

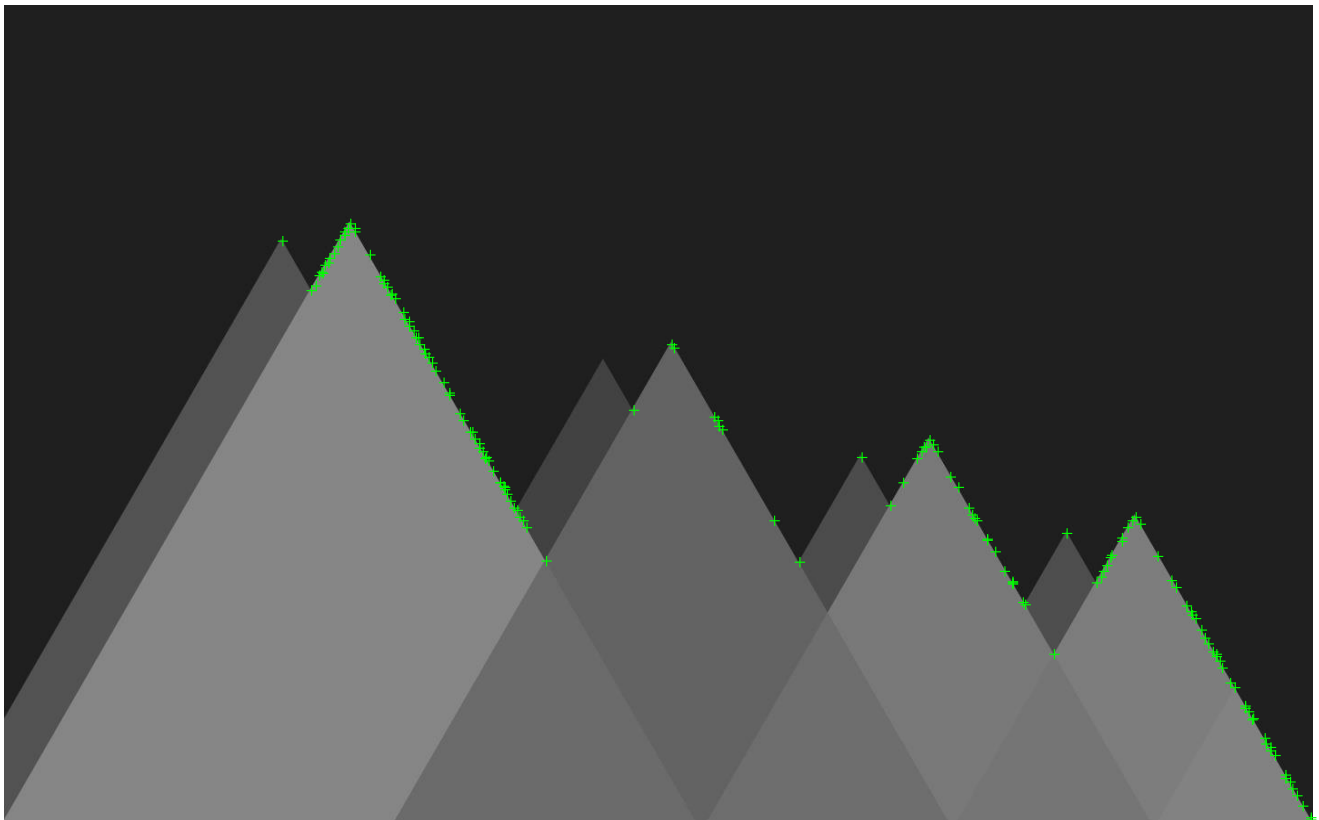
ΨΗΦΙΑΚΗ ΕΠΕΞΕΡΓΑΣΙΑ ΕΙΚΟΝΑΣ (ΕΡΓΑΣΙΑ 3)

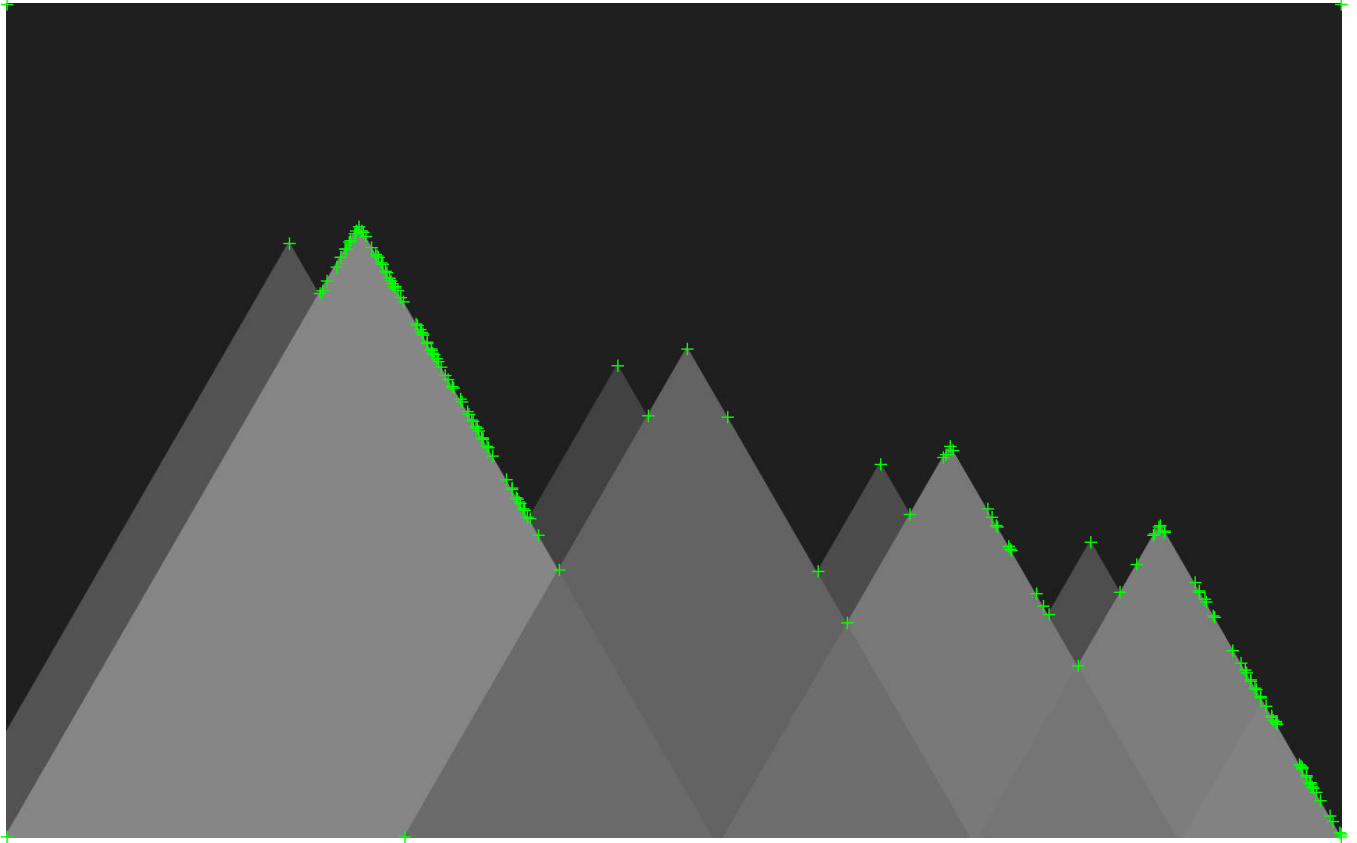
**ΟΝ/ΜΟ: ΝΗΡΑΣ ΔΗΜΗΤΡΗΣ
ΑΕΜ: 8057
email: diminira@auth.gr**

1.Harris Corner Detector

Στην ενότητα αυτή ζητήθηκε η υλοποίηση του αλγορίθμου Harris Corner Detector, σύμφωνα με το άρθρο 1 το οποίο παρατίθεται στην εκφώνηση, με σκοπό τον εντοπισμό σημείων ενδιαφέροντος. Η ανάπτυξη του αλγορίθμου έγινε ακολουθώντας κατά βάση τον αλγόριθμο που δίνεται στο άρθρο. Με σκοπό τη συγκράτηση μόνο των σημείων ενδιαφέροντος, χρησιμοποιήθηκε ένα threshold, το οποίο έπειτα από δοκιμές θέσαμε ίσο με 420 ούτως ώστε τα σημεία ενδιαφέροντος που προκύπτουν από την δική μας υλοποίηση να είναι περίπου ίσα με αυτά της έτοιμης συνάρτησης του MATLAB. Παρακάτω φαίνεται το αποτέλεσμα της δική μας συνάρτησης και της έτοιμης συνάρτησης του MATLAB εφαρμοζόμενες στην εικόνα originalImage.jpg.

detectHarrisFeatures





Όπως βλέπουμε η συνάρτησή μας έχει ανακαλύψει όλα τα σημεία ενδιαφέροντος, όπως έγινε και με την έτοιμη συνάρτηση, προσθέτοντας και μερικά ακόμη.

2. Surf Descriptor

Δεν υπήρξε υλοποίηση αυτής της συνάρτησης, επομένως χρησιμοποιήσαμε την έτοιμη συνάρτηση του MATLAB, `extractFeatures(image, intPoints)`

3.1. Αλγόριθμος RANSAC

Στην ενότητα αυτή ζητήθηκε η ανάπτυξη 3 συναρτήσεων της `possiblePairs(features1, features2)`, της `findAffineTransform(p1, p2)` και της `myransac(intPoints1, intPoints2, R)`.

Η συνάρτηση `possiblePairs(features1, features2)` δέχεται ως είσοδο δύο πίνακες που περιέχουν τους `descriptors(future vectors)` κάθε σημείου της εικόνας ως γραμμές, οι οποίο προκύπτουν εφαρμόζοντας τη συνάρτηση `extractFeatures`. Η συνάρτηση επιστρέφει έναν πίνακα δύο στηλών, με τιμές δείκτες στις γραμμές των

δύο πινάκων. Για την απόφαση εάν δύο σημεία μοιάζουν αρκετά χρησιμοποιήθηκε το μέσο τετραγωνικό σφάλμα των descriptor τους εάν είναι κάτω από ένα όριο (threshold) το οποίο έχει οριστεί ίσο με 10^3 . Ο αλγόριθμος σαρώνει όλες τις σειρές του feature vector της πρώτης εικόνας και αντίστοιχα όλες της δεύτερης εικόνας για να ανακαλύψει κάθε πιθανή ομοιότητα.

Η συνάρτηση `findAffineTransform(p1,p2)` λαμβάνει ως είσοδο δύο πίνακες διαστάσεων 3×2 , τον `p1` ο οποίος περιέχει τις συντεταγμένες 3 σημείων (ως γραμμές) και τον `p2` ο οποίος 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 ως εξής:

$$p2(i,:) = A * p1(i,:) + B \text{ για } i=1, 2, 3 \quad (1)$$

και επιστρέφει τον πίνακα A και το διάνυσμα B .

Έστω ο πίνακας $A = \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \alpha_{21} & \alpha_{22} \end{bmatrix}$ και το διάνυσμα $B = \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix}$. Παρατηρούμε πως έχουμε 6 αγνώστους επομένως θα χρειαστούμε 6 εξισώσεις για να λύσουμε το σύστημα. Οι πίνακες `p1` και `p2` είναι 3×2 πίνακες έχοντας συνολικά 3 ζεύγη συντεταγμένων.

Αναλύοντας την εξίσωση (1) έχουμε:

$$p2(1,1) = \alpha_{11} * p1(1,1) + \alpha_{12} * p1(1,2) + \beta_1$$

$$p2(1,2) = \alpha_{21} * p1(1,1) + \alpha_{22} * p1(1,2) + \beta_2$$

$$p2(2,1) = \alpha_{11} * p1(2,1) + \alpha_{12} * p1(2,2) + \beta_1$$

$$p2(2,2) = \alpha_{21} * p1(2,1) + \alpha_{22} * p1(2,2) + \beta_2$$

$$p2(3,1) = \alpha_{11} * p1(3,1) + \alpha_{12} * p1(3,2) + \beta_1$$

$$p2(3,2) = \alpha_{21} * p1(3,1) + \alpha_{22} * p1(3,2) + \beta_2$$

Βλέπουμε λοιπόν πως προέκυψαν 6 εξισώσεις με τις οποίες δεδομένων των `p1` και `p2` μπορούμε να βρούμε τον πίνακα A και το διάνυσμα B . Κατά την ανάλυση αυτή βλέπουμε τη σημαντικότητα του να υπάρχουν ακριβώς 3 σημεία, καθώς αν είχαμε λιγότερα δε θα μπορούσαμε να λύσουμε το σύστημα, καθώς δεν θα υπήρχαν επαρκείς εξισώσεις, ενώ αν είχαμε περισσότερα μπορεί να προκύπταν περισσότερες από μία λύσεις.

Στη συνέχεια υλοποιήθηκε η συνάρτηση `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 υπολογίζει η `possiblePairs` για τα ίδια σημεία. Ο αλγόριθμος ο οποίος περιγράφεται στην εκφώνηση εφαρμόστηκε επαναληπτικά για 1000 φορές, ούτως ώστε να έχουμε τη μεγαλύτερη δυνατή ακρίβεια που μπορούμε να επιτύχουμε. Η συνάρτηση δίνει ως έξοδο τον βέλτιστο πίνακα A και το βέλτιστο διάνυσμα B , τα οποία έχουν μετασχηματίσει την εικόνα. Για την καταμέτρηση των μετασχηματισμένων σημείων ενδιαφέροντος της πρώτης εικόνας τα οποία έχουν βρεθεί ικανά κοντά σε σημεία ενδιαφέροντος της δεύτερης χρησιμοποιήθηκε το μέσο

τετραγωνικό σφάλμα των σημείων και ένα ανώφλι (threshold) το οποίο τέθηκε ίσο με 10^{-2} . Δε λήφθηκε μέριμνα βέβαια για την περίπτωση στην οποία θα έχουμε 2 ή παραπάνω μέγιστα k .