Q1

1. H: Hue. S: Saturation. V: Value

For red and pink the saturation is the main difference.

1. Scale Invariant Feature Transform.

The eigenfaces correspond to the Eigen-vectors.

Q2

(1)

The size of the camera matrix P is 3 by 4.

K: the intrinsic matrix that contains the information of the camera including the focal length, optical center and distortion.

R: the rotation matrix that describe the transactions.

t: translation victor to describe the difference between the world coordinate system and camera coordinate system.

(2)

The size of the matrix K is 3 by 3.

The calibration parameters are α, β, s, u0 and v0.

K has 5 degrees of freedom.

6 point correspondences are minimally needed.

(3)

First Sobel then Gaussian: all the edges will be detected including some noise, and then all the lines will be blurred because of the Gaussian filter.

First Gaussian then Sobel: the image will be blurred by the Gaussian filter, so when applying Sobel filter there will be less noise because some noise are already smoothed by the Gaussian filter.

The second one often produces a better result because there will be less noise. This is also the one I preferred.

\*\* *The original question is not answerable.*

(S\*G)\*I will be better since it is more efficient.

(4)

The first one (vertical Sobel filter) will detect edges on vertical direction (compute the horizontal components of the image gradient), the second one (horizontal Sobel filter) will detect edges on the horizontal direction (compute the vertical components of the image gradient ).

The gradient direction (orientation of edge normal) is given by:



where f is the function to apply the filter to the image.

Q3

Median filter: (中值，排序后取中间值)

3 3 4

6 7 7

7 7 7

Vertical edge filter:

2 -398 4

1 105 2

-1 -95 5

Q4

(1)

Assume we want to classify the data to k groups.

Step 1: randomly pick k points as cluster centers.

Step2: for each point calculate the distances to each cluster center and assign the point to the closest center.

Step 3: recalculate the center of each cluster after the assignment.

Step 4: repeat step 2 and step 3 until there is no change (no points are reassigned).

(2)

Step 1: randomly pick 1 point as cluster centers.

Step2: for each point calculate the distances to each cluster center and assign the point to the closest center.

Step 3: recalculate the center of each cluster after the assignment.

Step 4: repeat step 2 and step 3 until there is no change (no points are reassigned).

Step 5: If there are more than M points assigned to the same center, randomly pick another cluster center.

Step 6: repeat step 2 to step 5 until there are no points are reassigned and all clusters have no more than M points.

Q5

(1)

Parameters: threshold t which limit how much a part of the image is close to a circle.

Step 1: initialize the radius r of the circle to 1.

Step 2: initialize the center c of a circle to [1, 1].

Step 3: generate a matrix f with size (2\*r+1, 2\*r+1). Set the element on the circle of with center [r, r] and radius r is set as 1 in the matrix. Use this as a circle filter to detect the circle in the image.

Step 4: for point [x, y] of the image starting from [1, 1], select the r\*r matrix centered on the pixel, calculate the AND with the matrix f, note it as matrix res.

Step 5: calculate the ones in the matrix res and the ones in the matrix f, compare if there are enough ones in the matrix res according to the threshold t.

Step 6: If there are enough ones in the matrix res, it means there are enough pixels in the image that are close enough to a circle. Set all the pixels apart from the circle to 0 and return the image.

Step 7: If there are not enough ones in the matrix res, increase the r to r+1 and reassign the center of the circle to [r, r].

Step 8: repeat step 4 to 7 until the filter is out of one border, then move to the next pixel until iterated all the pixels of the image.

(2)

Since the filter is a circle, with the AND operation, only pixels that are close enough to a circle will result to 1. For the line, there will be a few pixels that can result to 1 with the filter (maybe 2 pixels or less since there can be at most 2 intersection points for a line and a circle), which is not close enough to a circle, so the algorithm will ignore the line.

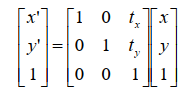
Q6

Size: 3 by 3 with 8 independent variables.

(2)

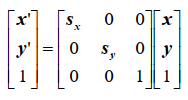
Yes. The matrix should be a transaction matrix that contains the rotation, translation, scaling and affine.

For translation, the matrix is like:

.

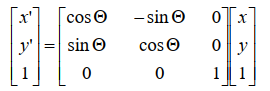
Calculate the tx and ty.

For scaling, the matrix is like:

.

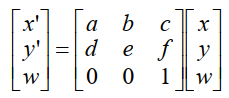
Calculate the sx and sy.

For rotation, the matrix is like:

.

Calculate the cosθ and sinθ.

For affine, the matrix is like:

.

Calculate the a, b, c, d, e, f (the w should be 1).

To combine them together, multiply all the transaction matrix.

(3)

Yes, should be (2, 4).

Q7