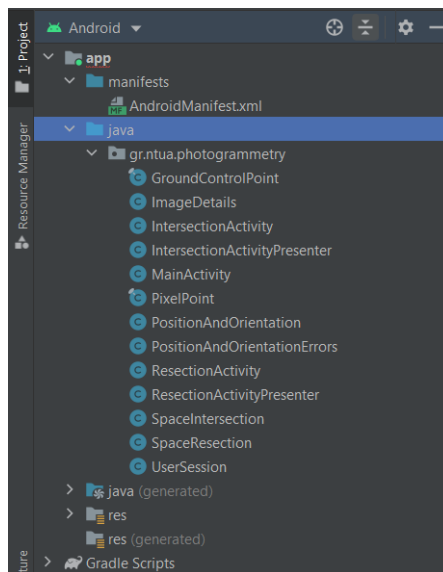
Εικόνα 24- Φάκελοι των οντοτήτων που αποτελούν την εφαρμογή

Το αρχείο 'AndroidManifest.xml' παρέχει πληροφορίες όπως το όνομα της εφαρμογής ('Photogrammetry'), άδεια ώστε σε κάθε στιγμή χρήσης της εφαρμογής η οθόνη να έχει 'κάθετο' προσανατολισμό, το όνομα της εικόνας που χρησιμοποιείται ως φόντο καθώς και το όνομα της οθόνης (activity) που εμφανίζεται κατά την ενεργοποίηση της εφαρμογής ('MainActivity').

```
AndroidManifest.xml x
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="gr.ntua.photogrammetry">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="Photogrammetry"
9         android:supportsRtl="true"
10        android:theme="@style/AppTheme">
11        <activity android:name=".IntersectionActivity" android:screenOrientation="portrait"/>
12        <activity android:name=".ResectionActivity" android:screenOrientation="portrait"/>
13        <activity android:name=".MainActivity" android:screenOrientation="portrait">
14            <intent-filter>
15                <action android:name="android.intent.action.MAIN" />
16
17                <category android:name="android.intent.category.LAUNCHER" />
18            </intent-filter>
19        </activity>
20
21        <meta-data
22            android:name="preloaded_fonts"
23            android:resource="@array/preloaded_fonts" />
24    </application>
25
26 </manifest>
```

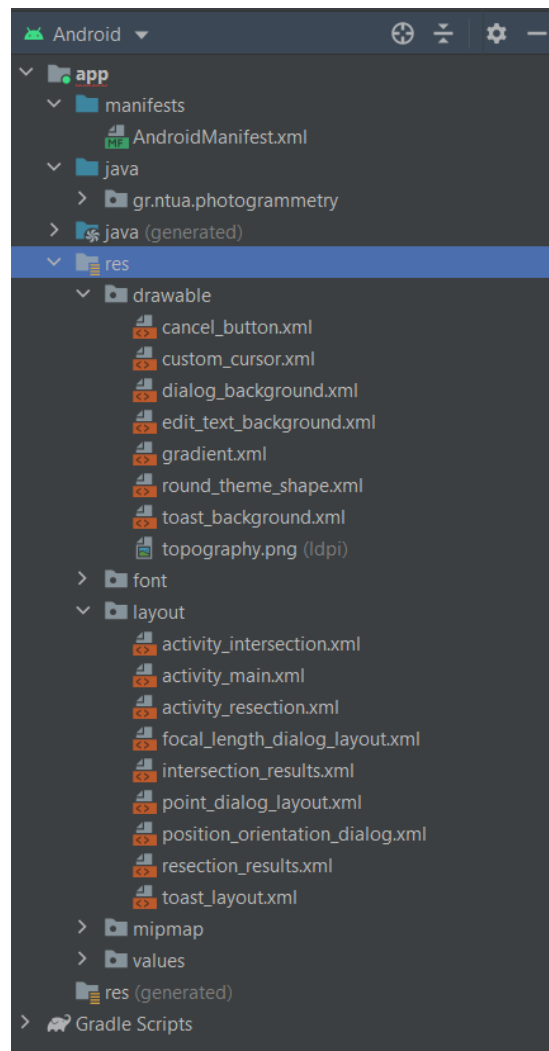
Εικόνα 25- Κώδικας που αποτελεί το 'AndroidManifest'

Στον φάκελο 'java' είναι αποθηκευμένες οι κλάσεις της εφαρμογής. Οι κλάσεις αποτελούν το λειτουργικό 'σώμα' κάθε προγράμματος στη Java.



Εικόνα 26- Περιεχόμενο φακέλου 'java'

Στο φάκελο 'res' εμφανίζονται υποφάκελοι οι οποίοι περιέχουν το γραφικό περιβάλλον της εφαρμογής όπως εικόνες, κουμπιά, χρώματα και φόντο.



**Εικόνα 27- Περιεχόμενο φακέλου 'res'**

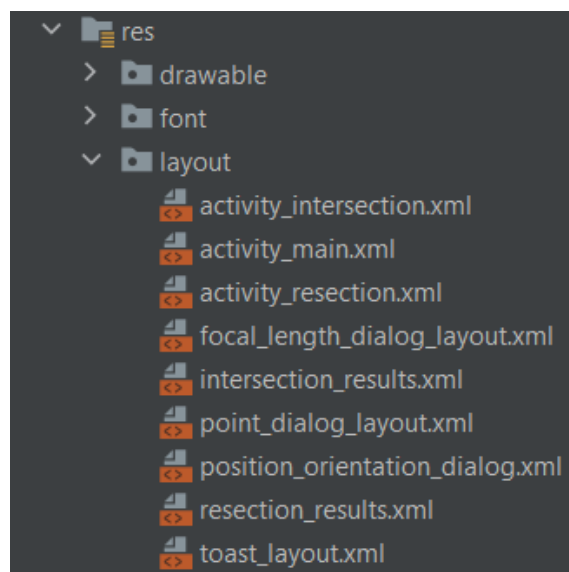
Στο κέντρο της πλατφόρμας βρίσκεται ο κώδικας του προγράμματος, ενώ γύρω του υπάρχουν αρκετά εικονίδια τα οποία αποτελούν εργαλεία για την τέλεση της εφαρμογής.

```
1 package gr.ntua.photogrammetry;
2
3 import ...
4
5
6
7
8
9 public class ImageDetails {
10     private Bitmap image;
11     private List<GroundControlPoint> groundControlPoints;
12     private Stack<PixelPoint> pixelPoints;
13     private PositionAndOrientation positionAndOrientation;
14
15     public ImageDetails(Bitmap image) {
16         this.image = image;
17         this.groundControlPoints = new ArrayList<>();
18         this.pixelPoints = new Stack<>();
19     }
20
21     public Bitmap getImage() { return image; }
22
23
24     public List<GroundControlPoint> getGroundControlPoints() { return groundControlPoints; }
25
26
27     public Stack<PixelPoint> getPixelPoints() { return pixelPoints; }
28
29
30
31
32
33     public int getTotalPointsProvided() { return groundControlPoints.size(); }
34
35 }
```

Εικόνα 28- Κώδικας και εργαλεία στο Android Studio

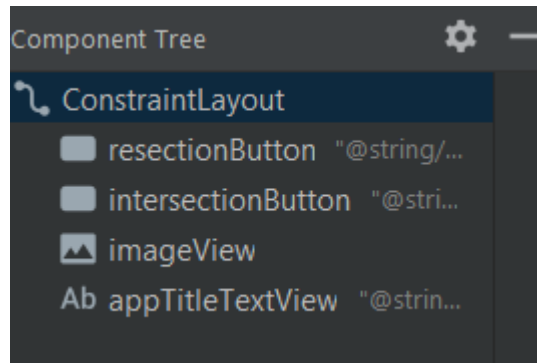
### 4.3 Μορφοποίηση της εφαρμογής

Οτιδήποτε βλέπει ένας χρήστης μιας εφαρμογής στην οθόνη του 'έξυπνου' τηλεφώνου του αποτελεί μία 'Activity'. Στην κάθε activity μπορούμε να δημιουργήσουμε γραφικά στοιχεία όπου 'μέσα' εκεί θα συμβαίνουν κάποιες λειτουργίες. Γι' αυτό το λόγο κάθε activity αποτελείται από ένα αρχείο με κατάληξη '.xml' που περιέχει αυτά που βλέπει ο χρήστης και ένα αρχείο με κατάληξη '.java' που είναι οι λειτουργίες της. Στο φάκελο 'res' συναντάται ο υποφάκελος 'layout'. Στο περιεχόμενό του βρίσκονται τα γραφικά περιβάλλοντα όλων των activities. Η πρώτη activity που εμφανίζεται κατά την ενεργοποίηση της εφαρμογής, όπως προαναφέρθηκε, είναι η 'MainActivity'. Το αρχείο 'activity\_main.xml' αποτελεί το γραφικό περιβάλλον της.



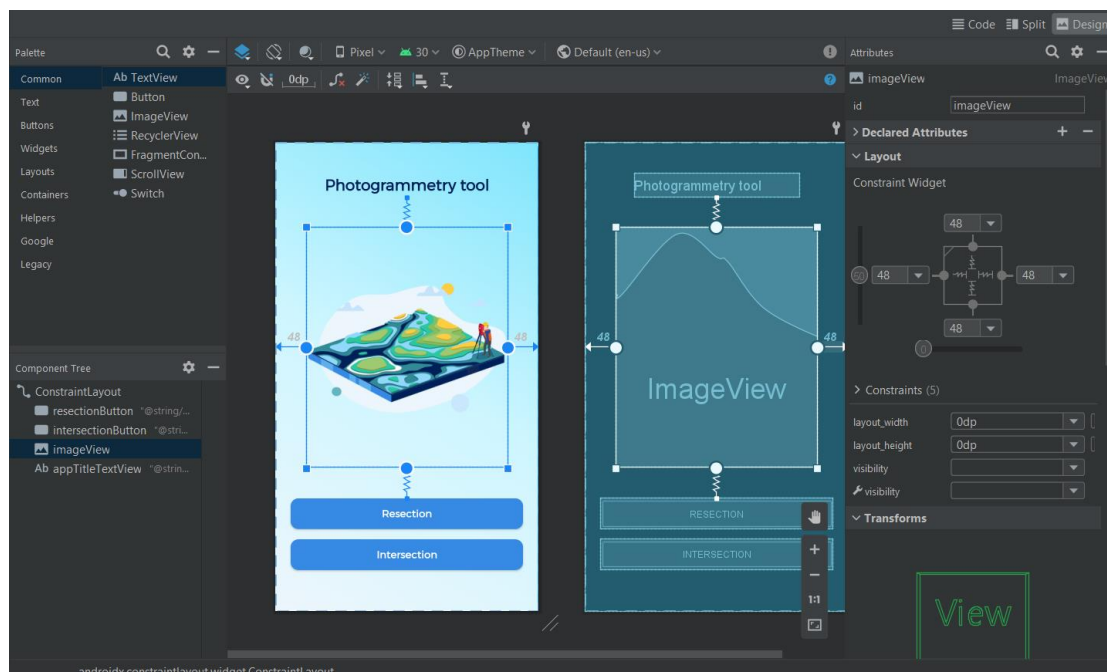
Εικόνα 29- Αρχεία '.xml'

Στη συγκεκριμένη εφαρμογή η αρχική οθόνη, αποτελείται από ένα κείμενο ('Photogrammetry tool'), από μία εικόνα και δύο κουμπιά ('Resection' και 'Intersection'). Η διαμόρφωση έγινε με βάση το μέγεθος της οθόνης που θα εκτελείται η εφαρμογή και όχι με συγκεκριμένες διαστάσεις για όλους τους τύπους οθόνης. Αυτό ονομάζεται 'constraint\_layout'.



**Εικόνα 30- Διαμόρφωση και οντότητες της 'MainActivity'**

Όλα τα στοιχεία που εμφανίζονται στην αρχική οθόνη είναι 'αγκιστρωμένα' μεταξύ τους, έτσι ώστε να μην αλλάζουν οι θέσεις τους άναρχα σε προβολή διαφορετικών οθονών. Η διαμόρφωση μιας activity στο Android Studio δε μπορεί να ολοκληρωθεί χωρίς τη χρήση κώδικα. Κάθε οντότητα που εμφανίζεται επιβάλλεται να δηλώνεται ως προς τις υπόλοιπες οντότητες και ως προς την οθόνη. Πρέπει επίσης να δηλώνονται οι πληροφορίες αυτές που της δίνουν σχήμα, μέγεθος, χρώμα αλλά και κωδική ονομασία.



**Εικόνα 31- Οθόνη 'MainActivity'**



```

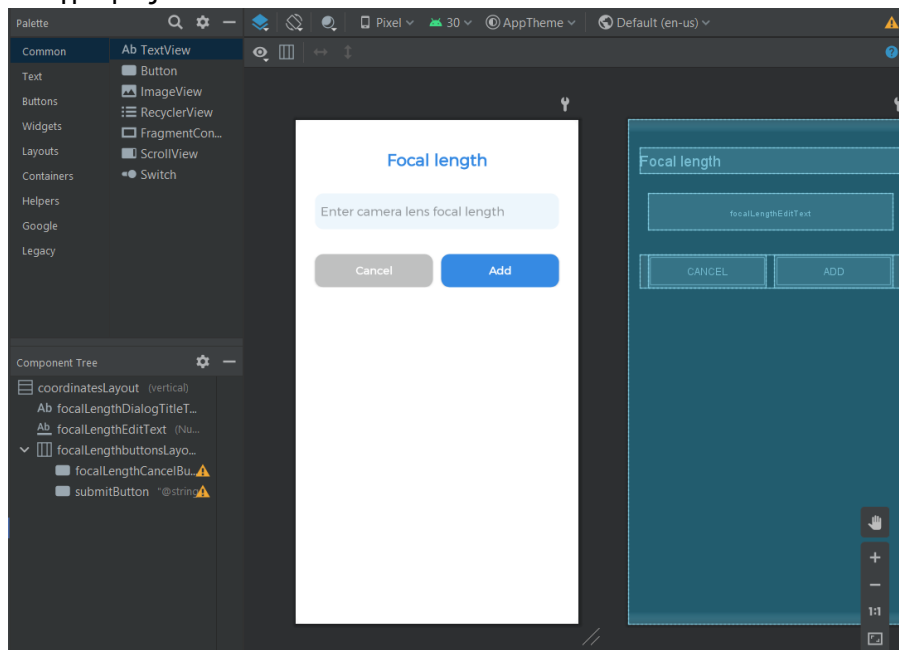
1 <?xml version="1.0" encoding="utf-8"?>
2 <androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     xmlns:app="http://schemas.android.com/apk/res-auto"
4     xmlns:tools="http://schemas.android.com/tools"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:background="@drawable/gradient"
8     tools:context=".MainActivity">
9
10    <Button
11        android:id="@+id/resectionButton"
12        android:layout_width="0dp"
13        android:layout_height="wrap_content"
14        android:layout_marginStart="24dp"
15        android:layout_marginEnd="24dp"
16        android:layout_marginBottom="16dp"
17        android:background="@drawable/round_theme_shape"
18        android:text="Resection"
19        android:textColor="@color/white"
20        android:textSize="16sp"
21        android:textStyle="bold"
22        app:layout_constraintBottom_toTopOf="@+id/intersectionButton"
23        app:layout_constraintEnd_toEndOf="parent"
24        app:layout_constraintHorizontal_bias="0.0"
25        app:layout_constraintStart_toStartOf="parent" />
26

```

**Εικόνα 32- Κώδικας για τη διαμόρφωση του φόντου της οθόνης και του κουμπιού 'Resection'**

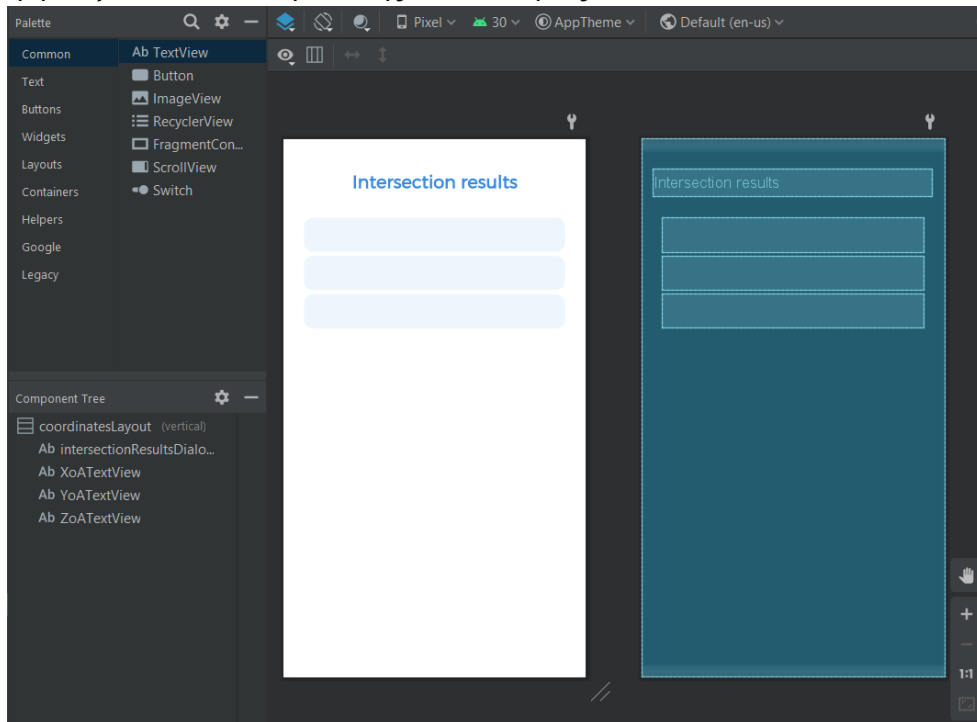
Με τον ίδιο τρόπο έχουν συνταχθεί και οι υπόλοιπες activities της εφαρμογής. Παρακάτω παρατίθενται όλες οι οθόνες που συναντώνται στην εφαρμογή καθώς και οι ονομασίες των αρχείων 'xml' τους.

- Η 'activity\_resection.xml' και η 'activity\_intersection.xml' είναι κενές οθόνες καθώς πάνω σε αυτές εμφανίζονται οι φωτογραφίες όπου επιλέγει ο χρήστης για να εκτελέσει την οπισθοτομία και την εμπροσθοτομία αντίστοιχα.
- Η 'Focal\_length\_dialog\_layout.xml' είναι η οθόνη όπου ο χρήστης καλείται να γράψει την εστιακή απόσταση της κάμερας του 'smartphone' όπου λήφθηκαν οι φωτογραφίες.

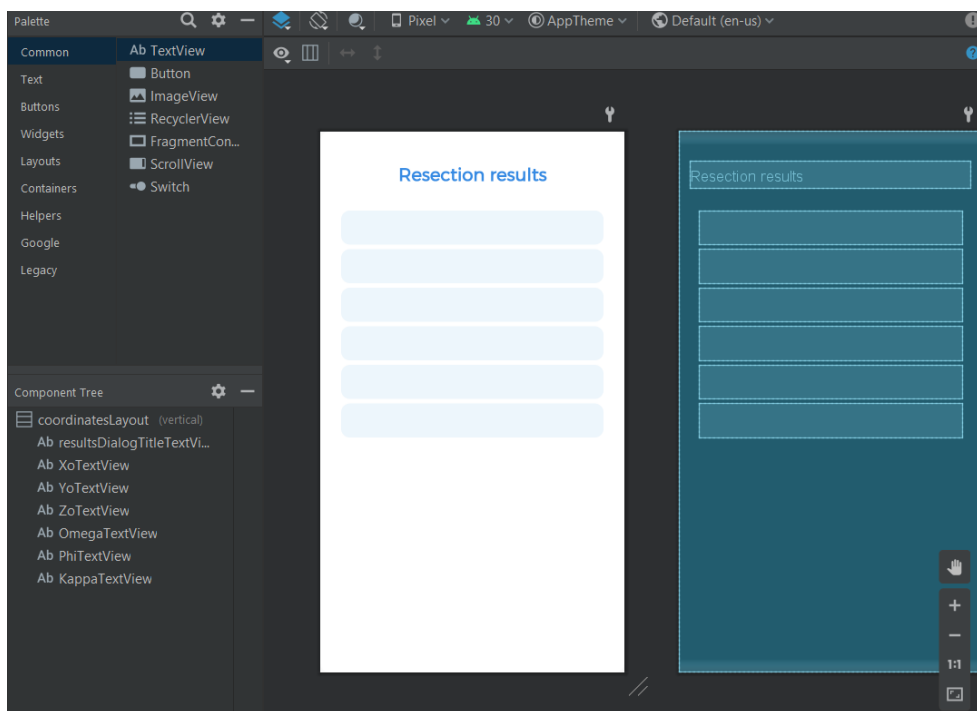


**Εικόνα 33- 'Focal\_length\_dialog\_layout.xml'**

- Η 'Intersection\_results.xml' είναι η οθόνη η οποία εμφανίζει τα αποτελέσματα της εμπροσθοτομίας στο χρήστη και αντίστοιχα η 'Resection\_results.xml' εμφανίζει τα αποτελέσματα της οπισθοτομίας.

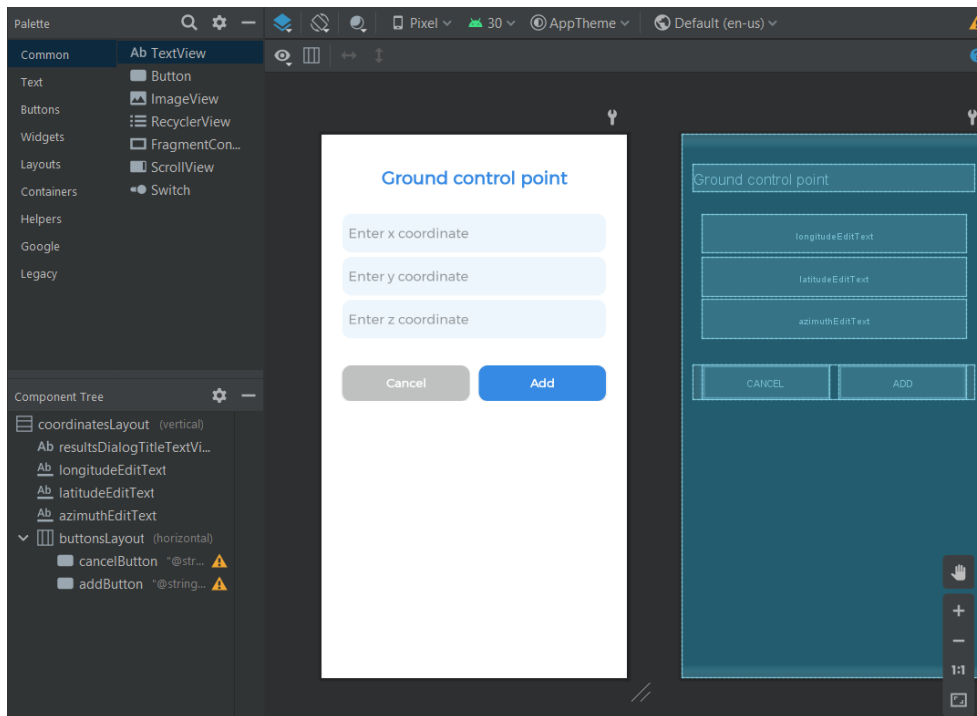


**Εικόνα 34- 'Intersection\_results.xml'**



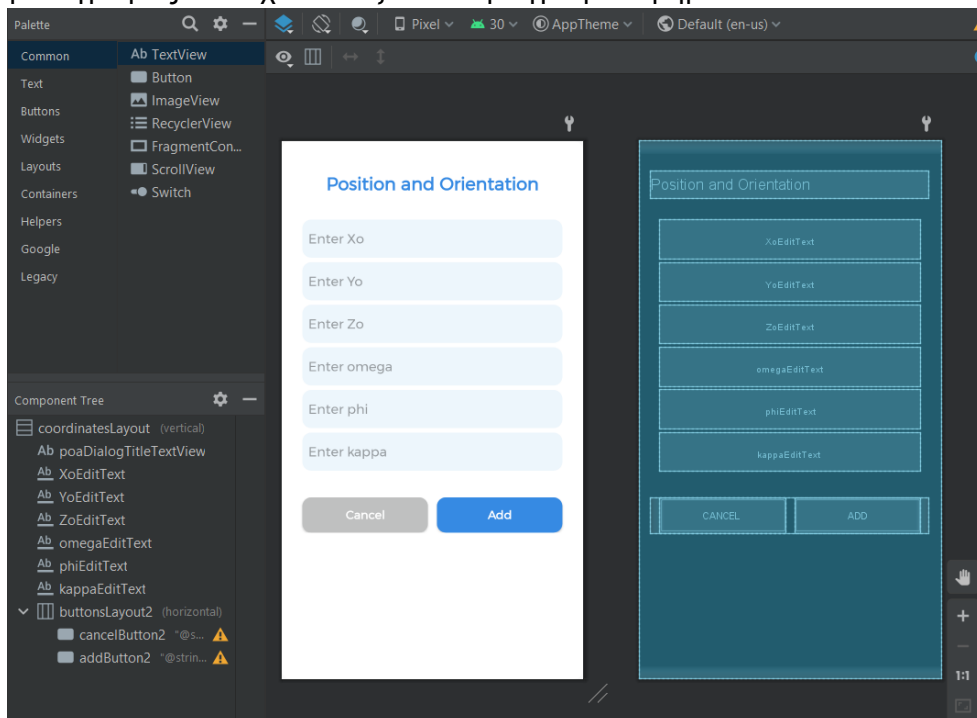
**Εικόνα 35- 'Resection\_results.xml'**

- Η 'Point\_dialog\_layout.xml' είναι η οθόνη η οποία ο χρήστης καλείται να γράψει τις φωτογραμμετρικές συντεταγμένες που αντιστοιχούν στο σημείο της φωτογραφίας που έχει επιλέξει στο προηγούμενο βήμα.



Εικόνα 36- 'Point\_dialog\_layout.xml'

- Η 'Position\_orientation\_dialog.xml' είναι η οθόνη που παρουσιάζεται ώστε ο χρήστης να γράψει τα στοιχεία του εξωτερικού προσανατολισμού της φωτογραφίας που έχει επιλέξει στο προηγούμενο βήμα.



Εικόνα 37- 'Position\_orientation\_dialog.xml'

- Η 'Toast\_layout.xml' είναι μια κενή οθόνη η οποία παρουσιάζεται με επικάλυψη τη τελευταία φωτογραφία που χρησιμοποιήθηκε μετά την ολοκλήρωση των εκάστοτε υπολογισμών.

## 4.4 Κώδικας

Όπως έχει αναφερθεί η Java είναι μια αντικειμενοστραφής γλώσσα προγραμματισμού. Ο βασικότερος τρόπος δόμησης ενός χώρου προβλήματος είναι η κατηγοριοποίηση των εννοιών σε αντικείμενα τα οποία ανήκουν σε κλάσεις αντικειμένων, οι οποίες με τη σειρά τους μπορεί να ανήκουν σε κάποιες άλλες κλάσεις κ.λ.π. Με αυτή τη λογική δημιουργήθηκε η εφαρμογή 'Photogrammetry'.

### 4.4.1 Κλάσεις δεδομένων

Μία κλάση δεδομένων λειτουργεί ως 'κοντέινερ' για δεδομένα που χρησιμοποιούνται από άλλες κλάσεις. Αυτές οι κλάσεις δεν περιέχουν καμία πρόσθετη λειτουργικότητα και δε μπορούν να λειτουργούν ανεξάρτητα από τα δεδομένα που κατέχουν. Για την επίλυση της οπισθοτομίας και της εμπροσθοτομίας χρειάζονται τα σημεία γνωστών φωτογραμμετρικών συντεταγμένων και τα εικονοστοιχεία στα οποία εμφανίζονται αντίστοιχα τα φωτοσταθερά. Για αυτόν το λόγο δημιουργήθηκαν δύο κλάσεις, που ονομάζονται, 'GroundControlPoint' και 'PixelPoint'. Αυτές οι δύο κλάσεις δεν έχουν μεταξύ τους καμία αλληλεπίδραση αλλά έχουν γραφτεί με την ίδια ακριβώς λογική κατασκευής. Η 'GroundControlPoint' αποτελεί αναπαράσταση ομαδοποίησης των συντεταγμένων των φωτοσταθερών (X,Y,Z), ενώ η 'PixelPoint' αποτελεί αναπαράσταση ομαδοποίησης των εικονοσυντεταγμένων (x,y).

```
public final class GroundControlPoint {  
  
    private double x;  
    private double y;  
    private double z;  
  
    public GroundControlPoint(double x, double y, double z) {  
        this.x = x;  
        this.y = y;  
        this.z = z;  
    }  
  
    public double getX() { return this.x; }  
  
    public double getY() { return this.y; }  
  
    public double getZ() { return this.z; }  
  
    public double[] getCoordinates() { return new double[]{this.x, this.y, this.z}; }  
}
```

Εικόνα 38- Μέρος του κώδικα της κλάσης 'GroundControlPoint'

```

public final class PixelPoint {

    private double x;
    private double y;

    public PixelPoint(double x, double y) {
        this.x = x;
        this.y = y;
    }

    public double getX() { return this.x; }

    public double getY() { return this.y; }

    public double[] getCoordinates() { return new double[]{this.x, this.y}; }
}

```

Εικόνα 39- Μέρος του κώδικα της κλάσης 'PixelPoint'

Στη αρχή κάθε κλάσης πρέπει να δηλώνονται οι μεταβλητές που χρησιμοποιούνται σε αυτήν. Στη συνέχεια καλείται ο 'κατασκευαστής' ο οποίος είναι υπεύθυνος για τη δημιουργία των αντικειμένων. Η λογική που κρύβεται πίσω από τη λέξη 'this.' είναι ότι το ίδιο το αντικείμενο το οποίο περιμένει από το χρήστη να πάρει τιμή δηλώνει τον εαυτό του ότι θα τη λάβει. Με τη χρήση της μεθόδου 'get' , παρέχεται πρόσβαση στις τιμές των μεταβλητών μελών της κλάσης. Το Android Studio κατά τη σύνταξη του εκάστοτε κώδικα βοηθάει τον προγραμματιστή, δημιουργώντας αυτόματα πεδία τα οποία πρέπει να συμπληρώσει, ώστε να αποφευχθούν λάθη τα οποία μπορούν να επιφέρουν τη μη εκτέλεση του προγράμματος π.χ. εξαιρέσεις. Στο συγκεκριμένο πρόγραμμα, καθώς ολοκληρώνεται η οπισθοτομία προκύπτουν κάποια δεδομένα τα οποία χρειάζεται να υπάρχουν όλα μαζί, σαν πακέτο. Για αυτό το λόγο δημιουργήθηκε μία ακόμα κλάση δεδομένων η οποία ονομάστηκε 'PositionAndOrientation'. Με αυτό τον τρόπο επιτυγχάνεται η ομαδοποίηση των (Χο,Υο,Ζο,ω,φ,κ). Αυτό μπορεί να επιφέρει όφελος σε μια ενδεχόμενη μελλοντική αναβάθμιση της εφαρμογής καθώς με αυτό τον τρόπο γίνεται εφικτή η αυτόματη εισαγωγή τιμών κατά την ενεργοποίηση της εμπροσθοτομίας.

```

public class PositionAndOrientation implements Serializable {
    private double Xo, Yo, Zo;
    private double omega, phi, kappa;
    private PositionAndOrientationErrors errors;

    public PositionAndOrientation(double Xo, double Yo, double Zo, double omega, double phi, double kappa) {
        this.Xo = Xo;
        this.Yo = Yo;
        this.Zo = Zo;
        this.omega = omega;
        this.phi = phi;
        this.kappa = kappa;
    }

    public double getXo() { return Xo; }

    public double getYo() { return Yo; }

    public double getZo() { return Zo; }

    public double getOmega() { return omega; }

    public double getPhi() { return phi; }

    public double getKappa() { return kappa; }
}

```

Εικόνα 40- Μέρος του κώδικα της κλάσης 'PositionAndOrientation'

#### 4.4.2 Κλάση Οπισθοτομίας

Η 'SpaceResection' είναι η κλάση η οποία εκτελεί την οπισθοτομία. Όπως επιβάλλεται στη Java, αρχικά πρέπει να δηλωθούν όλες οι μεταβλητές που χρειάζεται να εμφανιστούν στην κλάση.

```

public class SpaceResection {

    private final RealMatrix groundControlPoints;
    private final RealMatrix pixels;
    private final int numOfPoints;

    private final double photoScale;
    private final double photoDistance;
    private final double C;

    private RealMatrix R;
    private RealMatrix f;

    private RealMatrix A;
    private RealVector A1;
    private RealVector A2;
    private RealVector P;
    private RealVector B;

    private double Xo, Yo, Zo;
    private double omega, phi, kappa;

    private RealVector accuracy;

    private PositionAndOrientation positionAndOrientation;
}

```

Εικόνα 41- Δήλωση μεταβλητών στην κλάση 'SpaceResection'

Η βασική έκδοση της Java διαθέτει πίνακες αλλά δε μπορούν να εκτελέσουν ούτε τις απλές μαθηματικές πράξεις. Αυτό αντιμετωπίστηκε με εισαγωγή μιας βιβλιοθήκης γραμμικής Άλγεβρας στον κώδικα με όνομα 'Apache.commons.math3.linear'. Βιβλιοθήκη καλείται μια συλλογή από έτοιμα προγράμματα που περιέχουν

υποβοηθητικό κώδικα και δεδομένα, παρέχοντας, με αυτόν τον τρόπο, υπηρεσίες σε προγράμματα. Οι μεταβλητές που δηλώνονται στον κώδικα είναι τα σημεία φωτογραμμετρικών συντεταγμένων και οι εικονοσυντεταγμένες αυτών (σε μορφή πίνακα), μια ακέραια μεταβλητή με όνομα 'NumOfPoints', η κλίμακα ('photoscale'), η απόσταση που τραβήχτηκε η φωτογραφία ('photoDistance'), η εστιακή απόσταση ('C'), οι πίνακες 'R', 'f', 'A', τα διανύσματα 'A1','A2','P','U' και τα X<sub>0</sub>, Y<sub>0</sub>, Z<sub>0</sub>, ω, φ, κ. Με την ίδια λογική δηλώνονται και οι σταθερές 'RCC', ο μέγιστος αριθμός επαναλήψεων, τα όριο διαφοράς της απόστασης, το γωνιακό όριο, ο ελάχιστος αριθμός σημείων(4) που θα χρησιμοποιηθούν αλλά και ο μέγιστος(12).

```
private static final int X = 0;
private static final int Y = 1;
private static final int Z = 2;
private static final int RCC = 636620;
private static final int MAX_ITERATIONS = 50;
private static final double LINEAR_TOLERANCE = 0.1;
private final double ANGLE_TOLERANCE;

public static final int MINIMUM_POINTS_REQUIRED = 4;
public static final int MAXIMUM_POINTS_ALLOWED = 12;
```

Εικόνα 42- Δήλωση σταθερών στην κλάση 'SpaceResection'

Η μέθοδος ξεκινάει με τη συνθήκη ότι ο χρήστης έχει δηλώσει τουλάχιστον 4 σημεία. Η μεταβλητή 'numOfPoints' λαμβάνει τιμή αναλόγως με το μέγεθος του πίνακα που έχουν αποθηκευτεί οι συντεταγμένες των φωτοσταθερών. Ο πίνακας αυτός αλλά και ο πίνακας των εικονοσυντεταγμένων 'γεμίζει' με τιμές με τη βοήθεια της μεθόδου 'setRow' της 'commons.math3.linear'.

```
public SpaceResection(List<GroundControlPoint> groundControlPoints, List<PixelPoint> pixels, double focalLength) {
    // method needs at least 4 points to work
    if ((groundControlPoints.size() < MINIMUM_POINTS_REQUIRED) || (pixels.size() < MINIMUM_POINTS_REQUIRED)) {
        throw new IllegalArgumentException("Must provide at least 4 points");
    } else if (pixels.size() != groundControlPoints.size()) {
        throw new IllegalArgumentException("Must provide equal number of groundControlPoints and pixels");
    }

    this.numOfPoints = groundControlPoints.size();
    this.C = focalLength;

    // fill ground control points matrix from input list
    this.groundControlPoints = MatrixUtils.createRealMatrix(numOfPoints, columns: 3);
    for (int i = 0; i < this.numOfPoints; i++) {
        this.groundControlPoints.setRow(i, groundControlPoints.get(i).getCoordinates());
    }

    // fill pixels matrix from input list
    this.pixels = MatrixUtils.createRealMatrix(numOfPoints, columns: 2);
    for (int i = 0; i < this.numOfPoints; i++) {
        this.pixels.setRow(i, pixels.get(i).getCoordinates());
    }
}
```

Εικόνα 43- Μέρος του κώδικα της κλάσης 'SpaceResection'

Επόμενο βήμα αποτέλεσε ο προσδιορισμός της κλίμακας και της απόστασης φωτογράφισης. Υποθέτοντας ότι η φωτογραφία είναι αυστηρώς κατακόρυφη ισχύει

$$\frac{x_a}{X-X_0} = \frac{y_a}{Y-Y_0} = \frac{-c}{Z-Z_0} = \frac{1}{k}$$

Με γνωστή την κλίμακα υπολογίζεται μια προσέγγιση απόστασης φωτογράφισης

$$H = -(Z - Z_0) = c \cdot k$$

Σειρά έχει ο υπολογισμός του γωνιακού ορίου και η δημιουργία ενός πίνακα A του οποίου οι γραμμές είναι 2 \* τη μεταβλητή 'numOfPoints' και οι στήλες είναι 6 ( άγνωστοι της οπισθοτομίας ).

```
// approximate photo scale assuming it was shot vertically
this.photoScale = Math.abs((this.groundControlPoints.getEntry(0,0,X) - calculateAverage(X)) / this.pixels.getEntry(0,0,X)) * 1000;

// estimate the distance H from which the photo was taken
this.photoDistance = this.C * (this.photoScale / 1000);

// calculate angle tolerance
this.ANGLE_TOLERANCE = Math.atan(LINEAR_TOLERANCE / this.photoDistance) * RCC;

// initialize A matrix
this.A = MatrixUtils.createRealMatrix( rows: 2 * numOfPoints, columns: 6);

initialize();
```

Εικόνα 44- Μέρος του κώδικα της κλάσης 'SpaceResection'

Η τιμές των προσεγγιστικών γωνιών καθορίστηκαν ως '0' ενώ των X<sub>0</sub>, Y<sub>0</sub> όσο ο μέσος όρος των X και των Y των φωτοσταθερών. Αντίστοιχα, το Z<sub>0</sub> υπολογίστηκε από το άθροισμα της απόστασης λήψης με τον μέσο όρο των προσεγγιστικών φωτογραμμετρικών Z. Βασικό εργαλείο για την επίλυση της οπισθοτομίας αποτελεί ο πίνακας στροφής R<sub>ωφκ</sub> καθώς και ο πίνακας των μερικών παραγώγων του ως προς φ.

```
private RealMatrix calculateRotationMatrix() {

    double R[][] = new double[3][3];

    R[0][0] = Math.cos(Math.PI / 200 * phi) * Math.cos(Math.PI / 200 * kappa);
    R[0][1] = Math.sin(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Mat
    R[0][2] = -Math.cos(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Mat
    R[1][0] = -Math.cos(Math.PI / 200 / 200 * phi) * Math.sin(Math.PI / 200 * kappa)
    R[1][1] = -Math.sin(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Ma
    R[1][2] = Math.cos(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Mat
    R[2][0] = Math.sin(Math.PI / 200 * phi);
    R[2][1] = -Math.sin(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi);
    R[2][2] = Math.cos(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi);
```

Εικόνα 45- Υπολογισμός πίνακα στροφής της κλάσης 'SpaceResection'



```

private RealMatrix calculateFMatrix() {

    double f[][] = new double[3][3];

    f[0][0] = -Math.sin(Math.PI / 200 * phi) * Math.cos(Math.PI / 200 * kappa);
    f[0][1] = Math.sin(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi) * Math.cos(Math.PI / 200 * kappa);
    f[0][2] = -Math.cos(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi) * Math.cos(Math.PI / 200 * kappa);
    f[1][0] = Math.sin(Math.PI / 200 * phi) * Math.sin(Math.PI / 200 * kappa);
    f[1][1] = -Math.sin(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi) * Math.sin(Math.PI / 200 * kappa);
    f[1][2] = Math.cos(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi) * Math.sin(Math.PI / 200 * kappa);
    f[2][0] = Math.cos(Math.PI / 200 * phi);
    f[2][1] = Math.sin(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi);
    f[2][2] = -Math.cos(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi);

    return MatrixUtils.createRealMatrix(f);
}

```

**Εικόνα 46- Υπολογισμός πίνακα 'f' της κλάσης 'SpaceResection'**

Ακολουθεί ο υπολογισμός των διανυσμάτων 'A1', 'A2' που αποτελούν αριθμητές για το x και το y αντίστοιχα στη συνθήκη συγγραμμικότητας για κάθε φωτοσταθερό. Φυσικά με την ίδια μέθοδο υπολογίζεται και το διάνυσμα 'Π' που είναι ο παρονομαστής για τη συνθήκη συγγραμμικότητας για κάθε φωτοσταθερό.

```

private RealVector calculateA1Vector() {

    double[] A1 = new double[numOfPoints];

    for(int i = 0; i < numOfPoints; i++) {
        A1[i] = R.getEntry(0, 0, 0) * (groundControlPoints.getEntry(i, X) - Xo) + R.getEntry(0, 0, 1) * (groundControlPoints.getEntry(i, Y) - Yo)
    }

    return new ArrayRealVector(A1);
}

private RealVector calculateA2Vector() {

    double[] A2 = new double[numOfPoints];

    for(int i = 0; i < numOfPoints; i++) {
        A2[i] = R.getEntry(1, 0, 0) * (groundControlPoints.getEntry(i, X) - Xo) + R.getEntry(1, 0, 1) * (groundControlPoints.getEntry(i, Y) - Yo)
    }

    return new ArrayRealVector(A2);
}

private RealVector calculatePiVector() {

    double[] P = new double[numOfPoints];

    for(int i = 0; i < numOfPoints; i++) {
        P[i] = R.getEntry(2, 0, 0) * (groundControlPoints.getEntry(i, X) - Xo) + R.getEntry(2, 0, 1) * (groundControlPoints.getEntry(i, Y) - Yo)
    }

    return new ArrayRealVector(P);
}

```

**Εικόνα 47- Υπολογισμός διανυσμάτων 'A1', 'A2', 'Π' της κλάσης 'SpaceResection'**

Για τη δημιουργία του πίνακα A πρέπει να υπολογιστούν μερικές παράγωγοι αρχικής εξίσωσης x ως προς τις άγνωστες παραμέτρους  $X_0, Y_0, Z_0$  ( $\alpha_1, \alpha_2, \alpha_3$ ), οι μερικές παράγωγοι αρχικής εξίσωσης y ως προς τις άγνωστες παραμέτρους  $\omega, \phi, \kappa$  ( $\alpha_4, \alpha_5, \alpha_6$ ), οι μερικές παράγωγοι αρχικής εξίσωσης y ως προς τις άγνωστες παραμέτρους  $X_0, Y_0, Z_0$  ( $\beta_1, \beta_2, \beta_3$ ) και οι μερικές παράγωγοι αρχικής εξίσωσης y ως προς τις άγνωστες παραμέτρους  $\omega, \phi, \kappa$  ( $\beta_4, \beta_5, \beta_6$ ).

```

private double[] calculateCoefa2() {
    double a2[] = new double[numOfPoints];

    for(int i = 0; i < numOfPoints; i++) {
        a2[i] = (C / Math.pow(P.getEntry(i), 2)) * (-A1.getEntry(i) * R.getEntry(i, 2, 1) + P.getEntry(i) * R.getEntry(i, 0, 1));
    }

    return a2;
}

private double[] calculateCoefa3() {
    double a3[] = new double[numOfPoints];

    for(int i = 0; i < numOfPoints; i++) {
        a3[i] = (C / Math.pow(P.getEntry(i), 2)) * (-A1.getEntry(i) * R.getEntry(i, 2, 2) + P.getEntry(i) * R.getEntry(i, 0, 2));
    }

    return a3;
}

```

**Εικόνα 48- Υπολογισμός συνιστωσών 'a2', 'a3' της κλάσης 'SpaceResection'**

Όλες αυτές οι συνιστώσες ομαδοποιήθηκαν σε 2 πίνακες. Ο πίνακας 'dFx' αποτελείται από γραμμές ίσες με τον αριθμό των σημείων που χρησιμοποίησε ο χρήστης και από 6 στήλες. Το περιεχόμενό του είναι οι μερικές παράγωγοι της αρχικής εξίσωσης x. Αντίστοιχα ο πίνακας 'dFy' περιέχει τις μερικές παραγώγους της αρχικής εξίσωσης y.

```

// dFx matrix contains the partial derivatives of X
RealMatrix dFx = MatrixUtils.createRealMatrix(numOfPoints, columns: 6);
dFx.setColumn(i: 0, calculateCoefa1());
dFx.setColumn(i: 1, calculateCoefa2());
dFx.setColumn(i: 2, calculateCoefa3());
dFx.setColumn(i: 3, calculateCoefa4());
dFx.setColumn(i: 4, calculateCoefa5());
dFx.setColumn(i: 5, calculateCoefa6());

// dFy matrix contains the partial derivatives of Y
RealMatrix dFy = MatrixUtils.createRealMatrix(numOfPoints, columns: 6);
dFy.setColumn(i: 0, calculateCoefb1());
dFy.setColumn(i: 1, calculateCoefb2());
dFy.setColumn(i: 2, calculateCoefb3());
dFy.setColumn(i: 3, calculateCoefb4());
dFy.setColumn(i: 4, calculateCoefb5());
dFy.setColumn(i: 5, calculateCoefb6());

```

**Εικόνα 49- Υπολογισμός πινάκων 'dFx' και 'dFy' της κλάσης 'SpaceResection'**

Ο πίνακας 'A' 'γεμίζει' με μια συνθήκη 'for' όπου λειτουργεί με τη λογική ότι αν το 'i' που έχει αρχικοποιηθεί με τιμή '0' είναι μικρότερο από τον αριθμό των σημείων επί 2 τότε να ξεκινήσει να λαμβάνει τιμές ο πίνακας. Αν το 'i' είναι ζυγό ή '0' να εισέλθουν τιμές από τον πίνακα 'dFx'. Αν το 'i' είναι μονός αριθμός τότε να εισέρχονται τιμές από τον 'dFy'. Ακολουθεί η δημιουργία του πίνακα 'N' (ανάστροφος του 'A' επί τον

'A'), των διανυσμάτων 'Fx', 'Fy' ( $\frac{A1}{P}*(-C)$  ή  $\frac{A2}{P}*(-C)$ ). Το διάνυσμα 'L' 'γεμίζει' από τις διαφορές των εικονοσυντεταγμένων με τα αντίστοιχα στοιχεία των διανυσμάτων 'Fx', 'Fy'.

```
// fill A matrix
for(int i = 0; i < 2 * numOfPoints; i++) {
    if(i % 2 == 0) A.setRow(i, dFx.getRow(i / 2));
    else A.setRow(i, dFy.getRow(i / 2));
}

N = A.transpose().multiply(A);

RealVector Fx = A1.ebeDivide(P).mapMultiply(-C);
RealVector Fy = A2.ebeDivide(P).mapMultiply(-C);

RealVector L = new ArrayRealVector( size: 2 * numOfPoints);
for (int i = 0; i < numOfPoints; i++) {
    L.setEntry(i * 2, v: pixels.getEntry(i,X) - Fx.getEntry(i));
    L.setEntry(i * 2 + 1, v: pixels.getEntry(i,Y) - Fy.getEntry(i));
}
```

**Εικόνα 50- Υπολογισμός πινάκων 'A', 'N', 'Fx', 'Fy' και 'L' της κλάσης 'SpaceResection'**

Καταληκτικά δημιουργήθηκε το διάνυσμα 'dx' ( $dx = N1*AT*dl$ ). Στην περίπτωση που τουλάχιστον ένα στοιχείο του 'dx' είναι μεγαλύτερο από το όριο που έχει τεθεί στην αρχή του κώδικα τότε η επίλυση ακολουθεί τη γνωστή διαδικασία των διαδοχικών προσεγγίσεων, δεδομένου ότι από κάθε επίλυση του συστήματος προκύπτουν τιμές για τις διαφορικές μεταβολές των προσωρινών τιμών, ώστε αυτές προστιθέμενες να πλησιάσουν την καλύτερη τιμή. Σε αντίθετη περίπτωση, η οπισθοτομία έχει ολοκληρωθεί και τα  $X_0, Y_0, Z_0, \omega, \phi, \kappa$  διαθέτουν τιμές. Έχει προσεγγιστεί και ο πίνακας υπολοίπων ( $U = A*X - dl$ ) για τον υπολογισμό της ακρίβειας των αποτελεσμάτων.

```
// check if algorithm converged
if ((dx.getEntry(i 0) < LINEAR_TOLERANCE) && (dx.getEntry(i 1) < LINEAR_TOLERANCE) && (dx.getEntry(i 2) < LINEAR_TOLERANCE) &&
(dx.getEntry(i 3) < ANGLE_TOLERANCE) && (dx.getEntry(i 4) < ANGLE_TOLERANCE) && (dx.getEntry(i 5) < ANGLE_TOLERANCE)) {
    break;
}

// update values for next iteration
this.Xo += dx.getEntry(i 0);
this.Yo += dx.getEntry(i 1);
this.Zo += dx.getEntry(i 2);
this.omega += dx.getEntry(i 3) / 10000;
this.phi += dx.getEntry(i 4) / 10000;
this.kappa += dx.getEntry(i 5) / 10000;

this.U = A.operate(dx).subtract(L);
```

**Εικόνα 51- Έλεγχος συνθηκών και προσδιορισμός νέων ή τελικών τιμών της κλάσης 'SpaceResection'**

### 4.4.3 Κλάση Εμπροσθοτομίας

Με την ίδια λογική ξεκινάει και η κλάση της εμπροσθοτομίας. Στην αρχή δηλώνονται όλες οι απαραίτητες μεταβλητές και σταθερές. Έχοντας ως δεδομένα τα στοιχεία του εξωτερικού προσανατολισμού των δύο εικόνων που θα χρησιμοποιηθούν για την εμπροσθοτομία μπορούν να υπολογιστούν οι πίνακες στροφής τους.

```
private void initialize() {
    this.leftR = calculateRotationMatrix(positionsAndOrientations.get(LEFT_IMAGE));
    this.rightR = calculateRotationMatrix(positionsAndOrientations.get(RIGHT_IMAGE));

    RealVector leftXs = leftR.transpose().operate(pickedPoints.get(LEFT_IMAGE));
    RealVector rightXs = rightR.transpose().operate(pickedPoints.get(RIGHT_IMAGE));

    this.ZoA = (((positionsAndOrientations.get(RIGHT_IMAGE).getXo() - positionsAndOrientations.get(LEFT_IMAGE).getXo()) * (-leftXs.getEntry(0)) +
    this.XoA = positionsAndOrientations.get(LEFT_IMAGE).getXo() - ((leftXs.getEntry(0)) * (ZoA - positionsAndOrientations.get(LEFT_IMAGE).getZoC()
    this.YoA = positionsAndOrientations.get(LEFT_IMAGE).getYo() - ((leftXs.getEntry(1)) * (ZoA - positionsAndOrientations.get(LEFT_IMAGE).getZoC()
}

private RealMatrix calculateRotationMatrix(PositionAndOrientation positionAndOrientation) {
    double omega = positionAndOrientation.getOmega();
    double phi = positionAndOrientation.getPhi();
    double kappa = positionAndOrientation.getKappa();

    double R[][] = new double[3][3];

    R[0][0] = Math.cos(Math.PI / 200 * phi) * Math.cos(Math.PI / 200 * kappa);
    R[0][1] = Math.sin(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Math.cos(Math.PI / 200 * kappa) + Math.cos(Math.PI / 200 * omega) *
    R[0][2] = -Math.cos(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Math.cos(Math.PI / 200 * kappa) + Math.sin(Math.PI / 200 * omega) *
    R[1][0] = -Math.cos(Math.PI / 200 * phi) * Math.sin(Math.PI / 200 * kappa);
    R[1][1] = -Math.sin(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Math.sin(Math.PI / 200 * kappa) + Math.cos(Math.PI / 200 * omega) *
    R[1][2] = Math.cos(Math.PI / 200 * omega) * Math.sin(Math.PI / 200 * phi) * Math.sin(Math.PI / 200 * kappa) + Math.sin(Math.PI / 200 * omega) *
    R[2][0] = Math.sin(Math.PI / 200 * phi);
    R[2][1] = -Math.sin(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi);
    R[2][2] = Math.cos(Math.PI / 200 * omega) * Math.cos(Math.PI / 200 * phi);

    return MatrixUtils.createRealMatrix(R);
}
```

Εικόνα 52- Υπολογισμός πινάκων στροφής για την αριστερή και δεξιά φωτογραφία της κλάσης 'SpacelIntersection'

Επόμενο βήμα είναι η προσέγγιση των διορθώσεων των αρχικών τιμών των φωτογραμμετρικών συντεταγμένων ξεχωριστά για την αριστερή και τη δεξιά φωτογραφία και υπολογισμός του πίνακα 'A'.

```
int iterations = 1;
while (iterations < MAX_ITERATIONS) {
    // Point on left image
    leftU = leftR.getEntry(0, 0) * (XoA - positionsAndOrientations.get(LEFT_IMAGE).getXo()) + leftR.getEntry(0, 1) * (YoA - positionsAndOrientations.get(L
    leftV = leftR.getEntry(1, 0) * (XoA - positionsAndOrientations.get(LEFT_IMAGE).getXo()) + leftR.getEntry(1, 1) * (YoA - positionsAndOrientations.get(L
    leftW = leftR.getEntry(2, 0) * (XoA - positionsAndOrientations.get(LEFT_IMAGE).getXo()) + leftR.getEntry(2, 1) * (YoA - positionsAndOrientations.get(L

    // Point on right image
    rightU = rightR.getEntry(0, 0) * (XoA - positionsAndOrientations.get(RIGHT_IMAGE).getXo()) + rightR.getEntry(0, 1) * (YoA - positionsAndOrientations.g
    rightV = rightR.getEntry(1, 0) * (XoA - positionsAndOrientations.get(RIGHT_IMAGE).getXo()) + rightR.getEntry(1, 1) * (YoA - positionsAndOrientations.g
    rightW = rightR.getEntry(2, 0) * (XoA - positionsAndOrientations.get(RIGHT_IMAGE).getXo()) + rightR.getEntry(2, 1) * (YoA - positionsAndOrientations.g
}
```

Εικόνα 53- Υπολογισμός προσεγγιστικών τιμών των X,Y,Z του ίδιου σημείου για την αριστερή και δεξιά φωτογραφία της κλάσης 'SpacelIntersection'

```

// Calculate A matrix
A.setEntry( R 0, R 0, W -focalLength * ((leftR.getEntry( R 0, R 0) / leftW) - (leftR.getEntry( R 2, R 0) * (leftU / Math.pow(leftW, 2)))));
A.setEntry( R 0, R 1, W -focalLength * ((leftR.getEntry( R 0, R 1) / leftW) - (leftR.getEntry( R 2, R 1) * (leftU / Math.pow(leftW, 2)))));
A.setEntry( R 0, R 2, W -focalLength * ((leftR.getEntry( R 0, R 2) / leftW) - (leftR.getEntry( R 2, R 2) * (leftU / Math.pow(leftW, 2)))));
A.setEntry( R 1, R 0, W -focalLength * ((leftR.getEntry( R 1, R 0) / leftW) - (leftR.getEntry( R 2, R 0) * (leftV / Math.pow(leftW, 2)))));
A.setEntry( R 1, R 1, W -focalLength * ((leftR.getEntry( R 1, R 1) / leftW) - (leftR.getEntry( R 2, R 1) * (leftV / Math.pow(leftW, 2)))));
A.setEntry( R 1, R 2, W -focalLength * ((leftR.getEntry( R 1, R 2) / leftW) - (leftR.getEntry( R 2, R 2) * (leftV / Math.pow(leftW, 2)))));
A.setEntry( R 2, R 0, W -focalLength * ((rightR.getEntry( R 0, R 0) / rightW) - (rightR.getEntry( R 2, R 0) * (rightU / Math.pow(rightW, 2)))));
A.setEntry( R 2, R 1, W -focalLength * ((rightR.getEntry( R 0, R 1) / rightW) - (rightR.getEntry( R 2, R 1) * (rightU / Math.pow(rightW, 2)))));
A.setEntry( R 2, R 2, W -focalLength * ((rightR.getEntry( R 0, R 2) / rightW) - (rightR.getEntry( R 2, R 2) * (rightU / Math.pow(rightW, 2)))));
A.setEntry( R 3, R 0, W -focalLength * ((rightR.getEntry( R 1, R 0) / rightW) - (rightR.getEntry( R 2, R 0) * (rightV / Math.pow(rightW, 2)))));
A.setEntry( R 3, R 1, W -focalLength * ((rightR.getEntry( R 1, R 1) / rightW) - (rightR.getEntry( R 2, R 1) * (rightV / Math.pow(rightW, 2)))));
A.setEntry( R 3, R 2, W -focalLength * ((rightR.getEntry( R 1, R 2) / rightW) - (rightR.getEntry( R 2, R 2) * (rightV / Math.pow(rightW, 2)))));

```

**Εικόνα 54- Υπολογισμός πίνακα 'A' της κλάσης 'SpacelIntersection'**

Ακολουθεί ένας νέος υπολογισμός τιμών για τη σωστή προσέγγιση των συντεταγμένων του άγνωστου σημείου. Αφού έχουν εκτιμηθεί όλα τα παραπάνω μπορεί να δημιουργηθεί ο πίνακας 'dl', ο πίνακας 'N' (  $AT * A$  ), ο πίνακας 'U' ( $AT * dl$ ).

```

leftExoA = -focalLength * ((leftR.getEntry( R 0, R 0) * (XoA - positionsAndOrientations.get(LEFT_IMAGE).getXo()) + leftR.getEntry( R 0, R 1) *
leftFyoA = -focalLength * ((leftR.getEntry( R 1, R 0) * (XoA - positionsAndOrientations.get(LEFT_IMAGE).getXo()) + leftR.getEntry( R 1, R 1) *
rightExoA = -focalLength * ((rightR.getEntry( R 0, R 0) * (XoA - positionsAndOrientations.get(RIGHT_IMAGE).getXo()) + rightR.getEntry( R 0, R 1) *
rightFyoA = -focalLength * ((rightR.getEntry( R 1, R 0) * (XoA - positionsAndOrientations.get(RIGHT_IMAGE).getXo()) + rightR.getEntry( R 1, R 1) *

// Calculate dl matrix
dl.setEntry( R 0, W pickedPoints.get(LEFT_IMAGE).getEntry(X) - leftExoA);
dl.setEntry( R 1, W pickedPoints.get(LEFT_IMAGE).getEntry(Y) - leftFyoA);
dl.setEntry( R 2, W pickedPoints.get(RIGHT_IMAGE).getEntry(X) - rightExoA);
dl.setEntry( R 3, W pickedPoints.get(RIGHT_IMAGE).getEntry(Y) - rightFyoA);

N = A.transpose().multiply(A);
U = A.transpose().operate(dl);

```

**Εικόνα 55- Υπολογισμός πινάκων 'dl', 'N', 'U' της κλάσης 'SpacelIntersection'**

Καταληκτικά, υπολογίζεται ο πίνακας διαφορών ('dx') και γίνεται έλεγχος των απαιτούμενων ορίων. Αφού επαναληφθεί ο κώδικας όσες φορές επιβάλλεται για να επιτευχθεί η σύγκλιση παρουσιάζονται οι φωτογραμμετρικές συντεταγμένες του σημείου που έχει επιλέξει ο χρήστης.

```

dx = MatrixUtils.inverse(N).operate(U);

// check if algorithm converged
if ((Math.abs(dx.getEntry(X)) < TOLERANCE) && (Math.abs(dx.getEntry(Y)) < TOLERANCE) && (Math.abs(dx.getEntry(Z)) <
    break;
}

// update values for next iteration
XoA += dx.getEntry(X);
YoA += dx.getEntry(Y);
ZoA += dx.getEntry(Z);

iterations++;

```

**Εικόνα 56- Υπολογισμός πίνακα 'dx', συνθήκης επανάληψης και τελικών τιμών της κλάσης 'SpacelIntersection'**

#### 4.4.4 Κλάση 'UserSession'

Η κλάση UserSession σχεδιάστηκε με βάση το σχεδιαστικό πρότυπο 'Singleton'. Αυτό το πρότυπο εξασφαλίζει πως από αυτή την κλάση δε θα φτιαχτεί παραπάνω από ένα αντικείμενο. Με αυτό τον τρόπο εξασφαλίζεται ότι δε θα υπάρξει 'UserSession1' ή 'UserSession2' κ.τ.λ. Ο λόγος ύπαρξης αυτής της κλάσης είναι ότι κρατάει πληροφορίες ενώ η εφαρμογή 'τρέχει' αδιάκοπα. Ουσιαστικά αποτελεί ένα

κεντρικό σημείο αναφοράς. Η 'UserSession' κρατάει πληροφορίες για να μπορούν εύκολα να διανεμηθούν σε όποιο σημείο χρειαστεί το πρόγραμμα για να συνεχίσει να εκτελείται.

```
public void setFocalLength(double focalLength) { this.focalLength = focalLength; }

public void addGroundControlPoint(String image, GroundControlPoint gcp) {
    selectedImages.get(image).getGroundControlPoints().add(gcp);
}

public void addPixelPoint(String image, PixelPoint pixel) {
    selectedImages.get(image).getPixelPoints().push(pixel);
}

public void removePixelPoint(String image) { selectedImages.get(image).getPixelPoints().pop(); }

public int getPointsAdded(String image) {
    return selectedImages.get(image).getTotalPointsProvided();
}
```

Εικόνα 57- Μέρος του κώδικα της κλάσης 'UserSession'

#### 4.4.5 Κλάσεις Presenter

Ο τρόπος με τον οποίο λειτουργεί μια εφαρμογή για Android συσκευές είναι περίπλοκος. Όπως έχει αναφερθεί, εκτός από το γραφικό περιβάλλον που εμφανίζεται στην οθόνη του smartphone, υπάρχει και το λειτουργικό περιβάλλον. Οι διάφορες οθόνες που μπορεί να δει ένας χρήστης κατά τη διάρκεια εκτέλεσης της εφαρμογής ονομάζονται 'activities'. Οι 'activities' για να λειτουργήσουν πρέπει να αποκτήσουν δεδομένα τα οποία είναι αποθηκευμένα στη 'UserSession'. Αυτή η μεταφορά δεδομένων, καθώς και οι εντολές για το τι θα εμφανιστεί στην οθόνη επιτυγχάνεται από τις κλάσεις 'Presenter'. Στην εφαρμογή 'Photogrammetry' υπάρχουν τρεις 'activities'. Οι 'ResectionActivity' και 'IntersectionActivity' έχουν αντίστοιχα τις 'ResectionActivityPresenter' και 'IntersectionActivityPresenter'.

```
public class ResectionActivityPresenter {

    private UserSession userSession;
    private String image;
    private ResectionActivity view;
    private double focalLength;

    public ResectionActivityPresenter(ResectionActivity view, String image) {
        this.userSession = UserSession.getInstance();
        this.image = image;
        this.view = view;
    }
}
```

Εικόνα 58- Δηλώσεις της κλάσης 'ResectionActivityPresenter'

Όταν ο χρήστης κάνει κάτι στην οθόνη που του εμφανίζεται η αντίστοιχη 'activity' ενημερώνει την 'presenter' για την κίνηση αυτή. Η 'presenter' με τη σειρά της δίνει εντολή στην 'activity' για το πώς θα συνεχίσει τη ροή του προγράμματος. Αν το επόμενο βήμα του προγράμματος χρίζει ελέγχου των δεδομένων που έχουν συγκεντρωθεί τότε η 'presenter' καλεί τα δεδομένα από τη 'UserSession' για να τα ελέγξει. Ανάλογα από το αποτέλεσμα δίνεται η κατάλληλη εντολή.

```
public void userDrewPoint() {  
    view.disableDrawOnImage();  
    view.showConfirmButton();  
    view.hideCalculateButton();  
    view.showJoystick();  
}
```

Εικόνα 59- Μέρος του κώδικα της κλάσης 'ResectionActivityPresenter'

```
public void checkConditions() {  
    // Check for minimum points needed  
    if (userSession.getPointsAdded(image) >= SpaceResection.MINIMUM_POINTS_REQUIRED) {  
        view.showCalculateButton();  
    }  
}
```

Εικόνα 60- Μέρος του κώδικα της κλάσης 'ResectionActivityPresenter'

#### 4.4.6 Κλάσεις Activities

Οι 'activities' της εφαρμογής 'Photogrammetry' είναι η 'MainActivity', η 'ResectionActivity' και η 'IntersectionActivity'. Η 'MainActivity' δεν έχει δικιά της 'presenter' καθώς δεν έχει κάποιο λόγο να 'επικοινωνήσει' με τη 'UserSession'. Η χρήση της είναι απλή. Είναι η πρώτη οθόνη που εμφανίζεται κατά την εκκίνηση της εφαρμογής και διαθέτει δύο κουμπιά για την επιλογή εκτέλεσης οπισθοτομίας ή εμπροσθοτομίας. Ανάλογα με το τι επιλέξει ο χρήστης καλείται να ενεργοποιηθεί η αντίστοιχη 'activity' και το πρόγραμμα οδηγείται στη 'συλλογή' φωτογραφιών του smartphone ώστε να επιλεγθεί με ποια εικόνα θα 'δουλέψει'. Η 'ResectionActivity' και η 'IntersectionActivity' διαθέτουν τα κατάλληλα εργαλεία ώστε να συγκεντρωθούν τα δεδομένα που χρειάζονται να καταχωρηθούν.

```

private ResectionActivityPresenter presenter;
private ImageView photoImageView;
private ProgressBar progressBar;
private Button calculateButton, confirmButton;
private JoystickView joystick;
private EditText xEditText, yEditText, zEditText;

private View.OnTouchListener imageListener;

private float touchX, touchY;

private Bitmap lastImageVersion;

private Uri imageUri;

```

Εικόνα 61- 'Δηλώσεις' της κλάσης 'ResectionActivity'

Εκτός από τα κλασικά κουμπιά, η συγκεκριμένη εφαρμογή διαθέτει 'joystick' που δίνει την επιλογή στο χρήστη να επιλέξει με μεγάλη ακρίβεια όποιο pixel της εικόνας χρειάζεται. Η εικόνα 'διαβάζεται' ως 'bitmap'. Πιο συγκεκριμένα, 'διαβάζεται' σαν πίνακας με τις τιμές χρώματος που αντιστοιχούν σε κάθε κελί. Με το πάτημα της οθόνης από το χρήστη δημιουργείται μία τελεία στην οθόνη. Είναι πολύ δύσκολο να πατηθεί με την πρώτη προσπάθεια το σωστό εικονοστοιχείο καθώς το δάχτυλο καλύπτει δεκάδες pixel ταυτόχρονα. Κάθε φορά που κινείται το joystick το πρόγραμμα υπολογίζει τη γωνία κατά την οποία έχει κατεύθυνση το δάχτυλο. Στα 'μάτια' του χρήστη φαίνεται ότι κινείται η τελεία. Στην πραγματικότητα όμως η εικόνα ζωγραφίζεται στην οθόνη με την τελεία και κάθε φορά που το 'joystick' κινείται η εικόνα επαναζωγραφίζεται με τη τελεία σε νέα θέση.



```
joystick.setOnMoveListener(new JoystickView.OnMoveLis
@Override
public void onMove(int angle, int strength) {

    // Determine direction based on joystick angl
    if (angle > 350 || angle < 10) {
        touchX += PIXEL_OFFSET;
    }
    else if (angle < 80) {
        touchX += PIXEL_OFFSET;
        touchY -= PIXEL_OFFSET;
    }
    else if (angle < 100) {
        touchY -= PIXEL_OFFSET;
    }
    else if (angle < 170) {
        touchX -= PIXEL_OFFSET;
        touchY -= PIXEL_OFFSET;
    }
    else if (angle < 190) {
        touchX -= PIXEL_OFFSET;
    }
    else if (angle < 260) {
        touchX -= PIXEL_OFFSET;
        touchY += PIXEL_OFFSET;
    }
    else if (angle < 280) {
        touchY += PIXEL_OFFSET;
    }
}
```

Εικόνα 62- Απόσπασμα Νο1 από τον κώδικα του 'joystick' της κλάσης 'ResectionActivity'

```
float dst[] = new float[2];
// Get the matrix of ImageView
Matrix imageMatrix = photoImageView.getImageMatrix();
// Create an inverse matrix
Matrix inverseMatrix = new Matrix();
// Inverse, the inverse matrix is assigned
imageMatrix.invert(inverseMatrix);
// Get the value of the target point dst through the inverse matrix mapping
inverseMatrix.mapPoints(dst, new float[]{touchX, touchY});

// Determine the position of touchX, touchY on the Bitmap
int bitmapX = Math.round(dst[0]);
int bitmapY = Math.round(dst[1]);

// Create a new image bitmap and attach a brand new canvas to it
Bitmap bitmap = lastImageVersion;
Bitmap tempBitmap = Bitmap.createBitmap(bitmap.getWidth(), bitmap.getHeight(), Bitmap.Config.ARGB_8888);
Canvas tempCanvas = new Canvas(tempBitmap);

// Create custom paint
Paint paint = new Paint(Paint.ANTI_ALIAS_FLAG);
paint.setColor(getResources().getColor(R.color.red));
```

Εικόνα 63- Απόσπασμα Νο2 από τον κώδικα του 'joystick' της κλάσης 'ResectionActivity'

Είναι σημαντικό να αναφερθεί η διαδικασία κατά την οποία συλλέγονται οι τιμές των εικονοστοιχείων. Όπως είναι γνωστό, το σύστημα εικονοσυντεταγμένων του κινητού έχει ως αρχή το πάνω αριστερά ριχελ της οθόνης. Για την επίλυση των φωτογραμμετρικών προβλημάτων έπρεπε να δημιουργηθεί ένα καρτεσιανό σύστημα

όπου το κεντρικό pixel αποτελεί την αρχή των αξόνων με συντεταγμένες (0,0). Με την επιλογή του pixel από τον χρήστη, αυτή η πληροφορία αναλύεται σε pixel(x) και pixel(y). Η εικονοσυντεταγμένη x βρίσκεται αφαιρώντας το πλάτος της εικόνας διά 2, ενώ αντίστοιχα για την εικονοσυντεταγμένη y πρέπει να αφαιρεθεί από το ύψος της εικόνας διά 2.

```
public void translateCoordinatesToCenter() {  
    touchX = touchX - photoImageView.getWidth() / 2;  
    touchY = photoImageView.getHeight() / 2 - touchY;  
}
```

Εικόνα 64- Μετατροπή εικονοσυντεταγμένων ως προς το κεντρικό pixel της κλάσης 'ResectionActivity'

Μετά την εύρεση των εικονοσυντεταγμένων πρέπει αυτές να μετατραπούν από pixel σε χιλιοστά. Αυτό επιτυγχάνεται πολλαπλασιάζοντας τις εικονοσυντεταγμένες με το μέγεθος του εικονοστοιχείου που 'δίνει' ο κατασκευαστής διά χίλια, αφού το pixelsize είναι σε μικρόμετρα.

```
touchX = (float) (touchX * 1.43) / 1000;  
touchY = (float) (touchY * 1.43) / 1000;
```

Εικόνα 65- Μετατροπή εικονοσυντεταγμένων από pixel σε mm της κλάσης 'ResectionActivity'

#### 4.5 Διαδικασία δοκιμής της εφαρμογής

Από τη στιγμή που η εφαρμογή ολοκληρώθηκε έπρεπε να βρεθεί τρόπος να εξεταστούν τα αποτελέσματά της. Για την επίτευξη της αξιολόγησης χρησιμοποιήθηκε το κέντρο Μετρολογίας του κτιρίου Λαμπαδαρίου. Στο κέντρο μετρολογίας έχει δημιουργηθεί ένα φωτογραμμετρικό σύστημα συντεταγμένων, με βάση το οποίο λειτούργησε και η εφαρμογή. Αφού δόθηκε η πρόσβαση στην αίθουσα φωτογραφήθηκαν τα 138 σημεία γνωστών συντεταγμένων από δύο διαφορετικά βάθρα. Με την κατοχή των φωτογραμμετρικών συντεταγμένων τόσο πολλών σημείων καθώς και των βάθρων από 'που τραβήχτηκαν οι φωτογραφίες μπορούν να προκύψουν ακριβή συμπεράσματα για την επιτυχία της εφαρμογής. Για τις ανάγκες της φωτογράφισης χρησιμοποιήθηκε ένα smartphone Samsung Galaxy Note 10+, ένα τρίποδο για να εξασφαλιστεί η σωστή θέση του τηλεφώνου κατά τη φωτογράφιση και μια μετροταινία για την εύρεση κάποιων προσεγγιστικών τιμών.

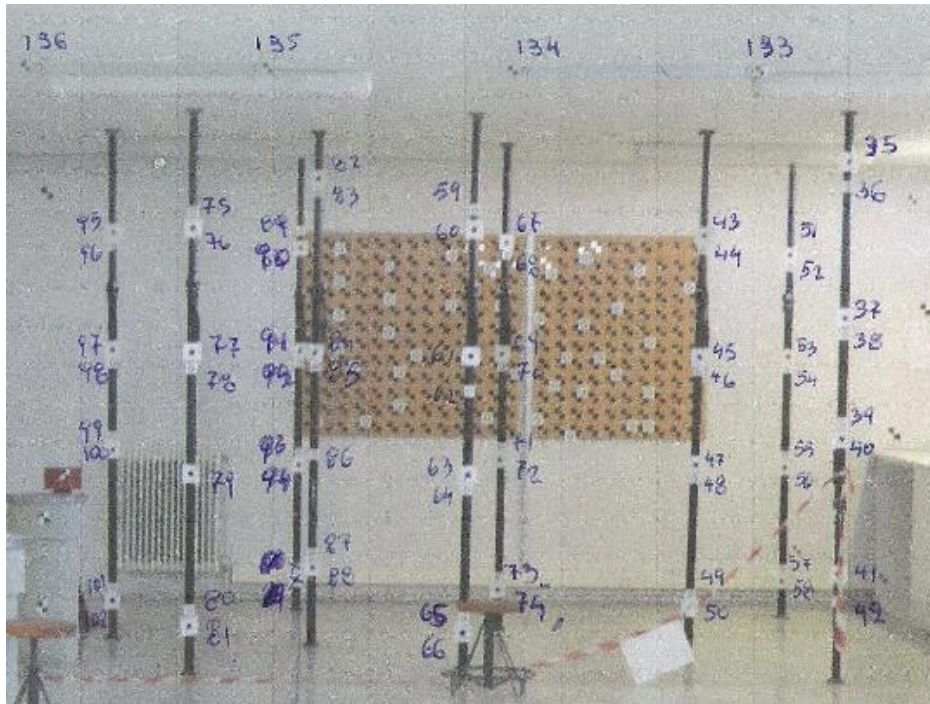


**Εικόνα 66- Λήψη φωτογραφιών**



**Εικόνα 67- Σημεία γνωστών φωτογραμμετρικών συντεταγμένων**

Για την εύρεση των συντεταγμένων του πεδίο ελέγχου χρησιμοποιήθηκε ένα σκαρίφημα με αρίθμηση των φωτοσταθερών καθώς και ένα αρχείο Word.



Εικόνα 68- Ένα κομμάτι του σκαριφήματος

	<b>ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ</b> ΣΧΟΛΗ ΑΓΡΟΝΟΜΩΝ ΚΑΙ ΤΟΠΟΓΡΑΦΩΝ ΜΗΧΑΝΙΚΩΝ ΤΟΜΕΑΣ ΤΟΠΟΓΡΑΦΙΑΣ ΚΕΝΤΡΟ ΜΕΤΡΟΛΟΓΙΑΣ					
	<b>ΗΜΕΡΟΜΗΝΙΑ ΜΕΤΡΗΣΗΣ</b>	<b>15 ΜΑΡΤΙΟΥ 2014</b>				
<b>ΟΡΓΑΝΟ ΜΕΤΡΗΣΗΣ</b>	<b>TDM 5005</b>					
<b>ΣΥΝΤΕΤΑΓΜΕΝΕΣ ΦΩΤΟΣΤΑΘΕΡΩΝ στο φωτογραμμετρικό σύστημα</b>						
Pt	$X\phi(m)$	$Y\phi(m)$	$Z\phi(m)$	$\sigma_x(mm)$	$\sigma_y(mm)$	$\sigma_z(mm)$
1	-21.0976	10.7257	-32.3838			
2	-21.0309	10.4946	-32.3990			
3	-21.0320	10.1921	-32.4004			
4	-21.3287	10.0948	-32.3469			
5	-21.3293	9.8952	-32.3479			
6	-21.4283	9.8951	-32.3309			
7	-21.5245	10.1960	-32.3136			
8	-21.5224	10.5965	-32.3107			
9	-21.7187	10.8979	-32.2767			

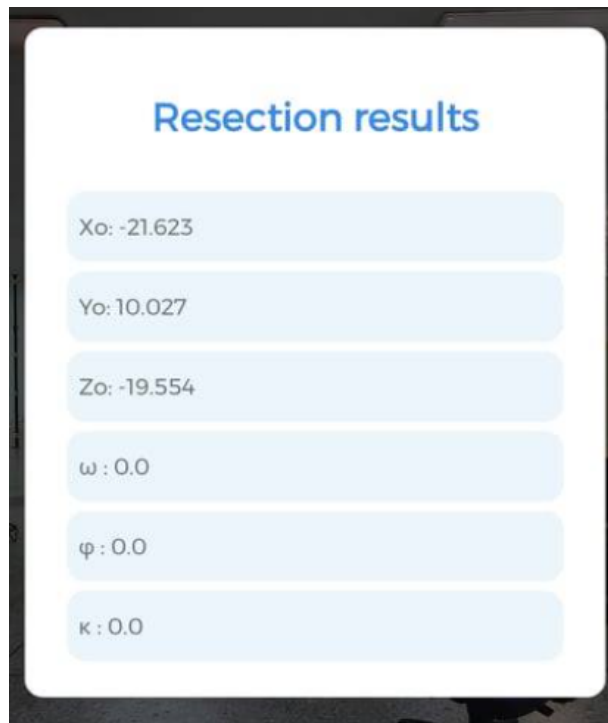
Εικόνα 69- 9 από τα 138 σημεία του αρχείου Word

Αναγκαία δεδομένα για την ομαλή λειτουργία της εφαρμογής αποτελούν η εστιακή απόσταση και το μέγεθος pixel του smartphone. Αυτά τα δεδομένα συλλέχθηκαν από τις πληροφορίες που είναι διαθέσιμες από τον κατασκευαστή. Συγκεκριμένα, από υπολογισμούς βρέθηκε το μέγεθος pixel και από τις λεπτομέρειες της φωτογραφίας βρέθηκε η εστιακή απόσταση. Αφού ολοκληρώθηκε η συλλογή των απαραίτητων δεδομένων 'έτρεξε' για πρώτη φορά η εφαρμογή. Τα αποτελέσματα στην αρχή δεν ήταν ενθαρρυντικά καθώς υπήρχαν αρκετά μεγάλες αποκλίσεις κυρίως για την προσέγγιση του  $Z_0$  για την οπισθοτομία αλλά και του  $Z_0A$  για την εμπροσθοτομία. Μετά από υποδείξεις του επιβλέποντα καθηγητή όσον αφορά τη διόρθωση του κώδικα αλλά και τη σωστή χρήση των αποκτηθέντων δεδομένων προέκυψαν τα επιθυμητά αποτελέσματα. Βασική προϋπόθεση για την επίτευξη 'σωστών'

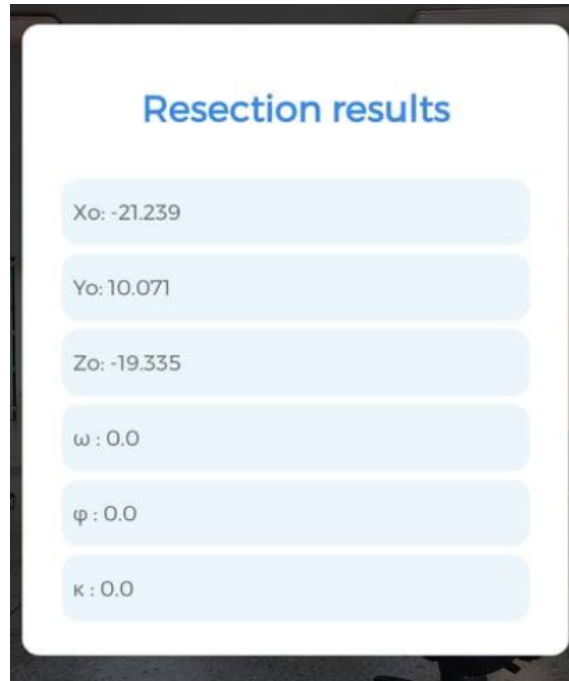
αποτελεσμάτων στην οπισθοτομία αποδείχθηκε πως είναι η επιλογή κατάλληλα κατανομημένων φωτοσταθερών ώστε να καλύπτεται όσο το δυνατόν μεγαλύτερο μέρος της φωτογραφίας.

#### 4.6 Αποτελέσματα της εφαρμογής

Για την ολοκλήρωση της εφαρμογής και την εμφάνιση επιθυμητών αποτελεσμάτων χρησιμοποιήθηκε μεγάλο πλήθος μετρήσεων. Στόχος αποτέλεσε να επιτευχθεί μια επαρκής εσωτερική ακρίβεια αλλά και μια επαρκής εξωτερική ακρίβεια. Εσωτερική ακρίβεια ονομάζεται η συνάφεια των μετρήσεων μεταξύ τους ενώ εξωτερική ακρίβεια θεωρείται η συνάφεια των μετρήσεων με τα αποτελέσματα των μετρήσεων που έχουν παραχθεί από άλλες μεθόδους. Η προσέγγιση των  $X_0$ ,  $Y_0$ ,  $Z_0$  για την οπισθοτομία αλλά και των  $X$ ,  $Y$ ,  $Z$  για την εμπροσθοτομία φτάνει να διαθέτει εσωτερική και εξωτερική ακρίβεια  $\pm 1m$ . Οι φωτογραμμετρικές συντεταγμένες του δεξιού βάθρου είναι  $X\delta = -21.8057\mu$ ,  $Y\delta = 9.82\mu$ ,  $Z\delta = -19,7800\mu$  ενώ του αριστερού βάθρου είναι  $X\alpha = -23.2588\mu$ ,  $Y\alpha = 10.18$ ,  $Z\alpha = -19.89\mu$ . Στο  $Y\delta$  έχουν προστεθεί 49εκ. και στο  $Y\alpha$  85 εκ. καθώς τόσο ήταν το ύψος του τριπόδου κατά τη φωτογράφιση. Παρακάτω παρατίθενται αποτελέσματα της οπισθοτομίας από δύο σετ φωτοσταθερών για τη φωτογραφία από το δεξί βάθρο και αποτελέσματα από δύο εμπροσθοτομίες. Το σημείο για το οποίο εκτελέστηκε η εμπροσθοτομία έχει συντεταγμένες  $X136 = -23.3571$ ,  $Y136 = 11.4143$  και  $Z136 = -25.8444$  (μετρημένες από Total Station).



Εικόνα 70- Αποτελέσματα 1ου σετ φωτοσταθερών του δεξιού βάθρου



Εικόνα 71- Αποτελέσματα 2ου σετ φωτοσταθερών του δεξιού βάρθρου



Εικόνα 72- Αποτελέσματα 1ου σετ φωτοσταθερών του αριστερού βάρθρου



Εικόνα 73- Αποτελέσματα 2ου σετ φωτοσταθερών του αριστερού βάρου



Εικόνα 74- Αποτελέσματα εμπροσθοτομίας για την εύρεση του σημείου 136



# ΚΕΦΑΛΑΙΟ 5

## Συμπερασματικά σχόλια

### 5.1 Αξιολόγηση και έλεγχοι

Αν αναλογιστεί κανείς πως η εφαρμογή λειτουργεί με βάση τις φωτογραφίες μιας ερασιτεχνικής φωτογραφικής μηχανής, τα σημεία γνωστών φωτογραμμετρικών συντεταγμένων επιλέγονται χειροκίνητα από το χρήστη και το γεγονός ότι η Java ως γλώσσα προγραμματισμού δεν είναι η ιδανική για την επίλυση πολύπλοκων μαθηματικών προβλημάτων είναι λογικό να μην προκύπτει ακρίβεια χιλιοστών. Το γεγονός όμως επίτευξης αποτελεσμάτων με ακρίβεια του ενός μέτρου αποτελεί επιτυχία καθώς εκπληρώνει το σκοπό ύπαρξης της εφαρμογής δηλαδή την ύπαρξη ενός εύχρηστου, γρήγορου και αποτελεσματικού εργαλείου για έναν μηχανικό του χώρου.

### 5.2 Μελλοντικές βελτιώσεις και προσθήκες

Η ραγδαία ανάπτυξη της τεχνολογίας δημιουργεί την εντύπωση πως τα smartphones θα εξελιχθούν ακόμα περισσότερο. Μια ενδεχόμενη βελτίωση της ποιότητας της φωτογραφικής μηχανής των 'έξυπνων' τηλεφώνων θα μπορούσε να βελτιστοποιήσει την ακρίβεια των αποτελεσμάτων. Όπως αναφέρθηκε και σε πολλές μελέτες πρέπει να βελτιωθούν επίσης τα επιταχυνσιόμετρα, τα γυροσκόπια, τα μαγνητόμετρα καθώς και η αξιοπιστία των δεκτών GNSS που είναι εξοπλισμένα. Πιθανή προσθήκη που θα μπορούσε να αναβαθμίσει την ποιότητα της εφαρμογής 'Photogrammetry' είναι η ένταξη σε αυτήν ενός αλγορίθμου για αυτόματο εντοπισμό των σημείων γνωστών φωτογραμμετρικών συντεταγμένων ώστε να αποφευχθεί η χειροκίνητη επιλογή τους. Χρήσιμη μπορεί να φανεί επίσης η ένταξη της δυνατότητας ο χρήστης να φωτογραφίζει κατευθείαν κατά την εκκίνηση της εφαρμογής ή η αυτόματη αποθήκευση των αποτελεσμάτων ή και η δημιουργία '.txt' αρχείου το οποίο να είναι συμβατό με κάποιο πρόγραμμα σχεδίασης. Καταληκτικά, σημαντική επέκταση μπορεί να αποτελέσει η ένταξη του υπολογισμού των ακριβειών καθώς και η σύνδεση των αποτελεσμάτων της οπισθοτομίας με τις αρχικές τιμές της εμπροσθοτομίας.



## Βιβλιογραφία

### Ελληνόγλωσση βιβλιογραφία

- Α. Γεωργόπουλος (2021)- Διαφάνειες διαλέξεων Φωτογραμμετρίας 1
- Α. Γεωργόπουλος (2021)- Διαφάνειες διαλέξεων Φωτογραμμετρίας 2
- Εκφώνηση άσκησης 3 του μαθήματος Φωτογραμμετρίας 1
- Εκφώνηση άσκησης 5 του μαθήματος Φωτογραμμετρίας 1
- Πέτσα Έ. (2000)- Έννοιες και Θεμελιώδη Προβλήματα της Φωτογραμμετρίας - Πρώτη Έκδοση

### Μεταφρασμένη βιβλιογραφία

- Karl Kraus (2010)- Βασικές έννοιες και μέθοδοι- Με τη συμβολή του καθηγητή Δρος P.WALDWHAUSSL- Ελεύθερη μετάφραση από το Γερμανικό πρωτότυπο στα Ελληνικά : Δρ Μηχανικός Ευάγγελος Βοζίκης με τη συμβολή του καθηγητή Δρος Ανδρέα Γεωργόπουλου-Εκδόθηκε από το Τεχνικό Επιμελητήριο Ελλάδας

### Ξενόγλωσση βιβλιογραφία

- Bakula and Flasiński (2014) – ‘Capabilities of a smartphone for georeferenced 3D model creation: an evaluation’
- Chen Yi and Lu Jue (2008)- ‘Performing space resection using total least squares’
- Fabio Poiesi and Paul Chippendale (2018) –‘3D reconstruction in your pocket’ - Mathematics for Computer Vision (MCV 2018) February 15-16th 2018, Trento, ITALY Demo Session 2: 3D Reconstruction
- Fabio Poiesi, Alex Locher, Paul Chippendale, Luc Van Gool, Fabio Remondino, Erica Nocerino (2017)- Cloud-based collaborative 3D reconstruction using smartphones - CVMP 2017, December 11–13, 2017, London, United Kingdom
- Francesca Fissore, Andrea Masiero, Marco Piragnolo, Francesco Pirotti, Alberto Guarnieri, Antonio Vettore (2018)- ‘Towards Surveying with a Smartphone’
- Hohyun Jeong, Hoyong Ahn, Dongyoon Shin, Yushin Ahn, Chuluong Choi (2019) – ‘A Comparative Assessment of the Photogrammetric Accuracy of Mapping Using UAVs with Smart Devices’ - Photogrammetric Engineering & Remote Sensing Vol. 85, No. 12, December 2019, 2019 American Society for Photogrammetry and Remote Sensing
- Marion Jaud, Matthieu Kervot, Christophe Delacourt, Stéphane Bertin (2019)- ‘Potential of Smartphone SfM Photogrammetry to Measure Coastal Morphodynamics’ Published: 26 September 2019

- Masaad and Omer (2014)- ‘Use of Scanned Aerial Images in Photogrammetric-Resection and Intersection’
- Melanie ELIAS, Christian KEHL, Danilo SCHNEIDER (2019) ‘Photogrammetric water level determination using smartphone technology’
- Melanie KIRCHHÖFER, Jim CHANDLER, Rene WACKROW (2011)- ‘Cultural heritage recording utilizing low-cost close range photogrammetry’
- Natan Micheletti, Jim H. Chandler, Stuart N. Lane (2014) – ‘Investigating the geomorphological potential of freely available and accessible structure-from-motion photogrammetry using a smartphone’
- Olaf Hellwich, Tomasz Bien, Timo Krueger, Dirk Mucha (2016)- ‘Patient registration using photogrammetric surface reconstruction from smartphone imagery’
- Oleg Muratov, Yury Slynko, Vitaly Chernov, Maria Lyubimtseva, Artem Shamsuarov, Victor Bucha, Samsung R&D Institute RUS (2016)- ‘3DCapture: 3D Reconstruction for a Smartphone’
- Paolo Dabove, Nives Grasso, Marco Piras (2019) – ‘Smartphone-Based Photogrammetry for the 3D Modeling of a Geomorphological Structure’
- Roman Shults (2017)- ‘New opportunities of low-cost photogrammetry for culture heritage preservation’ The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-5/W1, 2017 GEOMATICS & RESTORATION – Conservation of Cultural Heritage in the Digital Era, 22–24 May 2017, Florence, Italy
- Wojnarowski (2019)- Photogrammetric technology for remote high-precision 3D monitoring of cracks and deformation joints of buildings and constructions

## Διαδικτυακά βοηθήματα

- [https://developer.android.com/about/versions/11?gclid=Cj0KCQjwub-HBhCyARIsAPctr7zrHH-zgAsejxPU309m\\_jFMitHmT2jnqqlosAv\\_cPQsKKI\\_J\\_YbegMaAgF\\_EALw\\_wcB&gclsrc=aw.ds](https://developer.android.com/about/versions/11?gclid=Cj0KCQjwub-HBhCyARIsAPctr7zrHH-zgAsejxPU309m_jFMitHmT2jnqqlosAv_cPQsKKI_J_YbegMaAgF_EALw_wcB&gclsrc=aw.ds)