Homework 4 Questions

Instructions

- 4 questions.
- Write code where appropriate.
- Feel free to include images or equations.
- **Please use only the space provided and keep the page breaks.** Please do not make new pages, nor remove pages. The document is a template to help grading.
- If you really need extra space, please use new pages at the end of the document and refer us to it in your answers.

Questions

Q1: Imagine we were tasked with designing a feature point which could match all of the following three pairs of images. Which real world phenomena and camera effects might cause us problems? Use the OpenCV function $cornerHarris$ to investigate.

RISHLibrary — Chase — LaddObservatory

A1: There are several things to consider when creating feature points from image. The intensity and position of image pixels can vary according to position, direction, and angle of taking picture.

For example, in LaddObservatory, I can see that size of object changes. Since one photo is the magnified view of another photo, points outside the range of photo 1 may appear in photo 2.

In RISHLibrary photo, I can see that the direction of taking photos and intensity change. When creating feature point, it will be difficult to obtain pair of corresponding points if direction of gradient around the point is not adjusted or feature vector is not normalized.

In Chase photo, the direction of taking photo is also different, and photo 2 is unclear due to hand shake. Due to hand shake, a pixel and pixels close to it will have the same (or similar) values, which can cause problems (e.g. detect corner as flat region in corner detection).

Q2: In designing our feature point, what characteristics might we wish it to have? Describe the fundamental trade-off between feature point invariance and discriminative power. How should we design for this trade-off?

A2: It is important to find the same interesting point and create the same feature point regardless of location, direction, and angle of taking picture. At this time, various techniques can be used. For example, use differentiation in calculation process for invariance to intensity shift, perform feature vector normalization for invariance to intensity scaling, or use Laplacian response, scale-space blob detector, affine normalization when there is change in image scale and rotation. But the more I put these techniques in, the longer it will take. So, if I have a lot of data, I have to make a compromise.

Q3: In the Harris corner detector, what do the eigenvalues of the 'M' second moment matrix represent? Discuss both how they relate to image intensity and how we can interpret them geometrically.

A3: If eigenvalues are close to 0, it means that intensity change of image is small, so it is most likely flat region. If one of the two eigenvalues is particularly large, it is most likely edge because it means that intensity change in one axis direction is greater than intensity change in the other axis direction. If both eigenvalues have similarly large values, it is highly likely to be corner because intensity change in both axes is large.

The equation for indicating intensity change between a point and its surrounding points is shown in Figure 1.

$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u,y+v) - I(x,y)\right]^2 \qquad E(u,v) \approx [u \;\; v] \; M \begin{bmatrix} u \\ v \end{bmatrix}$$

Figure 1: E(u,v) expression

At this time, if equation is estimated as the shape of quadratic surface, M is obtained. M can be expressed as two eigenvalues and rotation matrix as shown in Figure 2, which represents geometrically estimated shape of quadratic surface. The cross section of quadratic surface is ellipse, where the two eigenvalues determine the axis length and the rotation matrix determines the direction(Figure2).

$$M = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$

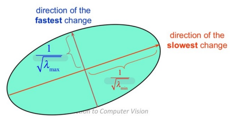

Figure 2: EpiscopalGaudi Corresponding points

Q4: Explain the difference between the Euclidean distance and the cosine similarity metrics between descriptors. What might their geometric interpretations reveal about when each should be used? Given a distance metric, what is a good method for feature descriptor matching and why?

A4: Euclidean distance is correlated with length of vector. This is because it calculates distance between points pointed to by vectors. On the other hand, cosine distance is only related to angle between two vectors, regardless of length of vector.

Therefore, if length of feature vector affects distance, it is better to use Euclidean distance. On the other hand, it is better to use cosine similarity if I want to find distance affected by the angle, regardless of length.