



Τεχνολογικό  
Πανεπιστήμιο  
Κύπρου

Σχολή Μηχανικής και  
Τεχνολογίας

Μεταπτυχιακή διατριβή

**Underwater Cyclops – Real Time Underwater Image  
Mapping based on Homography**

Στέλιος Νεοφυτίδης

Λεμεσός, Ιανουάριος 2021



ΤΕΧΝΟΛΟΓΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΥΠΡΟΥ  
ΣΧΟΛΗ ΜΗΧΑΝΙΚΗΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ  
ΤΜΗΜΑ ΠΟΛΙΤΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ  
ΓΕΩΠΛΗΡΟΦΟΡΙΚΗΣ

Μεταπτυχιακή διατριβή

Underwater Cyclops – Real Time Underwater Image Mapping based on  
Homography

του

Στέλιου Νεοφυτίδη

Λεμεσός, Ιανουάριος 2021

# Έντυπο έγκρισης

Μεταπτυχιακή διατριβή

## **Underwater Cyclops – Real Time Underwater Image Mapping based on Homography**

Παρουσιάστηκε από

Στέλιο Νεοφυτίδη

Επιβλέπων καθηγητής: Δρ. Σκαρλάτος Δημήτριος, Αναπληρωτής Καθηγητής

Υπογραφή \_\_\_\_\_

Μέλος επιτροπής:

Υπογραφή \_\_\_\_\_

Μέλος επιτροπής:

Υπογραφή \_\_\_\_\_

Τεχνολογικό Πανεπιστήμιο Κύπρου

Λεμεσός, Ιανουάριος 2021

## **Πνευματικά δικαιώματα**

Copyright © Στέλιος Νεοφυτίδης, 2021

Με επιφύλαξη παντός δικαιώματος. All rights reserved.

Η έγκριση της μεταπτυχιακής διατριβής από το Τμήμα Πολιτικών Μηχανικών και Μηχανικών Γεωπληροφορικής του Τεχνολογικού Πανεπιστημίου Κύπρου δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων του συγγραφέα εκ μέρους του Τμήματος.

Αυτή η διατριβή σημάνει το τέλος του Μεταπτυχιακού μου και την αρχή ενός νέου μονοπατιού, πολύ πιο δύσκολου, και ξέρω πως όπως και τώρα θα έχω δίπλα μου τους γονείς μου, την σύντροφο μου, την οικογένεια μου ολόκληρη και τους κοντινούς μου ανθρώπους να με στηρίζουν.

Θέλω λοιπόν, αυτούς τους ανθρώπους να τους ευχαριστήσω, για όλη τη στήριξη που μου παρείχαν κατά τη διάρκεια των σπουδών μου, που ήταν δίπλα μου όταν τα πράγματα ήταν σκούρα σε αυτό το επίπονο ταξίδι, που αποδείχτηκε τελικά, ένα τεράστιο σχολείο.

Με την ολοκλήρωση λοιπόν αυτής της διατριβής, θα έχω και επίσημα τον Μεταπτυχιακό Τίτλο «Γεωπληροφορικής και Γεωχωρικές Τεχνολογίες» με πόρτες ορθάνοιχτες να περιμένουν.

## ΠΕΡΙΛΗΨΗ

Μέσα από αυτή τη διατριβή μεταπτυχιακού επιπέδου υπήρχε σκοπός να δημιουργηθεί ένα λογισμικό Τεχνητής Νοημοσύνης (Artificial Intelligence) που θα εκτελείται με τη βοήθεια ενός μικροϋπολογιστή σε ένα υποβρύχιο ρομπότ (uROV) και να βοηθά τον χρήστη με αυτόματη χαρτογράφηση (automated mapping) κατά την πλοήγηση μέσα στο νερό σε πραγματικό χρόνο (Real-Time).

Το λογισμικό της διατριβής που πήρε το όνομα “Underwater Cyclops”, έχει ως στόχο να βοηθήσει τους ερευνητές του υποβρύχιου κόσμου στην πλοήγηση τους μέσα στο νερό με την αυτόματη δημιουργία ενός χάρτη από τα frames (καρέ) που βλέπει η camera του ROV φτιάχνοντας έτσι ένα μωσαϊκό με κύρια μέθοδο την Ομογραφία (Homography) σε πραγματικό χρόνο. Επίσης στόχος ήταν η διερεύνηση του κατά πόσο μπορεί η Ομογραφία να λειτουργήσει ορθά για τη λύση των προβλημάτων όπως για παράδειγμα η υποβρύχια πλοήγηση, ο χαμηλός φωτισμός και η ομοιομορφία στο υποβρύχιο περιβάλλον.

Ομογραφία είναι ο τρόπος που χαρακτηρίζεται η σχέση μεταξύ δύο εικόνων και χρησιμοποιείται σε πολλούς τομείς της Υπολογιστικής Όρασης (Computer Vision) όπως τον υπολογισμό κίνησης της κάμερας (camera motion estimation), αναγωγή συνόλου δεδομένων σε ένα σύστημα συντεταγμένων (image registration) ή ακόμα και για την διόρθωση μιας εικόνας (image rectification).

Μέσα από την διατριβή αυτή συμπεραίνεται ότι η Ομογραφία, χωρίς να συνδυαστεί με κάποια τεχνική, δεν μπορεί να βοηθήσει σε ένα τέτοιο πρόβλημα. Χρειάζεται την υποστήριξη μιας επιπλέον τεχνικής για σωστή εκτίμηση της θέσης του uROV χωρίς να δημιουργεί γεωμετρικές παραμορφώσεις στα αποτελέσματα.

### Λέξεις κλειδιά:

**Τεχνητή Νοημοσύνη:** Είναι ο κλάδος της πληροφορικής που προσπαθεί να μεταφέρει ανθρώπινη συμπεριφορά και ανθρώπινα χαρακτηριστικά σε μία μηχανή.

**Ομογραφία:** Ο τρόπος συσχέτισης δύο εικόνων.

**Υπολογιστική Όραση:** Είναι το πεδίο της Τεχνητής Νοημοσύνης όπου κατά τις έρευνες σε αυτό γίνεται προσπάθεια αλγοριθμικά να μεταφερθεί η αίσθηση της ανθρώπινης όρασης σε ένα υπολογιστή ή σε ένα ρομπότ.

**Μωσαϊκό:** Είναι η απεικόνιση μιας μεγάλης περιοχής χρησιμοποιώντας μικρότερες εικόνες, συνήθως από μία αλληλουχία εικόνων.

**Real-Time System:** Λογισμικό σύστημα το οποίο διεκπεραιώνει την αποστολή του σε πραγματικό χρόνο.



## **ABSTRACT**

The main purpose of this thesis was the development of a real time Artificial Intelligence (AI) software system that is being executed on a microcomputer and installed on an underwater ROV (Remotely Operated Vehicle) to assist the researcher on their work while they are navigating under the water.

The thesis' software being named "Underwater Cyclops" is targeting to assist the researchers in exploring the underwater world by generating an automatic map using ROV's camera's frames to create an image mosaic based on Homography in real-time and without any important latency. One of the thesis's purposes was to determine how Homography can be used under these circumstances and face problems like the underwater navigation, low light and uniform environment.

As Homography is being referred, is the way we describe how two images are related and is being used in a lot of Computer Vision(CV) work like Image Rectification, Image Registration and Camera Motion Estimation.

Through this dissertation it is concluded that Homography, without being combined with any technique, cannot help in such a problem. It needs the support of an additional technique to correctly estimate the position of the uROV without creating geometric distortions in the results.

### **Keywords:**

**Artificial Intelligence:** The Computer Science's field which is working on transferring the human behavior in a machine.

**Homography:** The way of describing two images are related.

**Computer Vision:** The field of Artificial Intelligence which is working on transferring the sense of Vision by using different algorithms on a computer or a robot.

**Image Mosaic:** The representation of a large area by using video's frames.

**Real Time System:** A software system which is completing its task in real time.

## ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

ΠΕΡΙΛΗΨΗ.....	vi
ABSTRACT.....	viii
ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ.....	ix
ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ.....	xi
ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ.....	xii
ΑΠΟΔΟΣΗ ΟΡΩΝ.....	xiv
1 Εισαγωγή.....	1
1.1 Βιβλιογραφική Ανασκόπηση.....	3
1.2 Τεχνικό Υπόβαθρο.....	5
1.2.1 Θεωρητικό Υπόβαθρο.....	6
1.2.1.1 Image Matching Algorithms.....	6
1.2.1.2 Oriented FAST Rotated BRIEF (ORB).....	6
1.2.1.3 Brute Force Matcher (BF Matcher).....	6
1.2.1.4 Detector.....	6
1.2.1.5 Descriptor.....	6
1.2.1.6 Homography.....	6
1.2.1.7 RANSAC.....	7
1.2.1.8 Perspective Transformation.....	7
1.2.2 Τεχνικό Υπόβαθρο.....	7
1.2.2.1 Python.....	7
1.2.2.2 PyCharms IDE.....	7
1.2.2.3 OpenCV.....	7
1.2.2.4 numpy.....	7
1.2.2.5 BlueROV2.....	8

1.2.2.6	Fathom Tether Cable.....	8
1.2.2.7	Raspberry Pi 3 Model B/ Microcomputers.....	8
2	Μεθοδολογία Έρευνας.....	9
3	Αποτελέσματα και Εφαρμογή.....	18
4	Συζήτηση αποτελεσμάτων .....	30
5	Συμπεράσματα και Μελλοντική Εργασία.....	34
	ΕΠΙΛΟΓΟΣ .....	36
	ΒΙΒΛΙΟΓΡΑΦΙΑ .....	38

## ΚΑΤΑΛΟΓΟΣ ΠΙΝΑΚΩΝ

Πίνακας 1: Αριθμητικές πληροφορίες αποτελεσμάτων .....	19
--	----

## ΚΑΤΑΛΟΓΟΣ ΔΙΑΓΡΑΜΜΑΤΩΝ

Διάγραμμα 1: Διάταξη υλικού (hardware) .....	11
Διάγραμμα 2: Διάγραμμα Ροής Διαδικασιών .....	13
Διάγραμμα 3: Τελικό προϊόν Dataset 1 .....	21
Διάγραμμα 4: Τελικό προϊόν Dataset 2 .....	21
Διάγραμμα 5: Τελικό προϊόν Dataset 3 .....	22
Διάγραμμα 6: Τελικό προϊόν Dataset 4 .....	22
Διάγραμμα 7: Σύγκριση χάρτη Dataset 3 με χάρτη Dataset 4 .....	23
Διάγραμμα 8: Τελικό προϊόν Dataset 5 .....	24
Διάγραμμα 9: Τελικό προϊόν Dataset 6 .....	24
Διάγραμμα 10: Τελικό προϊόν Dataset 7 .....	25
Διάγραμμα 11: Τελικό προϊόν Dataset 8 .....	25
Διάγραμμα 12: Τελικό προϊόν Dataset 480p .....	26
Διάγραμμα 13: Κακό τελικό προϊόν 1 .....	27
Διάγραμμα 14: Κακό τελικό προϊόν 2 .....	27
Διάγραμμα 15: Κακή αντιστοιχία σημείων 1 .....	28
Διάγραμμα 16: Κακή αντιστοιχία σημείων 2 .....	28
Διάγραμμα 17: Καλή αντιστοιχία που δίνει γεωμετρικά παραμορφωμένο χάρτη .....	29
Διάγραμμα 18: Καλή αντιστοιχία που δίνει μη γεωμετρικά παραμορφωμένο χάρτη ....	29
Διάγραμμα 19: Αποτέλεσμα από πηγή βιβλιογραφία υπ' αριθμόν 11 .....	32
Διάγραμμα 20: Αποτέλεσμα από πηγή βιβλιογραφία υπ' αριθμόν 15 .....	32
Διάγραμμα 21: Αποτέλεσμα από πηγή βιβλιογραφία υπ' αριθμόν 8 .....	33

## ΣΥΝΤΟΜΟΓΡΑΦΙΕΣ

Παρουσιάζονται συνοπτικά όλες οι σημαντικές συντομογραφίες που έχουν χρησιμοποιηθεί στο κείμενο της διατριβής και χρειάζονται επεξήγηση:

ΤΕΠΑΚ.:	Τεχνολογικό Πανεπιστήμιο Κύπρου
ROV:	Remotely Operated Vehicle
AI:	Artificial Intelligence
CV:	Computer Vision
SLAM:	Simultaneous Localization and Mapping
GNSS:	Global Navigation Satellite System
ANN:	Artificial Neural Network
AUV:	Autonomous Underwater Vehicle
RANSAC:	Random Sample Consensus

## ΑΠΟΔΟΣΗ ΟΡΩΝ

Data	Δεδομένα
Artificial Intelligence	Τεχνητή Νοημοσύνη
Computer Vision	Υπολογιστική Όραση
Python	Γλώσσα Προγραμματισμού
OpenCV	Βιβλιοθήκη εντολών προγραμματισμού Υπολογιστικής Όρασης
uROV	Υποβρύχιο τηλεκατευθυνόμενο όχημα από απόσταση
Frames	Καρέ
Real Time	Πραγματικός Χρόνος
Artificial Neural Network	Τεχνητό Νευρωνικό Δίκτυο
Visual Odometry	Οπτική Οδομετρία
Data Association	Σύγκριση Δεδομένων
Computational Time	Υπολογιστικός Χρόνος
Image Stitching	Συρραφή εικόνων
Homography	Ομογραφία

# 1 Εισαγωγή

Κατά την εισαγωγή θα αναλυθεί το θέμα αυτής της πτυχιακής εργασίας, ο σκοπός που έχει, τα προβλήματα που λύνονται μέσα από όλη την έρευνα που έγινε αλλά και από την ανάπτυξη του Real-Time λογισμικού Underwater Cyclops, θα παρουσιαστεί η βιβλιογραφική ανασκόπηση με προσπάθειες που έγιναν για παρόμοιο σκοπό και θα τελειώσει με μία ανάλυση του τεχνικού υπόβαθρου που χρησιμοποιήθηκε για την δημιουργία του λογισμικού και το οποίο χωρίζεται στο θεωρητικό υπόβαθρο και στο υπόβαθρο υλοποίησης.

Το θέμα της συγκεκριμένης μεταπτυχιακής διατριβής ήταν η ανάπτυξη ενός λογισμικού συστήματος, το οποίο θα λειτουργεί σε πραγματικό χρόνο και σε αλληλεπίδραση μαζί με ένα microcomputer εγκατεστημένο σε ένα ROV να δημιουργεί ένα ορθοφωτογραφικό μωσαϊκό που θα απεικονίζει ένα χάρτη για να τον συμβουλευτεί κάποιος ερευνητής κατά τις υποβρύχιες του ερευνητικές εργασίες.

Γενικός σκοπός της διατριβής αυτής, είναι η χρήση τεχνολογιών προγραμματισμού Τεχνητής Νοημοσύνης και πιο συγκεκριμένα του πεδίου της Υπολογιστικής Όρασης σε συνδυασμό μαζί με γνώσεις Φωτογραμμετρίας για την παραγωγή του λογισμικού συστήματος το οποίο θα μπορεί να χειρίζεται εικόνες που λήφθηκαν σε υποβρύχιες συνθήκες με ότι αυτό συνεπάγεται και να δημιουργείται ένας «χάρτης» που θα βοηθά τους ερευνητές. Επιπλέον σκοπός ήταν η μελέτη των συνθηκών που κάνουν τα πράγματα πιο δύσκολα για έναν ερευνητή κάτω από το νερό, η μελέτη αλγορίθμων που μπορούν να βρουν τα κοινά σημεία σε δύο εικόνες και πως αντιστοιχούνται αλλά και η υλοποίησή τους σε πηγαίο κώδικα. Τα προβλήματα που αντιμετωπίζονται κάτω από την επιφάνεια της θάλασσας από τους ερευνητές κατά τη διάρκεια εργασιών τους, είναι συνθήκες σκότους, συνθήκες ομοιομορφίας του περιβάλλοντος που πολύ συχνά οδηγούν σε μια κατάσταση αποπροσανατολισμού και δυσκολίες πλοήγησης που υπάρχουν. Αξίζει επίσης να αναφερθεί ότι είναι λίγα τα λογισμικά που διεκπεραιώνουν τέτοιου είδους εργασίες.

Επίσης ως σκοπός της συγκεκριμένης ερευνητικής διπλωματικής εργασίας είναι η διερεύνηση του κατά πόσο με τη χρήση της μεθόδου της Ομογραφίας μπορεί να προκύψει ένα τέτοιο λογισμικό.



Η ιδέα για να προχωρήσει η ερευνητική ομάδα στην υλοποίηση του λογισμικού Underwater Cyclops ήρθε διαβάζοντας το ερευνητικό άρθρο[1] μιας ερευνητικής ομάδας ενός πανεπιστημίου στη Μασσαλία της Γαλλίας, όπου επικεντρώθηκαν στην ανακατασκευή ενός φωτογραφικού μωσαϊκού το οποίο αποτελείται από επικαλυπτόμενες φωτογραφίες συνδεδεμένες βάσει σημείων κλειδί.

Σημαντικό μέρος του συστήματος είναι η ευχρηστία του και η ικανότητα του να το χειρίζεται κάποιος εύκολα, νοουμένου ότι υπάρχουν κάποιες απαραίτητες γνώσεις εγκατάστασης, προγραμματισμού και χρήσης συστημάτων τεχνητής νοημοσύνης. Το Underwater Cyclops, κρίνοντας και από τη βιβλιογραφία είναι ένα από τα λίγα λογισμικά που επιχειρεί να χειριστεί καταστάσεις υποβρύχιας πλοήγησης και θα αποτελέσει ένα ισχυρό εργαλείο στα χέρια των ερευνητών του Photogrammetric-Vision Lab του ΤΕΠΑΚ, καθώς πολλές έρευνες του εργαστηρίου αφορούν την υποβρύχια εξερεύνηση, κυρίως για αρχαιολογικούς σκοπούς.

Δεν λύνονται στην ολότητα τους τα προβλήματα ακόμα και νοουμένης της ύπαρξης του λογισμικού Underwater Cyclops. Το λογισμικό θα μπορούσε να λύνει τα προβλήματα που αντιμετωπίζονται αν γινόταν μία ελαφρώς διαφορετική προσέγγιση εξ αρχής αφού δυστυχώς, παρατηρούνται στα τελικά αποτελέσματα κάποιες γεωμετρικές παραμορφώσεις, που οφείλονται στην μέθοδο της Ομογραφίας την οποία εισηγήθηκε ο επιβλέπων καθηγητής. Σημειωτέο όμως ότι η όλη διατριβή είχε πειραματική και ερευνητική υπόσταση, που προνοούσε τη μελέτη της μεθόδου της Ομογραφίας για να λυθούν τέτοια θέματα και ο ερευνητής ήταν εξ αρχής προετοιμασμένος ότι πιθανόν να μην βρεθεί λύση με το πέρας και την ολοκλήρωση της παρούσας διπλωματικής εργασίας. Ωστόσο βγαίνουν σημαντικά συμπεράσματα που θα βοηθήσουν την κοινή ερευνητική γνώμη.

Ακόμη ένα από τα κυριότερα προβλήματα που οδηγούν τους ερευνητές να στρέψουν το βλέμμα τους σε λύσεις SLAM και Visual Odometry σε υποβρύχια καταστάσεις είναι η έλλειψη ενός παγκόσμιου υποβρύχιου συστήματος θέσεως όπως τα επίγεια GNSS συστήματα.

Συνοψίζοντας, πολλά είναι τα προβλήματα για τους ερευνητές που ασχολούνται με έρευνες στον υποβρύχιο κόσμο και δυστυχώς ένα λογισμικό που χρησιμοποιεί αποκλειστικά την μέθοδο της Ομογραφίας δεν μπορεί να δώσει μία χειροπιαστή λύση.

## 1.1 Βιβλιογραφική Ανασκόπηση

Αρκετά είναι τα παραδείγματα στη παγκόσμια ερευνητική βιβλιογραφία με διάφορες προσπάθειες ερευνητικών ομάδων στην ανάγκη τους για ένα εργαλείο που θα τους βοηθά στην υποβρύχια εξερεύνηση και πλοήγηση. Στο υποκεφάλαιο 1.1 Βιβλιογραφική Ανασκόπηση της παρούσας μεταπτυχιακής διατριβής θα αναλυθούν διάφορες προσπάθειες και θα κριθεί η καταλληλότητα τους για την δική μας περίπτωση.

Μια προσπάθεια ερευνητικής ομάδας ήταν η ανάπτυξη ενός Semi-Global Visual Odometry [2] συστήματος, βασισμένο σε stereo σύστημα φωτογραφικών λήψεων, το οποίο και χρησιμοποιήθηκε και σε περιήγηση στο ναυάγιο Xlendi στη Μάλτα, ο λόγος που στη συγκεκριμένη περίπτωση δεν μπορούσε να ακολουθηθεί μία τέτοια προσέγγιση ήταν γιατί το Underwater Cyclops ήταν βασισμένο σε monocular σύστημα φωτογραφικών λήψεων.

Εξαιρετική είναι και η προσπάθεια των Guangchao Hou et al. [3] σε μια πρόσφατη έρευνα του 2019, όπου κατάφεραν να δημιουργήσουν ένα Simultaneous Localization and Mapping (SLAM) αλγόριθμο, βασισμένο σε ένα Artificial Neural Network (ANN) τον ARB-SLAM αλλά αναπτύχθηκε για την χρήση του σε AUV και όχι ROV όπως τυγχάνει στη παρούσα προσπάθεια. Γιατί σε αντίθεση τα AUVs είναι αυτόνομα και είναι εντελώς διαφορετική η φιλοσοφία με την οποία λειτουργούν αλλά και η φιλοσοφία η οποία πρέπει να ακολουθηθεί για να αναπτυχθεί ένα σύστημα το οποίο θα αλληλοεπιδρά με ένα ROV.

Άλλοι ερευνητές σε άλλη προσέγγιση, προσπάθησαν να κατασκευάσουν από την αρχή και τον δικό τους αλγόριθμο [4] αντί της χρήσης έτοιμων αλγορίθμων όπως BRIEF,ORB,FAST και αυτή η προσπάθεια όπως και άλλες είναι βασισμένη και αυτή σε stereo σύστημα. Γενικώς αυτό δημιουργούσε πολλά ερωτήματα ως προς την υλοποίηση σε θέματα πηγαίου κώδικα ενός αλγορίθμου φτιαγμένου από άλλες ομάδες, που σημαίνει κόστος χρόνου. Ακόμη το ότι η προσέγγιση απασχολούσε stereo σύστημα σήμαινε ότι δεν καθίσταται δυνατόν να χρησιμοποιηθεί στην παρούσα έρευνα.

Η ερευνητική ομάδα των Ferrara et al., το 2009, προχώρησε στην υλοποίηση SLAM όπου τα δεδομένα για την δημιουργία του μωσαϊκού συνόδευαν δεδομένα laser triangulation optical correlation, που στη παρούσα περίπτωση δεν ήταν δυνατό να

δοκιμαστεί κάτι τέτοιο, γιατί το εργαστήριο Photogrammetric-Vision Lab δεν ήταν εφοδιασμένο με τέτοιο εξάρτημα για το ROV.

Η ίδια σχεδόν ομάδα με μια πολύ κοντινή προσέγγιση σε αυτή της διατριβής, παρουσίασε μία έρευνα [5] σε συνέδριο στον Πλατανιά των Χανίων το 2013 που αφορούσε μία προσέγγιση που χρησιμοποιεί τους αλγόριθμους SURF και BRIEF για να λύσει τα προβλήματα του data association και computational time αλλά διαφοροποιείται καθώς χρησιμοποιεί και underwater laser triangulation altimeter για τον υπολογισμό του βάθους από το ROV. Ο ερευνητής του εργαστηρίου που ανέλαβε την διεκπεραίωση της συγκεκριμένης διατριβής δεν είχε τη δυνατότητα να προβεί σε μία τέτοια λύση για το λόγο ότι δεν υπήρχε διαθέσιμο στο εργαστήριο ένα underwater laser triangulation altimeter.

Για το λόγο ότι η επόμενη προσέγγιση [6] η οποία θα αναλυθεί, αφορά post-processing λύση και δεν υπήρχαν φανερά ιδιαίτερα χρήσιμα στοιχεία για να βοηθηθεί η παρούσα περίπτωση, που αφορά real-time και in situ κατάσταση.

Η ομάδα του E. Nocerino et al., σε πρόσφατη έρευνα [7] του 2020 δημοσίευσε την προσπάθεια τους για ενίσχυση των μέσων πλοήγησης με ROV, όπου υπήρχαν 2 πτυχές, η μία πτυχή όπου μιλούσε για την ανακατασκευή ενός δυσδιάστατου μωσαϊκού και η δεύτερη αφορούσε μία προβολή των γεωμετρικά μετασχηματισμένων εικόνων σε ένα επίπεδο νέφος σημείων (point cloud), η συγκεκριμένη έρευνα βοήθησε πολύ ώστε να ξεδιαλυθεί το αλγοριθμικό μοτίβο που έπρεπε να ακολουθηθεί στη περίπτωση αυτής της διατριβής, αφού κατά κάποιον τρόπο είναι συνδυασμός των δύο πτυχών της συγκεκριμένης έρευνας. Ένα ορθοφωτογραφικό μωσαϊκό, προοπτικά μετασχηματισμένο το οποίο αναπαραγάγεται σε πραγματικό χρόνο.

Πολύ σημαντικό σε αυτή την έρευνα, ήταν το αλγοριθμικό ερώτημα του πως θα ανιχνεύονται τα σημαντικά σημεία πάνω σε 2 συνεχόμενες (sequential) και επικαλυπτόμενες (overlapped) εικόνες και πως θα αντιστοιχούνταν τα κοινά σημεία, έτσι έγινε μία εκτενής έρευνα στους διάφορους αλγόριθμους.

Σύμφωνα με επιστημονικό κείμενο [8] που αναλύει και συγκρίνει Image Matching αλγόριθμους καλύτερος αλλά και πιο αργός σε υπολογιστικό χρόνο κατά την επεξεργασία θεωρείται ο Scale Invariant Feature Transform (SIFT), ακολουθούν ο Speed Up Robust Feature (SURF), Binary Robust Independent Elementary Features (BRIEF) και τέλος ο oriented FAST rotated BRIEF (ORB) αλγόριθμος.

Ο SIFT [9] επικεντρώνεται στα δυνατά χαρακτηριστικά που μπορεί να ανιχνεύσει στους υπάρχοντες detectors και descriptors κρατώντας τα σε μια «νοητή» βάση δεδομένων και με το που εμφανιστεί μία νέα εικόνα με τα ίδια χαρακτηριστικά, γίνεται η ταυτοποίηση και αντιστοιχεί τις εικόνες μεταξύ τους.

Ένας άλλος πολύ ικανός αλγόριθμος για τέτοιου είδους εργασίες είναι ο SURF [10] του Herbert Bay και της ομάδας του, ο οποίος ανιχνεύει ένα descriptor βασισμένος στο Haar Wavelet [11] της εκάστοτε εικόνας.

Βασισμένος στη δυαδική κωδικοποίηση μίας εικόνας ο αλγόριθμος BRIEF [12] προσπαθεί να υπολογίσει τον descriptor της εικόνας μέσα από μικρά και γρήγορα density tests.

Στην συγκεκριμένη υλοποίηση χρησιμοποιήθηκε ο αλγόριθμος ORB [13] που είναι μία συνένωση των αλγορίθμων FAST και BRIEF ο οποίος αναλύθηκε πιο πάνω. Ο αλγόριθμος Features from Accelerated Segment Test (FAST) [14], αναλύει σε μικρά segments την εκάστοτε εικόνα προσπαθώντας να ανιχνεύσει τις ενδιαφέρουσες γωνίες σε κάθε segment και σημαντικά σημεία. Ο λόγος που χρησιμοποιήθηκε από τον ερευνητή ο αλγόριθμος ORB είναι για περιοριστικούς λόγους που υπήρξαν στην υλοποίηση και γιατί θεωρείται μία αποτελεσματική εναλλακτική λύση. Σε συνδυασμό με τον ORB χρησιμοποιήθηκε ο Brute Force Matcher γιατί σύμφωνα με μία έρευνα [15] ο συνδυασμός τους μπορεί να δώσει πολύ καλά αποτελέσματα όπως επίσης και πολύ γρήγορη επεξεργασία σε υπολογιστικό χρόνο.

## 1.2 Τεχνικό Υπόβαθρο

Το τεχνικό υπόβαθρο της παρούσας διατριβής χωρίζεται σε δύο μέρη, το θεωρητικό υπόβαθρο στο οποίο θα αναλυθούν όλες οι έννοιες και όλες οι τεχνολογίες που χρησιμοποιούνται στη συγκεκριμένη πτυχιακή αλλά και που ερευνήθηκαν για την καλύτερη κατανόηση της γενικής θεματολογίας και το υπόβαθρο υλοποίησης όπου θα αποδοθεί περιγραφή για όλα τα εργαλεία και όλες τις γλώσσες προγραμματισμού που χρησιμοποιήθηκαν για την πραγμάτωση της παρούσας εργασίας.

## 1.2.1 Θεωρητικό Υπόβαθρο

### 1.2.1.1 *Image Matching Algorithms*

Αλγόριθμοι που ο σκοπός τους είναι να βρουν σε μία εικόνα αναφοράς και μίας εικόνας στόχου ή σε ένα δυναμικό ζεύγος (αλλάζει, καθώς εφαρμόζεται σε μία αλληλουχία σχετικών εικόνων) εικόνων κοινά σημεία που στην ουσία χαρακτηρίζουν την επικαλυπτόμενη περιοχή που περιγράφει το υπό επεξεργασία ζεύγος εικόνων.

### 1.2.1.2 *Oriented FAST Rotated BRIEF (ORB)*

Μία αλγοριθμική υλοποίηση Image Matching που είναι η συνένωση του αλγόριθμου FAST με τον αλγόριθμο BRIEF που είναι ικανό να αντιστοιχίσει δύο εικόνες ανεξαρτήτως του προσανατολισμού της κάθε εικόνας που πρέπει να επεξεργαστεί και δεν ενοχλείται από τον θόρυβο (noise) που υπάρχει σε μια εικόνα. Πολύ γρήγορος σε θέματα υπολογιστικού χρόνου και αποτελεί μία αρκετά καλή και εύκολη προσέγγιση σε επίπεδο πηγαίου κώδικα.

### 1.2.1.3 *Brute Force Matcher (BF Matcher)*

Ένας απλός αλγόριθμος σε θέμα υλοποίησης, ουσιαστικά θεωρεί τον descriptor της μίας εικόνας ως σημείο αναφοράς (descriptor αναφοράς) και τον συγκρίνει με την άλλη εικόνα κάνοντας κάποιους υπολογισμούς μέτρησης της απόστασης μεταξύ των σημείων που ανιχνεύει και επιστρέφει το σημείο με την μικρότερη απόσταση από τον descriptor αναφοράς.

### 1.2.1.4 *Detector*

Αλγόριθμος [15] που διαλέγει ενδιαφέροντα σημεία σε μία εικόνα βάσει κάποιων κριτηρίων που ορίζονται από τον εκάστοτε αλγόριθμο.

### 1.2.1.5 *Descriptor [16]*

Ένα διάνυσμα τιμών οι οποίες χαρακτηρίζουν μία εικόνα και την καθιστούν μοναδική.

### 1.2.1.6 *Homography [17]*

Ένας 3x3 πίνακας ο οποίος δείχνει την συσχέτιση μεταξύ 2 εικόνων με 8 DoF (Degrees of Freedom) όσο αφορά τον μετασχηματισμό τους.

### **1.2.1.7 RANSAC**

Ένας αλγόριθμος [18] για την εκτίμηση παραμέτρων ενός μοντέλου με τυχαία δειγματοληψία από τα ενδιαφερόμενα δεδομένα (data) χρησιμοποιώντας την μέθοδο ελάχιστων τετραγώνων. Χρησιμοποιείται κατά κόρον στο πεδίο της Υπολογιστικής Όρασης σε ερευνητικές περιπτώσεις όπως Structure from Motion, Image Stitching κλπ.

### **1.2.1.8 Perspective Transformation**

Η μετατροπή της αρχικής προβολής μίας εικόνας σε προοπτική προβολή [19].

## **1.2.2 Τεχνικό Υπόβαθρο**

### **1.2.2.1 Python**

Γλώσσα προγραμματισμού με κύριο πλεονέκτημα την εύκολη εκμάθηση από κάποιον που έχει βασικές γνώσεις προγραμματισμού και κύριο μειονέκτημα την ανάγκη μεγάλου υπολογιστικού χρόνου κατά την επεξεργασία. Χρησιμοποιείται σε διάφορα tasks που αφορούν το πεδίο της Υπολογιστικής Όρασης.

### **1.2.2.2 PyCharms IDE**

Περιβάλλον ανάπτυξης λογισμικών συστημάτων ειδικό για την γλώσσα προγραμματισμού Python. Πολύ φιλικό περιβάλλον προς τον προγραμματιστή με κύριο χαρακτηριστικό την εκτέλεση(execution/run) του πηγαίου κώδικα χωρίς την χρήση Command Line Interpreter.

### **1.2.2.3 OpenCV**

Ανοιχτού πηγαίου κώδικα βιβλιοθήκη [20] εντολών που χρησιμοποιείται κυρίως σε δουλειές που αφορούν την επεξεργασία εικόνας και την Υπολογιστική Όραση. Διαθέσιμη σε πολλές γλώσσες προγραμματισμού όπως C++ και Python.

### **1.2.2.4 numpy**

Ανοιχτού πηγαίου κώδικα βιβλιοθήκη [21] εντολών της Python που βοηθά στην πραγμάτωση δύσκολων μαθηματικών εντολών ή μελέτη μαθηματικών μοντέλων μέσα από το προγραμματισμό.

### **1.2.2.5 BlueROV2**

Ένα από τα πιο προσιτά υψηλής απόδοσης ROV στην παγκόσμια αγορά. Δίνει την ευκαιρία στο χρήστη να το διαμορφώσει όπως ακριβώς θέλει μέσα από τα διάφορα εξαρτήματα που κυκλοφορούν από την εταιρεία. Χρησιμοποιεί Raspberry Pi 3 Model B και λογισμικά ανοικτού πηγαίου κώδικα (open source code).

### **1.2.2.6 Fathom Tether Cable**

Τα καλώδια τα οποία έρχονται από την εταιρεία Blue Robotics είναι τα Fathom Tether, τύπου UTP 26AWG (ο τύπος αυτός, είναι και αυτό που υποδηλώνει ότι μπορούν να κάνουν τόσο βαρετές δουλειές, υποβρύχια και σε αρκετό βάθος), σε ζεύγη τύπου CAT5. Λειτουργεί ακριβώς όπως ένα CAT5, άρα έχει ένα bandwidth της τάξεως του 1 Gbps που συνεπάγεται σε περίπου 125 MB/s.

### **1.2.2.7 Raspberry Pi 3 Model B/ Microcomputers**

Τα Raspberry Pi είναι μία σειρά μικροϋπολογιστών μονής πλακέτας κατασκευασμένα από το Raspberry Pi Foundation που εδρεύει στο Ηνωμένο Βασίλειο. Χρησιμοποιούνται σε πολλές εφαρμογές ρομποτικής [22].

Το συγκεκριμένο που είναι συνδεδεμένο στο ROV που σχετίζεται με την έρευνα έχει τα πιο κάτω τεχνικά χαρακτηριστικά:

1. Quad Core 1.2GHz Broadcom BCM2837 64bit CPU
2. 1GB RAM
3. BCM43438 wireless LAN and Bluetooth Low Energy (BLE) on board
4. 100 Base Ethernet
5. 40-pin extended GPIO
6. 4 USB 2 ports
7. 4 Pole stereo output and composite video port
8. Full size HDMI
9. CSI camera port for connecting a Raspberry Pi camera
10. DSI display port for connecting a Raspberry Pi touchscreen display
11. Micro SD port for loading your operating system and storing data
12. Upgraded switched Micro USB power source up to 2.5A

## 2 Μεθοδολογία Έρευνας

Στο κεφάλαιο «Μεθοδολογία Έρευνας» θα αναλυθεί ο τρόπος με τον οποίο λύθηκε το υπάρχον πρόβλημα της υποβρύχιας πλοήγησης με ROV για σκοπούς έρευνας. Τα βήματα θα αναλυθούν με τρόπο ώστε να μπορεί ο αναγνώστης να εκτελέσει τη λύση για τα δικά του προβλήματα έχοντας τις απαραίτητες γνώσεις τα απαραίτητα εργαλεία και το αναγκαίο υλικό (hardware).

Το πρώτο βήμα πάντα σε μία έρευνα, είναι η μελέτη της βιβλιογραφίας που εγκλείει παρόμοιες περιπτώσεις έτσι ώστε ο ερευνητής να μπορέσει να σχηματίσει πιθανές δικές τους λύσεις και να πάρει απόψεις από υλοποιήσεις που απέτυχαν ή υλοποιήσεις οι οποίες δεν πρόκειται να βοηθήσουν στη δική του περίπτωση.

Εφόσον το εγχείρημα αυτό απαιτούσε γνώσεις και σε υλικό (hardware) όπως και σε λογισμικό (software) αφού εμπειρίκλεια προβληματισμούς και σκέψεις που αφορούσαν και τα 2 κομμάτια της υλοποίησης. Αρχικά, έγιναν οι σκέψεις σε θέματα υλικού και έγινε μία εκτενής έρευνα η οποία οδήγησε σε 3 πιθανές προοπτικές που θα έδιναν την λύση στους προβληματισμούς του ερευνητή.

Ένα από τα κυριότερα προβλήματα που έτυχαν σε θέματα υλικού είναι ότι λόγω του ότι όλη η διαδικασία γίνεται υποβρύχια δεν υπάρχει καλή συνδεσιμότητα σε Bluetooth και Wi-Fi, ο λόγος είναι ότι η εκπεμπόμενη συχνότητα είναι στα 2.4GHz και της συγκεκριμένης συχνότητας τα μόνια, κάτω από το νερό εξασθενούν [23] και το σήμα εξασθενεί, αυτό συνεπάγεται σε ένα εύρος του σήματος κάποιων ιντσών (inches) .

Οι 3 πιθανές λύσεις αναλύονται και περιγράφονται πιο κάτω:

### **Λύση 1:**

Χρήση μίας GoPro Hero 3/4/5 Action Camera η οποία θα εγκλειόταν μέσα σε ένα υδατοστεγές PVC κουτί για την ενίσχυση του Wi-Fi σήματος που θα έστελνε στο Raspberry Pi το live streaming που θα μεταδίδεται σε έναν ηλεκτρονικό υπολογιστή με τη βοήθεια του λογισμικού ArduSub και θα χρησιμοποιείται για την αποκοπή frames (καρέ) τα οποία θα χρησιμοποιούνταν για την παραγωγή του τελικού προϊόντος.



***Η Λύση 1 απορρίφθηκε για το λόγο ότι θα υπήρχαν δυσκολίες στη κατασκευή του PVC κουτιού σε συνδυασμό με τους προβληματισμούς για την αντοχή της μπαταρίας της κάμερας με ενεργοποιημένο το Wi-Fi.***

### **Λύση 2:**

Χρήση μίας GoPro Hero 3/4/5 Action Camera η οποία θα έστελνε τα δεδομένα με την χρήση ενός Wi-Fi Cable Extender σε ένα δεύτερο υπολογιστή (ο πρώτος θα χρησιμοποιούταν μόνο για το live streaming στο ArduSub) για να μην ρισκάρουμε την υπερφόρτωση του πρώτου ηλεκτρονικού υπολογιστή, και ακολουθώντας μετά παρόμοια βήματα με τη Λύση 1.

***Η Λύση 2 απορρίφθηκε λόγω περιορισμών του προϋπολογισμού καθώς ένα Wi-Fi Cable Extender κοστίζει πάρα πολλά στην περίπτωση τη δική μας, καθώς χρειαζόμασταν τεράστιο σε μήκος καλώδιο. Ακόμη δυσκολίες υπήρχαν και στην χρήση δεύτερου υπολογιστή.***

### **Λύση 3:**

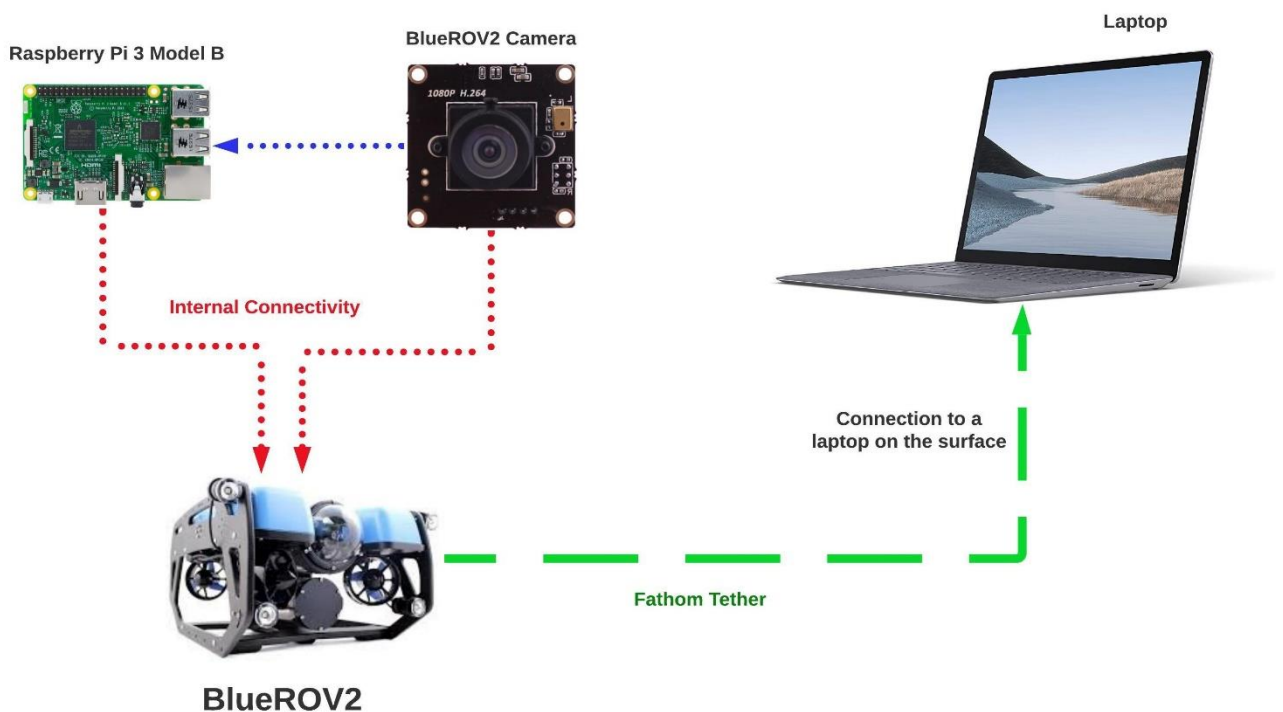
Χρήση του υπάρχοντος Raspberry Pi μαζί με μία Pi Camera (ενδεικνυόμενη κάμερα από την κατασκευάστρια εταιρεία του μικροϋπολογιστή για αποφυγή καθυστερήσεων και τεχνικών προβλημάτων συνδεσιμότητας) που θα στέλνει τα δεδομένα με την χρήση του καλωδίου βαρειών εργασιών Fathom Tether στον υπολογιστή που θα υπάρχει στην ακτή και ακολουθώντας μετά παρόμοια βήματα με τη Λύση 1.

***Ακολουθήθηκε η Λύση 3 γιατί όλα τα απαραίτητα σε υλικό ήταν ήδη διαθέσιμα από το εργαστήριο Photogrammetric-Vision Lab και γιατί ήταν η πιο απλή σε συνδεσιμότητα και διάταξη υλικού.***

Γενικότερα προβλήματα που προέκυψαν κατά την πρώτη διερεύνηση για την διάταξη της προσπάθειας αυτής σε υλικό εκτός από τις συνδεσιμότητες Bluetooth και Wi-Fi, ήταν η συνδεσιμότητα της GoPro με το Raspberry Pi καθώς μέσα από έρευνα παρατηρήθηκε ότι πολλοί χρήστες ανέφεραν προβλήματα στην επικοινωνία των 2 συσκευών.

Μετέπειτα στάδιο, ήταν η επιλογή της γλώσσας προγραμματισμού που θα χρησιμοποιείτο για την υλοποίηση του λογισμικού. Έγινε η επιλογή της Python παρά το ότι υστερεί σε θέματα απόδοσης σε υπολογιστικό χρόνο από ότι η επιλογή της C++. Επιλέγηκε η Python λόγω της ευκολίας της για την υλοποίηση του κώδικα και της γενικότερης χρήσης της.

Στο Διάγραμμα 1 που φαίνεται πιο κάτω φαίνεται η διάταξη του υλικού.



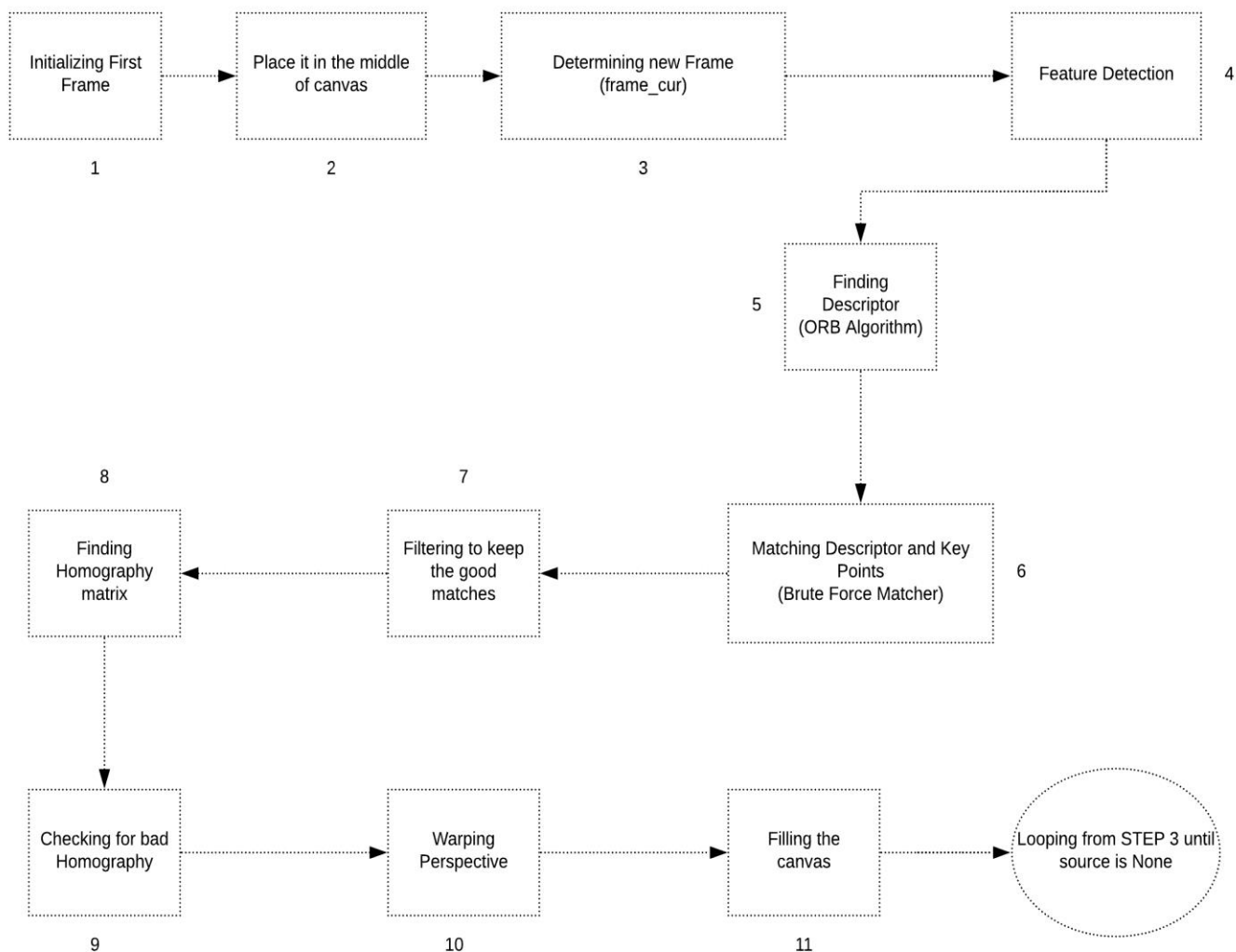
**Διάγραμμα 1:** Διάταξη υλικού (hardware)

Αναλυτικά η διάταξη του υλικού, όπως φαίνεται στο Διάγραμμα 1, η κάμερα του BlueROV2 είναι εγκατεστημένη στο Raspberry Pi 3 Model B που είναι εγκατεστημένο πάνω στο ROV. Το BlueROV2 με ένα καλώδιο Fathom Tether στέλνει τα δεδομένα στον υπολογιστή που βρίσκεται εγκατεστημένο στην ακτή. Η κάμερα που έρχεται από την κατασκευάστρια εταιρεία Blue Robotics του BlueROV2 είναι συνδεδεμένη στο Raspberry Pi σε συγκεκριμένη θύρα (port) όπως ενδείκνυται από την Blue Robotics. Το Raspberry Pi είναι στερεωμένο μέσα στο κυρίως μέρος του BlueROV2.

Η διάταξη αυτή, είναι η διάταξη η οποία αποφασίστηκε μετά από την αρχική διερεύνηση που έγινε και αφορά την Λύση 3 όπως αυτή περιγράφεται σε προηγούμενη παράγραφο του συγκεκριμένου κεφαλαίου.

Επόμενο στάδιο και αφού επιλέχθηκε και η γλώσσα προγραμματισμού που θα υλοποιείτο πάνω σε αυτή το λογισμικό, έπρεπε να γίνει η εκμάθηση της γλώσσας από τον ερευνητή ώστε να βρίσκεται σε πολύ καλό επίπεδο για να μπορεί να αντιμετωπίσει ένα πρόβλημα Υπολογιστικής Όρασης όπως το παρόν. Πριν ξεκινήσει η υλοποίηση, έγινε εκτενής μελέτη σε θέματα Υπολογιστικής Όρασης και Τεχνητής Νοημοσύνης, όπως και μελέτη διαφόρων παραδειγμάτων που επιλύουν παρόμοια προβλήματα ή κομμάτια του παρόντος προβλήματος για να είναι δυνατόν να αφομοιωθεί και να σχεδιαστεί καλά μια μελλοντική λύση όσο αφορά το λογισμικό. Επομένως, αφού ήταν σίγουρο πως για την επίλυση τέτοιου προβλήματος θα γινόταν η χρήση της ανοιχτού πηγαίου κώδικα βιβλιοθήκη OpenCV, μελετήθηκε το documentation (έγγραφο τεκμηρίωσης) της βιβλιοθήκης για να βρεθούν ρουτίνες (routines) που θα ήταν χρήσιμες για την υλοποίηση.

Στην επόμενη σελίδα, στο Διάγραμμα 2, φαίνεται το πως υλοποιήθηκε σε κατά μέρη σημεία το λογισμικό της παρούσας πτυχιακής εργασίας. Αποφασίστηκε αυτή η ροή διαδικασιών και επεξεργασίας των δεδομένων για να παραχθεί το τελικό προϊόν μετά από πολλή έρευνα και μελέτη παρόμοιων υλοποιήσεων ή υλοποιήσεων με τον ίδιο σκοπό.



**Διάγραμμα 2:** Διάγραμμα Ροής Διαδικασιών

Στο Διάγραμμα 2 φαίνεται το Διάγραμμα Ροής Διαδικασιών όπως αναφέρθηκε και πιο πάνω, δηλαδή με ποιο τρόπο το συγκεκριμένο λογισμικό σύστημα επεξεργάζεται τα δεδομένα που λαμβάνει (στην προκειμένη περίπτωση, τα καρτέ από το live streaming) έτσι ώστε να φτάσει να παραγάγει το τελικό προϊόν (ένας χάρτης πλοήγησης σε πραγματικό χρόνο).

Πιο κάτω θα ακολουθήσει η επεξήγηση της κάθε διαδικασίας ξεχωριστά με παράθεση κώδικα σε μορφή ψευδοκώδικα (pseudocode) όπου δύναται και όλων των ενδιαφερόντων μαθηματικών εκφράσεων που χαρακτηρίζουν την κάθε διαδικασία.

### **Διαδικασία 1:**

Στη Διαδικασία 1 γίνεται η αρχικοποίηση του πρώτου frame που λαμβάνεται από το live camera feed.

1. Set a variable as the first frame of the live camera feed.

### **Διαδικασία 2:**

Στη Διαδικασία 2, το πρώτο frame τοποθετείται στη μέση του canvas για να ακολουθήσει η δημιουργία του χάρτη βάσει αυτού.

1. Set a canvas with dimensions x, y.
2. Calculate the middle of canvas (x, y) with dimensions (x, y) or set an offsetX and an offsetY.

Below it's described mathematically how offsetX and offsetY are set:

$$offsetX = H \begin{bmatrix} 1.02 & -3.77 & offsetX \\ 2.65 & -2.38 & 7.45 \\ -4.09 & 1.72 & -2.34 \end{bmatrix}$$

$$offsetY = H \begin{bmatrix} 1.02 & -3.77 & 7.45 \\ 2.65 & -2.38 & offsetY \\ -4.09 & 1.72 & -2.34 \end{bmatrix}$$

$H \rightarrow$  Homography Matrix, example

3. Paste the frame to that point.

### **Διαδικασία 3:**

Μέσα από την επεξεργασία της Διαδικασίας 3, ορίζεται μία νέα μεταβλητή που περιέχει ένα frame που είναι ουσιαστικά το παρόν frame.

1. Set a new variable as each current frame of the live camera feed.

### **Διαδικασίες 4-6:**

Στη συγκεκριμένη συνέχεια διαδικασιών γίνεται η εύρεση και η σχεδίαση των key points του εκάστοτε frame που περνά από επεξεργασία από το λογισμικό με τη χρήση ORB και Brute Force Matcher.

1. Create an ORB detector.
2. Create a Brute Force Matcher.

3. Detect with ORB the key points.
4. Compute the descriptors.
5. Match with BF Matcher the current descriptor with previous descriptor
6. Draw the key points.

#### **Διαδικασία 7:**

Στην παρούσα διαδικασία γίνεται ένας γρήγορος έλεγχος ώστε να χρησιμοποιηθούν στην συνέχεια μόνο τα «καλά» matches.

1. Set a limit.
2. Set length of matches.
3. If length of matches is bigger than the limit

return

#### **Διαδικασία 8:**

Στην Διαδικασία 8 γίνεται ένα από τα πιο σημαντικά βήματα της υλοποίησης, καθώς σε αυτή τη διαδικασία γίνεται η εύρεση του πίνακα ομογραφίας (Homography Matrix). Η μέθοδος (method) που χρησιμοποιήθηκε για να βρεθεί η ομογραφία ήταν η RANSAC [24] με threshold 2,0 (methodThreshold).

1. Set key points of previous frame (kpprev)
2. Set key points of current frame (kpcur)
3. findHomography(kpprev, kpcur, method, methodThreshold)

findHomography is an OpenCV module for calculating easily the homography matrix (H).

Ο πίνακας Ομογραφίας είναι ένας 3x3 πίνακας με οκτώ DoF (degrees of freedom) που υπολογίζεται αυτόματα από την εντολή findHomography. Έχει τη μορφή που φαίνεται πιο κάτω.

$$H = \begin{bmatrix} 1.02 & -3.77 & 7.45 \\ 2.65 & -2.38 & 1.19 \\ -4.09 & 1.72 & -2.34 \end{bmatrix}$$

*H* → Homography Matrix, example

Η ομογραφία περιγράφει τον μετασχηματισμό μεταξύ δύο επιπέδων και χαρακτηρίζεται από την παρακάτω μαθηματική έκφραση.

$$s \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} h \\ y \\ 1 \end{bmatrix}$$

### **Διαδικασία 9:**

Συμπεριλαμβάνεται ένα πολύ σημαντικό κομμάτι της υλοποίησης στη Διαδικασία 9 καθώς για τον έλεγχο «κακής» ομογραφίας, χρειάζεται ο ορισμός των γωνιών που θα χρησιμοποιηθούν και στη Διαδικασία 10.

1. Set corner0, corner1, corner2, corner3.  
More explanation in Procedure 10.
2. Apply homography on corners.
3. Check if they are still clockwise between one another

### **Διαδικασία 10:**

Μία από τις σημαντικότερες διαδικασίες, καθώς σε αυτή εφαρμόζεται ο προοπτικός μετασχηματισμός (perspective transformation) σε κάθε frame που περνά από την επεξεργασία.

1. Set corner0, corner1, corner2, corner3.  
corner0 = [0,0]  
corner1 = [width,0]  
corner2 = [width, height]  
corner3 = [0, height]  
Width and Height are referred to frame's width and height.
2. Set an array of corners.  
corners = {corner0, corner1, corner2, corner3}
3. Apply perspective transformation to frame's corners.  
transformedCorners = perspectiveTransformation(corners,H)

Ο προοπτικός μετασχηματισμός είναι ο άμεσα συσχετισμένος μετασχηματισμός με την Ομογραφία και μαθηματικά εκφράζεται όπως παρακάτω:

Ουσιαστικά μετασχηματίζεται η πηγή (αρχικό frame, src) χρησιμοποιώντας τον πίνακα Ομογραφίας (H), για να καταλήξουμε στο μετασχηματισμένο frame (dst).

$$dst(x, y) = src \left( \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}, \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}} \right)$$

### Διαδικασία 11:

Στην Διαδικασία 11 υλοποιείται ο τρόπος με τον οποίο δημιουργείται ο χάρτης.

1. Apply transformedCorners on frame.  
outputImage is resulting after the appliance of transformedCorners on frame and after warping it perspective.
2. Copy the outputImage on canvas based on H.

Επαναλαμβάνονται οι διαδικασίες 3 έως 11 μέχρι να επεξεργαστεί όλο το live camera feed μέσω επαναληπτικού βρόχου. Το γεγονός ότι οι διαδικασίες αυτές εκτελούνται όσο υπάρχει το camera feed, σημαίνει και την υλοποίηση του συστήματος σε Real-Time.

**while** source is not **None**

execute Procedure 3 to Procedure 11

**exit**

Με το πέρας της συγγραφής του πηγαίου κώδικα ακολουθώντας το διάγραμμα ροής διαδικασιών που σχεδιάστηκε και τέθηκε υπόψη για την διεκπεραίωση του λογισμικού, έγιναν κάποιες αρχικές δοκιμές. Παρατηρήθηκε από τις δοκιμές ότι υπήρχαν αρκετά προβλήματα και διαστρεβλώσεις στα παραγόμενα ορθοφωτομωσαϊκά. Μετά από μελέτη διαπιστώθηκε ότι μπορούν να γίνουν αλλαγές στις παραμέτρους [25] του αλγόριθμου ORB, ώστε να μπορεί να δημιουργεί καλύτερες αντιστοιχίες.

Πιο κάτω αναφέρονται οι τιμές που δόθηκαν στις παραμέτρους σε μορφή ψευδοκώδικα. Μετά από κάθε συμβολισμό # (π.χ. nlevels = 100) αναφέρεται η default τιμή κάθε παραμέτρου.

```
detector = ORBcreate  
(  
500, # nfeatures = 500
```



```
1.2f, # scaleFactor = 1.2f
4, # nlevels = 8
21, # edgeThreshold = 31
0, # firstLevel = 0
2, # WTA_K = 2
HARRIS_SCORE, # scoreType = HARRIS_SCORE
21, # patchSize = 31
20 # fastThreshold = 20
)
```

Ακόμη, μία παράμετρος που δοκιμάστηκε αρκετά, ήταν το πόσες αντιστοιχίες από όσες εντοπίζει κατά την επεξεργασία ο αλγόριθμος ORB να λαμβάνει υπόψη του. Παρατίθεται πιο κάτω ο ψευδοκώδικας.

```
matches = matches[(len(matches), 40)]
```

Ως len αναφέρεται ο αγγλικός όρος length, άρα ουσιαστικά ο πίνακας αντιστοιχιών να έχει ένα μέγεθος 40 αντιστοιχιών.

## **Επιπρόσθετη Λύση**

Το λογισμικό που δημιουργήθηκε για τους σκοπούς αυτής της διπλωματικής εργασίας, με μεγάλη ευκολία για τον ερευνητή, μπορεί να δημιουργηθεί σε 2 εκδόσεις. Η διαφορά της δεύτερης έκδοσης, Underwater Cyclops ft. FPS Limitation, όπως την ονομάζει ο δημιουργός, είναι ότι επιλέγει ορισμένα από τα frames της πηγής με συγκεκριμένο ρυθμό (που μπορεί να ρυθμιστεί εύκολα μέσω αλλαγής στο κώδικα) για να δημιουργήσει το μωσαϊκό. Δεν θα αναλυθεί περαιτέρω η συγκεκριμένη λύση, στη συνέχεια της διπλωματικής εργασίας, καθώς δεν ήταν εξ αρχής σκοπός, ούτε δίνει ιδιαίτερη διαφορά στη λύση, εκτός από κάποια πιο γρήγορη σε δευτερόλεπτα υπολογιστικού χρόνου λύση.

## **3 Αποτελέσματα και Εφαρμογή**

Στο κεφάλαιο «Αποτελέσματα και Εφαρμογή» θα αναλυθούν αρχικά τα Datasets που είχε στη διάθεση του ο ερευνητής για να μπορεί να ελέγξει το λογισμικό και να πραγματοποιήσει οποιουσδήποτε ελέγχους αλλά και βελτιώσεις στο λογισμικό ώστε να λειτουργεί με το καλύτερο δυνατό υπολογιστικό χρόνο. Ακολούθως στο παρόν κεφάλαιο θα αναλυθούν τα αποτελέσματα, και αυτά που υπήρξαν κοντά στα αναμενόμενα, αλλά και τα αποτελέσματα τα οποία δείχνουν τις αδυναμίες του λογισμικού. Τέλος, θα επεξηγηθεί ακριβώς η εφαρμογή του λογισμικού Underwater Cyclops.

Η μελέτη και υλοποίηση λογισμικού, που περιγράφονται στην συγκεκριμένη διατριβή, αφορούν την πραγματοποίηση ενός λογισμικού που σε πραγματικό χρόνο θα επεξεργάζεται φωτογραφικά καρέ που λαμβάνονται από ένα uROV. Ο ερευνητής δεν είχε στη διάθεση του τέτοια δεδομένα για να ελέγξει την ακρίβεια του συστήματος που υλοποίησε αλλά δεδομένα που λήφθηκαν από άλλου είδους κάμερα και με κατάδυση. Συγκεκριμένα ήταν μία κάμερα, στερεωμένη σε μία μπάρα, και κατά τη διάρκεια της υποβρύχιας περιήγησης του δύτε, λάμβανε δεδομένα.

Στον Πίνακα 1 που φαίνεται πιο κάτω περιγράφονται οι πληροφορίες (No of Frames, Recording Time, Duration, Processing Time) που κατεγράφησαν μέσα από επτά (7) datasets που παράχθηκαν μέσα από άλλα πιο μεγάλα datasets. Ο λόγος είναι ότι κατά τα μεγάλα datasets υπήρχαν στιγμές που οι λήψεις γινόντουσαν σε περιοχές χωρίς λεπτομέρειες και υπήρχαν διαστρεβλώσεις όπως γεωμετρικές παραμορφώσεις [26] στα αποτελέσματα (θα αναλυθούν πιο κάτω) ή σε περιοχές εκτός νερού.

Τα αποτελέσματα για τα 7 datasets θα περιγραφούν και παρακάτω στη διατριβή, με την παράθεση των χαρτών που παράγονται κατά την επεξεργασία και που καθιστούν το τελικό προϊόν του λογισμικού.

Επίσης όπως ήδη αναφέρθηκε, η λύση δεν είναι η καλύτερη για αυτό σε αρκετές περιπτώσεις δεν λειτουργεί και τα αποτελέσματα είναι γεωμετρικά παραμορφωμένα. Θα παρατεθούν διαγράμματα που θα περιγράφουν και αυτές τις περιπτώσεις.

Dataset #	No of Frames	Duration (s)	Processing Time (s)
Dataset 1	1367	56	90.979
Dataset 2	740	30	49.318
Dataset 3	2592	108	170.410
Dataset 4	3497	145	240.205
Dataset 5	2277	94	151.141
Dataset 6	1446	60	97.189
Dataset 7	496	30	28.475
Dataset 8	792	33	54.465

**Πίνακας 1:** Αριθμητικές πληροφορίες αποτελεσμάτων

Για τη λήψη των αριθμητικών πληροφοριών που περιγράφουν την απόδοση του λογισμικού σε θέματα υπολογιστικού χρόνου, των αποτελεσμάτων, που πάρθηκαν μετά το τρέξιμο των επτά (7) datasets, χρησιμοποιήθηκε ένας top notch technology υπολογιστής με τα ακόλουθα τεχνικά χαρακτηριστικά:

**RAM:** 32GB

**GPU:** Asus RoG Strix 2060S Advanced Edition 8GB VRAM

**CPU:** AMD Ryzen 7 3700X 8 Cores, 16 Threads

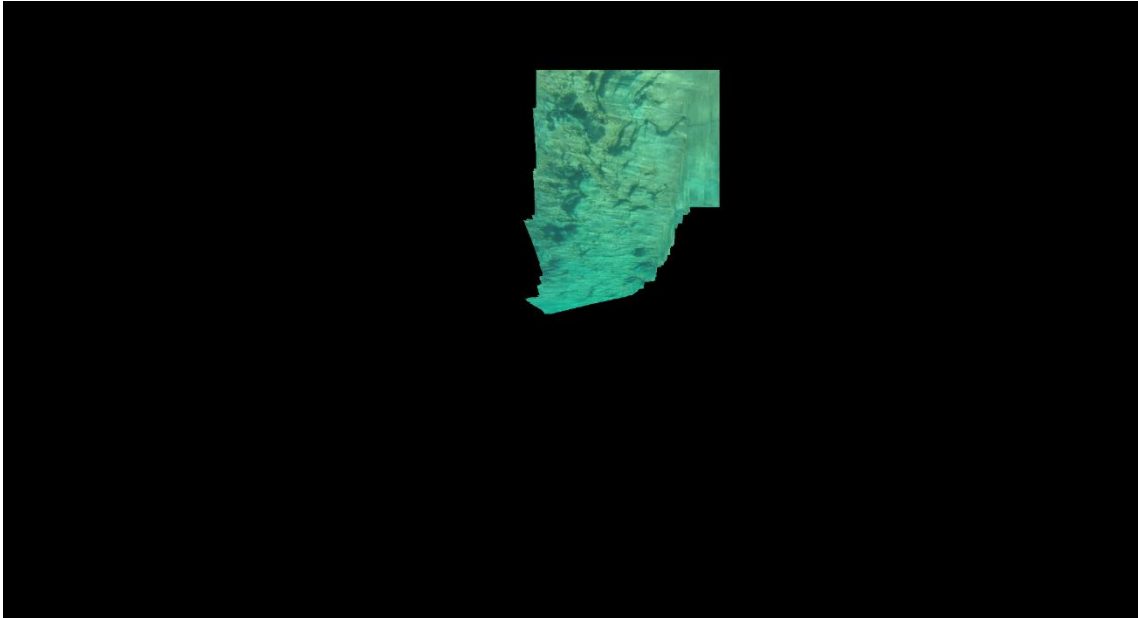
Με σύντομους και απλούς μαθηματικούς υπολογισμούς με τον συγκεκριμένο υπολογιστή ο ερευνητής μπορεί να επεξεργαστεί σε 33 sec μία αλληλουχία 500 καρέ (frames).

Το συγκεκριμένο αριθμητικό δεδομένο, που περιγράφει τον ρυθμό επεξεργασίας των 500 καρέ είναι κατά προσέγγιση του αποτελέσματος .

Τέτοιοι υπολογισμοί γίνονται για να μπορεί να υπάρξει ένα γενικό συμπέρασμα σε θέματα ταχύτητας του λογισμικού, όταν θα υπάρχει η καθυστέρηση (delay) της απόσπασης και για το λόγο ότι στο πεδίο δεν θα μπορεί να υπάρξει ένας υπολογιστής που να παρέχει τέτοια υπολογιστική δύναμη, το λογισμικό θα χρειάζεται περισσότερο χρόνο για την επεξεργασία και παραγωγή του τελικού προϊόντος.

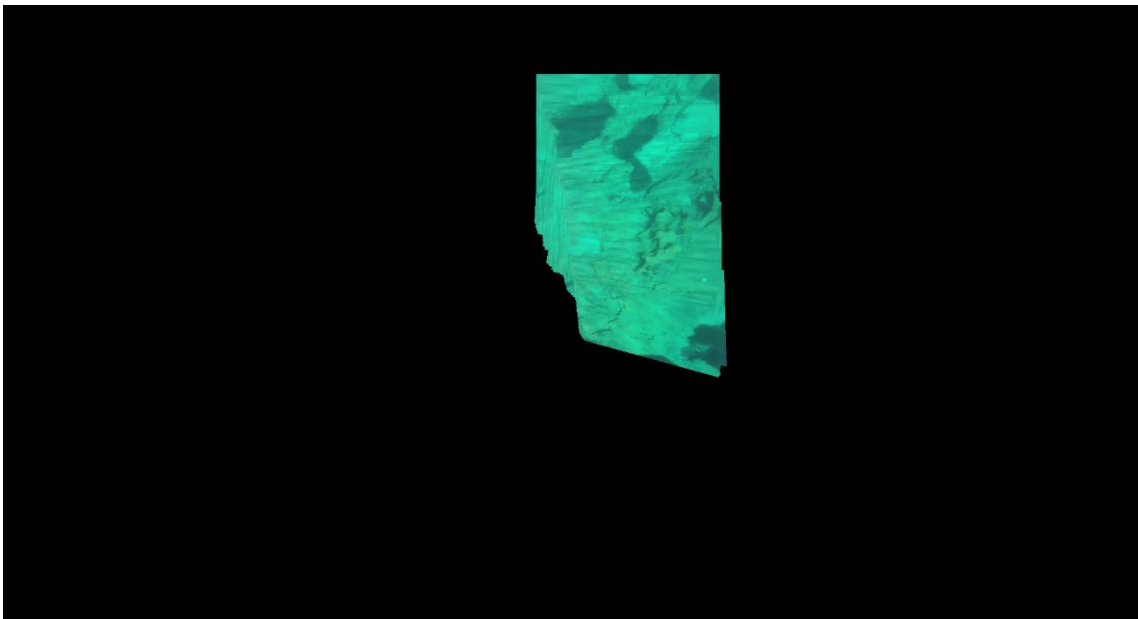
Στα Διαγράμματα που ακολουθούν φαίνεται ο χάρτης που περιγράφει το τελικό αποτέλεσμα για κάθε dataset. Τα datasets, από το Dataset 1 μέχρι και το Dataset 8 αφορούν βίντεο (αλληλουχίες συνεχόμενων καρέ) που με τη χρήση λογισμικού επεξεργασίας βίντεο, υποβαθμίστηκε στα 240p.

Ο λόγος που υποβαθμίστηκαν τα datasets στα 240p, είναι γιατί το λογισμικό, ενώνει τα frames, στις πραγματικές τους διαστάσεις, άρα η συνένωση frame διαστάσεων των 1080p θα δυσκόλευε πολύ στην παρουσίαση των αποτελεσμάτων.



**Διάγραμμα 3:** Τελικό προϊόν Dataset 1

Στο Διάγραμμα 3, φαίνεται ο παραγόμενος χάρτης από το Dataset 1, πάνω από ένα βράχο. Μπορούμε να δούμε ότι αρχικά, στο πάνω μέρος του χάρτη, τα frames, κολλάνε το ένα στο άλλο με τις σωστές τους διαστάσεις, ακολούθως, όσο τα frames περνάνε, μικραίνει και δημιουργείται μια παραμόρφωση.



**Διάγραμμα 4:** Τελικό προϊόν Dataset 2

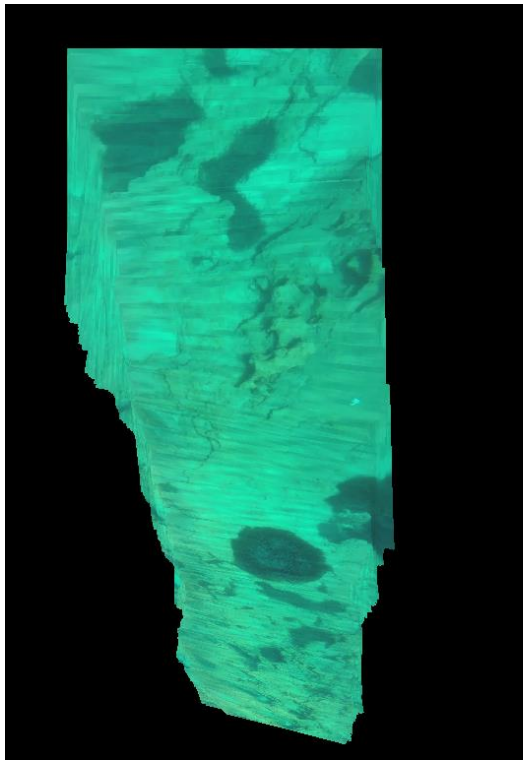
Παρατηρούμε στο Διάγραμμα 4, που φαίνεται ο χάρτης του Dataset 2, ότι τα frames καθώς περνάει ο χρόνος, δεν παραμορφώνονται όσο σε άλλες περιπτώσεις, και αυτό πιθανόν να οφείλεται στο ότι κατά τη διάρκεια του βίντεο, δεν αλλάζει τόσο πολύ τη τοποθεσία του ο δύτες, άρα ουσιαστικά ο αλγόριθμός βρίσκει περισσότερα «καλύτερα» κοινά σημεία σε κάθε ζεύγος frame σχεδόν καθ' όλη την επεξεργασία.



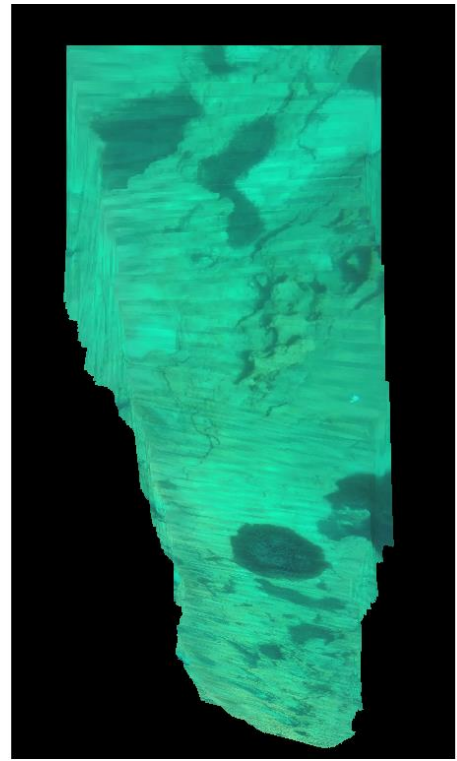
**Διάγραμμα 5:** Τελικό προϊόν Dataset 3



**Διάγραμμα 6:** Τελικό προϊόν Dataset 4



**Dataset 3**

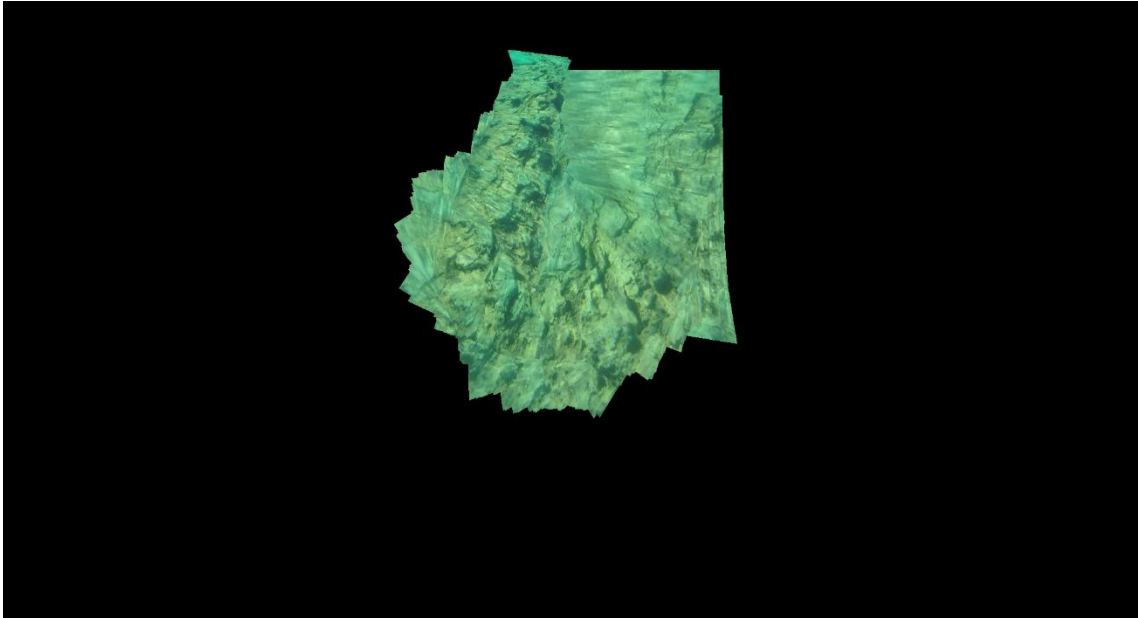


**Dataset 4**

**Διάγραμμα 7:** Σύγκριση χάρτη Dataset 3 με χάρτη Dataset 4

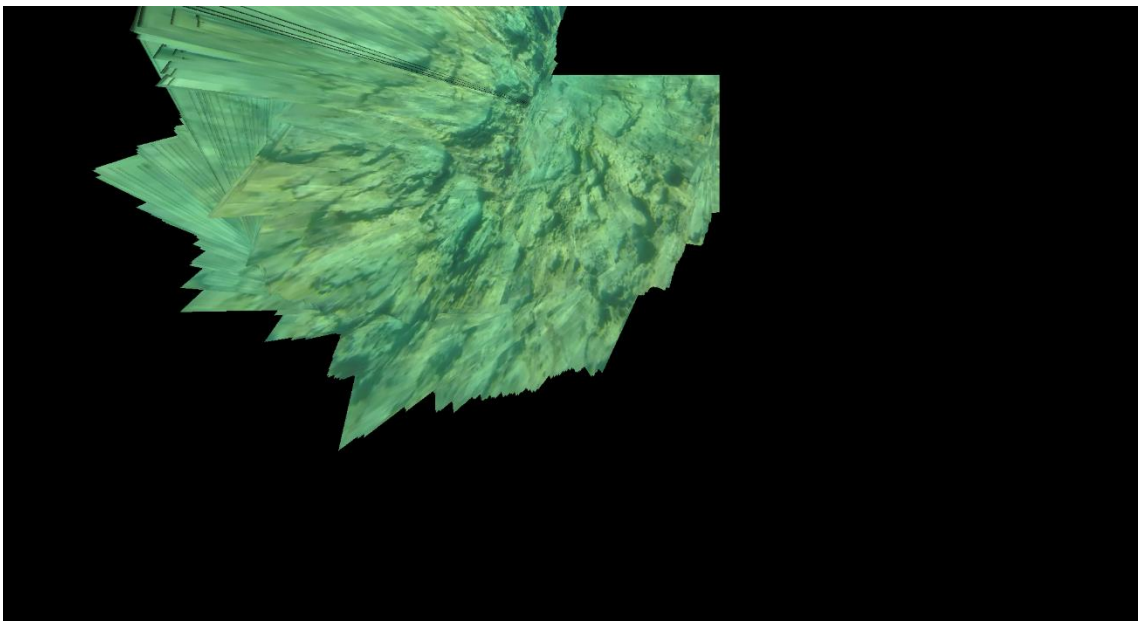
Στο Διάγραμμα 5 παρουσιάζεται ο χάρτης από το Dataset 3, και στο Διάγραμμα 6 ο χάρτης από το Dataset 4. Το Dataset 4 είναι ουσιαστικά το Dataset 3 και αρκετά επιπλέον frames. Εδώ παρουσιάζεται και φαίνεται αισθητά το πρόβλημα του λογισμικού που παράχθηκε.

Μπορούμε με γυμνό μάτι να δούμε ότι ενώ ο χάρτης του Dataset 4 θα έπρεπε να είναι πολύ διαφορετικός καθώς έχει περίπου 1000 frames περισσότερα από το Dataset 3, και να περιγράφει μεγαλύτερη έκταση του βυθού, έχει κάποιες ελάχιστες διαφορές με το χάρτη του Dataset 3.



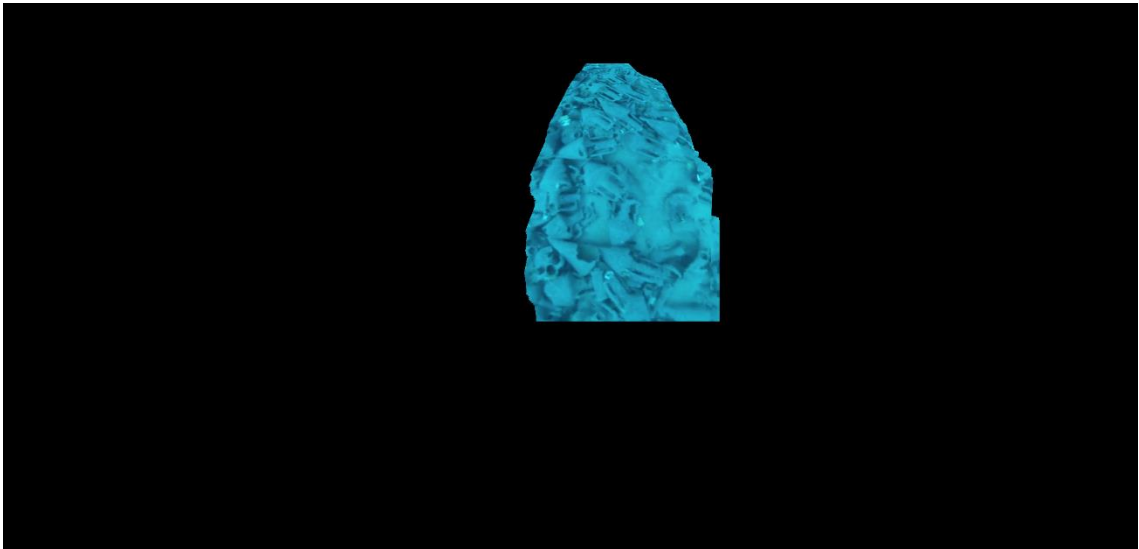
**Διάγραμμα 8:** Τελικό προϊόν Dataset 5

Το τελικό προϊόν του Dataset 5, όπως φαίνεται στο Διάγραμμα 8, είναι ένα παράδειγμα που δείχνει και κάποια από τα θετικά χαρακτηριστικά του λογισμικού, καθώς φαίνεται ξεκάθαρα ότι ο αλγόριθμος αντιλαμβάνεται την στροφή που κάνει η κάμερα πάνω από την υπό μελέτη περιοχή. Ωστόσο δεν ελλείπουν οι γεωμετρικές παραμορφώσεις.



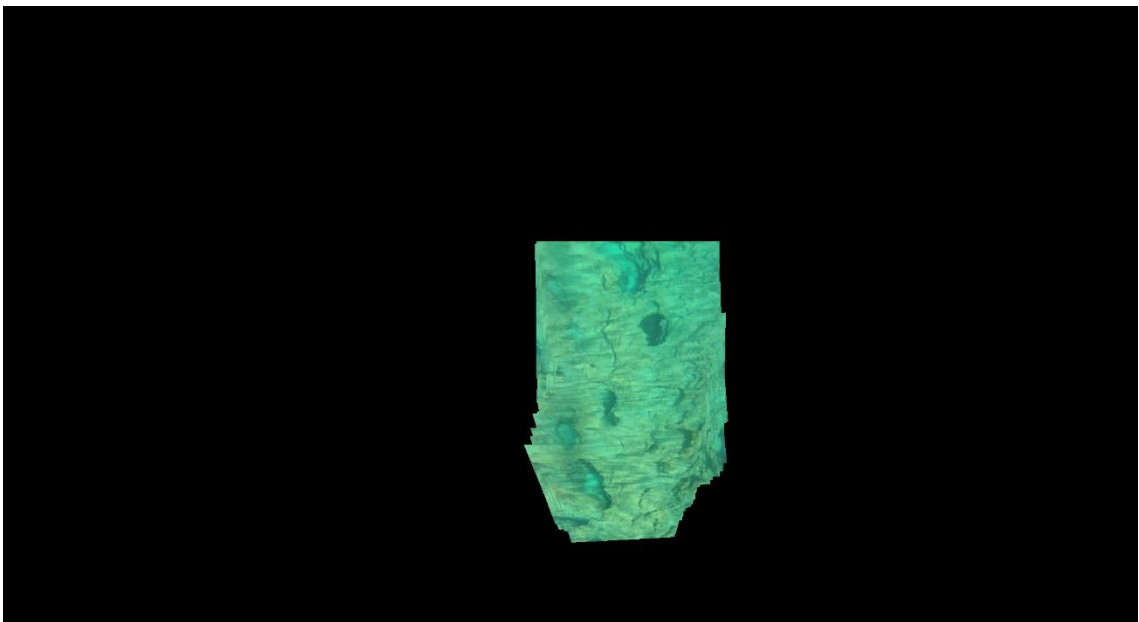
**Διάγραμμα 9:** Τελικό προϊόν Dataset 6

Ακόμη ένα παράδειγμα, που φαίνεται ότι το λογισμικό μπορεί να αντιλαμβάνεται στροφές και να τις καταγράφει στον παραγόμενο χάρτη. Και εδώ παρατηρούνται αρκετές διαστρεβλώσεις.



**Διάγραμμα 10:** Τελικό προϊόν Dataset 7

Ένα από τα καλύτερα παραγόμενα αποτελέσματα, καθώς φαίνεται ξεκάθαρα ο χάρτης που περνάει πάνω από μία υποβρύχια έκταση, αρχαιολογικά ταξινομημένη.



**Διάγραμμα 11:** Τελικό προϊόν Dataset 8

Και σε αυτή τη περίπτωση υπάρχουν παραμορφώσεις, ελάχιστες όμως σε σύγκριση με άλλες αλληλουχίες από frames.



Δοκιμές έγιναν επίσης και σε κάποια datasets των 480p και πιο κάτω στο Διάγραμμα 12 θα παρουσιαστεί το αποτέλεσμα σε μία από αυτές τις δοκιμές.

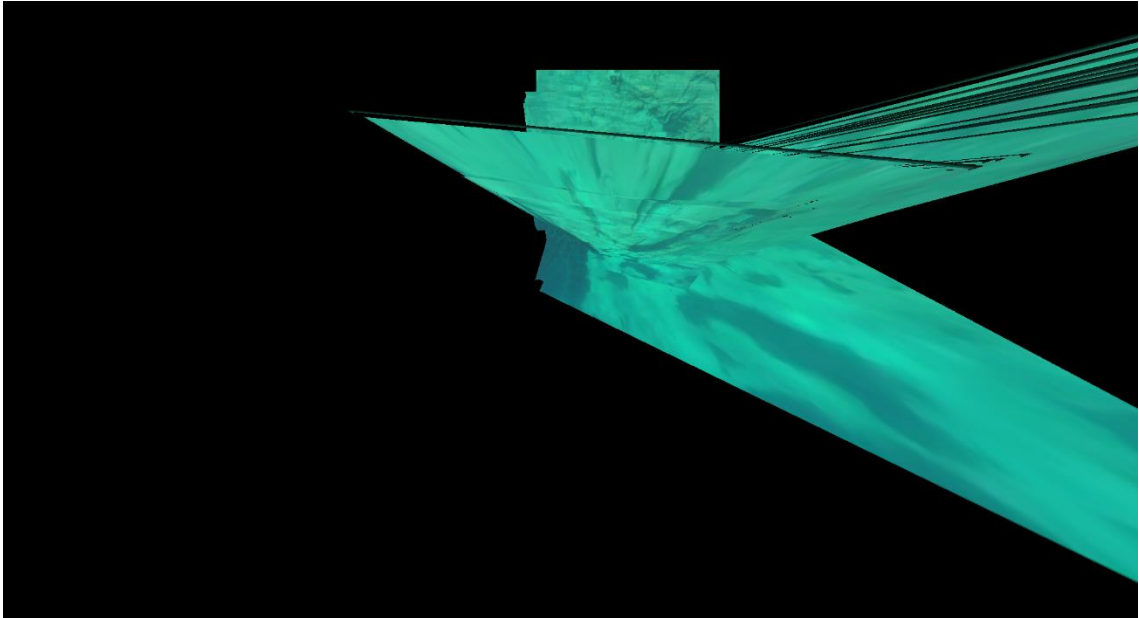


**Διάγραμμα 12:** Τελικό προϊόν Dataset 480p

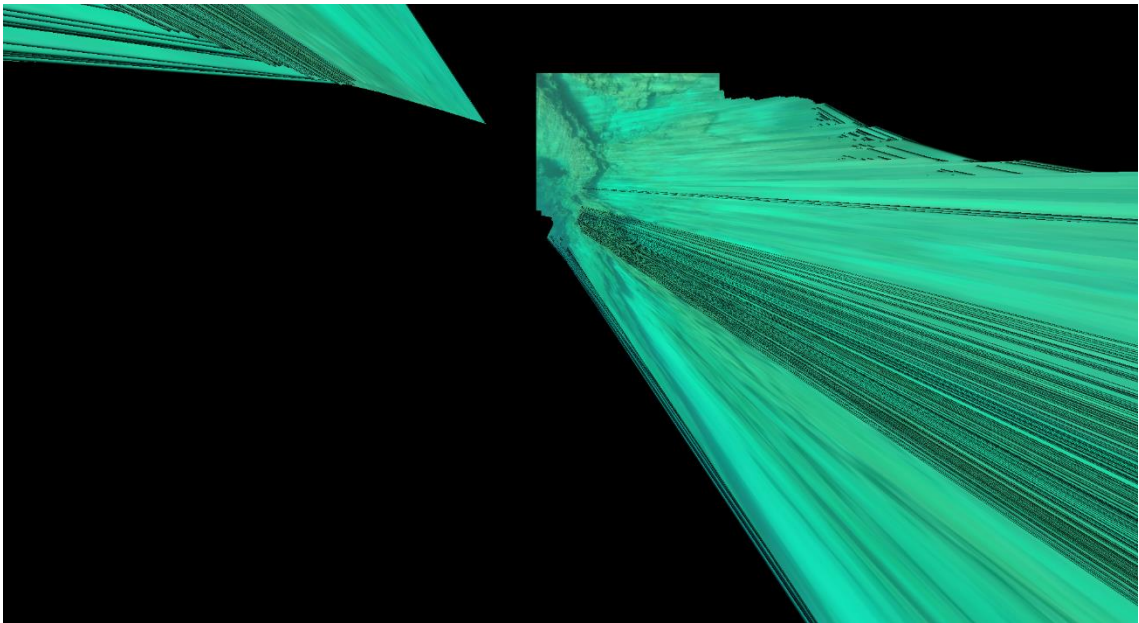
Βλέπουμε ότι στη περίπτωση επεξεργασίας των 480p, στο Διάγραμμα 12, έχουμε πιο καθαρή εικόνα όσο αφορά τις λεπτομέρειες του βυθού, αλλά δεν είναι αυτό που ενδιαφέρει σε μια τέτοια έρευνα, αλλά η καταγραφή της πορείας στο βυθό.

Αναφέρθηκε πολλάκις κατά τη διάρκεια της περιγραφής των τελικών αποτελεσμάτων ότι τα αποτελέσματα υστερούν στο να ακολουθούν τη γεωμετρική υπόσταση της προσέγγισης που ακολουθήθηκε σε αυτή την διπλωματική εργασία και αυτό οφείλεται ξεκάθαρα στην χρήση της μεθόδου της Ομογραφίας (Homography). Περαιτέρω το ζήτημα θα αναλυθεί στο Κεφάλαιο 4, «Συζήτηση Αποτελεσμάτων».

Υπήρξαν εννοείται και δοκιμές σε διάφορα βίντεο που απέτυχαν παταγωδώς όπου παρατίθενται πιο κάτω στα Διαγράμματα 13 και 14.



**Διάγραμμα 13:** Κακό τελικό προϊόν 1

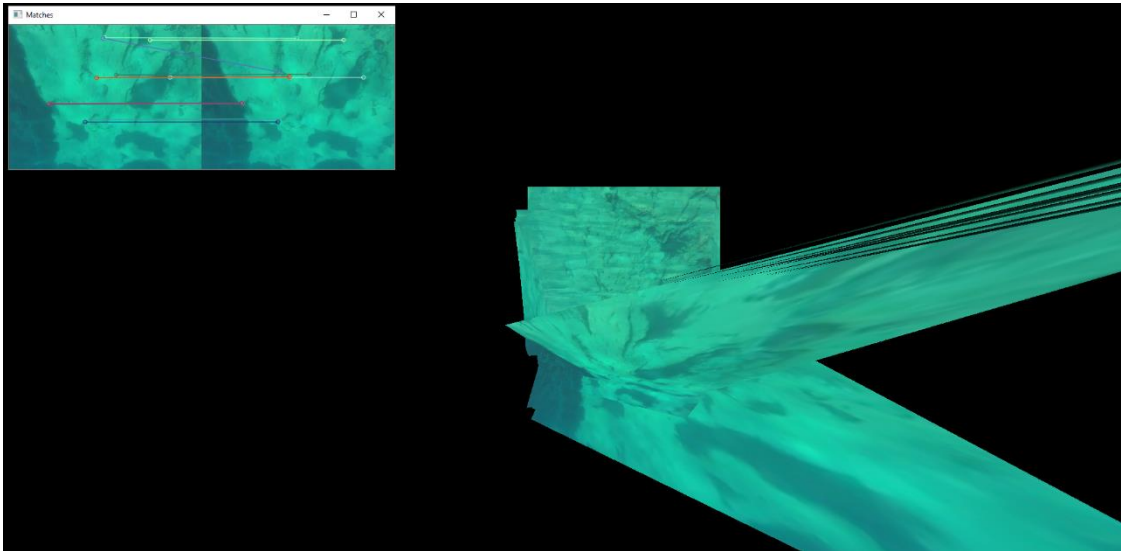


**Διάγραμμα 14:** Κακό τελικό προϊόν 2

Στα Διαγράμματα 13 και 14, είναι εμφανές ότι σε κάποιες περιπτώσεις το λογισμικό δεν απέδωσε κάποιο ευδιάκριτο τελικό αποτέλεσμα. Αυτό οφείλεται είτε λόγω της περιοχής του πυθμένα που καταγράφηκε στα καρέ είτε απότομων κινήσεων της κάμερας.

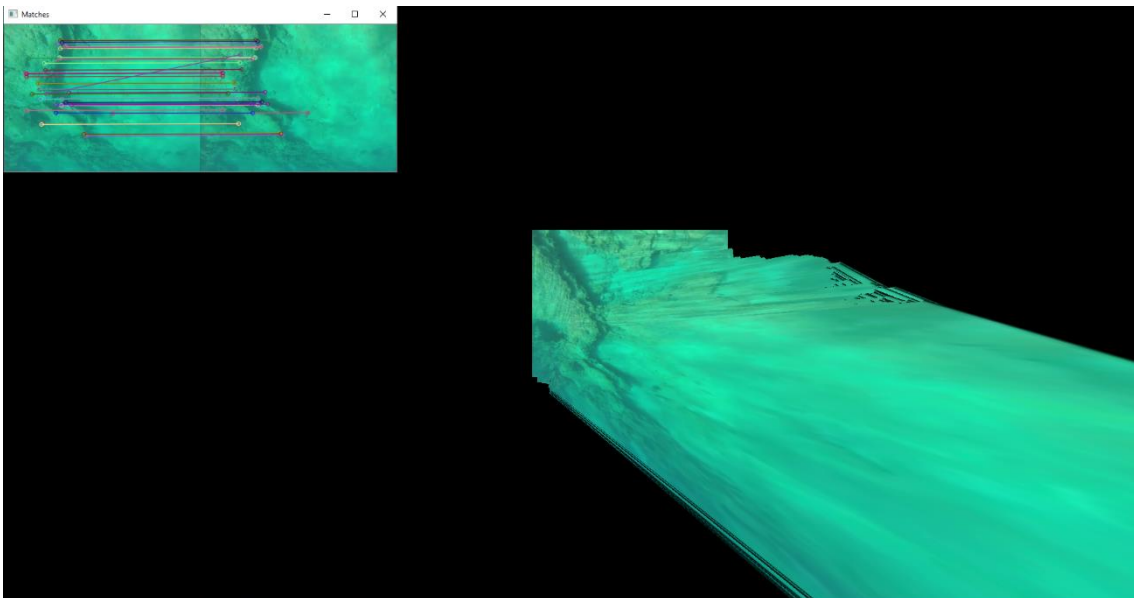
Πιο αναλυτικά αυτό που συμβαίνει είναι ότι δεν υπήρχαν καλά σημεία ή λεπτομερές σημεία που να μπορούν να αντιστοιχηθούν από τον αλγόριθμο ORB κατά την επεξεργασία που τύγχαναν τα εκάστοτε datasets από το λογισμικό Underwater Cyclops.

Φαίνεται ξεκάθαρα στο Διάγραμμα 14 και στο Διάγραμμα 15 παρακάτω.



**Διάγραμμα 15:** Κακή αντιστοιχία σημείων 1

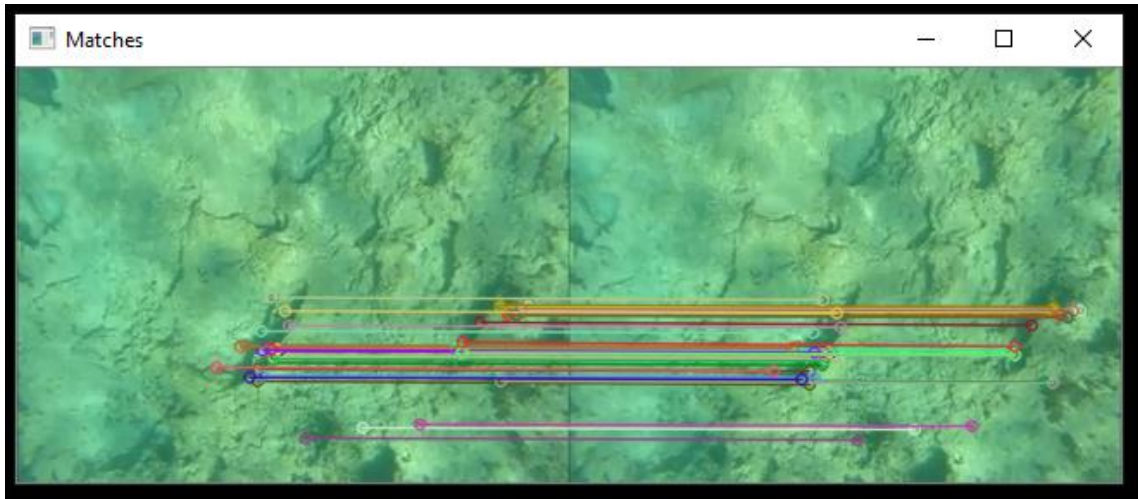
Στο Διάγραμμα 15, ο αριθμός των καλών αντιστοιχιών φθίνει, και επίσης βλέπουμε και διαγώνια αντιστοιχία που σημαίνει μία εντελώς λανθασμένη αντιστοιχία σημείων, καθώς είναι εντελώς διαφορετικά σημεία και παρατηρείται εύκολα με γυμνό μάτι.



**Διάγραμμα 16:** Κακή αντιστοιχία σημείων 2

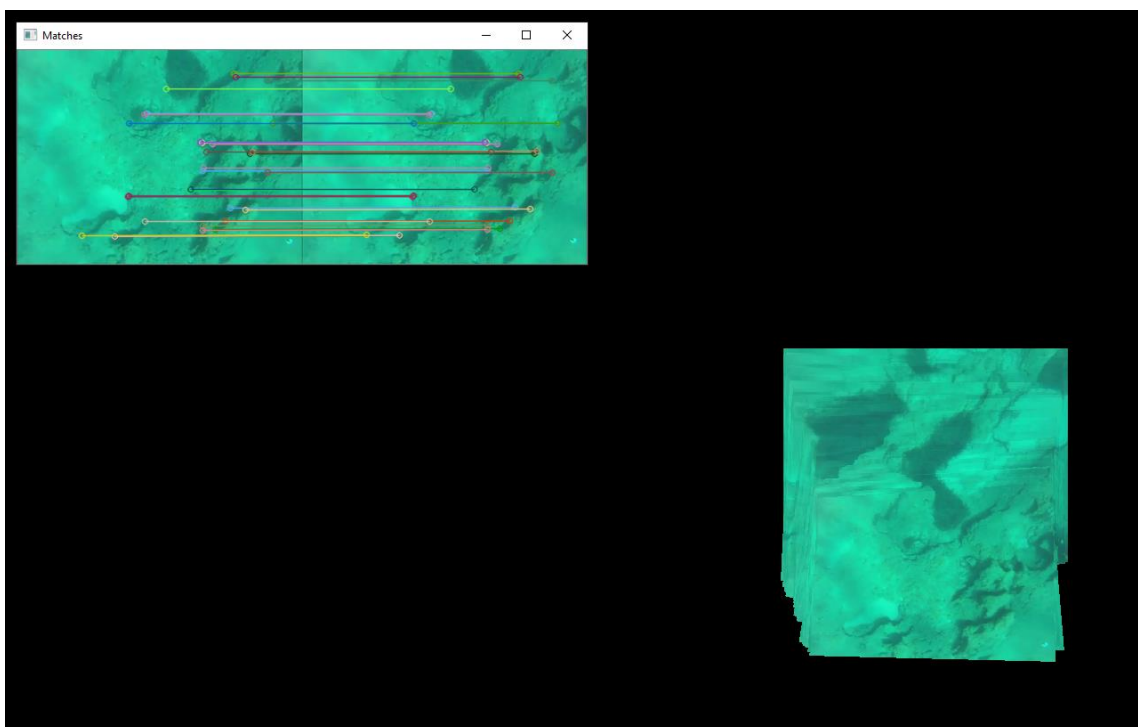
Στο Διάγραμμα 16 παρόλο που είναι αρκετές οι αντιστοιχίες, βλέπουμε επίσης διαγώνιες αντιστοιχίες, και επίσης να μην είναι απλωμένες στην ολότητα των καρτέ.

Επίσης παρατηρήθηκε ότι ακόμα και μια καλή αντιστοιχία σημείων μπορεί να μας δώσει ένα γεωμετρικά παραμορφωμένο αποτέλεσμα.



**Διάγραμμα 17:** Καλή αντιστοιχία που δίνει γεωμετρικά παραμορφωμένο χάρτη

Στο Διάγραμμα 17, παρόλο που φαίνεται μία πληθώρα καλών αντιστοιχιών σε ένα συγκεκριμένο ζεύγος καρτέ του Dataset 5, είναι όλες μαζεμένες σε μία συγκεκριμένη περιοχή των καρτέ, άρα αυτό σημαίνει ότι δεν θα ενωθούν σωστά τα δύο καρτέ.



**Διάγραμμα 18:** Καλή αντιστοιχία που δίνει μη γεωμετρικά παραμορφωμένο χάρτη

Αντίθετα στο Διάγραμμα 18, μπορεί να μην έχουμε τόσο πολλές αντιστοιχίες, αλλά είναι σωστά απλωμένες στα frames, σε σημεία που υπάρχουν λεπτομέρειες και βλέπουμε ότι ο αλγόριθμος αγνοεί τις σχεδόν λείες στο μάτι επιφάνειες και σχεδιάζει έναν αρκετά καλό χάρτη.

Αναφέρεται ότι στο παράθυρο matches που εμφανίζεται ότι εμφανίζονται μόνο οι θεωρητικά «καλύτερες» αντιστοιχίες, όπως κρίνει ο αλγόριθμος και όχι όλες οι αντιστοιχίες που αναγνωρίζει ο ORB.

Όπως εξηγήθηκε και σε προηγούμενο κεφάλαιο το λογισμικό Underwater Cyclops παράχθηκε για να βοηθά τους ερευνητές που ασχολούνται με τα ευρήματα στον υποβρύχιο κόσμο μέσω uROV. Το Underwater Cyclops, πιθανόν να μην μπορεί να βοηθήσει επακριβώς, στην απεικόνιση του πυθμένα για τον χειρισμό ενός uROV αλλά με κάποιες αλλαγές στη διάταξη του υλικού (hardware) θα προκύψει ένα λογισμικό που θα μπορεί να δημιουργεί ένα ορθοφωτομωσαικό των ενδιαφέροντων υποβρύχιων περιοχών για εκ των ύστερων μελέτη. Περισσότερο θα αναλυθεί στο Κεφάλαιο 5 «Συμπεράσματα και Μελλοντική Εργασία».

#### **4 Συζήτηση αποτελεσμάτων**

Στο Κεφάλαιο 4 της παρούσας διπλωματικής εργασίας θα συζητηθούν τα αποτελέσματα στην ολότητα τους και όχι η περιγραφή τους. Θα αναλυθούν οι λόγοι οι οποίοι οδήγησαν σε κακά αποτελέσματα όπως για παράδειγμα οι γεωμετρικές παραμορφώσεις που παρατηρούνται στα αποτελέσματα. Ακόμη θα αναφερθεί εκ νέου ο σκοπός της διατριβής, και θα παρατεθούν αποτελέσματα από άλλες ερευνητικές ομάδες, προς σύγκριση με αυτά που προέκυψαν από την παρούσα διατριβή. Σκοπός της διπλωματικής εργασίας, όπως αναφέρθηκε ήδη από την αρχή της, ήταν να διερευνηθεί το πρόβλημα της πλοήγησης κάτω από την επιφάνεια του νερού με τη χρήση uROV παράγοντας έναν «χάρτη» από επικαλυπτόμενες εικόνες (τα καρτέ που καταγράφονται από τη κάμερα του uROV).

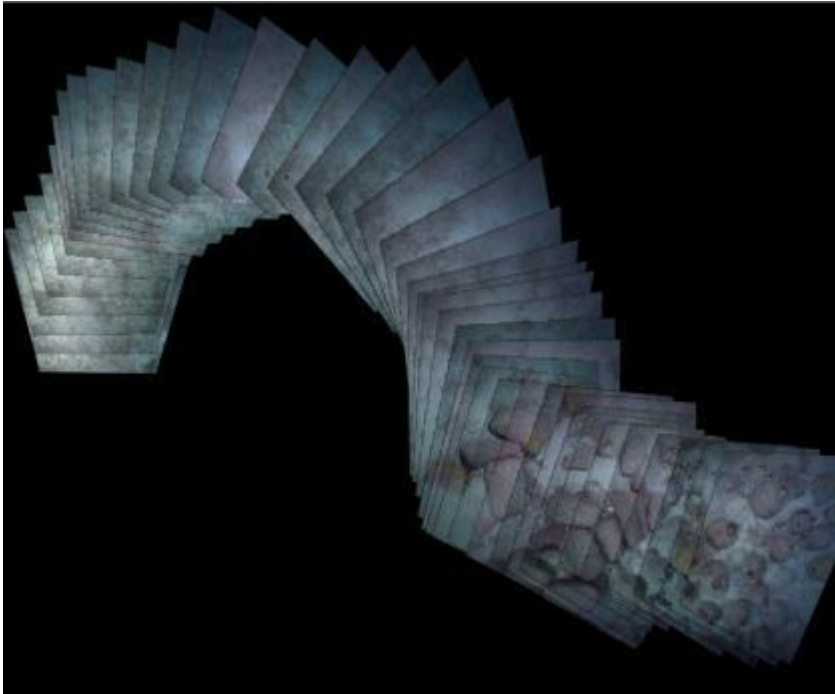
Στους παραγόμενους χάρτες από το λογισμικό που παράχθηκε στα πλαίσια αυτής της έρευνας παρατηρούνται σε όλους μηδενός εξαιρουμένου γεωμετρικές παραμορφώσεις. Ο λόγος είναι γιατί η πραγμάτωση του λογισμικού Underwater Cyclops έγινε με τη χρήση εκτός από αλγορίθμους όπως ο ORB ή ο Brute Force Matcher που αν θεωρηθούν

ως δευτερεύοντα συστατικά στοιχεία, το κύριο συστατικό στοιχείο του λογισμικού ήταν η διεκπεραίωση με την μέθοδο της Ομογραφίας (Homography). Η μέθοδος της Ομογραφίας καθίσταται εφικτή μόνο σε περιπτώσεις όπου η  $Z$  Ομογενές Συντεταγμένη ισούται συνέχεια με 0. Στη περίπτωση των σετ δεδομένων που χρησιμοποιήθηκαν η  $Z$  είναι συνεχώς μεταβαλλόμενη, αφού στην συγκεκριμένη περίπτωση όπου είναι μία κάμερα που κινείται καθώς κινείται ο δύτες που την χειρίζεται, η  $Z$  μεταβάλλεται.

Ακόμη, με ευκολία γίνεται κατανοητό, ότι υπάρχουν διαστρεβλώσεις στα αποτελέσματα και στα τελικά παράγωγα του λογισμικού, διότι η λήψη έγινε από δύτε, και εύκολα μπορεί να γίνει αντιληπτό ότι η κίνηση ενός ανθρώπου κάτω από την επιφάνεια του νερού, είναι πολύ διαφορετική από την κίνηση ενός uROV. Ακόμη, αν παρατηρηθούν λεπτομερώς τα datasets, παρατηρείται ένα συνεχές τρέμουλο ανά διαστήματα, στο οποίο οφείλονται ανωμαλίες στα αποτελέσματα.

Κατά το Κεφάλαιο 4 «Αποτελέσματα» συζητήθηκε και το κομμάτι του υπολογιστικού χρόνου, όπου με απλούς μαθηματικούς τύπους αναφέρθηκε ότι γίνεται επεξεργασία 500 καρέ σε 33 sec κατά προσέγγιση. Σε μία in situ περίπτωση όμως, όπου δεν θα υπάρχει η δυνατότητα ενός υπολογιστή με τόσο καλά τεχνικά χαρακτηριστικά όπως αυτός που χρησιμοποιήθηκε σε αυτή τη πρώτη πειραματική προσέγγιση. Ακόμη in situ τα δεδομένα δεν θα είναι άμεσα προσπελάσιμα από το λογισμικό, αλλά σε πραγματικό χρόνο θα μεταφέρονται μέσω Ethernet καλωδίου από το uROV. Αυτό συνεπάγεται στο ότι σε μία in situ διερεύνηση ενός οποιουδήποτε προβλήματος προκύψει αυτής της φύσης, θα έχουμε πιο αργούς υπολογιστικούς χρόνους.

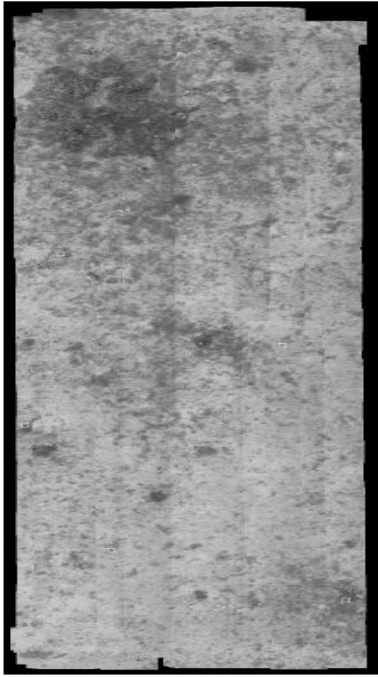
Στα Διαγράμματα 19-21 που παρατίθενται πιο κάτω φαίνονται τα αποτελέσματα από άλλες ερευνητικές προσεγγίσεις που δημοσιεύτηκαν σε ερευνητικά περιοδικά. Φαίνεται ότι έχουν διαφορές με τα αποτελέσματα που παράγονται από το λογισμικό Underwater Cyclops αλλά αν αναλογιστεί κανείς, ότι οι προσεγγίσεις έγιναν με διαφορετικές μεθόδους και με τον συνδυασμό διαφορετικών ειδών δεδομένων για να πετύχουν το καλύτερο δυνατόν αποτέλεσμα.



**Διάγραμμα 19:** Αποτέλεσμα από πηγή βιβλιογραφία υπ' αριθμόν 11



**Διάγραμμα 20:** Αποτέλεσμα από πηγή βιβλιογραφία υπ' αριθμόν 15



**Διάγραμμα 21:** Αποτέλεσμα από πηγή βιβλιογραφία υπ' αριθμόν 8

Ξεκινώντας από το αποτέλεσμα που φαίνεται στο Διάγραμμα 21 είναι εντελώς διαφορετικό από τα αποτελέσματα που παρατίθενται στα Διαγράμματα του Κεφαλαίου 3, καθώς χρησιμοποιήθηκε η μέθοδος των Binary Features.

Σε σύγκριση των παραγόμενων χαρτών που παρουσιάζονται στα Διαγράμματα 3-11 με τα αποτελέσματα των Διαγραμμάτων 20 και 21, φαίνεται ότι από το Underwater Cyclops παράγονται «χάρτες» πολύ κοντά σε αυτούς που παράγονται από τους «χάρτες» των Διαγραμμάτων που θεωρήθηκαν από τον φοιτητή ως σημεία αναφοράς για να φτάσει στην πραγμάτωση του λογισμικού που χρησιμοποιεί την μέθοδο της Ομογραφίας.

Τα συγκρινόμενα αποτελέσματα των Διαγραμμάτων 20 και 21, υπόκεινται επίσης σε γεωμετρικές παραμορφώσεις καθώς και σε αυτές τις περιπτώσεις, βλέπουμε τις διαστάσεις των καρέ, σε κάποια σημεία κατά την συνένωση καρέ σε καρέ, να αλλάζουν και να μην παραμένουν σταθερές. Προφανώς οι παραμορφώσεις σε αυτές τις περιπτώσεις, δεν προκύπτουν λόγω των αλγορίθμων που χρησιμοποιούνται αλλά λόγω δυσκολιών στις λήψεις που προκύπτουν από παραμέτρους του υποβρύχιου περιβάλλοντος.



Μέσα από αυτή τη σύντομη σύγκριση και τη γενική συζήτηση για τα αποτελέσματα του Underwater Cyclops προκύπτουν θετικά συμπεράσματα που θα αναλυθούν στο Κεφάλαιο 5 .

## **5 Συμπεράσματα και Μελλοντική Εργασία**

Στο παρόν κεφάλαιο θα αναλυθούν τα συμπεράσματα στα οποία καταλήγει ο φοιτητής μετά από την όλη διεκπεραίωση αυτής της διπλωματικής εργασίας αλλά και τα χαρακτηριστικά τα οποία αντιλαμβάνεται με την λεπτομερή χρήση του λογισμικού που παράχθηκε για το σκοπό της παρούσας διατριβής. Επίσης, στο κεφάλαιο θα αναφερθούν λεπτομερώς όλα τα επόμενα στάδια που θα πραγματοποιηθούν είτε για την βελτίωση του Underwater Cyclops αλλά και ιδέες που θα τεθούν προς υλοποίηση και προέκυψαν μέσα από αυτή τη διατριβή.

Αρχικό συμπέρασμα που προκύπτει είναι ότι το πρόβλημα που υπάρχει, δεν μπορεί να λυθεί με την χρήση της μεθόδου της Ομογραφίας (Homography) καθώς όπως διαπιστώνεται οι γεωμετρικές παραμορφώσεις προκύπτουν από την μεταβαλλόμενη ομογενές συντεταγμένη  $Z$  (Homogenous Coordinate  $Z$ ) ενώ η Ομογραφία προϋποθέτει ότι για να δώσει σωστά αποτελέσματα η  $Z$  πρέπει να είναι πάντα ίση με μηδέν. Άρα όπως ο δύτες κινείται με τη κάμερα κάτω από την επιφάνεια της θάλασσας για να καταγράψει τα σετ δεδομένων, έτσι θα κινείται και το uROV και η  $Z$  θα είναι μεταβαλλόμενη και σε αυτή τη περίπτωση. Ακόμη παρατηρήθηκε, ότι το λογισμικό δεν μπορεί να χρησιμοποιηθεί για σετ δεδομένων όπου τα καρέ έχουν απότομες αλλαγές γιατί αυτό θα προκαλέσει ανωμαλίες στα αποτελέσματα. Απότομες αλλαγές στα καρέ μπορούν να προκύψουν είτε γιατί το uROV θα πέσει σε ένα ισχυρό ρεύμα με αποτέλεσμα να το μετακινήσει απότομα είτε από λάθος χειρισμό του χειριστή που θα βρίσκεται στη στεριά. Άρα αυτό ισχύει και για in situ περιπτώσεις. Στις περιοχές της Κύπρου που θα τύγχαναν μελέτης με χρήση uROV και του Underwater Cyclops, σχεδόν ποτέ δεν παρουσιάζονται ισχυρά ρεύματα, άρα συμπεραίνεται ότι αν το uROV σταματούσε πάνω από τη περιοχή μελέτης, και με ένα σύστημα στη κάμερα, που θα χειρίζεται κάποιος την κλίση της και την οπτική γωνία της τότε το Underwater Cyclops θα μπορούσε να δημιουργήσει το ορθοφωτομωσαικό της περιοχής υπό μελέτη για περαιτέρω επεξεργασία στο εργαστήριο. Σίγουρα δεν θα μπορούσε να λειτουργήσει για να λύσει τα διάφορα προβλήματα πλοήγησης που ταλανίζουν τους ερευνητές.

Ακόμη συγκρίνοντας τα αποτελέσματα του Underwater Cyclops με αποτελέσματα άλλων ερευνητικών ομάδων, το λογισμικό που παράχθηκε στα πλαίσια της διατριβής βρίσκεται σε πολύ καλό δρόμο και με κάποιες βελτιώσεις θα φτάσει στο αρχικό επιθυμητό αποτέλεσμα και θα ανταγωνίζεται στα ίσια λογισμικά μεγάλων ερευνητικών εργαστηρίων.

Αφού ο ερευνητής κατέληξε και στα συμπεράσματα του μέσα από όλη αυτή την επίπονη διαδικασία, καταλήγει και στα όσα πρέπει να γίνουν στο μέλλον.

Εφόσον ήδη υπάρχει ένας κορμός του λογισμικού βασισμένος στην Ομογραφία, μπορεί σε συνδυασμό με κάποια άλλη μέθοδο που θα αντιλαμβάνεται και θα επεξεργάζεται την μεταβολή της ομογενούς συντεταγμένης  $Z$  όπου θα την λαμβάνει υπόψη κατά τη διάρκεια της παραγωγής του ορθοφωτομωσικού, όπως επίσης μπορεί να συνδυαστεί και η Ομογραφία με κάποια άλλη μέθοδο Pose Estimation (Εκτίμηση Θέσης). Ακόμη για καλύτερη υλοποίηση στο τομέα του υπολογιστικού χρόνου, θα ήταν καλό να εφαρμοστεί μόνιμα στην επόμενη έκδοση του λογισμικού Underwater Cyclops ένας Optimized FPS Limiter (Βελτιωμένος Περιοριστής FPS) αφού έτσι θα είναι λιγότερα τα επεξεργαζόμενα καρέ. Ακόμη το υπάρχον λογισμικό, μπορεί να συνδυαστεί με κάποια άλλη διάταξη υλικού (hardware) όπου να το καθιστά ικανό να φτιάχνει σωστά ορθοφωτομωσικά χωρίς γεωμετρικές παραμορφώσεις και ανωμαλίες. Μία πιθανή διάταξη υλικού θα πρέπει να γνωστοποιεί στο λογισμικό ότι η  $Z$  ομογενής συντεταγμένη θα παραμένει σταθερή στο μηδέν. Ακόμη κρίνεται πλέον ορθότερο από τον ερευνητή η επεξεργασία μέσω του αλγόριθμου ORB για εύρεση αντιστοιχιών μεταξύ ζευγών καρέ να γίνεται στο raw (ακατέργαστο) frame που δεν θα υπόκειται σε καμιάς μορφής downscaling ή upscaling (όπως π.χ. στη συγκεκριμένη περίπτωση που γίνεται downscale των datasets στα 240p) και μετά την επεξεργασία, η ένωση και επικάλυψη των καρέ μεταξύ τους, να γίνεται αφού υποστούν ένα downscale για να φτάσουν στο μέγεθος ενός tile (π.χ. 40x40). Αυτό θα βοηθήσει και σε θέματα υπολογιστικού χρόνου πολύ πιθανόν αλλά και στην καλύτερη επεξεργασία των ζευγών καρέ από τον αλγόριθμο ORB.

Ακόμη, για την αποφυγή γενικότερα των γεωμετρικών παραμορφώσεων αλλά και για να δοθεί η όσο το δυνατόν βέλτιστη λύση γίνεται, ο ερευνητής θα υλοποιήσει πηγαίο κώδικα για ένα νέο χαρακτηριστικό της νέας έκδοσης του Underwater Cyclops το οποίο κατά την επικάλυψη των καρέ μεταξύ τους, δεν θα σχεδιάζονται στο ορθοφωτομωσικό

οι περιοχές που δεν χαρακτηρίζονται από αντιστοιχίες που εντοπίζονται από τον αλγόριθμο ORB.

Τέλος, μετά την ανάπτυξη των βελτιώσεων και την παραγωγή ενός καλύτερου λογισμικού, που θα μπορεί να χειρίζεται καλύτερα τις περιπτώσεις υποβρύχιων πλοηγήσεων, ο ερευνητής έχει ως στόχο τη δημιουργία μιας πιο lite (απλής-ελαφριάς) έκδοσης για τα έξυπνα κινητά τηλέφωνα. Διαθέσιμη η εφαρμογή μόνο για τα κινητά τηλέφωνα τα οποία μπορούν να χρησιμοποιηθούν κάτω από το νερό για όσους έχουν χόμπι τις καταδύσεις και θα ήθελαν να αποτυπώνουν το ορθοφωτομωσικό της υποβρύχιας περιδιάβασης τους.

Συνοπτικά, μπορεί ολοκληρωτικά ο στόχος να μην έχει επιτευχθεί από τον ερευνητή αλλά σίγουρα προέκυψαν αρκετά ενθαρρυντικά στοιχεία και δεδομένα για τη συνέχιση της παρούσας ερευνητικής εργασίας.

## **ΕΠΙΛΟΓΟΣ**

Απώτερος στόχος και σκοπός της παρούσας διατριβής, μεταπτυχιακού επιπέδου, ήταν η χρήση φωτογραμμετρικών τεχνικών όπως η Ομογραφία και ο αλγόριθμος ORB στην παρούσα περίπτωση, υλοποιημένες σε πηγαίο κώδικα, σε συνδυασμό με τεχνολογίες Τεχνητής Νοημοσύνης για την παραγωγή ενός λογισμικού που θα καθορίζει κατά πόσο είναι εφικτό με την μέθοδο της Ομογραφίας να λυθεί το πρόβλημα της υποβρύχιας πλοήγησης με ένα uROV.

Συμπεράθηκε μέσα από την πραγμάτωση όλης αυτής της επίπονης έρευνας και μελέτης ότι η Ομογραφία για λόγους που εξηγήθηκαν εκτενώς στα προηγούμενα κεφάλαια δεν μπορεί να δώσει τη λύση σε τέτοιου είδους προβλήματα. Σε περίπτωση όμως που η Ομογραφία συνδυαστεί με κάποια άλλη μέθοδο ή κάποια άλλη διάταξη υλικού (θα επιφέρει όμως περιορισμούς) μπορεί να λύσει το πρόβλημα και να δίνει πάρα πολύ καλά αποτελέσματα, χωρίς παραμορφώσεις και ανωμαλίες.

Μέσα από τα αποτελέσματα που προκύπτουν μετά την χρήση του λογισμικού Underwater Cyclops και τα διαθέσιμα datasets από το εργαστήριο Photogrammetric-Vision Lab, διαπιστώνεται ότι η προσπάθεια που καταβλήθηκε, βάζει τον ερευνητή σε πολύ καλό δρόμο για την συνέχεια.

Εν κατακλείδι, η διατριβή με τίτλο «Underwater Cyclops - Real Time Underwater Image Mapping based on Homography» καθίσταται ως η κατάκτηση της απαραίτητης γνώσης και ο πρόδρομος για μια βελτιωμένη και αποδοτικότερη έκδοση του λογισμικού Underwater Cyclops, που θα λύσει τα χέρια στην επιστημονική κοινότητα της φωτογραμμετρίας.

## ΒΙΒΛΙΟΓΡΑΦΙΑ

1. Nocerino, E., Menna, F., Chemisky, B., and Drap, P. (2020). "REAL-TIME IMAGE MOSAICING: AN AID TO UNDERWATER IMAGE NAVIGATION". XXIV ISPRS Congress.
2. Nawaf, M. M., Hijazi, B., Merad, D. D., and Drap, P. (2016). "Guided Underwater Survey Using Semi-Global Visual Odometry." *15th International Conference on Computer Applications and Information Technology in the Maritime Industries, At Lecce, (May)*, 288–300.
3. Hou, G., Shao, Q., Zou, B., Dai, L., Zhang, Z., Mu, Z., Zhang, Y., and Zhai, J. (2019). "A novel underwater simultaneous localization and mapping online algorithm based on neural network." *ISPRS International Journal of Geo-Information*, 9(1).
4. Nawaf, M. M., Merad, D., Royer, J. P., Boï, J. M., Saccone, M., Ellefi, M. Ben, and Drap, P. (2018). "Fast visual odometry for a low-cost underwater embedded stereo system†." *Sensors (Switzerland)*, 18(7).
5. Ferreira, F., Veruggio, G., Caccia, M., Zereik, E., and Bruzzone, G. (2013). "A real-time mosaicking algorithm using binary features for ROVs." *2013 21st Mediterranean Conference on Control and Automation, MED 2013 - Conference Proceedings, IEEE*, 1267–1273.
6. Li, Y., Randall, C. J., Woesik, R. van, and Ribeiro, E. (2019). "Underwater video mosaicing using topology and superpixel-based pairwise stitching." *Expert Systems with Applications, Elsevier Ltd*, 119, 171–183.
7. Nocerino, E., Menna, F., Chemisky, B., and Drap, P. (2020). "3D SEQUENTIAL IMAGE MOSAICING for UNDERWATER NAVIGATION and MAPPING." *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, 43(B2), 991–998.
8. Karami, E., Prasad, S., and Shehata, M. (2017). "Image matching using SIFT, SURF, BRIEF and ORB: Performance comparison for distorted images." *arXiv*.
9. Clemons, J. (2007). "Sift : Scale Invariant Feature Transform." *Transform*, 2, 91–110.
10. Bay, H., Ess, A., Tuytelaars, T., and Van Gool, L. (2008). "Speeded-Up Robust Features (SURF)." *Computer Vision and Image Understanding*, 110(3), 346–359.
11. Pang, Y., Li, W., Yuan, Y., and Pan, J. (2012). "Fully affine invariant SURF for image matching." *Neurocomputing, Elsevier*, 85, 6–10.
12. Daniilidis, K., Maragos, P., and Paragios, N. (2010). *Computer vision - ECCV 2010*. Springer, Berlin.
13. Rublee, E., Rabaud, V., Konolige, K., and Bradski, G. (2011). "ORB: An efficient alternative to SIFT or SURF." *Proceedings of the IEEE International Conference on Computer Vision, IEEE*, 2564–2571.
14. Viswanathan, D. G. (2011). "Features from Accelerated Segment Test (FAST) Deepak Geetha Viswanathan 1."
15. Noble, F. K. (n.d.). "Comparison of OpenCV ' s Feature Detectors and Feature Matchers." IEEE.
16. Gil, A., Mozos, O. M., Ballesta, M., and Reinoso, O. (2010). "A comparative evaluation of interest point detectors and local descriptors for visual SLAM." *Machine Vision and Applications*, 21(6), 905–920.

17. Li, X., Liu, Y., Wang, Y., and Yan, D. (2005). "Computing homography with RANSAC algorithm: a novel method of registration". *Electronic Imaging and Multimedia Technology IV*.
18. Li, H., Qin, J., Xiang, X., Pan, L., Ma, W., and Xiong, N. (2018). "An Efficient Image Matching Algorithm Based on Adaptive Threshold and RANSAC". *IEEE Access*, 6, 66963-66971.
19. Liu, X., Cheng, Z., and Yi, F. (2016). "A road detection technology based on reverse perspective transformation." *Proceedings - 2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2016*, 1, 126–129.
20. Bradski, G., and Kaehler, A. (2016). *Learning OpenCV*. O'Reilly Media, Incorporated, Cambridge.
21. Bressert, E. (2012). *SciPy and NumPy*. O'Reilly Media.
22. Grimmett, R. (n.d.). *Raspberry Pi Robotics Projects - Second Edition*. PACKT.
23. von Lukas, U. F., Quarles, J., Kaklis, P., and Dolereit, T. (2015). "Underwater Mixed Environments." *Virtual Realities: International Dagstuhl Seminar, Dagstuhl Castle, Germany, June 9-14, 2013, Revised Selected Papers*, G. Brunnett, S. Coquillart, R. van Liere, G. Welch, and L. Váša, eds., Springer International Publishing, Cham, 56–76.
24. Li, H., Qin, J., Xiang, X., Pan, L., Ma, W., and Xiong, N. N. (2018). "An efficient image matching algorithm based on adaptive threshold and RANSAC." *IEEE Access*, IEEE, 6, 66963–66971.
25. "OpenCV: cv::ORB Class Reference." (2021). Docs.opencv.org, <[https://docs.opencv.org/3.4/db/d95/classcv\\_1\\_1ORB.html](https://docs.opencv.org/3.4/db/d95/classcv_1_1ORB.html)> (Jan. 24, 2021).
26. Costa, M., and Bonetti, L. (2018). "Geometrical distortions in geographical cognitive maps." *Journal of Environmental Psychology*, Elsevier Ltd, 55, 53–69.