Exam in SSY097

March 23rd, 2019

Allowed materials : Pen/pencil, eraser.

The exam consists of six problems. Make sure that you have them all.

- Motivate all answers carefully.
- Use a new paper for each new numbered problem.
- Write on one side of the papers only.
- Write your anonymous number on each new page.
- Avoid using a red pen.
- If you want the result registered as SSY096, write this on the cover page.

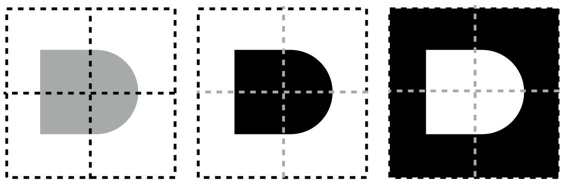
Grades

$\geq$ **8 points** Grade: 3

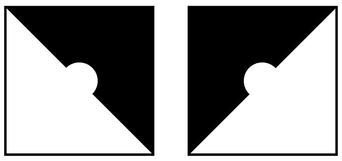
$\geq$ **11 points** Grade: 4

$\geq$ **14 points** Grade: 5

1 SIFT, 3 points



(a) Consider the three patches P, Q, and R shown above. For each patch, sketch a SIFT-like descriptor by drawing its vector bouquet representation. The dashed lines indicate four regions per patch that are used to compute individual bouquets.
How many elements does each descriptor have?
Compare the descriptors for P and Q after L2-normalization, *i.e.*, after dividing each descriptor element by the L2 norm of the descriptor.



(b) Patch T shown above is a rotated version of patch S. Describe how SIFT achieves rotation invariance, *i.e.*, how SIFT is able to extract the same descriptor for both S and T.

(c) Consider the task of descriptor matching between two images $\mathcal{P}$ and $\mathcal{Q}$. For simplicity, we use 1D descriptors in this task. Thus, the descriptor distance between two descriptors p, q is given by their absolute distance $|p - q|$.
The descriptors extracted from image $\mathcal{P}$ are

$$p_1 = (10.0), \qquad p_2 = (3.0), \qquad p_3 = (0.5), \qquad p_4 = (5.0) \ .$$

The descriptors extracted from image $\mathcal{Q}$ are

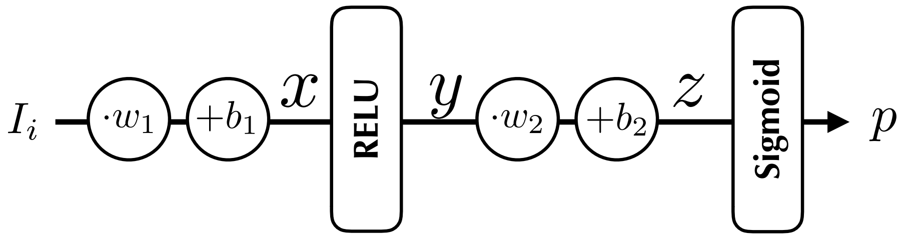
$$q_1 = (4.0), \qquad q_2 = (8.0), \qquad q_3 = (1.0), \qquad q_4 = (14.0) \ .$$

Compute the matches that pass Lowe's ratio test[1] with a threshold of 0.4 when matching features from $\mathcal{P}$ against features from $\mathcal{Q}$.

[1]Also known as the SIFT ratio test.

2 Statistical Learning, 3 points

Consider the simple neural network shown below that performs binary classification on scalar input values.



(a) Compute the loss L_i for a negative example I_i. As in the lecture and the lab, use the negative log-likelihood loss.

(b) Given a negative example $I_i = 1$, compute the derivatives

$$\frac{\partial L_i}{\partial w_1}, \qquad \frac{\partial L_i}{\partial b_1}, \qquad \frac{\partial L_i}{\partial w_2}, \qquad \frac{\partial L_i}{\partial b_2}$$

through the backpropagation algorithm. To this end, first perform a forward pass to compute values for x, y, z, and p. Next, use the chain rule to derive formulas for the derivatives in the backward pass. Compute the actual values for the derivatives using your equations and the values for x, y, z, and p computed during the forward pass.
The current values for w_1, b_1, w_2, and b_2 are

$$w_1 = 10, \qquad b_1 = 5, \qquad w_2 = 1, \qquad b_2 = -15 \ .$$

(c) Using a learning rate of 0.1, what is the next value for b_2?

3 Point Set Registration, 3 points

(a) Consider a transformation of the form

$$\begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \begin{pmatrix} a & 0 & -b \\ 0 & 1 & 0 \\ b & 0 & a \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \\ t_z \end{pmatrix} \quad . \tag{1}$$

Explain how to construct a solver that can be used inside RANSAC to estimate a transformation between two sets of 3D points. To this end, derive a linear system $\mathtt{M}\theta = \mathbf{v}$, where the vector θ contains the parameters of the transformation.
What is the minimal number of correspondences that are needed by the solver to compute a transformation?

(b) Assuming that $a^2 + b^2 = 1$, what type of transformations can be expressed by Eq. 1? Justify your answer.

(c) Let t_{solve} be the time required by the solver from **(a)** to compute a transformation from a minimal number of matches. Let t_{verify} be the time required to determine whether a correspondence is an inlier to a transformation. Given n matches, the run-time of RANSAC for T iterations is thus given by

$$t_{\text{RANSAC}} = \sum_{i=1}^{T} \left(t_{\text{solve}} + n \cdot t_{\text{verify}}\right) \quad . \tag{2}$$

For efficient solvers such as the one from **(a)** and n large enough, we typically have $t_{\text{solve}} << n \cdot t_{\text{verify}}$, *i.e.*, inlier counting is significantly more expensive than computing a model.
The solver from **(a)** is non-minimal in the sense that the minimal number of correspondences needed to compute a transformation provides one more equation than necessary. Describe how the additional equation can be used to accelerate RANSAC. Justify your answer.

4 RANSAC and Camera Pose Estimation, 3 points

Given n 2D-3D matches $\mathcal{M} = \{(\mathbf{x}_i, \mathbf{X}_i)\}$ between 2D image coordinates $\mathbf{x}_i$ and 3D point positions $\mathbf{X}_i$, camera pose estimation is the task of estimating the pose `R`, $\mathbf{t}$ of the camera, *i.e.*, of computing the rotation `R` and translation $\mathbf{t}$ that best explains the matches.

(a) Write a function `ransac_pose_estimation` that computes a camera pose inside RANSAC. The function takes as input the intrinsic camera calibration matrix `K`, a set of n 2D-3D matches $\mathcal{M} = \{(\mathbf{x}_i, \mathbf{X}_i)\}$, and a threshold τ on the reprojection error that is used to distinguish between inliers and outliers. You can assume that the following functions are available:

`R, t = minimal_solver(M')` computes a camera pose from three 2D-3D matches.

`error = reprojection_error(x_i, X_i, K, R, t)` computes the reprojection error of a 2D-3D match $(\mathbf{x}_i, \mathbf{X}_i)$ for given `K`, `R`, and $\mathbf{t}$.

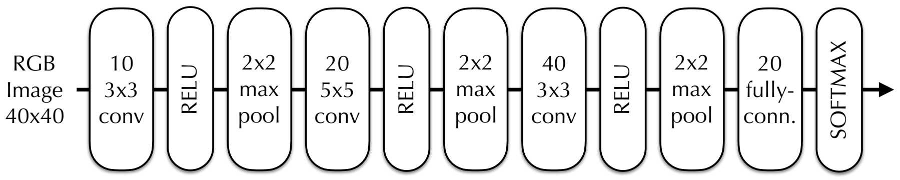
`ransac_pose_estimation` should return the pose with the largest number of inliers. You can write Matlab code, pseudocode, or a mixture of both.

(b) Given a 2D-3D match $(\mathbf{x}_i, \mathbf{X}_i)$, a pose `R`, $\mathbf{t}$, and the intrinsic camera calibration matrix `K`, derive a formula for computing the reprojection error of the match. Introduce all intermediate variables that you use. Explain your formula.

(c) Assuming that you are given an estimate ε for the inlier ratio, derive a formula for the maximum number of iterations $k_{\max}$ RANSAC needs to take, such that the probability of not generating a single all-inlier sample[2] in all $k_{\max}$ iterations is equal to or below a threshold τ. Assume that in each iteration, RANSAC needs to draw a random sample of size m. Explain your formula.

[2]An all-inlier sample is a random sample that only contains inliers and no outliers.

5 Deep Learning, 3 points



(a) Consider the convolutional neural network shown above. For each layer of the network with trainable parameters, write down the number of trainable parameters. Assume that the convolutional layers use padding and explain your answers.
You can choose to ignore bias terms when computing the number of parameters. Indicate whether or not you are using bias terms in your computations.

(b) In the context of overfitting, explain what "transfer learning" is and how it can be used to reduce overfitting.

(c) Name another technique that can be used to reduce overfitting. Explain how the technique works and how it helps to reduce overfitting.

6 Generative Adversarial Networks, 3 points

(a) Generative Adverserial Networks (GANs) are one approach to generative modeling discussed in the lecture. GANs can be used to generate novel images from random input vectors. Draw the general architecture of a GAN and describe each of its parts.

(b) The training objective of a GAN is given by

$$\min_G \max_D E_{\mathbf{x}\sim p_{\text{data}}}[\log(D(\mathbf{x})] + E_{\mathbf{z}\sim p_{\text{model}}}[\log(1 - D(G(\mathbf{z})))] \ . \tag{3}$$

Explain the individual terms in Eq. 3.

(c) Training GANs is notoriously hard. One challenge is that the discriminator and the generator need to be sufficiently balanced to enable a GAN to learn to generate realistic images. Explain what happens if the discriminator becomes too powerful too soon during training. Explain what happens if the discriminator is not powerful enough.