ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ

ΚΑΙ ΜΗΧΑΝΙΚΩΝ Η/Υ

Ψηφιακή Επεξεργασία Εικόνας

-Εργασία 2-

Hough, Harris, Rot και αποκοπή εικόνων

Ονοματεπώνυμο : ΝΑΣΤΟΣ ΒΙΚΤΩΡ

ΑΕΜ : 9297

Email : viktorna@auth.gr

ΥΠΕΥΘΥΝΟΣ ΚΑΘΗΓΗΤΗΣ

ΝΤΕΛΟΠΟΥΛΟΣ ΑΝΑΣΤΑΣΙΟΣ

ΘΕΣΣΑΛΟΝΙΚΗ , ΙΟΥΝΙΟΣ 2020

➢ Γενική περιγραφή εργασίας

Η παρούσα εργασία αποτελείται από δύο ενότητες. Στην πρώτη ενότητα καλούμαστε να υλοποιήσουμε μια σουίτα από ρουτίνες τις οποίες θα πρέπει να χρησιμοποιήσουμε για να δώσουμε λύση στο τελικό πρόβλημα(ενότητα 2). Η πρώτη ενότητα απαιτεί την υλοποίηση των :

- Μετασχηματισμός Hough
- Harris corner detector
- Περιστροφή εικόνας

➢ 1.1 Hough Transform

Σε αυτό το κομμάτι της εργασίας, καλούμαστε να υλοποιήσουμε τον μετασχηματισμό Hough, και πιο συγκεκριμένα τη συνάρτηση :

$$function\ [H, L, res] = myHoughTransform(img\_binary, Drho, Dtheta, n)$$

η οποία λαμβάνει ως είσοδο μία binary εικόνα *img_binary* η οποία έχει προέλθει από κατωφλίωση της εξόδου ενός edge detector για μία grayscale εικόνα, την διακριτότητα *Drho* στην διάσταση ρ στο πεδίο του Hough μετρούμενη σε pixels και την διακριτότητα *Dtheta* του θ στο πεδίο του Hough μετρούμενη σε rads.

Η συνάρτηση επιστρέφει τον πίνακα μετασχηματισμού Hough H και τον πίνακα **L** με τις παραμέτρους *rho* και *theta* των *n* ισχυρότερων ευθειών, των ευθειών δηλαδή που αντιστοιχούν στα *n* μεγαλύτερα τοπικά μέγιστα του πίνακα **H**. Η συνάρτηση επιστρέφει επίσης το πλήθος *res* των σημείων της εικόνας εισόδου που δεν ανήκουν στις n ευθείες που έχουν εντοπιστεί.

Κώδικας Matlab

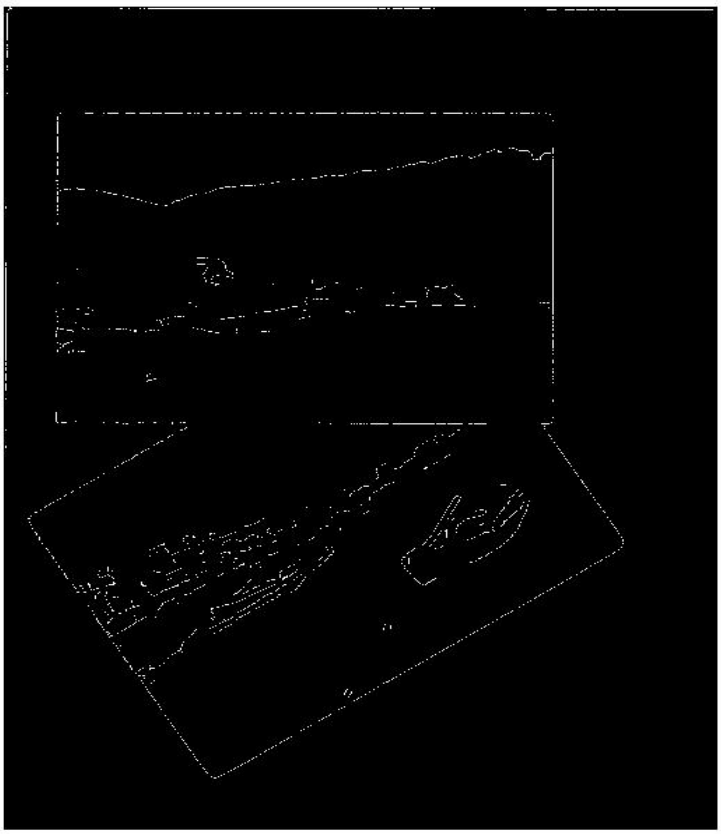
Η εικόνα του αρχείου deliverable_1.m , πριν τοποθετηθεί σαν όρισμα στη συνάρτηση που αναφέρθηκε

παραπάνω, υπέστη κάποιες τροποποιήσεις οι οποίες είναι οι εξής :

- Αλλαγή μεγέθους μέσω της συνάρτησης imresize().
- Εφαρμογή Gaussian φίλτρου το οποίο ρυθμίζεται μέσω του sigma και καθιστά την εικόνα πιο smooth.

$$imgaussfilt(image, sigma)$$

- Η χρήση του 'Sobel' edge detector, το οποίο προκαλεί αλλαγές στην ένταση της εικόνας. Το αποτέλεσμα φαίνεται παρακάτω :



Εικόνα 1. Edge detector : Sobel

Στη συνέχεια ακολούθησε η υλοποίηση της συνάρτησης **myHoughTransform**() η οποία αποτελείται από τρία βασικά κομμάτια :

1. Hough transform(πίνακας μετασχηματισμού H).
2. Hough peaks (κορυφές της εικόνας αποθηκευμένες σε έναν πίνακα

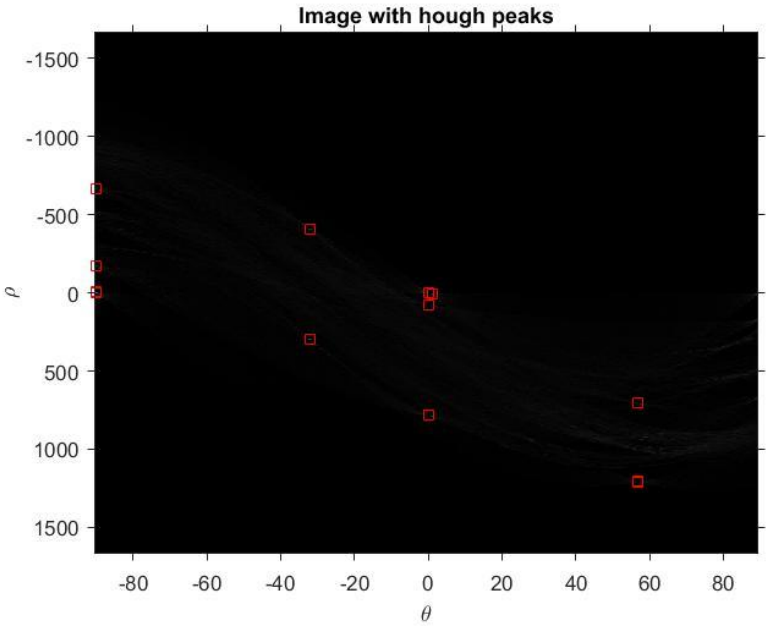
$$L = [rhos, thetas]$$

).

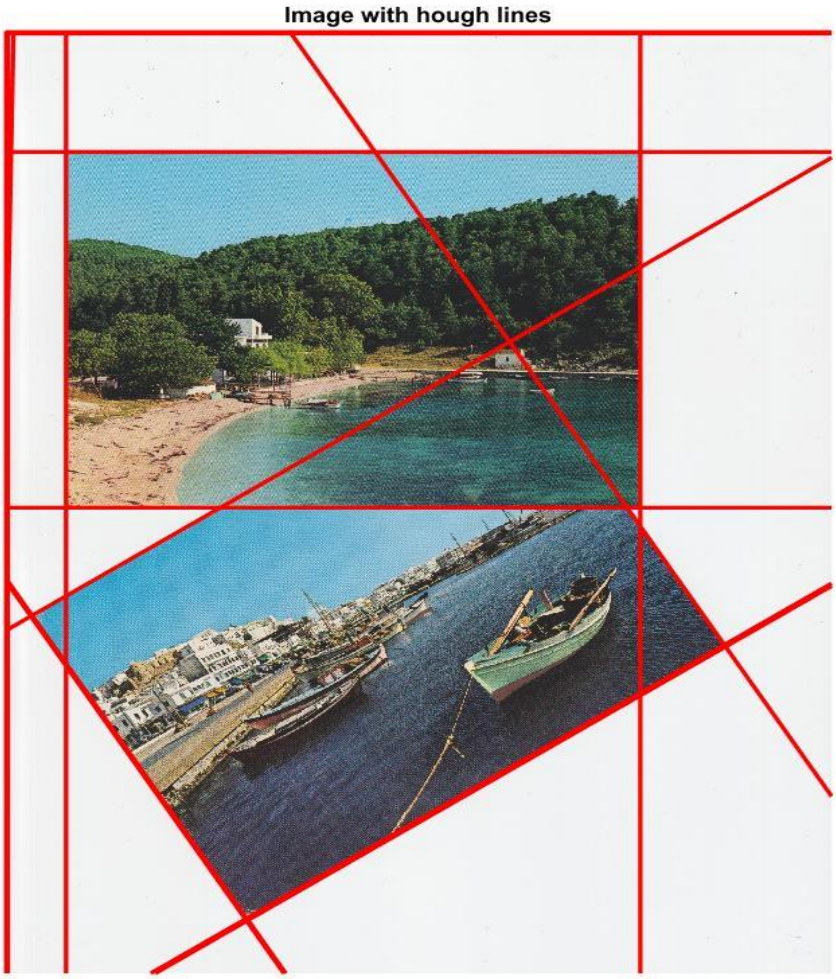
3. Hough lines (Οι ευθείες που εντοπίστηκαν στην εικόνα εισόδου).
   Η βασική θεωρητική θέση του Μετασχηματισμού Γραμμών Hough είναι πως κάθε σημείο – pixel σε μία δυαδική εικόνα μπορεί να είναι τμήμα κάποιας γραμμής. Μία γραμμή στο χώρο της εικόνας μπορεί να εκφραστεί με δύο μεταβλητές είτε στο Καρτεσιανό σύστημα αξόνων (Cartesian Coordinate System) είτε στο Πολικό σύστημα αξόνων (Polar Coordinate System).

Αποτελέσματα Matlab

Παρακάτω φαίνονται τα αποτελέσματα του κώδικα Matlab για n = 15 και sigma = 3.75 . Παρατηρώ πως για n = 15 έχω τη βέλτιστη λύση.



Εικόνα 2. Hough Peaks



Εικόνα 3. Hough Lines

➢ 1.2 Harris corner detector

Στο συγκεκριμένο κομμάτι της εργασίας καλούμαστε να υλοποιήσουμε τη συνάρτηση

$$function\ corners\ =\ myDetectHarrisFeatures(I)$$

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.

Η συνάρτηση επιστρέφει τις γωνίες που εντοπίστηκαν πάνω στην grayscale εικόνα εισόδου μέσω ενός κόκκινου τετραγώνου 5 × 5 pixels με κέντρο την κάθε εντοπισμένη κορυφή.

Η εικόνα εισόδου του κώδικα Matlab, τροποποιεί και πάλι το μέγεθος της μέσω της συνάρτησης imresize(). Στη συνέχεια υλοποιείται η συνάρτηση **myDetectHarrisFeatures()**, η οποία εφαρμόζει κατάλληλη μάσκα για τον υπολογισμό των μερικών παραγώγων και εκτυπώνει τις επιθυμητές γωνίες, των οποίων ο αριθμός εξαρτάται από το όριο threshold. Όσο το threshold μειώνεται, ο αριθμός των επιστρεφόμενων γωνιών αυξάνεται.

Αποτελέσματα Matlab

Παρακάτω φαίνεται η εκτύπωση της εικόνας εισόδου για threshold = 0.01 .

Εικόνα 4. Harris corner detector

➢ 1.3 Rotation

Σε αυτό το κομμάτι της εργασίας καλούμαστε να υλοποιήσουμε τη συνάρτηση

$$function\ rotImg\ =\ myImgRotation(img, angle)$$

η οποία θα λαμβάνει σαν είσοδο μία εικόνα img και θα την περιστρέφει αντίστροφα από την φορά του ρολογιού κατά γωνία angle σε rads.

Η εικόνα εισόδου του κώδικα Matlab, τροποποιεί για άλλη μια φορά το μέγεθος της μέσω της συνάρτησης imresize(). Στη συνέχεια υλοποιείται η συνάρτηση **myImgRotation()**, η οποία εκτυπώνει την περιστρεφόμενη εικόνα, αναλόγως του ορίσματος που δίνεται στην κλήση της.

<u>Αποτελέσματα Matlab</u>

Παρακάτω φαίνονται τα αποτελέσματα για γωνίες περιστροφής $54° \ x \ pi/180° \ rads$ και $213° \ x \ pi/180° \ rads$ αντίστοιχα.

Εικόνα 5. Rotate Image/ 54° x pi/180° rads

Εικόνα 6. Rotate Image/ 213° x pi/180° rads