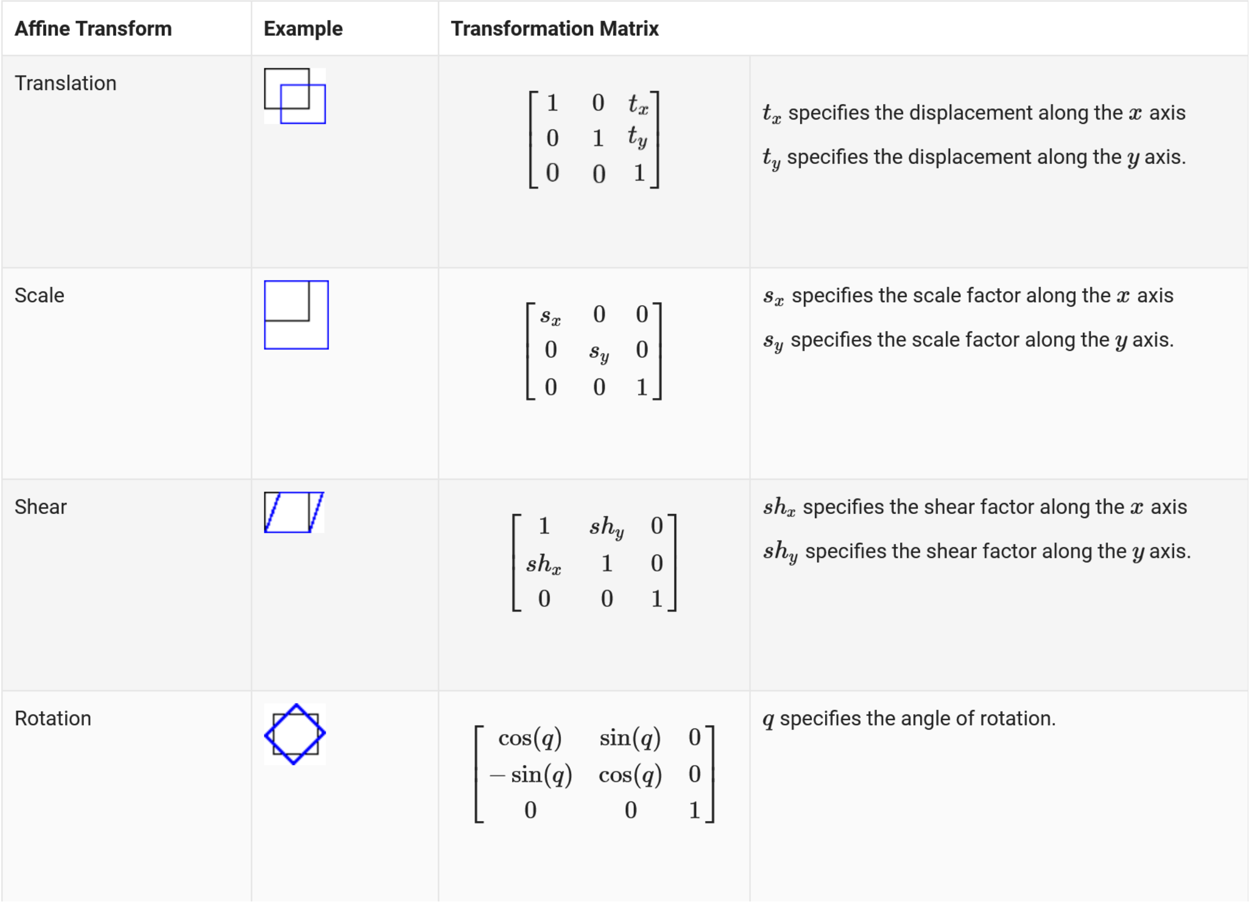
LAB 6: IMAGE TRANSFORMATIONS (AFFINE TRANSFORMATIONS)



Affine Transform	Example	Transformation Matrix	
Translation		$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	t_x specifies the displacement along the x axis t_y specifies the displacement along the y axis.
Scale		$\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	s_x specifies the scale factor along the x axis s_y specifies the scale factor along the y axis.
Shear		$\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	sh_x specifies the shear factor along the x axis sh_y specifies the shear factor along the y axis.
Rotation		$\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$	q specifies the angle of rotation.

[1]

To apply an affine transformation on an image, do the following:

1. Define transformation matrix

```
A = [2 0 0; 0.33 1 0; 0 0 1];
```

2. Create an affinetform2d object from the transformation matrix.

```
tform = affinetform2d(A)
```

3. Apply the geometric transformation to the image, and display the result.

```
J = imwarp(I,tform);
```

Experiment No. 31 Image Scaling

1. Read an image into the workspace and show it.

[1] Ref: https://www.mathworks.com/discovery/affine-transformation.html

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

figure();

imshow(I);

axis on;
```

2. Define transformation matrix, downscale the image by factor 0.5.

```
A = [0.5 0 0; 0 0.5 0; 0 0 1];
```

3. Create an affinetform2d object from the transformation matrix.

```
tform = affinetform2d(A);
```

4. Apply the geometric transformation to the image, and display the result.

```
J = imwarp(I,tform);

imshow(J);

hold off;

axis on;
```

Experiment No. 32 Image Rotation

1. Read an image into the workspace.

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

figure();

imshow(I);

axis on;
```

2. Define transformation matrix, rotate the image by 45 about the origin.

```
A = [cos(45) sin(45) 0; -sin(45) cos(45) 0; 0 0 1];
```

3. Create an affinetform2d object from the transformation matrix.

```
tform = affinetform2d(A)
```

4. Apply the geometric transformation to the image, and display the result.

```
J = imwarp(I,tform);

figure();
```

```
imshow(J);

hold off;

axis on;
```

Experiment No. 33 Image Translation

1. Read an image into the workspace.

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

   figure();

   imshow(I);

   axis on;
   ```

2. Define transformation matrix, shift the image 3 units to the right and 4 units down.

   ```
   A = [1 0 3; 0 1 -4; 0 0 1];
   ```

3. Create an affinetform2d object from the transformation matrix.

   ```
   tform = affinetform2d(A);
   ```

4. Apply the geometric transformation to the image, and display the result.

   ```
   J = imwarp(I,tform);

   figure();

   imshow(J);

   hold off;

   axis on;
   ```

Experiment No. 34 Image Shearing

1. Read an image into the workspace.

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

   figure();

   imshow(I);

   axis on;
   ```

2. Define transformation matrix that Shears the image by factor 3 along the x-axis (horizontal shearing).

```
A = [1 3 0; 0 1 0; 0 0 1];
```

3. Create an affinetform2d object from the transformation matrix.

```
tform = affinetform2d(TR);
```

4. Apply the geometric transformation to the image, and display the result.

```
J = imwarp(I,tform);

figure();

imshow(J);

hold off;

axis on;
```

Experiment No. 35 Image Reflection

Reflect around the x-axis

1. Read an image into the workspace.

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

figure();

imshow(I);

axis on;
```

2. Define transformation matrix.

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

3. Create an affinetform2d object from the transformation matrix.

```
tform = affinetform2d(A);
```

4. Apply the geometric transformation to the image, and display the result.

```
J = imwarp(I,tform);

figure();

imshow(J);

hold off;
```

```
axis on;
```

Reflect around the y-axis

1. Read an image into the workspace.

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

   figure();

   imshow(I);

   axis on;
   ```

2. Define transformation matrix.

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

3. Create an affinetform2d object from the transformation matrix.

   ```
   tform = affinetform2d(A);
   ```

4. Apply the geometric transformation to the image, and display the result.

   ```
   J = imwarp(I,tform);

   figure();

   imshow(J);

   hold off;

   axis on;
   ```

Experiment No. 36 Composite Transformations

To perform composite transformations:

Create a matrix that represents the individual transformations, then create the composite transformation by multiplying the matrices together, and finally store the transformation matrix as an `affinetform2d` object.

Note: The order of the transformation matrices matters.

Rotation Followed by Translation Example

1. Read an image into the workspace.

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

   figure();

   imshow(I);
   ```

```
axis on;
```

2. Define rotation transformation matrix.

```
R = [cos(30) sin(30) 0; -sin(30) cos(30) 0; 0 0 1];
```

3. Define transformation matrix that translate the image 3 units to the left.

```
T = [0 0 3; 0 1 0; 0 0 1];
```

4. Perform rotation first and translation second. Using the premultiply matrix convention, the translation matrix T is on the left and the rotation matrix R is on the right.

```
TR = T*R
```

5. Create an affinetform2d object from the transformation matrix.

```
tform = affinetform2d(TR);
```

6. Apply the geometric transformation to the image, and display the result.

```
J = imwarp(I,tform);

figure();

imshow(J);

hold off;

axis on;
```

Task:

- Read an image.
- Construct the transformation matrix of composite transformations as follows:
    - Rotate 60^o anti-clockwise around the origin.
    - Scale down by 0.5 factor.
    - Reflect around the y-axis.
- Apply the composite transformations.
- Show the original and transformed images in the same figure.