

Επεξεργασία Εικόνας & Βίντεο

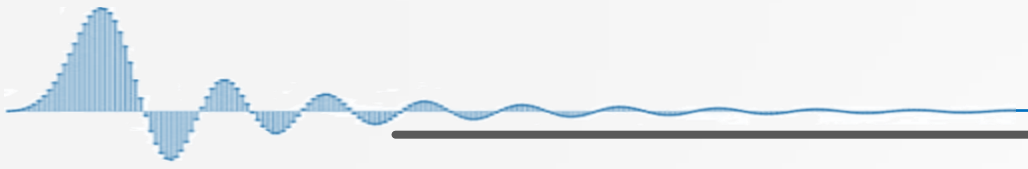


04. Βελτίωση της εικόνας με
Γεωμετρικούς
Μετασχηματισμούς

Εισηγητής: Νικόλαος Γιαννακέας

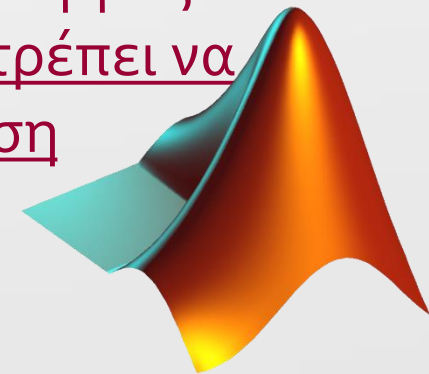
Επίκουρος Καθηγητής, Σημάτων & Συστημάτων





Εισαγωγή

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 **αγχίγραμμος μετασχηματισμός** (Affine Transform). Ο μετασχηματισμός μας επιτρέπει να μετασχηματίσουμε τις συντεταγμένες της αρχικής εικόνας με χρήση πινάκων:





Εισαγωγή

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

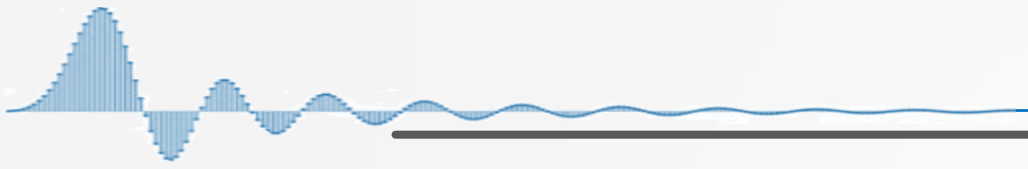
$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

Sh_x : Εκφράζει το μέγεθος της παραμόρφωσης στην οριζόντια διάσταση

Παραμόρφωση (Shear) στον οριζόντιο άξονα

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

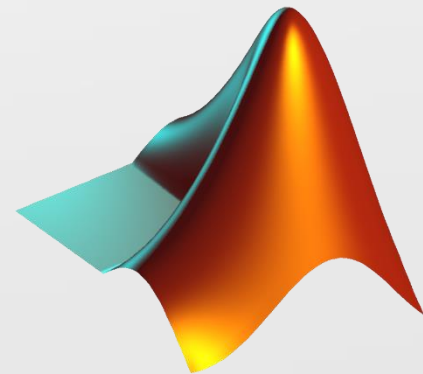
Sh_y : Εκφράζει το μέγεθος της παραμόρφωσης στην κατακόρυφη διάσταση



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).



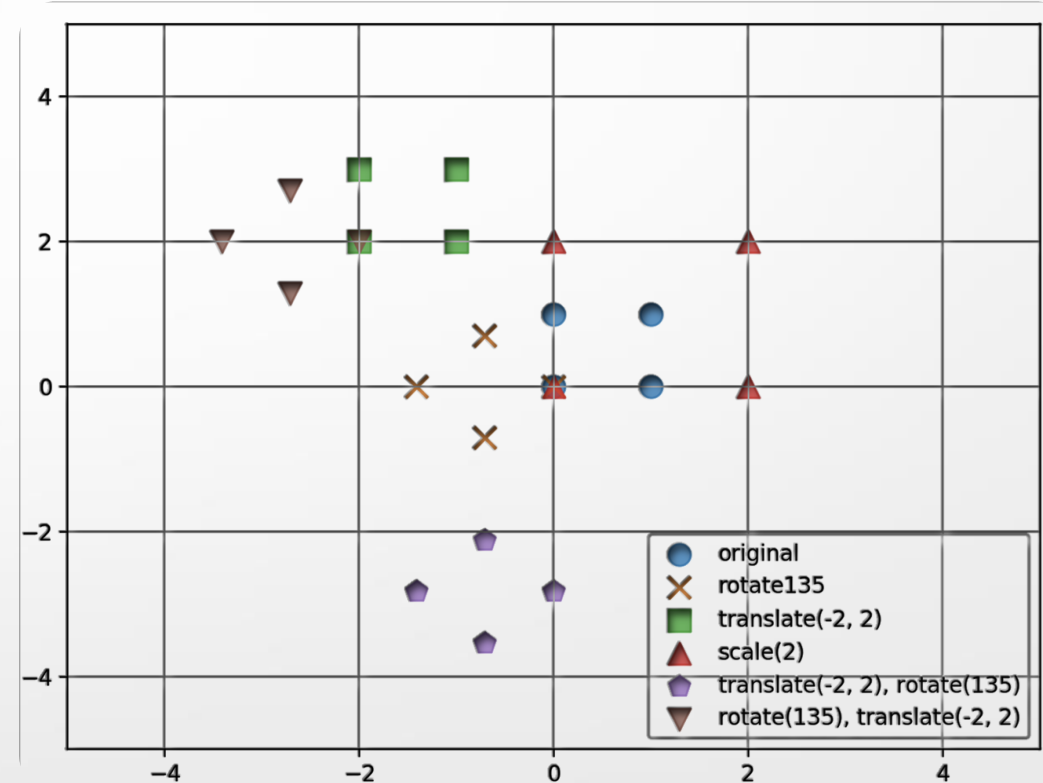


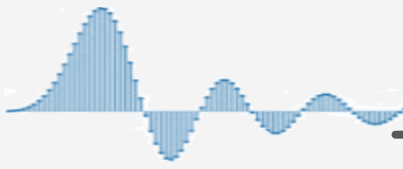
Γεωμετρικοί Μετασχηματισμοί

Αγχίγραμμος Μετασχηματισμός (Affine Transform)

Ο Αγχίγραμμος Μετασχηματισμός με επιτρέπει με **χρήση πινάκων** να εφαρμόσουμε πολλούς γεωμετρικούς μετασχηματισμούς είτε μεμονωμένα είτε όλους μαζί:

- Κλιμάκωση (Scaling)
- Κατοπτρισμός (Reflection)
- Περιστροφή (Rotation)
- Μετατόπιση (Translation)
- Οριζόντια παραμόρφωση (Horizontal Shear)
- Κατακόρυφη Παραμόρφωση (Vertical Shear)

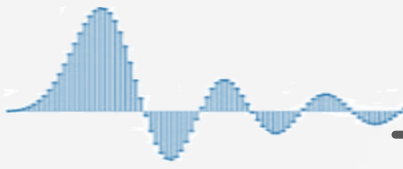




Γεωμετρικοί Μετασχηματισμοί

Αγχίγραμμος Μετασχηματισμός (Affine Transform)

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



Γεωμετρικοί Μετασχηματισμοί

Αγχίγραμμος Μετασχηματισμός (Affine Transform)

- 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

- 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



Γεωμετρικοί Μετασχηματισμοί

Αγχίγραμμος Μετασχηματισμός (Affine Transform)

- Παραμόρφωση (Shear) στον κατακόρυφο άξονα

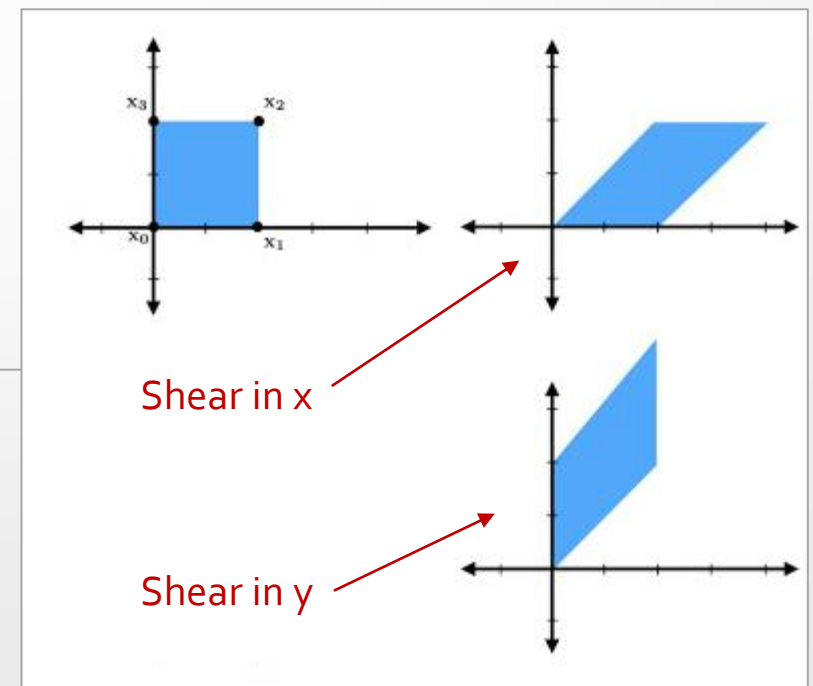
$$\begin{bmatrix} x_{transformed} \\ y_{transformed} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ Sh_y & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_{initial} \\ y_{initial} \\ 1 \end{bmatrix}$$

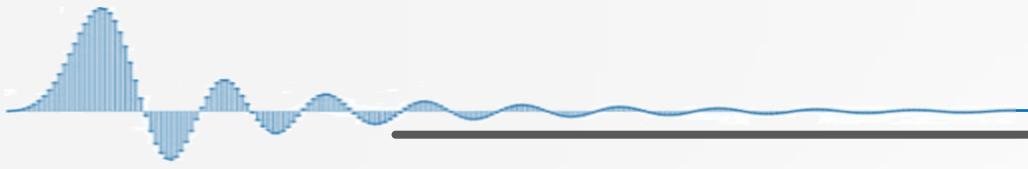
Sh_x : Εκφράζει το μέγεθος της παραμόρφωσης στην οριζόντια διάσταση

- Παραμόρφωση (Shear) στον οριζόντιο άξονα

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

Αρχικό Σχήμα





Αρχικοποίηση Παραμέτρων

ΛΥΣΗ

```
angle_rad = deg2rad(47);
```

→ Η γωνία περιστροφής σε rad

```
scaleY = 0.6
```

→ Κλιμάκωση στην κατακόρυφη διάσταση

```
scaleX = 0.6
```

→ Κλιμάκωση στην οριζόντια διάσταση

```
translateX = -300;
```

→ Μετακίνηση στην οριζόντια διάσταση

```
translateY = 400;
```

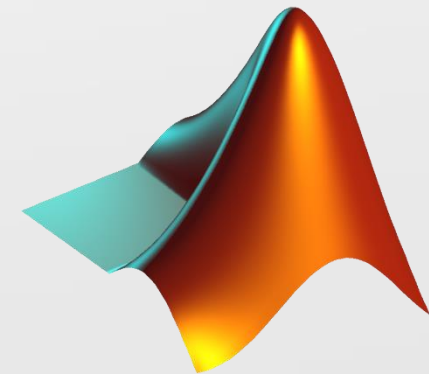
→ Μετακίνηση στην κατακόρυφη διάσταση

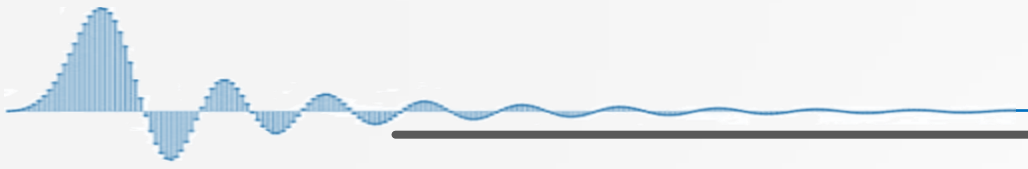
```
Sx = 0.8;
```

→ Παραμόρφωση στον άξονα X

```
Sy = 0.7;
```

→ Παραμόρφωση στον άξονα Y





Περιστροφή (Rotation)

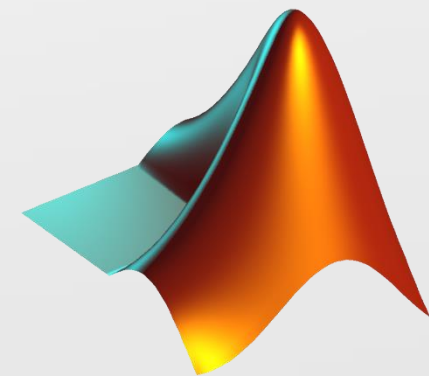
ΛΥΣΗ

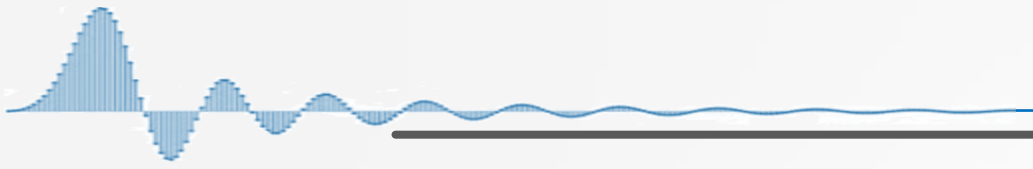
```
Matrix_for_rotation = [ cos(angle_rad) sin(angle_rad) 0;  
                        -sin(angle_rad) cos(angle_rad) 0;  
                        0      0      1];
```

- 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





Κατοπτρισμός (mirroring)

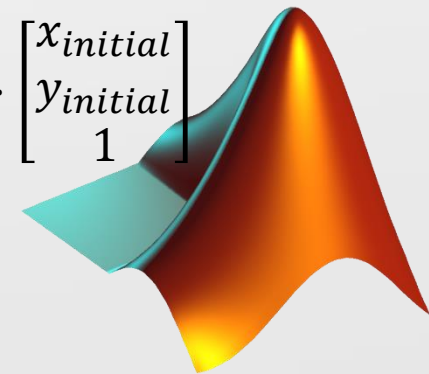
ΛΥΣΗ

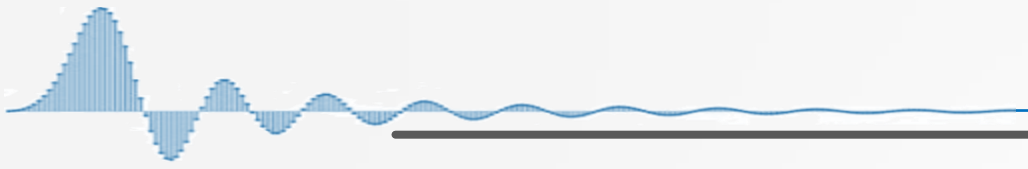
```
Matrix_for_reflection = [1 0 0;  
                        0 -1 0;  
                        0 0 1];
```

- Κατοπτρισμός (mirroring) με βάση τον οριζόντιο άξονα
- Κατοπτρισμός (mirroring) με βάση τον κατακόρυφο άξονα

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

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





Κλιμάκωση (Scaling)

ΛΥΣΗ

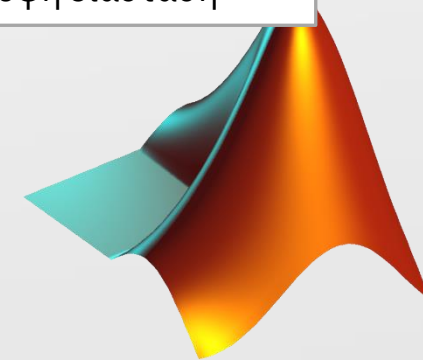
```
Matrix_for_scaling = [scaleX  0  0;  
                      0  scaleY 0;  
                      0  0  1];
```

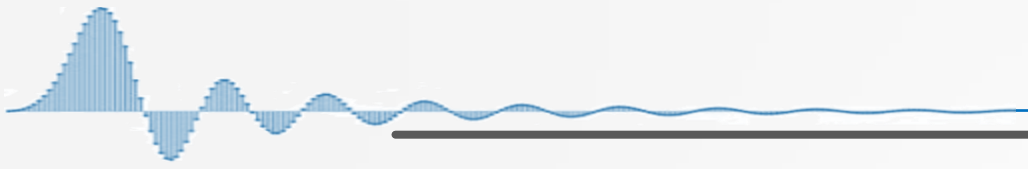
- Κλιμάκωση (Scaling)

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

S_x : Πολλαπλασιάζει το μέγεθος στην οριζόντια διάσταση

S_y : Πολλαπλασιάζει το μέγεθος στην κατακόρυφη διάσταση





Μετατόπιση (Translation)

ΛΥΣΗ

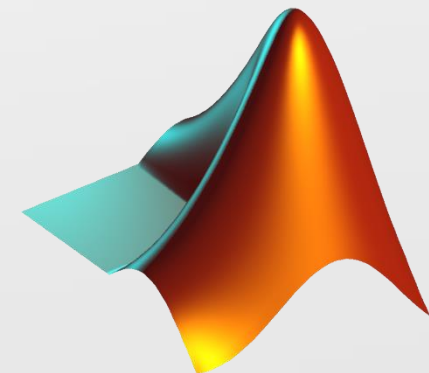
```
Matrix_for_translate = [1  0  translateX;  
                        0  1  translateY;  
                        0  0   1  ];
```

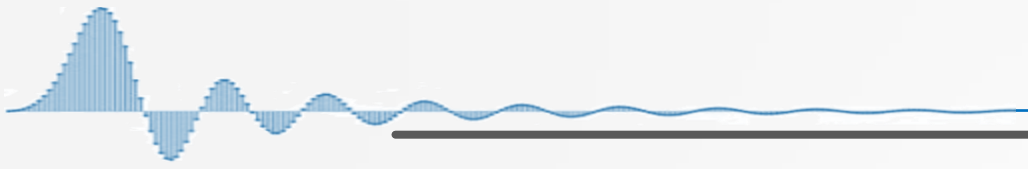
- Μετατόπιση (Translation)

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

t_x : απόσταση στον οριζόντιο άξονα σε pixels

t_y : απόσταση στον κατακόρυφο άξονα σε pixels





Παραμόρφωση (Shear)

ΛΥΣΗ

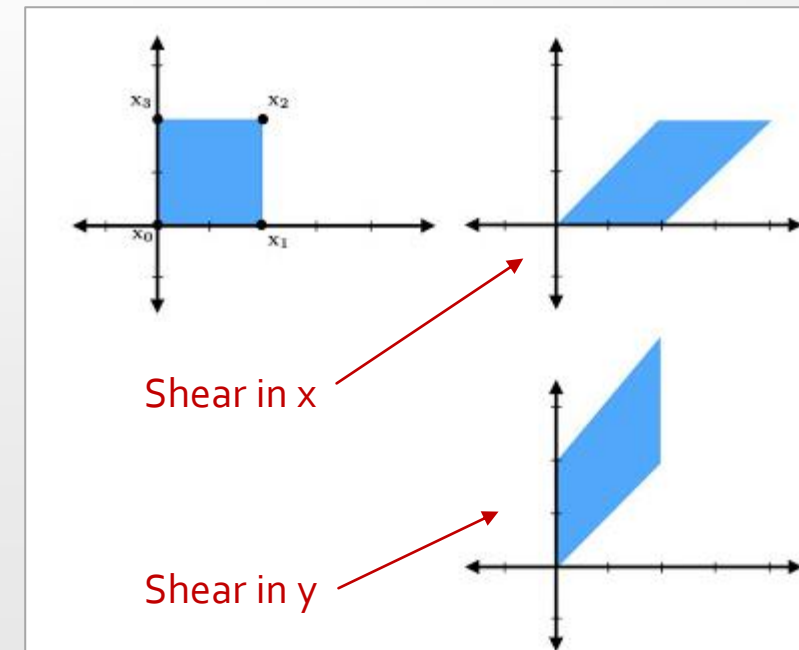
$$\text{Matrix_for_shearY} = \begin{bmatrix} 1 & 0 & 0; \\ & S_y & 1 & 0; \\ & 0 & 0 & 1 \end{bmatrix};$$

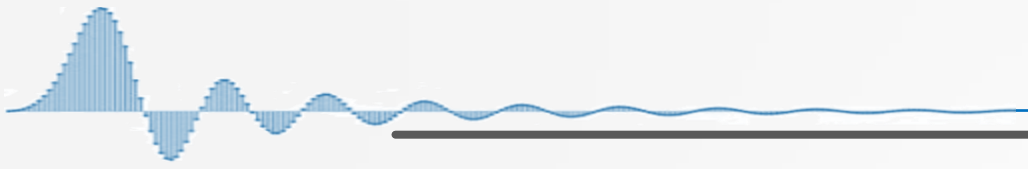
- Παραμόρφωση (Shear) στον κατακόρυφο άξονα

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

S_{h_x} : Εκφράζει το μέγεθος της παραμόρφωσης στην οριζόντια διάσταση

Αρχικό Σχήμα





Παραμόρφωση (Shear)

ΛΥΣΗ

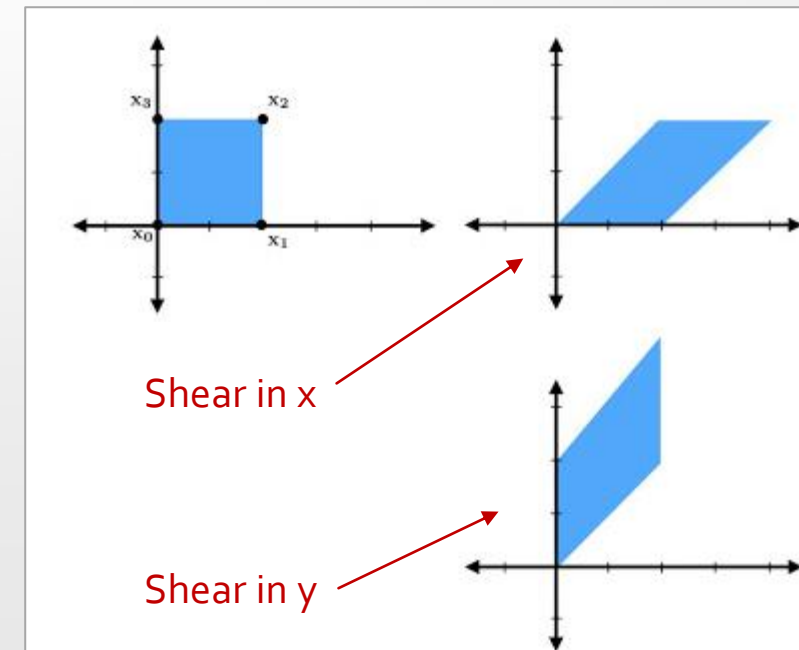
$$\text{Matrix_for_shearX} = \begin{bmatrix} 1 & 0 & 0; \\ 0 & 1 & 0; \\ 0 & Sx & 1 \end{bmatrix};$$

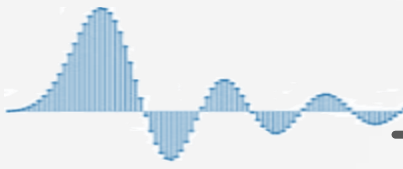
- Παραμόρφωση (Shear) στον οριζόντιο άξονα

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

Sh_x : Εκφράζει το μέγεθος της παραμόρφωσης στην οριζόντια διάσταση

Αρχικό Σχήμα



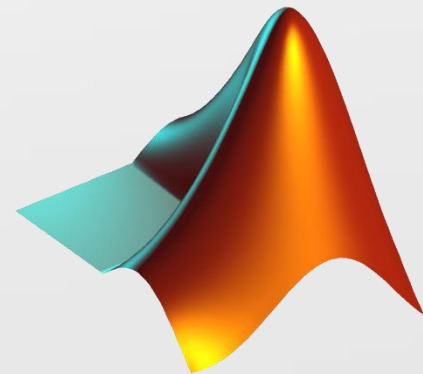


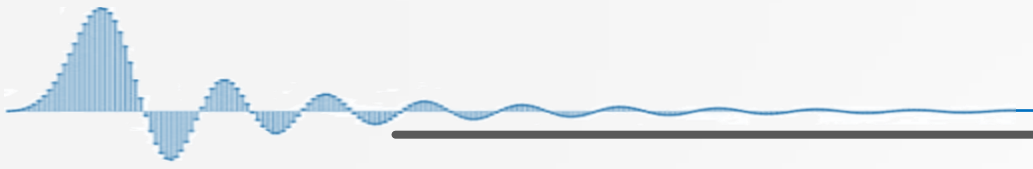
ΠΕΙΡΑΜΑΤΙΚΗ ΔΙΑΔΙΚΑΣΙΑ

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





ΛΥΣΗ

```
I = imread('trapezio.png');
```

```
I = logical(I(:,:,1));
```

↓
Οποιοδήποτε μη μηδενικό στοιχείο του I μετατρέπεται σε λογικό 1 (true) και τα μηδενικά μετατρέπονται σε λογικό 0 (false).

```
figure, imshow(I)
```

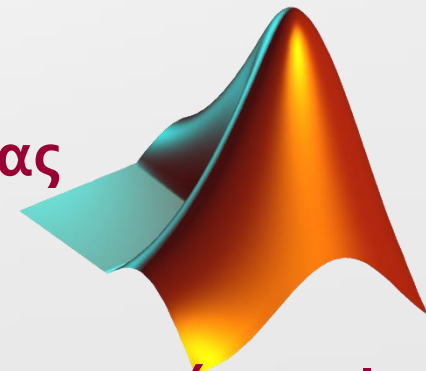
```
I_transform = zeros(size(I,1),size(I,2));
```

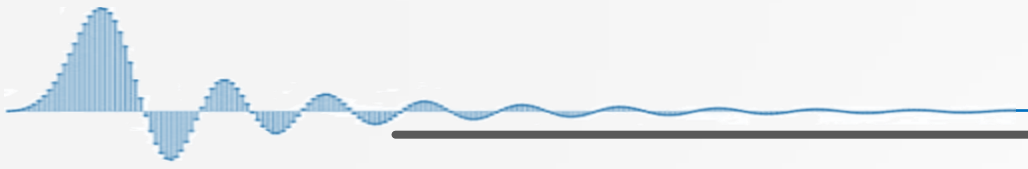
→ Διάβασμα εικόνας

→ Μετατροπή εικόνας σε δυαδική

→ Εμφάνιση εικόνας

→ Αρχικοποιούμε ένα πίνακα `filtered` με μέγεθος το μέγεθος του πίνακα I





Αρχικοποίηση Παραμέτρων

ΛΥΣΗ

```
angle_rad = deg2rad(47);
```

→ Η γωνία περιστροφής σε rad

```
scaleY = 0.6
```

→ Κλιμάκωση στην κατακόρυφη διάσταση

```
scaleX = 0.6
```

→ Κλιμάκωση στην οριζόντια διάσταση

```
translateX = -300;
```

→ Μετακίνηση στην οριζόντια διάσταση

```
translateY = 400;
```

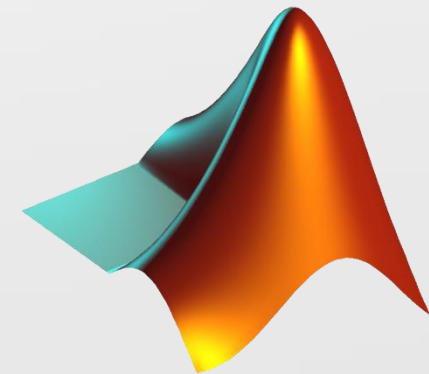
→ Μετακίνηση στην κατακόρυφη διάσταση

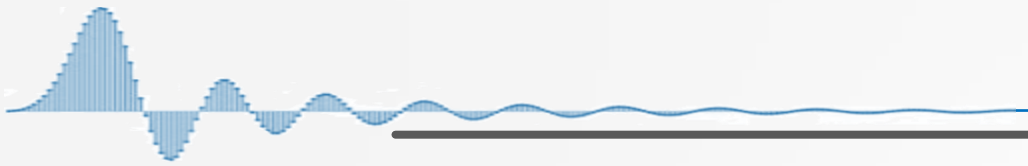
```
Sx = 0.8;
```

→ Παραμόρφωση στον άξονα X

```
Sy = 0.7;
```

→ Παραμόρφωση στον άξονα Y





```
I = imread('trapezio.png');  
I = logical(I(:,:,1));  
figure, imshow(I)  
I_transform = zeros(size(I,1),size(I,2));
```

```
Matrix_for_rotation = [ cos(angle_rad) sin(angle_rad) 0;  
                       -sin(angle_rad) cos(angle_rad) 0;  
                       0 0 1];
```

Περιστροφή (Rotation)

```
Matrix_for_reflection = [1 0 0;  
                         0 -1 0;  
                         0 0 1];
```

Κατοπτρισμός (mirroring)

```
Matrix_for_scaling = [scaleX 0 0;  
                      0 scaleY 0;  
                      0 0 1];
```

Κλιμάκωση (Scaling)

```
Matrix_for_translate = [1 0 translateX;  
                        0 1 translateY;  
                        0 0 1 ];
```

Μετατόπιση (Translation)

```
Matrix_for_shearY = [1 0 0;  
                    Sy 1 0;  
                    0 0 1];
```

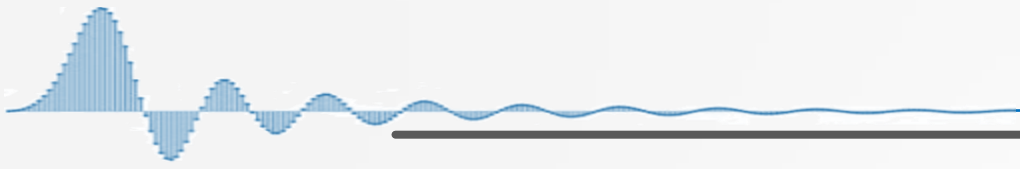
Παραμόρφωση (Shear) στον κατακόρυφο άξονα

```
Matrix_for_shearX = [1 0 0;  
                    0 1 0;  
                    0 Sx 1];
```

Παραμόρφωση (Shear) στον οριζόντιο άξονα

ΛΥΣΗ

```
angle_rad = deg2rad(47);  
scaleY = 0.6  
scaleX = 0.6  
translateX = -300;  
translateY = 400;  
Sx = 0.8;  
Sy = 0.7;
```



ΛΥΣΗ

`affine_matrix = Matrix_for_translate;` → **Επιλογή πίνακα**

`for i = 1:size(I,1)` → **Πλήθος γραμμών πίνακα**

`for j = 1:size(I,2)` → **Πλήθος στηλών πίνακα**

`XY_new = round(affine_matrix*[i;j;1]);` → **Αντικατάσταση επιλεγμένου πίνακα στον αρχικό πίνακα και στρογγυλοποίηση**

`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`

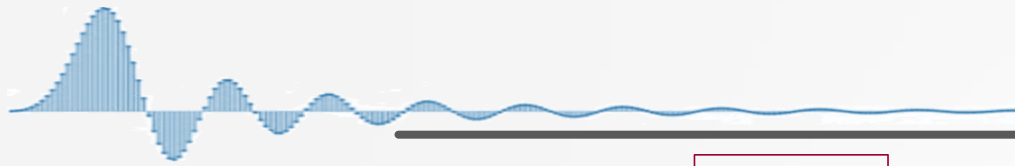
`end`

`end`

`figure, imshow(I_transform)`

Ελέγχουμε αν η θέση του XY είναι στα όρια της εικόνας

`XY_new(1) = x άξονας`
`XY_new(2) = y άξονας`



```

I = imread('trapezio.png');
I = logical(I(:,:,1));
figure, imshow(I)
I_transform = zeros(size(I,1),size(I,2));

```

```

angle_rad = deg2rad(47);
scaleY = 0.6
scaleX = 0.6
translateX = -300;
translateY = 400;
Sx = 0.8;
Sy = 0.7;

```

```

Matrix_for_rotation = [ cos(angle_rad) sin(angle_rad) 0;
                       -sin(angle_rad) cos(angle_rad) 0;
                       0 0 1];

```

```

Matrix_for_reflection = [1 0 0;
                        0 -1 0;
                        0 0 1];

```

```

Matrix_for_scaling = [scaleX 0 0;
                     0 scaleY 0;
                     0 0 1];

```

```

Matrix_for_translate = [1 0 translateX;
                       0 1 translateY;
                       0 0 1 ];

```

```

Matrix_for_shearY = [1 0 0;
                    Sy 1 0;
                    0 0 1];

```

```

Matrix_for_shearX = [1 0 0;
                    0 1 0;
                    0 Sx 1];

```

```
affine_matrix = Matrix_for_translate;
```

```

for i = 1:size(I,1)
  for j = 1:size(I,2)
    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)
```

ΛΥΣΗ

