BISHOP HEBER COLLEGE (AUTONOMOUS)
DEPARTMENT OF DATA SCIENCE

Name: PAVITHIRAN.V

Roll.no: 235229122

EXERCISE 1: ROTATION

Perform a rotation of an image by 45 degrees counterclockwise. Display the original and rotated images side by side.

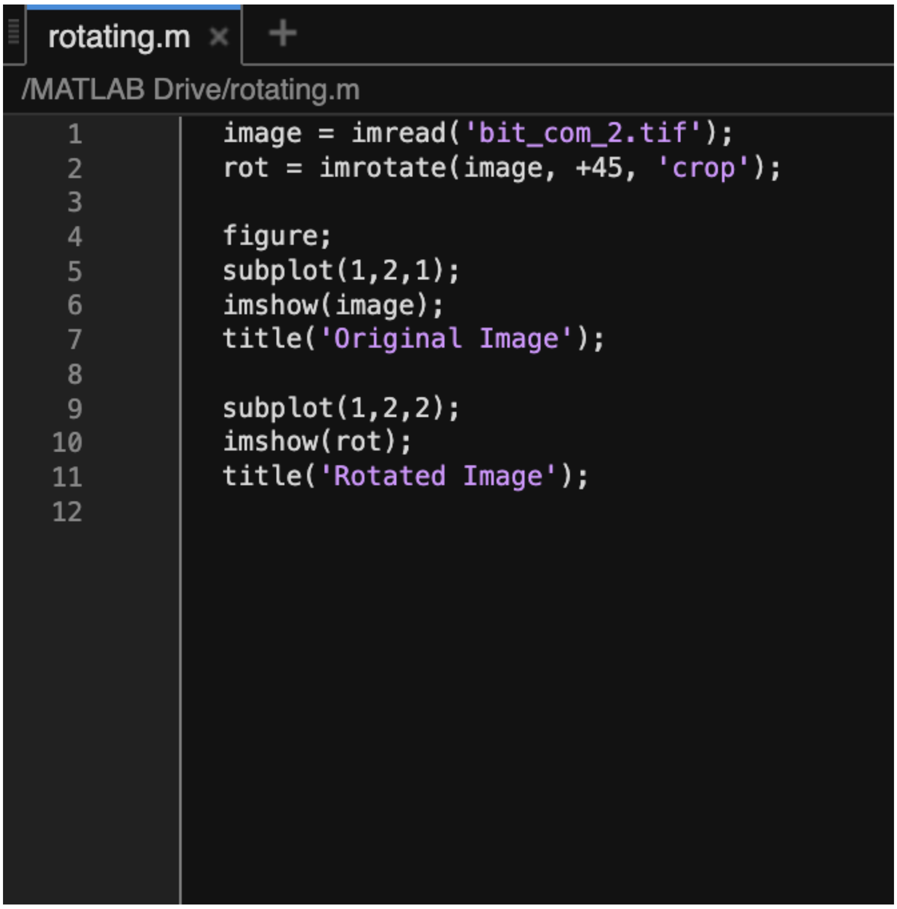
Step 1: Read the original image using the 'imread' function
Replace 'input _image jpg' with the actual path and filename of your image.

Step 2: Perform the rotation using the 'imrotate function
The imrotate function takes the original image and the rotation angle O as input and returns the rotated image.

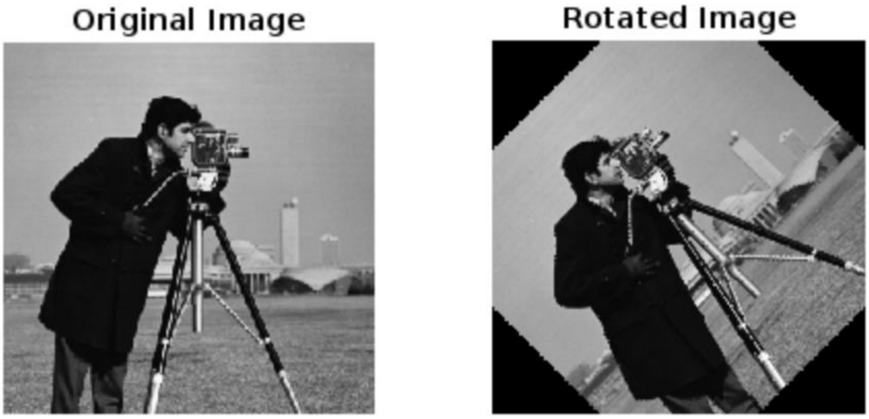
Step 3: Display the original and rotated images side by side
Create a figure with two subplots using the subplot function. The first subplot displays the original image using 'imshow', and the second subplot displays the rotated image. The title function is used to provide titles for each subplot.

CODE:

```
image = imread('bit_com_2.tif');
rot = imrotate(image, +45, 'crop');

figure;
subplot(1,2,1);
imshow(image);
title('Original Image');

subplot(1,2,2);
imshow(rot);
title('Rotated Image');
```



```
image = imread('bit_com_2.tif');
rot = imrotate(image, +45, 'crop');

figure;
subplot(1,2,1);
imshow(image);
title('Original Image');

subplot(1,2,2);
imshow(rot);
title('Rotated Image');
```



Original Image

Rotated Image

EXERCISE 2: SCALING

Implement a scaling transformation on an image by a factor of 1.5. Display the original and scaled images side by side.
Step 1: Read the original image using the 'imread' function
Replace 'input _image.jpg' with the actual path and filename of your image.

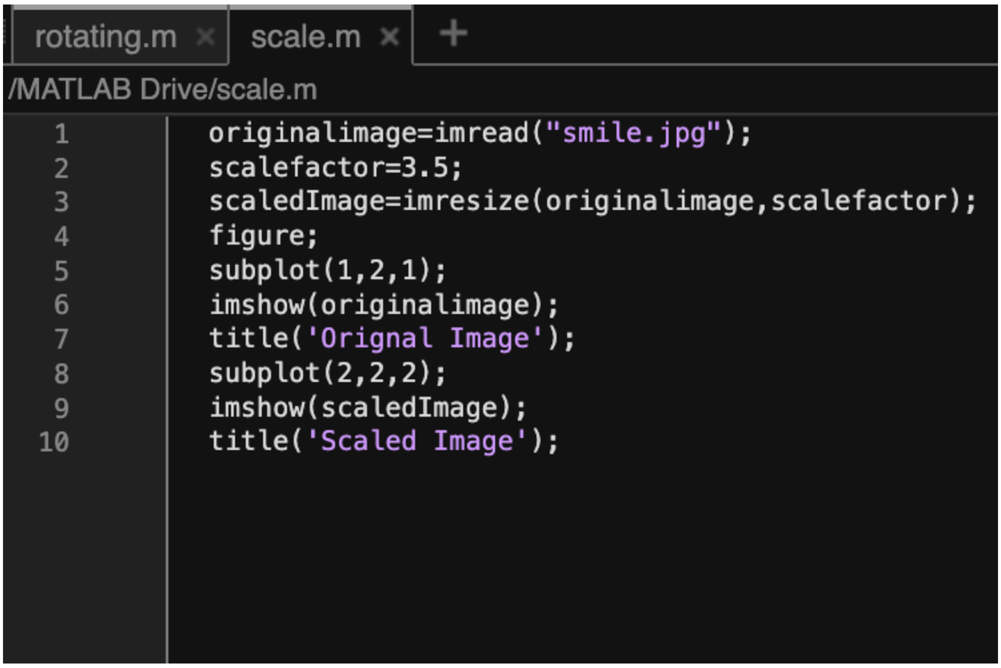
Step 2: Perform the scaling transformation using the 'imresize function
The imresize function takes the original image and the scaling factor as input and returns the scaled image.

Step 3: Display the original and scaled images side by side

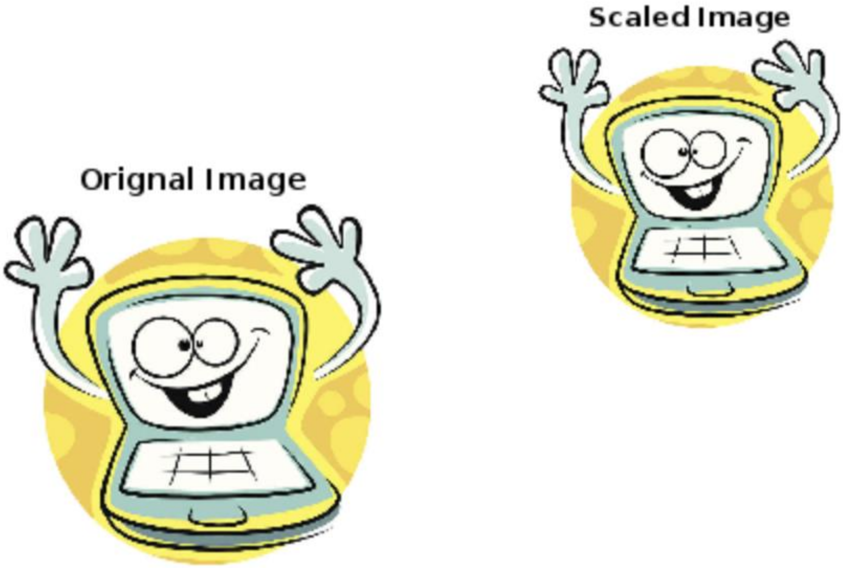
Problem Solve
Create a figure with two subplots using the subplot function. The first subplot displays the original image using 'imshow', and the second subplot displays the scaled image. The title function is used to provide titles for each subplot.

Code:

```
originalimage=imread("smile.jpg");
scalefactor=3.5;
scaledImage=imresize(originalimage,scalefactor);
figure;
subplot(1,2,1);
imshow(originalimage);
title('Orignal Image');
subplot(2,2,2);
imshow(scaledImage);
title('Scaled Image');
```



```
originalimage=imread("smile.jpg");
scalefactor=3.5;
scaledImage=imresize(originalimage,scalefactor);
figure;
subplot(1,2,1);
imshow(originalimage);
title('Orignal Image');
subplot(2,2,2);
imshow(scaledImage);
title('Scaled Image');
```



EXERCISE 3: TRANSLATION

Apply a translation to an image by shifting it 30 pixels to the right and 20 pixels downward. Display the original and translated images side by side.

Step 1: Read the original image using the 'imread' function
Replace 'input image.jpg' with the actual path and filename of your image.

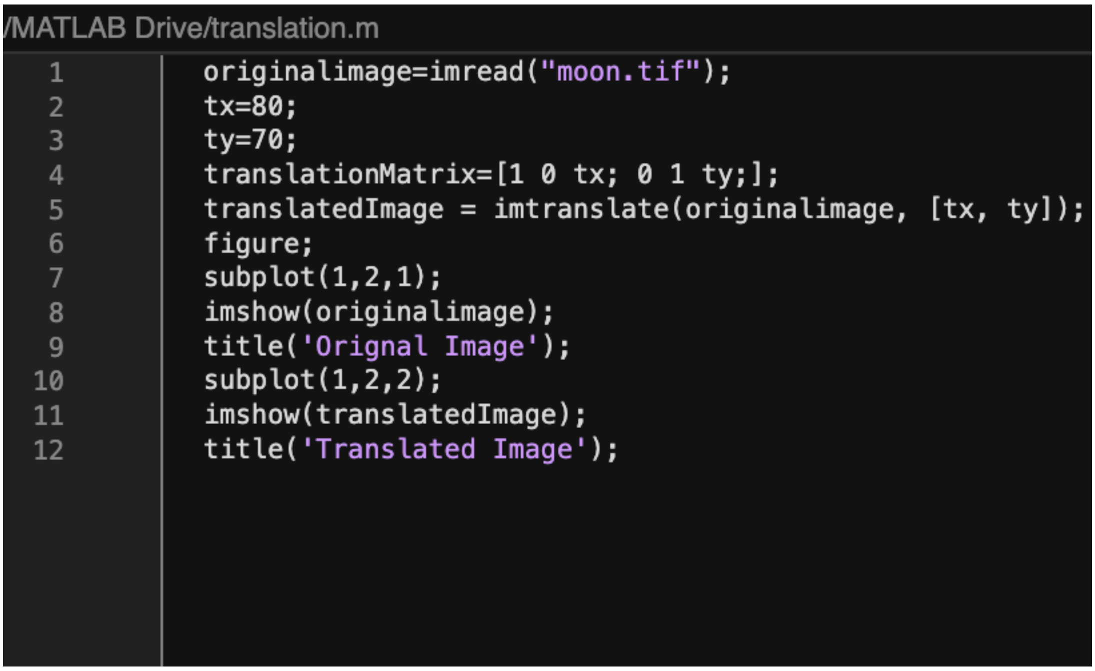
Step 2: Define the translation amounts in pixels.
translationMatrix = [1 0 tx; 0 1 ty];

Step 3: Create a 2x3 translation matrix using the translation amounts.
The translation matrix represents the translation transformation and specifies the shift in the x-axis (x) and y-axis (ty).

Step 4: Perform the translation using the 'imwarp' function
The 'imwarp' function applies the translation to the original image using the translation matrix. The affine2d' function creates an affine transformation obiect based on the translation matrix.

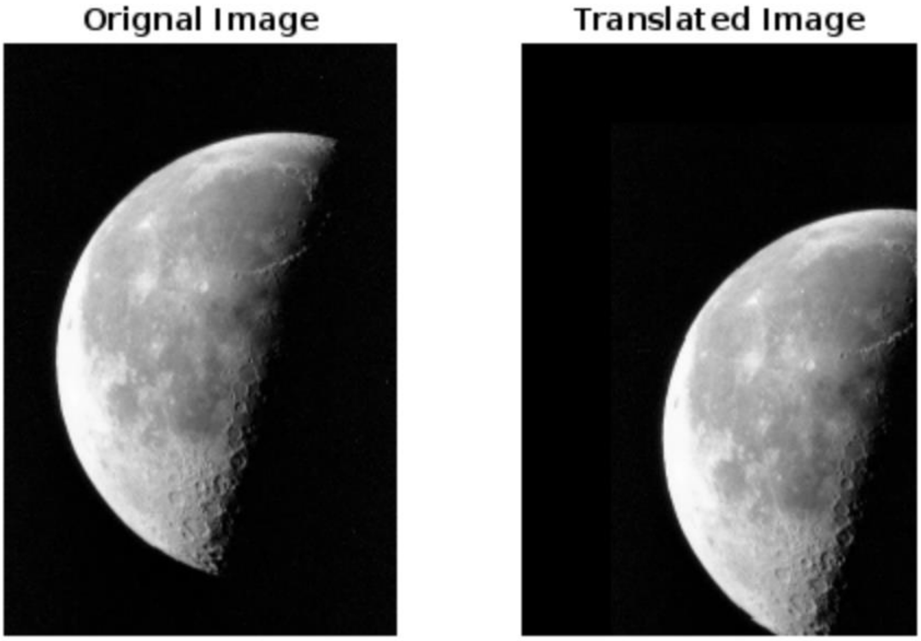
Code:

```
originalimage=imread("moon.tif");
tx=80;
ty=70;
translationMatrix=[1 0 tx; 0 1 ty;];
translatedImage = imtranslate(originalimage, [tx, ty]);
figure;
subplot(1,2,1);
imshow(originalimage);
title('Orignal Image');
subplot(1,2,2);
imshow(translatedImage);
title('Translated Image');
```



/MATLAB Drive/translation.m

```
originalimage=imread("moon.tif");
tx=80;
ty=70;
translationMatrix=[1 0 tx; 0 1 ty;];
translatedImage = imtranslate(originalimage, [tx, ty]);
figure;
subplot(1,2,1);
imshow(originalimage);
title('Orignal Image');
subplot(1,2,2);
imshow(translatedImage);
title('Translated Image');
```



Orignal Image

Translated Image

EXERCISE 4: SHEARING

Perform a shear transformation on an image by shearing it along the x-axis. Display the original and sheared images side by side.

Step 1: Read the original image using the imread function
Replace 'input image.jpg' with the actual path and filename of your
SKILL ATTAINMEN
i. Developing a sol shearing.
ii. Hands-on exper transformations tc it. Interpreting and images, compare
image.

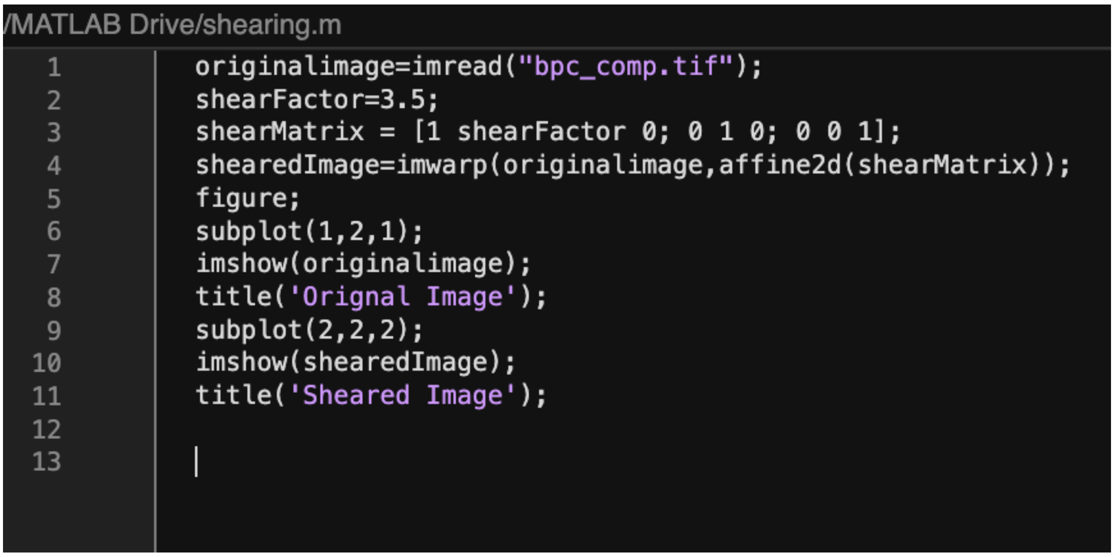
Step 2: Define the shear factor:
shearFactor = 0.5; % Shear factor along the x-axis

Step 3: Create a shear transformation matrix:
Problem Solve
The shear matrix represents the shear transformation and specifies the shear factor along the x-axis (shearFactor).
Test Cases an

Step 4: Perform the shear transformation using the imwarp function
The 'imwarp' function applies the shear transformation to the original image using the shear matrix. The 'affine2d' function creates an affine transformation obiect based on the shear matrix.

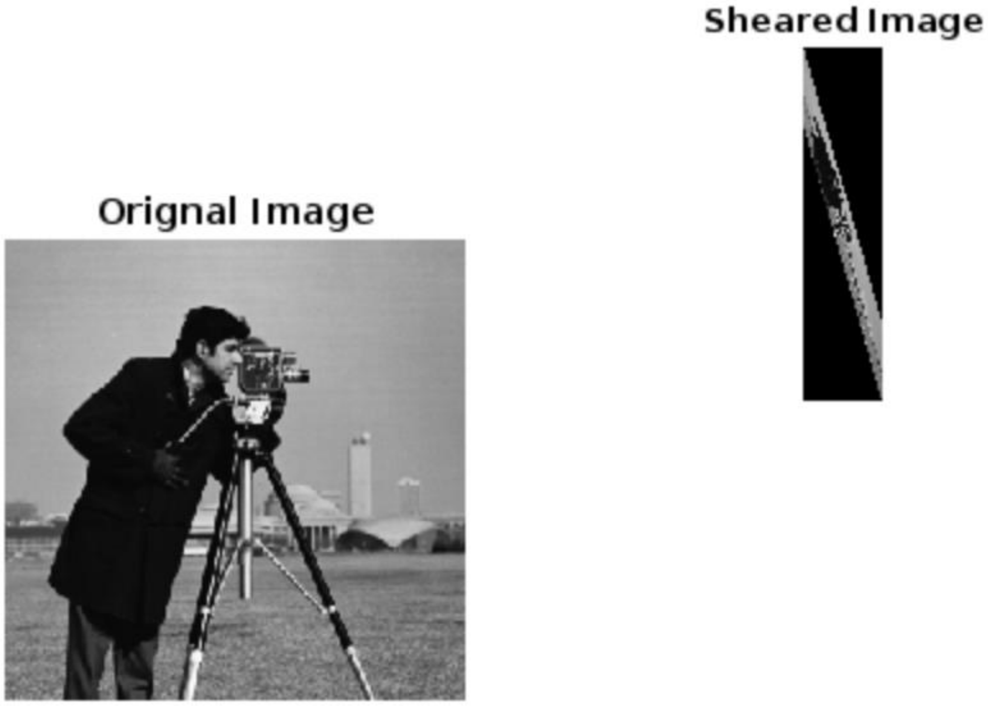
Code:

```
originalimage=imread("bpc_comp.tif");
shearFactor=3.5;
shearMatrix = [1 shearFactor 0; 0 1 0; 0 0 1];
shearedImage=imwarp(originalimage,affine2d(shearMatrix));
figure;
subplot(1,2,1);
imshow(originalimage);
title('Orignal Image');
subplot(2,2,2);
imshow(shearedImage);
title('Sheared Image');
```



/MATLAB Drive/shearing.m

```
originalimage=imread("bpc_comp.tif");
shearFactor=3.5;
shearMatrix = [1 shearFactor 0; 0 1 0; 0 0 1];
shearedImage=imwarp(originalimage,affine2d(shearMatrix));
figure;
subplot(1,2,1);
imshow(originalimage);
title('Orignal Image');
subplot(2,2,2);
imshow(shearedImage);
title('Sheared Image');
```



Sheared Image

Orignal Image

EXERCISE 5:

Combine multiple transformations (e.g., rotation, scaling, translation) on an image and display the final transformed image.

Step 1: Read the original image using the imread function
Replace 'input image.jpg' with the actual path and filename of your image.

Step 2: Define the transformation parameters

Step 3: Create the transformation matrices
The rotation matrix represents the rotation transformation, the scaling matrix represents the scaling transformation, and the translation matrix represents the translation transformation.

Step 4: Combine the transformations
combinedMatrix = translationMatrix * scalingMatrix * rotationMatrix;

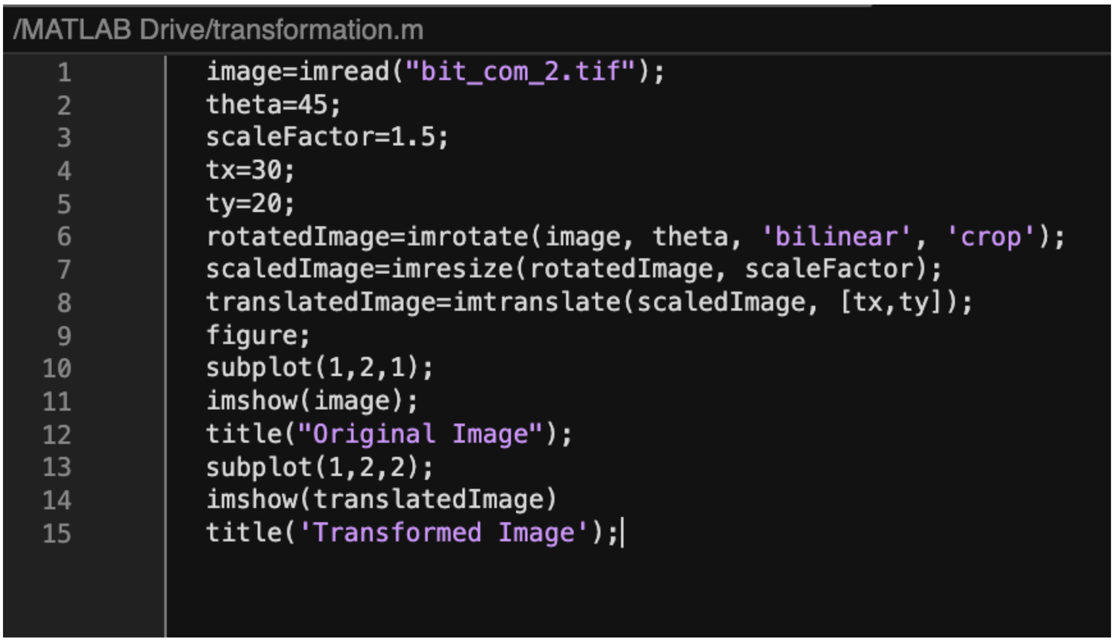
Step 5: Perform the combined transformation using the imwarp function
The 'imwarp' function applies the combined transformation to the original image using the combined matrix. The 'affine2d' function creates an affine transformation obiect based on the combined matrix.

Step 6: Display the original and transformed images side by side
Create a figure with two subplots using the subplot function. The first subplot displays the original image using 'imshow, and the second subplot displays the transformed image. The title function is used to provide titles for each subplot.

Code:

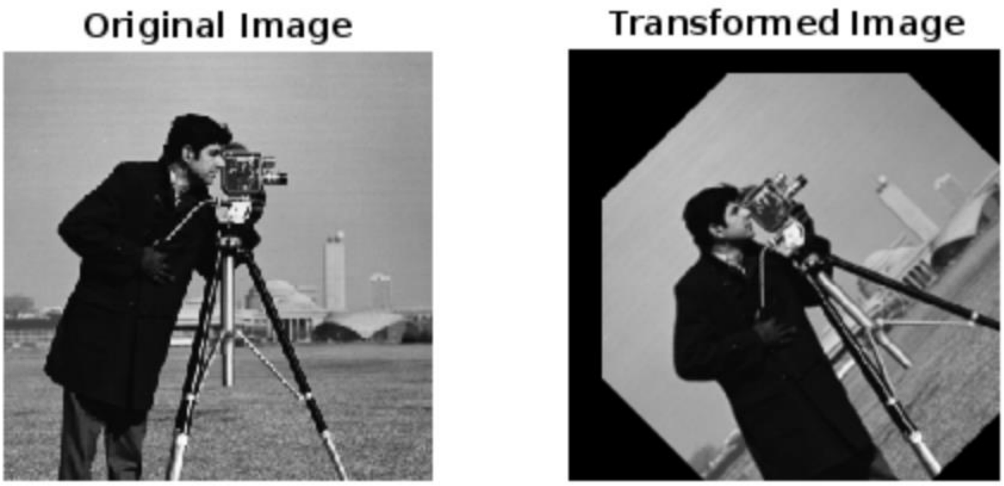
```
image=imread("bit_com_2.tif");
theta=45;
scaleFactor=1.5;
tx=30;
ty=20;
rotatedImage=imrotate(image, theta, 'bilinear', 'crop');
scaledImage=imresize(rotatedImage, scaleFactor);
translatedImage=imtranslate(scaledImage, [tx,ty]);
figure;
subplot(1,2,1);
imshow(image);
title("Original Image");
subplot(1,2,2);
```

```
imshow(translatedImage)
title('Transformed Image');
```



/MATLAB Drive/transformation.m

```
image=imread("bit_com_2.tif");
theta=45;
scaleFactor=1.5;
tx=30;
ty=20;
rotatedImage=imrotate(image, theta, 'bilinear', 'crop');
scaledImage=imresize(rotatedImage, scaleFactor);
translatedImage=imtranslate(scaledImage, [tx,ty]);
figure;
subplot(1,2,1);
imshow(image);
title("Original Image");
subplot(1,2,2);
imshow(translatedImage)
title('Transformed Image');
```



Original Image

Transformed Image