



Επεξεργασία Εικόνας και Βίντεο

4^η Εργαστηριακή Άσκηση

Βελτίωση της εικόνας με Γεωμετρικούς Μετασχηματισμούς

Περιεχόμενα

1. Εισαγωγή στους Γεωμετρικούς Μετασχηματισμούς
2. Πίνακες μετασχηματισμών
3. Ασκήσεις
4. Λύσεις Ασκήσεων



ΕΙΣΑΓΩΓΗ

Εκτός από την βελτίωση της εικόνας με χρήση φίλτρων, η οποία αποσκοπεί κυρίως στην βελτίωση η αποκατάσταση των χαρακτηριστικών της φωτεινότητας της εικόνας, η αποκατάσταση των γεωμετρικών χαρακτηριστικών της είναι εξίσου σημαντική. Από την δημιουργία της μια εικόνα ενδέχεται να είναι παραμορφωμένη. Σκεφτείτε ένα έγγραφο το οποίο έχει σκαναριστεί με μια μικρή γωνία, ή μια φωτογραφία η οποία λόγω προοπτικής περιέχει αντικείμενα παραμορφωμένα. Ο πιο γνωστός μετασχηματισμός για να επαναφέρουμε την εικόνα και να αποκαταστήσουμε τις γεωμετρικές παραμορφώσεις είναι ο αγκίγραμμος μετασχηματισμός (Affine Transform). Ο μετασχηματισμός μας επιτρέπει να μετασχηματίζουμε τις συντεταγμένες της αρχικής εικόνας με χρήση πινάκων:

Ταυτότητα (Identity)

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

Κατοπτρισμός (mirroring) με βάση τον οριζόντιο άξονα

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

Κατοπτρισμός (mirroring) με βάση τον Κατακόρυφο άξονα

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

Κλιμάκωση (Scaling)

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

S_x : Πολλαπλασιάζει το μέγεθος στην οριζόντια διάσταση
 S_y : Πολλαπλασιάζει το μέγεθος στην κατακόρυφη διάσταση

Περιστροφή (Rotation)

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

θ : περιστροφή σε μοίρες

Μετατόπιση (Translation)

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

t_x : απόσταση στον οριζόντιο άξονα σε pixels
 t_y : απόσταση στον κατακόρυφο άξονα σε pixels

Παραμόρφωση (Shear) στον κατακόρυφο άξονα

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

Sh_x : Εκφράζει το μέγεθος της παραμόρφωσης στην οριζόντια διάσταση

Παραμόρφωση (Shear) στον οριζόντιο άξονα

$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

Sh_y : Εκφράζει το μέγεθος της παραμόρφωσης στην κατακόρυφη διάσταση

Σχήμα 1: Πίνακες μετασχηματισμών



ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

Άσκηση 1

Υλοποιήστε με κώδικα MATLAB τους πίνακες του μετασχηματισμός Affine με τις απαιτούμενες παραμέτρους. Συγκεκριμένα υλοποιήστε πίνακες για περιστροφή (rotation), κατοπτρισμό (mirroring-reflection), κλιμάκωση (scale), μετατόπιση (translation), και τους δύο πίνακες για την παραμόρφωση (shear).

Άσκηση 2

Εισάγεται στο workspace της MATLAB την εικόνα με όνομα 'trapezio.png' και εμφανίστε την εικόνα σε νέο παράθυρο. Χρησιμοποιώντας της παραμέτρους και τους πίνακες που χρησιμοποιήσατε στην άσκηση 1 πραγματοποιείτε διαδοχικά:

- i. Περιστροφή 47 μοιρών
- ii. Σμίκρυνση του σχήματος κατά 40% και στις δύο κατευθύνσεις
- iii. Μεταφορά του σχήματος κατακόρυφα κατά -300 pixels και οριζόντια κατά 400 pixels
- iv. Παραμόρφωση ShearY κατά 0.7
- v. Όλα τα παραπάνω μαζί



ΛΥΣΕΙΣ

ΑΣΚΗΣΗ 1

```
clear, close all, clc

%Parameteres
angle_rad = deg2rad(47);           % Η γωνία στρώσεως σε rad
scaleY = 0.6;                       % Klimakosi_ston_katakoryfo
scaleX = 0.6;                       % Klimakosi_ston_orizontio
translateX = -300;                  % Metakinshh ston orizontio
translateY = 400;                   % Metakinshh ston katakorifo
Sx = 0.8;                           % Paramorfosi ston x
Sy = 0.7;                           % Paramorfosi ston y

% Define matrix for Rotation
Matrix_for_rotation = [ cos(angle_rad) sin(angle_rad) 0;
                        -sin(angle_rad) cos(angle_rad) 0;
                        0 0 1];

% Define matrix for Reflection
Matrix_for_reflection = [1 0 0;
                         0 -1 0;
                         0 0 1];

% Define matrix for Scale
Matrix_for_scaling = [scaleX 0 0;
                      0 scaleY 0;
                      0 0 1];

% Define matrix for translation
Matrix_for_translate = [1 0 translateX;
                        0 1 translateY;
                        0 0 1];

% Define matrix for vertical Shear
Matrix_for_shearY = [1 0 0;
                     Sy 1 0;
                     0 0 1];

% Define matrix for horizontal Shear
Matrix_for_shearX = [1 0 0;
                     0 1 0;
                     0 Sx 1];
```

ΑΣΚΗΣΗ 2

ΑΣΚΗΣΗ 1

```
clear, close all, clc
I = imread('trapezio.png');      %eisagw thn eikona
I = logical(I(:,:,1));           %epidei thn eisagwi rgb metatrepw se binary
figure, imshow(I)                % optikopoihs ths eikonas
I_transform = zeros(size(I,1),size(I,2));

%Parameteres
angle_rad = deg2rad(47);         % Η γωνία στρώσεως σε rad
scaleY = 0.6;                    % Klimakosi_ston_katakoryfo
```



```

scaleX = 0.6; % Klimakosi_ston_orizontio
translateX = -300; % Metakinshsh ston orizontio
translateY = 400; % Metakinshsh ston katakorifo
Sx = 0.8; % Paramorfosi ston x
Sy = 0.7; % Paramorfosi ston y

% Define matrix for Rotation
Matrix_for_rotation = [ cos(angle_rad) sin(angle_rad) 0;
                        -sin(angle_rad) cos(angle_rad) 0;
                        0 0 1];

% Define matrix for Reflection
Matrix_for_reflection = [1 0 0;
                         0 -1 0;
                         0 0 1];

% Define matrix for Scale
Matrix_for_scaling = [scaleX 0 0;
                      0 scaleY 0;
                      0 0 1];

% Define matrix for translation
Matrix_for_translate = [1 0 translateX;
                        0 1 translateY;
                        0 0 1 ];

% Define matrix for vertical Shear
Matrix_for_shearY = [1 0 0;
                     Sy 1 0;
                     0 0 1];

% Define matrix for horizontal Shear
Matrix_for_shearX = [1 0 0;
                     0 1 0;
                     0 Sx 1];

affine_matrix =Matrix_for_translate;

%Matrix_for_translate*Matrix_for_rotation*Matrix_for_scaling*Matrix_for_shearY;;

for i = 1:size(I,1) % grammes
    for j = 1:size(I,2) % sthles
        XY_new = round(affine_matrix*[i;j;1]);
        if XY_new(1) > 0 & XY_new(1) < size(I,1) & XY_new(2) > 0 & XY_new(2) < size(I,2)
            I_transform(XY_new(1), XY_new(2)) = I(i,j);
        end
    end
end
figure, imshow(I_transform)

```