

# Ψηφιακή Επεξεργασία Εικόνας

3<sup>η</sup> Εργασία Μαθήματος 2022 - 2023

Αλεξόπουλος Δημήτριος ΑΕΜ 10091

[aadimitri@ece.auth.gr](mailto:aadimitri@ece.auth.gr)

# Περιεχόμενα

1	<b>Εισαγωγή: Image Registration</b>	4
2	<b>Local Descriptor</b>	5
2.1	my_local_descriptor . . . . .	5
2.2	my_local_descriptor_upgrade . . . . .	5
2.3	Αποτελέσματα - Σχολιασμοί . . . . .	6
3	<b>Harris Corner Detector</b>	8
3.1	is_corner . . . . .	9
3.2	my_detect_harris_features . . . . .	10
3.3	Αποτελέσματα - Σχολιασμοί . . . . .	10
4	<b>Matching Descriptors - RANSAC</b>	12
4.1	descriptor_matching . . . . .	12
4.2	my_ransac . . . . .	12
4.3	helpers . . . . .	14
4.4	Αποτελέσματα - Σχολιασμοί . . . . .	14
5	<b>Image Stitching</b>	16
5.1	my_stitch . . . . .	16
5.2	Αποτελέσματα - Σχολιασμοί . . . . .	16

# Κατάλογος Σχημάτων

1.1	Φωτογραφία 1 . . . . .	4
1.2	Φωτογραφία 2 . . . . .	4
3.1	Γωνίες Φωτογραφίας 1 . . . . .	10
3.2	Γωνίες Φωτογραφίας 2 . . . . .	10
4.1	<i>inliers/outliers</i> Φωτογραφίας 1 . . . . .	14
4.2	<i>inliers/outliers</i> Φωτογραφίας 2 . . . . .	14
5.1	Συνενωμένη εικόνα αστικής περιοχής . . . . .	17
5.2	Salt and pepper noise . . . . .	17
5.3	Filtered image . . . . .	17
5.4	Δασική φωτογραφία 1 . . . . .	18
5.5	Δασική φωτογραφία 2 . . . . .	18
5.6	Συνενωμένη εικόνα δασικής περιοχής . . . . .	19

# 1 Εισαγωγή: Image Registration

Αντικείμενο της παρούσας εργασίας είναι η ευθυγράμμιση και η συνένωση δύο διαφορετικών εικόνων σε μια ενιαία για τη δημιουργία μιας συνεχτικής αναπαράστασης. Οι εικόνες αυτές μπορεί να παρουσιάζουν διαφοροποιήσεις όσον αφορά το μέσο λήψης, την γωνία και τις συνθήκες λήψης.

Προκειμένου να επιλύσουμε το παραπάνω πρόβλημα θα υλοποιήσουμε, αρχικά, μια σουίτα από ρουτίνες που θα χρησιμοποιηθούν για την τελική ευθυγράμμιση και συνένωση των δύο εικόνων. Συγκεκριμένα, θα υλοποιήσουμε:

- έναν τοπικό περιγραφέα
- τον Harris Corner Detector
- αντιστοίχιση σημείων με την μέθοδο RANSAC

Για τους σκοπούς της εργασίας θα παρουσιάσουμε αναλυτικά τα αποτελέσματα των επιμέρους συναρτήσεων, καθώς και της τελικής συνένωσης, έχοντας ως δείγμα τις δύο παρακάτω εικόνες (αεροφωτογραφίες) της Θεσσαλονίκης:



Σχήμα 1.1: Φωτογραφία 1



Σχήμα 1.2: Φωτογραφία 2

Για τον έλεγχο της ευρωστίας του αλγορίθμου που θα υλοποιήσουμε, θα επιβεβαιώσουμε τελικά την λειτουργία του και σε ένα διαφορετικό σετ εικόνων.

## 2 Local Descriptor

Πρώτο μας μέλημα είναι η κατασκευή μιας ρουτίνας που να υλοποιεί τον υπολογισμό ενός απλού *rotation invariant* περιγραφέα της γειτονιάς ενός σημείου  $p = [p_1, p_2]$ . Ο συγκεκριμένος *local descriptor* κατασκευάζεται σαρώνοντας διαδοχικούς ομόκεντρους κύκλους με κέντρο το  $p$  και ακτίνες  $\rho = \rho_m : \rho_s : \rho_M$ , όπου τόσο η μικρότερη/μεγαλύτερη ακτίνα όσο και το βήμα  $\rho_s$  είναι παράμετροι του αλγορίθμου υπολογισμού.

Οι κύκλοι σαρώνονται σε  $2\pi/N$  σημεία και, έτσι, για καθέναν από τους κύκλους υπολογίζεται ένα διάνυσμα  $x_\rho = [x_{\rho,0}, \dots, x_{\rho,N-1}]$ . Τα στοιχεία του διανύσματος υπολογίζονται από παρεμβολή των στοιχείων της εικόνας με βάση τη θέση στην οποία αντιστοιχούν. Η βασική έκδοση τους περιγραφέα είναι ένα διάνυσμα  $d$  με τόσα στοιχεία όσα οι κύκλοι  $((\rho_M - \rho_m)/\rho_s)$  που το καθένα έχει τιμή ίση προς το μέσο όρο του αντίστοιχου  $x_\rho$ .

Έπειτα από την υλοποίηση της βασικής εκδοχής του περιγραφέα θα προχωρήσουμε και στην υλοποίηση μιας παραλλαγής του που παραμένει *rotation invariant*, αλλά περιέχει πλουσιότερη πληροφορία.

### 2.1 my\_local\_descriptor

Κατασκευάζουμε την βασική εκδοχή του τοπικού περιγραφέα που περιγράφηκε παραπάνω υλοποιώντας την συνάρτηση  $\text{descriptor} = \text{my\_local\_descriptor}(\text{img}, p, \text{rhom}, \text{rhoM}, \text{rhostep}, N)$  με τα παρακάτω ορίσματα κι εξόδους:

- *img*: η εικόνα εισόδου
- *p*: το σημείο της γειτονιάς του οποίου επιθυμούμε να βρούμε τον περιγραφέα
- *rhom, rhoM*: η μικρή και μεγάλη αντίστοιχα ακτίνα των ομόκεντρων κύκλων
- *rhostep*: το βήμα αύξησης της ακτίνας
- *N*: ο αριθμός σαρώσεων πάνω σε έναν ομόκεντρο κύκλο
- *descriptor*: το διάνυσμα εξόδου του περιγραφέα

Η συνάρτηση περιέχει την λογική της διαδοχικής σάρωσης ομόκεντρων κύκλων γύρω από το σημείο ενδιαφέροντος και της παρεμβολής για την παραγωγή του διανύσματος του περιγραφέα. Λόγω της ιδιότητας των κύκλων η μέθοδος αυτή είναι *rotation invariant*.

### 2.2 my\_local\_descriptor\_upgrade

Στην προηγούμενη προσέγγιση, της συνάρτησης `my_local_descriptor()`, υπολογίζαμε τον μέσο όρο των τιμών των εικονοστοιχείων κατά μήκος κάθε κύκλου. Αυτή η προσέγγιση παρέχει μια ενιαία αντιπροσωπευτική τιμή για κάθε κύκλο, ωστόσο, με τη λήψη του μέσου όρου, χάνουμε κάποια πληροφορία σχετικά με την κατανομή των τιμών των εικονοστοιχείων εντός κάθε κύκλου.

Επιχειρούμε, λοιπόν, να εκπροσωπήσουμε κάθε κύκλο με βάση ένα ιστόγραμμα. Κατασκευάζουμε ένα ιστόγραμμα των τιμών των εικονοστοιχείων εντός κάθε κύκλου κι αντί να υπολογίζουμε τον μέσο όρο των τιμών, καταγράφουμε τη συχνότητα των διαφορετικών τιμών των εικονοστοιχείων στη γειτονιά του σημείου ενδιαφέροντος. Αυτό το ιστόγραμμα αναπαριστά την κατανομή των τιμών των εικονοστοιχείων, παρέχοντας, έτσι, πιο λεπτομερείς πληροφορίες σχετικά με τις διακυμάνσεις της έντασης εντός της γειτονιάς.

Κάθε *bin* του ιστογράμματος αντιπροσωπεύει ένα εύρος τιμών των εικονοστοιχείων κι αυτό αποτυπώνει τόσο τις κυρίαρχες τιμές των εικονοστοιχείων, όσο και τυχόν παραλλαγές ή μοτίβα που υπάρχουν στη γειτονιά.

Με βάση αυτήν την λογική κατασκευάζουμε την συνάρτηση `descriptor = my_local_des(descriptor_upgrade(img, p, rhom, rhoM, rhostep, N, num_bins)`, η οποία περιέχει τα ίδια ορίσματα με την προηγούμενη υλοποίηση κι επιπλέον την παράμετρο `num_bins` που μας επιτρέπει να ρυθμίσουμε τον αριθμό των *bins* του κάθε ιστογράμματος.

## 2.3 Αποτελέσματα - Σχολιασμοί

Θα παρουσιάσουμε τα αποτελέσματα των δύο τοπικών περιγραφέων για τα παραχάτω σημεία. Οι υπολογισμοί θα γίνουν με  $rhom = 5$ ,  $rhoM = 20$ ,  $rhostep = 1$  και  $N = 8$ .

- $p = [100, 100]$ :

$$d_{avg} = [148.88, 147.83, 147.08, 161.08, 160.54, \\ 156.00, 157.83, 151.33, 149.71, 146.25, \\ 136.71, 128.63, 114.50, 117.04, 118.38]$$

- $q = [200, 200]$ :

$$d_{avg} = [75.25, 80.33, 79.58, 81.96, 91.25, \\ 94.04, 89.29, 75.63, 82.21, 83.13, \\ 80.29, 87.21, 84.92, 86.46, 80.29]$$

- $q = [202, 202]$ :

$$d_{avg} = [109.21, 90.67, 87.00, 89.00, 89.71, \\ 78.67, 74.96, 72.21, 76.25, 78.29, \\ 80.75, 82.50, 86.25, 84.54, 76.33]$$

Παρατηρούμε ότι σε κοντινές περιοχές οι περιγραφές δεν διαφέρουν κατά πολύ, ωστόσο σε περιοχή διαφορετικής υφής οι διαφορές ειναι εμφανείς. Ο περιγραφές με την χρήση ιστογράμματος δεν γίνεται εύκολα αντιληπτός, καθώς απαιτούνται πολλά *bins* για την εύρυθμη λειτουργία του, ωστόσο παραθέτουμε ένα παράδειγμα για το σημείο  $p = [202, 202]$  και για 8 *bins*:

$$d_{hist} = [0.0, 1.0, 3.0, 3.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0, 6.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 5.0, 1.0, \\ 0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 4.0, 2.0, 0.0, 0.0, 0.0, 0.0, 0.0, 3.0, 2.0, 3.0, 0.0, 0.0, 0.0, 0.0,$$

0.0, 1.0, 3.0, 3.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 2.0, 2.0, 3.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 3.0, 1.0,  
1.0, 3.0, 0.0, 0.0, 0.0, 0.0, 0.0, 3.0, 0.0, 2.0, 3.0, 0.0, 0.0, 0.0, 0.0, 3.0, 0.0, 1.0, 4.0, 0.0,  
0.0, 0.0, 0.0, 3.0, 0.0, 0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 3.0, 1.0, 1.0, 1.0, 1.0, 1.0, 0.0, 0.0,  
4.0, 0.0, 0.0, 0.0, 2.0, 0.0, 2.0, 0.0, 0.0, 4.0, 0.0, 0.0, 2.0, 1.0, 1.0, 0.0, 0.0, 4.0, 0.0, 0.0,  
3.0, 1.0, 0.0, 0.0, 0.0]

### 3 Harris Corner Detector

Για τον εντοπισμό σημείων ενδιαφέροντος θα υλοποιήσουμε τον αλγόριθμο *Harris corner detector*, του οποίου θα παρουσιάσουμε την βασική ιδέα. Ας υποθέσουμε ότι  $w(x_1, x_2)$  είναι μία διδιάστατη συνάρτηση που έχει μη μηδενικές τιμές κοντά στην αρχή των αξόνων και “πεθαίνει” καθώς το  $(x_1, x_2)$  απομακρύνεται από το  $(0, 0)$ . Για παράδειγμα:

$$w(x_1, x_2) = \exp \left\{ -\frac{x_1^2 + x_2^2}{2\sigma^2} \right\}$$

Αν  $I(x_1, x_2)$  είναι η φωτεινότητα μιας *gray scale* εικόνας, τότε η συνάρτηση

$$E(x_1, x_2; p_1, p_2) = \sum_{u_1, u_2} w(u_1, u_2) \|I(p_1 + u_1 + x_1, p_2 + u_2 + x_2) - I(p_1 + u_1, p_2 + u_2)\|^2$$

παρουσιάζει την εξής συμπεριφορά:

- Αν το σημείο  $(p_1, p_2)$  βρίσκεται σε μία σχετικά ομαλή περιοχή της εικόνας, τότε  $E(x_1, x_2; p_1, p_2) \approx 0$ .
- Αν το  $(p_1, p_2)$  βρίσκεται πάνω ή πολύ κοντά σε μία ακμή η οποία σε σχέση με το μέγεθος της μη μηδενικής περιοχής του παραπέραν  $w(x_1, x_2)$  είναι ευθύγραμμη, τότε  $E(x_1, x_2; p_1, p_2) \approx 0$  όταν η ολίσθηση  $(x_1, x_2)$  είναι (σχεδόν) παράλληλη με την ακμή, ενώ θα παίρνει τιμές  $>> 0$  όταν η ολίσθηση  $(x_1, x_2)$  είναι προς την κάθετη στην ακμή κατεύθυνση.
- Αν το  $(p_1, p_2)$  βρίσκεται πάνω ή πολύ κοντά σε μία γωνία, τότε  $E(x_1, x_2; p_1, p_2) >> 0$  για οποιαδήποτε κατεύθυνση της ολίσθησης  $(x_1, x_2)$ .

Συνεπώς, η μελέτη της  $E(x_1, x_2; p_1, p_2)$  ως συνάρτηση των  $(x_1, x_2)$  μας επιτρέπει να αποφασίζουμε αν το σημείο  $(p_1, p_2)$  βρίσκεται σε περιοχή ομαλής φωτεινότητας, σε ακμή ή σε γωνία. Χρησιμοποιώντας ανάπτυγμα *Taylor* γύρω από το σημείο  $(p_1 + u_1, p_2 + u_2)$  ως προς τις μεταβλητές  $(x_1, x_2)$ , η  $E(x_1, x_2; p_1, p_2)$  γράφεται:

$$E(x_1, x_2; p_1, p_2) \approx \sum_{u_1, u_2} w(u_1, u_2) \|x_1 I_1(p_1 + u_1, p_2 + u_2) + x_2 I_2(p_1 + u_1, p_2 + u_2)\|^2$$

όπου  $I_1(x_1, x_2) = \frac{\partial I(x_1, x_2)}{\partial x_1}$ ,  $I_2(x_1, x_2) = \frac{\partial I(x_1, x_2)}{\partial x_2}$  και έχουμε παραλείψει τους όρους υψηλότερης τάξης.

Εναλλακτικά, μπορούμε να γράψουμε:

$$E(x_1, x_2; p_1, p_2) \approx [x_1 \quad x_2] M(p_1, p_2) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

όπου

$$M(p_1, p_2) = \sum_{u_1, u_2} w(u_1, u_2) A(u_1, u_2; p_1, p_2)$$

και

$$A(u_1, u_2; p_1, p_2) = \begin{bmatrix} I_1(p_1 + u_1, p_2 + u_2)^2 & I_1(p_1 + u_1, p_2 + u_2)I_2(p_1 + u_1, p_2 + u_2) \\ I_1(p_1 + u_1, p_2 + u_2)I_2(p_1 + u_1, p_2 + u_2) & I_2(p_1 + u_1, p_2 + u_2)^2 \end{bmatrix}$$

Πρακτικά, οι μερικές παράγωγοι που συμμετέχουν στις παραπάνω εκφράσεις υπολογίζονται με τη χρήση κατάλληλων συνελικτικών μασκών. Ο υπολογισμός τους μάλιστα γίνεται μια και έξω για όλη την εικόνα. Οι τρεις εναλλακτικές συμπεριφορές της συνάρτησης  $E(x_1, x_2; p_1, p_2)$  που είδαμε παραπάνω αντιστοιχούν σε τρεις αντίστοιχες διαφορετικές εκδοχές για τις ιδιοτιμές του πίνακα  $M(p_1, p_2)$ :

- Αν το σημείο  $(p_1, p_2)$  βρίσκεται σε μία σχετικά ομαλή περιοχή της εικόνας, τότε  $\lambda_1 \approx 0$  και  $\lambda_2 \approx 0$ .
- Αν το  $(p_1, p_2)$  βρίσκεται πάνω ή πολύ κοντά σε μία ακμή, τότε  $\lambda_1 \gg \lambda_2 \approx 0$ .
- Αν το  $(p_1, p_2)$  βρίσκεται πάνω ή πολύ κοντά σε μία γωνία, τότε  $\lambda_1 \approx \lambda_2 \gg 0$ .

Συνεπώς, η απόφαση για το αν η περιοχή του σημείου  $(p_1, p_2)$  είναι ομαλή, βρίσκεται σε ακμή ή σε γωνία μπορεί να ληφθεί από τη μελέτη των δύο ιδιοτιμών του πίνακα  $M(p_1, p_2)$ . Για λόγους υπολογιστικής πολυπλοκότητας, καθώς ο υπολογισμός των ιδιοτιμών απαιτεί την εκτέλεση τετραγωνικών ριζών, χρησιμοποιούμε την παρακάτω μετρική γωνιότητας:

$$R(p_1, p_2) = \det(M(p_1, p_2)) - k \text{Trace}(M(p_1, p_2))^2 \quad (3.1)$$

όπου  $k > 0$  μια οριζόμενη από εμάς παράμετρος. Επειδή

$$R(p_1, p_2) = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2$$

η τιμή  $R$  θα είναι θετική όταν το σημείο  $(p_1, p_2)$  είναι κοντά σε γωνίες, αρνητική στη γειτονιά ακμών και κοντά στο μηδέν σε ομοιόμορφες περιοχές. Χρησιμοποιώντας κατώφλια, μπορούμε να επιλέξουμε τις πλέον ευδιάχριτες γωνίες, και τελικά να έχουμε στην διάθεσή μας τις θέσεις των πιζελς όπου εντοπίζονται γωνίες.

### 3.1 is\_corner

Για να υλοποιήσουμε την παραπάνω λογική, κατασκευάζουμε, αρχικά, την συνάρτηση  $c = \text{is\_corner}(img, p, k, Rthres)$ , οποία έχει τα εξής ορίσματα κι εξόδους:

- $img$ : η εικόνα εισόδου
- $p$ : διάνυσμα σημείων ενδιαφέροντος
- $k$ : παράμετρος  $k$  της εξίσωσης (3.1)
- $Rthres$ : το επιλεγμένο από εμάς κατώφλι
- $c$ : το διάνυσμα εξόδου

Η συνάρτηση αυτή επιστρέφει ένα διάνυσμα με *true/false* αν η *grayscale* εικόνα *img* εμφανίζει γωνία στα *pixels p*. Η παράμετρος εισόδου  $R_{thres}$  ορίζει το κατώφλι πάνω από το οποίο η τιμή της  $R(p_1, p_2)$  θεωρείται αρκετά θετική ώστε να εντοπίζεται γωνία. Ο λόγος που επιλέχθηκε η είσοδος και η έξοδος της συνάρτησης να είναι διανύσματα είναι η μείωση της πολυπλοκότητας του αλγορίθμου.

### 3.2 my\_detect\_harris\_features

Έπειτα υλοποιούμε την συνάρτηση *corners = my\_detect\_harris\_features(img)*, η οποία χρησιμοποιεί την συνάρτηση *is\_corner* για τον εντοπισμό των γωνιών μιας εικόνας. Έχει τα παρακάτω ορίσματα κι εξόδους:

- *img*: η εικόνα εισόδου
- *corners*: πίνακας με τις συντεταγμένες των σημείων της εικόνας όπου εντοπίζεται γωνία

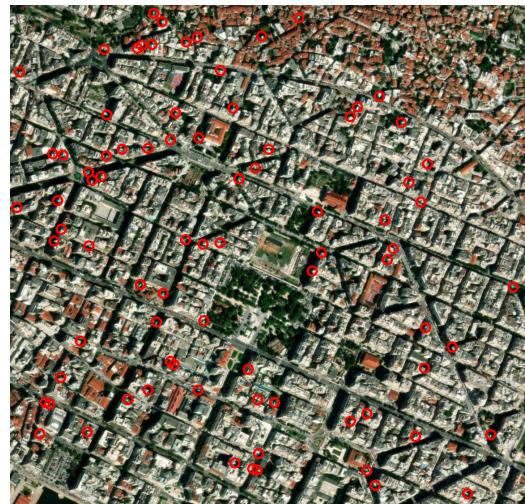
Η συνάρτηση υλοποιεί τον αλγόριθμο εύρεσης γωνιών, όπως περιγράφηκε παραπάνω. Η μεταβλητή *img* είναι ένας πίνακας 2 διαστάσεων και περιέχει μία εικόνα σε *gray scale*, με τιμές πραγματικούς αριθμούς στο διάστημα  $[0, 1]$ . Σκανάρει τα σημεία της εικόνας και με βάση την συνάρτηση *is\_corner()* αποφασίζει ποιά από αυτά είναι γωνίες. Για την αποφυγή πολλαπλών γειτονικών σημείων και την μείωση της πολυπλοκότητας για κάθε γωνία επιλέγεται μια ακτίνα εντός της οποίας αποκλείονται τα σημεία ως πιθανές γωνίες.

### 3.3 Αποτελέσματα - Σχολιασμοί

Για την επίδειξη της λειτουργίας του αλγορίθμου ακολουθούν οι δύο εικόνες των Σχημάτων 1.1 και 1.2 με σημειωμένες τις γωνίες που εντοπίστηκαν:



Σχήμα 3.1: Γωνίες Φωτογραφίας 1



Σχήμα 3.2: Γωνίες Φωτογραφίας 2

Επιλέχθηκε αρκετά υψηλό *Rthres*, ώστε να προκύψουν λίγα σημεία και να είναι αποδοτικός ο αλγόριθμος. Μπορούμε να επιβεβαιώσουμε την λειτουργία του και με την χρήση της συνάρ-

της `cv2.cornerHarris(img_gray, blockSize = 2, ksize = 3, k = 0.04)` της βιβλιοθήκης *openCV* της *python*. Τα αποτελέσματα με την κατάλληλη ρύθμιση των παραμέτρων είναι πανομοιότυπα με αυτά των εικόνων 3.1 και 3.2.

## 4 Matching Descriptors - RANSAC

Έχοντας βρει με αυτόν τον τρόπο τα ιδιάζοντα σημεία και με την βοήθεια των περιγραφέων που κατασκευάσαμε στις προηγούμενες ενότητες μπορούμε να αντιστοιχίσουμε τα ανιχνευμένα σημεία μεταξύ των δύο εικόνων. Για να το πετύχουμε αυτό, υπολογίζουμε τις Ευκλείδειες αποστάσεις των περιγραφέων για όλα τα ζεύγη ανιχνευμένων σημείων και δημιουργούμε έναν πίνακα αποστάσεων, με διαστάσεις  $N_1 \times N_2$ , όπου  $N_i$  το πλήθος ανιχνευμένων σημείων της  $i = 1, 2$  εικόνας.

Με βάση αυτόν τον μη συμμετρικό πίνακα αποστάσεων, μπορούμε να δούμε πόσο 'καλά' αντιστοιχίζεται ένα ανιχνευμένο σημείο της 1ης εικόνας με κάποιο ανιχνευμένο σημείο της 2ης εικόνας. Η μέθοδος αυτή περιλαμβάνει έλεγχο και άρα υπολογισμό όλων των δυνατών συνδυασμών ζευγών ανιχνευμένων σημείων και χαρακτηρίζεται ως exhaustive search. Παρατηρούμε ότι ένα σημείο της μίας εικόνας μπορεί να αντιστοιχηθεί 'καλά' σε πολλαπλά σημεία της άλλης εικόνας, ενώ κάποιες αντιστοιχίεις μπορεί να είναι εσφαλμένες (outliers).

### 4.1 descriptor\_matching

Τηλοποιούμε με βάση την παραπάνω λογική την συνάρτηση  $matching\_points = descriptor\_matching(points1, points2, img1, img2, percentageThreshold)$ , η οποία έχει τα εξής ορίσματα κι εξόδους:

- $points1, points2$ : οι γωνίες που εντοπίστηκαν στην 1η και 2η φωτογραφία αντίστοιχα
- $img1, img2$ : οι δύο φωτογραφίες
- $percentageThreshold$ : ποσοστό για την επιλογή κατωφλίου περιορισμού των σημείων
- $matching\_points$ : πίνακας  $2 \times n$  με τα ζεύγη ανιχνευμένων σημείων που είναι πιο πιθανό να εχουν αντιστοιχηθεί

Για να περιορίσουμε το πλήθος των πιθανών ζευγών σημείων εφαρμόζουμε ένα κατώφλι στην απόσταση των περιγραφέων. Συγκεκριμένα θέλουμε να κρατήσουμε ενα ποσοστό,  $percentageThreshold$  (πχ 30%), των πιθανών ζευγών σημείων, οπότε θα εφαρμόσουμε το κατάλληλο κατώφλι που διατηρεί αυτό το επιλεγένο ποσοστό από τα  $N_1 \times N_2$  πιθανά ζεύγη σημείων.

### 4.2 my\_ransac

Για να βρούμε, έπειτα, τον μετασχηματισμό περιστροφής κατά γωνία  $\theta$  και μετατόπισης κατά διάνυσμα  $\vec{d}$  αρχεί να έχουμε 2 σημεία που να αντιστοιχίζονται σωστά στις εικόνες, δηλαδή 2 ζεύγη σημείων. Αν  $P_1, P_2$  είναι οι συντεταγμένες του ίδιου σημείου στις δύο εικόνες τότε θεωρούμε ότι

$$P_2 = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} P_1 + \vec{d}$$

Θεωρούμε  $H = \{\theta, \vec{d}\}$ , το οποίο αναπαριστά το μετασχηματισμό περιστροφής και μετατόπισης ώστε τα κοινά τους σημεία να συμπίπτουν.

Αντί να χρησιμοποιήσουμε τα δύο 'καλύτερά' ζεύγη σημείων, θα υλοποιήσουμε μια πιο εύρωστη (*robust*) τεχνική που λαμβάνει υπόψη όλα τα ζεύγη σημείων που έχουμε βρει να αντιστοιχίζονται 'καλύτερά', αγνοώντας όσα δεν συμφωνούν (outliers). Για να το πετύχουμε αυτό θα χρειαστεί να υλοποιήσουμε τον αλγόριθμο *RANdom SAmple Consensus* (*RANSAC*). Η εκτίμηση της βέλτιστης λύσης προκύπτει από το μέγεθος του 'σετ συμφωνίας' (*consensus set*), δηλαδή επιλέγεται ως τελικό μοντέλο αυτό στο οποίο συμφωνεί το μεγαλύτερο σετ δεδομένων.

Επαναλαμβανόμενα βήματα *RANSAC*:

1. Επίλεξε τυχαία δύο ζεύγη σημείων (*random minimal set*)
2. Υπολόγισε το μετασχηματισμό  $H$  που ορίζεται από αυτά τα δύο ζεύγη
3. Υπολόγισε το *score* ως το άθροισμα των Ευκλείδειων αποστάσεων ανάμεσα στα υπόλοιπα ζεύγη σημείων που προκύπτουν μετά την εφαρμογή του μετασχηματισμού.

Ο αλγόριθμος επαναλαμβάνεται  $N$  φορές και έχει ως παράμετρο την ακτίνα  $r$ , που χρησιμοποιείται ως κριτήριο κατάταξης του κάθε ζεύγους σημείων σε *inlier* ή *outlier*. Στο τέλος των  $N$  επαναλήψεων επιλέγεται ο μετασχηματισμός με το βέλτιστο *score* και γίνεται ο διαχωρισμός των *inliers* και *outliers* με βάση την ακτίνα  $r$ .

Τλοποιούμε με βάση την παραπάνω λογική την συνάρτηση  $\{H, inlier\_matchingPoints, out - outlier\_matchingPoints\} = my\_RANSAC(matching\_points, r, N)$ , η οποία έχει τα παρακάτω ορίσματα κι εξόδους:

- *matching\_points*: πίνακας  $2 \times n$  με τα ζεύγη ανιχνευμένων σημείων που είναι πιο πιθανό να εχουν αντιστοιχηθεί
- *r*: ακτίνα ως κριτήριο για την διαφοροποίηση των *inliers* από τα *outliers*
- *N*: αριθμός επαναλήψεων
- *H*: ο βέλτιστος μετασχηματισμός
- *inlier\_matchingPoints, out - outlier\_matchingPoints*: λίστες με τα *inliers* σημεία και τα *outliers*

Η συνάρτηση επιστρέφει τον βέλτιστο μετασχηματισμό  $H$  που προκύπτει από την είσοδο *matchingPoints* (δηλαδή  $H.\theta = \theta$  και  $H.\vec{d} = \vec{d}$ ), καθώς επίσης και τα ζεύγη σημείων που θεωρήθηκαν *inliers* και *outliers*.

Αντί να υπολογίζουμε ως σκορ το ελάχιστο άθροισμα των Ευκλείδειων αποστάσεων ανάμεσα στα υπόλοιπα ζεύγη σημείων που προκύπτουν μετά την εφαρμογή του μετασχηματισμού, μετράμε τον αριθμό των ζευγών που με εφαρμογή του κάθε μετασχηματισμού επιφέρουν αποστάσεις μικρότερες ενός *threshold*. Έτσι, επιλέγουμε ως βέλτιστο τον μετασχηματισμό που πετυχαίνει το υψηλότερο σκορ, δηλαδή συγκεντρώνει τις περισσότερες ψήφους από σημεία που οι αποστάσεις του είναι μικρότερες του *threshold*.

### 4.3 helpers

Για πιο ευανάγνωστη υλοποίηση της παραπάνω συνάρτησης κατασκευάσαμε δύο βοηθητικές συνάρτησεις. Αρχικα, την συνάρτηση  $H = calculate\_h(pair1, pair2)$ , η οποία υπολογίζει τον μετασχηματισμό  $H$  με βάση την ανάλυση που προηγήθηκε και περιλαμβάνει τα εξής ορίσματα κι εξόδους:

- $pair1, pair2$ : τα δύο ζεύγη σημείων
- $H$ : ο μετασχηματισμός που ορίζουν

Έπειτα υλοποιούμε την βοηθητική συνάρτηση  $transformed\_points = transform\_points(points, H)$ , η οποία εφαρμόζει τον παραπάνω μετασχηματισμό σε μια λίστα από σημεία κι έχει τα παρακάτω ορίσματα κι εξόδους:

- $points$ : η λίστα με τα σημεία που επιθυμούμε να μετασχηματίσουμε
- $H$ : ο μετασχηματισμός που θα εφαρμοστεί
- $transformed\_points$ : η λίστα με τα μετασχηματισμένα σημεία

### 4.4 Αποτελέσματα - Σχολιασμοί

Θα παρουσιάσουμε τα σημεία που εντοπίσαμε πάνω στις δύο εικόνες και τη μεταξύ τους αντιστοίχιση. Σημειώνουμε με γκρι τετράγωνα τα ανιχνευμένα σημεία που αποτελούν *outliers* στη μεταξύ τους αντιστοίχιση και με διαφορετικά χρώματα, κοινά ανά ζεύγη ανιχνευμένων σημείων, τα σημεία που αντιστοιχίζονται 'καλά' μεταξύ των δύο εικόνων:



Σχήμα 4.1: *inliers/outliers* Φωτογραφίας 1



Σχήμα 4.2: *inliers/outliers* Φωτογραφίας 2

Παρατηρούμε ότι έχει γίνει πολύ καλή αντιστοίχιση των ιδιαζόντων σημείων της εικόνας 1.1 με τα ιδιάζοντα σημεία της εικόνας 1.2. Αυτό σημαίνει ότι έχουμε στην διάθεσή μας σημεία τα οποία μπορούν αξιόπιστα και εύρωστα με βάση την παραπάνω ανάλυση να εντοπίσουν τον

βέλτιστο μετασχηματισμό που εάν εφαρμοστεί στην εικόνα 1.1 θα την φέρει σε μορφή κατάλληλη για να ενωθεί με την εικόνα 1.2. Η ένωση αυτή των δύο εικόνων θα υλοποιηθεί στην επόμενη ενότητα.

## 5 Image Stitching

Έχοντας πλεόν στην διάθεσή μας τον βέλτιστο μετασχηματισμό  $H$ , μπορούμε να τον εφαρμόσουμε στα σημεία της εικόνας 1.1 ώστε να την ευθυγραμμίσουμε και τελικά να την ενώσουμε με την εικόνα 1.2.

### 5.1 my\_stitch

Τηλοποιούμε, λοιπόν, την συνάρτηση  $stitched = myStitch(img1, img2, H)$ , η οποία έχει τα εξής απλά ορίσματα κι εξόδους:

- $img1, img2$ : οι δύο εικόνες που επιθυμούμε να συνενώσουμε
- $stitched$ : η συνενωμένη εικόνα

Η συνάρτηση αυτή δημιουργεί έναν καμβά κατάλληλων διαστάσεων κι έπειτα γεμίζει ένα άλλο της συνενωμένης εικόνας με βάση τις τιμές των πίζελς των δύο αρχικών εικόνων κι ανάλογα με την ύλη στην οποία ανατρέχει. Ο αλγόριθμος αυτός δεν είναι ο πιο αποδοτικός, ωστόσο για τις δεδομένες διαστάσεις των εικόνων αυτό δεν αποτελεί σημαντικό πρόβλημα.

### 5.2 Αποτελέσματα - Σχολιασμοί

Θα παρουσιάσουμε, αρχικά, το αποτέλεσμα της συνένωσης των δύο φωτογραφιών των Σχημάτων 1.1 και 1.2 και στη συνέχεια θα ελέγξουμε τον αλγόριθμό μας σε ένα διαφορετικό σετ εικόνων, μιας δασικής περιοχής.

Παρακάτω βλέπουμε την συνενωμένη εικόνα της αστικής περιοχής. Παρατηρούμε ότι η αντιστοίχιση έχει γίνει ικανοποιητική και με την κατάλληλη ρύθμιση των παραμέτρων μπορούμε να λάβουμε ένα αποτέλεσμα που είναι συνεπές με κάθε εκτέλεση του αλγορίθμου. Μικρές διαφορές στον υπολογισμό του μετασχηματισμού  $H$  μετά από κάθε εκτέλεση παρατηρούνται λόγω μη βέλτιστη επιλογής των παραμέτρων:

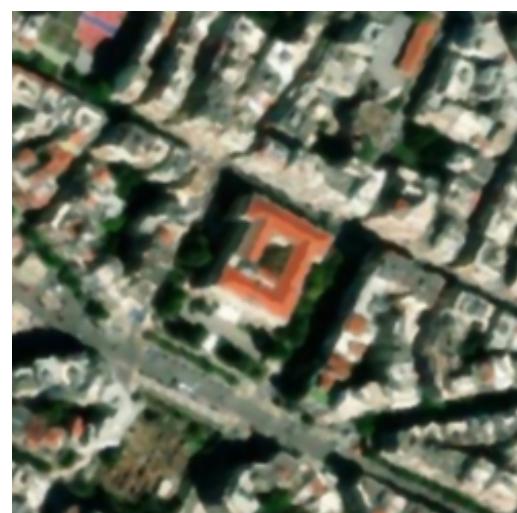


Σχήμα 5.1: Συνενωμένη εικόνα αστικής περιοχής

Εάν μεγενθύνουμε την παραπάνω εικόνα ωσα παρατηρήσουμε κάποιο salt and pepper noise που οφείλεται στην διαδικασία που ακολουθήσαμε για την συνένωση και στις στρογγυλοποιήσεις. Γι' αυτόν τον λόγο εφαρμόζουμε στην εικόνα ένα medianBlur φίλτρο κι εξομαλύνουμε την εικόνα. Χάνουμε, ωστόσο, λίγο σε ευχρίνεια:



Σχήμα 5.2: *Salt and pepper noise*



Σχήμα 5.3: *Filtered image*

Δοκιμάζουμε, τώρα, να εφαρμόσουμε τον αλγόριθμο που υλοποιήσαμε σε ένα σετ δύο φωτογραφιών μιας δασικής περιοχής:



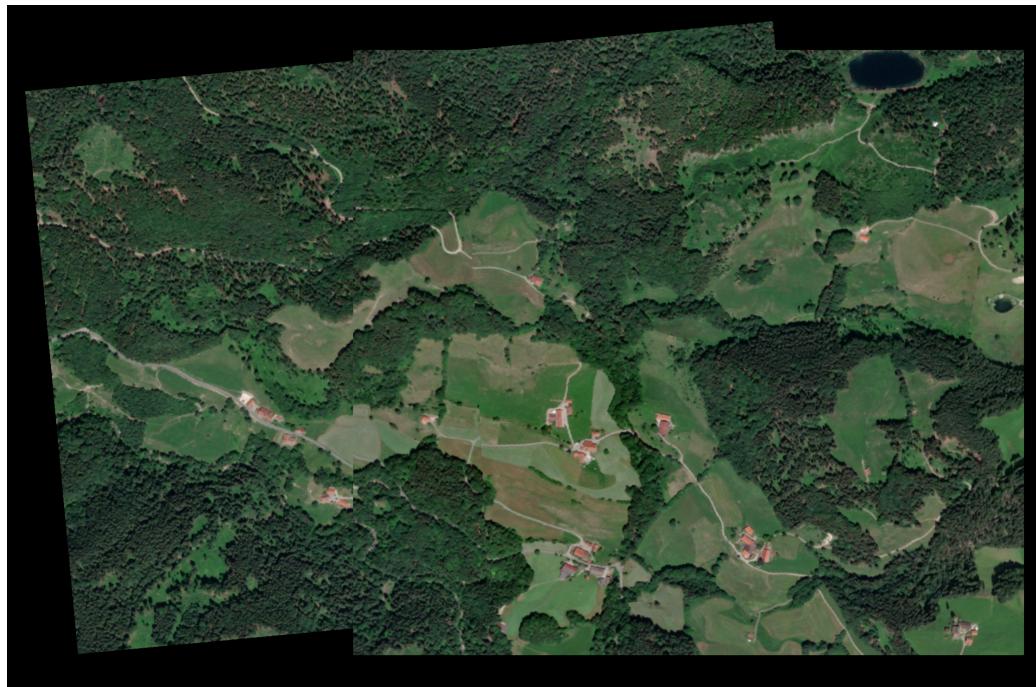
Σχήμα 5.4: Δασική φωτογραφία 1



Σχήμα 5.5: Δασική φωτογραφία 2

Παρατηρούμε ότι με το όριο  $R_{thres}$  που έχουμε θέσει στο προηγούμενο σετ εικόνων, οι γωνίες που εντοπίζονται δεν αρκούν για την εύρεση του κατάλληλου μετασχηματισμού  $H$ . Το γεγονός αυτό φαίνεται λογικό από τη στιγμή που η δασική περιοχή έχει πολύ λιγότερες γωνίες από μια αστική περιοχή.

Για τον λόγο αυτό θα χρειαστεί να μειώσουμε το κατώφλι  $R_{thres}$ , ώστε να αυξηθεί ικανοποιητικά ο αριθμός των ιδιαζόντων σημάτων των δύο εικόνων. Από εκεί και πέρα ο αλγόριθμος λειτουργεί ικανοποιητικά και με τον ίδιο τρόπο που λειτουργούσε και στο πρώτο σετ εικόνων:



Σχήμα 5.6: Συνενωμένη εικόνα δασικής περιοχής

Αξίζει να σημειώσουμε ότι οι δύο αεροφωτογραφίες στην αστική περιοχή εχουν την ίδια χωρική ανάλυση (*Ground Sampling Distance - GSD*) και το ίδιο συμβαίνει και γι' αυτές στις δασικές εικόνες. Εαν δεν ίσχυε αυτή η συνθήκη, για παράδειγμα εαν οι δύο φωτογραφίες είχαν ληφθεί από διαφορετικό ύψος με τον ίδιο αισθητήρα κάμερας, ο αλγόριθμος δεν θα διέλευε, καθώς στηρίζεται στις αποστάσεις (σε πιξελ) μεταξύ σημείων των δύο εικόνων για να εντοπίσει τα αντίστοιχα ιδιάζοντα σημεία που δίνουν τον βέλτιστο μετασχηματισμό  $H$ . Η λύση που θα μπορούσε να προταθεί είναι η κατάλληλη αλλαγή μεγέθους της μίας εκ των δύο εικόνων με γραμμική παρεμβολή, ώστε να αποκτήσει την ίδια χωρική πυκνότητα με την δεύτερη εικόνα.