

# Ψηφιακή Επεξεργασία Εικόνας -Εργασία 3-

---

## Harris corner detector, SURF descriptor & Panorama stitching

A. Ντελόπουλος

Άνοιξη 2016

### Εισαγωγικά

Το θέμα της παρούσας εργασίας είναι η ανάλυση εικόνας, και η χρήση της ανάλυσης για την συνένωση πολλαπλών εικόνων σε μία. Το σενάριο που θα υλοποιήσετε περιλαμβάνει

1. Τον εντοπισμό σημείων ενδιαφέροντος (interest points/salient points) με τη χρήση του Harris corner detector.
2. Την εξαγωγή χαρακτηριστικών (descriptors) στα σημεία ενδιαφέροντος με τον αλγόριθμο SURF.
3. Την συνένωση εικόνων σε ένα πανόραμα (panorama stitching) με χρήση ενός RANSAC αλγόριθμου.

### 1 Harris corner detector

Για τον εντοπισμό σημείων ενδιαφέροντος θα υλοποιήσετε τον αλγόριθμο Harris corner detector που πρωτοπροτάθηκε στο το άρθρο [1]. Ακολουθεί η βασική ιδέα. Ας υποθέσουμε ότι  $w(x_1, x_2)$  είναι μία διδιάστατη συνάρτηση που έχει μή μηδενικές τιμές κοντά στην αρχή των αξόνων και "πεθαίνει" καθώς το  $(x_1, x_2)$  απομακρύνεται από το  $(0, 0)$ . Για παράδειγμα:

$$w(x_1, x_2) = \exp\left\{-\frac{x_1^2 + x_2^2}{2\sigma^2}\right\} \quad (1)$$

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

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

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 παραθύρου  $w(x_1, x_2)$  - 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  $\gg 0$  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  $E(x_1, x_2; p_1, p_2) \gg 0$  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

$$E(x_1, x_2; p_1, p_2) \approx \sum_{u_1, u_2} w(u_1, u_2) \|x_1 I_1(p_1 + u_1, p_2 + u_2) + x_2 I_2(p_1 + u_1, p_2 + u_2)\|^2 \quad (3)$$

όπου  $I_1(x_1, x_2) = \partial I(x_1, x_2) / \partial x_1$ ,  $I_2(x_1, x_2) = \partial I(x_1, x_2) / \partial x_2$  και έχουμε παραλήψει τους όρους υψηλότερης τάξης. Εναλλακτικά,

$$E(x_1, x_2; p_1, p_2) \approx [x_1, x_2] \mathbf{M}(p_1, p_2) \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad (4)$$

όπου

$$\mathbf{M}(p_1, p_2) = \sum_{u_1, u_2} w(u_1, u_2) \mathbf{A}(u_1, u_2; p_1, p_2) \quad (5)$$

και

$$\mathbf{A}(u_1, u_2; p_1, p_2) = \begin{bmatrix} I_1(p_1 + u_1, p_2 + u_2)^2 & I_1(p_1 + u_1, p_2 + u_2) I_2(p_1 + u_1, p_2 + u_2) \\ I_1(p_1 + u_1, p_2 + u_2) I_2(p_1 + u_1, p_2 + u_2) & I_2(p_1 + u_1, p_2 + u_2)^2 \end{bmatrix} \quad (6)$$

Πρακτικά οι μερικές παράγωγοι που συμμετέχουν στις παραπάνω εκφράσεις υπολογίζονται με τη χρήση κατάλληλων συνελκτικών μασκών όπως έχουμε εξηγήσει στη θεωρία. Ο υπολογισμός τους μάλιστα γίνεται μια κι 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  $\mathbf{M}(p_1, p_2)$ :

- 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  $\mathbf{M}(p_1, p_2)$ . Για λόγους υπολογιστική πολυπλοκότητας - καθώς ο υπολογισμός ιδιοτιμών απαιτεί υπολογισμό τετραγωνικών ριζών - χρησιμοποιούμε την παρακάτω μετρική γωνιότητας:

$$R(p_1, p_2) = \det(\mathbf{M}(p_1, p_2)) - k \text{Trace}(\mathbf{M}(p_1, p_2))^2 \quad (7)$$

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

$$R(p_1, p_2) = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \quad (8)$$

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. Κατάλληλα κατώφλια μπορούν να χρησιμοποιηθούν για να επιλεγούν οι πλέον ευδιάκριτες γωνίες.

Κατασκευάστε την συνάρτηση

```
corners=myDetectHarrisFeatures(I)
```

η οποία υλοποιεί τον αλγόριθμο, όπως περιγράφεται παραπάνω. Η μεταβλητή  $I$  είναι ένας πίνακας 2 διαστάσεων και περιέχει μία εικόνα σε gray scale, με τιμές πραγματικούς αριθμούς στο διάστημα  $[0, 1]$ . Η μεταβλητή  $corners$  είναι ένας πίνακας δύο στηλών, και κάθε του γραμμή αντιστοιχεί στις συντεταγμένες μίας γωνίας. Ο αλγόριθμος είναι ήδη υλοποιημένος στο Image Processing toolbox του Matlab. Οι παρακάτω εντολές επιδεικνύουν τη χρήση του:

```
1 I=imread('cameraman.tif');
2 corners=detectHarrisFeatures(I);
3 figure
4 hold on
5 imshow(I);
6 plot(corners);
7 hold off
```

Μπορείτε να χρησιμοποιήσετε τη μεταβλητή  $corners.Location$  για να επαληθεύσετε την υλοποίησή σας συγκρίνοντας τα δικά σας αποτελέσματα με αυτά της έτοιμης συνάρτησης. Η ορθότητα της υλοποίησης δεν εξαρτάται προφανώς από την σειρά με την οποία παρατίθενται οι γωνίες.

## 2 SURF descriptor

Στη συνέχεια, θα υλοποιήσετε τον αλγόριθμο υπολογισμού του περιγραφέα SURF, και συγκεκριμένα την απλουστευμένη εκδοχή U-SURF. Ο αλγόριθμος περιγράφεται στην Ενότητα 4 του άρθρου [2]. Η προτεινόμενη απλουστευμένη εκδοχή είναι κατάλληλη για τη συσχέτιση σημείων που εμφανίζονται σε εικόνες που δεν έχουν σημαντική περιστροφή ή μία σε σχέση με την άλλη.

Υλοποιήστε τη συνάρτηση

```
features=extractSurfFeatures(I,intPoints)
```

η οποία υλοποιεί τον αλγόριθμο U-SURF. Ως είσοδος δίνεται μία εικόνα  $I$ , με τις ίδιες προδιαγραφές του προηγούμενου ερωτήματος, και ένα σύνολο σημείων ενδιαφέροντος  $intPoints$ , με τις ίδιες προδιαγραφές όπως η έξοδος της συνάρτησης  $myDetectHarrisFeatures$ . Η έξοδος  $features$  είναι ένας πίνακας ίσων γραμμών με τον πίνακα  $intPoints$ , και κάθε γραμμή είναι το διάνυσμα χαρακτηριστικών (feature vector) του αντίστοιχου σημείου ενδιαφέροντος.

Ο υπολογισμός του SURF descriptor είναι διαθέσιμος και στο image processing toolbox της Matlab μέσω της συνάρτησης  $extractFeatures$ .

## 3 Panorama stitching

Στην ενότητα αυτή θα δημιουργήσετε τα εργαλεία για την συνένωση τμημάτων μίας εικόνας. Προς εφαρμογή, δίνεται μία αρχική εικόνα  $I$ , και τρία χωρία αυτής. Το κάθε

χωρίο προκύπτει από έναν γραμμικό μετασχηματισμό και μία μετατόπιση της αρχικής εικόνας,  $v$

### 3.1 Αλγόριθμος RANSAC για τον υπολογισμό του μετασχηματισμού

Κατασκευάστε τη συνάρτηση

```
R=possiblePairs(features1,features2)
```

η οποία λαμβάνει ως είσοδο δύο πίνακες που περιέχουν τους descriptors (feature vectors) κάθε σημείου κάθε εικόνας, ως γραμμές. Για παράδειγμα, ο descriptor του 5ου σημείου ενδιαφέροντος της πρώτης εικόνας είναι το διάνυσμα  $feature1(5, :)$ . Οι δύο πίνακες πρέπει να έχουν ίδιο αριθμό στηλών. Η συνάρτηση επιστρέφει έναν πίνακα δύο στηλών, με τιμές δείκτες στις γραμμές των δύο πινάκων. Για παράδειγμα, αν η  $i$  γραμμή του πίνακα  $R$  είναι  $R(i, :)= [8 \ 3]$ , συνεπάγεται ότι το 8ο σημείο ενδιαφέροντος της πρώτης εικόνας μοιάζει αρκετά με το 3ο της δεύτερης εικόνας. Δεν απαγορεύεται ένα σημείο μίας εικόνας να μοιάζει με περισσότερα από ένα (ή και κανένα) σημεία της άλλης εικόνας. Για να αποφασίσετε αν μοιάζουν αρκετά δύο σημεία, χρησιμοποιήστε το μέσο τετραγωνικό σφάλμα των descriptors τους.

Στη συνέχεια, κατασκευάστε τη συνάρτηση

```
[A,B]=findAffineTransform(p1,p2)
```

η οποία λαμβάνει ως είσοδο δύο πίνακες διαστάσεων  $3 \times 2$ . Ο πίνακας  $p1$  περιέχει τις συντεταγμένες 3 σημείων (ως γραμμές), ενώ ο πίνακας  $p2$  περιέχει τις συντεταγμένες των ίδιων σημείων (και πάλι σε γραμμές) αφού έχουν μετασχηματισθεί μέσω ενός affine μετασχηματισμού, οποίος υπολογίζεται με έναν πίνακα γραμμικού μετασχηματισμού  $A$  και ένα διάνυσμα μετατόπισης  $B$ . Με άλλα λόγια

```
p2(i,:) = A*p1(i,:) + B για i=1, 2, 3
```

Η συνάρτηση υπολογίζει τον πίνακα  $A$  και το διάνυσμα  $B$ . Στην αναφορά σας, παρουσιάστε μαθηματικά τη λύση την οποία θα υλοποιήσετε. Η λύση θα πρέπει να δείχνει και τον λόγο για τον οποίο απαιτούνται ακριβώς 3 σημεία. Υλοποιήσεις οι οποίες δεν περιλαμβάνουν την μαθηματική λύση δεν θα γίνουν δεκτές.

Μπορείτε τώρα να κατασκευάσετε την συνάρτηση

```
[A,B]=myransac(intPoints1,intPoints2,R)
```

η οποία υλοποιεί έναν RANSAC αλγόριθμο για να βρει τον affine μετασχηματισμό ο οποίος μετασχηματίζει μία εικόνα  $I_1$  σε μία άλλη εικόνα  $I_2$ , όπου  $intPoints1$  και  $intPoints2$  είναι τα σημεία ενδιαφέροντος της πρώτης και δεύτερης εικόνας αντίστοιχα, όπως έχουν βρεθεί από την `myDetectHarrisFeatures`, και  $R$  είναι ο πίνακας που υπολογογίζει η `possiblePairs` για τα ίδια σημεία. Για να βρεθεί ο μετασχηματισμός, κανονικά πρέπει να εξεταστούν εξαντλητικά όλες οι πιθανές αντιστοιχίες σημείων της πρώτης εικόνας στη δεύτερη. Η εξαντλητική αναζήτηση ωστόσο είναι ιδιαίτερα χρονοβόρα. Το πρόβλημα αυτό λύνει ο αλγόριθμος τύπου RANSAC που θα υλοποιήσετε, ο οποίος περιγράφεται στην ακόλουθη παράγραφο.

Ο αλγόριθμος εφαρμόζει επαναληπτικά μία διαδικασία η οποία περιλαμβάνει ένα στάδιο τυχαίας επιλογής, και στο τέλος της διαδικασίας προκύπτει μία πιθανή λύση. Εάν η διαδικασία επαναληφθεί αρκετές φορές, είναι πρακτικά αδύνατο να μην βρεθεί η βέλτιστη λύση. Συγκεκριμένα, ο δικός σας αλγόριθμος θα πρέπει να υλοποιεί τα παρακάτω βήματα.

1. Επιλογή 3 σημείων από την πρώτη εικόνα και 3 σημείων από τη δεύτερη εικόνα. Η επιλογή μπορεί να γίνει εύκολα με κατάλληλο χειρισμό του πίνακα R που υπολογίζει η `possiblePairs`. Η επιλογή των 3 σημείων από κάθε εικόνα θεωρεί ότι το πρώτο σημείο της δεύτερης εικόνας είναι το πρώτο μετασχηματισμένο της πρώτης, κλπ. Η επιλογή γίνεται με τυχαίο τρόπο, ώστε να υπάρχει διαφοροποίηση σε κάθε επανάληψη. Επίσης, δεν επιτρέπεται να επιλεγθεί 2 ή 3 φορές το ίδιο σημείο μίας εικόνας σε κάθε επανάληψη (περιγράψτε στην αναφορά τι πρόβλημα δημιουργείται αν συμβεί κάτι τέτοιο).
2. Δεδομένων των 3 και 3 σημείων, υπολογίζονται οι πίνακες μετασχηματισμού  $A_i$  και  $B_i$ , όπου  $i$  είναι ο Α.Α. της επανάληψης.
3. Μετασχηματίζονται όλα τα σημεία ενδιαφέροντος της πρώτης εικόνας με χρήση των  $A_i$  και  $B_i$ .
4. Καταμετρούνται πόσα μετασχηματισμένα σημεία ενδιαφέροντος της πρώτης εικόνας έχουν βρεθεί ικανά κοντά σε σημεία ενδιαφέροντος της δεύτερης εικόνας (έστω  $k_i$ ). Το κριτήριο ικανά κοντά υλοποιείται με ανωφλίωση της ευκλείδειας απόστασης των σημείων. Επίσης, δεν είναι αποδεκτό δύο σημεία της μίας εικόνας να είναι ικανά κοντά στη ίδιο σημείο της άλλης εικόνας.

Τα παραπάνω βήματα πρέπει να εκτελεσθούν επαναληπτικά αρκετές φορές (πχ 1000). Μετά το πέρας των επαναλήψεων, επιλέγονται οι πίνακες  $A_n$  και  $B_n$  για τους οποίους το  $k_n$  είναι μέγιστο στο σύνολο των  $k_i$  που υπολογίστηκαν στις επαναλήψεις. Σε περίπτωση που δεν υπάρχει μοναδικό μέγιστο, θα πρέπει να κάνετε κάποιον επιπλέον χειρισμό, τον οποίο θα πρέπει αναλυτικά να αιτιολογήσετε στην αναφορά σας.

### 3.2 Segment stitching

Μία εικόνα περιγράφεται από έναν πίνακα δύο διαστάσεων, θεωρώντας ότι γνωρίζουμε τα δείγματα της εικόνας σε ακέραιες θέσεις. Εναλλακτικά, η εικόνα μπορεί να περιγραφεί ως ένας πίνακας 3 στηλών, όπου κάθε γραμμή αντιστοιχεί σε ένα σημείο/δείγμα της εικόνας, και οι τρεις στήλες περιέχουν τις 2 συντεταγμένες και την τιμή της φωτεινότητας του σημείου. Η εναλλακτική αυτή αναπαράσταση επιτρέπει την ύπαρξη μη-ακέραιων συντεταγμένων.

Κατά τη διαδικασία συνένωσης ενός χωρίου σε μία εικόνα, οι γνωστές φωτεινότητες των σημείων του χωρίου στις μετασχηματισμένες θέσεις (ώστε να ταιριάζουν με την εικόνα) δεν θα είναι γνωστές σε ακέραιες θέσεις. Για τον λόγο αυτό, θα χρειαστεί να κατασκευάσετε κάποιου είδους παρεμβολή (πχ γραμμική) δύο διαστάσεων. Δεν επιτρέπεται να χρησιμοποιήσετε έτοιμες συναρτήσεις όπως η `interp1`.

Κατασκευάστε την συνάρτηση

```
canvas=stitch(canvas,image,segment)
```

η οποία λαμβάνει ως είσοδο μία εικόνα `image` και ένα χωρίο `segment`, υπολογίζει την σχετική θέση του χωρίου ως προς την εικόνα, και βάφει το αποτέλεσμα στη μεταβλητή `canvas`, την οποία και επιστρέφει.

Τέλος, κατασκευάστε το script `demo3a` το οποίο επιδεικνύει την μεθοδολογία που έχει περιγραφεί στις εικόνες που περιλαμβάνονται στο αρχείο `set_1.zip`. Συγκεκριμένα, συνενώνει τα 3 χωρία που δίνονται ως προς την πρωτότυπη εικόνα, και κατασκευάζει μία δεύτερη εικόνα η οποία έχει προκύψει από την κατάλληλη τοποθέτηση των 3 χωρίων σε ένα άσπρο καμβά.

## 4 Εφαρμογή σε πραγματικό παράδειγμα

Στη συνέχεια, κατασκευάστε το script `demo3b`, το οποίο εφαρμόζει την ίδια μεθοδολογία για το ζευγάρι πραγματικών εικόνων που δίνεται στο αρχείο `set_2.zip`. Οι βασικές διαφορές με το προηγούμενο ερώτημα είναι οι ακόλουθες.

Δεν υπάρχει διαθέσιμη η αρχική εικόνα. Συνεπώς, οι διαστάσεις του καμβά θα πρέπει να είναι αρκετά μεγάλες ώστε να χωράνε ολόκληρο το αποτέλεσμα. Επίσης, δεν υπάρχει κάποιο παγκόσμιο σύστημα αναφοράς για να ευθυγραμμιστούν απόλυτα τα χωρία.

Λόγω προοπτικής, δεν είναι δυνατό να συμπέσουν απόλυτα τα σημεία ενδιαφέροντος κάθε εικόνας. Δεν υπάρχει δηλαδή κάποιος *affine* μετασχηματισμός βάσει του οποίου έχουν μετασχηματιστεί τα κοινά μέρη των εικόνων, αλλά ένας προοπτικός μετασχηματισμός στον χώρο των 3 διαστάσεων. Έτσι, ποτέ δεν θα συμπέσουν ακριβώς τα σημεία ενδιαφέροντος των δύο εικόνων. Ωστόσο, αγνοήστε την παραπάνω πρόταση, ρυθμίζοντας κατάλληλα τα ανώφλια που χρησιμοποιείτε.

## 5 Παραδοτέα

Παραδώστε μία αναφορά με τις περιγραφές και τα συμπεράσματα που σας ζητούνται στην εκφώνηση. Έμφαση θα πρέπει να δοθεί στη θεωρητική υποστήριξη των αποτελεσμάτων σας, και όχι στην προγραμματιστική. Επίσης, ο κώδικας θα πρέπει να είναι σχολιασμένος ώστε να είναι κατανοητό τι ακριβώς λειτουργία επιτελεί (σε θεωρητικό επίπεδο, όχι σε επίπεδο κλίσης συναρτήσεων).

Επίσης:

- Υποβάλετε ένα και μόνο αρχείο, τύπου `zip`.
- Το όνομα του αρχείου πρέπει να είναι `AEM1.zip`, όπου `AEM1` είναι το `AEM` του φοιτητή της ομάδας.
- Το προς υποβολή αρχείο πρέπει να περιέχει τα αρχεία κώδικα `Matlab` και το αρχείο `report.pdf` το οποίο θα είναι η αναφορά της εργασίας.
- Η αναφορά πρέπει να είναι ένα αρχείο τύπου `PDF`, και να έχει όνομα `report.pdf`.
- Μην υποβάλετε αρχεία που δεν χρειάζονται για την λειτουργία του κώδικά σας, ή φακέλους/αρχεία που δημιουργεί το λειτουργικό σας, πχ ```Thumbs.db```, ```.DS_Store```, ```.directory```.
- Για την ονομασία των αρχείων, φακέλων, κλπ που περιέχονται στο προς υποβολή αρχείο, χρησιμοποιείτε μόνο αγγλικούς χαρακτήρες, και όχι ελληνικούς ή άλλα σύμβολα, πχ ```#```, ```$```, ```%``` κλπ.
- Όλα τα αρχεία κώδικα πρέπει να είναι αρχεία κειμένου τύπου `UTF-8`.

## Αναφορές

- [1] C. Harris and M. Stephens, *A Combined Corner and Edge Detector*, Proceedings of the 4th Alvey Vision Conference, August 1988, pp. 147-151
- [2] H. Bay, T. Tuytelaars, L. Van Gool, *SURF: Speeded Up Robust Features*, Computer Vision – ECCV 2006, Vol. 3951 of the series Lecture Notes in Computer Science pp 404-417