Problem 1 (27 points)

We want to perform calibration of an old iPod camera. We are interested only in the intrinsic parameters \( \alpha_u, \alpha_v, u_0, v_0 \) and the pixel size. Table 1 gives the data for 10 points, where \((r, c)\) represent the pixel coordinates and \((x, y, z)\) represent the coordinates of the 3D points in the camera reference frame. They are measured in millimeter.

\[
\begin{bmatrix}
    r & c & x & y & z \\
    428 & 290 & -28 & -32 & 264 \\
    427 & 403 & -28 & -4 & 264 \\
    812 & 310 & 156 & -32 & 482 \\
    808 & 658 & 156 & 94 & 482 \\
    252 & 333 & -133 & -20 & 701 \\
    560 & 336 & 27 & -20 & 701 \\
    498 & 427 & -5 & 28 & 701 \\
    559 & 427 & 27 & 52 & 701 \\
    497 & 472 & -5 & 28 & 701 \\
    558 & 473 & 27 & 52 & 701
\end{bmatrix}
\] (1)

1) What is the minimum number of points we need to solve this calibration problem? (Don’t say six)
2) Calculate the intrinsic parameters \( \alpha_u, \alpha_v, u_0, v_0 \).
3) Obtain the pixel size \( s_x, s_y \).

Problem 2 (10 points)

We want to implement the Hough transform to detect the vertical lines in an image. By redefining the origin of the reference frame at the top left corner of the image, a vertical line corresponds to \( \theta = \pi/2 \). The accumulator result for detecting vertical lines for the images of figures 1 and 3 are shown in figures 2 and 4, respectively.

1) Why is the accumulator a 1D function in this case?
2) Write code to implement the Hough transforms to detect vertical lines in the image shown in figure 5, where a line corresponds to 10 votes at least. Show the result of the accumulator, you can show the numbers or use function stem as shown in figure 2 and 4.
3) Show the Hough space \((\theta, \rho)\) corresponding to the vertical lines in the image of figure 5.

The image of figure 5 can be obtained as follows:
\[
I=\text{zeros}(25,25); I(2:22,5)=1; I(4:15,12)=1; I(8:15,7)=1; I(8:15,20)=1; I(22,10:25)=1;
\]

Problem 3 (8 points)

In this problem we want to use neural network to classify handwritten numbers. We are restricted to number 8 and 9 as shown in figure 6. The images are resized to 25 by 25 pixels.

1) Write code to implement a neural network to classify the numbers. The maximum number of features you can use is 625. Draw the boundary line with the data if you use one or two features. You need to provide the numbers of the weights and bias in table or graph format if you use more than three features.
2) Test your code with additional handwritten numbers (8s and 9s only). Discuss your results.
Problem 4 (30 points)

Figure 7 shows the Hough transform of a simple geometric shape. Plot the geometric shape in a figure similar to 8. The origin is defined at the top left corner.
Fig. 5. Image for problem 2

Fig. 6. Figure for problem 3

Fig. 7. Hough transform of a geometric shape

Fig. 8.