Exercise 5 – Image saliency

Overview

In this exercise, we are going to work on different methods for image saliency, which refers to identifying important features in an image to compress its information. We will start by looking into different image filtering techniques. Subsequently, we will look at methods for edge and corner extraction before we will finally compare the properties of different point features.

Q1 Image filtering

During the lecture, you have been introduced to correlation and convolution for image filtering. We will have a look at their properties in this exercise.

1. **1D image filter**

   (a) Consider the 1D image given below (Table 1). Correlate the image with the listed filters and state for each the effect it has on the resulting image. Use the first and last image values to fill the filter at the left and right image borders.

...	2	3	5	8	4	5	4	9	1	3	...

Table 1: 1D image.

   i. $F_1 = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \end{bmatrix}$

   ii. $F_2 = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}$

   iii. $F_3 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$

   (b) What will be the effect on the image for filter F_2 given above if you use the convolution operation instead of the correlation operation?

2. **Smoothing filters**

   (a) A common choice for creating a smoothing image filter is to discretize a Gaussian filter. A (still continuous) filter for the 1D case is given below. Explain the influence of the parameter σ on the resulting image.

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) \tag{1}$$

   (b) Consider now the 2D Gaussian filter below. Something is off here! Transform the filter into a proper image filter.

$$F = \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

(c) (MATLAB) We will now apply a 2D smoothing filter to an image. Please refer to `ex_5_smoothing.m` for further instructions.

3. **Separable filters**

(a) Consider the separable filter F given below. Which are the corresponding row and column factor that this filter can be separated into?

$$F = \frac{1}{256}\begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

(b) Separable filters allow for computational speed-up through their decomposition. How many multiplications and additions need to be computed for a separable filter and non-separable filter of size $N \times N$. How could an additional speed-up be achieved for the separable filter?

4. **True or False?**

(a) Let F, G and I be signals. Denoting the convolution operator as "$*$", the expression "(F$*$G)$*$I=F$*$(G$*$I)" holds.

Answer:

1. **1D image filter**

(a) i. F_1 is a mean or averaging filter that smooths the image.

...	$\frac{7}{3}$	$\frac{10}{3}$	$\frac{16}{3}$	$\frac{17}{3}$	$\frac{17}{3}$	$\frac{13}{3}$	6	$\frac{14}{3}$	$\frac{13}{3}$	$\frac{7}{3}$	...

ii. F_2 shifts the image 1 pixel to the left.

...	3	5	8	4	5	4	9	1	3	3	...

iii. F_3 has no effect on the image such that the filtered image will be equal to the input image.

...	2	3	5	8	4	5	4	9	1	3	...

(b) Using F_3 for a convolution operation shifts the image 1 pixel to the right. The convolution operation is equivalent to the correlation operation with a flipped filter before correlating.

2. **Smoothing filters**

(a) The parameter σ controls the amount of smoothing in the resulting image. If increased, the resulting image gets smoothed more.

(b) The filter is not yet normalized. A proper smoothing filter (where all values add up to 1), is given below:

$$F = \frac{1}{273}\begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

(c) Please refer to `ex_5_smoothing_solutions.m` for further instructions.

3. **Separable filters**

(a) Row and column vector:

$$v_r = \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix},\ v_c = \begin{bmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{bmatrix}$$

(b) i. *Non-separable filter*:
Number of multiplications: N^2
Number of additions: $N^2 - 1$

ii. *Separable filter*:
Number of multiplications: $2N$
Number of additions: $2(N-1)$

iii. We could achieve additional speed-up for separable filters in case the filter has constant value. In this case the operations reduce to:
Number of multiplications: 1
Number of additions: $2(N-1)$

4. **True or False?**

(a) True, convolution is an associative operation. Correlation is not an associative operation.

Q2 Edge detection

After an introduction into basic image filtering in the first exercise, we will now have a look into how we can extract salient image regions from the image via edge detection.

1. **Edge properties in images**

(a) How can you identify edges in images? Please describe two different options.

(b) How can you achieve image sharpening? Please describe two different options.

2. **2D Edge detection**

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	10	10	10	10	1	1
1	1	10	10	10	10	1	1
1	1	10	10	10	10	1	1
1	1	10	10	10	10	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Table 2: 2D input image

(a) Consider the 2D image given above (Table 2). Given the following filters for edge detection

$$F_x = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}, \qquad F_y = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

filter the image to calculate the edge strength $|\nabla F| = \sqrt{F_x^2 + F_y^2}$. Then mark all pixels with an edge strength $|\nabla F| > 20$. You can use the matrices given below in Figure 1 and 2 to fill in your solution. Use the boundary values to fill the filter at the image borders.

(b) We have used the boundary values to pad the image at the borders in the previous question. What would happen if we instead decided to use a black padding (pixel values of 0) at the image borders?

(c) The filters given above detect edges in horizontal (F_x) and vertical (F_y) direction. How could you construct a filter that detects edges at 45° of rotation?

(a) Filtered image using F_x

(b) Filtered image using F_y

Figure 1: Filtered images

Figure 2: Edge strength $|\nabla F|$

3. **True or False?**
   (a) Smoothing masks always sum to 0.
   (b) Smoothing masks may have negative values.
   (c) Low pass filters can be used for image sharpening.
   (d) Edges occur at the zero-crossings of the 1st order derivatives of an image.
   (e) Let the x-axis be aligned with the horizontal axis. Assuming that image I is noise free, the partial derivative $\partial \mathrm{I}/\partial x$ will highlight the horizontal edges in image I.

Answer:

1. **Edge properties in images**
   (a) Edges are characterized through intensity changes in the image. They can therefore be detected by
      i. a minimum/maximum of a first order derivative of the image.
      ii. the zero-crossing of a second order derivative of the image.
   (b) Image sharpening can be achieved using
      i. a low-pass filter by first smoothing an image and then subtracting the smoothed image from the original image.

ii. a high-pass filter by first differentiating an image to enhance regions with a change in intensity and then adding the original image to the differentiated image.

2. **2D Edge detection**

(a) see below

1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	10	10	10	10	1	1
1	1	10	10	10	10	1	1
1	1	10	10	10	10	1	1
1	1	10	10	10	10	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1

Table 3: 2D input image

0	0	0	0	0	0	0	0
0	9	9	0	0	-9	-9	0
0	18	18	0	0	-18	-18	0
0	27	27	0	0	-27	-27	0
0	27	27	0	0	-27	-27	0
0	18	18	0	0	-18	-18	0
0	9	9	0	0	-9	-9	0
0	0	0	0	0	0	0	0

Table 4: Filtered image using F_x

0	0	0	0	0	0	0	0
0	9	18	27	27	18	9	0
0	9	18	27	27	18	9	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	-9	-18	-27	-27	-18	-9	0
0	-9	-18	-27	-27	-18	-9	0
0	0	0	0	0	0	0	0

Table 5: Filtered image using F_y

0	0	0	0	0	0	0	0
0	12.73	20.12	27	27	20.12	12.73	0
0	20.12	25.5	27	27	25.5	20.12	0
0	27	27	0	0	27	27	0
0	27	27	0	0	27	27	0
0	20.12	25.5	27	27	25.5	20.12	0
0	12.73	20.12	27	27	20.12	12.73	0
0	0	0	0	0	0	0	0

Table 6: Edge strength $|\nabla S|$

(b) If we used a black padding, which corresponds to adding pixels with a value of 0, we would create artifact edges at the image boundaries.

(c) The filters given below will detect edges at $-45°$ and $45°$. The two filters are just one example. There exist multiple solutions.

$$F_{-45} = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}, \qquad F_{45} = \begin{bmatrix} 0 & -1 & -1 \\ 1 & 0 & -1 \\ 1 & 1 & 0 \end{bmatrix}$$

3. **True or False?**

(a) False: Smoothing masks always sum to 1. Derivative masks always sum to 0.

(b) False: Smoothing masks have no negative values. Derivative masks have negative values.

(c) True: Low pass filters can be used for image sharpening by subtracting a smoothed image from the original image.

(d) False: Edges occur at the maxima/minima of the *1st* order derivative of an image and the zero-crossings of the *2nd* order derivative of an image.

(e) False: The partial derivative $\partial I/\partial x$ will highlight the vertical edges in the image I.

Q3 Point features

Robots typically perceive their environment through sensors with a limited field of view and from different viewpoints. For the robot to build a consistent map of the environment, these different views have to be correlated. Point features allow us to relate different views (represented through images in our case) relative to each other by extracting distinctive structures from the image and matching these across the images.

1. **Keypoint properties**
   (a) Define what is understood under a feature detector and a feature descriptor and their relation to each other.
   (b) Based on what you learned in the lecture: What are important requirements for feature descriptors? Shortly describe for each requirement what happens if it isn't fulfilled.
2. **Harris corner detector**
   (a) To detect corners, the Harris corner detector looks at the eigenvalues of the moment matrix M.
   $$M = \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} = R^{-1} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} R$$
   Please mark the regions of eigenvalues for edges, corners and flat regions in Figure 3 and state which properties hold in these regions.

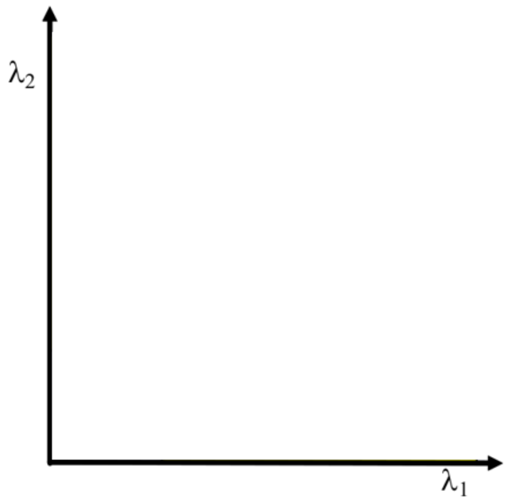

Figure 3: Eigenvalue properties for Harris corner detection

3. **SIFT features**
   (a) Which are the main processing stages when using a SIFT feature detector and descriptor?
4. **Feature matching for panorama creation**
   (MATLAB) After learning about feature detectors and descriptors in the previous questions, we are now finally able to apply the concepts to stitch together images to create a panorama. Please refer to `ex_5_pointFeatures.m` for further instructions.
5. **True or False?**
   (a) The SIFT detection and description method is one of the most computationally efficient approaches to extracting image features.
   (b) When analyzing the 2x2 Second Moment Matrix M of an image area during Harris Corner detection, the criterion for this area to be named a Harris Corner is that both eigenvalues of M must be small.
   (c) Harris corners are invariant to scale.

(d) SIFT feature detector uses LoG kernel to detect keypoints because of its efficiency.

(e) SIFT feature descriptors can not be combined with the Harris corner detector.

Answer:

1. **Keypoint properties**

(a) A feature detector is used to find salient image points, so-called keypoints. Keypoints are typically characterized through image intensity changes, such as for example corners. An example for a feature detector is the Harris corner detector. Feature descriptors are then used to describe the keypoints to allow for matching between consecutive images. The descriptor typically consists of characteristics of an image patch around the keypoint. The SIFT feature descriptor is for example based on gradient orientations in the image patch surrounding a keypoint.

(b) It is important that feature descriptors

i. are distinctive,

ii. do not vary much in appearance across images (ideally invariant to changes in rotation, scale and illumination) and

iii. can be detected and matched very fast.

If these requirements are not fulfilled

i. there might me multiple candidates detected with similar properties such that no unique feature matching can be derived,

ii. it might not be possible to match correct feature pairs across images as the feature descriptor are different for each image (resulting in wrong image alignment) and

iii. high run time might limit applicability of the features descriptors.

2. **Harris corner detector**

(a) The corresponding eigenvalue properties for flat, edge and corner regions in an image can be seen below.

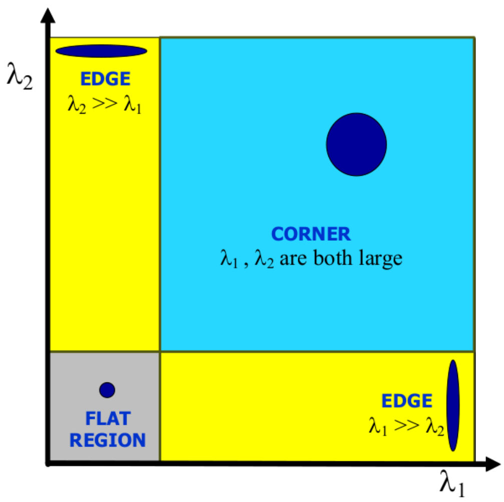

Figure 4: Eigenvalue properties for Harris corner detection

3. **SIFT features**

(a) The main processing stages when using a SIFT feature detector and descriptor are:

i. Extract keypoints + scale

ii. Assign keypoint orientation

iii. Generate keypoint descriptor

4. **Feature matching for panorama creation**

   Please refer to `ex_5_pointFeatures_solutions.m` for further instructions.

5. **True or False?**

   (a) False: While SIFT features show very good performance, they come at the drawback of high computational cost.

   (b) False: See overview above. The criterion is for both eigenvalues to be large.

   (c) False: Harris corners are not invariant to scale.

   (d) False: The SIFT feature detector uses the DoG kernel to detect keypoints which is an approximation of the LoG kernel for more efficiency.

   (e) False: SIFT feature descriptors can be combined with the Harris corner detector, which is also what was done in the Matlab exercise. You can mix detectors and descriptors based on the requirements of your application.