

Ψηφιακή Επεξεργασία Εικόνας

Αναφορά Εργασίας 2

Μαστραλέξη Χριστίνα Μαρία (9284)

Εισαγωγή

Στην παρούσα εργασία σκοπός μας είναι να δημιουργήσουμε ένα toolbox που αποτελείται από 3 ρουτίνες: τον μετασχηματισμό Hough, τον Harris Corner Detector και την περιστροφή εικόνας.

1.1 Hough Transform

Σύντομη Θεωρητική Ανάλυση

Στην προσπάθεια εντοπισμού των ακμών μιας εικόνας, ο μετασχηματισμός Hough ακολουθείται συχνά σαν μεθοδολογία στο ζήτημα της συνένωσης ακμών. Η ανάγκη αυτή προκύπτει μετά την κατωφλίωση της εικόνας μας, εφόσον έχει γίνει κατανοητό ότι στις περισσότερες περιπτώσεις οι ακμές που προκύπτουν από αυτή την διαδικασία είναι διακεκομμένες ή έχουν δημιουργηθεί λόγω της επιλογής κατωφλίου ή και θορύβου. Έχουμε, δηλαδή, σαν αποτέλεσμα ένα σύνολο σημείων τα οποία είναι υποψήφια σημεία ακμών της εικόνας. Ο μετασχηματισμός Hough λύνει αυτό το πρόβλημα με την δημιουργία ενός πίνακα συσσώρευσης, κάθε κελί του οποίου παραμετροποιεί μια ευθεία (ρ, θ) . Κάθε σημείο της κατωφλιωμένης εικόνας που είναι πιθανό σημείο ακμής ψηφίζει τα αντίστοιχα κελιά των ευθειών στις οποίες θα μπορούσε να ανήκει και όπως γίνεται φανερό πλέον το πρόβλημα ανάγεται στην διαδικασία εύρεσης των κελιών με τις περισσότερες ψήφους, στην εύρεση συγκεκριμένα των τοπικών μεγίστων του πίνακα Hough. Η διαδικασία αυτή που βασίζεται στο γεγονός ότι μπορούμε να παραμετροποιήσουμε μια ευθεία γνωρίζοντας τις παραμέτρους ρ και θ έχει διάφορες παραλλαγές και για άλλα είδη καμπυλών εκτός των ευθειών.

Αλγόριθμος και αποτελέσματα

Η συνάρτηση που καλούμαστε να υλοποιήσουμε είναι η:

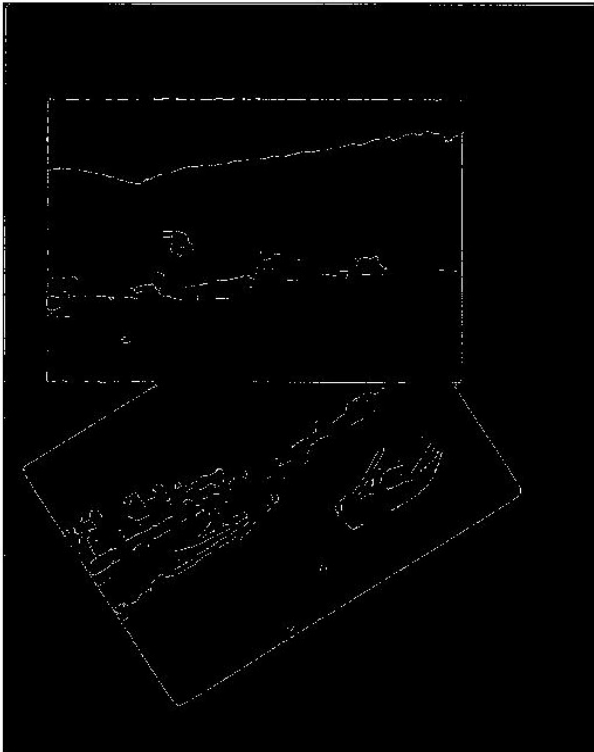
```
1 function [H, L, res] = myHoughTransform(img_binary, Drho, Dtheta, n).
```

Ως είσοδο λαβάνει μια binary εικόνα, η οποία έχει προέλθει από κατωφλίωση της εξόδου του edge detector, τις διακριτότητες Drho και Dtheta, στην διάσταση ρ στο πεδίο του Hough μετρούμενη σε pixels και του θ στο πεδίο του Hough μετρούμενη σε rads αντίστοιχα. Τέλος, n είναι το πλήθος των ισχυρότερων ευθειών που θέλουμε να εξάγουμε, δηλαδή τα n τοπικά μέγιστα του πίνακα H. Η συνάρτηση επιστρέφει τον πίνακα συσσώρευσης H, τον πίνακα L με τις παραμέτρους (ρ, θ) για κάθε μια από τις n ευθείες, και το πλήθος res των σημείων της εικόνας εισόδου που δεν ανήκουν στις n ευθείες που έχουν εντοπιστεί. Η πρακτική χρήση της μεταβλητής res είναι να μπορούμε να ελέγξουμε κατά πόσο το n που έχουμε χρησιμοποιήσει βρίσκει ένα σημαντικό ποσοστό ευθειών στην εικόνα. Για παράδειγμα, αν το res πέσει πολύ χαμηλά σαν ποσοστό, άρα έχουν βρεθεί οι περισσότερες ευθείες, δεν θα είχε κάποιο νόημα να αυξήσουμε περεταίρω το n.

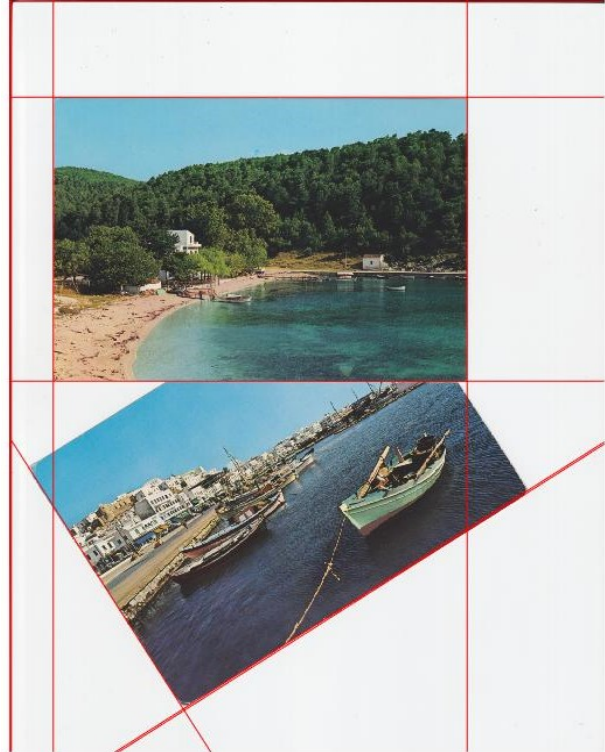
Η binary εικόνα προκύπτει από την αρχική που γίνεται grayscale και τροφοδοτείται στον sobel edge detector του matlab, ο οποίος χρησιμοποιήθηκε λόγω καλύτερων αποτελεσμάτων στην συγκεκριμένη περίπτωση. Για λόγους ευκολίας είναι χρήσιμο στο σημείο αυτό να σημειωθεί ότι κάνοντας downsampling στην εικόνα και χρησιμοποιώντας το imgaussfilt (smoothing filter) του matlab πετυχαίνουμε καλύτερους χρόνους και καλύτερα αποτελέσματα αλγοριθμικά. Στη συνέχεια, ορίζουμε τις διαμερίσεις του ρ και του θ . Το εύρος τιμών του θ θα είναι από -90° έως 89° , οι οποίες στα πλαίσια της εργασίας μετατρέπονται σε rad, όπως ζητά η εκφώνηση ενώ το ρ μπορεί να πάρει τιμές από ρ_{max} έως $-\rho_{max}$, όπου $\rho_{max} = \sqrt{N_1^2 + N_2^2}$ και N_1, N_2 οι διαστάσεις της εικόνας. Ο πίνακας H έχει διαστάσεις ίσες με το πλήθος των διαμερίσεων ρ και θ εφόσον κάθε κελί του αντιστοιχίζεται σε ένα (ρ, θ) ζεύγος.

Έπειτα, η binary εικόνα σκανάρεται από τον αλγόριθμο και εντοπίζονται τα λευκά pixels, δηλαδή τα σημεία που είναι υποψήφια για σημεία ακμής που υπάρχει στην εικόνα. Κάθε φορά που εντοπίζεται ένα τέτοιο σημείο ο αλγόριθμος βρίσκει τις ευθείες στις οποίες φαίνεται να ανήκει σύμφωνα με τον έλεγχο που γίνεται και ψηφίζει το αντίστοιχο κελί. Στο τέλος αυτών των ελέγχων έχουμε πλέον τον πίνακα συσσώρευσης H.

Τα τοπικά μέγιστα του πίνακα βρέθηκαν με την βοήθεια της συνάρτησης imregionalmax του matlab που επιστρέφει έναν πίνακα ιδίων διαστάσεων με τον H, ο οποίος περιέχει 1 στα σημεία τοπικών μεγίστων και 0 οπουδήποτε αλλού. Μετά από κατάλληλες πράξεις με τον πίνακα H προκύπτει ένας πίνακας ο οποίος περιέχει όλα τα τοπικά μέγιστα του H και τις τιμές τους. Έχουμε ήδη μια τιμή n η οποία εκφράζει το πλήθος των τοπικών μεγίστων που χρειάζεται να εξάγουμε από τον πίνακα που έχει πλέον προκύψει. Με μια τετριμμένη

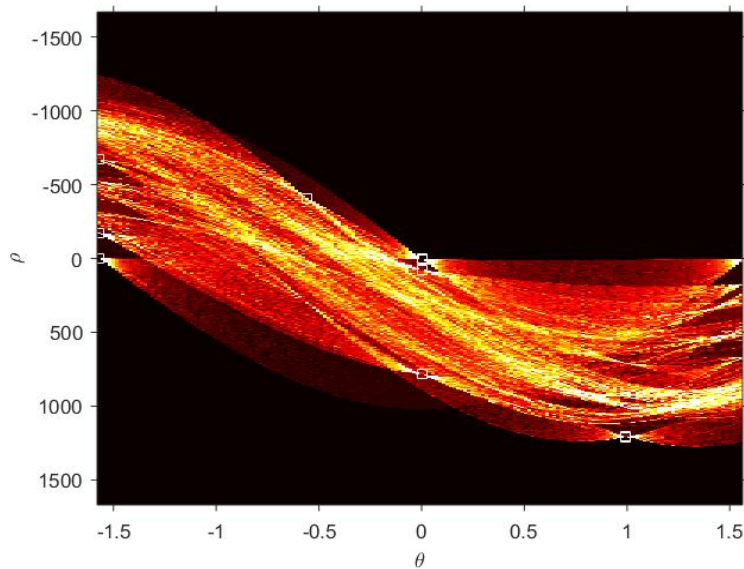


(α)



(β)

Εικόνα 1: (α) Η binary εικόνα μετά τον edge detector (β) Η αρχική εικόνα με τις ευθείες που βρέθηκαν με τον μετασχηματισμό Hough



Εικόνα 2: Ο πίνακας μετασχηματισμού και οι κορυφές(πίνακας L) που εντοπίστηκαν

επαναληπτική διαδικασία, βρίσκοντας κάθε φορά το μέγιστο και έπειτα μηδενίζοντας το έως ότου φτάσουμε να έχουμε τα επιθυμητά n στοιχεία, έχουμε πλέον έναν πίνακα peaks ο οποίος περιέχει την γραμμή και την στήλη

των τοπικών μεγίστων στον πίνακα H, και έναν πίνακα L με τις παραμέτρους (ρ,θ) που προκύπτουν.

Τα αποτελέσματα του deliverable1.m για εφαρμογή στην εικόνα im2.jpg φαίνονται στις παραπάνω εικόνες. Στην Εικόνα 1(β) έχουμε την αρχική εικόνα εισόδου και τις ευθείες που εντοπίστηκαν μετά από τον μετασχηματισμό Hough χρωματισμένες με κόκκινο χρώμα. Η συγκεκριμένη εικόνα έχει προκύψει για τις τιμές ρ και θ που αναφέρθηκαν παραπάνω και n=10. Στην Εικόνα 2 βρίσκεται την αναπαράσταση του πίνακα συσσώρευσης H και οι κορυφές στον πίνακα L. Όπως παρατηρούμε, ο αλγόριθμος εντόπισε σε πολύ ικανοποιητικό βαθμό τις ευθείες του περιγράμματος των εικόνων, αποτέλεσμα που θα φανεί ιδιαίτερα χρήσιμο στην δεύτερη ενότητα της εργασίας. Θα μπορούσαμε να αυξήσουμε το n για την ανίχνευση και όσων ευθειών του περιγράμματος των εικόνων δεν βρέθηκαν, διότι προέκυψαν αναπόφευκτα και κάποιες "διπλές" ευθείες, με κίνδυνο βέβαια να αρχίσουν να εντοπίζονται και ευθείες στο περιεχόμενο των φωτογραφιών, οι οποίες δεν είναι τόσο σημαντικές.

1.2 Harris Corner Detector

Σύντομη Θεωρητική Ανάλυση

Ο Harris Corner Detector καθορίζει αν ένα σημείο βρίσκεται σε γωνία, σε ακμή ή αν είναι ένα ομαλό σημείο. Ένα σημείο θα μπορούσε να αναγνωριστεί αν είναι γωνία κοιτώντας το από ένα μικρό παράθυρο. Μετατοπίζοντας το παράθυρο σε οποιαδήποτε κατεύθυνση θα πρέπει να δίνει μεγάλη αλλαγή στην ένταση σε τουλάχιστον δύο κατευθύνσεις. Ειδικότερα, ο Harris Corner Detector είναι βασισμένος στην τοπική συνάρτηση αυτοσυσχέτισης, η οποία περιγράφει την τοπική αλλαγή φωτεινότητας σε μία "γειτονιά" ενός σημείου. Η βασική ιδέα δίνεται από την εκφώνηση και χάριν συντομίας θα αποδοθεί πολύ συνοπτικά. Δεδομένης μιας μετατόπισης (Δx,Δy) και του σημείου (x,y) ορίζουμε:

$$c(x, y) = \sum_{x_i, y_i} [I(x_i, y_i) - I(x_i + \Delta x, y_i + \Delta y)]^2$$

, όπου I(x,y) η ένταση της. Η ένταση των σημείων του παραθύρου (x_i, y_i) εξομαλύνεται με την χρήση φίλτρου Gauss με κέντρο το (x,y): $w(x_i, y_i) = \exp(-\frac{x_i^2 + y_i^2}{2\sigma^2})$. Από το πρώτο μέρος του αναπτύγματος Taylor έχουμε:

$$c(x, y) = \sum [\Delta x \Delta y] M [\Delta x \Delta y]'$$

. Ο πίνακας M ορίζεται ως εξής: $M = \begin{bmatrix} I_x^2 & I_x \cdot I_y \\ I_x \cdot I_y & I_y^2 \end{bmatrix}$, όπου I_x και I_y είναι οι μερικοί παράγωγοι της έντασης της εικόνας στις κατευθύνσεις x και y αντίστοιχα και έχουν βρεθεί με την χρήση των μασκών που έχουν αναφερθεί στη θεωρία. Ο αλγόριθμος προτείνει τη χρήση ενός μεγέθους R που περιλαμβάνει την ορίζουσα και το ίχνος του πίνακα M. Σε ένα σημείο (x₀, y₀) ορίζεται: $R(x_0, y_0) = \det(M) - k \cdot \text{Trace}(M)$, όπου το k είναι μια θετική παράμετρος συνήθως με τιμές 0.04-0.06. Η απόκριση R παίρνει μεγάλες θετικές τιμές στα σημεία που είναι γωνίες ενώ παίρνει αρνητικές τιμές στα σημεία που είναι ακμές και τιμές περίπου ίσες με μηδέν σε σημεία που είναι σε ομαλές περιοχές.

Αλγόριθμος και αποτελέσματα

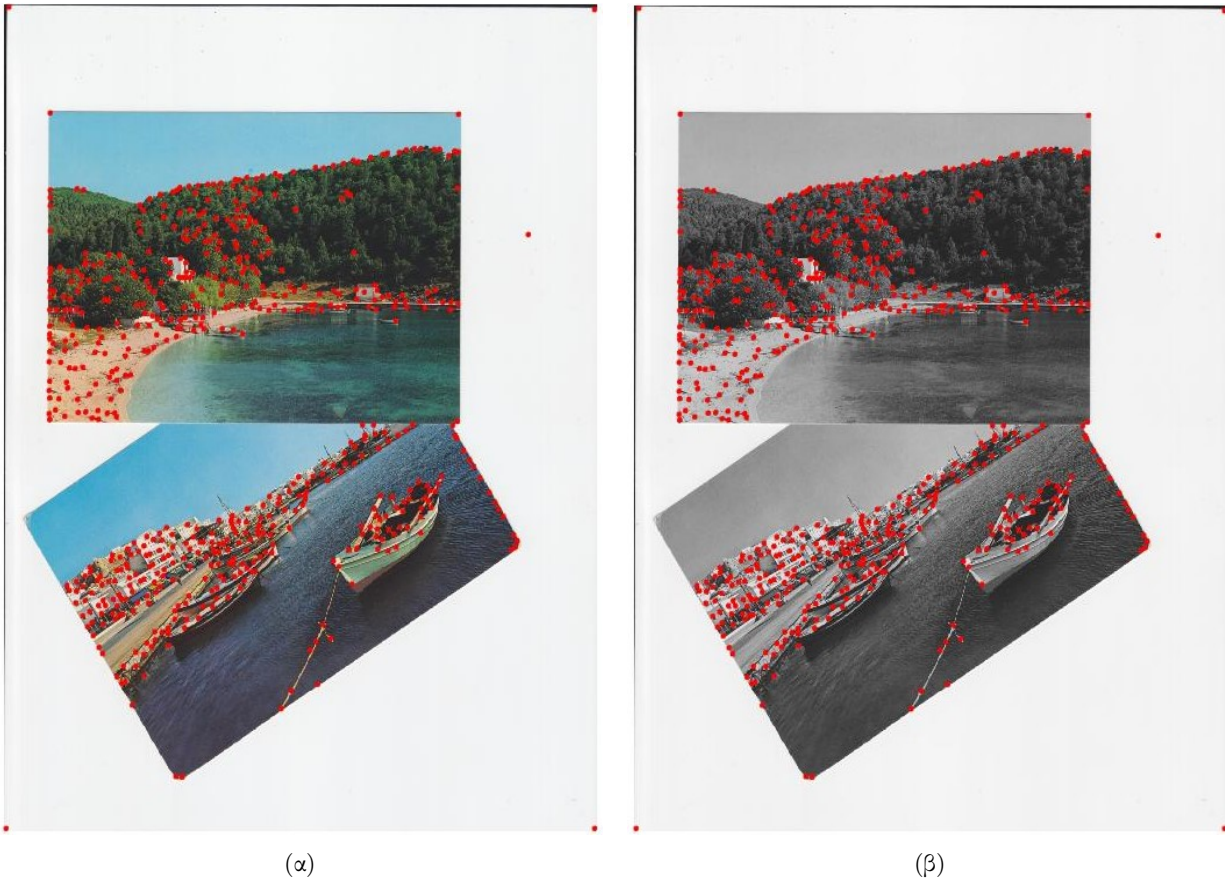
Η συνάρτηση που καλούμαστε να υλοποιήσουμε είναι η:

```
1 function corners = myDetectHarrisFeatures(I),
```

η οποία θα υλοποιεί τον αλγόριθμο όπως περιγράφηκε στην θεωρία. Παίρνει σαν είσοδο μια εικόνα grayscale με τιμές φωτεινότητας από το 0 έως το 1 και επιστρέφει τον πίνακα corners κάθε γραμμή του οποίου αντιστοιχεί στις συντεταγμένες μιας γωνίας. Συνοπτικά τα βήματα του αλγορίθμου είναι:

- Υπολογισμοί των I_x και I_y, δηλαδή των μερικών παραγώγων της έντασης της εικόνας στις κατευθύνσεις x και y αντίστοιχα με την βοήθεια των μασκών $dx = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$, $dy = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$.
- Υπολογισμοί των I_x, I_y, I_x · I_y.
- Υπολογισμοί των ορισμάτων του πίνακα M με την χρήση ενός φίλτρου εξομάλυνσης Gauss.

- Υπολογισμός της απόκριση του R .
- Εύρεση των σημείων με μεγάλη τιμή του R , μεγαλύτερη και από το όριο κατωφλίσωσης, δηλαδή ($R > threshold$).
- Εύρεση των τοπικών μεγίστων του R , τα οποία είναι και τα σημεία ενδιαφέροντος στην συγκεκριμένη περίπτωση που ψάχνουμε τις γωνίες.



Εικόνα 3: Οι γωνίες που εντοπίστηκαν στην εικόνα αποτυπωμένες πάνω (α) στην αρχική και (β) στη grayscale εκδοχή της

Τα αποτελέσματα που βλέπουμε στις παραπάνω εικόνες είναι οι γωνίες που ανιχνεύθηκαν με τον αλγόριθμο που περιγράφηκε προηγουμένως και είναι αποτυπωμένες με ένα κόκκινο τετράγωνο διαστάσεων 5×5 pixels. Παρατηρούμε ότι για τις παραμέτρους που επιλέχθηκαν στην συγκεκριμένη κλήση της συνάρτησης ($\sigma=6$, $threshold=0.003$, $order=9$), ο αλγόριθμος εντοπίζει γωνίες τόσο στην περίμετρο της σκαναρισμένης εικόνας και των εικόνων που είναι τοποθετημένες εσωτερικά όσο και μέσα στο περιεχόμενο των φωτογραφιών. Είναι προφανές ότι αλλάζοντας τις παραμέτρους αυτές θα πέρναμε ένα ελαφρώς διαφορετικό αποτέλεσμα. Για παράδειγμα, ελαττώνοντας το σ θα βρίσκαμε περισσότερες γωνίες με το ίδιο $threshold$, οπότε γίνεται αντιληπτό ότι η επιλογή των παραμέτρων αυτών είναι ένα ανοιχτό θέμα ως προς τον χρήστη ανάλογα με τον αποτέλεσμα που θέλει να πετύχει. Στην συγκεκριμένη περίπτωση αυτό που μας ενδιαφέρει περισσότερο είναι να εντοπίζει ικανοποιητικά τις 4 γωνίες του περιγράμματος κάθε φωτογραφίας, ενώ προφανώς δεν μπορεί να αποφευχθεί και ο εντοπισμός γωνιών στο εσωτερικό των φωτογραφιών.

1.3 Rotation

Αλγόριθμος και αποτελέσματα

Η συνάρτηση που καλούμαστε να υλοποιήσουμε είναι η:

```
1 function rotImg = myImgRotation(img, angle),
```

η οποία λαμβάνει σαν είσοδο μια εικόνα `img` και την γωνία `angle` σε rad, και επιστρέφει στην έξοδο την περιστραμμένη με αντισωρολογιακή φορά εικόνα. Η συνάρτηση αυτή λειτουργεί ανεξαρτήτως του αριθμού των καναλιών της εικόνας εισόδου. Η εικόνα εξόδου `rotImg` έχει διαστάσεις τέτοιες ώστε κατά την περιστροφή της εικόνας να μην χάνεται καθόλου πληροφορία και να χωράει ολόκληρη η εικόνα εξόδου ενώ έχουμε υποθέσει ότι το φόντο είναι μαύρο.

Ο αλγόριθμος κάνει ανάθεση στην εικόνα εξόδου `rotImg` από την αρχική και έπειτα ακολουθούμε διαδικασία `bilinear interpolation`. Αυτό συμβαίνει γιατί ένα pixel (i,j) της εικόνας εξόδου έχει προκύψει από ένα pixel (x,y) της εικόνας εισόδου. Η θέση (x,y) όμως δεν είναι σίγουρο ότι βρίσκεται πάνω σε ένα ακέραιο pixel της εικόνας εισόδου οπότε πρέπει να γίνει παρεμβολή για να βρεθεί η τιμή του.

Συνοπτικά τα βήματα του αλγορίθμου είναι:

- Υπολογισμός της διάστασης της νέας εικόνας με βάση τις διαστάσεις (ri,ci) της αρχικής εικόνας και την γωνία περιστροφής:

```
1 rows=ceil(ri*abs(cos(angle))+ci*abs(sin(angle))); %rows
2 cols=ceil(ri*abs(sin(angle))+ci*abs(cos(angle))); %columns
```

- Αρχικοποίηση του πίνακα `rotImg` με μηδενικά. Οι διαστάσεις του `rotImg` είναι αυτές που βρέθηκαν παραπάνω και τα κανάλια όσα και της αρχικής εικόνας.
- Εύρεση του κέντρου της αρχικής εικόνας και της τελικής εικόνας με σκοπό την διευκόλυνση των πράξεων παρακάτω.
- Σάρωση της εικόνας `rotImg` για κάθε pixel (i,j) της οποίας υπολογίζεται το pixel (x,y) της αρχικής εικόνας του οποίου η τιμή θα του ανατεθεί.
- Bilinear Interpolation και ανάθεση της τιμής στο pixel που εξετάζουμε. Όπως αναφέρθηκε στη θεωρία, στη διαδικασία του `bilinear interpolation` για κάθε pixel της τελικής εικόνας έχει υπολογισθεί το pixel της αρχικής από το οποίο πρέπει να αντλήσουμε την τιμή. Επειδή όμως οι δείκτες που βρήκαμε δεν είναι απαραίτητα ακέραιοι αριθμοί υπολογίζουμε το `weighted average` των γειτονικών pixels όπως εξηγήθηκε στην θεωρία και αναθέτουμε την τιμή. Αυτό ισχύει για οποιονδήποτε αριθμό καναλιών.

Στο `deliverable3.m` καλείται ο αλγόριθμος και τα αποτελέσματα φαίνονται στις Εικόνες 4 και 5 για τις γωνίες περιστροφής που ζητήθηκαν. Στην Εικόνα 6 μπορούμε να παρατηρήσουμε την ίδια ακριβώς εφαρμογή αλλά για grayscale εικόνες, πράγμα που αποδεικνύει την λειτουργία του αλγορίθμου ανεξάρτητα από τον αριθμό των καναλιών της εικόνας που δίνεται στην είσοδο.



Εικόνα 4: Η εικόνα περιστραμμένη κατά $54^\circ \times \pi/180$



Εικόνα 5: Η εικόνα περιστραμμένη κατά $213^\circ \times \pi/180$



(α) Περιστροφή κατά $54^\circ \times \pi/180$



(β) Περιστροφή κατά $213^\circ \times \pi/180$

Εικόνα 6: Εφαρμογή για grayscale εικόνες