
```

%NAME: Krishna Kodali
%INST: IIT Bhubaneswar
%DATE: 29/11/2020
%CATEGORY: BTech
%BRANCH: Computer Science
%Roll Number: 17CS01008
%Assignment-06
%Affline Transformations
%Removing previous Buffer
clc;clear;close all;

%Question-1: Take an image and apply affline transformation.

Image = imread('cameraman.tif');
Image = double(Image)./255;
n = size(Image,1);m = size(Image,2);
I1 = affine_transformation(Image,90,1,1,0,-m);
I2 = affine_transformation(Image,0,0.5,0.5,0,0);
I3 = affine_transformation(Image,0,1,1,n/2,m/2);
subplot(2,2,1);imshow(Image);title('sample img');
subplot(2,2,2);imshow(I1);title('90 ACW rotation');
subplot(2,2,3);imshow(I2);title('Scaled version with same origin');
subplot(2,2,4);imshow(I3);title('Origin shifted center');

function Img = affine_transformation(img,theta,sx,sy,tx,ty)
    n = size(img,1);    m = size(img,2);
    V_M = [[cosd(theta); sind(theta); 0],[-sind(theta);
    cosd(theta) ;0],[0;0;1]];
    S_M = [[sx ;0;0],[0;sy;0],[0;0;1]];
    T_M = [[1;0;0],[0;1;0],[tx;ty;1]];
    Img = zeros(size(img));
    for x=1:n
        for y=1:m
            A = V_M*S_M*T_M;
            cord_mat = (inv(A))*[x;y;1];
            pix = bilinear_interpolation(img,cord_mat(1),cord_mat(2));
            Img(x,y) = pix;
        end
    end
end

function pix = bilinear_interpolation(img,x,y)
    n = size(img,1);m=size(img,2);
    x1 = floor(x); y1 = floor(y);
    x2 = ceil(x); y2 = ceil(y);
    if(x1<=0)
        x1=1;
    end
    if(y1<=0)
        y1=1;
    end
    if(x1>=n)
        x1=n-1;

```

```

end
if(y1>=m)
    y1=m-1;
end
if(y2<=0)
    y2=1;
end
if(x2<=0)
    x2=1;
end
if(y2>m)
    y2=m;
end
if(x2>n)
    x2=n;
end
if(y2==y1&&x2~=x1)
    pix = ((x2-x)/(x2-x1))*img(x1,y1)+((x-x1)/(x2-x1))*img(x2,y1);
elseif(x2==x1&&y2~=y1)
    pix = ((y2-y)/(y2-y1))*img(x1,y1)+((y-y1)/(y2-y1))*img(x1,y2);
elseif(x2==x1&&y2==y1)
    pix = img(x1,y1);
else
    pix_h1 = ((y2-y)/(y2-y1))*img(x1,y1)+((y-y1)/(y2-
y1))*img(x1,y2);
    pix_h2 = ((y2-y)/(y2-y1))*img(x2,y1)+((y-y1)/(y2-
y1))*img(x2,y2);
    pix = ((x2-x)/(x2-x1))*pix_h1+((x-x1)/(x2-x1))*pix_h2;
end
end
end

```

sample img



90 ACW rotation



Scaled version with same origin



Origin shifted center



Published with MATLAB® R2020a