1. Geometric transformations

Approach:

This transformation needs three steps:

(1) Transfer the original coordinate system Ⅰ to the coordinate system Ⅱ with the center of the image as the origin.

(2) Rotation by theta degree in the coordinate system II.

(3) Transfer II to I back.

Set the width of the original picture=w, the height=h

Thus, the transformation matrix is:

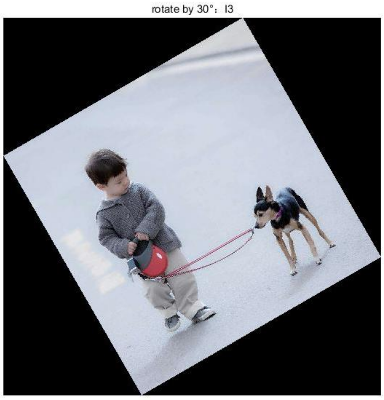
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -0.5w & 0.5h & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0.5W & 0.5H & 1 \end{bmatrix}$$

$$W = w*\cos\theta + h*\sin\theta$$

$$H = w*\sin\theta + h*\cos\theta.$$

Result:

Figure 1: SrcImage   Figure 2: rotated by 30 degree   Figure 3: rotated by 60 degree



2. Planar projective distortion

i)

First, using ginput( ) to pick four points, top left, top right, bottom left, bottom right. Obtain the height and width of the new rectangle from the original quadrilateral.

Then get the original vertex:

y=[dot(1,1) dot(2,1) dot(3,1) dot(4,1)];

x=[dot(1,2) dot(2,2) dot(3,2) dot(4,2)];

and the new vertex:

Y=[dot(1,1) dot(1,1) dot(1,1)+h dot(1,1)+h];

X=[dot(1,2) dot(1,2)+w dot(1,2) dot(1,2)+w];

In my program, I chose a new rectangle, and we can make it any other way. Then

Simultaneous solution of equations, coefficients of equations:
A=[x(1) y(1) 1 0 0 0 -X(1)*x(1) -X(1)*y(1);
0 0 0 x(1) y(1) 1 -Y(1)*x(1) -Y(1)*y(1);
x(2) y(2) 1 0 0 0 -X(2)*x(2) -X(2)*y(2);
0 0 0 x(2) y(2) 1 -Y(2)*x(2) -Y(2)*y(2);
x(3) y(3) 1 0 0 0 -X(3)*x(3) -X(3)*y(3);
0 0 0 x(3) y(3) 1 -Y(3)*x(3) -Y(3)*y(3);
x(4) y(4) 1 0 0 0 -X(4)*x(4) -X(4)*y(4);
0 0 0 x(4) y(4) 1 -Y(4)*x(4) -Y(4)*y(4)];
The solution of the four points is also the global transformation coefficient.
Then we can get:
a=fa(1);b=fa(2);c=fa(3);
d=fa(4);e=fa(5);f=fa(6);
g=fa(7);h=fa(8);
rot=[d e f;
a b c;
g h 1];
Then transform the image’s left, right, left, lower right point. We can compute the height and width of the transformed image. Gets the offset from the negative axis in the y and x direction respectively. Then reverse search the original image from the transformed image to avoid voids. Get the coordinates in the original image, because [YW XW W]=fa*[y x 1], so here is [YW XW W],W=gy+hx+1; It's the same thing as solving the equation [pix(1)*(gy+hx+1) pix(2)*(gy+hx+1)]=[y x], finding y and x, and finally pix=[y x]; Use Nearest neighbor interpolation to get points: imgn(i+delta_y,j+delta_x)=img(round(pix(1)),round(pix(2))); The following pictures shows the input image and the output image after distortion removal respectively.


ii)The same way as above, set: rot1=[1 0 2;0 -1 2;0 0 1]; rot2=[-2 0 5; 0 2 5; 0 0 1]; and the newly generated perspectives


3. Camera Calibration

Approach: First, detect checkerboards in image. Generate world coordinates of the corners of the squares. Then, calibrate the camera. Visualize pattern locations.

Result analysis:

4. Fourier Transforms

i)

Approach:

Result analysis:

fig1. image A fig2. magnitude-only reconstruction fig3. phase-only reconstruction.

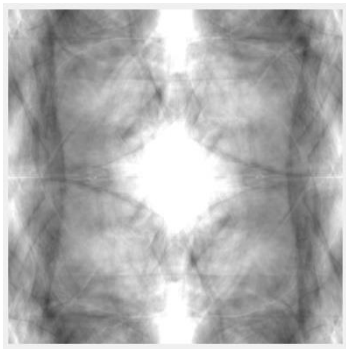
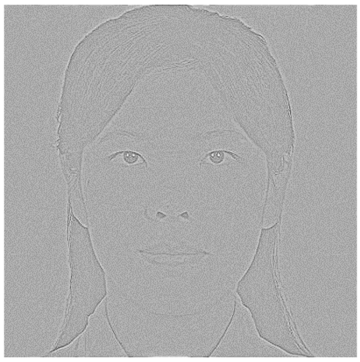
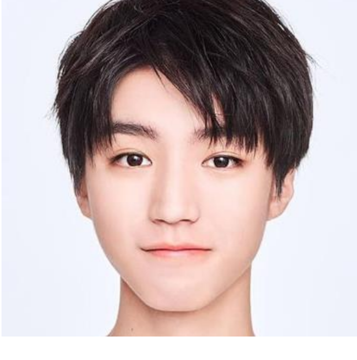
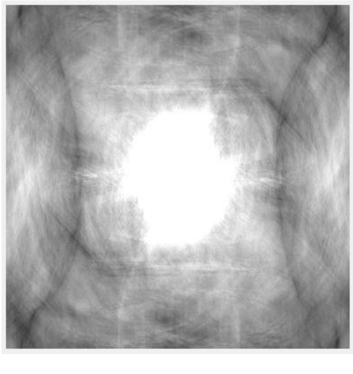
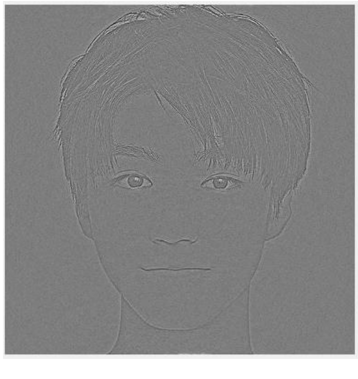
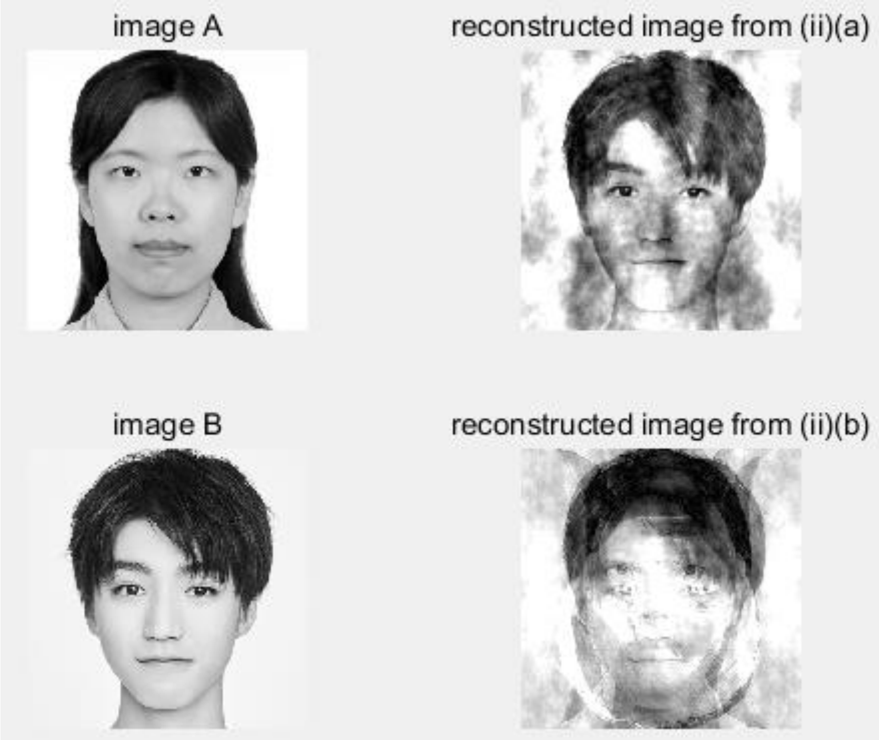




fig4. image B fig5. magnitude-only reconstruction fig6. phase-only reconstruction.

ii)



Result analysis: According to the comparison of experimental results, phase spectrum plays a more important role in the reconstruction.

From (i), Test image reconstructed from MAGNITUDE spectrum only. The phase values determine the shift in the sinusoid components of the image. With zero phase, all the sinusoids are centered at the same location and you get a symmetric image whose structure has no real correlation with the original image at all. Being centered at the same location means that the sinusoids are a maximum at that location, and is why there is a big white patch in the middle of Figure2. The phase-only reconstruction preserve features because of the principle of phase congruency. At the location of edges and lines, most of the sinusoid components have the same phase. So we can see that the phase information is most important.

And intensity values of HIGH frequency (edges, lines) pixels are comparatively more than LOW frequency pixels. Changing the magnitude of the various component sinusoids changes the shape of the feature. When you do a phase-only reconstruction, you set all the magnitudes to one, which changes the shape of the features, but not their location. In many images the low frequency components have a magnitude higher than the high frequency components, so phase-only reconstruction does look like a high-pass filter. In short, phase contains the information about the locations of features.