CS512 - Assignment 1

Mohammadreza Asherloo
Department of Mechanical, Materials and Aerospace Engineering
Illinois Institute of Technology

October 22, 2020

Question 1.a:

There is an algorithm for corner detection:

1. We need to find the correlation matrix of gradients in local window.
2. Then find eigen values of this correlation matrix
3. If the eigen values are large enough, we will have a corner in that window

For finding principal directions, we need to plot the orientation histogram and count the directions.

Question 1.b:

We need to find vector V so that PCA vector projection onto V will be minimized.

Question 1.c:

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 0 & 2 \\ 0 & 3 \\ 0 & 4 \\ 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} = \begin{bmatrix} 4 & 6 \\ 6 & 44 \end{bmatrix}$$

Question 1.d:

We can detect a corner if multiplication of eigen values are larger than the threshold.

Question 1.e:

With this algorithm we can delete the windows that contain the same corner and keep the window with the most accurate location of the corner.
We select all the windows that have the multiplication of eigen values more than the threshold, then sort them in decreasing order, select the top of the list and delete all other windows.

Question 1.f:

In Harris corner detection we have a new variable named (cornerness) and we compute this variable based on the correlation matrix and we have a corner in that window where the cornerness is high.

Question 1.g:

We need to convert all the points (x_i) to project the gradient at (x_i) onto $(x_i - p)$ and the best p is that minimizes the sum of $(x_i - p)$.

Question 1.h:

We will split each point to cells with overlap. Then create orientation histogram in each cell with any of the known methods and after creating this histogram we will create the histogram for the entire image.
Desired properties are translation, scale, rotation and illumination invariance

Question 1.i:

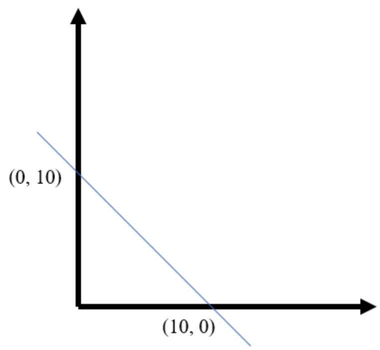
There are some steps to do SIFT feature extraction:
Potential location for finding features then we localize the feature keypoints. And then we will assign orientation to this keypoints and after that we will describe them as high dimensional vectors.

Question 2.a:

Because of 2 reasons. Firstly, we don't know the range of the slope. Secondly, we can-not represent the vertical lines.

Question 2.b:

$a = cos(\theta)$
$b = sin(\theta)$
$c = -10$



Question 2.c:

Each point in parameter space is a wave lines and for finding the line equation, we need to find the intersection of these lines.

Question 2.d:

Each point represents a line in the parameter space and for detecting the line we need to find the intersection of these lines which gives us the common parameters of these points.

Question 2.e:

If we use larger bin size, it would be more efficient but we will lose the ability of having a accurate localization.

Question 2.f:

If we know the normal for the desired line, we can have a vote on parameter space and narrow down the vote to a small section of the curve in parameter space.

Question 2.g:

We will have three dimensions. radius of circle and the coordination of the circle's origin.

Question 3.a:

The disadvantage of this model is the error objective that is used which computes the vertical distance between the points and the line. This problem exacerbates when the slope of the line gets close to 90 degrees.

Question 3.b:

$$ax + by + c = 0 \Longrightarrow e^T p = ax + by + c \Longrightarrow e = \begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}$$

Question 3.d:

When the point is on the line, the result of $e^T p$ will be zero and if the point is not on the line, the result of $e^T p$ will show the geometric distance between the point and the line. We need to minimize sum of these distances between points and the line to find the best line fit.
The equation is SL = 0 which S is the correlation matrix of the points and L is the line coefficients.

Question 3.e:

$ax^2 + bxy + cy^2 + dx + ey + f = 0$ and the constraint for these coefficients is $b^2 - 4ac < 0$

Question 3.f:

$L = \lambda_1 S^{-1} cL$ and we need to find the eigen vector of $S^{-1}c$ belonging to its negative eigen value which will be the answer.
Points closer to short axis are more important because based on the distance equation which is $q_i = \frac{d_i}{d_i * r_i}$ if r is smaller, distance will be larger.

Question 3.g:

$E(L) = \sigma \frac{e^T L}{\nabla e^T L}$ should be minimized. But the complication is that there isn't an explicit solution for this equation and this should be solved with the iterative numerical solution.

Question 3.h:

$E[\phi(s)] = \int_{\phi(s)} (\alpha(s) E_{cont} + \beta(s) E_{curv} + \gamma(s) E_{img})\, ds$
which E_{cont} is the variable shows the continuity of the curve, E_{curve} is the variable shows the curvature of the line and E_{img} is the variable shows the image energy. And $\alpha(s)$, $\beta(s)$ and $\gamma(s)$ are the coefficient of different energy terms.

Question 3.i:

Continuity energy is calculated by step forward discreet gradient which is:
$E_{cont} = |\frac{d\phi}{ds}|^2$
Curvature energy is calculated by:
$E_{curv} = |\frac{d^2\phi}{ds^2}|^2$

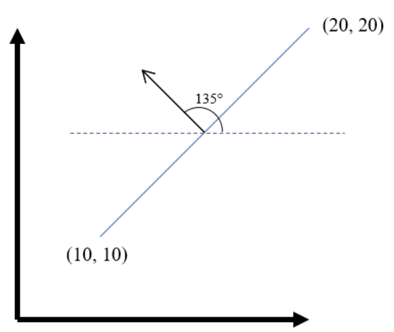
Question 3.i:

If we want to relax any of the energies in the function, we can put its coefficient to be zero. For example if we let $\alpha(s) = 0$ the continuity energy will lose its effect.

Question 4.a:

$$\text{S} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} = \begin{bmatrix} 4 & 11 \\ 11 & 25 \end{bmatrix}$$

Question 4.b:



$\theta = 135 \Longrightarrow -\frac{\sqrt{2}}{2}x + \frac{\sqrt{2}}{2}y = 0 \Longrightarrow y = x$

$n_x = cos(\theta) = cos(135) = -\frac{\sqrt{2}}{2}x$
$n_y = sin(\theta) = sin(135) = \frac{\sqrt{2}}{2}x$

Question 4.c:

$x + 2y + 3 = 0 \Longrightarrow x = 2 \Longrightarrow y = -2.5$

Question 4.d:

$d = xCos(0) + ySin(0) = x + 0 = x = 1$

Question 4.e:

$$S = \begin{bmatrix} 1 & 3 \\ 2 & 4 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 3 & 4 & 1 \end{bmatrix} = \begin{bmatrix} 10 & 14 & 4 \\ 14 & 20 & 6 \\ 4 & 6 & 2 \end{bmatrix}$$

Question 4.f:

$d(p, f) = |p - x^*| = \frac{|f(p)|}{|\nabla f(x^*)|} = \frac{1}{2}$

Question 4.g:

$d(p, f) = |p - x^*| = \frac{|f(p)|}{|\nabla f(p)|} = \frac{1}{2}$

Question 4.h:

$E_{cont} = |\sqrt{(3-2)^2 + (4-3)^2}|^2 = 2$
$E_{curv} = |\sqrt{(3 - 2 * 2 + 1)^2 + (4 - 3 * 2 + 2)^2}|^2 = |\sqrt{0^2 + 0^2}|^2 = 0$

Question 4.i:

β should be zero for a tight fitting on a corner.

Question 5.a:

Outliers are the points with the longest distance between them and the model in robust estimation model. They make the initial model is inaccurate and makes the model to the wrong direction.

Question 5.b:

$E(\theta) = \int \alpha_\sigma(d(x_i, \theta))$
This is the general model and if we put $\alpha_\sigma = x^2$ we will have the mean square error function.

Question 5.d:

$estimator = \frac{x}{x+\sigma} = \frac{1}{1+1} = \frac{1}{2}$

Question 5.e:

Algorithm:

1. draw n points from the data set
2. fit a model on these points
3. then find the inliers which means points with a distance of smaller that a threshold
4. fit the model again on these points
5. update the parameters (k: number of trials, t: distance threshold)

We need to draw the largest number of points with smallest error

Question 5.f:

Parameters:
n: number of points drawn for each evaluation
d: minimum number of points needed to estimate the model
k: number of trials $= \frac{log(1-p)}{log(1-w^n)}$
t: distance threshold
w: probability of a point to be an inlier
p: probability of having at least a trial without any outlier

Question 5.g:

$k = \frac{log(1-p)}{log(1-w^n)} = \frac{log(1-0.99)}{log(1-0.9^2)} = \frac{-2}{-0.72} = 2.78 \Longrightarrow k = 3$

Question 6.a:

There are two objectives:

1. distinguish the object in the picture from the background
2. find contours of objects in the picture

Question 6.b:

Agglomerative approach:

1. put each pixel in different cluster
2. merge clusters with small distance
3. continue until clusters are satisfactory

Divisive approach:

1. put each pixel in one cluster
2. split clusters to distant clusters
3. continue until clusters are satisfactory

Question 6.c:

We initiate with a number of random cluster centers then assign pixels to each cluster based on their distance to each cluster. After that calculate the center of the clusters again and assign the pixels again as well. Continue until the change of centers is smaller than a threshold.

Question 6.d:

In graph cut we don't have a cost of size of the cluster created with cutting the links and we only cut the links based on the similarity of their nodes but in normalized cut we take the cost of creating the cluster into account based on the size of the cluster.
It is necessary to use normalized cut because with graph cut we will eventually cut a single node as a result of minimizing the cost.

Question 6.e:

Similarity matrix: $\begin{bmatrix} W_{1,1} & . & . & . & W_{1,100} \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ W_{100,1} & . & . & . & W_{100,100} \end{bmatrix}$

Weighted degree of matrix: $\begin{bmatrix} d_1 & . & . & . & . & . \\ . & d_2 & . & . & . & . \\ . & . & d_3 & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & d_{100} \end{bmatrix}$

Question 6.f:

We assign 1 for every node that we take and assign -1 for every other node. In this way, if a node is 1, it belongs to cluster A and if it is -1, it belongs to cluster B. To determine the cut we the following formula:

$N_{cut}(A, B) = \frac{cut(A,B)}{vol(A)} + \frac{cut(A,B)}{vol(B)}$

Question 6.g:

We need to minimize the $N_{cut}(x) = \frac{y^T(D-W)y}{y^T Dy}$ so that $y^T D1 = 0$

Question 6.h:

The solution is: $(D - W)y = \lambda Dy$ which yields two eigen vectors. The first eigen vector belonging to $\lambda = 0$ and the second smaller eigen vector which is the solution.

Question 6.i:

In this approach we map the image to lower dimensional vectors using principal component analysis and after that we measure the similarity with templates in lower dimensional space. The disadvantage of this approach is that it is less sensitive to small changes in the picture.

Question 6.j:

The idea is that we detect points of interest and extract the features of the image with any desired method, put these features into clusters and create a code word for them, compute the distribution of code words in each class then classify the image using these distribution of code words.

The disadvantage of this approach is that it does not take the relationship between parts into account so if we have a picture of separated parts, it will be falsely classified.