# CS663: Digital Image Processing - Homework 1

Harsh | Pranav | Swayam

August 23, 2024

## 1 Homework 1 - Question 6

### (a)

The MATLAB code for this question is as follows:

```
n = 12;
choice = 1; %1 for ginput, else feed directly

if choice == 1
    for i = 1:n
        figure(1); imshow(im1/255);
        [x1(i),y1(i)] = ginput(1);
        figure(2); imshow(im2/255);
        [x2(i),y2(i)] = ginput(1);
    end
end

disp("Figure 1 goi1 selected points are:")
disp([x1, y1])
disp("Figure 2 goi2 selected points are:")
disp([x2, y2])
```

After running the above code, we get the following selected points for the two images:

```
Terminal Out:
Figure 1 goi1 selected points are:
202.3677 512.4066 277.6984 179.0214 316.9202 484.0798 264.9358 118.3210
424.9358 373.8852 367.3482 201.1226 256.7646 294.7412 251.4728 218.4767
219.4105 301.2782 175.8307 246.1809 178.3210 19.5661 210.0720 243.0681

Figure 2 goi2 selected points are:
236.6089 570.6167 313.8074 212.9514 353.0292 537.3093 299.1770 150.6946
467.2704 414.3521 406.2588 162.5233 277.9319 313.7296 271.0837 238.0875
238.7101 321.5117 192.6401 266.4144 191.3949 32.0175 230.3054 255.2082
```

### (b)

The MATLAB code for this question is as follows:

```
P2 = [x2; y2; ones(1,n)];
P1 = [x1; y1; ones(1,n)];
A = P2*P1'*inv(P1*P1'); % Moore-Penrose pseudo-inverse

disp("Matrix A:")
disp(A);
```

After running the above code, we get the following matrix $A$:

```
Terminal Out:
Matrix A:
    1.1047   -0.0012    1.2435
   -0.0032    1.0283   12.6563
         0         0    1.0000
```

Hence, the image `goi1` was transformed using the rotation, scaling, shearing and translation transformations to get the image `goi2`. This can be mathematically achieved using the matrix $A$.

### (c)

The MATLAB code for this question is as follows:

```
im3 = zeros(size(im1));

for i = 1:H
    for j = 1:W
        dxy = A\[j i 1]';
        xx = round(dxy(1));
        yy = round(dxy(2));

        if xx > 0 && xx <= W && yy > 0 && yy <= H
            im3(i,j) = im1(yy,xx); % nearest neighbor
        end
    end
end

figure(1); imshow(im1/255);
figure(2); imshow(im2/255);
figure(3); imshow(im3/255);

% Save all the images
saveas(figure(1), 'nn_im1.png');
saveas(figure(2), 'nn_im2.png');
saveas(figure(3), 'nn_im3.png');
```

The images obtained after the nearest neighbor interpolation are as follows:

Figure 1: `goi1`

Figure 2: `goi2`

Figure 3: Interpolated Image (Nearest Neighbor)

### (d)

The MATLAB code for this question is as follows:

```
im4 = zeros(size(im1));

for i = 1:H
    for j = 1:W
        dxy = A\[j i 1]';
        xx = round(dxy(1));
        yy = round(dxy(2));

        if xx > 0 && xx <= W && yy > 0 && yy <= H
            w1 = (xx+1-dxy(1))*(yy+1-dxy(2)); % bilinear
            w4 = (dxy(1)-xx)*(dxy(2)-yy);
            w3 = (yy+1-dxy(2))*(dxy(1)-xx);
            w2 = (xx+1-dxy(1))*(dxy(2)-yy);
            im4(i,j) = im1(yy,xx)*w1+im1(yy+1,xx)*w2+im1(yy,xx+1)*w3+im1(yy+1,xx+1)*w4;
        end
    end
end

figure(1); imshow(im1/255);
figure(2); imshow(im2/255);
figure(3); imshow(im4/255);

% Save all the images
saveas(figure(1), 'bilinear_im1.png');
saveas(figure(2), 'bilinear_im2.png');
saveas(figure(3), 'bilinear_im4.png');
```

The images obtained after the bilinear interpolation are as follows:

Figure 4: goi1

Figure 5: goi2

Figure 6: Interpolated Image (Bilinear)

## (e)

For estimating the affine transformation matrix, we need at least three non-collinear points.

If all the selected points are collinear, then the following issues may arise:

1. The system of equations we formed to solve for the affine transformation matrix becomes *underdetermined*, that there is not enough independent equations to uniquely solve for all parameters.

2. The computation will not capture any affine transformation that includes rotations, scaling, or translations.

3. The affine transformation cannot be estimated accurately because the equations would not span the required two-dimensional space. It essentially becomes a problem in one-dimensional space along the line.