

Ψηφιακή Επεξεργασία Εικόνας

Εργασία 3

Local Descriptor.....	2
function myLocalDescriptor.....	3
Δοκιμές.....	4
function d = myLocalDescriptorUpgrde.....	6
Δοκιμές.....	6
Harris Corner Detector.....	8
function isCorner.....	8
function myDetectHarrisFeatures.....	10
Matching descriptors.....	15
function descriptorMatching.....	15
RANSAC.....	16
function myRANSAC.....	16
Συνένωση.....	18
function myStitch.....	18

Γκατζή Κωνσταντίνα
AEM: 10037
mail: gikonstan@ece.auth.gr

Local Descriptor

Ο περιγραφέας χρησιμοποιείται για να δίνει πληροφορίες για ένα συγκεκριμένο pixel της εικόνας. Η εικόνα που χρησιμοποιείται για να δοκιμαστεί είναι αυτή.

Image 1: Im1 used to test descriptor



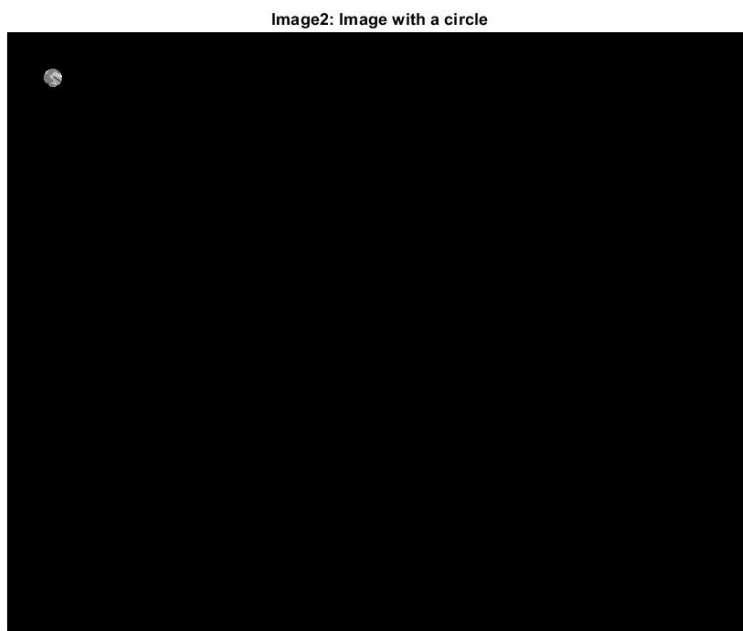
function myLocalDescriptor

Η συνάρτηση αυτή υλοποιεί τον περιγραφέα. Δέχεται σαν είσοδο την εικόνα I και το συγκεκριμένο pixel p για το οποίο θέλουμε να εξαγάγουμε πληροφορίες.

Η είσοδος ρ_{hom} είναι η αρχική ακτίνα του κύκλου που χρησιμοποιούμε σαν μάσκα γύρω από το pixel. Η είσοδος ρ_{hstep} είναι το βήμα με το οποίο αυξάνεται η ακτίνα του κύκλου, ενώ η ρ_{hoM} είναι η μέγιστη ακτίνα που θα φτάσει. Η είσοδος N χρησιμοποιείται για να χωριστεί ο κύκλος σε κομμάτια και να πάρουμε τιμές από συγκεκριμένα σημεία από αυτά τα κομμάτια. Η έξοδος d είναι ένα διάνυσμα μεγέθους όσο οι κύκλοι και περιέχει τις πληροφορίες για το pixel.

Στη συνάρτηση πραγματοποιείται μια επαναληπτική διαδικασία για την τιμή της ακτίνας, με αρχική, τελική τιμή και βήμα όσα θέτουμε στην είσοδο. Σε κάθε επανάληψη δημιουργείται ένα πλαίσιο με τις διαστάσεις της εικόνας και μια μάσκα με διαφορετική ακτίνα ανάλογα με την επανάληψη, χρησιμοποιώντας την εξίσωση του κύκλου γύρω από το pixel.

Θέτω μια εικόνα `imageWithCircle` ίση με την αρχική και τις τιμές της εκτός των θέσεων του κύκλου μηδενικές, δηλαδή μαύρο χρώμα.



Έπειτα εφόσον έχουμε τον κύκλο χρειάζεται να τον χωρίσουμε σε N μέρη. Αυτό υλοποιείται με τη χρήση πολικών συντεταγμένων. Το σημείο σε κάθε κομμάτι θα έχει συντεταγμένες ίσες με αυτές του pixel συν την ακτίνα επί το συνημίτονο του $2j\pi/N$ για τον άξονα x και το αντίστοιχο ημίτονο για τον άξονα y , όπου το j ξεκινάει από την τιμή 0 και καταλήγει στην $N-1$ έτσι ώστε να συμπληρώσει κύκλο. Σε περίπτωση που αυτό το σημείο που προκύπτει είναι εκτός των ορίων της εικόνας, ενεργοποιείται η μεταβλητή `empty=1`.

Για κάθε κύκλο το διάνυσμα d είναι ο μέσος όρος των σημείων του κύκλου, ενώ αν η μεταβλητή `empty` είναι ίση με 1, το διάνυσμα d μένει άδειο.

Δοκιμές

Μετά από περιστροφή της εικόνας κατά γωνία 3 μοιρών δοκιμάστηκε ο περιγραφέας με αρχική ακτίνα 5, τελική ακτίνα 20, βήμα 1, αριθμό τμημάτων κύκλου 8 , στο pixel [100 100]. Προέκυψε το διάνυσμα d

1	2	3	4	5	6	7	8
0.2404	0.4664	0.244	0.2442	0.4741	0.4815	0.2453	0.4773

9	10	11	12	13	14	15	16
0.4419	0.2107	0.2098	0.4841	0.4465	0.2122	0.4446	0.3562

Μετά την περιστροφή της εικόνας κατά 88 μοίρες προκύπτει διάνυσμα d

1	2	3	4	5	6	7	8
0.1445	0.4621	0.1978	0.1822	0.4215	0.3961	0.1362	0.2784

9	10	11	12	13	14	15	16
0.254	0.0895	0.1286	0.4721	0.493	0.1542	0.472	0.4999

Για το pixel [200 200] στην αρχική εικόνα προκύπτει διάνυσμα d

1	2	3	4	5	6	7	8
0.1712	0.2991	0.1609	0.1575	0.314	0.3132	0.1965	0.3562

9	10	11	12	13	14	15	16
0.3185	0.1483	0.196	0.3667	0.3664	0.2266	0.3523	0.3494

Για το pixel [202 202] προκύπτει

1	2	3	4	5	6	7	8
0.1995	0.3753	0.19	0.1732	0.3585	0.3547	0.1814	0.3046

9	10	11	12	13	14	15	16
0.2959	0.1597	0.1817	0.3398	0.3289	0.2278	0.3538	0.3622

Παρατηρείται ότι σε διπλανά pixels οι τιμές του περιγραφέα είναι παρόμοιες.

function d = myLocalDescriptorUpgrde

Η συνάρτηση αυτή με ίδιες εισόδους και εξόδους με την προηγούμενη, είναι ο περιγραφέας δικής μου έμπνευσης. Το διαφορετικό είναι ότι σε κάθε κύκλο αντί να πάρουμε τιμές από τα σημεία της περιμέτρου του κύκλου, παίρνουμε τιμές από όλα τα σημεία της ακτίνας και υπολογίζουμε τον μέσο όρο έτσι ώστε να έχουμε περισσότερη πληροφορία.

Στην περίπτωση που η ακτίνα είναι υπερβολικά μικρή παίρνουμε τις τιμές του pixel μόνο.

Δοκιμές

Η δοκιμή για το Pixel [100 100] με περιστροφή της εικόνας κατά 3 μοίρες δίνει διάνυσμα d

1	2	3	4	5	6	7	8
0.4698	0.47251	0.47429	0.4761	0.47751	0.4779	0.4779	0.4783

9	10	11	12	13	14	15	16
0.4783	0.4767	0.4757	0.47574	0.4743	0.47269	0.4725	0.47127

Για περιστροφή κατά 88 μοίρες

1	2	3	4	5	6	7	8
0.1682	0.1778	0.195	0.21	0.2137	0.224	0.2317	0.232

9	10	11	12	13	14	15	16
0.2345	0.2369	0.2419	0.2438	0.2462	0.2464	0.2463	0.24812

Για το Pixel [200 200]

1	2	3	4	5	6	7	8
0.64386	0.6402	0.628	0.6186	0.6174	0.612	0.6081	0.6103

9	10	11	12	13	14	15	16
0.6048	0.6015	0.6004	0.5998	0.6001	0.5958	0.5962	0.5934

Για το pixel [202 202]

1	2	3	4	5	6	7	8
0.4244	0.42931	0.4309	0.4322	0.4342	0.434	0.4319	0.4303

9	10	11	12	13	14	15	16
0.429	0.4269	0.4252	0.4256	0.4228	0.4214	0.4243	0.4251

Στον δικό μου περιγραφέα όλα τα στοιχεία του διανύσματος έχουν όμοια αποτελέσματα διότι παίρνω όλα τα στοιχεία της ακτίνας των σημείων. Στα διπλανά pixels ισχύει το ίδιο συμπέρασμα με τον άλλο περιγραφέα, είναι δηλαδή οι τιμές κοντινές.

Harris Corner Detector

Η μέθοδος αυτή χρησιμοποιείται για να εντοπιστούν τα σημεία ενδιαφέροντος στην εικόνα. Χρησιμοποιώντας τις παρακάτω συναρτήσεις μελετώνται τα pixels όλης της εικόνας. Χρησιμοποιεί επαναληπτικές διαδικασίες οπότε αυτή η ρουτίνα είναι αργή.

function isCorner

Η συνάρτηση αυτή δέχεται σαν είσοδο την εικόνα, τις συντεταγμένες από ένα συγκεκριμένο pixel, την παράμετρο k και το κατώφλι. Έξοδο δίνει την binary μεταβλητή που δηλώνει αν το pixel της εισόδου δείχνει γωνία ή όχι. Εφαρμόζει τον αλγόριθμο Harris Corner Detector. Χρησιμοποιεί την μάσκα με τιμές που αποσβαίνουν όσο απομακρύνεται από το κέντρο, όπως υποδείχθηκε στη εκφώνηση.

Συγκεκριμένα:

$w =$

0.8825	0.9394	0.8825
0.9394	1.0000	0.9394
0.8825	0.9394	0.8825

Στη συνέχεια υπολογίζονται οι μερικές παράγωγοι της εικόνας .

Η μερική παράγωγος ως προς τον άξονα x υπολογίζεται με συνέλιξη με τη μάσκα

1	0	-1
1	0	-1
1	0	-1

Η μερική παράγωγος ως προς τον άξονα y υπολογίζεται με συνέλιξη με τη μάσκα

1	1	1
0	0	0
-1	-1	-1

Για να βρούμε τον πίνακα M, χρειαζόμαστε το τετράγωνο της μερικής παραγώγου ως προς x, το τετράγωνο της μερικής παραγώγου ως προς y και το γινόμενο τους. Αφού η μάσκα w είναι 3x3 χρειαζόμαστε από τις εικόνες αυτές που προέκυψαν την 3x3 περιοχή γύρω από το επιλεγμένο pixel.

Εφαρμόζουμε συνέλιξη της μάσκας με αυτές τις περιοχές των εικόνων και αθροίζουμε τα στοιχεία των αποτελεσμάτων. Ο πίνακας M αποτελείται από αυτά τα στοιχεία όπως ακριβώς δηλώνει η εκφώνηση.

Τέλος υπολογίζουμε την τιμή $R = \det(M) - k \cdot \text{trace}(M)^2$, η οποία θα αποφασίσει αν το σημείο περιέχει γωνία ή όχι. Αν η τιμή είναι μεγαλύτερη από το κατώφλι μόνο θεωρείται γωνία και έτσι δίνεται τιμή στην binary μεταβλητή του output.

function myDetectHarrisFeatures

Η συνάρτηση αυτή εφαρμόζει τα αποτελέσματα της `isCorner` σε όλη την εικόνα. Δέχεται σαν είσοδο μόνο την εικόνα και η έξοδος της είναι το διάνυσμα με τις συντεταγμένες όλων των γωνιών και ο αντίστοιχος περιγραφέας της.

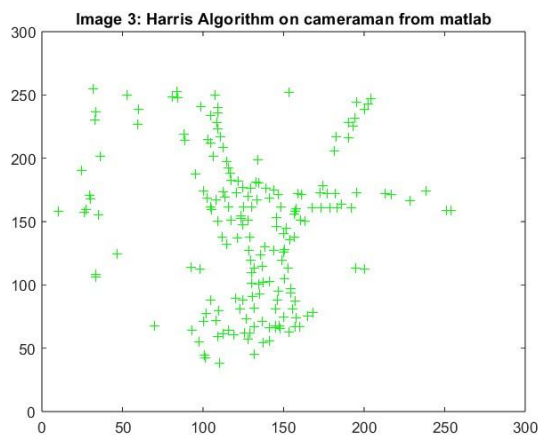
Αρχικά στη συνάρτηση αντιμετωπίζεται το πρόβλημα που προκύπτει αν η εικόνα έχει πολύ μεγάλες διαστάσεις. Εφόσον αποδείχθηκε ότι σε διπλανά pixels οι περιγραφείς έχουν παρόμοιες τιμές, οι διαστάσεις της εικόνας μπορεί να μειωθεί χωρίς να προκύψει πρόβλημα από τις πληροφορίες. Η διαφορά στην εικόνα μετά την μείωση της διάστασης δεν είναι εμφανής, αλλά η ταχύτητα του αλγορίθμου είναι βελτιωμένη.

Σαρώνεται όλη η εικόνα με βήμα 3, διότι η μάσκα είναι 3x3 και για κάθε pixel εφαρμόζεται η συνάρτηση `isCorner` και αποφασίζεται αν περιέχει γωνία ή όχι. Στην περίπτωση που περιέχει αποθηκεύονται οι συντεταγμένες της.

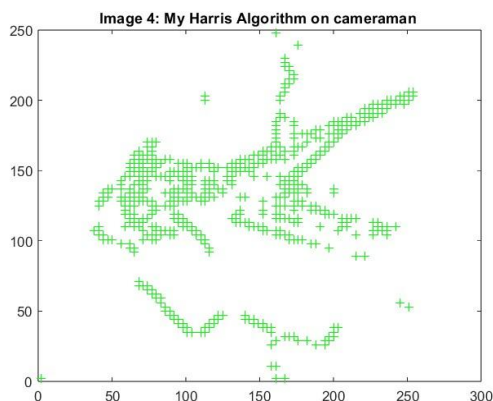
Αφού αποθηκεύσει τις συντεταγμένες της γωνίας εφαρμόζεται η συνάρτηση για τη εύρεση του περιγραφέα στο συγκεκριμένο pixel. Στην περίπτωση που ο περιγραφέας προκύψει κενός, ορίζεται μηδενικό διάνυσμα για να καταφέρουν να αποθηκευτούν όλα στη σειρά στον πίνακα της εξόδου.

Η συνάρτηση δοκιμάστηκε αρχικά στην εικόνα `cameraman` και πραγματοποιήθηκε σύγκριση με την έτοιμη συνάρτηση της `matlab`.

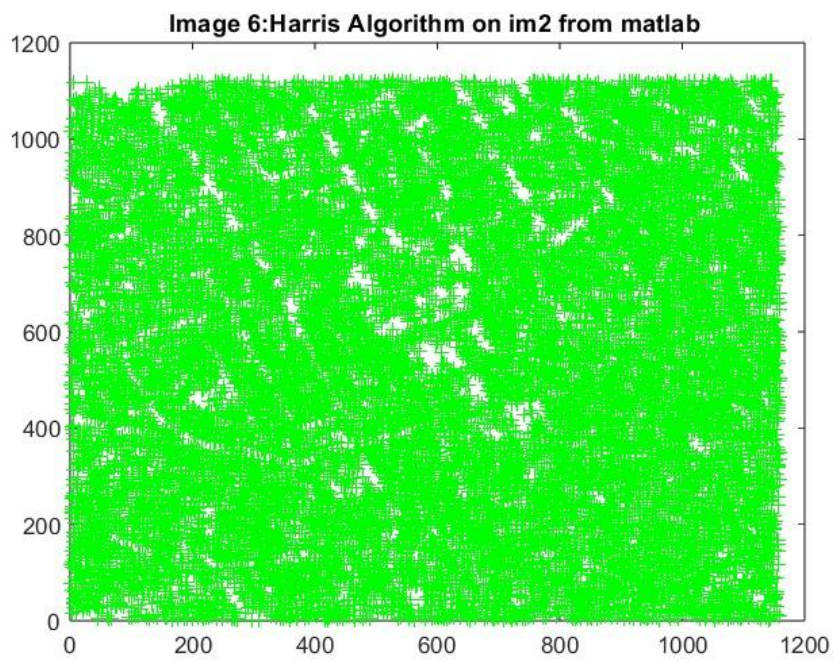
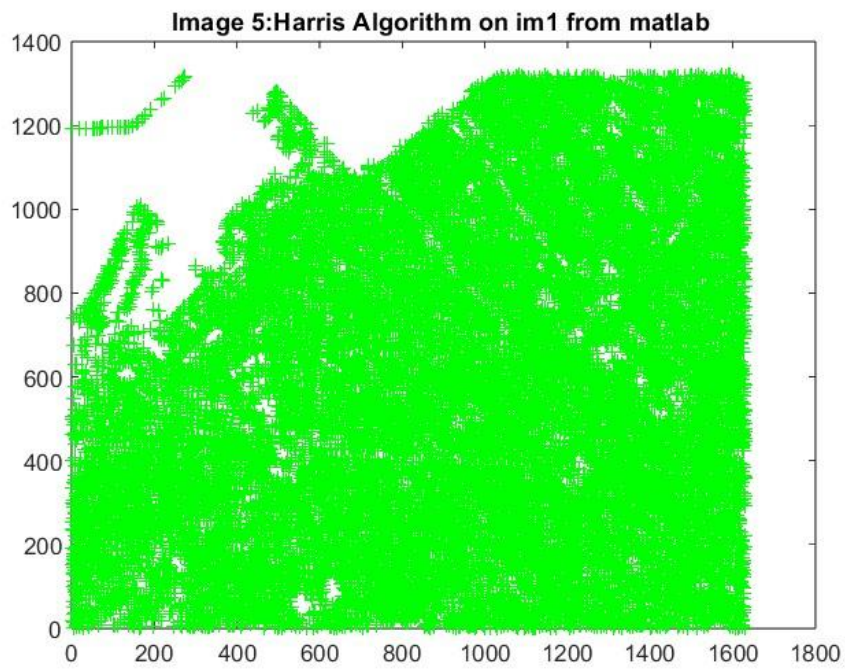
Το αποτέλεσμα της έτοιμης συνάρτησης είναι

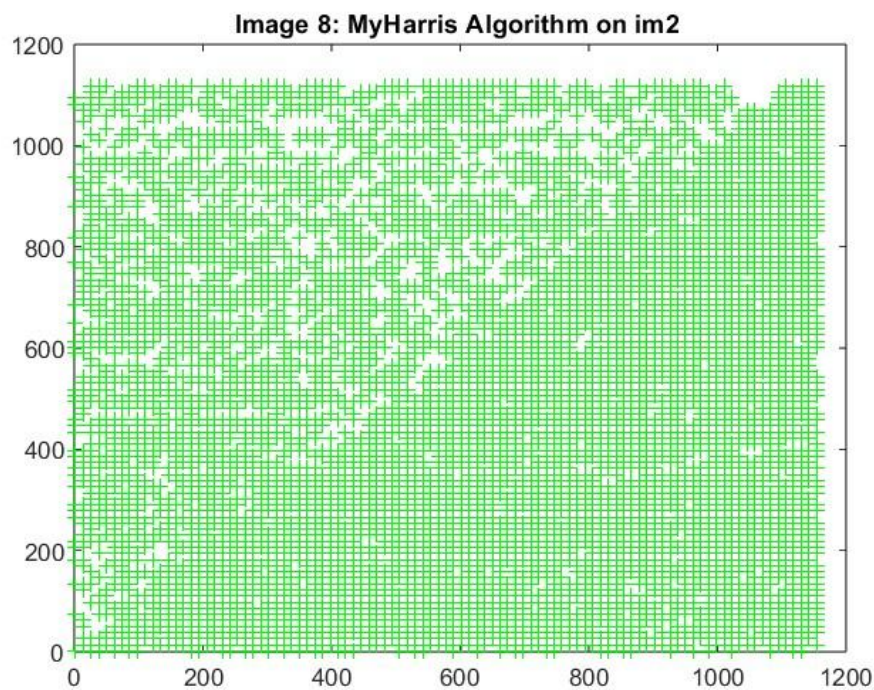
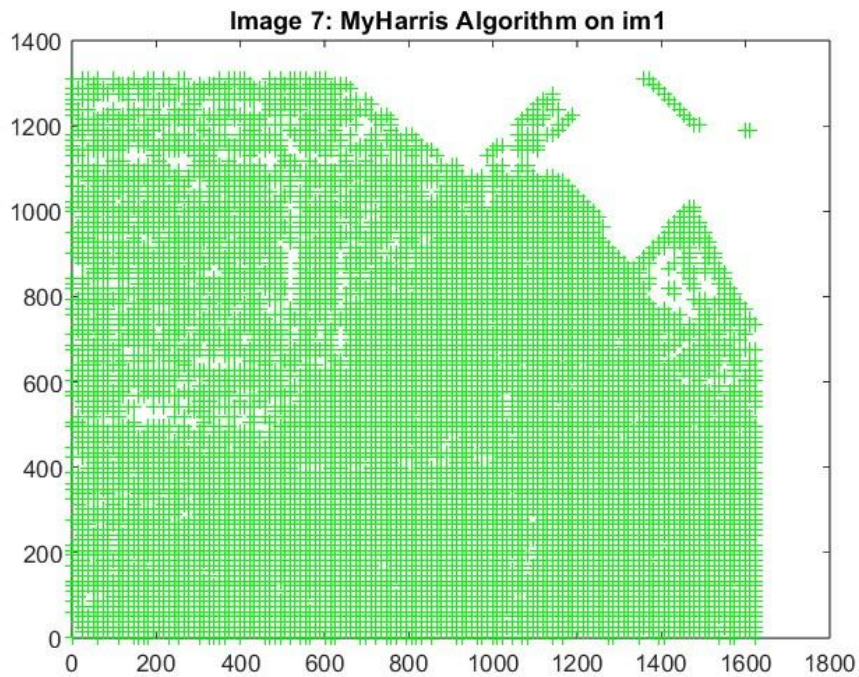


Το αποτέλεσμα της δικής μου συνάρτησης είναι



Αντίστοιχα το αποτέλεσμα για τις εικόνες im1 και Im2





Φαίνεται ότι τα αποτελέσματα είναι πανομοιότυπα, με μοναδικό πρόβλημα μια περιστροφή που διορθώνεται σε επόμενη συνάρτηση.
Παρακάτω δοκιμάζεται ο αλγόριθμος στις εικόνες με το δάσος.

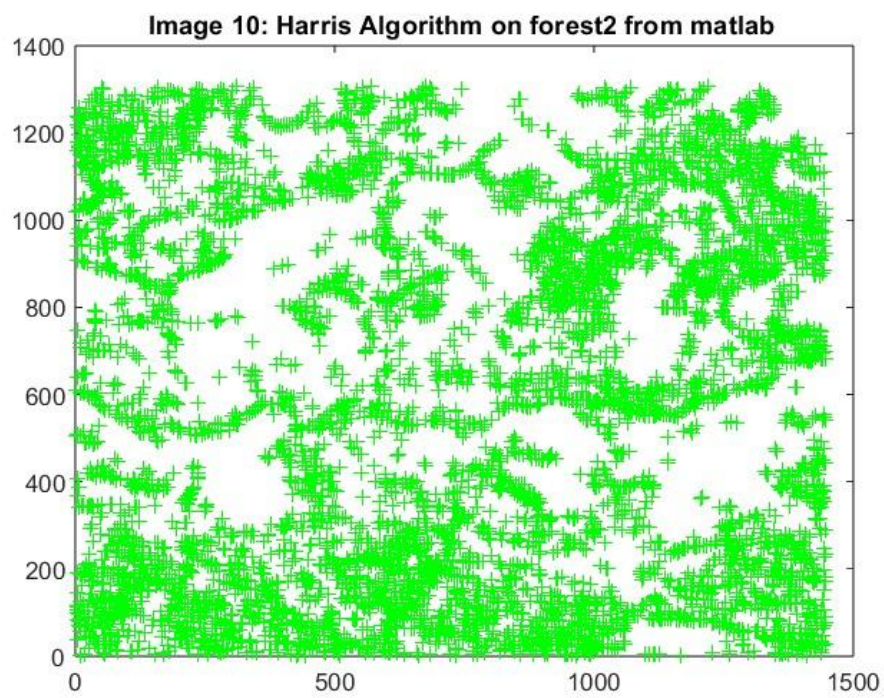
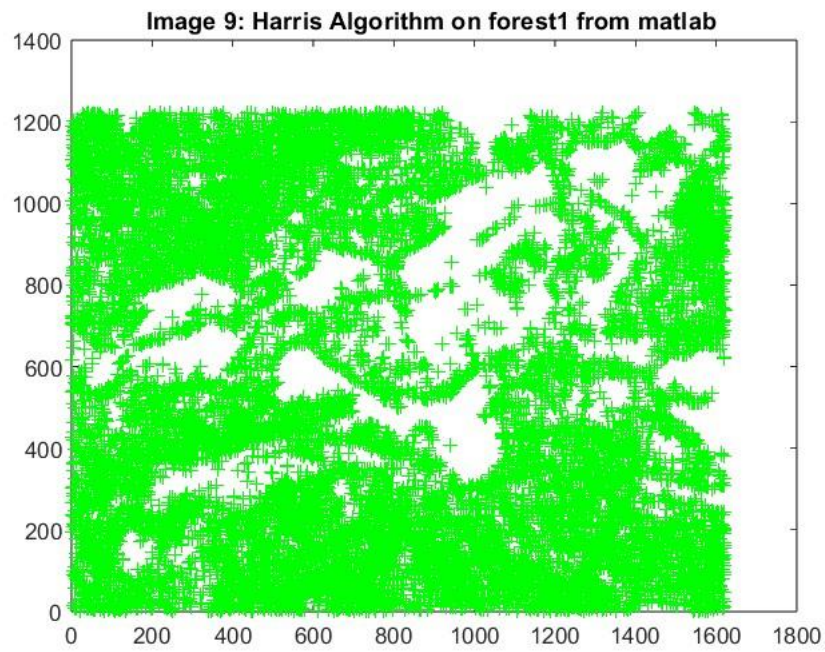


Image 11: My Harris Algorithm on forest1

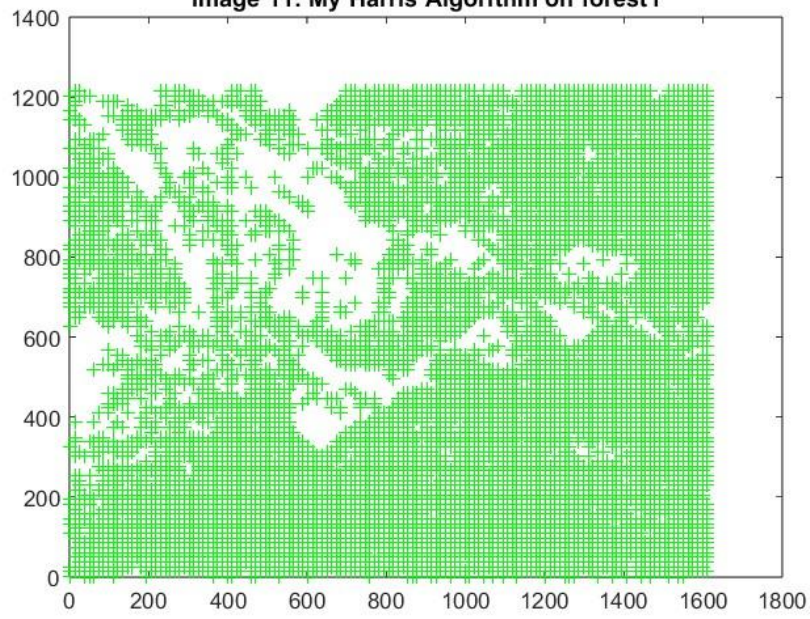
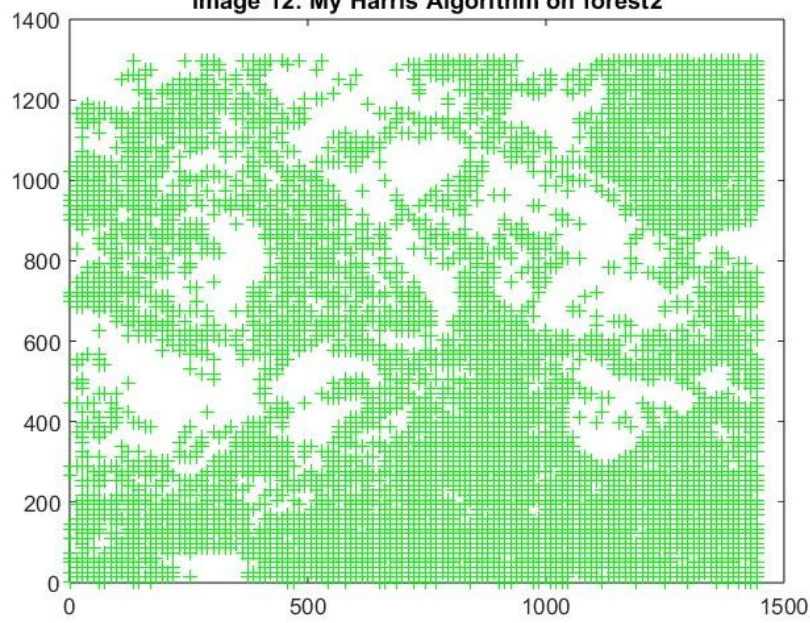


Image 12: My Harris Algorithm on forest2



Matching descriptors

Η μέθοδος αυτή μελετάει όλα τα πιθανά ζεύγη σημείων και αποφασίζει ποια ταιριάζουν.

function descriptorMatching

Η συνάρτηση αυτή χρησιμοποιείται για να μελετήσει τα πιθανά ζεύγη ανάλογα με την ευκλείδεια απόστασή τους. Σαν εισόδους δέχεται τις λίστες των συντεταγμένων από τα pixels που περιέχουν γωνίες από τις 2 εικόνες, τις λίστες με τα διανύσματα των περιγραφών για τα συγκεκριμένα pixels με την ίδια σειρά, και το κατώφλι για την απόσταση. Έξοδος είναι η λίστα `matchingPoints` που περιέχει δείκτες στις λίστες των σημείων για τα ζεύγη που θεωρεί ότι ταιριάζουν. Η πρώτη στήλη περιέχει δείκτες στη λίστα σημείων από την πρώτη εικόνα, ενώ η δεύτερη στήλη για την δεύτερη εικόνα.

Αρχικά αφαιρεί τα σημεία που περιέχουν μηδενικούς περιγραφείς και τους περιγραφείς αυτούς. Υπολογίζει σε μια μεταβλητή το άθροισμα από όλα τα διανύσματα περιγραφών. Έπειτα βρίσκει τις συντεταγμένες από τα μηδενικά και τα αφαιρεί, όπως και τα αντίστοιχα σημεία με τις συντεταγμένες αυτές.

Υπολογίζεται ο πίνακας με τις ευκλείδειες αποστάσεις όλων των πιθανών ζευγών. Βρίσκονται οι συντεταγμένες των σημείων που έχουν απόσταση κάτω από το επιτρεπτό κατώφλι. Η συντεταγμένες στον άξονα x είναι οι δείκτες στα στοιχεία της πρώτης λίστας, ενώ στον άξονα y της δεύτερης λίστας. Αποθηκεύονται στην μεταβλητή εξόδου.

RANSAC

function myRANSAC

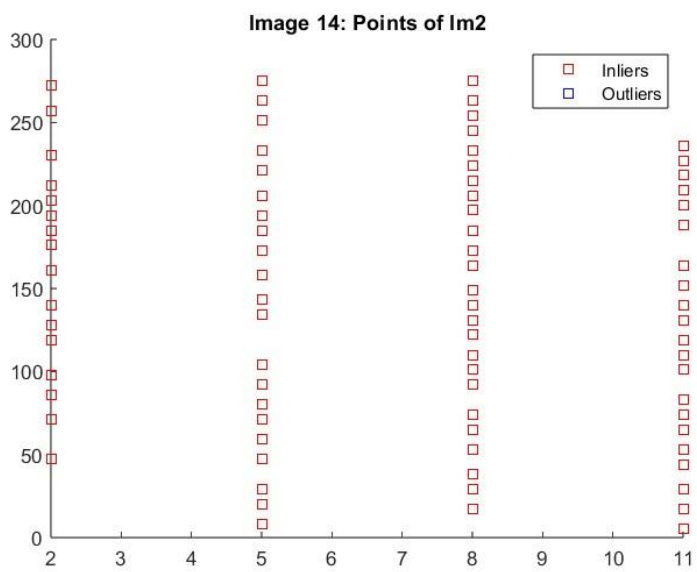
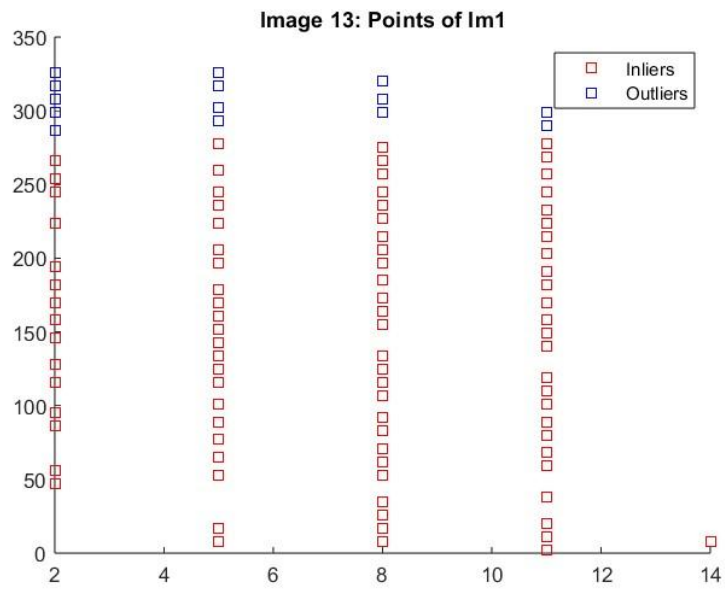
Η συνάρτηση αυτή δέχεται σαν εισόδους την λίστα `matchingPoints` από την προηγούμενη συνάρτηση, το `r` κατώφλι απόστασης, το `N` που είναι ο αριθμός φορών που πρέπει να εφαρμοστεί ο αλγόριθμος, και τη λίστα με τις συντεταγμένες των σημείων που περιέχουν γωνίες στις 2 εικόνες. Για εξόδους δίνει τον `H` μετασχηματισμό από το επιλεγμένο ζευγάρι και τις λίστες `inliers` και `outliers` που περιέχουν δείκτες στη λίστα `matchingPoints` και αναγνωρίζουν ποια σημεία θεωρούνται ότι αντιστοιχήθηκαν καλά και ποια όχι.

Η μεταβλητή `best` χρησιμοποιείται για να αποθηκεύσει τον δείκτη του καλύτερου ζεύγους και η `ebest` την τιμή. Χρησιμοποιείται επαναληπτική διαδικασία με βάση τις φορές που έχει οριστεί ότι πρέπει να επαναλαμβάνεται ο αλγόριθμος.

Αρχικά επιλέγεται τυχαία ένα ζεύγος από τη λίστα `matchingPoints` και χρησιμοποιώντας τη λίστα αυτή που περιέχει δείκτες στις λίστες των σημείων των 2 εικόνων βρίσκονται οι συντεταγμένες των σημείων. Υπολογίζεται στην μεταβλητή `dist` τη ευκλείδεια απόσταση των 2 ζευγών. Έπειτα υπολογίζεται η γωνία ανάμεσα τους και αποθηκεύεται ο μετασχηματισμός `H` στον πίνακα `h` για κάθε επανάληψη.

Χρησιμοποιείται άλλη μια επαναληπτική διαδικασία για όλα τα υπόλοιπα σημεία της `matchingPoints`. Εφαρμόζεται ο μετασχηματισμός σε κάθε σημείο και υπολογίζεται η ευκλείδεια απόσταση τους. Στη συνέχεια αθροίζονται όλες οι αποστάσεις και ελέγχεται το αν η απόσταση του κάθε σημείου είναι μικρότερη από κατώφλι. Σε περίπτωση που είναι μικρότερη η απόσταση αποθηκεύεται ο δείκτης του ζεύγους της `matchingPoints` στην μεταβλητή `in` για τα `inlier` σημεία, ενώ μηδενίζεται η μεταβλητή `out` για τα `outlier` σημεία, αλλιώς στην μεταβλητή `out` και μηδενίζεται η μεταβλητή `in`.

Αναγνωρίζεται σε ποια επανάληψη βρέθηκε η μικρότερη απόσταση και αποθηκεύεται ο συγκεκριμένος μετασχηματισμός και τα μη μηδενικά στοιχεία των `in` και `out`. Δίνονται τα διαγράμματα που προέκυψαν



Συνένωση

function myStitch

Η συνάρτηση χρησιμοποιείται για να συνενωθούν οι 2 εικόνες στο σημείο που πρέπει. Δέχεται σαν εισόδους τις 2 εικόνες και τον μετασχηματισμό που πρέπει να εφαρμοστεί στην δεύτερη εικόνα και σαν έξοδο δίνει την ενωμένη εικόνα.

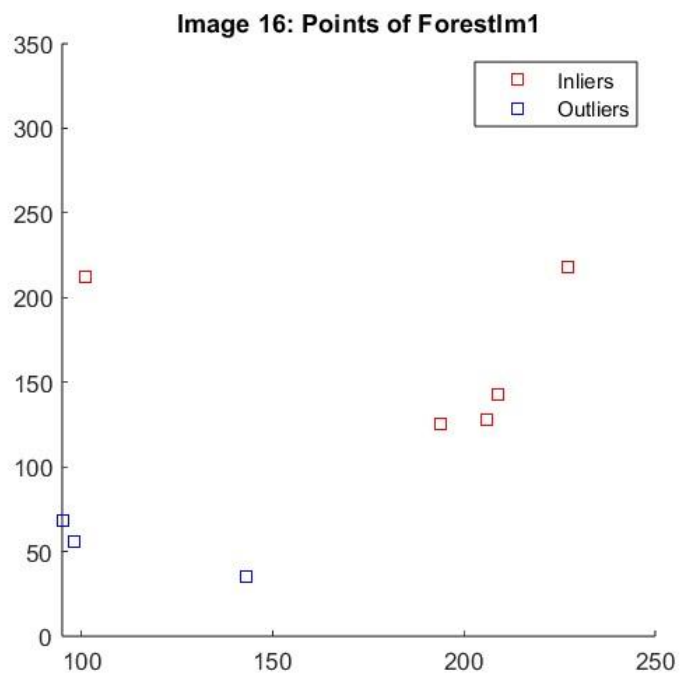
Αρχικά δημιουργείται μια εικόνα με τις διαστάσεις της δεύτερης εικόνας. Υπολογίζεται ο γεωμετρικός μετασχηματισμός που πρέπει να εφαρμοστεί και εφαρμόζεται. Οι 2 εικόνες περιστρέφονται 90 μοίρες διότι οι συντεταγμένες των σημείων δίνονται με περιστροφή στην αρχή.

Τέλος οι 2 εικόνες συνενώνονται και εμφανίζεται η τελική εικόνα.

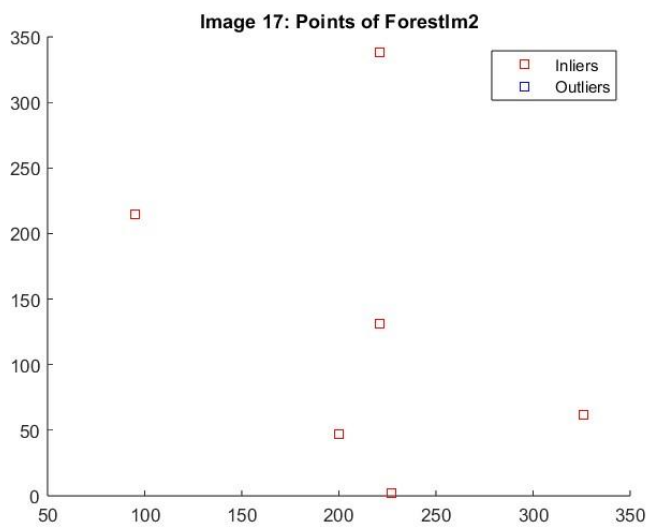
Image 15: Stitched Image



Οι συναρτήσεις εφαρμόστηκαν και στην εικόνα με το δάσος και αυτά είναι τα αποτελέσματα που προέκυψαν.

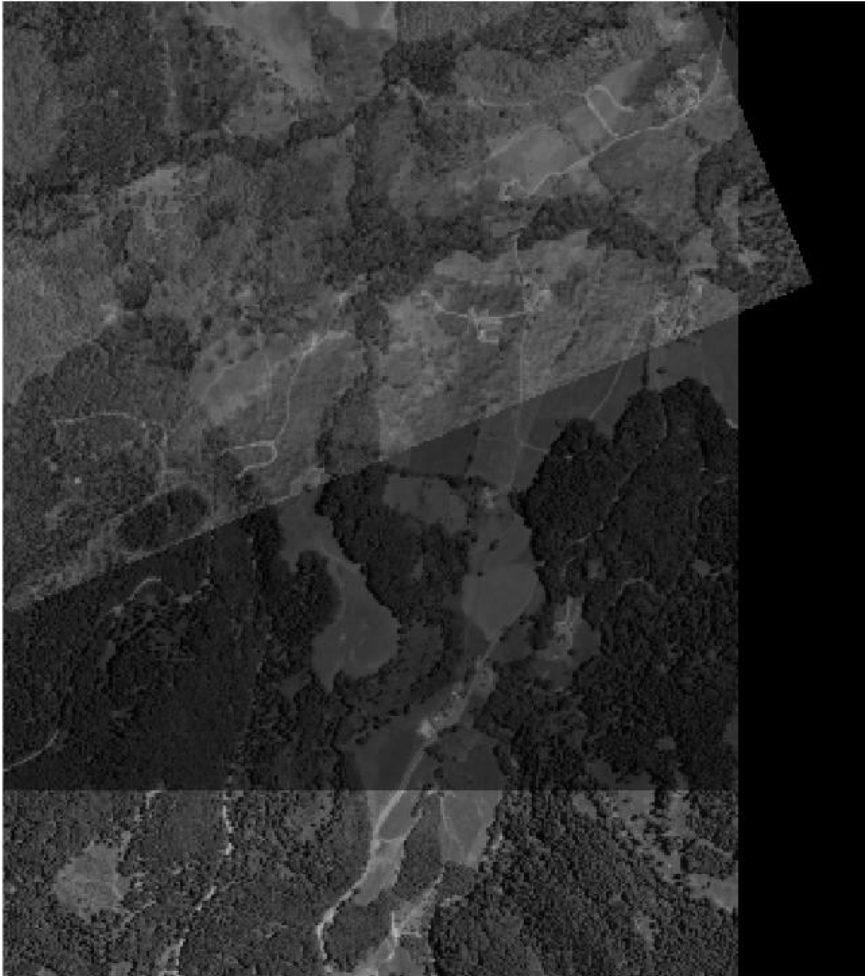


2



Και τέλος, η ενωμένη εικόνα του δάσους.

Image 18: Stitched Image



Αν η φωτογραφία είχε ληφθεί από διαφορετικό ύψος, ο μετασχηματισμός μετατόπισης και περιστροφής δεν θα ήταν αρκετός. Θα χρειαζόταν μεγένθυση ή συρρίκνωση της μιας εικόνας.