



ΑΡΙΣΤΟΤΕΛΕΙΟ
ΠΑΝΕΠΙΣΤΗΜΙΟ
ΘΕΣΣΑΛΟΝΙΚΗΣ

8^ο Εξάμηνο

«Ψηφιακή Επεξεργασία Εικόνας»

Ακαδ. έτος 2021-22 | Ημ.Παράδοσης 10/5/2022

1^η Εργασία : «Salient Points»

του Σταύρου Βασίλειου Μπουλιόπουλου 9671

Διδάσκων θεωρίας: Αναστάσιος Ντελόπουλος

Επιβλέπων εργασίας: Λεωνίδας Αλαγιαλόγλου

Περιεχόμενα

| | |
|--|------------------------------|
| 1.Εισαγωγή | 1 |
| 2. Εκτέλεση των αρχείων κώδικα και προβλήματα που αντιμετώπισα ... | Error! Bookmark not defined. |
| 3. Παρουσίαση δομής κώδικα και αποτελεσμάτων | 2 |

1.Εισαγωγή

Στα πλαίσια της πρώτης εργασίας του μαθήματος μας ζητήθηκε να υλοποιήσουμε αλγορίθμους προκειμένου να

κάνουμε match στοιχεία βάσει κάποιων αλγορίθμων και να κάνουμε **2 εικόνες stitch** σε ένα κοινό κανβά/πανόραμα.

2. Εκτέλεση των αρχείων κώδικα και προβλήματα που αντιμετώπισα

Τα **αρχεία προς εκτέλεση** είναι τα **“rotation1_1.m”, “descriptor1_2.m” και το “harrisCd1_3.m”**

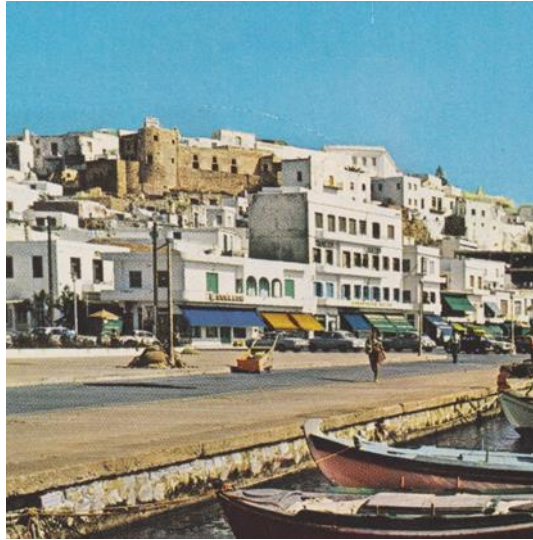
Δεν χρησιμοποιήθηκε η λογική της προτεινόμενης συνάρτησης **isCorner** της εκφώνησης και **δεν** υλοποιήθηκε ύστερα από προσπάθειες και δεν εκτελούνται σωστά και αποτελεσματικά οι σχετικοί κώδικες για το **2^ο κεφάλαιο** της εκφώνησης και του εκτελέσιμου αρχείου “stitch2.m”. Οπότε δεν θα γίνει αναφορά στο 2^ο κεφάλαιο αν και άφησα κάποιες σχετικές συναρτήσεις που είχα προσπαθήσει να γράψω.

3. Παρουσίαση δομής κώδικα και αποτελεσμάτων

Αρχικά, έπρεπε αρχικά να προσέξω και να μελετήσω την εισαγωγή και εξαγωγή/προβολή εικόνας, να μελετήσω τους αλγόριθμους της εργασίας σε άρθρα ή papers ή στο Youtube ή σε άλλες πηγές, ώστε να αφομοιώσω τις λογικές πίσω από αυτούς. Έτσι λοιπόν, ξεκινώντας με την πρώτη υπόενοτητα για την **περιστροφή εικόνας** ακολούθησα την εξής λογική :

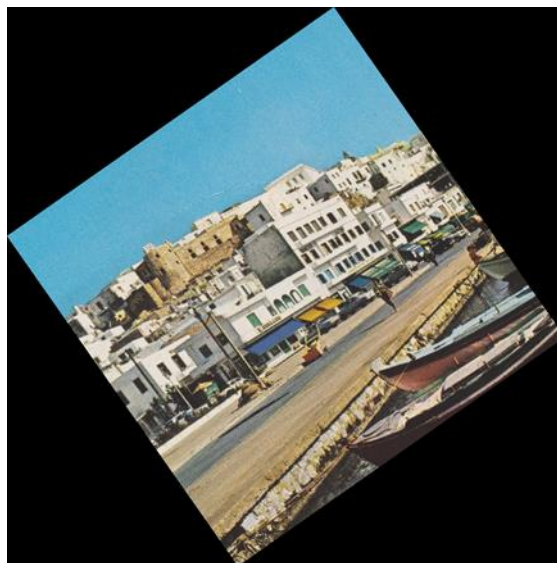
- i) Εισάγουμε μια εικόνα και τις μοίρες που θέλουμε να την περιστρέψουμε
- ii) Θέτω τον πίνακα A 3x3 περιστροφής
- iii) Ελέγχω αν η εικόνα είναι RGB(3D) ή gray scale(1D) για να χειριστώ αναλόγως αργότερα τον υπολογισμό του μέσου βάρους από γειτονικά pixel σε 1 ή και στα 3 χρώματα.
- iv) Φέρνω την αρχή στο κέντρο του πίνακα
- v) Βάσει του κέντρου αυτού και του πίνακα A υπολογίζω τους μετασχηματισμούς των σημείων

- vi) Κατασκευάζω την εικόνα με μαύρο φόντο και τοποθετώ όσα σημεία έχουν υπολογιστεί
- vii) Γεμίζω τα κενά της εικόνας με φίλτρο διαμέσου βάσει των γειτονικών του pixel.

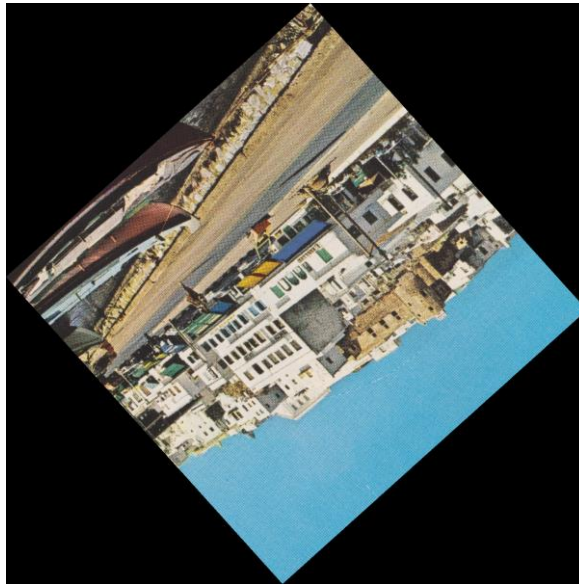


Εικόνα 1: Εικόνα προς δοκιμή

Με είσοδο της παραπάνω εικόνας πήραμε τις παρακάτω 2 εικόνες.



Εικόνα 2: Αντιωρολογιακή περιστροφή κατά 35 μοίρες



Εικόνα 3: Αντιωρολογιακή περιστροφή κατά 222 μοίρες

Στην δεύτερη υποενότητα υλοποίησα μόνο τον **myLocalDescriptor** γιατί προσπαθούσα να καταλάβω την λογική του stitch 2 εικόνων και δεν ασχολήθηκα πραγματικά για την υλοποίηση του myLocalDescriptorUpgrade και την αναλυτικότερη έρευνα άλλων feature descriptors (GLOH, SIFT κτλπ.). Στην υλοποίηση ακολούθησα τα βήματα της εκφώνησης ως προς τις στρογγυλοποιήσεις, τις παραδοχές και τον αλγόριθμο. Τα αποτελέσματα φαίνονται με disp() εκτελώντας το **descriptor1_2.m**.

Τέλος, στην Τρίτη υποενότητα για τον Harris corner detector η λογική είναι η εξής βάσει του αλγορίθμου :

- i) Έχω τις παραμέτρους ώστε να κάνω trial-error για να δω ικανοποιητικά αποτελέσματα
- ii) Θέτω μάσκες παραγώγων
- iii) Υπολογίζω παράγωγους
- iv) Κάνω smooth την εικόνα με Gaussian Filter

v) Υπολογίζω τα στοιχεία του πίνακα M

$$M = \sum w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

vi) Υπολογίζω το μέτρο του Harris R

$$R = \det M - k(\text{trace } M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$\text{trace } M = \lambda_1 + \lambda_2$$

vii) Τέλος συγκρίνω την γειτονιά των σημείων και κάνω κατωφλίωση για να καταλήξω στα σημεία που μου δίνουν τους corners.

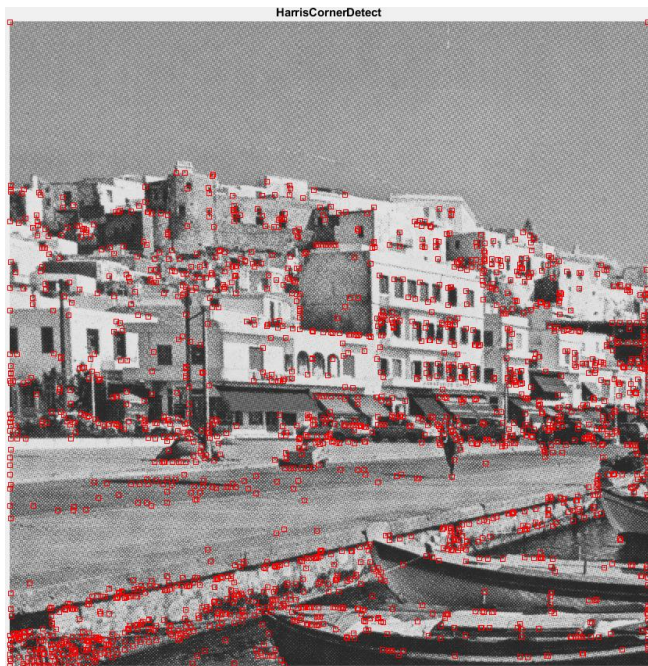
(Πηγές στον φάκελο κατάθεσης)



Εικόνα 3 Ιx οριζόντιος



Εικόνα 4 Ιγ κάθετος



Εικόνα 5 Harris Corners

