Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Ψηφιακή Επεξεργασία Εικόνας

Εργασία 1

Κούτση Χριστίνα
ΑΕΜ: 9871
email: cvkoutsi@ece.auth.gr

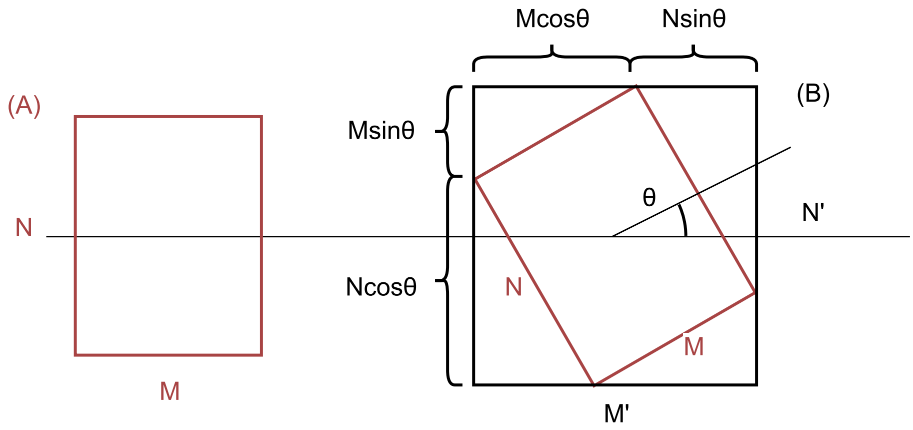
Μάιος, 2022

1 Εισαγωγικά

1.1 Περιστροφή της εικόνας κατά γωνία ϑ

Στο πρώτο ερώτημα μας ζητάται η κατασκευή μιας συνάρτησης η οποία θα δέχεται ως ορίσματα μία εικόνα και μία γωνία και θα περιστρέφει την γωνία κατά τις μοίρες που υποδεικνύει η γωνία

- Εύρεση διαστάσεων της μετασχηματισμένης εικόνας
  Στο παρακάτω σχήμα έχουμε την μετασχηματισμένη εικόνα (εικόνα (A)) και την τελική εικόνα που θα μας επιστρέψει η συνάρτηση, η οποία θα είναι η ανεστραμμένη εικόνα με μαύρο background.

  Παρατηρώ ότι οι δύο εικόνες δεν έχουν ίδιες διαστάσεις. Συγκεκριμένα, αν η εικόνα (A) έχει διαστάσεις MxN, η εικόνα (B) θα έχει διαστάσεις M′xN′ οι οποίες θα δίνονται από τη σχέση:

  $$\begin{cases} M' = M\,|\cos\vartheta\,| + N\,|\sin\vartheta\,| \\ N' = M\,|\sin\vartheta\,| + N\,|\cos\vartheta\,| \end{cases}$$

  Η απόλυτη τιμή μπαίνει έτσι ώστε να έχουμε θετικό N′ και M′ ανεξάρτητα σε ποιο τεταρτημόριο ανήκει η γωνία περιστροφής.

- Εύρεση πίνακα A
  Για την ανάλυσή μας, θεωρούμε ότι η αρχική εικόνα παριστάνεται από

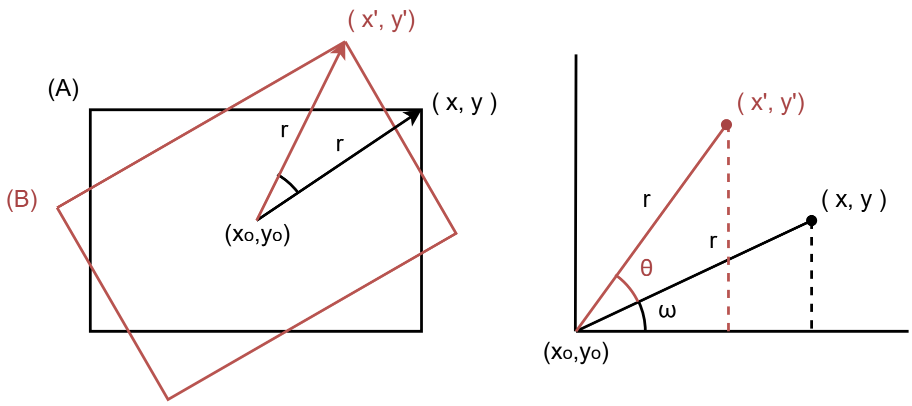
το σχήμα (A) και η τελική ανεστραμμένη εικόνα παριστάνεται από το σχήμα (B). Θεωρούμε ότι η αρχική εικόνα έχει κέντρο

$$(x_0, y_0) = (\frac{M}{2}, \frac{N}{2})$$

και περιστρέφεται γύρω από αυτό. Η ανεστραμμένη εικόνα θα έχει κέντρο

$$(x'_0, y'_0) = (\frac{M'}{2}, \frac{N'}{2})$$

λόγω της αλλαγής του μεγέθους της εικόνας από NxM σε N'xM'.



Για τις νέες συντεταγμένες (x',y') της ανεστραμμένης εικόνας ισχύει:

$$x' - x'_0 = rcos(\omega + \theta)$$
$$y' - y'_0 = rsin(\omega + \theta)$$

Από τις τριγωνομετρικές ταυτότητες

cos(A+B) = cosAcosB - sinAsinB
sin(A+B) = sinAcosB + cosAsin B

έχουμε:

$$x' - x'_0 = rcos\omega cos\theta - rsin\omega sin\theta$$
$$y' - y'_0 = rsin\omega cos\theta + rcos\omega sin\theta$$

'Ομως από το σχήμα έχουμε

$$\text{x - } x_0 = \text{rcos}\omega$$
$$\text{y - } y_0 = \text{rsin}\omega$$

Επομένως:

$$\begin{cases} x' = (x - x_0)cos\theta - (y - y_0)sin\theta + x'_0 \\ y' = (x - x_0)sin\theta + (y - y_0)cos\theta + y'_0 \end{cases}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x - x_0 \\ y - y_0 \end{bmatrix} + \begin{bmatrix} x'_0 \\ y'_0 \end{bmatrix}$$

Επομένως ο πίνακας μετασχηματισμού είναι ο

$$A = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

Λύνοντας ως προς τις συντεταγμένες x,y της αρχικής εικόνας, έχουμε:

$$\begin{cases} x = (x' - x'_0)cos\theta + (y' - y'_0)sin\theta + x_0 \\ y = -(x' - x'_0)sin\theta + (y' - y'_0)cos\theta + y_0 \end{cases} \qquad (1)$$

Επομένως, κάθε σημείο (x',y') της ανεστραμμένης εικόνας αντιστοιχεί το σημείο (x,y) της αρχικής εικόνας, το οποίο δίνεται από τη σχέση (1).

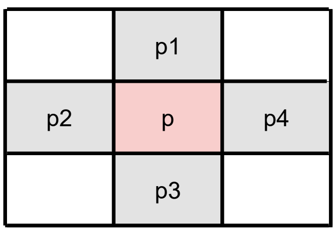
- Αλγόριθμος Bilinear Interpolation
  Έστω pixel p, όπως στην εικόνα 1. Εφαρμόζουμε bilinear interpolation για το pixel p το οποίο έχει 4 γειτονικά pixels, έχουμε:

$$p = \frac{p1 + p2 + p3 + p4}{4}$$

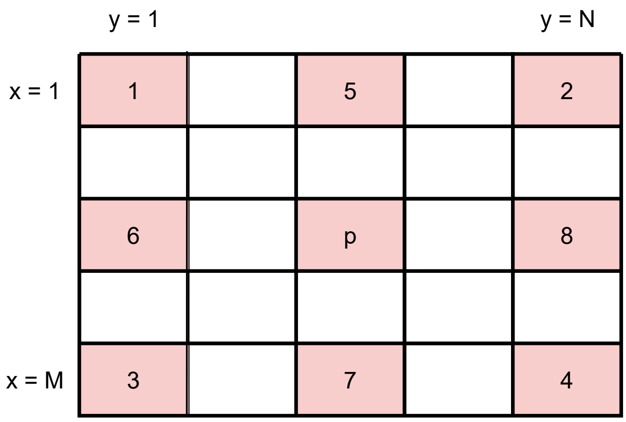
  Σε μία εικόνα όμως δεν έχουν όλα τα pixels 4 γείτονες (εικόνα 2), επομένως θα πρέπει να διακρίνουμε περιπτώσεις:

  - Ακριανά pixels με 3 γειτονικά pixels (5,6,7,8)
  - Γωνιακά pixels με 2 γειτονικά pixels (1,2,3,4)

  Αναλόγως σε ποια άκρη ή γωνία της εικόνας βρίσκεται το pixel θα πρέπει να λάβουμε τα κατάλληλα pixels για το bilinear interpolation ώστε να μην βγούμε από τα όρια της αρχικής εικόνας.



Εικόνα 1



Εικόνα 2

Έστω p = (x,y) και f εικόνα διάστασης MxN. Εφαρμόζοντας τα παραπάνω σε ψευδοκώδικα έχουμε:

Algorithm 1: Bilinear Interpolation

Input: x,y
Output: f(x,y)

```
value ← 0
neighbors ← 0
/* Check if p is in the boundaries of the image */
if x > 1 then
    value ← value + f(x − 1, y)
    neighbors ← neighbors + 1
if x < M then
    value ← value + f(x + 1, y)
    neighbors ← neighbors + 1
if y > 1 then
    value ← value + f(x, y − 1)
    neighbors ← neighbors + 1
if y > 1 then
    value ← value + f(x, y + 1)
    neighbors ← neighbors + 1
/* Get the value of the image */
f(x, y) ← value/neighbors
```

- Εύρεση της μετασχηματισμένης εικόνας
  Για κάθε pixel της μετασχηματισμένης εικόνας θα πρέπει να βρούμε ποια αντιστοιχούν σε pixel της αρχικής εικόνας και ποια θα κρατήσουν την default τιμή, δηλαδή θα παραμείνουν μαύρα. Αυτό το κάνουμε εφαρμόζοντας την παρακάτω διαδικασία

  1. Για κάθε pixel (x′,y′) της μετασχηματισμένης εικόνας, βρίσκουμε το pixel (x,y) της αρχικής εικόνας εφαρμόζοντας την σχέση (1).
  2. Ελέγχουμε αν το (x,y) βρίσκεται μέσα στα όρια της αρχικής εικόνας.
  3. Εφαρμόζουμε bilinear interpolation για να βρούμε την τιμή του pixel (x′,y′)

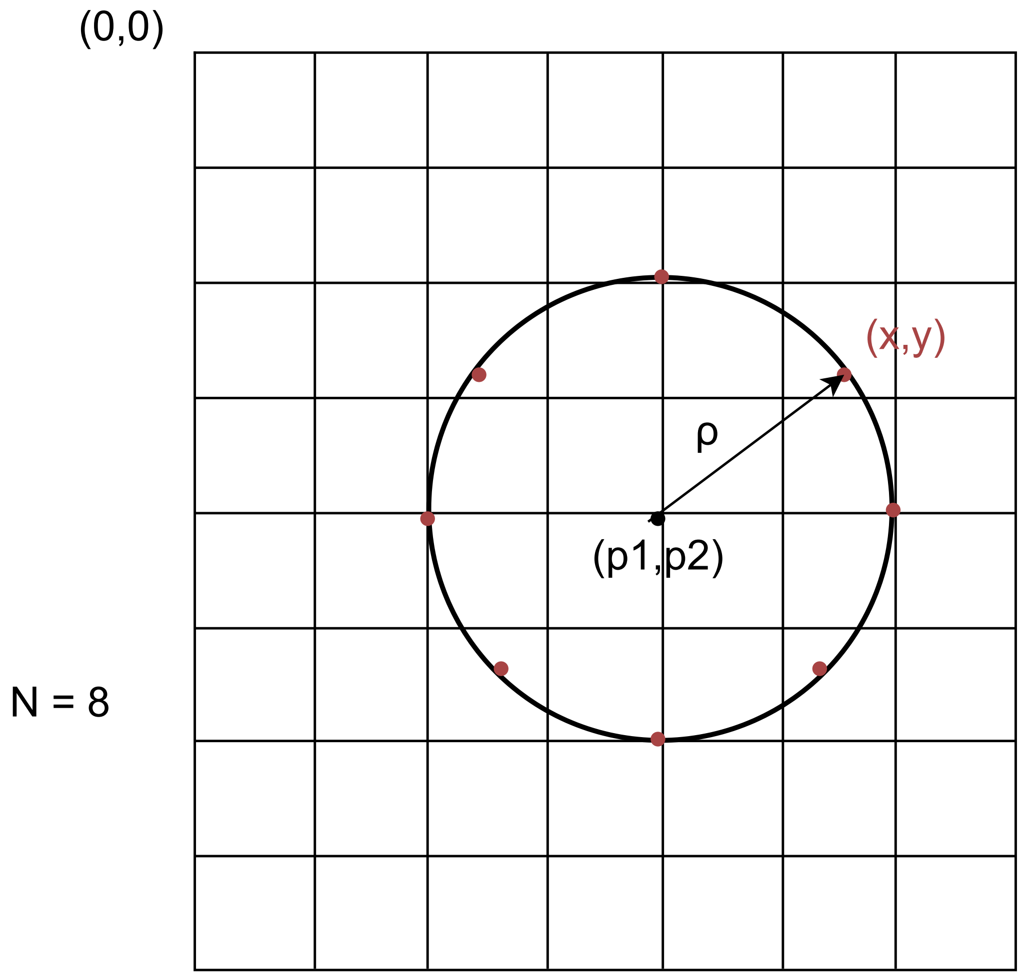
Εφαρμόζοντας την παραπάνω διαδικασία για $\theta_1 = 35°$ και $\theta_2 = 222°$ έχουμε:

(a) $\theta_1 = 35°$

(b) $\theta_2 = 222°$

1.2 Ένας **Local Descriptor**

Έστω σημείο p = [p1,p2] και θέλουμε να κατασκευάσουμε έναν local descriptor σαρώνοντας ομόκεντρο κύκλο ακτίνας ρ σε N σημεία [x,y].



Τα σημεία [x,y] περιγράφονται από τη σχέση

$$\begin{cases} x = p2 + \rho cos\theta \\ y = p1 - \rho sin\theta \end{cases}$$