CS-E4850 Computer Vision

Exam 11th of December 2020, Lecturer: Juho Kannala

There are plenty of questions. Possibly many more than can be solved in the given time but answer as many as you can in the available time. The number of points awarded from different parts is shown in parenthesis at the end of each question. The maximum score from the whole exam is 42 points.

The exam must be taken completely alone. Showing or discussing it with anyone is forbidden!

1. Image filtering

   (a) Filter image J with the gaussian filter G using zero padding. (1 p)

$$J = \begin{array}{|c|c|c|}\hline 0 & 1 & 0 \\ \hline 0 & 2 & 0 \\ \hline 0 & 3 & 0 \\ \hline \end{array} \qquad G = \frac{1}{16} \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

   (b) Is it more efficient to filter an image with two 1D filters as opposed to a 2D filter? Why? How does the computational complexity relate to the size of the filter kernel (with K×K pixels) in both cases? (1 p)

   (c) Is the following convolution kernel separable? If so, separate it. (1 p)

$$H = \begin{array}{|c|c|c|c|}\hline 1 & 2 & 3 & 4 \\ \hline 2 & 4 & 6 & 8 \\ \hline 3 & 6 & 9 & 12 \\ \hline 4 & 8 & 12 & 16 \\ \hline \end{array}$$

   For the image I below apply the following filters to the pixel at the center (marked with a box). Round the results to the nearest integer value.

$$I = \begin{array}{|c|c|c|c|c|}\hline 2 & 3 & 4 & 5 & 6 \\ \hline 3 & 4 & 5 & 6 & 8 \\ \hline 4 & 5 & \boxed{6} & 8 & 5 \\ \hline 5 & 7 & 8 & 9 & 3 \\ \hline 9 & 10 & 9 & 4 & 3 \\ \hline \end{array}$$

   (e) 3×3 box filter (i.e. averaging in a 3×3 neighborhood). (0.5 p)

   (f) 3×3 median filter. (0.5 p)

   (g) Why is the Gaussian filter a better smoothing filter than a box filter? How can it be implemented fast? (1 p)

   (h) Compute the edge direction and magnitude (that is, the direction and magnitude of image gradient) at the center pixel using the masks of the Sobel edge detector (S_1 and S_2 below). (1 p)

$$S_1 = \frac{1}{8} \begin{array}{|c|c|c|}\hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} \qquad S_2 = \frac{1}{8} \begin{array}{|c|c|c|}\hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

(i) The binary pixel array on the left below was convolved with an unknown kernel $\boxed{?}$ to produce the result on the right. The output is limited to the same size as input and zero padding was used at the boundaries. Specify the kernel as an array. What task does it accomplish in computer vision. (1 p)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

$* \boxed{?} \Rightarrow$

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	-1	1	0	0	1	-1	0	0
0	0	-1	1	0	0	1	-1	0	0
0	0	-1	1	0	0	1	-1	0	0
0	0	-1	1	0	0	1	-1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

2. Image formation

Assume that all coordinate frames are right handed.

Consider a camera with the following camera projection matrix:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -10 \\ 0 & 1 & 0 & 10 \end{bmatrix}$$

(a) There is a triangle located in the world coordinate system with vertices: $(10, 0, 0)$, $(0, 10, 0)$, and $(0, 0, 10)$. Calculate the coordinates of the projection and draw the image of this triangle as seen by our camera. (1 p)

(b) What happens if you try to compute the projection of $(0, -10, -10)$, and why? . (1 p)

Now we switch to a camera with the following intrinsic parameters:

- A 10 mm focal distance
- Rectangular pixels, 2.5 micron wide and 2 micron tall (one mm is 1000 micron)
- A camera sensor with 4000 pixels horizontally and 3000 vertically
- The principal point is at image point π =(2000,1500) pixels
- No distortion

The world reference system is the same as the camera's canonical reference system (camera is at the world origin and pointed towards the positive z-axis), except that the units for the axes are in centimeters. The cameras pixel image reference system measures image coordinates in pixels, and its origin is in the upper left corner of the image.

(c) Calculate the intrinsic, extrinsic and projection matrices. (1 p)

(d) A point $\mathbf{X}$ has coordinates (0, -10, 40) centimeters in the world reference system. Calculate the coordinates of the projection of X in the pixel image reference system. (1 p)

Now we move the above camera from the origin of the world coordinate system 20 cm in the direction of the positive y-axis and rotate the camera 45° counterclockwise around the x-axis. Note: The rotation is counterclockwise when viewed by an observer looking along the rotation axis towards the origin from the positive side of the x-axis.

(f) Calculate the extrinsic camera matrix and give the coordinates of the projection of $\mathbf{X}$ in the pixel image reference system. (1 p)

3. Triangulation

Two cameras are looking at the same scene. The projection matrices of the two cameras are $\mathbf{P}_1$ and $\mathbf{P}_2$. They see the same 3D point $\mathbf{X} = (X, Y, Z)^\top$. The observed coordinates for the projections of point $\mathbf{X}$ are $\mathbf{x}_1$ and $\mathbf{x}_2$ in the two images, respectively. The numerical values are as follows:

$$\mathbf{P}_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad \mathbf{P}_2 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 1 \end{bmatrix}, \quad \mathbf{x}_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}, \quad \mathbf{x}_2 = \begin{bmatrix} \frac{3}{4} \\ 0 \end{bmatrix}.$$

(a) Present a derivation for the linear triangulation method and explain how $\mathbf{X}$ can be solved using that approach in the general case (i.e. no need to compute with numbers in this subtask). (1 p)

(b) Compute the 3D coordinates of the point $\mathbf{X}$ using the given numeric values for the camera projection matrices and image points. It is sufficient to just give the result. (Hint: You can calculate this with a computer or using pen and paper. In the latter case it may be easiest to write the projection equations in homogeneous coordinates by explicitly writing out the unknown scale factors, and to solve X, Y, Z and the scale factors directly from those equations.) (1 p)

(c) A third camera $\mathbf{P}_3$ is added to the scene. Describe how the linear triangulation method above can be extended to use the information from all the three cameras. (1 p)

(d) If there is noise (i.e. measurement errors) in the observed image coordinates of point $\mathbf{X}$, the linear triangulation method above is not the optimal choice but a nonlinear approach can be used instead. What error function is typically minimized in the nonlinear approach? (1 p)

(e) How does the nonlinear triangulation approach differ from the bundle adjustment procedure which is commonly used in structure-from-motion problems (i.e. how is the bundle adjustment problem different)? (1 p)

4. Local feature detection and description using SIFT

(a) Explain the difference between a feature detector and descriptor. (1 p)

(b) Is Harris corner detector rotation invariant? Could Harris corner detector and normalized cross-correlation based matching be used to match corner features in images related by a rotation? (1 p)

(c) How do we compute a histogram of gradient orientations when generating a SIFT descriptor? (1 p)

Let's assume that we detected SIFT regions from two images (i.e. circular regions with assigned orientations) of the same textured plane.

(f) What is the minimum number of SIFT region correspondence pairs needed for computing a similarity transformation between the pair of images? (1 p)

(g) Describe RANSAC-based procedure for estimating the similarity transformation in a real world use case where there are both correct and incorrect correspondences among the SIFT region correspondences. (1 p)

5. Epipolar geometry

We have a camera pair with projection matrices $P = \begin{bmatrix} I & 0 \end{bmatrix}$ and $P' = \begin{bmatrix} R & \mathbf{t} \end{bmatrix}$ where R and $\mathbf{t}$ are such that the essential matrix E for the camera pair is the following:

$$E = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix}$$

.

(a) Give one possible value for the unit-norm vector $\mathbf{t}$ that points from the second camera center to the first camera center and for the rotation matrix R between the two cameras. Justify your answer briefly. (1 p)

(b) What is the angle between the optical axes of the two cameras? No need to justify your answer, if you are sure about it. (1 p)

(c) A point in the second image has coordinates $\mathbf{b} = \begin{bmatrix} 0.5 & 1 & 1 \end{bmatrix}^T$ in the canonical camera reference system (units are focal distances). Write the equation of the epipolar line of $\mathbf{b}$ in the canonical image reference system of the first camera. Show your derivation. Remember that the canonical (i.e. normalized) image coordinates of a point are the same as its first two canonical camera coordinates. (1 p)

6. Image retrieval

(a) When matching features across two images, why does it make sense to use the ratio: (distance to best match) / (distance to second best match), as a way to judge if we have found a good match? (1 p)

(b) How do we use clustering to compute a bag-of-words image representation? Describe the process. (1 p)

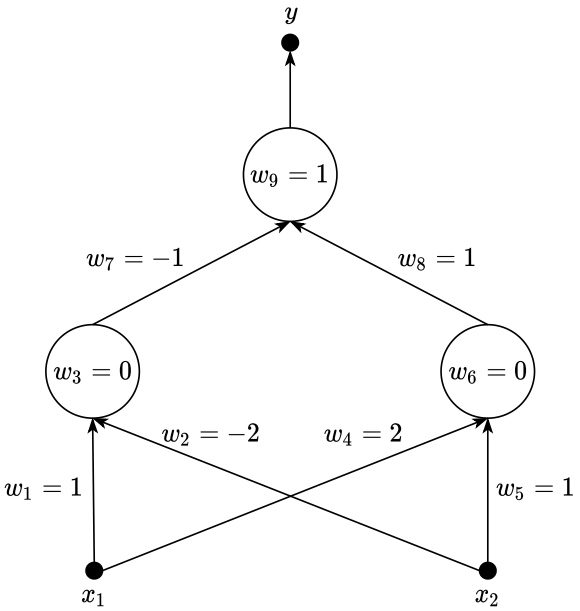
(c) How can we find to which cluster we should assign a new feature, which was not part of the set of features used to compute the clustering? (1 p)

(d) When is it more efficient to create an inverted file index to match a query image to other images in the database, rather than comparing the query to all database images without an index? (1 p)

(e) Why do we need to measure both precision and recall in order to score the quality of retrieved results? (1 p)

7. Neural networks and object detection

The small neural net in the figure below uses ReLU as the nonlinearity at the output of each neuron. The values specified in the hollow circles are biases, and the values along the edges are gains.



(a) Are all the layers in the network above fully connected? (1 p)

(b) What is the output y from the net above when the input is as follows? (1 p)

$$x_1 = 0 \quad \text{and} \quad x_2 = 3$$

(c) What is the gradient $\mathbf{g}$ of the output y of the network above with respect to the weight vector

$$\mathbf{w} = [w_1,\ w_2,\ w_3,\ w_4,\ w_5,\ w_6,\ w_7,\ w_8,\ w_9]^T$$

when the input has the values given in the previous problem? Just give the result if you are confident of your answer. (2 p)

(d) With image data convolutional neural networks are much more popular than fully connected neural networks. Why is this? (1 p)

(e) Especially deep convolutional neural networks have proven to be effective. What function do the earlier layers (a.k.a. the base network) of a deep convolutional neural network serve and why are they often re-used from pre-existing networks such as VGG16. (1 p)

(f) SSD object detector evaluates only a small set (e.g. 4) of default boxes of different aspect ratios at each location. How can it detect large and small objects if the boxes are of fixed size? (1 p)

8. Feature tracking

Let $I(\mathbf{x})$ and $J(\mathbf{x})$ be two grayscale images of the same scene taken from slightly different viewpoints and possibly slightly different orientations. We'd like to track

a point $\mathbf{x}_I$ in image I to it's coordinate $\mathbf{x}_J$ in image J. That is we'd like to know the two dimensional displacement $\mathbf{d}^*$ of point $\mathbf{x}_I$ such that:

$$\mathbf{x}_J = \mathbf{x}_I + \mathbf{d}^*$$

To approximate $\mathbf{d}^*$ we look at a window (small square) $W(\mathbf{x}_I)$ of odd side-length 2h+1 pixels centered around the point $\mathbf{x}_I$ in image I and search for $\mathbf{d}$ that minimizes the dissimilarity between the windows in both images:

$$\mathbf{d}^* = \arg\min_{\mathbf{d}} \epsilon(\mathbf{d})$$

where the dissimilarity $\epsilon(\mathbf{d})$ is defined as a sum over the whole image $\mathbf{x} = (x_1, x_2)$:

$$\epsilon(\mathbf{d}) = \sum_{\mathbf{x}} [J(\mathbf{x} + \mathbf{d}) - I(\mathbf{x})]^2 w(\mathbf{x} - \mathbf{x}_I)$$

$w(\mathbf{x})$ is the indicator function of a $W(\mathbf{x})$:

$$w(\mathbf{x}) = \begin{cases} 1 & \text{if } |x_1| \leq h \text{ and } |x_2| \leq h \\ 0 & \text{otherwise.} \end{cases}$$

We assume that the motion of the camera between the two images is so small that the magnitude of $\mathbf{d}^*$ is much smaller than the diameter of $W(\mathbf{x}_I)$ and use an iterative approach where the problem is now to find a step displacement $\mathbf{s}_t$ that, when added to $\mathbf{d}_t$, yields the new displacement $\mathbf{d}_{t+1}$ at each iteration t such that $\epsilon(\mathbf{d}_t + \mathbf{s}_t)$ is minimized. We add $\mathbf{d}_t$ into $\mathbf{x}$ as follows $J_t(\mathbf{x}) = J(\mathbf{x} + \mathbf{d}_t)$ and approximate the image function $J_t(\mathbf{x} + \mathbf{s}_t)(= J(\mathbf{x} + \mathbf{d}_t + \mathbf{s}_t))$ with its first-order Taylor expansion:

$$J_t(\mathbf{x} + \mathbf{s}_t) \approx J_t(\mathbf{x}) + [\nabla J_t(\mathbf{x})]^T \mathbf{s}_t$$

Minimizing $\epsilon(\mathbf{d}_t + \mathbf{s}_t)$ leads to a linear system of equations $A\mathbf{s}_t = \mathbf{b}$ where

$$A = \sum_{\mathbf{x}} \nabla J_t(\mathbf{x})[\nabla J_t(\mathbf{x})]^T w(\mathbf{x} - \mathbf{x}_I) \quad \text{and} \quad \mathbf{b} = \sum_{\mathbf{x}} \nabla J_t(\mathbf{x})[I(\mathbf{x})] - J_t(\mathbf{x})] w(\mathbf{x} - \mathbf{x}_I)$$

The overall displacement is then the sum of all the steps:

$$\mathbf{d}^* = \sum_t \mathbf{s}_t$$

(a) Show that minimizing $\epsilon(\mathbf{d_t} + \mathbf{s}_t)$ leads to a linear system of equations $A\mathbf{s}_t = \mathbf{b}$. (1 p)

NOTE: the problems (b)-(e) below don't require that you have solved problem (a).

Assuming a window size of 3×3 ($h = 1$) and an initial guess of displacement $\mathbf{d}_0 = [0, 0]^T$. For a particular value of $\mathbf{x}_I$, the two components of $\nabla J_0(\mathbf{x})$ inside the window $W(\mathbf{x}_I)$ are:

$$\frac{\partial J_0}{\partial x_1} = \begin{bmatrix} 10 & 10 & 10 \\ 10 & 10 & 10 \\ 10 & 10 & 10 \end{bmatrix} \quad \text{and} \quad \frac{\partial J_0}{\partial x_2} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

and the difference between the two images is:

$$I(\mathbf{x}) - J_0(\mathbf{x}) = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

(b) Compute A and $\mathbf{b}$. (1 p)

(c) Does the feature at $\mathbf{x}_I$ suffer from the aperture problem? Briefly justify your answer. (1 p)

(d) Give the minimum-norm solution $\mathbf{s}_0$ to the linear system $A\mathbf{s}_0 = \mathbf{b}$ (1 p)

(e) Assume that further iterations of the Lucas-Kanade algorithm do not change the solution $\mathbf{s}_0$ much. Does your answer to the previous question imply that the image motion between I and J at $\mathbf{x}_I$ is approximately horizontal? Briefly justify your answer. (1 p)