|  |
| --- |
| Ατομική Διπλωματική Εργασία  **Αλγόριθμοι και Εφαρμογή για Δημιουργία χαρτών UAV (Drones)**  Νικόλας Χατζηγιάννης  **ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**  UCY-new-logo  **ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**  **Μάιος 2018** |

**ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ**

**Αλγόριθμοι και Εφαρμογή για την Δημιουργία Χαρτών UAV(Drones)**

Νικόλας Χατζηγιάννης

Επιβλέπων Καθηγητής

Βάσος Βασιλείου

Η Ατομική Διπλωματική Εργασία υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων απόκτησης του πτυχίου Πληροφορικής του Τμήματος Πληροφορικής του Πανεπιστημίου Κύπρου

Μάιος 2018

**Περίληψη**

Αυτή η πτυχιακή έρευνα, έχει μελετήσει τα πρόβλημα και δυσκολίες που αντιμετωπίζουν οι ομάδες δάσωσης και έρευνας και συνεπώς την υλοποίηση μιας εφαρμογής, η οποία θα συνενώνει εικόνες που προέρχονται από το Μη Επανδρωμένο Αεροσκάφος και θα έχει σαν έξοδο ένα χάρτη της περιοχής που σαρώθηκε. Μέσα από αυτή την μελέτη επινοήθηκε και ένας καινούργιος αλγόριθμος, ο Αλγόριθμος του Καμβά. Αυτός ο Αλγόριθμος χρησιμοποιεί σφαιρική τριγωνομετρία και αποκλιμάκωση για την τοποθέτηση των εικόνων σε ένα άδειο καμβά χωρίς την χρήση εννοιών που περιγράφονται στους Αλγορίθμους Συνένωσης μέσω σημείων-κλειδιών. Οι αλγόριθμοι και οι εφαρμογές είναι υλοποιημένοι σε γλώσσα προγραμματισμού Java και Python, ενώ οι γραφικές διαπροσωπίες δημιουργήθηκαν με την χρήση της βιβλιοθήκης Swing. Τέλος έχουν γίνει κάποια πειράματα για την σύγκριση της αποδοτικότητας των αλγόριθμων βάσει του όγκου των δεδομένων εισόδου, αλλά και την σύγκριση του αλγόριθμου που προτείνεται στην πτυχιακή εργασία, με τους υπόλοιπους. Τα αποτελέσματα αυτά μας οδηγούν στο συμπέρασμα πως κανένας αλγόριθμος δεν είναι ο καταλληλότερος εφόσον ο καθένας έχει μειονεκτήματα.

**Περιεχόμενα**

Κεφάλαιο **1 Εισαγωγή**…………...……………………………………..………...…… **1**

1.1 Πρόβλημα.............................................................................................. 1

1.2 Στόχος.................................................................................................... 2

1.3 Ερευνητικό Υπόβαθρο........................................................................... 3

1.3.1 Προηγούμενη Διπλωματική Εργασία...................................... 3

1.3.2 Ψηφιακή Εικόνα και η Επεξεργασία της................................. 4

1.3.3 Η Έννοια και Χρήση του Stitching.......................................... 7

1.3.4 Μαθηματικοί Υπολογισμοί και Αποστάσεις........................... 9

1.4 Μεθοδολογία....................................................................................... 11

1.5 Οργάνωση Διπλωματικής Εργασίας..................................................... 12

Κεφάλαιο **2 Εφαρμογή Υπολογισμού Αποτυπώματος Φωτογραφίας από Προκαθορισμένο Υψόμετρο** .................................................................... **14**

2.1 Εφαρμογή............................................................................................. 14

2.1.1 Ψευδοκώδικας Αλγορίθμου.................................................... 15

2.1.2 Λεπτομερής Περιγραφή Αλγορίθμου..................................... 15

2.1.3 Βοηθητική Εφαρμογή, Δεδομένα Εισόδου

και Εγκυρότητα τους.............................................................. 19

2.1.4 Δεδομένα Εξόδου και Αρχεία Αναφορών............................... 21

2.2 Επεξήγηση Γραφικής Διαπροσωπίας Χρήστη....................................... 22

Κεφάλαιο **3 Εφαρμογή Δημιουργίας Χάρτη από Φωτογραφίες UAV**…................... **26**

3.1 Εφαρμογή.............................................................................................. 26

3.1.1 Τύπος αντικειμένου BlackBox …......................................... 27

3.1.2 Ψευδοκώδικας Αλγορίθμου Καμβά........................................ 29

3.1.3 Λεπτομερής Περιγραφή Αλγορίθμου Καμβά......................... 30

3.1.4 Ψευδοκώδικας Αλγορίθμου Stitching.................................... 35

3.1.5 Λεπτομερής Περιγραφή Αλγορίθμου Stitching...................... 36

3.1.6 Βοηθητική Εφαρμογή, Δεδομένα Εισόδου

και Εγκυρότητα τους……………………………………… 38

3.1.7 Δεδομένα Εξόδου και Αρχεία Αναφορών............................... 41

3.2 Επεξήγηση Γραφικής Διαπροσωπίας Χρήστη....................................... 42

Κεφάλαιο **4 Πειραματική αξιολόγηση**…...................................................................... **45**

4.1 Σύγκριση Αλγορίθμων Δημιουργίας Χάρτη.......................................... 45

4.2 Εκτέλεση Αλγορίθμων για Σύγκριση Χρόνου εκτέλεσης...................... 45

4.3 Εκτέλεση Αλγορίθμων με Εικόνες Διαφορετικών Μεγεθών................. 48

4.4 Εκτέλεση Αλγορίθμων με Διαφορετική Ποσότητα Εικόνων................. 49

4.5 Εκτέλεση Αλγορίθμων σε Διαφορετικές Συσκευές............................... 51

4.6 Έλεγχος Μονάδων (JUNIT).................................................................. 52

Κεφάλαιο **5 Συμπεράσματα**…....................................................................................... **54**

5.1 Σύγκριση Αποτελεσμάτων των Πειραμάτων......................................... 54

5.2 Δυσκολίες που Αντιμετωπίστηκαν........................................................ 55

5.3 Μελλοντική Ερευνητική Εργασία......................................................... 55

**Βιβιλιογραφία**............................................................................................ **56**

Παράρτημα  **Α**….............................................................................................................. **60**

Α.1 Διάγραμμα Κλάσης Drone Algorithm.................................................. 60

A.2 Διάγραμμα Κλάσης Canvas Algorithm................................................. 61

A.3 Διάγραμμα Κλάσης Main App.............................................................. 62

Παράρτημα  **B**….............................................................................................................. **63**

B.1 Διαπροσωπία αντικειμένου Drone…................................................... 63

B.2 Διαπροσωπία αντικειμένου Area…...................................................... 67

B.3 Διαπροσωπία αντικειμένου BlackBox……………………................. 69

B.4 Διαπροσωπία αντικειμένου ImgRGB24............................................... 70

**Κεφάλαιο 1**

**Εισαγωγή**

1.1 Πρόβλημα 1

1.2 Στόχος 2

* 1. Ερευνητικό υπόβαθρο 3

1.3.1 Προηγούμενη Διπλωματική Εργασία 3

1.3.2 Ψηφιακή Εικόνα και η Επεξεργασία της 4

1.3.3 Η Έννοια και χρήση του Stitching 7

1.3.4 Μαθηματικοί Υπολογισμοί και Αποστάσεις. 9

* 1. Μεθοδολογία 11

1.5 Οργάνωση Διπλωματικής Εργασίας 12

**1.1 Πρόβλημα**

Σε ορισμένα περιστατικά όπως πυρκαγιές, ναυάγια ή φυσικές καταστροφές, οι ομάδες ερευνάς και διάσωσης έχουν να αντιμετωπίσουν παράγοντες όπως δύσβατες περιοχές ή περιοχές μεγάλων εκτάσεων. Σε τέτοια περιστατικά πολύ συχνά υπάρχουν απώλειές ανθρωπίνων ζωών ή εξάπλωση και αναζωογόνηση πυρκαγιών λόγω του ότι η εύρεση ή η εκτίμηση τους είναι χρονοβόρα, δύσκολη ή μη εφικτή λόγο των προαναφερθέντων.

Το έργο ομάδων έρευνας και διάσωσης ή ομάδων εκτίμησης ζημίων, που προκλήθηκαν από φυσικές καταστροφές και μη, είναι δύσκολο λόγο δύσβατων περιοχών. Το έργο αυτών των ομάδων γίνεται ακόμα πιο δύσκολο όταν λόγο έλλειψης προσωπικού πρέπει να καλυφθούν μεγάλες έκτασης και ή κούραση του προσωπικού καθώς και ο χρόνος περιορίζει της ενέργειες που μπορεί να εκτελεσθούν. Ο παράγοντες αυτοί μπορεί να καταστήσουν τις αποστολές αυτές πιο κρίσιμες όταν εμπλακεί και ο παράγοντας ανθρώπινες ζωές.

Συνοπτικά οι κύριοι παράγοντες που θεωρούνται κρίσιμη και υπαίτιοι για το πρόβλημα αυτό είναι ο χρόνος, η δυσκολία ανάπτυξης αποστολής για διάσωση λόγο καιρικών συνθηκών ή δυσκολία πρόσβασης των ομάδων σε περιοχές λόγο της γεωγραφικής τοπολογίας, το κόστος, το ανθρώπινο δυναμικό και το ανθρώπινο σφάλμα όσο αφορά τον υπολογισμό των ζημιών και εύρεσης θυμάτων.

* 1. **Στόχος**

Τα Μη Επανδρωμένα Αεροσκαφών (UAV) είναι μια λύση που μπορεί να βοηθήσει σε περιπτώσεις έκτακτης ανάγκης. Τα Μη Επανδρωμένα Αεροσκάφη, κοινός γνωστά ως drone, είναι αεροσκάφη στα οποία η πλοήγηση τους γίνεται απομακρυσμένα μέσω τηλεχειριστηρίου. Στα Μη Επανδρωμένα Αεροσκάφη κινούνται χωρίς την παρουσία πιλότου εντός του αεροσκάφους. Για να γίνεται ο χειρισμός του Μη Επανδρωμένου αεροσκάφους πρέπει να αποκατασταθεί ένα δίκτυο επικοινωνίας μεταξύ του αεροσκάφους και του χειριστηρίου του. Ο χειρισμός των UAV μπορεί να γίνει εξίσου είτε χειροκίνητα από κάποιο άτομο ή αυτοματοποιημένα με την εισαγωγή κάποιου μοντέλου αποστολής.

Η χρήση των UAV σε αποστολές έρευνας ή διάσωσης μειώνουν τον αντίκτυπο που προκαλούν οι παράγοντες της προσβασιμότητας λόγο γεωγραφικής τοπολογίας, η δυσκολία των ομάδων όσο αφορά της καιρικές συνθήκες και έλλειψης ή εξάντλησης ανθρώπινου δυναμικού που αναφέραμε πριν σε σημαντικό βαθμό. Τι γίνεται όμως, όσο αφορά τους παράγοντες του χρόνου, του κόστους και του ανθρώπινου σφάλματος ;

Κατά μεγάλο ποσοστό η τεχνολογία Μη Επανδρωμένων Αεροσκαφών έρχεται σε συνδυασμό με την συλλογή βίντεο και φωτογραφείων τα οποία γίνεται λήψη τους μέσο μιας, μεσαίας μέχρι υψηλών αναλύσεων, κάμερας που επιτάσσεται στα κύρια συστατικά του Drone, και η αποθήκευση τους σε μια κάρτα μνήμης (Micro SD) η οποία περισυλλέγετε και γίνεται η ανάκτηση της σε κάποιο ηλεκτρονικό υπολογιστή. Μέσω αυτών των φωτογραφείων ή βίντεο μπορεί να γίνει περισυλλογή πληροφοριών ,σε ατομική προσπέλαση του υλικού, για την ανάλυση και την εκτίμηση των ζημιών που μπορεί να προκληθεί από τις καταστροφές ή ακόμα και την εύρεση θυμάτων μέσω εφαρμογών αναγνώρισης αντικειμένων συμβάλλοντας θετικά στην απόσβεση των παραγόντων του κόστους, του χρόνου και μείωση του ανθρώπινου σφάλματος.

Συνοπτικά ο στόχος της διπλωματικής εργασίας είναι η ανάπτυξη και υλοποίηση μιας εφαρμογής, η οποία θα συγκεντρώνει τις φωτογραφίες και θα κάνει συνένωση τους σε μια ενιαία φωτογραφία , εκτελώντας κάποιους συγκεκριμένους αλγόριθμους, τον αλγόριθμο του Καμβά και τον αλγόριθμό Stitching. Παράδειγμα, σε ένα χάρτη που προήλθε από φωτογραφίες που λήφθηκαν από θερμική κάμερα από μια αποστολή πτήσης του drone, να μπορούμε να εξάγουμε εκτίμησή για αν μια εστία φωτιάς αναζωπυρώνεται ή την ανίχνευση κάποιου ατόμου στο σκοτάδι. Λόγω της απότομης διαφοράς θερμοκρασίας μεταξύ οντοτήτων, παραδείγματος χάρη το ανθρώπινο σώμα κυμαίνεται γύρω στους 36ο κελσίου και το περιβάλλον γύρω του κυμαίνεται στους 21ο κελσίου στην μέση κατάσταση, θα υπάρχει και η αντίστοιχη εναλλαγή χρώματος μεταξύ των δύο οντοτήτων στην φωτογραφία που θα ληφθεί από το drone, καθιστώντας την χρονοβόρα έρευνα εύρεσης κάποιου θύματος ή μιας εστίας φωτιάς, με την χρήση αλγορίθμου αναγνώρισης πάνω στον χάρτη, πιο εύκολη και συγκριτικά πιο επακριβές όσο αφορά την θέση του στο χάρτη παρά την ατομική προσπέλαση κάθε φωτογραφίας ξεχωριστά από της μονάδες έρευνας. Συνοπτικά η εξαγωγή του χάρτη είναι σημαντικής σημασίας γιατί με την χρήση του μπορεί να γίνει ανίχνευση διάφορων πληροφοριών χωρίς χρονοτριβές.

* 1. **Ερευνητικό υπόβαθρο**

**1.3.1 Προηγούμενη Διπλωματική Εργασία**

Αυτή η διπλωματική εργασία έχει ως σκοπό την συνέχεια μιας προηγούμενης διπλωματικής μελέτης, με τίτλο «Ευφυής αυτονόμηση λειτουργειών μη επανδρωμένων αεροσκαφών για αποστολές εκτάκτου ανάγκης»[1]. Η Προηγούμενη διπλωματική εργασία είναι βασισμένη στο επώνυμο project «PREDICATE», το οποίο έγινε σε συνεργασία με το ερευνητικό κέντρο ΚΙΟΣ που έχει ως έδρα το Πανεπιστήμιο Κύπρου.

Η Προηγούμενη διπλωματική εργασία είχε ως σκοπό την δημιουργία μιας εφαρμογής που θα τίθεται σε λειτουργεία από ομάδες έρευνας και διάσωσης για την εύρεση αγνοουμένων, φυγάδων και αναζωπύρωση εστιών φωτιάς από πυρκαγιές, με την βοήθεια ενός drone και την υλοποίηση αλγορίθμων των βασικών λειτουργειών του για εκτίμηση των κινδύνων. Μέσω της εφαρμογής το drone εκτελεί αυτόνομα την αποστολή-δρομολόγιο που του ανατέθηκε και ο χρήστης μπορεί να παρακολουθήσει την αποστολή σε πραγματικό χρόνο ή να ανάκτηση σε κάποια συσκευή το βίντεο ή τις φωτογραφίες της αποστολής. Το υλικό μπορεί να είναι υψηλής ανάλυσης τις πλείστες περιπτώσεις από μια κάμερα υψηλής ευκρίνειας, είτε από θερμική κάμερα για την εύρεση στην αλλαγή της θερμοκρασίας μέσο της ανάλλαγή χρωμάτων.

Στην αυτή την διπλωματική εργασία το ερευνητικό υπόβαθρο μελετά την υλοποίηση μιας εφαρμογής, όπου γίνεται συνένωση του φωτογραφικού υλικού που ανακτάτε από το μη επανδρωμένο αεροσκάφος και η δημιουργία μιας καινούργιάς φωτογραφίας, η οποία μπορεί να θεωρηθεί σαν χάρτης μια περιοχής που σαρώθηκε από το drone.

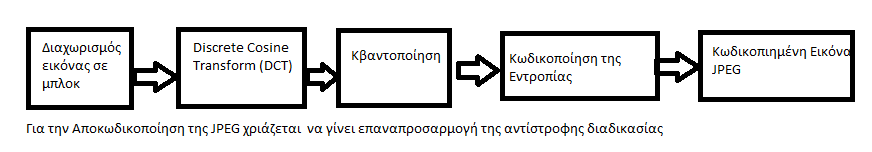
Όπως προαναφέραμε η εξαγωγή του χάρτη είναι σημαντικής ουσίας γιατί με την χρήση του μπορεί να γίνει ανίχνευση διάφορων πληροφοριών. Έτσι για αρχή έπρεπε να γίνει εύρεση και κατανόηση των απαραίτητων συστατικών των οποίων θα συμβάλλαν στην επιτυχία του αποτελέσματος. Τα απαραίτητα συστατικά γενικευμένα είναι το drone data log αρχείου το οποίο μέσα περιέχει πληροφορίες για την αποστολή που διεξάχθηκε και τις φωτογραφίες που λήφθηκαν, η συλλογή των φωτογραφείων και η ψηφιακή τους αναπαράσταση στον ηλεκτρονικό υπολογιστή, καθώς και το φωτογραφικό αποτύπωμα μιας φωτογραφίας, δηλαδή το μήκος και το πλάτος σε αληθινές αποστάσεις σύμφωνα με το ύψος που λήφθηκε η φωτογραφία.

**1.3.2 Ψηφιακή Εικόνα και η Επεξεργασία της**

Για την Ανάπτυξη και την υλοποίηση της εφαρμογής χρειάστηκε καταρχάς, να μελετηθεί διάφορα είδη ψηφιακής εικόνας καθώς και το πως μπορούμε να τις επεξεργαστούμε. Το Drone λόγο του λογισμικού του και την κάμερα που χρησιμοποιεί γνωρίζουμε ότι η εικόνα που παράγει είναι εικόνα τύπου JPEG ([Joint Photographic Experts Group](https://en.wikipedia.org/wiki/Joint_Photographic_Experts_Group)) [2].

Οι εικόνες τύπου JPEG χρησιμοποιούν μια μέθοδο, όπου είναι βασισμένη στην συμπίεση απώλειας για ψηφιακές εικόνες, για να μειώσουν την χωρητικότητα που καταλαμβάνει η εικόνα στην μνήμη της συσκευής κατά αναλογία συμπίεσης 10:1 με μια μικρή απώλεια στην ποιότητα τις εικόνας[3]. Τέτοιου είδους συμπιέσεις ψηφιακής εικόνας, είναι ευπρόσδεκτες για εταιρείες που παράγουν κάμερες για ψηφιακή φωτογράφιση, καθώς και η χρήση των καμερών αυτών, όπως και σε συστήματα UAV.

Η κωδικοποίηση της εικόνας JPEG[4] γίνεται με την ακολουθεί διαδικασία που φαίνεται στην Εικόνα Εικ.1.1 :



(Εικ. 1.1)

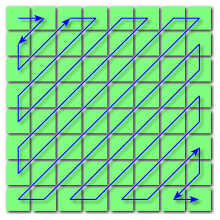
Η κωδικοποίηση της εικόνας JPEG γίνεται με την ακόλουθη διαδικασία που φαίνεται στην Εικόνα. Κάθε κανάλι πρέπει να χωριστεί σε 8\*8 μπλοκ. Η απόδοση της ελάχιστης κωδικοποιημένης μονάδας με μέγεθος 8\*8 ή 16\*8, εξαρτώνται από το χρώμα του υποδείγματος. Στην δημιουργία του βίντεο αυτά ονομάζονται μικρομπλοκ[5].

Αν τα δεδομένα του καναλιού δεν αντιπροσωπεύουν ακέραιο αριθμό από μπλοκ τότε πρέπει να συμπληρωθεί το κενό με κάποιας μορφής ψευδομεταβλητών δεδομένων. Συμπληρώνοντας τις άκριες με ένα χρώμα θα δημιουργήσει ringing artifacts στο περίγραμμά της εικόνας. Η επαναχρησιμοποίηση των ακρινών pixel είναι κοινή τεχνική που χρησιμοποιείται για να μειώσει τα κενά.

Μετά των διαχωρισμό των μπλοκ, κάθε κομμάτι 8\*8 μπλοκ μετατρέπεται σε συχνό-πεδίο ορισμού που χρησιμοποιεί δύο διαστάσεις τύπου-II discrete cosine transform (DCT) [6,7]. Πριν να υπολογιστεί το DCT του 8\*8 μπλοκ οι τιμές του μετατρέπονται από θετικό αριθμό σε 0. Από 8-bit εικόνα, με κάθε πρόσθεση στο αρχικό μπλοκ μπαίνει στο εύρος (0,255). Η τιμή που βρίσκεται στο μέσο του εύρους θα αφαιρεθεί από κάθε είσοδο για να δημιουργήσει ένα εύρος δεδομένων στο κέντρο του αριθμού 0.

Το επόμενο στάδιο είναι ο κβαντισμός [8], που είναι βασισμένος στην ανατομία του ανθρώπινου ματιού. Το ανθρώπινο μάτι μπορεί να εντοπίσει μικρές διαφορές στην φωτεινότητα σε μια μεγάλη επιφάνεια αλλά είναι δύσκολο να ξεχωρίσει την ακριβή δυναμική έντονων διαφορών στην φωτεινότητα. Αυτό επιτρέπει την μείωση του όγκου πληροφοριών στα συστατικά υψηλής συχνότητας. Η υλοποίηση αυτού μπορεί να γίνει διαιρώντας κάθε συστατικό με μια σταθερά του ίδιου του συστατικού και αν χρειάζεται να στρογγυλοποιηθεί στο πιο κοντινό ακέραιο. Η στρογγυλοποίηση του αριθμού αυτού είναι η μόνη διαδικασία που μπορεί να μην είναι τόσο ακριβής αν ο υπολογισμός του DCT έχει γίνει σωστά. Σαν αποτέλεσμα αυτού, τα συστατικά υψηλής συχνότητας είναι 0 και τα υπόλοιπα μετατρέπονται σε μικρούς αρνητικούς η θετικούς αριθμούς, στα οποία χρησιμοποιούντα πολύ λίγα bit για να αναπαρασταθούν.

Στο τελευταίο στάδιο, η κωδικοποίηση της Εντροπίας[9] είναι μια μορφή χωρίς απώλειες δεδομένων συμπίεσης. Η διάταξη των συστατικών της εικόνας γίνεται σε “zigzag”, όπως βλέπουμε και στην Εικόνα Είκ.1.2 σειρά χρησιμοποιώντας των αλγόριθμο run-length encoding που ομαδοποιεί παρόμοιες συχνότητες μαζί, εισάγοντας τα μηδενικού μήκους κωδικοποίηση, και μετά χρησιμοποίει Huffman κωδικοποίηση [10] σε ότι έχει απομείνει.



Εικ. 1.2

Λόγο της διαδικασίας που είδαμε πιο πάνω και της πολυπλοκότητας, αποφασίστηκε να γίνει χρήση εικόνων τύπου BMP (Device-independent bitmaps) [11] , η οποίες είναι απλά μια σειριακή ακολουθία της εικόνας σε byte. Γενικότερα δεν θα χρειαστεί διαδικασία κωδικοποίησης και αποσυμπιεστής για να έχουμε τα δεδομένα μας σε ακολουθίες byte, στην περίπτωση της JPEG.

Για την Χρήση της BMP, έπρεπε να γίνει μελέτη στην δομή των δεδομένων της εικόνας και των χαρακτηριστικών της [12].

* + 1. **Η Έννοια και χρήση του Stitching**

Stitching είναι μια έννοια η οποία χρησιμοποιείται ευρύτατα στον χώρο και τον κλάδο της επεξεργασίας ψηφιακής κάμερας. Πολλές γλώσσες προγραμματισμού όπως η Python ανέπτυξαν βιβλιοθήκες [13,14] για την υλοποίηση αλγορίθμων βασισμένων στο Stitching.

Το stitching εικόνων είναι ο συνδυασμός 2 ή περισσότερων εικόνων που καλύπτουν η μία την άλλη για να διαμορφώσουν μια μεγαλύτερη εικόνα.

Ο αλγόριθμος που ακολουθήθηκε για να εξάγουμε το αποτέλεσμα είναι βασισμένος σε κάποια βήματα [15,16] :

1. Εντοπισμός keypoint (SIFT) [17,18]

2. Αντιστοίχιση keypoints [19,20]

3. Εκτίμηση Ομογραφίας [21,22,23] (Χρησιμοποιώντας τον αλγόριθμο RANSAC)

4. Παρουσίαση της εικόνας και Στρέβλωση [24]

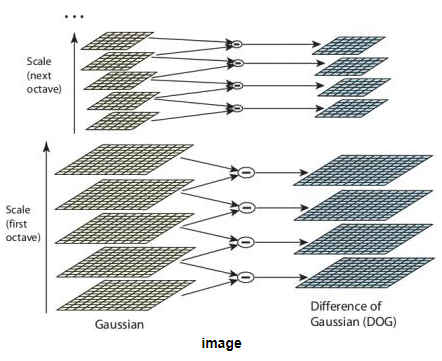
Για να γίνει η αντιστοίχιση των εικόνων πρέπει να βρεθεί ένα σύνολο από ξεχωριστά κύρια σημεία (keypoint). Τα keypoint πρέπει να είναι ξεχωριστά και επαναλαμβανόμενα. Να οριστεί μια περιοχή γύρω από κάθε κύριο σημείο και να το διαχωρίσει και κανονικοποιήσει το περιεχόμενο της περιοχής. Μετά πρέπει να υπολογιστεί ένας λογικός περιγραφέας για την κανονικοποίηση της περιοχής . Στο τέλος πρέπει να ταυτοποιηθούν οι περιγραφής.

Το αποτέλεσμα Stitching μπορεί να επηρεαστεί από πολλούς παράγοντες όπως:

Προβλήματα stitching :

* Διαφορετική ένταση σε όλη την εικόνα
* Διαφορετική ένταση και αντίθεση ανάμεσα στα πλαίσια
* Παραμόρφωση φακού
* Κίνηση στις εικόνες
* Αποπροσανατολισμός

Το ισχυρότερο μέρος του Stitching είναι η ανίχνευση και η αντιστοίχιση των “keypoints”. Ο αλγόριθμος SIFT (Scale-Invariant Feature Transform)[25] , εξάγει τα βασικά σημεία και υπολογίζουν τους περιγραφικούς δείκτες. Ο αλγόριθμος SIFT, χρησιμοποιεί τη “Difference of Gaussians” (DoG) [26]. H DoG παράγεται ως η διαφορά μεταξύ δύο διαφορετικών θολώσεων μεταξύ δύο κλιμακωτών παραμέτρων μιας εικόνας μέσω της Γκαουσιανής Πυραμίδας και την εφαρμογή της μεταξύ διαφορετικών οκτάβων, όπως φαίνεται και στην Εικόνα Εικ. 1.2



(Εικ. 1.2)

Έτσι γίνεται και ο εκτοπισμός πιθανών keypoint. Αμέσως μετά γίνεται χρήση κατωφλίου για να γίνει εξάλειψη των keypoint που έχουν χαμηλή αντίθεση και τα ακραία keypoint, δηλαδή αυτά που βρίσκονται κοντά στην περιφέρεια της εικόνας, αφήνοντας έτσι τα εναπομείναντα δυνατά σε αντίθεση και όσο πιο κοντά στο κέντρο της εικόνας.

H αντιστοίχιση των όμοιων keypoint, γίνεται με αναγνώριση των γειτονικών keypoint, δηλαδή το κατά πόσο η περιοχή γύρο από τα δυο πιθανά για αντιστοίχιση keypoint περιλαμβάνει όμοια δευτερεύοντα keypoint. Εάν τα γειτονικά keypoint διαφέρουν μεταξύ των εικόνων κατά περισσότερο από 0.8, τότε απορρίπτονται, έτσι αφανίζοντας το 90% των keypoint και απομένουν μόνο αυτά , που στο τέλος αντιστοιχούνται.

* + 1. **Μαθηματικοί Υπολογισμοί και Αποστάσεις.**

Για τον υπολογισμό κάποιων αποστάσεων χρειάστηκαν οι τύποι από κάποιες φόρμουλες όπως η φόρμουλα για το Field of View [27,28] του Drone και η Haversine Formula[29,30] η οποία Εκτιμά την απόσταση μεταξύ δυο γεωγραφικών σημείων στην επιφάνεια της γης.

Για τον υπολογισμό του Field of View έγινε εξαγωγή των υπολογισμών μέσο του γενικευμένου τύπου του Field of View στην Ψηφιακή επεξεργασία εικόνας. Χρησιμοποιώντας των γενικευμένο τύπο, εξάγονται οι φόρμουλες για την παραγωγή του Αποτυπώματος.

Οι μετατροπές όσο αφορούν την εξαγωγή των υπολογιστικών πράξεων είναι βάση του τύπου του Field Of View[40], αντικαθιστώντας την ίριδα του ανθρώπου και την απόσταση μεταξύ του αντικειμένου που παρακολουθείτε με το υψόμετρο, δηλαδή την απόσταση από τον φακό της κάμερας του Drone ως την επιφάνεια της γης, αυτό συνεπώς μας δίνει μια προσέγγιση του υπολογισμού, τόσο ακριβές με σχεδόν μηδενική απόκλιση.

Όσο αφορά την εισαγωγή των γονιών αντίβαρου, απλά έγινε αλλαγή του αλγορίθμου με υπολογισμούς ως προς την οπτική γωνία του αλγορίθμου. Δηλαδή την εισαγωγή της εφαπτομένης ως προς την γωνία του αντίβαρου.

Η φόρμουλα haversine καθορίζει την απόσταση δύο σημείων σε μια σφαίρα λαμβάνοντας υπόψη τα μήκη και τα γεωγραφικά πλάτη τους. Είχε σημαντικό ρόλο όσο αφορά τον προσανατολισμό και αποτελεί μια ειδική περίπτωση μιας γενικότερης φόρμουλας στης σφαιρικής τριγωνομετρία, ο νόμος των haversines[30], που συσχετίζει τις πλευρές και τις γωνίες των σφαιρικών τριγώνων.

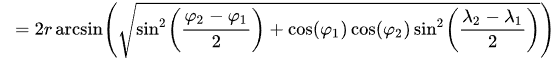
Οι εξής παράγοντες που χρειάζονται για τον υπολογισμό αυτό είναι

• d είναι η απόσταση μεταξύ των δύο σημείων (κατά μήκος ενός μεγάλου κύκλου της σφαίρας), Δηλαδή το αποτέλεσμα μας.

• r είναι η ακτίνα της σφαίρας, Η ακτίνα της Γης.

• φ1, φ2: γεωγραφικό πλάτος του σημείου 1 και γεωγραφικό πλάτος του σημείου 2, σε ακτίνια(Radians)

• λ1, λ2: μήκος του σημείου 1 και γεωγραφικό μήκος του σημείου 2, σε ακτίνια (radians)



Ο τύπος, είναι κατά προσέγγιση 0.7% όταν εφαρμόζεται στη Γη, για το λόγο ότι δεν είναι μια τέλεια σφαίρα. Η "ακτίνα της γης" κυμαίνεται από 6356.752 χλμ., στους πόλους έως 6378.137 χλμ. στον ισημερινό. Το πιο σημαντικό είναι ότι η ακτίνα καμπυλότητας μιας γραμμής βορρά-νότου στην επιφάνεια της γης είναι κατά 1% μεγαλύτερη στους πόλους (≈6399.594 χλμ.), Από ό, τι στον ισημερινό (≈6335.439 χλμ.).

* 1. **Μεθοδολογία**

Η μεθοδολογία που ακολουθήθηκε για την εκπόνηση της διπλωματικής εργασίας, ήταν όπως προαναφέρθηκε, η μελέτη για τα βασικά συστατικά που θεωρούνται απαραίτητα για το επιτυχές αποτέλεσμα της εφαρμογής.

Πρώτα έπρεπε να γίνει κατανόηση του φωτογραφικού αποτυπώματος μιας φωτογραφίας και πως οι πληροφορίες αυτές θα μας ήταν χρήσιμες. Μέσω αυτού γίνεται μια δημιουργία μιας βοηθητικής αυτόνομης υπηρεσίας η οποία λαμβάνει σαν είσοδο πληροφορίες σχετικά με το drone και της φωτογραφίες και δημιουργά μια εκτίμηση των αποστάσεων που καλύπτονται από τις φωτογραφίες.

Στη συνέχεια έπρεπε να γίνει μελέτη η οποία θα αφορούσε την κατανόηση για την δομή του data log file και η δημιουργία μια μικρής βοηθητικής εφαρμογής όπου θα έκανε αποσφαλμάτωση το αρχείο, για εγκυρότητα όσο αφορά την δομή του και της πληροφορίες που μπορεί να ληφθούν από αυτό.

Σε κατοπινό στάδιο έπρεπε να γίνει μελέτη και εξαγωγή αλγορίθμων οι οποίοι θα συνέβαλαν στην δημιουργία του χάρτη. Ο πρωταρχικός αλγόριθμος Stitching, είναι αλγόριθμος του οποίου γίνετε χρήση βιβλιοθήκης OpenCV και κατά την εκτέλεση του, γίνεται εύρεση “key point” ανάμεσα σε δυο φωτογραφίες και της συνενώνει με την ακριβής χρήση της έννοιας “stitching”. Ο δεύτερος αλγόριθμος, o οποίος ονομάστηκε Canvas Algorithm, δημιουργήθηκε με βάση την τοποθέτηση των φωτογραφείων σε ένα καμβά, κάνοντας ανάλυση και υπολογισμό των γεωγραφικών συντεταγμένων των φωτογραφείων με την βοήθεια του αποτυπώματος της φωτογραφίας και του drone data log file. Τέλος γίνεται συνένωση των προαναφερθέντων σε μια ενιαία εφαρμογή με την χρήση γραφικών διπροσωπιών.

Για την υλοποίηση, όσο αφορά της βοηθητικές εφαρμογές και τον Αλγόριθμο του Καμβά, γίνεται η χρήση της γλώσσας προγραμματισμού Java με την βοήθεια του εργαλείου IntelliJ IDEA[31] και της βιβλιοθήκης JavaSE 1.8 . Για την υλοποίηση του Αλγορίθμου Stitching γίνεται η χρήση της γλώσσας προγραμματισμού Python (Python 3.6)[32] με την επιπρόσθετη χρήση των βιβλιοθηκών OpenCV-Python, ΟpenCV-Contrib-Python, imutils και numpy. Για την αποσφαλμάτωση των συναρτήσεων των βασικών κλάσεων που χρησιμοποιούνται γίνετε η χρήση JUNIT [33]. Για την τεκμηρίωση του κώδικα γίνεται χρήση Javadoc. Η δημιουργία των γραφικών διπροσωπιών γίνεται την χρήση του plugin Windows Builder και βιβλιοθήκης SWING[34].

Για την δημιουργία των διαγραμμάτων κλάσεων γίνεται με την χρήση του plugin Object Aid Class Diagram [35]. Για την επίβλεψη και οργάνωση την ανάπτυξη του κώδικα έγινε χρήση του GitHub όπου και βρίσκεται αναρτημένος και ο κώδικας, καθώς και το doc file [36]με την τεκμηρίωση.

**1.5 Οργάνωση Διπλωματικής Εργασίας**

Αυτή η διπλωματική εργασία έχει διαχωριστεί και οργανωθεί ως εξής. Στο δεύτερο κεφάλαιο θα δούμε την παρουσίαση της εφαρμογής Αποτυπώματος του Drone, το κεφάλαιο περιλαμβάνει τον ψευδοκώδικα που ακολουθήθηκε για την δημιουργία των μεθόδων υπολογισμού, καθώς και την λεπτομερή περιγραφή του. Αμέσως μετά ακολουθούν Περιγραφές σχετικά με την βοηθητική εφαρμογή DroneDataEditror, τα δεδομένα εξόδου, τα δεδομένα εισόδου, τις αναφορές που εκδίδονται από την εφαρμογή και την εγκυρότητα των δεδομένων. Το Κεφάλαιο τελειώνει με την Παρουσίαση των Γραφικών Διαπροσωπιών της εφαρμογής και του DroneDataEditor.

Στο τρίτο κεφάλαιο θα δούμε την παρουσίαση της εφαρμογής των αλγορίθμων Stitching και Καμβά, το κεφάλαιο περιλαμβάνει τον ψευδοκώδικα που ακολουθήθηκε για την δημιουργία των μεθόδων υπολογισμού και την εξαγωγή του τελικού αποτελέσματος του κάθε Αλγορίθμου ξεχωριστά σε διαφορετικά υποκεφάλαια, καθώς και την λεπτομερή περιγραφή τους. Επίσης σε ξεχωριστό κεφάλαιο ανατέθηκε η επεξήγηση του αντικειμένου BlackBox, το οποίο είναι ζωτικής σημασίας για την εκτέλεση και των δυο Αλγορίθμων. Αμέσως μετά ακολουθούν Περιγραφές σχετικά με την βοηθητική εφαρμογή DroneRouteEditror, τα δεδομένα εξόδου, τα δεδομένα εισόδου, τις αναφορές που εκδίδονται από την εφαρμογή και την εγκυρότητα των δεδομένων. Το Κεφάλαιο τελειώνει με την Παρουσίαση των Γραφικών Διαπροσωπιών της εφαρμογής και του DroneRouteEditor.

Στο τέταρτο κεφάλαιο γίνεται πειραματική αξιολόγηση των δυο αλγορίθμων όσο αφορά την εκτέλεση με διαφοροποίηση σε παραμέτρους σημαντικούς για αυτή , ως προς το μέγεθος των εικόνων, την ποσότητα των εικόνων, τον χρόνο εκτέλεσης κ.τ.λ. Επίσης γίνεται αναφορά στον τρόπο ελέγχου των δεδομένων του κώδικα του αλγορίθμου του καμβά, και τα στατιστικά του όσο αφορά τα JUNIT αρχεία που δημιουργήθηκαν για αυτό τον σκοπό.

Στο πέμπτο κεφάλαιο γίνεται σύγκριση των αλγορίθμων αποτελεσματικά και η σύγκριση τους σε ακραίες περιπτώσεις. Επίσης γίνεται αναφορά στις δυσκολίες που αντιμετωπίστηκα κατά την υλοποίηση των εφαρμογών και την ανάπτυξη των αλγορίθμων. Τέλος γίνεται αναφορά για μελλοντική ερευνητική εργασία που μπορεί να γίνει για την βελτιστοποίηση των αλγορίθμων, καθώς και για την εισαγωγή επιπρόσθετων αρχιτεκτονικών όσο αφορά την ανάπτυξη συστήματος.

**Κεφάλαιο 2**

**Εφαρμογή Υπολογισμού Αποτυπώματος Φωτογραφίας από Προκαθορισμένο Υψόμετρο**

2.1 Εφαρμογή 14

2.1.1 Ψευδοκώδικας Αλγορίθμου 15

2.1.2 Λεπτομερείς Περιγραφή Αλγορίθμου 15

2.1.3 Βοηθητική εφαρμογή, Δεδομένα Εισόδου και Εγκυρότητα τους 19

2.1.4 Δεδομένα Εξόδου και Αρχεία Αναφορών 21

2.2 Επεξήγηση Γραφικής Διαπροσωπία Χρήστη 22

**2.1 Εφαρμογή**

Η Εφαρμογή αυτή έχει ως σκοπό τον υπολογισμό των αποστάσεων που καλύπτονται από τις φωτογραφίες. Χρησιμοποιείτε για την δημιουργία μιας αναφοράς για πολλαπλές φωτογραφίες καθώς μπορεί να χρησιμοποιηθεί και σαν υπολογιστική μηχανή για στιγμιαίους υπολογισμούς όπως βλέπουμε στην εικόνα πιο κάτω στο κεφάλαιο αυτό (Είκ. 2.7). Επίσης ο αλγόριθμος της εφαρμογής αυτής αποτελεί και σαν συστατικό για την εκτέλεση του αλγορίθμου καμβά που θα μελετηθεί στο επόμενο κεφάλαιο. Επίσης η εφαρμογή περιλαμβάνει βοηθητική εφαρμογή για την δημιουργία και την εγκυρότητα του DroneData.txt που χρησιμοποιείται σαν είσοδος για τον υπολογισμό. Ο πηγαίος κώδικας της εφαρμογής μπορεί να βρεθεί στο GitHub ενώ σημαντικά κομμάτια του αλγορίθμου αναρτούνε στο Παράρτημα Β, ενώ το διαγράμματα κλάσεων αναρτούνε στο παράρτημα Α.

**2.1.1 Ψευδοκώδικας Αλγορίθμου**

Ο αλγόριθμος αποτελείτε από τρεις φάσεις οι οποίες διαχωρίζονται σε δύο υποφάσεις η κάθε μια αντίστοιχα, όπως φαίνεται και στον πίνακα πιο κάτω (Εικ. 2.1) . Ο υπολογισμός γίνεται με χρήση αντικειμένου με τύπο κλάσης Drone και την επιστρέφει μια λίστα αντικειμένου τύπου Area.



Εικ.2.1

**2.1.2 Λεπτομερής Περιγραφή Αλγορίθμου**

Στην πρώτη φάση του αλγορίθμου γίνεται η δημιουργία μιας λίστας με τύπο αντικειμένου Drone. Η κλάση τύπου Drone, η οποία αναρτάτε στο Παράρτημα Β, αποτελείτε από τα βασικά πεδία μεταβλητών :

* **private** **String** dronetype ⇨ Το μοντέλο της κάμερας του Drone.
* **private** **String** imagetitle ⇨ Ο τίτλος του αρχείου της εικόνας.
* **private** **double** x\_sensor ⇨ Ο αισθητήρας για το πλάτος της εικόνας σε μιλίμετρα.
* **private** **double** y\_sensor ⇨ Ο αισθητήρας για το ύψος της εικόνας σε μιλίμετρα .
* **private** **double** x\_gimbal\_angle ⇨ Η γωνία αντίβαρου για το πλάτος της εικόνας σε μοίρες.
* **private** **double** y\_gimbal\_angle ⇨ Η γωνία αντίβαρου για το ύψος της εικόνας σε μοίρες.
* **private** **double** altitude ⇨ Το ύψος του Drone τη στιγμή λήψης της εικόνας σε

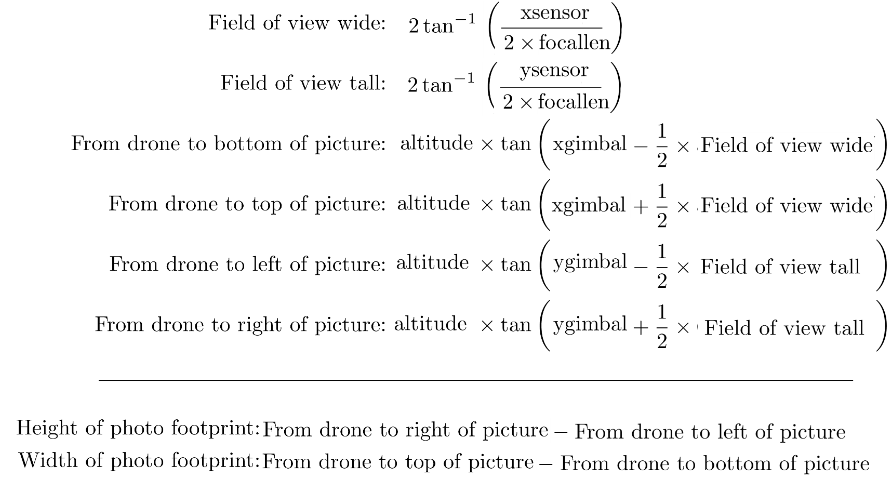
μέτρα.

* **private** **double** focallen ⇨ Το εστιακό μήκος του φακού της κάμερας σε μιλίμετρα.
* **private** **double** latitude ⇨ Το γεωγραφικό πλάτος της εικόνας τη στιγμή της λήψης της εικόνας.
* **private** **double** longitude ⇨ Το γεωγραφικό μήκος της εικόνας τη στιγμή της λήψης της εικόνας.

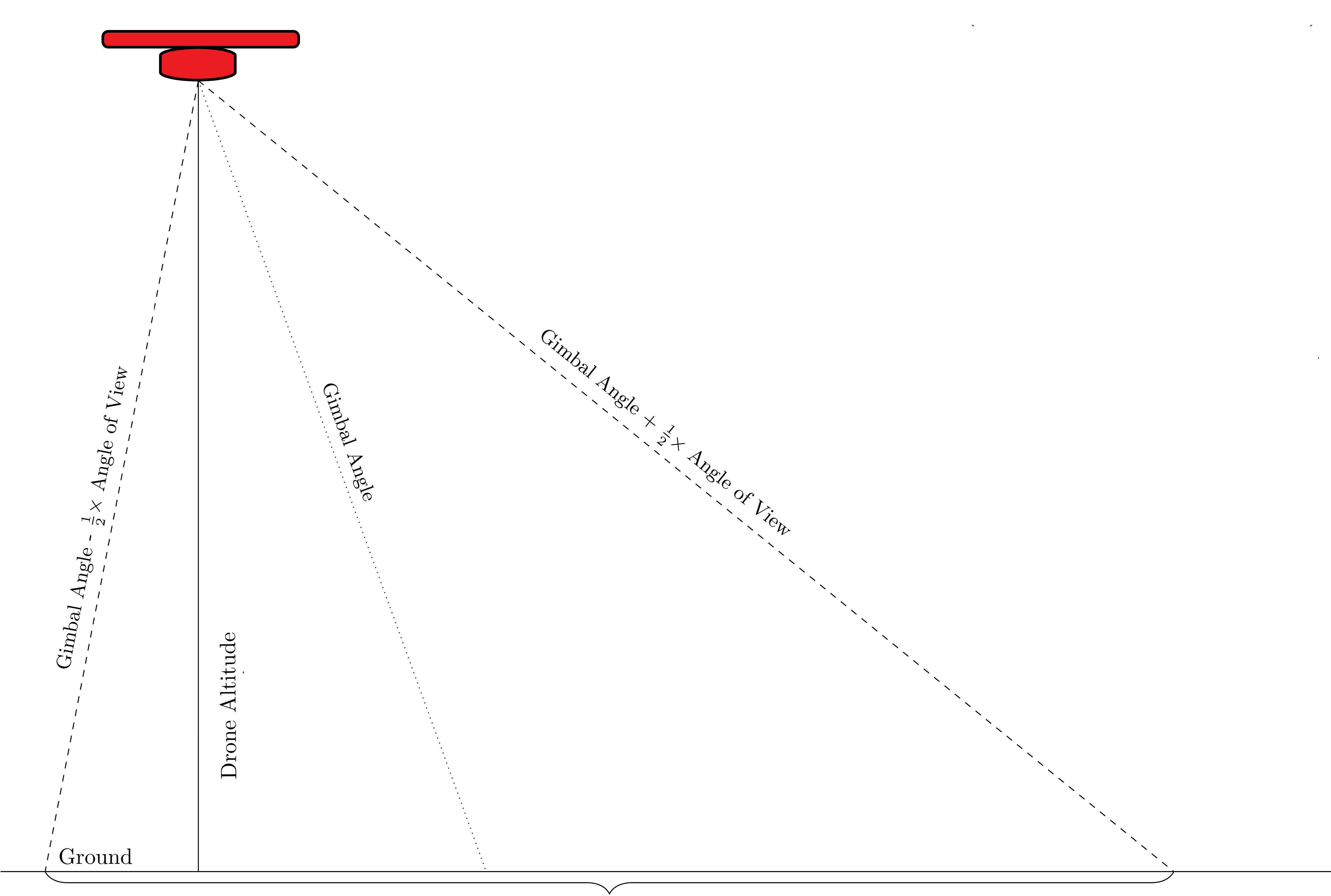
Από τα βασικά πεδία που είναι απαραίτητα για τον υπολογισμό του αποτυπώματος είναι ο αισθητήρας για το πλάτος της εικόνας, ο αισθητήρας για το ύψος της εικόνας, η γωνία αντίβαρου για το πλάτος της εικόνας, η γωνία αντίβαρου για το ύψος της εικόνας, το ύψος του Drone την στιγμή λήψης της εικόνας και το εστιακό μήκος του φακού της κάμερας.

Εωσότου δημιουργηθεί η λίστα γίνεται ανάγνωση του Αρχείου DroneData.txt (Κεφ. 2.1.3) και φορτώνουμε την λίστα μας με τα αντικείμενα Drone στα οποία με την ανάθεση των πεδίων κατάλληλα με τις τιμές που δόθηκαν από το αρχείο. Σημειώνουμε ο αριθμός των στιγμιότυπων Drone ισούται με τον αριθμό των φωτογραφιών που είναι εγγεγραμμένες στο αρχείο.

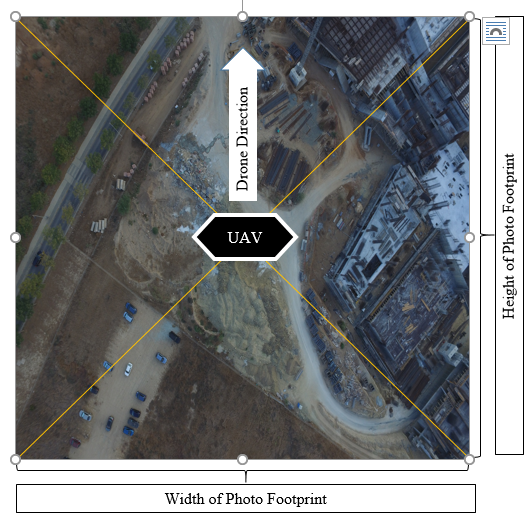
Στην δεύτερη φάση του αλγορίθμου γίνεται υπολογισμός των πραγματικών αποστάσεων του αποτυπώματος μέσω μιας σειρά μαθηματικών υπολογισμών όπου υλοποιούνται στις διάφορες μεθόδους της κλάσης CalculationProcedures.java. Στην Εικόνα Εικ. 2.2 φαίνονται οι εξισώσεις η οποίες υπολογίζουν σταδιακά τις απόστασης του αποτυπώματος της φωτογραφίας του drone όπου πάρθηκαν από την ιστοσελίδα stack exchange αυτούσιες . Στην Εικόνα Εικ 2.3 βλέπουμε την συσχέτιση του ύψους και της γωνίας αντίβαρου και το πως Τριγωνομετρικά έγινε ο υπολογισμός της απόσπασης για την εύρεση των πραγματικών αποστάσεων του αποτυπώματος.



(Εικ.2.2)



(Εικ.2.3)

Η απεικόνιση των διαστάσεων εικονικά φαίνεται στην παρακάτω εικόνα. Σημειώνουμε ότι το πραγματικό μήκος και πλάτος εξαρτώνται από την κατεύθυνση του UAV και αντίστοιχα το πλάτος τις εικόνας σε pixels (width x height) ⬄ (height x width)

(Εικ.2.4)

Στην τρίτη φάση του αλγορίθμου γίνεται ανάθεση των υπολογισμών σε ένα καινούργιο τύπο

κλάσης Area , η οποία αναρτάτε στο παράρτημα Β, με πεδία μεταβλητών

* **private** **String** imagetitle ⇨ Το όνομα του αρχείου που αντιστοιχούν οι αποστάσεις.
* **private** **double** height ⇨ Το πραγματικό μήκος που καλύπτει η εικόνα σε μετρά.
* **private** **double** width ⇨ Το πραγματικό πλάτος που καλύπτει η εικόνα σε μετρά.

Στην ουσία το αντικείμενο Area έχει τον ρόλο τις συμπίεσης του αντικειμένου Drone για να

μπορεί να είναι πιο ευέλικτο κατά το πέρασμα του στην εφαρμογή του καμβά (Κεφ.3). Η ποσότητα των αντικειμένων τύπου Area που δημιουργούνται είναι αντίστοιχή της ποσότητας των Drone που είναι σαν είσοδος για τον υπολογισμό. Όπως και κατά την αρχικοποίηση του αλγορίθμου έτσι και τώρα δημιουργείται μια λίστα τύπου Area όπου επιστρέφεται και σαν τιμή επιστροφής από τον executioner τις εφαρμογής για να περάσει στον αλγόριθμο του καμβά όπως προαναφέραμε. Κατά τον τερματισμό του αλγορίθμου παράγεται και ένα αρχείο AreaReport.txt όπου περιέχει τα αποτελέσματα του αλγορίθμου σε μορφή Αναφοράς (Κεφ. 2.4).

**2.1.3 Βοηθητική εφαρμογή, Δεδομένα Εισόδου και Εγκυρότητα τους**

Για τον υπολογισμό τον αποστάσεων του αποτυπώματος της φωτογραφίας του Drone χρειάζεται με κάποιο τρόπο εισαγωγή δεδομένων στην εφαρμογή υπολογισμού. Η μορφή εισόδου, θεωρήθηκε και ως η πιο απλή, είναι κάποιο αρχείο κειμένου (.txt) το οποίο δημιουργείτε ή επεξεργάζεται από μια βοηθητική εφαρμογή επονομαζόμενη «DroneDataEditor».

Ο DroneDataEditor αποτελείτε από 4 σημαντικούς τομείς, όπως φαίνεται και στην Εικόνα Εικ.2.8 αργότερα. Μέσω αυτού του editor μπορεί ο χρήστης να δημιουργήσει ή να επεξεργαστεί αντίστοιχα ένα αρχείο DroneData.txt . Στην περίπτωση της δημιουργίας ο χρήστης έχει δυο επιλογές, την επιλογή manual add και την επιλογή Upload add. Στην επιλογή manual μπορεί να προσθέσει δεδομένα εισόδου απλά γεμίζοντας τα αντίστοιχα πεδία. Στην δεύτερη επιλογή μπορεί να ανεβάσει μια εικόνα και αυτόματα τα στοιχεία θα ανακτηθούν στα πεδία εισόδου. Αυτό που αξίζει να σημειωθεί είναι ότι για την uploading add, δημιουργήθηκαν ειδικές μέθοδοι οι οποίες μέσο της χρήσης της βιβλιοθήκης JAVAXT να μπορούμε να πάρουμε της πληροφορίες από τα details της εικόνας[37,38,39] που θέλουμε να κάνουμε add. Παράδειγμα μιας από αυτές τις μεθόδους είναι οι παρακάτω και περιέχεται στο αρχείο GUIProcedures.java :

/\*Αυτή η συνάρτηση παίρνει το τύπο της κάμερας ο οποίος περιέρχεται στις

\* Λεπτομέρειες της εικόνας

\* **@param** path - path to imag3

\* **@return** camera type String.

\*/

**public** String showImageCameraType(String path) {

//Δημιουργία του αντικειμένου

javaxt.io.Image image = **new** javaxt.io.Image(path);

//Χρήση της getExifTags() για την ανάκτηση του HashMap

java.util.HashMap<Integer, Object> exif = image.getExifTags();

// exif.get(0x0110) = Το κλειδί 0x0110 μας δίνει την τιμή που βρίσκεται //στο συγκεκριμένο σημείο του HashMap, στην περίπτωση αυτή του μοντέλου //της κάμερας. Γίνεται casting σε τύπο δεδομένου String.

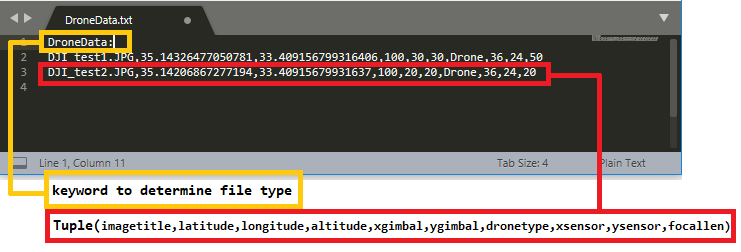
String temp = (String) exif.get(0x0110);

**return** temp;

}

Πληροφοριακά η φωτογραφίες που μπορούν να γίνουν upload πρέπει να είναι εικόνες που προέρχονται από κάποιο drone έτσι ώστε να περιέχουν και τις ειδικές πληροφορίες για ανάκτηση όπως Camera Type, GPS συντεταγμένες κ.τ.λ. Κομμάτι της GUIProcedures.java βρίσκεται στο Παράρτημα Β.

Το αρχείο αυτό αποτελείτε από δυο σημαντικά συστατικά. Το πρώτο συστατικό είναι η λέξη-κλειδί (keyword) «DroneData:», η οποία έχει δύο ρόλους. Ξεχωρίζει το είδος του αρχείου, εάν αυτό είναι το αρχείο που έχει τα απαραίτητα δεδομένα για την είσοδο της εφαρμογής υπολογισμού και επιπρόσθετα σηματοδοτεί την έναρξη του reader να ανακτήσει τα δεδομένα του αρχείου και να τα αναθέσει στις δομές. Το δεύτερο συστατικό είναι οι πλειάδες(tuples) των δεδομένων. Κάθε πλειάδα αναπαριστά τα δεδομένα που θα χρειαστούν, καθώς και πληροφορίες που σηματοδοτούν την κάθε εικόνα που θα χρειαστεί να υπολογιστεί το αποτύπωμά της. Το ποσό των εικόνων που θα υπολογιστούν είναι το σύνολο των πλειάδων και κάθε πλειάδα αποτελείτε από τα πεδία (imagetitle, latitude, longitude, altitude, x\_gimbal, y\_gimbal, dronetype, x\_sensor, y\_sensor, focallen). Μια αναπαράσταση του αρχείου είναι όπως αυτή που βλέπουμε στην Εικόνα Εικ. 2.5



(Εικ.2.5)

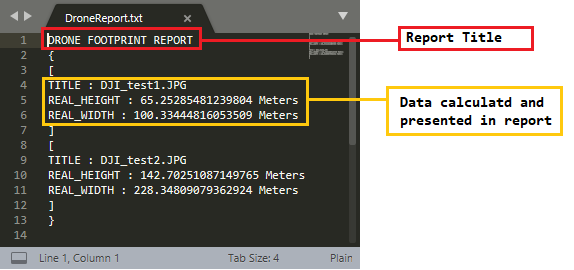
Για την εγκυρότητα του αρχείου λήφθηκαν υπόψη οι ακόλουθοι κανόνες :

1. Η λέξη κλειδί, η οποία πάντα πρέπει να βρίσκεται στην πρώτη γραμμή του αρχείου και σηματοδοτεί τον reader ότι αρχίζουν οι πλειάδες δεδομένων.
2. H λέξη κλειδί είναι η « DroneData: ».
3. Στην δεύτερη γραμμή του αρχείου πάντα βρίσκεται η πρώτη πλειάδα δεδομένων.
4. Το αρχείο πρέπει να περιέχει μόνο τις πλειάδες και την λέξη κλειδί.
5. Τα δεδομένα της πλειάδας πάντα πρέπει να είναι στη ακολουθία (imagetitle, latitude, longitude, altitude, x\_gimbal, y\_gimbal, dronetype, x\_sensor, y\_sensor, focallen).
6. Οι τύποι δεδομένου της πλειάδας είναι αντίστοιχα (String, double, double, double, double, double, String, double, double, double).
7. Τα δεδομένα τις κάθε πλειάδας διαχωρίζονται με «,» χωρίς κανένα κενό ενδιάμεσα.
8. Στο τέλος τις κάθε πλειάδας υπάρχει πάντα χαρακτήρας «\n».
9. Στο τέλος του κάθε αρχείου υπάρχει χαρακτήρας EOF.

Κατά την επιλογή του αρχείου στην εφαρμογή του αλγορίθμου, ο validator του αρχείου, ενεργοποιείτε αυτόματα και ενημερώνει για την εγκυρότητα του αρχείου με το κατάλληλο μήνυμα.

**2.1.4 Δεδομένα Εξόδου και Αρχεία Αναφορών**

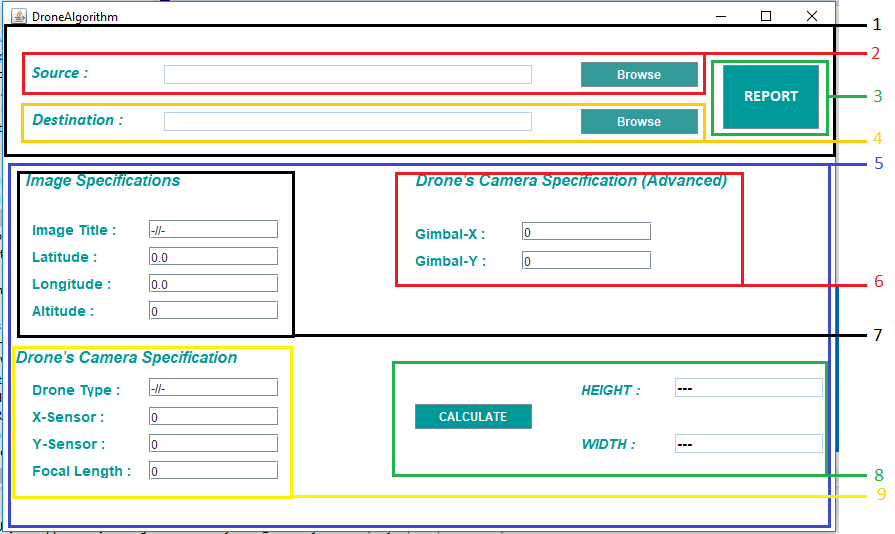
Μετά τον τερματισμό του αλγορίθμου και όταν οι υπολογισμοί γίνουν, ο Phase Executioner της εφαρμογής επιστρέφει μια λίστα τύπου Area, η οποία εμπλέκεται στον αλγόριθμο του καμβά (Κεφ.3). Επιπρόσθετα η εφαρμογή δημιουργεί ένα report με τα αποτελέσματα της εφαρμογής σε ένα αρχείο κειμένου(.txt) και με ονομασία AreaReport.txt. Αυτό αποτελείται από τον τίτλο του Report και τα δεδομένα που υπολογίστηκαν διατεταγμένα σε πλειάδες χρονισμένες με αγκύλες και άγκιστρα. Παράδειγμα του Report παρουσιάζεται στην Εικόνα Εικ. 2.6.



(Εικ.2.6)

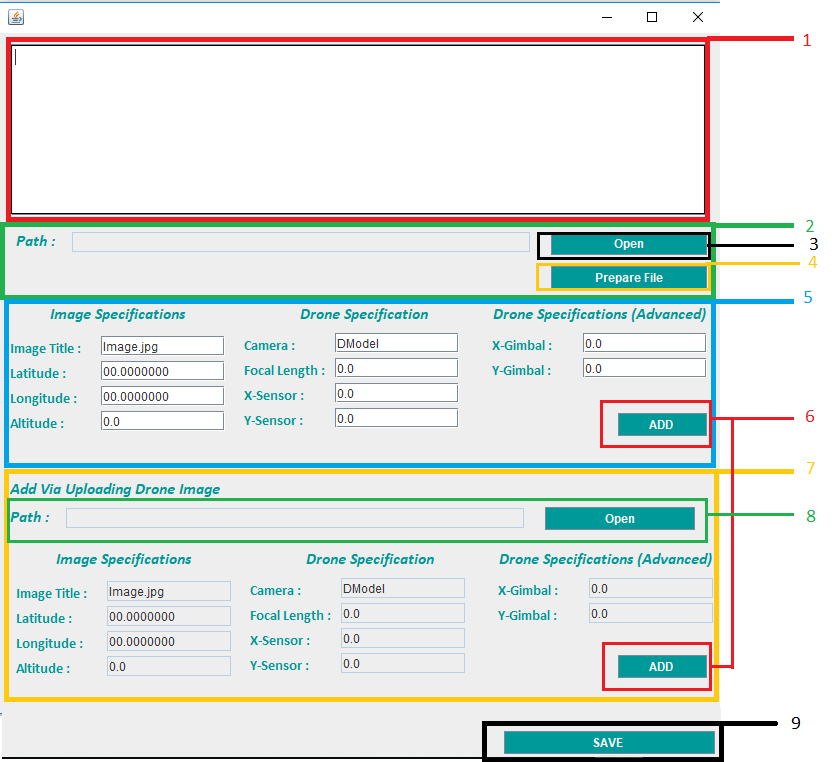
**2.2 Επεξήγηση Γραφικής Διαπροσωπίας Χρήστη**

Για την αναπαράσταση της εφαρμογής και των βοηθητικών εφαρμογών δημιουργήθηκαν κάποιες γραφικές διαπροσωπίες χρήστη (GUI) στα οποία γίνεται αυτοματοποιημένη η εκτέλεση του αλγορίθμου για παραγωγή της αναφοράς των αποτελεσμάτων, καθώς ο χρήστης μπορεί να δημιουργήσει και να επεξεργαστή δια δραστικά το αρχείο DroneData.txt όπου περιέχει τα δεδομένα εισόδου για εκτέλεση της υπολογιστικής εφαρμογής. Καταρχάς, όπως βλέπουμε στην Εικόνα Εικ. 2.7 την γραφική διαπροσωπία για την εκτέλεση του αλγορίθμου και στην Εικόνα Εικ. 2.8 την βοηθητική εφαρμογή για δημιουργίας.



(Εικ.2.7)

Κατά το άνοιγμα της διαπροσωπίας του αλγορίθμου, ο χρήστης μπορεί να διακρίνει τον διαχωρισμό τις εφαρμογής στους δυο βασικούς τομείς της. Τον τομέα όπου γίνεται η εκτέλεση της εφαρμογής για την αναφορά και (1) και στον δεύτερο τομέα την υπολογιστική μηχανή για στιγμιαίους υπολογισμούς κάποιου αποτυπώματος (5). Στον τομέα εκτέλεσης αναγνωρίζουμε δυο σημεία, το σημείο Source (2) και το σημείο Destination (4), και το κουμπί (3). Τα σημεία Source και Destination αποτελούνται αντίστοιχα από ένα πεδίο κειμένου(text field), που είναι Edit Disable και ένα κουμπί που επονομάζετε Browse. Όταν πατηθεί το κουμπί Browse και στα δυο σημεία, ανοίγει ένα παράθυρο System Explorer για επιλογή αρχείου. To αρχείο του Source είναι το αρχείο DroneData.txt όπου περιέχει τα δεδομένα εισόδου, Το αρχείο του destination αντιπροσωπεύει το αρχείο AreaReport.txt όπου θα αποθηκευτεί η αναφορά των αποτελεσμάτων. Όταν επιλεχθεί το αρχείο του αναγράφεται το απόλυτο μονοπάτι στο πεδίο κειμένου για να γίνει προφορική εξέταση αν το αρχείο είναι αυτό που θέλει ο χρήστης. Όταν επιλεχθούν και τα δυο αρχεία τότε το κουμπί Report θα μπορεί να ενεργοποιηθεί για να γίνει παραγωγή της αναφοράς. Ο τομέας 2 περιλαμβάνει 3 κατηγορίες Δεδομένων εισόδου την Image Specification (7), την Drone’s Camera Specification (9) και την Advanced Drone’s Camera Specification (6). Κάθε κατηγορία περιλαμβάνει συγκεκριμένα πεδία εισόδου. Όλα τα πεδία εισόδου είναι υποχρεωτικά να καταγραφούν και επίσης υπάρχει type validation για τα πεδία που πρέπει να περιέχουν μεταβλητές double. Όταν τα πεδία συμπληρωθούν τότε ο χρήστης πατά το κουμπί Calculate και γίνεται ο υπολογισμός των αποστάσεων και εμφανίζονται στα αντίστοιχα πεδία κειμένου όπως φαίνεται στο κουτί (8) της εικόνας.



(Εικ.2.8.)

Ο Editor διαχωρίζεται σε 4 βασικούς τομείς. Στον πρώτο τομέα (1) βρίσκεται το πάνελ το οποίο βλέπουμε το αρχείο και μπορούμε να το επεξεργαστούμε αντίστοιχα, εφόσον κληθεί κάποια add. Τα δεδομένα εμφανίζονται σε αυτό. Στον δεύτερο τομέα (2) ανοίγουμε το αρχείο για να μπορούμε να το επεξεργαστούμε πατώντας το κουμπί open (3). Εάν το αρχείο είναι καινούργιο η πρέπει να επαναδημιουργιθεί πατούμε το κουμπί PREPARE FILE (4) και αυτόματα αχειροποίητε με την keyword του αρχείου «DroneData:». Στον τρίτο τομέα (5) μπορούμε να κάνουμε add Manual όπου συμπληρώνουμε τα πεδία και πατούμε το κουμπί ADD (6) για να προσθέσουμε τα δεδομένα. Στον τέταρτο τομέα (7) γίνεται η Upload Add, επιλέγοντας το αρχείο της εικόνας πατώντας το κουμπί open (8) και μετά το κουμπί add. Όταν τελειώσουμε με την επεξεργασία πατούμε το κουμπί Save για να αποθηκεύσουμε το αρχείο (9).

**Κεφάλαιο 3**

**Εφαρμογή Δημιουργίας Χάρτη από Φωτογραφίες UAV**

3.1 Εφαρμογή 26

3.1.1 Τύπος αντικειμένου BlackBox 27

3.1.2 Ψευδοκώδικας Αλγορίθμου Καμβά 29

3.1.3 Λεπτομερής Περιγραφή Αλγορίθμου Καμβά 30

3.1.4 Ψευδοκώδικας Αλγορίθμου Stitching 35

3.1.5 Λεπτομερής Περιγραφή Αλγορίθμου Stitching 36

3.1.6 Βοηθητική εφαρμογή, Δεδομένα Εισόδου και Εγκυρότητα τους 38

3.1.7 Δεδομένα Εξόδου και Αρχεία Αναφορών 41

3.2 Επεξήγηση Γραφικής Διαπροσωπία Χρήστη 42

**3.1 Εφαρμογή**

Η Εφαρμογή Δημιουργίας Χάρτη έχει από φωτογραφίες UAV είναι όπως προαναφέρθηκε και στο πρώτο κεφάλαιο είναι μεγάλης σημασίας για τον λόγο ότι έχει σκοπό να βελτιστοποιήσει την διαδικασία των ερευνητικών ομάδων, όσο αφορά τον χρόνο και την ποιότητα απόδοσης τους. Η εφαρμογή αποτελείτε από την χρήση δύο αλγορίθμων οι οποίοι έχουν σκοπό την δημιουργία μιας ενιαίας εικόνας-χάρτη, από την συλλογή των φωτογραφιών που συλλέχθηκαν από μια αποστολή του Μη Επανδρωμένου Αεροσκάφους.

Οι δυο αλγόριθμοι έχουν σαν είσοδο ένα αρχείο κειμένου με ονομασία DroneRoute.txt το οποίο προσομοιώνει την πτήση του UAV, τον κατάλογο των φωτογραφιών όπου θα επεξεργαστούν, και την χρήση κάποιων τύπων αντικειμένων οι οποίες έχουν ζωτική σημασία για την επεξεργασία και το τελικό αποτέλεσμα της εφαρμογής. Επίσης για κάθε φάση του αλγορίθμου εξάγεται και μια επιπρόσθετη αναφορά οι οποία περιέχει πληροφορίες όσο για την απόδοση όσο και για την εξέταση και αποσφαλμάτωση των δεδομένων. Εφόσον κάποιος από τους αλγόριθμους εκτελεσθεί η εφαρμογή παράγει τον χάρτη και των αποθηκεύει για μεταγενέστερη χρήση, έρευνα και ανάλυση.

Η εφαρμογή παρουσιάζεται στον χρήστη, με την χρήση Γραφικής διαπροσωπίας(GUI). Επίσης η εφαρμογή περιλαμβάνει βοηθητική εφαρμογή για την δημιουργία και την εγκυρότητα του DroneRoute.txt που χρησιμοποιείται σαν είσοδος για τον υπολογισμό. Ο πηγαίος κώδικας της εφαρμογής μπορεί να βρεθεί στο GitHub ενώ σημαντικά κομμάτια του αλγορίθμου αναρτούνε στο Παράρτημα Β.

**3.1.1 Τύπος αντικειμένου BlackBox**

Το βασικότερο στοιχείο όσο αφορά την εκτέλεση των δύο αλγορίθμων είναι η χρήση του αντικειμένου BlackBox. Το BlackΒox συμβολικά ονομάστηκε έτσι για τον λόγο ότι προσομοιώνει τις πληροφορίες της πτήσης του Drone και μέσα από αυτό μπορούμε να αναλύσουμε τα δεδομένα, όπως το μαύρο κουτί του αεροπλάνου.

Σαν τύπος δεδομένων το αντικείμενο αυτό, απαρτίζεται από κάποιες μεταβλητές και κάποιές σταθερές :

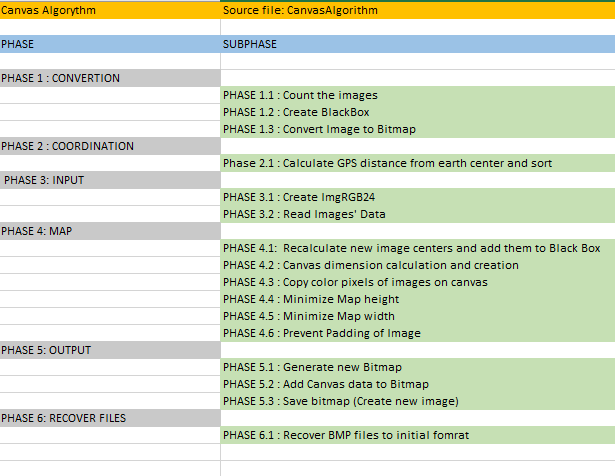
* **private** **final** **static** **int** ***BOX\_SIZE*** = 9 ⇨ Σταθερά για την ένδειξη του μεγέθους του πίνακα.
* **private** **final** **static** **int** ***IMAGE\_FORMAT*** = 0 ⇨ Σταθερά ένδειξης στην θέση 0 του πίνακα βρίσκεται η πληροφορία που αφορά το αρχικό είδος της εικόνας (π.χ. JPG).
* **private** **final** **static** **int** ***IMAGE\_TITLE*** = 1 ⇨ Σταθερά ένδειξης στην θέση 1 του πίνακα βρίσκεται η πληροφορία που αφορά τον τίτλο του αρχείου.
* **private** **final** **static** **int** ***LATITUDE*** = 2 ⇨ Σταθερά ένδειξης στην θέση 2 του πίνακα βρίσκεται η πληροφορία που αφορά την συντεταγμένη του γεωγραφικού πλάτους τη στιγμή που λήφθηκε η φωτογραφία σε μοίρες.
* **private** **final** **static** **int** ***LONGITUDE*** = 3 ⇨ Σταθερά ένδειξης στην θέση 3 του πίνακα βρίσκεται η πληροφορία που αφορά την συντεταγμένη του γεωγραφικού μήκους τη στιγμή που λήφθηκε η φωτογραφία σε μοίρες.
* **private** **final** **static** **int** ***CENTER\_X*** = 4 ⇨ Σταθερά ένδειξης στην θέση 4 του πίνακα βρίσκεται η πληροφορία που αφορά την συντεταγμένη χ του κέντρου στο καμβά (μόνο για αλγόριθμο Καμβά).
* **private** **final** **static** **int** ***CENTER\_Y*** = 5 ⇨ Σταθερά ένδειξης στην θέση 5 του πίνακα βρίσκεται η πληροφορία που αφορά την συντεταγμένη ψ του κέντρου στο καμβά (μόνο για αλγόριθμο Καμβά).
* **private** **final** **static** **int** ***INTERCECT\_DISTANCE*** = 6 ⇨ Σταθερά ένδειξης στην θέση 6 του πίνακα βρίσκεται η πληροφορία που αφορά την απόσταση της φωτογραφίας από το σημείο τομείς του ισημερινού και του πρώτου μεσημβρινού με γεωγραφικές συντεταγμένες (0,0) σε απόσταση μέτρων.
* **private** **final** **static** **int** ***IMAGE\_HEIGHT*** = 7 ⇨ Σταθερά ένδειξης στην θέση 7 του πίνακα βρίσκεται η πληροφορία που αφορά το ύψος της εικόνας σε pixels.
* **private** **final** **static** **int** ***IMAGE\_WIDTH*** = 8 ⇨ Σταθερά ένδειξης στην θέση 8 του πίνακα βρίσκεται η πληροφορία που αφορά το πλάτος της εικόνας σε pixels.
* **private** String[][] BlackBoxArray ⇨ Ο πίνακας των δεδομένων. Το ποσό των γραμμών του πίνακα καθιστά και το ποσό των φωτογραφιών που θα επεξεργαστούν.

Το BlackBox όπως προαναφέραμε, έχει σημαντικό ρόλο στην εκτέλεση της εφαρμογής. Σε θεωρητικό επίπεδο συλλέγει πληροφορίες και βοηθά τους αλγόριθμούς για να βελτιστοποιηθούνε όσο αφορά την επεξεργασία των εικόνων, κρατώντας κρίσιμες πληροφορίες όπως τα κέντρα των εικόνων για την τοποθέτηση τους στον καμβά, τον υπολογισμό της απόστασης από το σημείο τομής των πρώτου μεσημβρινού και του ισημερινού, καθορίζοντας σε πιο τεταρτημόριο βρισκόμαστε. Επίσης διαθέτει το όνομα αρχείου της κάθε εικόνας, το αρχικό format της για να γίνει επαναφορά της στο τέλος, καθώς και το πλάτος και ύψος της εικόνας σε pixels για να γίνει υπολογισμός του μεγέθους του καμβά. Επίσης μπορεί να χρησιμοποιηθεί και σαν ιστορικό της εκτέλεσης του αλγορίθμου, καθώς και σαν μονάδα αποσφαλμάτωσης για τον έλεγχο τον υπολογισμών κατά την διάρκεια της επεξεργασίας.

Η κλάση με τον πηγαίο κώδικα του BlackBox βρίσκεται αναρτημένη στο Παράρτημα Β.

**3.1.2 Ψευδοκώδικας Αλγορίθμου Καμβά**

Ο αλγόριθμος του καμβά αποτελείτε από 6 βασικές φάσεις οι οποίες διαχωρίζονται σε υποφάσεις η κάθε μια αντίστοιχα όπως φαίνεται και στον πίνακα πιο κάτω στην Εικόνα Εικ. 3.1. Η εκτέλεση του αλγορίθμου γίνεται σε συνεργασία με τον αλγόριθμο του αποτυπώματος που περιεγράφηκε στο Υποκεφάλαιο 2.1. Για να γίνει εφικτή η εκτέλεση του αλγορίθμου χρειάζεται σαν δεδομένα εισόδου το φάκελο με τις εικόνες που θα επεξεργαστούν, το DroneRoute.txt και το αρχείο DroneData.txt. Σαν δεδομένο εξόδου έχουμε τις αναφορές του BlackBox και τον χάρτη.



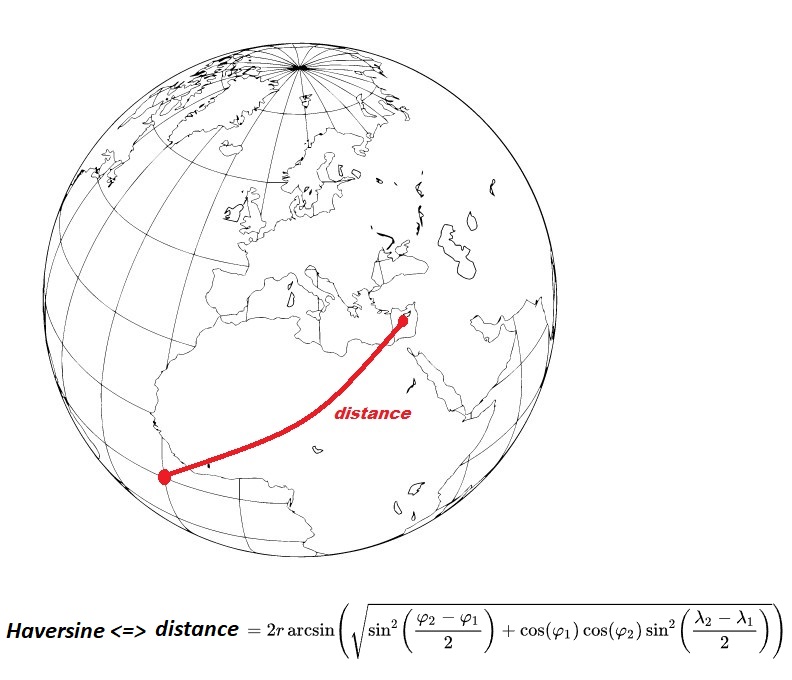
(Εικ.3.1)

**3.1.3 Λεπτομερής Περιγραφή Αλγορίθμου Καμβά**

Η πρώτη φάση του αλγορίθμου του Καμβά αποτελείτε από 3 υποφάσεις οι οποίες αποτελούνται από το μέτρημα των εικόνων που θα επεξεργαστούν κατά την εκτέλεση του Αλγορίθμου, την δημιουργία του αντικειμένου BlackBox, και την μετατροπή των εικόνων σε αρχεία .bmp (bitmap image file).

Για την καταμέτρηση των εικόνων, δημιουργήθηκε μια συνάρτηση η οποία ανοίγει τον φάκελο τον οποίο βρίσκονται μέσα οι αποθηκευμένες εικόνες, και με βάση το extension καταμετρά πόσες εικόνες βρίσκονται μέσα στον φάκελο. Η καταμέτρηση γίνεται μόνο σε εικόνες τύπου .bmp και .JPG σε προσωρινή φάση. Στο πέρας της καταμέτρησης ο αριθμός που θα υπολογιστή, είναι ο αριθμός των γραμμών για την δημιουργία του αντικειμένου BlackBox η οποία είναι και η επόμενη υποφάση. Κατά την δημιουργία του BlackBox, εκτελείτε και η ανάγνωση του αρχείου DroneRoute.txt , από την οποία γίνετε ανάκτηση των δεδομένων του IMAGE\_FORMAT, IMAGE\_TITLE και των συντεταγμένων των εικόνων , LATITUDE και LONGITUDE, όπου και αποθηκεύονται και στις κατάλληλες θέσεις του BLACKBOXARRAY. Σε περίπτωση ανυπαρξίας, η λάθους ανοίγματος του αρχείου DroneRoute.txt ή ακόμα και αν ο αριθμός των εικόνων που καταμετρήθηκαν δεν αντιστοιχεί στον αριθμό των εικόνων που αναγράφονται στο αρχείο εγείρεται εξαίρεση και ενημερώνει με κατάλληλο μήνυμα των χρήστη. Για την λήξη της πρώτης φάσης γίνεται η μετατροπή των αρχείων σε .bmp για τους λόγους που αναφέρονται και στο ερευνητικό υπόβαθρο του Κεφαλαίου 1 πιο αναλυτικά και αντικαθιστάτε το όνομα του αρχείου με το καινούργιο extension .bmp σε περίπτωση που το αυθεντικό αρχείο εικόνας ήταν .JPG.

Στη δεύτερη φάση του αλγορίθμου γίνεται ο υπολογισμός της αποστάσεις μεταξύ των γεωγραφικών συντεταγμένων της εικόνας και του κέντρου τομής του πρώτου μεσημβρινού και του ισημερινού όπως φαίνεται στην πιο κάτω εικόνα (Εικ.3.2) που τίθεται ως παράδειγμα.

****

(Εικ.3.2)

Δεν χρησιμοποιείται η γνωστή φόρμουλα της Ευκλείδειας απόστασης μεταξύ δυο σημείων για τον λόγο της σφαιρικότητας στην επιφάνεια της γης. Έτσι για τον υπολογισμό της απόστασης χρησιμοποιήθηκε η φόρμουλα Haversine ,η οποία επεξηγείτε λεπτομερώς στο Ερευνητικό Υπόβαθρο και κομμάτι του πηγαίου κώδικα για την εκτέλεση της υπάρχει στο Παράρτημα Β, κάνοντας χρήση της για τον υπολογισμό των αποστάσεων σε όλες τις εικόνες και αποθηκεύουμε τις αποστάσεις στην στήλη INTERCECT\_DISTANCE, για κάθε εικόνα αντίστοιχα. Ακολούθως, με την χρήση του αλγορίθμου quick sort ταξινομούμε τον πίνακα BLACKBOXARRAY με βάση την απόσταση σε αύξουσα σειρά αναγνωρίζοντας έτσι πια εικόνα είναι πιο κοντά στο κέντρο τομής (0,0) και συνεπώς η εικόνα που θα βρίσκεται πιο κοντά στο κατω-αριστερα σημείο του καμβά.

Εφόσον μέχρι στιγμής έχουμε ενημερώσει το BLACKBOXARRAY και με τις αποστάσεις και το ταξινομήσαμε, προχωράμε στην τρίτη φάση του αλγορίθμου που έχει να κάνει σχετικά με την ανάγνωση των εικόνων.

Στην τρίτη φάση γίνεται χρήση ενός αντικειμένου imgRGB24 ,που αντιπροσωπεύει την εικόνα BMP RGB24 (24-bit bitmap) σύμφωνα και με το ερευνητικό υπόβαθρο. Το αντικείμενο δημιουργείτε στην αρχή της φάσης και έχει ως πεδία τις ακόλουθες μεταβλητές:

**private** String[] ImageTitle ⇨ Ο τίτλος του αρχείου της εικόνας.

/////////////////// BMP HEADER///////////////////////////////////////////////////////////////////////////

**private** String Signature ⇨ Το magic number της εικόνας.

**private** String Size ⇨ To μέγεθος του bmp file.

**private** String AppSpec1 ⇨ Application Specific (Δεν χρησιμοποιείτε συνήθως).

**private** String AppSpec2 ⇨ Application Specific (Δεν χρησιμοποιείτε συνήθως).

**private** String Offset ⇨ Σε πιο offset αρχίζουν τα δεδομένα.

/////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

///////////////////// DIB HEADER///////////////////////////////////////////////////////////////////////////

**private** String DIB size ⇨ Μέγεθος του DIB header.

**private** String Width ⇨ Πλάτος της εικόνας σε pixels.

**private** String Height ⇨ Ύψος της εικόνας σε pixels.

**private** String NumofPlanes ⇨ Ο αριθμός των planes.

**private** String BitperPx ⇨ Πόσα bit αντιστοιχούν σε εάν pixel.

**private** String CompressionArray ⇨ Τρόπος συμπίεσης του πίνακα των pixel.

**private** String RawBitmapData ⇨ Μέγεθος πίνακα χρωματότητας.

**private** String PrintResolutionWidth ⇨ Ανάλυση εικόνας κατά πλάτος.

**private** String PrintResolutionHeight ⇨ Ανάλυση εικόνας κατά ύψος.

**private** String NumOfColorsPallete ⇨ Αριθμός των χρωμάτων τις παλέτας.

**private** String ImportantColors ⇨ Σημαντικοί χρωματισμοί.

///////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

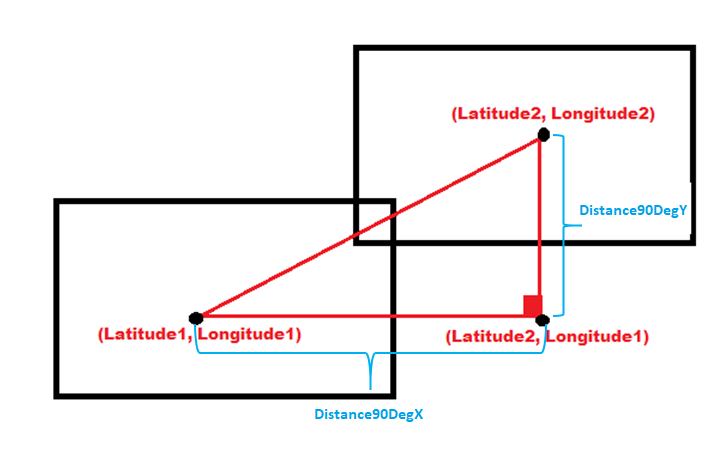
///////////////////// BITMAP DATA/////////////////////////////////////////////////////////////////////////

**private** String[][] ColorArray ⇨ Πίνακας των pixel (χρωματισμοί).

////////////////////////////////////////////////////////////////////////////////////////////////////////////////////////

Ο πηγαίος κώδικας του αντικειμένου βρίσκεται στο Παράρτημα Β. Η δημιουργία του αντικειμένου αυτού έχει ως σκοπό την αποθήκευση του πίνακα των χρωμάτων τις εικόνας για να μπορεί να επεξεργαστεί μετά. Για κάθε εικόνα δημιουργείτε και ένα αντικείμενο. Μετά την δημιουργία των ImgRGB24 και στην συνέχεια γίνεται ανάγνωση των εικόνων από τα αρχεία τους. Κατά την ανάγνωση αριχκοποιούμε ένα FileInputStream για το αρχείο της εικόνας και με την βοήθεια τις συνάρτησης convertByte2String, η οποία μετατρέπει ένα πίνακα από byte στο αντίστοιχο 16αδικό σύστημα (ΗΕΧ), προσθέτει τις πληροφορίες στο κατάλληλο πεδίο του ImgRGB24. Όταν ο reader φτάσει στο πίνακα των χρωματισμών χρησιμοποιεί μια έξτρα μέθοδο, η οποία αντιστρέφει τον πίνακα των pixel για να διαβαστεί αντίθετα από την δομή της εικόνας, για την καλύτερη επεξεργασία της. Όταν τερματίσει η διαδικασία διαβάσματος, το ύψος και το πλάτος της εικόνας αποθηκεύονται στις στήλες IMAGE\_HEIGHT και IMAGE\_WIDTH του πίνακα BLACKBOXARRAY αντίστοιχα για κάθε εικόνα.

Η τέταρτη φάση του αλγορίθμου του καμβά και η πιο κρίσιμη είναι η δημιουργία της τελικής εικόνας. Πρώτα πρέπει να γίνει κατανομή των κέντρων τις κάθε εικόνας, για τον λόγο αυτό δημιουργήθηκε η μέθοδος imageCenterAdjustment, όπου ακολουθεί τα εξής βήματα. Πρώτα γίνεται ανασκόπηση του πίνακα για να πάρουμε την εικόνα στην πρώτη γραμμή του BLACKBOXARRAY, δηλαδή αυτή που σύμφωνα με την ταξινόμηση του πίνακα είναι πιο κοντινή στο σημείο τομής (0,0) του ισημερινού και πρώτου μεσημβρινού. Έπειτα καλούμε την εφαρμογή για τον υπολογισμό των αποστάσεων πλάτους και μήκους των αποτυπωμάτων των εικόνων δίνοντας σαν είσοδο το αρχείο DroneData.txt (Κεφ. 2). Εωσότου υπολογιστούν τα αποτυπώματα καταχωρούμε στις κατάλληλες τοπικές μεταβλητές. Αμέσως μετά ανακτούμε από το BlackBox τα γεωγραφικά πλάτη και μήκη των εικόνων καθώς και οι διαστάσεις τους αντίστοιχα. Μετά σειρά έχει ο υπολογισμός του scale\_x που αφορά τα πλάτη και το scale\_y που αφορά το ύψος των εικόνων, δηλαδή Scalex = πλάτος εικόνας σε pixel / απόσταση πλάτους αποτυπώματος σε μέτρα και Scaley = ύψος εικόνας σε pixel / απόσταση ύψους αποτυπώματος σε μέτρα. Αυτοί οι υπολογισμοί μας δίνουν πόσα pixel αντιστοιχούν σε ένα μέτρο στις εικόνες. Το επόμενο στάδιο είναι ο υπολογισμός των αποστάσεων που φαίνονται στην ακολουθεί εικόνα χρησιμοποιώντας την Haversine Formula, όπως φαίνονται στην εικόνα (Εικ.3.3)



(Εικ.3.3)

Γνωρίζοντας τις αποστάσεις και πολλαπλασιάζοντας με το κατάλληλο scale για την κάθε απόσταση, παίρνουμε την απόσταση σε pixel που απέχουν τα δύο κέντρα μεταξύ τους δηλαδή το κέντρο τους στον καμβά. Τα κέντρα που υπολογίζονται αποθηκεύονται στο BLACKBOXARRAY στις θέσεις CENTER\_X και CENTER\_Y τερματίζοντας έτσι την μέθοδο και συνεπώς τον αλγόριθμο υπολογισμού.

Στην επόμενη υποφάση της δημιουργίας του Καμβά. πρέπει να υπολογιστεί το μέγεθος του καμβά. Το μέγεθος αρχικοποιείται σύμφωνα με την ποσότητα των εικόνων. Λόγο του ότι υπάρχει επισκιάσει μεταξύ των εικόνων που λαμβάνονται από το Drone. Γνωρίζουμε ήδη ότι η εικόνες τοποθετούνται με ασφάλεια σε ένα καμβά (Image Width\*2 X Image Heigth\*2) διαστάσεων. Στην επόμενη υποφάση με την χορήγηση των IMAGE\_TITLE, CENTER\_X και CENTER\_Y από το BLACKBOXARRAY.

Στην επόμενη υποφάση γίνετε αντιγραφή των πινάκων χρωματισμών των εικόνων στον Καμβά. Με τη χρήση της συνάρτησης System.arrayCopy() γίνεται αντιγραφή των πινάκων στις συντεταγμένες τις οποίες πήραμε κατά την εξαγωγή των κέντρων CENTER\_X και CENTER\_Y κατά την προηγούμενη φάση. Κατά την αντιγραφή των πινάκων γίνεται απαλοιφή των δεδομένων των Αντικειμένων ImgRGB24 για να απελευθερωθεί μνήμη στο σύστημα και να μην οδηγηθεί σε Εξαίρεση λόγο έλλειψής μνήμης.

Κατά της επόμενες δύο φάσεις γίνεται σμίκρυνση του Καμβά σε κατά μήκος και κατά πλάτος. Επεξηγηματικά κατά την δημιουργία του καμβά υπάρχουν στήλες και γραμμές, οι οποίες δεν περιέχουν σημαντικές πληροφορίες παρα μόνο pixels με λευκό χρώμα. Για τον λόγο αυτό αποκόβουμε αυτές τις στήλες και γραμμές από την τελική εικόνα για να βελτιστοποιήσουμε το μέγεθος και την εκτέλεση του αλγορίθμου.

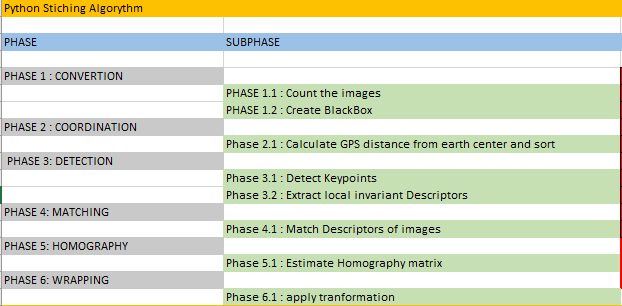
Στην τελευταία υποφάση λόγο του τύπου εικόνας που επιλέχθηκε, υπάρχει υ παρουσία “padding”. Το Padding υπάρχει στη δομή δεδομένων της εικόνας για να συγκρατά στοιχισμένα τα pixels. Για να το αποφύγουμε αυτό απλά αφαιρούμε της στήλες μέχρι το πλάτος του πίνακα να έχει υπόλοιπο 0 εάν διαιρεθεί με το 3 δηλαδή να είναι πολλαπλάσιο του 3.

Στην πέμπτη φάση του αλγορίθμου επεξεργαζόμαστε τα δεδομένα τις εικόνας μας όσο αφορούν τις επικεφαλίδες ώστε να αντιστοιχούν στο μέγεθος του καμβά και το μέγεθος του αρχείου. Μετατρέπουμε σειριακά την επικεφαλίδα και τα χρώματα του πίνακα από 16αδικό σε byte και την γράφουμε στο καινούργιο αρχείο της εικόνας και το αποθηκεύουμε.

Στη τελευταία φάση του αλγορίθμου μετατρέπουμε της εικόνες στο αρχικό τους τύπο αρχείου, δηλαδή εάν οι εικόνες ήταν tτύπου JPEG και τις μετατρέψαμε σε BMP τότε τις επαναφέρουμε σε JPEG. Έτσι τερματίζει η εκτέλεση του αλγορίθμου

**3.1.4 Ψευδοκώδικας Αλγορίθμου Stitching**

Ο αλγόριθμος του Stitching αποτελείτε από 6 βασικές φάσεις οι οποίες διαχωρίζονται σε υποφάσεις η κάθε μια αντίστοιχα όπως φαίνεται και στον πίνακα πιο κάτω στην Εικόνα Εικ. 3.4. Οι πρώτες δυο φάσεις του αλγορίθμου του Stitching είναι πανομοιότυπες με αυτές της εκτέλεσης του Αλγορίθμου του Καμβά, περιλαμβάνουν την καταμέτρηση των εικόνων και την δημιουργία του BlackBox αντικειμένου το οποίο περιέχει σημαντικές πληροφορίες όσο αφορά την εκτέλεση του Αλγορίθμου. Η φάση τρία διαφοροποιείτε όσο αφορά την ταξινόμηση των δεδομένων στον πίνακα ΒlackBoxArray του αντικειμένου, ως προς το γεωγραφικό πλάτος και όχι την INTERCECT\_DISTANCE. Η εκτέλεση του αλγορίθμου γίνεται με την χρήση “exec” από την Java για εκτέλεση στην πλατφόρμα της Python. Για να γίνει εφικτή η εκτέλεση του αλγορίθμου χρειάζεται σαν δεδομένα εισόδου το φάκελο με τις εικόνες που θα επεξεργαστούν, το αρχείο DroneData.txt και ένα αρχείο κειμένου που περιλαμβάνει με τα μονοπάτια των τοποθεσιών των εικόνων, το οποίο δημιουργείτε αυτόματα. Σαν δεδομένο εξόδου έχουμε τις αναφορές του BlackBox και τον χάρτη.



(Εικ. 3.4)

**3.1.5 Λεπτομερείς Περιγραφή Αλγορίθμου Stitching**

Η πρώτη φάση του Αλγορίθμου του Stitching αποτελείτε από 2 πανομοιότυπες υποφάσεις, με τον Αλγόριθμο του Καμβά, οι οποίες εκτελούν το μέτρημα των εικόνων που θα επεξεργαστούν κατά την εκτέλεση του Αλγορίθμου Stitching και την δημιουργία του αντικειμένου BlackBox. Ο Αλγόριθμος αυτός δεν χρειάζεται μετατροπή των εικόνων σε αρχεία BMP, για τον λόγο ότι δεν θα πραγματοποιήσει την συνένωση με τον τρόπο του προηγούμενου Αλγορίθμου, και έτσι την παραβλέπει.

Η καταμέτρηση γίνεται με το κάλεσμα της μεθόδου καταμέτρησης του Καμβά και έτσι γίνεται μόνο σε εικόνες τύπου .bmp και .JPG. Στο πέρας της καταμέτρησης ο αριθμός που θα υπολογιστή, όπως είδαμε και στο προηγούμενο κεφάλαιο , είναι ο αριθμός των γραμμών για την δημιουργία του αντικειμένου BlackBox, η οποία είναι και η επόμενη υποφάση. Κατά την δημιουργία του BlackBox, εκτελείτε και η ανάγνωση του αρχείου DroneRoute.txt , από την οποία γίνετε ανάκτηση των δεδομένων του IMAGE\_FORMAT, IMAGE\_TITLE και των συντεταγμένων των εικόνων , LATITUDE και LONGITUDE, όπου και αποθηκεύονται και στις κατάλληλες θέσεις του BLACKBOXARRAY. Σε περίπτωση ανυπαρξίας, η λάθους ανοίγματος του αρχείου DroneRoute.txt ή ακόμα και αν ο αριθμός των εικόνων που καταμετρήθηκαν δεν αντιστοιχεί στον αριθμό των εικόνων που αναγράφονται στο αρχείο εγείρεται εξαίρεση και ενημερώνει με κατάλληλο μήνυμα των χρήστη. Εφόσον Δεν γίνεται κάποια μετατροπή στα αρχεία των εικόνων η πρώτη φάση τερματίζει σε αυτό το σημείο.

Στη δεύτερη φάση του Αλγορίθμου του Stitching και με την χρήση του αλγορίθμου quick sort ταξινομούμε τον πίνακα BLACKBOXARRAY με βάση το μικρότερο Latitude δηλαδή την εικόνα που βρίσκετε πιο “ Δυτικά ”, στην λίστα τον φωτογραφιών. Ο λόγος που γίνεται αυτή η ταξινόμηση είναι γιατί ο αλγόριθμος που χρησιμοποιείτε για την συνένωση κάνει μόνο δεξιά μετατόπιση, στις εικόνες που επεξεργάζονται.

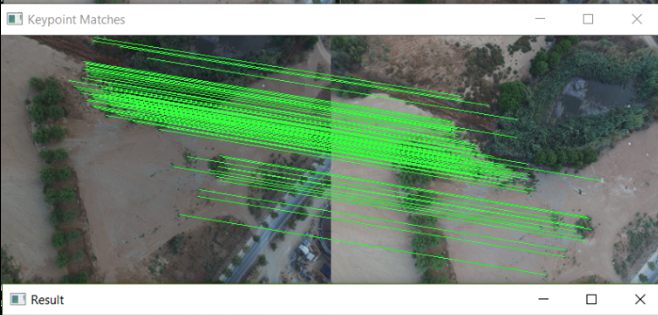
Στην τρίτη και τέταρτη την φάση του Αλγορίθμου πρέπει να γίνει εντοπισμός των keypoints και αντιστοίχιση τους . Για να γίνει ο εντοπισμός και η εξαγωγή των περιγραφέων όπως φαίνεται στην Εικόνα Εικ. 3.5, η οποία πάρθηκε από την ιστοσελίδα της OpenCV όπου υλοποιούνται οι συναρτήσεις της SIFT όπως και αυτή που πάραξε την εικόνα αυτή χρησιμοποιώντας την συνάρτηση,

img=cv2.drawKeypoints(gray,kp,img,flags=cv2.DRAW\_MATCHES\_FLAGS\_DRAW\_RICH\_KEYPOINTS)

γίνεται η χρήση του περιγραφέα SIFT του opencv\_contrib, και αργότερα η αντιστοίχισή τους όπως στην Εικόνα Εικ. 3.6 . Ο περιγραφέας SIFT, είναι ένας πολύ ισχυρός αλγόριθμος CV για την αναγνώριση και την αντιστοίχιση των keypoint δυο εικόνων. Η βασικές λειτουργίες του Αλγορίθμου SIFT αναρτιούνται στο Παράρτημα Β.



(Εικ. 3. 5)



(Εικ. 3.6)

Στην Πέμπτη φάση του αλγορίθμου που έχουμε την δημιουργία του πίνακα ομογραφίας. Για να δημιουργηθεί ο πίνακας ομογραφίας. Ο πίνακας ομογραφίας θα χρησιμοποιήσει τα σημεία αντιστοίχισης, για να εκτιμήσει ένα σχετικό μετασχηματισμό προσανατολισμού στις δύο εικόνες, δηλαδή Εάν η εικόνα χρειάζεται να περιστραφεί στις τρεις διαστάσεις για να εφαρμόσουν όλες οι αντιστοιχίες και να πετύχει το Stitching. Εφόσον ο πίνακας δημιουργηθεί βάση του αλγορίθμου RANSAC, οι εικόνες είναι έτοιμες να συνενωθούν.

Στη τελευταία φάση του αλγορίθμου γίνεται η συνένωση. Κατά την συνένωση γίνετε στρέβλωση της εικόνας έτσι ώστε να μπορέσει να εφαρμόσει περισσότερές αντιστοιχίσεις.

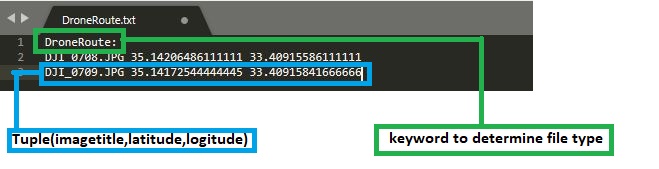
**3.1.6 Βοηθητική εφαρμογή, Δεδομένα Εισόδου και Εγκυρότητα τους**

Για την λειτουργεία των αλγορίθμων χρειαζόμαστε να ξέρουμε συγκεκριμένα κάποια δεδομένα. Αυτά τα δεδομένα είναι το όνομα του αρχείου της εικόνας, το γεωγραφικό πλάτος και το γεωγραφικό μήκος που λήφθηκε η φωτογραφία. Αυτά τα στοιχεία μπορούμε να τα συλλέξουμε μέσω κάποιου αρχείου Drone data log αρχείου το οποίο δημιουργείτε από το σύστημα του Drone και αντιπροσωπεύει το ιστορικό λήψης των φωτογραφιών μιας αποστολής. Λόγο της διαφορετικότητας μεταξύ εταιριών που δημιουργούν τέτοια αρχεία , αποφασίστηκε να δημιουργηθεί μια βοηθητική εφαρμογή επονομαζόμενη «DroneRouteEditor», η οποία παράγει ένα αρχείο DroneRoute.txt το οποίο προσαρμόζει τα δεδομένα των αρχείων log του drone έτσι ώστε να μπορούν να διαβαστούν χωρίς σφάλματα από τους Αλγορίθμους .

Ο DroneRoute Editor αποτελείτε από 4 σημαντικούς τομείς, όπως φαίνεται και στην Εικόνα Εικ. 3.10 στο κεφάλαιο των διπροσωπιών. Μέσω αυτού του editor μπορεί ο χρήστης να δημιουργήσει ή να επεξεργαστεί αντίστοιχα ένα αρχείο DroneRoute.txt . Στην περίπτωση της δημιουργίας ο χρήστης έχει δυο επιλογές, την επιλογή manual add και την επιλογή Upload add. Στην επιλογή manual μπορεί να προσθέσει δεδομένα εισόδου απλά γεμίζοντας τα αντίστοιχα πεδία, ακολουθώντας την ίδια λογική με τον DroneDataEditror. Στην δεύτερη επιλογή μπορεί να ανεβάσει μια εικόνα και αυτόματα τα στοιχεία θα ανακτηθούν στα πεδία εισόδου.

Πληροφοριακά η φωτογραφίες που μπορούν να γίνουν upload πρέπει να είναι εικόνες που προέρχονται από κάποιο drone έτσι ώστε να περιέχουν και τις ειδικές πληροφορίες για ανάκτηση όπως GPS συντεταγμένες κ.τ.λ. Κομμάτι της GUIProcedures.java βρίσκεται στο Παράρτημα Β.

Το αρχείο αυτό αποτελείτε από δυο σημαντικά συστατικά. Το πρώτο συστατικό είναι η λέξη-κλειδί (keyword) «DroneRoute:», η οποία έχει δύο ρόλους. Ξεχωρίζει το είδος του αρχείου, εάν αυτό είναι το αρχείο που έχει τα απαραίτητα δεδομένα για την είσοδο της εφαρμογής για την εκτέλεση του αλγορίθμου και επιπρόσθετα σηματοδοτεί την έναρξη του reader να ανακτήσει τα δεδομένα του αρχείου και να τα αναθέσει στις δομές του blackboxarray. Το δεύτερο συστατικό είναι οι πλειάδες(tuples) των δεδομένων. Κάθε πλειάδα αναπαριστά τα δεδομένα που θα χρειαστούν, καθώς και πληροφορίες που σηματοδοτούν την κάθε εικόνα που θα χρειαστεί να υπολογιστεί το αποτύπωμά της. Το ποσό των εικόνων που θα υπολογιστούν είναι το σύνολο των πλειάδων και κάθε πλειάδα αποτελείτε από τα πεδία (ImageTitle,Latidute,Logitude). Μια αναπαράσταση του αρχείου είναι όπως αυτή που βλέπουμε στην Εικόνα Εικ. 3.7



(Εικ. 3.7)

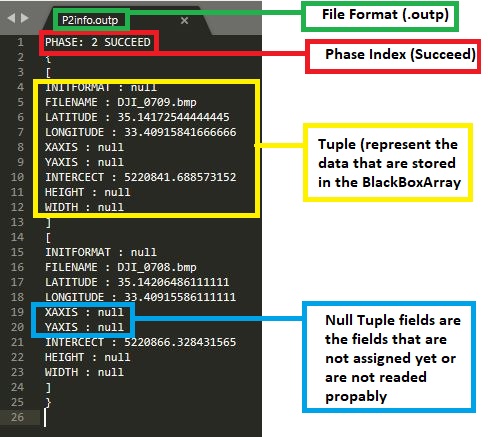
Για την εγκυρότητα του αρχείου λήφθηκαν υπόψη οι ακόλουθοι κανόνες :

1. Η λέξη κλειδί, η οποία πάντα πρέπει να βρίσκεται στην πρώτη γραμμή του αρχείου και σηματοδοτεί τον reader ότι αρχίζουν οι πλειάδες δεδομένων.
2. H λέξη κλειδί είναι η « DroneRoute: ».
3. Στην δεύτερη γραμμή του αρχείου πάντα βρίσκεται η πρώτη πλειάδα δεδομένων.
4. Το αρχείο πρέπει να περιέχει μόνο τις πλειάδες και την λέξη κλειδί.
5. Τα δεδομένα της πλειάδας πάντα πρέπει να είναι στη ακολουθία (imagetitle, latitude, longitude).
6. Οι τύποι δεδομένου της πλειάδας είναι αντίστοιχα (String, double, double).
7. Τα δεδομένα τις κάθε πλειάδας διαχωρίζονται με « » χωρίς επιπρόσθετα κενά ενδιάμεσα.
8. Στο τέλος τις κάθε πλειάδας υπάρχει πάντα χαρακτήρας «\n».
9. Στο τέλος του κάθε αρχείου υπάρχει χαρακτήρας EOF.

Κατά την επιλογή του αρχείου στην εφαρμογή του αλγορίθμου, ο validator του αρχείου, ενεργοποιείτε αυτόματα και ενημερώνει για την εγκυρότητα του αρχείου με το κατάλληλο μήνυμα. Εάν δεν είναι συμβατό το αρχείο ο validator εγείρει εξαίρεση και ενημερώνετε η διεπαφή αυτόματα.

**3.1.7 Δεδομένα Εξόδου και Αρχεία Αναφορών**

Μετά τον τερματισμό τις πρώτης, της δεύτερης και της τρίτης φάσης του Αλγορίθμου του Καμβά και μετά τον τερματισμό τις πρώτής και δεύτερης φάσης του αλγορίθμου του Stitching, ο Phase Executioner των δυο Αλγορίθμων αντίστοιχα, τυπώνει μια αναφορά όσο αφορά τα δεδομένα που περιέχονται στο BlackBox για την επικύρωση των δεδομένων σε περίπτωση σφάλματος, αλλά και για την στατιστική τους χρήση. Επίσης σε μεταγενέστερη έκδοση της εφαρμογής μπορεί να χρειαστούν και σαν αρχεία ιστορικού για την χρήση της εφαρμογής. Πιο κάτω στην Εικόνα Εικ. 3.8 βλέπουμε παράδειγμα από μια αναφορά στο μετά την φάση 2.

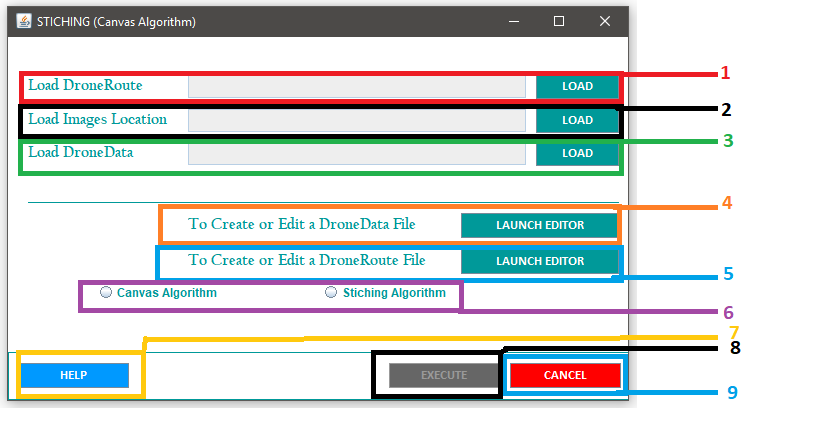


(Εικ.3.8)

Τα δεδομένο εξόδου που παράγεται από τους αλγορίθμους είναι ο Χάρτης.

**3.2 Επεξήγηση Γραφικής Διαπροσωπίας Χρήστη**

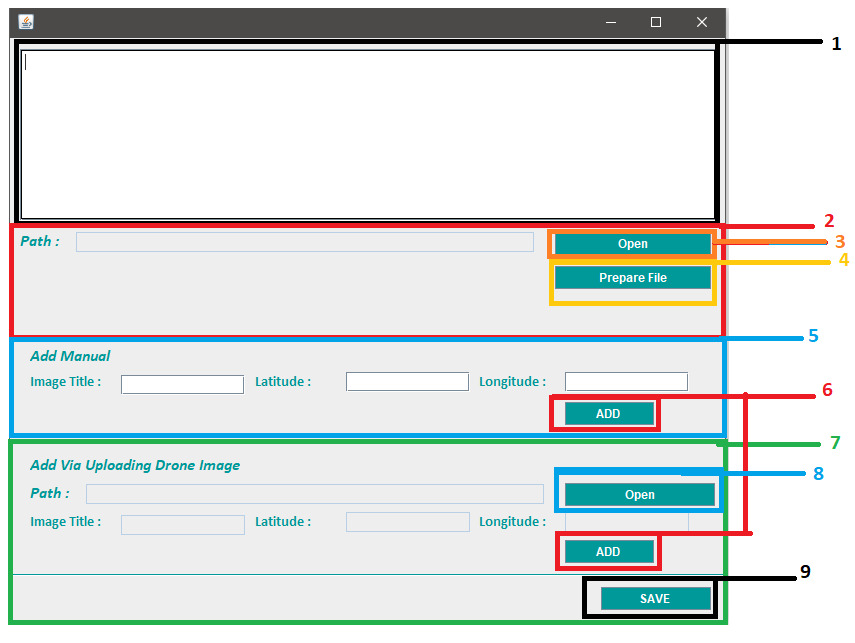
Για την αναπαράσταση της εφαρμογής και των βοηθητικών εφαρμογών δημιουργήθηκαν κάποιες γραφικές διαπροσωπίες χρήστη (GUI) στα οποία γίνεται αυτοματοποιημένη η εκτέλεση των Αλγορίθμων του Καμβά και του Stitching, καθώς ο χρήστης μπορεί να δημιουργήσει και να επεξεργαστή δια δραστικά το αρχείο DroneRoute.txt όπου περιέχει τα δεδομένα εισόδου για εκτέλεση της υπολογιστικής εφαρμογής. Καταρχάς, όπως βλέπουμε στην Εικόνα Εικ. 3.9 την γραφική διαπροσωπία για την εκτέλεση του αλγορίθμου και στην Εικόνα Εικ. 3.10 την βοηθητική εφαρμογή για δημιουργίας.

****

(Εικ. 3.9)

Κατά το άνοιγμα της διαπροσωπίας :

1. Γραμμή για την επιλογή μέσο ενός Browse Window την τοποθεσία που βρίσκεται το αρχείο DroneRoute.txt το αρχείο κατά την επιλογή του ελέγχεται για εγκυρότητα. Έγκυρο ή μη Εμφανίζεται μήνυμα στον χρήστη για εγκυρότητα ή εξαίρεση. Εάν το Αρχείο είναι έγκυρο τότε εμφανίζεται το μονοπάτι του στο πεδίο κειμένου.
2. Γραμμή για την επιλογή μέσο ενός Browse Window την τοποθεσία που βρίσκεται το ο κατάλογος των Εικόνων που θα επεξεργαστούν. Κατά την επιλογή του καταλόγου, εμφανίζεται το μονοπάτι του στο πεδίο κειμένου.
3. Γραμμή για την επιλογή μέσο ενός Browse Window την τοποθεσία που βρίσκεται το αρχείο DroneData.txt το αρχείο κατά την επιλογή του ελέγχεται για εγκυρότητα. Έγκυρο ή μη Εμφανίζεται μήνυμα στον χρήστη για εγκυρότητα ή εξαίρεση. Εάν το Αρχείο είναι έγκυρο τότε εμφανίζεται το μονοπάτι του στο πεδίο κειμένου.
4. Σε περίπτωση μη υπαρκτού ή μη συμβατού αρχείου DroneData, μπορεί ο χρήστης να καλέσει τον DroneDataEditor για να μπορεί να δημιουργήσει ένα συμβατό αρχείο.
5. Σε περίπτωση μη υπαρκτού ή μη συμβατού αρχείου DroneRoute, μπορεί ο χρήστης να καλέσει τον DroneRouteEditor για να μπορεί να δημιουργήσει ένα συμβατό αρχείο.
6. Επιλογή αλγορίθμου εκτέλεσης.
7. Κουμπί για βοήθεια, Αποτελείτε με πληροφορίες για της λειτουργίες.
8. “Disable” κουμπί εκτέλεσης, μόνο όταν επιλεχθεί αλγόριθμός και υπάρχει εγκυρότητα όσο αφορά τα μονοπάτια, μπορεί να πατηθεί και να εκτελεστεί ο αλγόριθμος που επιλέχθηκε.
9. Κουμπί για ακύρωση εκτέλεσης.



(Εικ. 3.10)

Ο DroneRouteEditor διαχωρίζεται σε 4 βασικούς τομείς. Στον πρώτο τομέα (1) βρίσκεται το πάνελ το οποίο βλέπουμε το αρχείο και μπορούμε να το επεξεργαστούμε αντίστοιχα, εφόσον κληθεί κάποια add ή φορτωθεί ένα αρχείο, τα δεδομένα εμφανίζονται σε αυτό. Στον δεύτερο τομέα (2) ανοίγουμε το αρχείο για να μπορούμε να το επεξεργαστούμε πατώντας το κουμπί open (3). Εάν το αρχείο είναι καινούργιο η πρέπει να επαναδημιουργιθεί πατούμε το κουμπί PREPARE FILE (4) και αυτόματα αρχικοποιείται με την keyword του αρχείου «DroneRoute:». Στον τρίτο τομέα (5) μπορούμε να κάνουμε add Manual όπου συμπληρώνουμε τα πεδία και πατούμε το κουμπί ADD (6) για να προσθέσουμε τα δεδομένα. Στον τέταρτο τομέα (7) γίνεται η Upload Add, επιλέγοντας το αρχείο της εικόνας πατώντας το κουμπί open (8) και μετά το κουμπί add. Όταν τελειώσουμε με την επεξεργασία πατούμε το κουμπί Save για να αποθηκεύσουμε το αρχείο (9).

**Κεφάλαιο 4**

**Πειραματική αξιολόγηση**

4.1 Σύγκριση Αλγορίθμων Δημιουργίας Χάρτη 45

4.2 Εκτέλεση Αλγορίθμων για Σύγκριση Χρόνου εκτέλεσης. 45

4.3 Εκτέλεση Αλγορίθμων με Εικόνες Διαφορετικών Μεγεθών 48

4.4 Εκτέλεση Αλγορίθμων με Διαφορετική Ποσότητα Εικόνων 49

4.5 Εκτέλεση Αλγορίθμων σε Διαφορετικές Συσκευές 51

4.6 Έλεγχος Μονάδων (JUNIT) 52

**4.1 Σύγκριση Αλγορίθμων Δημιουργίας Χάρτη**

Οι αλγόριθμοι ως αποτέλεσμα έχουν την δημιουργία κάποιου χάρτη μέσω των εικόνων που θα δοθούν ως είσοδο στην εφαρμογή για την δημιουργία του χάρτη. Σημαντικό για την σύγκριση των αλγορίθμων είναι η διαφορετικότητα όσο αφορά την γλώσσα που προγραμματίστηκαν. Ο αλγόριθμος του καμβά προγραμματίστηκε σε γλώσσα προγραμματισμού Java, ενώ ο αλγόριθμος του Stitching προγραμματίστηκε σε γλώσσα προγραμματισμού Python. Επιπρόσθετα για την σύγκριση των αποτελεσμάτων έγινε και η ανάλυση των αλγορίθμων πειραματικά όπως και φαίνεται και στα επόμενα υποκεφάλαια πιο συγκεκριμένα. Επίσης για τον κώδικα του αλγορίθμου του Καμβά δημιουργήθηκαν και κάποιοι αρχεία αποσφαλμάτωσης τα οποία θα παρουσιάσουμε και πιο κάτω.

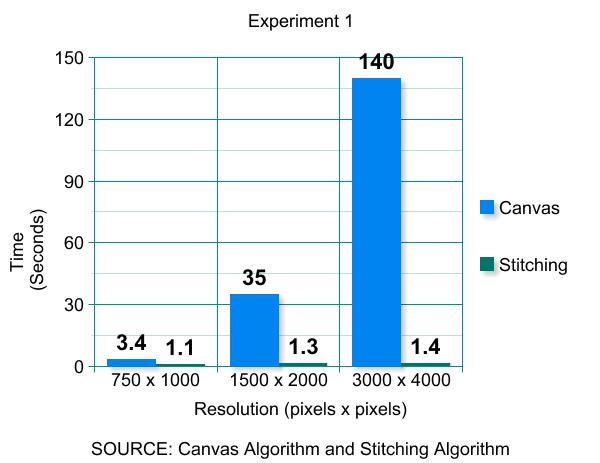
**4.2 Εκτέλεση Αλγορίθμων για Σύγκριση Χρόνου Εκτέλεσης.**

Το πρώτο πείραμα που έγινε για την σύγκριση των δύο αλγορίθμων, αφορούσε τον χρόνο εκτέλεσης των δυο αλγορίθμων. Για το πείραμα αυτό έγινε εκτέλεση και των δύο αλγορίθμων στις Εικόνες (Εικ 4.1 και 4.2) που φαίνονται πιο κάτω.

(Εικ. 4.1 και 4.2)

Οι εικόνες είναι διαστάσεων 3000 x 4000. Λόγο τού μεγέθους του αλγορίθμου του Καμβά όσο αφορά την δομή του ο χρόνος εκτέλεσης του αλγορίθμου είναι μεγαλύτερος του αλγορίθμου του Stitching. Το αποτέλεσμα του πειράματος είναι αντίστοιχα, 140 δευτερόλεπτα και 1.4 δευτερόλεπτα για την εκτέλεση των αλγορίθμων. Για εικόνες 1500 x 2000, οι χρόνοι εκτέλεσης είναι αντίστοιχα, 35 δευτερόλεπτα και 1.3 δευτερόλεπτα. Για εικόνες 750 x 1000, οι χρόνοι εκτέλεσης είναι αντίστοιχα, 3.3 δευτερόλεπτα και 1.2 δευτερόλεπτα. Σύμφωνα με την γραφική παράσταση Γρ.Παρ.1 που ακολουθεί βλέπουμε την διαφορά στους χρόνους εκτέλεσης των δύο αλγορίθμων. Επίσης φαίνονται οι χάρτες που παράγονται από τους αλγορίθμους Καμβά και του Stitching στα αποτελέσματα Απ. 1 και Απ. 2 αντίσοιχα .



(Γρ.Παρ.1)



(Απ. 1)



(Απ. 2)

**4.3 Εκτέλεση Αλγορίθμων με Εικόνες Διαφορετικών Μεγεθών**

Το δεύτερο πείραμα που έγινε για την σύγκριση των δύο αλγορίθμων, αφορούσε την είσοδο των δύο αλγορίθμων. Για το πείραμα αυτό έγινε εκτέλεση και των δύο αλγορίθμων στις Εικόνες (Εικ 4.1 και 4.2) που χρησιμοποιήθηκαν και στο πείραμα 1 που φαίνονται πιο πάνω για την εκτέλεση για εισόδους διαφορετικού μεγέθους. Τα αποτελέσματα είναι Απ. 3 για τον αλγόριθμο του Stitching.



(Aπ .3)

Ο αλγόριθμος του Καμβά δεν μπορεί να ανταποκριθεί σε εικόνες διαφορετικού μεγέθους ενώ ο Αλγόριθμος του Stitching μπορεί να ανταποκριθεί για τον λόγο ότι o αλγόριθμος εξαγωγής των keypoint και της αντιστοίχισης τους, μπορεί να αντιστοιχίσει περιγραφές keypoint διαφορετικών μεγεθών, εφόσον ο πίνακας ομογραφίας δημιουργείτε κανονικά αλλά με λιγότερα. Οι εικόνες είναι διαστάσεων 3000x4000 και 1500x2000 αντίστοιχα.

**4.4 Εκτέλεση Αλγορίθμων με Διαφορετική Ποσότητα Εικόνων**

Το τρίτο πείραμα που έγινε για την σύγκριση των δύο αλγορίθμων, αφορούσε τον χρόνο εκτέλεσης των δυο αλγορίθμων. Για το πείραμα αυτό έγινε εκτέλεση και των δύο αλγορίθμων στις Εικόνες (Εικ 4.3, 4.4 και 4.5) που φαίνονται πιο κάτω.

(Εικ. 4.3,4.4 και 4.5)

Οι εικόνες είναι διαστάσεων 3000 x 4000. Λόγο τού μεγέθους του αλγορίθμου του Καμβά όσο αφορά την δομή του ο χρόνος εκτέλεσης του αλγορίθμου είναι μεγαλύτερος του αλγορίθμου του Stitching. Το αποτέλεσμα του πειράματος είναι αντίστοιχα, 7 δευτερόλεπτα και 360 δευτερόλεπτα για την εκτέλεση των αλγορίθμων. Για εικόνες 1500 x 2000, οι χρόνοι εκτέλεσης είναι αντίστοιχα, 48 δευτερόλεπτα και 375 δευτερόλεπτα. . Για εικόνες 750 x 1000, οι χρόνοι εκτέλεσης είναι αντίστοιχα, 230 δευτερόλεπτα και 360 δευτερόλεπτα. Σύμφωνα με την γραφική παράσταση Γρ.Παρ.2 που ακολουθεί βλέπουμε την διαφορά στους χρόνους εκτέλεσης των δύο αλγορίθμων. Επίσης φαίνονται οι χάρτες που παράγονται από τους αλγορίθμους Καμβά και του Stitching στα αποτελέσματα Απ. 5 και Απ. 6 αντίσοιχα .



(Γρ.Παρ.2)



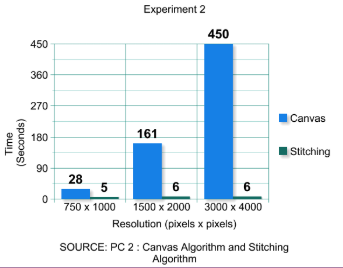
(Απ . 6)

**4.5 Εκτέλεση Αλγορίθμων σε Διαφορετικές Συσκευές**

Το τέταρτο πείραμα που έγινε για την σύγκριση των δύο αλγορίθμων, αφορούσε τον χρόνο εκτέλεσης των δυο αλγορίθμων. Για το πείραμα αυτό έγινε εκτέλεση και των δύο αλγορίθμων στις Εικόνες (Εικ 4.1 και 4.2) που φαίνονται πιο πάνω, σε συσκευές με τα ακόλουθα χαρακτηριστικά .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Η/Υ | CPU | RAM | GPU | O.S. |
| 1 | Intel Core I7-6700HQ CPU @ 2.6 GHz | 4.00 GB  DDR3 | NVIDIA GeForce GTX 950M | Windows 10 Pro |
| 2 | Intel Core I7-6700K | 16.00 GB DDR4 | NVIDIA GeForce GTX 1060 | Windows 10 Pro |

Οι εικόνες είναι διαστάσεων 3000 x 4000. Λόγο τού μεγέθους του αλγορίθμου του Καμβά όσο αφορά την δομή του ο χρόνος εκτέλεσης του αλγορίθμου είναι μεγαλύτερος του αλγορίθμου του Stitching. Το αποτέλεσμα του πειράματος είναι αντίστοιχα, 28 δευτερόλεπτα και 5 δευτερόλεπτα για την εκτέλεση των αλγορίθμων. Για εικόνες 1500 x 2000, οι χρόνοι εκτέλεσης είναι αντίστοιχα, 161 δευτερόλεπτα και 6 δευτερόλεπτα. Για εικόνες 750 x 1000, οι χρόνοι εκτέλεσης είναι αντίστοιχα, 450 δευτερόλεπτα και 6 δευτερόλεπτα. Σύμφωνα με την γραφική παράσταση Γρ.Παρ.3 που ακολουθεί βλέπουμε την διαφορά στους χρόνους εκτέλεσης των δύο αλγορίθμων.

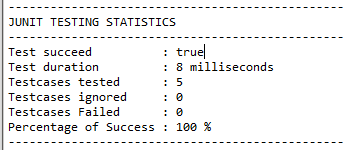


(Γρ.Παρ.3)

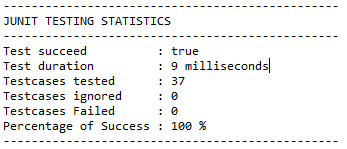
**4.6 Έλεγχος Μονάδων (JUNIT)**

Για την αποσφαλμάτωση του αλγορίθμου του Καμβά, χρησιμοποιήθηκε η βιβλιοθήκη JUNIT σε όλες τις μεθόδους που δημιουργήθηκαν για την εκτέλεση του αλγορίθμου. Κατά την εκτέλεση των JUNIT αρχείων πάρθηκαν τα πιο κάτω αποτελέσματα Εικ 4.6.

Εφαρμογή Drone Foot Print



Εφαρμογή για δημιουργεία χάρτη



**Κεφάλαιο 5**

**Συμπεράσματα**

5.1 Σύγκριση Αποτελεσμάτων των Πειραμάτων 54

5.2 Δυσκολίες που Αντιμετωπίστηκαν 55

5.3 Μελλοντική Ερευνητική Εργασία 55

**5.1 Σύγκριση Αποτελεσμάτων των Πειραμάτων**

Σύμφωνα με τα αποτελέσματα των πειραμάτων που διεξάχθηκαν και μελετήθηκαν στο κεφάλαιο 4, έχουμε βγάλει τα ακόλουθα συμπεράσματα. Σύμφωνα με τα αποτελέσατα των πειραμάτων 1 και 4 και όπως προβλήθηκε, ο αλγόριθμος του Καμβά λόγο μεγαλύτερης πολυπλοκότητας από αυτόν του Stitching, χρειάζεται περισσότερο χρόνο εκτέλεσης. Ο περισσότερος χρόνος που καταναλόνεται είναι κατά το διάβασμα των αρχείων, εφόσον δεν γίνεται χρήση κάποιας βιβλιοθήκης για την ταχύτερη ανάγνωση του. Επίσης η Python διατρέχει ταχύτερη πλατφόρμα, από την πλατφόρμα της Java. Στον αγλόριθμο του Stitching γίνεται χρήση βιβλιοθηκών για την ανάγνωση των εικόνων.

Σύμφωνα με το δεύτερο πείραμα αντιληφθήκαμε ότι ο αλγόριθμος του Καμβά περιορίζεται όσο αφορά την διαφορετικότητα μεταξύ των μεγεθ΄βν εικόνων που πρέπει να επεξεργαστούν ταυτόχρονα λόγο διαφοράς στο padding των εικόνων.

Τέλος σύμφωνα με το τρίτο πείραμα ο αλγόριθμος του Καμβά είναι ταχύτερος σε σύκριση με αυτό του αλγορίθμου του Stitching , όσο αφορά την πολλάπλή είσοδο εικόνων για επεξεργασία. Αυτό όμως εξαρτάτε από ον αριθμό της εισόδου και το μέγεθος των εικόνων.

**5.2 Δυσκολίες που Αντιμετωπίστηκαν**

Οι κύριες δυσκολίες που αντιμετωπίστηκαν, ήταν με τον αλγόριθμο του Καμβά. Η πιο κύρια δυσκολία ήταν η δυσκολία με το πώς να καταφέρουμε να χειριστούμε το padding. Το padding ήταν μια σημαντική παράμετρος, που στην αρχή της υλοποίησης του αλγορίθμου δεν λήφθηκε υπόψη και δημιούργησε προβλήματα κατά την τύπωση των εικόνων στον καμβά και αργότερα την εγγραφή του αρχείου, δημιουργώντας μια εικόνα μη συμβατή στο προβλεπόμενο αποτέλεσμα. Η δεύτερη δυσκολία όσο αφορά τον αλγόριθμο του καμβά ήταν η μαζική ανάγνωση των εικόνων λόγο της πολυπλοκότητας του αναγνώστη της εικόνας. Τα δύο αυτά προβλήματα λήφθηκαν και κατέστησαν τον αλγόριθμο επιτυχές στην τύπωση του χάρτη και την ανάγνωση των εικόνων. Άλλο πρόβλημα που αντιμετωπίσαμε ήταν η διαφορετικότητα μεταξύ των αρχείων log που συνόδευαν τις εικόνες των αποστολών. Για την λύση αυτής της δυσκολίας δημιουργήσαμε το DroneData αρχείο για να υπάρχει συμβατικότητα, καθώς και την βοηθητική εφαρμογή τύπου μπορείς να το δημιουργήσεις και αν το επεξεργαστείς. Η τελευταία δυσκολία που προέκυψε ήταν η ανάκτηση των πληροφοριών από την εικόνα, το οποίο λύθηκε με την χρήση της βιβλιοθήκης JavaXT και την χρήση του exif tag.

**5.3 Μελλοντική Ερευνητική Εργασία**

Αυτή η Ερευνιτική Εργασία θα μπορούσε να σταθεί σαν κίνιτρο για την δημιουργία πιο ολοκληρωμένων εφαρμογών όσο αφορά την δημιουργία χαρτών από εικόνες Drone. Στο κώδικα που υλοποιήθηκε υπάρχει πρόνια για δημιουργία μιας ολοκληρωμένης εφαρμογής χρησιμοποιόντας σαν αρχεία ιστορικού τις αναφορές του αντικειμένου ΒlackΒox. Έτσι θα μπορεί να επαναληφθη η διαδικασία εκτελεσης για πολύ μεγαλύτερο αριθμό εικόνων χωρίς την επαναεκτέλεση του αλγορίθμου από την αρχή. Επίσης θα μπορούσε να δημιουργιθεί κάποιο δύκτιο το οποίο θα συνδεεί το Drone μαζί με κάποιο εξυπηρετητή και να γίνεται κατέβασμα της φωτογραφίας από το Drone κατά την λήψη της. Καθιστώντας έτσι την χρήση των αλγορίθμων ταχύτερη. Τέλος θα μπορούσε να γίνει εύρεση πιο ειδικών βιβλιοθηκών και υπολογισμών αποστάσεων έτσι ώστε να μειώσουν την πολυπλοκότητα, να αυξήσουν την ταχύτητα και να μεγιστοποιήσουν την ακρίβεια του Αλγορίθμου του Καμβά.

**Βιβλιογραφία**

[1] George Heracleous “Ευφυής αυτονόμηση λειτουργειών μη επανδρωμένων αεροσκαφών για αποστολές εκτάκτου ανάγκης.” (2017).

[2] Prashanth, H. S., H. L. Shashidhara, and Balasubramanya Murthy KN. "Image scaling comparison using universal image quality index." Advances in Computing, Control, & Telecommunication Technologies, 2009. ACT'09. International Conference on. IEEE, 2009.

[3] Haines, Richard F., and Sherry L. Chuang. "The effects of video compression on acceptability of images for monitoring life sciences experiments." (1992).

[4] Pennebaker, William B., and Joan L. Mitchell. JPEG: Still image data compression standard. Springer Science & Business Media, 1992.

[5] Khan, Muhammad Imran, Varun Jeoti, and Muhammad Asif Khan. "Perceptual encryption of JPEG compressed images using DCT coefficients and splitting of DC coefficients into bitplanes." Intelligent and Advanced Systems (ICIAS), 2010 International Conference on. IEEE, 2010.

[6] Ahmed, Nasir, T\_ Natarajan, and Kamisetty R. Rao. "Discrete cosine transform." IEEE transactions on Computers 100.1 (1974): 90-93.

[7] Robinson, Jonathan, and Vojislav Kecman. "Combining support vector machine learning with the discrete cosine transform in image compression." IEEE Transactions on Neural Networks 14.4 (2003): 950-958.

[8] Watson, Andrew B., Quingmin J. Hu, and John F. McGowan. "Digital video quality metric based on human vision." Journal of Electronic imaging 10.1 (2001): 20-30.

[9] Chrysafis, Christos, and Antonio Ortega. "Efficient context-based entropy coding for lossy wavelet image compression." Data Compression Conference, 1997. DCC'97. Proceedings. IEEE, 1997.

[10] Aggarwal, Manoj, and Ajai Narayan. "Efficient huffman decoding." Image Processing, 2000. Proceedings. 2000 International Conference on. Vol. 1. IEEE, 2000.

[11] Bourke, Paul. "BMP image format." BMP Files. July (1998).

[12] Miano, John. Compressed image file formats: Jpeg, png, gif, xbm, bmp. Addison-Wesley Professional, 1999.

[13] Bradski, Gary, and Adrian Kaehler. Learning OpenCV: Computer vision with the OpenCV library. " O'Reilly Media, Inc.", 2008.

[14] Kaehler, Adrian, and Gary Bradski. Learning OpenCV 3: computer vision in C++ with the OpenCV library. " O'Reilly Media, Inc.", 2016.

[15] Szeliski, Richard. "Image alignment and stitching: A tutorial." Foundations and Trends® in Computer Graphics and Vision2.1 (2006): 1-104.

[16] Brown, Matthew, and David G. Lowe. "Automatic panoramic image stitching using invariant features." International journal of computer vision 74.1 (2007): 59-73.

[17] Rublee, Ethan, et al. "ORB: An efficient alternative to SIFT or SURF." Computer Vision (ICCV), 2011 IEEE international conference on. IEEE, 2011.

[18] Juan, Luo, and Gwun Oubong. "SURF applied in panorama image stitching." Image Processing Theory Tools and Applications (IPTA), 2010 2nd International Conference on. IEEE, 2010.

[20] Agarwal, Anubhav, C. V. Jawahar, and P. J. Narayanan. "A survey of planar homography estimation techniques." Centre for Visual Information Technology, Tech. Rep. IIIT/TR/2005/12(2005).

[21] Dubrofsky, Elan. "Homography estimation." Diplomová práce. Vancouver: Univerzita Britské Kolumbie (2009).

[22] Jain, Paresh Kumar. "Homography estimation from planar contours." 3D Data Processing, Visualization, and Transmission, Third International Symposium on. IEEE, 2006.

[23] Derpanis, Konstantinos G. "Overview of the RANSAC Algorithm." Image Rochester NY 4.1 (2010): 2-3.

[24] Zhang, Fan, and Feng Liu. "Parallax-tolerant image stitching." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.

[25]Lowe, David G. "Object recognition from local scale-invariant features." Computer vision, 1999. The proceedings of the seventh IEEE international conference on. Vol. 2. Ieee, 1999.

[26] Heymann, S., et al. "SIFT implementation and optimization for general-purpose GPU." (2007).

[27] Dennison, John Charles, Clayton Houston Davis, and Brian Edward Frazier. "Determining angular rate for line-of-sight to a moving object, with a body-fixed imaging sensor." U.S. Patent No. 8,946,606. 3 Feb. 2015.

[28] Danelljan, Martin, et al. "A low-level active vision framework for collaborative unmanned aircraft systems." Workshop at the European Conference on Computer Vision. Springer, Cham, 2014.

[29] Robusto, C. Carl. "The cosine-haversine formula." The American Mathematical Monthly 64.1 (1957): 38-40.

[30] Veness, Chris. "Calculate distance and bearing between two Latitude/Longitude points using Haversine formula in JavaScript." Movable Type Scripts (2011).

[31] IntelliJ, I. D. E. A. "the most intelligent Java IDE." JetBrains[online].[cit. 2016-02-23]. Dostupné z: https://www. jetbrains. com/idea (2011).

[32] Gries, Paul, Jennifer Campbell, and Jason Montojo. Practical Programming: An Introduction to Computer Science Using Python 3.6. Pragmatic Bookshelf, 2017.

[33] Massol, Vincent, and Ted Husted. Junit in action. Manning Publications Co., 2003.

[34] Eckstein, Robert, Marc Loy, and Dave Wood. Java swing. O'Reilly & Associates, Inc., 1998.

[35] Blaha, Michael, and James Rumbaugh. Object-oriented modeling and design with UML. Upper Saddle River: Pearson Education, 2005.

[36] Kramer, Douglas. "API documentation from source code comments: a case study of Javadoc." Proceedings of the 17th annual international conference on Computer documentation. ACM, 1999.

[37] Camera & Imaging Products Association. "Exchangeable image file format for digital still cameras: Exif Version 2.3." CIPA DC-008 Translation-2010 (2010). (exif)

[38] Hong, Sungjin, and Kyung-Hyune Rhee. "An approach for the similar file detection with GPS information." Computers, Networks, Systems and Industrial Engineering (CNSI), 2011 First ACIS/JNU International Conference on. IEEE, 2011.

[39] Camera & Imaging Products Association. "Exchangeable image file format for digital still cameras: Exif Version 2.3." CIPA DC-008 Translation-2010 (2010).

[40] Bovik, Alan C., ed. *The essential guide to image processing*. Academic Press, 2009.

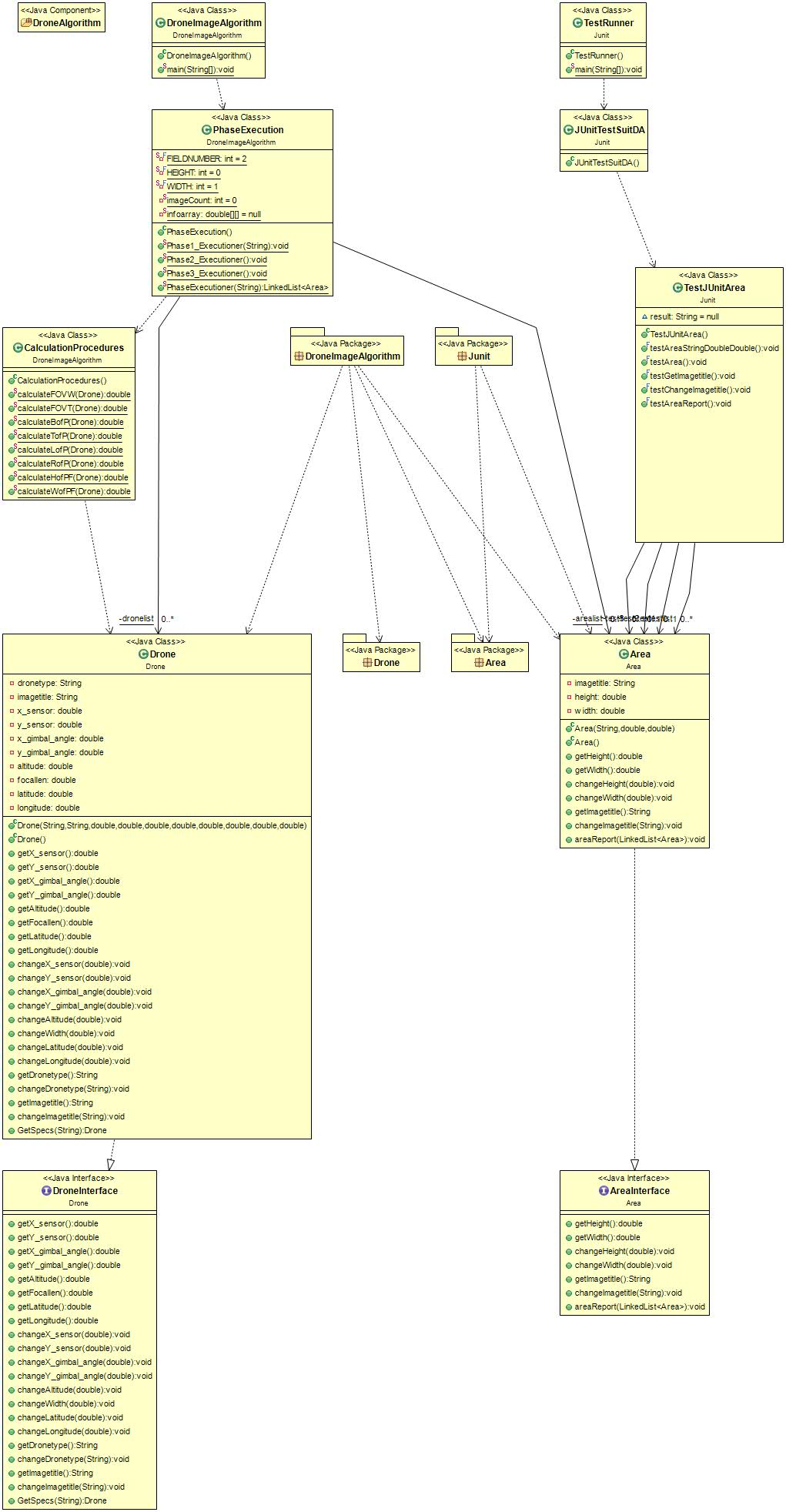
**Παράρτημα Α**

Α.1 Διάγραμμα Κλάσης Drone Algorithm 60

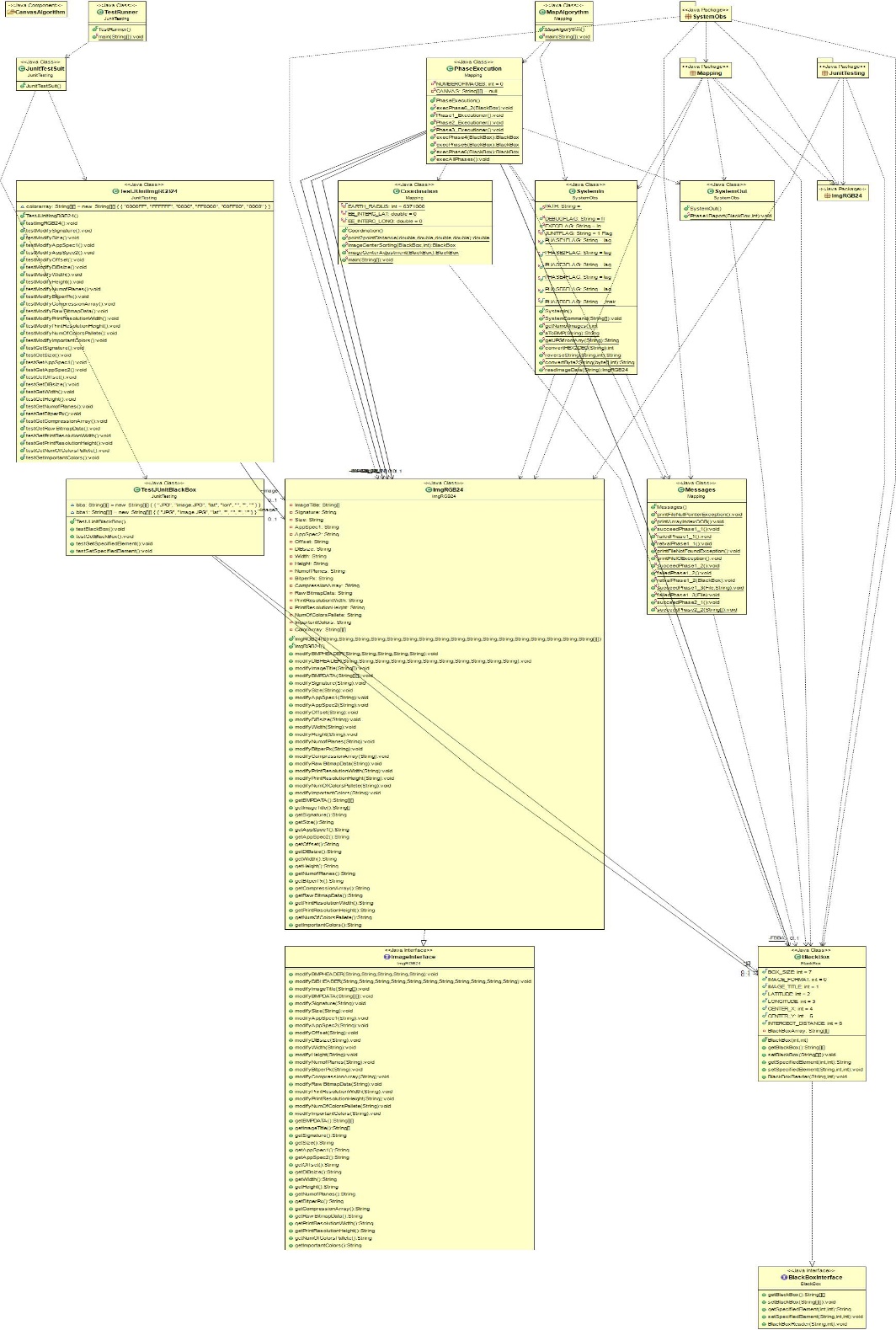
A.2 Διάγραμμα Κλάσης Canvas Algorithm 61

A.3 Διάγραμμα Κλάσης Main App 62

**Α.1 Διάγραμμα Κλάσης Drone Algorithm**



**A.2 Διάγραμμα Κλάσης Canvas Algorithm**



**A.3 Διάγραμμα Κλάσης Main App**



**Παράρτημα Β**

B.1 Διαπροσωπία αντικειμένου Drone 63

B.2 Διαπροσωπία αντικειμένου Area 67

B.3 Διαπροσωπία αντικειμένου BlackBox 69

B.4 Διαπροσωπία αντικειμένου ImgRGB24 70

Το παράρτημα Β περιλαμβάνει κομμάτια από τον πηγαίο κώδικα των κλάσεων των αλγορίθμων για περεταίρω μελέτη και κατανόηση τους

**B.1 Διαπροσωπία αντικειμένου Drone**

**package** Drone;

/\*\*

\* Drone Object Interface

\*

\* **@author** NXATZ

\*

\*/

**public** **interface** DroneInterface {

/\*\*

\* Get x\_sensor.

\* <p>

\* Object Method that returns x\_sensor

\*

\* **@return** x\_sensor -

\*/

**public** **double** getX\_sensor();

/\*\*

\* Get y\_sensor.

\* <p>

\* Object Method that returns y\_sensor

\*

\* **@return** y\_sensor

\*/

**public** **double** getY\_sensor();

/\*\*

\* Get x\_gimbal\_angle.

\* <p>

\* Object Method that returns x\_gimbal\_angle

\*

\* **@return** x\_gimbal\_angle

\*/

**public** **double** getX\_gimbal\_angle();

/\*\*

\* Get y\_gimbal\_angle.

\* <p>

\* Object Method that returns the y\_gimbal\_angle

\*

\* **@return** y\_gimbal\_angle

\*/

**public** **double** getY\_gimbal\_angle();

/\*\*

\* Get altitude.

\* <p>

\* Object Method that returns the altitude

\*

\* **@return** altitude

\*/

**public** **double** getAltitude();

/\*\*

\* Get focallen.

\* <p>

\* Object Method that returns the focallen

\*

\* **@return** focallen

\*/

**public** **double** getFocallen();

/\*\*

\* Get latitude.

\* <p>

\* Object Method that returns the latitude

\*

\* **@return** latitude

\*/

**public** **double** getLatitude();

/\*\*

\* Get longitude.

\* <p>

\* Object Method that returns the longitude

\*

\* **@return** longitude - longitude

\*/

**public** **double** getLongitude();

/\*\*

\* Change x\_sensor

\* <p>

\* Object Method that returns the changed x\_sensor

\*

\* **@param** x\_sensor

\* - x sensor

\*/

**public** **void** setX\_sensor(**double** x\_sensor);

/\*\*

\* Change y\_sensor

\* <p>

\* Object Method that returns the changed y\_sensor

\*

\* **@param** y\_sensor

\* - y\_sensor

\*/

**public** **void** setY\_sensor(**double** y\_sensor);

/\*\*

\* Change x\_gimbal\_angle

\* <p>

\* Object Method that returns the changed x\_gimbal\_angle

\*

\* **@param** x\_gimbal\_angle

\* - x\_gimbal\_angle

\*/

**public** **void** setX\_gimbal\_angle(**double** x\_gimbal\_angle);

/\*\*

\* Change y\_gimbal\_angle

\* <p>

\* Object Method that returns the changed y\_gimbal\_angle

\*

\* **@param** y\_gimbal\_angle

\* - y\_gimbal\_angle

\*/

**public** **void** setY\_gimbal\_angle(**double** y\_gimbal\_angle);

/\*\*

\* Change altitude.

\* <p>

\* Object Method that returns the changed altitude

\*

\* **@param** altitude

\* - altitude

\*/

**public** **void** setAltitude(**double** altitude);

/\*\*

\* Change focallen

\* <p>

\* Object Method that returns the changed focallen

\*

\* **@param** focallen

\* - focallen

\*/

**public** **void** setWidth(**double** focallen);

/\*\*

\* Change image latitude

\* <p>

\* Object Method that returns the changed latitude

\*

\* **@param** latitude

\* - latitude

\*/

**public** **void** setLatitude(**double** latitude);

/\*\*

\* Change longitude

\* <p>

\* Object Method that returns the changed longitude

\*

\* **@param** longitude

\* -longitude

\*/

**public** **void** setLongitude(**double** longitude);

/\*\*

\* Get Drone type

\*

\* **@return** dronetype drone type

\*/

**public** String getDronetype();

/\*\*

\* Change drone type

\*

\* **@param** dronetype

\* drone type to change

\*/

**public** **void** setDronetype(String dronetype);

/\*\*Get drone image title

\*

\* **@return** image

\*/

**public** String getImagetitle();

/\*\*Change drone image title

\*

\* **@param** imagetitle image title change

\*/

**public** **void** setImagetitle(String imagetitle);

/\*\*read and get drone specs

\*

\* **@param** specs Drone specs

\*

\* **@return** Drone object

\*/

**public** Drone GetSpecs(String specs)

**B.2 Διαπροσωπία αντικειμένου Area**

package Area;

import java.io.IOException;

import java.util.LinkedList;

/\*\*

\* Area Object Interface

\*

\* @author NXATZ

\*

\*/

public interface AreaInterface {

/\*\*

\* Get Image height.

\* <p>

\* Object Method that returns the real image height

\*

\* @return height - The real height in meters

\*/

public double getHeight();

/\*\*

\* Get Image width.

\* <p>

\* Object Method that returns the real image width

\*

\* @return width - The real width in meters

\*/

public double getWidth();

/\*\*

\* Change image height.

\* <p>

\* Object Method that returns the changed height

\*

\* @param height

\* - The change

\*/

**public void changeHeight(double height);**

/\*\*

\* Change image width.

\* <p>

\* Object Method that returns the changed width

\*

\* @param width

\* - The change

\*/

public void changeWidth(double width);

/\*\*

\* Get Image title.

\* <p>

\* Object Method that returns the real image title

\*

\* @return width - The real title

\*/

public String getImagetitle();

/\*\*

\* Change image title.

\* <p>

\* Object Method that returns the changed title

\*

\* @param imagetitle

\* image title

\*/

public void changeImagetitle(String imagetitle);

/\*\*Print in a file the report

\*

\* @param arealist area list

\* @param path the path to save

\* @throws IOException IO exception

\*/

public void areaReport(LinkedList<Area> arealist,String path) throws IOException;

}

**B.3 Διαπροσωπία αντικειμένου BlackBox**

**package** BlackBox;

/\*\*

\* **@author** NXATZ

\*

\*/

**public** **interface** BlackBoxInterface {

/\*\*

\* Get the new black box array

\*

\* **@return** the blackBox

\*/

String[][] getBlackBox();

/\*\*

\* Set update a new black box array

\*

\* **@param** blackBox

\* the blackBox to set

\*/

**public** **void** setBlackBox(String[][] blackBox);

/\*\*

\* Retrieve a specified position value

\*

\* **@param** Row

\* the row

\* **@param** Col

\* the column

\*

\* **@return** the value of the position

\*/

**public** String getSpecifiedElement(**int** Row, **int** Col);

/\*\*

\* Modified a specified position value

\*

\* **@param** element

\* the element to change

\* **@param** Row

\* the row

\* **@param** Col

\* the column

\*/

**public** **void** setSpecifiedElement(String element, **int** Row, **int** Col);

/\*\*

\* Retrieve the table with flight's data BMP title,latitude and longitude

\* coordinates

\*

\* **@param** path

\* path to flight data log file

\* **@param** num

\* number of images taken from drone

\*/

**public** **void** BlackBoxReader(String path, **int** num);

}

**B.4 Διαπροσωπία αντικειμένου ImgRGB24**

**package** ImgRGB24;

/\*\*

\* **@author** NXATZ

\*

\*/

**public** **interface** ImageInterface {

/\*\*

\* BMP header setter

\*

\* **@param** Signature

\* File format signature

\* **@param** Size

\* Size of file

\* **@param** AppSpec1

\* Always set to 00 00

\* **@param** AppSpec2

\* Always set to 00 00

\* **@param** Offset

\* Offset size to the pixel data

\*/

**public** **void** modifyBMPHEADER(String Signature, String Size, String

AppSpec1, String AppSpec2, String Offset);

/\*\*

\* DIB header setter

\*

\* **@param** DIBsize

\* Size of Dibheader header

\* **@param** Width

\* Image width

\* **@param** Height

\* Image Height

\* **@param** NumofPlanes

\* Number of planes on image

\* **@param** BitperPx

\* Size per pixel : 24 bit

\* **@param** CompressionArray

\* Compression array standards

\* **@param** RawBitmapData

\* Image size without header

\* **@param** PrintResolutionWidth

\* Screen resolution width

\* **@param** PrintResolutionHeight

\* Screen resolution height

\* **@param** NumOfColorsPallete

\* Number of colors in the pallete

\* **@param** ImportantColors

\* Important colors

\*/

**public** **void** modifyDIBHEADER(String DIBsize, String Width, String Height,

String NumofPlanes, String BitperPx,

String CompressionArray,String RawBitmapData, String PrintResolutionWidth,

String PrintResolutionHeight,

String NumOfColorsPallete,

String ImportantColors);

/\*\*

\* ImageTitle setter

\*

\* **@param** flightBlackBox

\* Image filename

\*

\*/

**public** **void** modifyImageTitle(String[] flightBlackBox);

/\*\*

\* BMP Pixel array setter

\*

\* **@param** ColorArray

\* Image pixel colors

\*/

**public** **void** modifyBMPDATA(String[][] ColorArray);

/\*\*

\* Signature setter

\*

\* **@param** Signature

\* File format signature

\*/

**public** **void** modifySignature(String Signature);

/\*\*

\* Size setter

\*

\* **@param** Size

\* Size of file

\*/

**public** **void** modifySize(String Size);

/\*\*

\* AppSpec1 setter

\*

\* **@param** AppSpec1

\* Always set to 00 00

\*/

**public** **void** modifyAppSpec1(String AppSpec1);

/\*\*

\* AppSpec2 setter

\*

\* **@param** AppSpec2

\* Always set to 00 00

\*/

**public** **void** modifyAppSpec2(String AppSpec2);

/\*\*

\* Offset setter

\*

\* **@param** Offset

\* Offset size to the pixel data

\*/

**public** **void** modifyOffset(String Offset);

/\*\*

\* DIB size setter

\*

\* **@param** DIBsize

\* Size of Dibheader header

\*/

**public** **void** modifyDIBsize(String DIBsize);

/\*\*

\* Width setter

\*

\* **@param** Width

\* Image width

\*/

**public** **void** modifyWidth(String Width);

/\*\*

\* Height setter

\*

\* **@param** Height

\* Image Height

\*/

**public** **void** modifyHeight(String Height);

/\*\*

\* NumofPlanes setter

\*

\* **@param** NumofPlanes

\* Number of planes on image

\*/

**public** **void** modifyNumofPlanes(String NumofPlanes);

/\*\*

\* BitperPx setter

\*

\* **@param** BitperPx

\* Size per pixel : 24 bit

\*/

**public** **void** modifyBitperPx(String BitperPx);

/\*\*

\* CompressionArray setter

\*

\* **@param** CompressionArray

\* Compression array standards

\*/

**public** **void** modifyCompressionArray(String CompressionArray);

/\*\*

\* RawBitmapData setter

\*

\* **@param** RawBitmapData

\* Image size without header

\*/

**public** **void** modifyRawBitmapData(String RawBitmapData) ;

/\*\*

\* PrintResolutionWidth setter

\*

\* **@param** PrintResolutionWidth

\* Screen resolution width

\*/

**public** **void** modifyPrintResolutionWidth(String PrintResolutionWidth) ;

/\*\*

\* PrintResolutionHeight setter

\*

\* **@param** PrintResolutionHeight

\* Screen resolution height

\*/

**public** **void** modifyPrintResolutionHeight(String PrintResolutionHeight) ;

/\*\*

\* NumOfColorsPallete setter

\*

\* **@param** NumOfColorsPallete

\* Number of colors in the pallete

\*/

**public** **void** modifyNumOfColorsPallete(String NumOfColorsPallete) ;

/\*\*

\* ImportantColors setter

\*

\* **@param** ImportantColors

\* Important colors

\*/

**public** **void** modifyImportantColors(String ImportantColors) ;

/\*\*

\* ColorArray getter

\*

\* **@return** ColorArray

\*/

**public** String[][] getBMPDATA() ;

/\*\*

\* ImageTitle getter

\*

\* **@return** ImageTitle

\*/

**public** String[] getImageTitle() ;

/\*\*

\* Signature getter

\*

\* **@return** Signature

\*/

**public** String getSignature() ;

/\*\*

\* Size getter

\*

\* **@return** Size

\*/

**public** String getSize() ;

/\*\*

\* AppSpec1 getter

\*

\* **@return** AppSpec1

\*/

**public** String getAppSpec1() ;

/\*\*

\* AppSpec2 getter

\*

\* **@return** AppSpec2

\*/

**public** String getAppSpec2() ;

/\*\*

\* Offset getter

\*

\* **@return** Offset

\*/

**public** String getOffset() ;

/\*\*

\* DIBsize getter

\*

\* **@return** DIBsize

\*/

**public** String getDIBsize() ;

/\*\*

\* Width getter

\*

\* **@return** Width

\*/

**public** String getWidth() ;

/\*\*

\* Height getter

\*

\* **@return** Height

\*/

**public** String getHeight() ;

/\*\*

\* NumofPlanes getter

\*

\* **@return** NumofPlanes

\*/

**public** String getNumofPlanes() ;

/\*\*

\* BitperPx getter

\*

\* **@return** BitperPx

\*/

**public** String getBitperPx() ;

/\*\*

\* CompressionArray getter

\*

\* **@return** CompressionArray

\*/

**public** String getCompressionArray() ;

/\*\*

\* RawBitmapData getter

\*

\* **@return** RawBitmapData

\*/

**public** String getRawBitmapData() ;

/\*\*

\* PrintResolutionWidth getter

\*

\* **@return** PrintResolutionWidth

\*/

**public** String getPrintResolutionWidth() ;

/\*\*

\* PrintResolutionHeight getter

\*

\* **@return** PrintResolutionHeight

\*/

**public** String getPrintResolutionHeight() ;

/\*\*

\* NumOfColorsPallete getter

\*

\* **@return** NumOfColorsPallete

\*/

**public** String getNumOfColorsPallete() ;

/\*\*

\* ImportantColors getter

\*

\* **@return** ImportantColors

\*/

**public** String getImportantColors() ;

}