AU7005 Computer Vision (Spring 2024) **Homework 2**

Unit III: Representation

Lecture: Xu Zhao zhaoxu@sjtu.edu.cn

TA: Baoxing Li lbx11sjtu@sjtu.edu.cn

Problem 1. Local feature detection

The **structure tensor** is a matrix derived from the gradient of an image, which is useful in local feature detection. It summarizes the predominant directions of the gradient in a specified neighborhood of a pixel, and the degree to which those directions are coherent. For simplification, we define the image gradient of each pixel as follows:

$$\frac{\partial}{\partial x}\boldsymbol{I}(x,y)=\boldsymbol{I}(x+1,y)-\boldsymbol{I}(x-1,y),\ \frac{\partial}{\partial y}\boldsymbol{I}(x,y)=\boldsymbol{I}(x,y+1)-\boldsymbol{I}(x,y-1),$$

where $\boldsymbol{I}(x,y)$ is the intensity value of image $\boldsymbol{I}$ in (x,y). Then the structure tensor M is defined as:

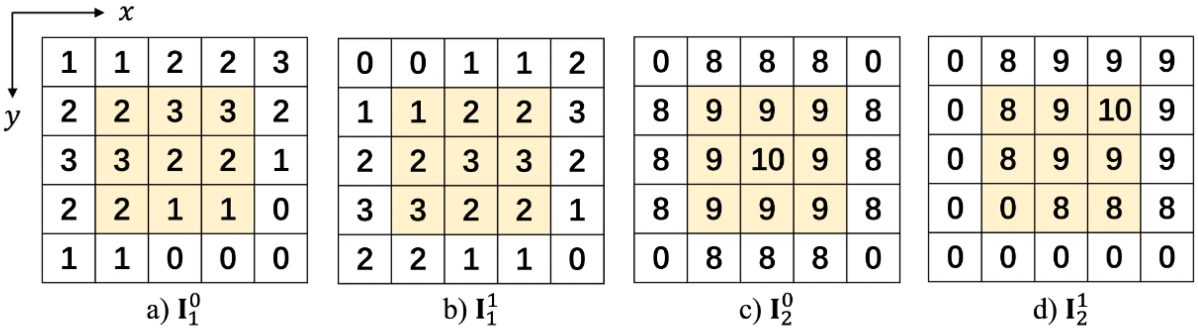
$$M=\sum_{x,y} w(x,y)\begin{bmatrix}\boldsymbol{I}_x^2 & \boldsymbol{I}_x\boldsymbol{I}_y\\ \boldsymbol{I}_x\boldsymbol{I}_y & \boldsymbol{I}_y^2\end{bmatrix},$$

where $\boldsymbol{I}_x=\partial\boldsymbol{I}(x,y)/\partial x$, $\boldsymbol{I}_y=\partial\boldsymbol{I}(x,y)/\partial y$, $w(x,y)$ is a window function, and $w(x,y)=1$ only if (x,y) is in the area of interest. The eigenvalues of the structure tensor are often used to describe the local image properties.

Given two images, $\boldsymbol{I}_1^0$ and $\boldsymbol{I}_2^0$ shown in Figure 1 (a, c), can you analysis the local image properties of the highlighted regions by using the structure tensor M? Then if we want to use the Harris corner detector to extract local image features from $\boldsymbol{I}_1^0$ and $\boldsymbol{I}_2^0$, what is an appropriate threshold range on the *cornerness score* (refer to slides-10), where $k=0.05$, to distinguish the two images? (20 points)

Grading standards. Totally 20 points, 10 for the calculation of two structure tensors, 5 for the analysis of local image properties, 5 for the threshold range of the Harris corner detector.

Answer here.



1	1	2	2	3
2	2	3	3	2
3	3	2	2	1
2	2	1	1	0
1	1	0	0	0

a) $\mathbf{I}_1^0$

0	0	1	1	2
1	1	2	2	3
2	2	3	3	2
3	3	2	2	1
2	2	1	1	0

b) $\mathbf{I}_1^1$

0	8	8	8	0
8	9	9	9	8
8	9	10	9	8
8	9	9	9	8
0	8	8	8	0

c) $\mathbf{I}_2^0$

0	8	9	9	9
0	8	9	10	9
0	8	9	9	9
0	0	8	8	8
0	0	0	0	0

d) $\mathbf{I}_2^1$

Figure 1: Here are four 5×5 images with different intensity values, where the highlighted regions within the 3×3 windows are the areas of interest. You will use these images in problem 1 and problem 2.

Problem 2. Optical flow estimation

Optical flow is the apparent motion of brightness patterns in the image. Given two subsequent frames, the apparent motion field between them can be estimated if three assumptions are satisfied: brightness constancy, small motion, and spatial coherence. Figure 1 (a-b) and (c-d) are two pairs of images satisfying the above assumptions. For these images, the derivative of time is defined as:

$$\frac{\partial}{\partial t}\boldsymbol{I}(x,y) = \boldsymbol{I}^1(x,y) - \boldsymbol{I}^0(x,y)$$

We assume that the pixels of each image within the highlighted 3×3 window have the same motion, denoted as $(u,v)^T$, where u and v are the velocity of pixels along x-axis and y-axis correspondingly. Can you estimate the motions $(u_1,v_1)^T$ and $(u_2,v_2)^T$ for $\boldsymbol{I}_1$ and $\boldsymbol{I}_2$ using the Lucas-Kanade algorithm? Then discuss your results: Are they consistent with the actual motions of the objects? And why? (15 points)

Hint. Substitute the image gradient of $\boldsymbol{I}_1^0$ and $\boldsymbol{I}_2^0$ calculated in problem 1 into the brightness constancy equations.

Grading standards. Totally 15 points, 12 for the calculation of motion, 3 for the discussion of results.

Answer here.

Problem 3. Convolutional neural networks

Convolutional neural network (CNN) is a popular deep-learning-based representation in computer vision. A CNN model is typically composed by convolutional layer, pooling layer, and fully-connected (FC) layer. Table 1 shows the architecture of a CNN model, which is used for image classification. The input size is $C \times H \times W$, where $C = 3$ is the channel size, $H = W = 256$ are the height and width of the input image. Please fill in the blanks in the table. Then choose appropriate activation functions for the output layer and the hidden layers. (15 points)

Layer name	Operations	Output size
Conv1	convolution 7×7, channel 32, stride 2, padding 3	?
Pool1	max pooling 3×3, stride 2, padding 3	?
Conv2	convolution 5×5, channel 64, stride 5, padding 3	?
Conv3	?	$128 \times 6 \times 6$
Pool2	?	$128 \times 1 \times 1$
FC1	?	10×1

Table 1: The architecture of a CNN model.

Grading standards. Totally 15 points, 2 for each blank, 3 for the activation functions.

Answer in Table 1.

Problem 4. Image retrieval (Code & Report)

As shown in Figure 2, we have collected 25 photos of different buildings in SJTU as a database. Given a query image, please design an image retrieval algorithm to find the same buildings as the query image from the database. Your algorithm should use local image features. (50 points)

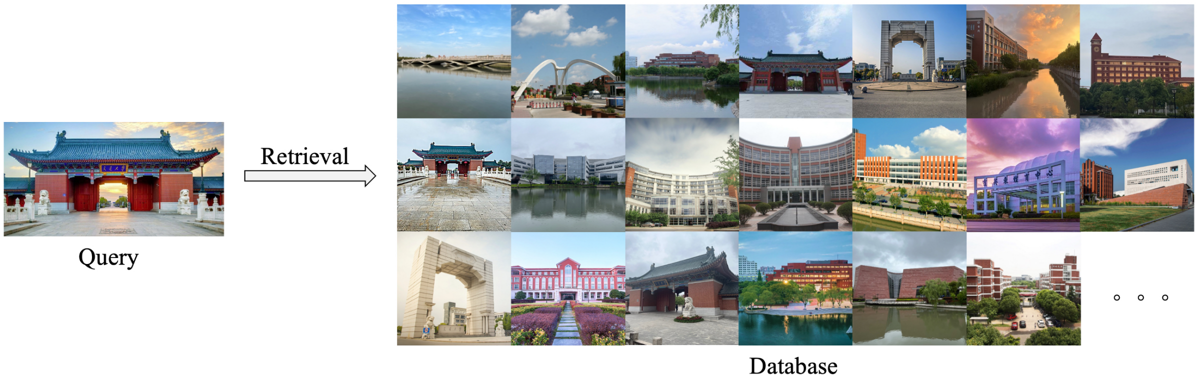


Figure 2: Image retrieval.

Hint.

- You can use any OpenCV function as long as you clearly explain how it works in your algorithm.
- Possibly useful tools: SIFT feature descriptor, bag-of-words model, k-nearest neighbors algorithm, ...

Grading standards.

- Local image feature detection, description and visualization. (20 points)
- Feature matching and visualization. (10 points)
- Scoring the matching results and retrieving the similar photos. (10 points)
- Evaluation and discussion of the results. (10 points)

Code in Jupyter Notebook, then report the algorithm and results here.