Exam in SSY097

March 17, 2018

Allowed materials: 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.
- Only 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

≥ 8 **points** Grade: 3

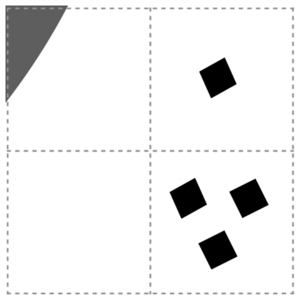
≥ 11 **points** Grade: 4

≥ 14 **points** Grade: 5

1 SIFT, 3 points

A SIFT-like descriptor (as the one in Lab 1) was computed for the image patch, Q, below. The regions used are indicated with grey dashed lines so these lines are not part of the actual image. Note that unlike for original SIFT only four regions are used. Illustrate (for example using vector bouquets) what the SIFT-like descriptor for Q will look like. It should be clear from your description how many elements there are in the descriptor vector.

(a)



Q

(b) How is the descriptor for $2Q$ different from that of Q? Motivate your answer.

(c) How is the descriptor for $Q + 10$ different from that of Q? Motivate your answer.

(d) How is the descriptor for $1 - Q$ different from that of Q? Motivate your answer.[1]

[1] You can assume that the values in Q are between 0 and 1.

2 Statistical learning, 3 points

A simple 3-class classifier works in the following way,

$$y_1 = I \cdot w_1 + c_1, \tag{1}$$
$$y_2 = I \cdot w_2 + c_2, \tag{2}$$
$$y_3 = I \cdot w_3 + c_3, \tag{3}$$

where the dot refers to the dot product and I is an 20×20 input image. The output class probabilities (for the 3 classes) are then given by softmax

$$p_1 = \frac{e^{y_1}}{e^{y_1} + e^{y_2} + e^{y_3}}, \quad p_2 = \frac{e^{y_2}}{e^{y_1} + e^{y_2} + e^{y_3}}, \quad p_3 = \frac{e^{y_3}}{e^{y_1} + e^{y_2} + e^{y_3}} \tag{4}$$

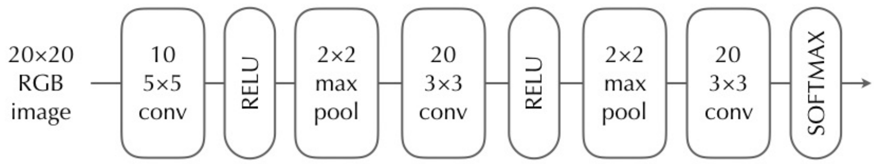
To train the parameters in w_i and c_i, we try to minimize the negative log likelihood over a training set:

$$L(\theta) = -\sum_{i \in S_1} \ln p_{1,i} - \sum_{i \in S_2} \ln p_{2,i} - \sum_{i \in S_3} \ln p_{3,i} \tag{5}$$

where S_k refers to the examples in the training set with label i and $p_{k,i}$ refers to the resulting p_k when example image I_i is used as input.

(a) Derive the update rules for w_1 and c_1 if we use stochastic gradient descent with learning rate μ. Give formulas for the three cases: example from class 1, example from class 2 and example from class 3. Motivate your answer by showing your derivations. (The two last cases will be very similar.)

(b) Consider training the network below with stochastic gradient descent. How many elements does the gradient vector of the partial loss, ∇L_i, have. (The max pooling layers have stride 2 and no padding is used.)



3 Image registration, 3 points

In order to align two images using an affine transformation, we want to write a function `ransac_affine` that takes two images `source_img` and `target_img` and outputs a transformation given by a matrix A and a translation vector t.

(a) Write down the Matlab code required to define this function. You can use the functions

```
[points, descriptors] = extractSIFT(img)

corrs = matchFeatures(desc, desc_tilde)

[Ahyp, thyp] = minimal_solver(pts, pts_tilde)

nbr_outliers = count_outliers(A, t, pts, pts_tilde)
```

without describing their contents. It doesn't matter if you do syntax errors as long as the idea is understandable (and correct).

(b) A student uses five measurements for the minimal solver instead of three, as *more data is always better*. Explain why this is not a good idea.

4 Triangulation, 3 points

Consider two cameras with camera matrices P and $\tilde{P}$. We have matched Sift points with coordinates u and $\tilde{u}$ and want to use triangulation to find the corresponding 3D point U.

(a) Write down the relationship that should hold for these variables (if the Sift points were correctly matched).

(b) Explain how to construct a minimal solver for triangulation. Start from the camera equations in (a) and derive equations on the form $M\theta = v$ that can be easily solved in Matlab.

Assume that we used Ransac to triangulate a 3D point from five views. Comment on the certainty of the estimate if we got

(c) 1 inlier

(d) 2 inliers

(e) 3 inliers

5 Mixed, 1.5 + 1.5 points

(a) We are trying to estimate a transformation using Ransac and we are trying to choose between two different transformation models. Below are some characteristics for the two models.

Model 1:

- Minimal solver requires 3 measurements
- Minimal solver takes 1 time unit to run
- Estimating the loss takes 0.01 time units per measurement

Model 2:

- Minimal solver requires 4 measurements
- Minimal solver takes 0.01 time units to run
- Estimating the loss takes 0.01 time units per measurement

On average how much longer will it take to find a good solution with the slower model if the rate of outliers is 90% and we have 100 measurements in total. (An approximate answer is enough.)

(b) To find edges in an image, I, we have computed gradients at each pixel and the magnitudes (lengths) of these gradients. These form a new image M, that is shown below. High values correspond to edge points.

$$M = \begin{bmatrix} 9 & 10 & 6 & 1 & 2 & 1 \\ 1 & 7 & 12 & 5 & 4 & 1 \\ 1 & 1 & 9 & 12 & 5 & 1 \\ 1 & 1 & 4 & 8 & 11 & 6 \\ 1 & 2 & 2 & 1 & 8 & 10 \end{bmatrix}. \tag{6}$$

For a point to be an edge point, it is required to have a larger response than its neighbours perpendicular to the edge. In this case we have edge points at $(2,1)$, $(3,2)$, $(4,3)$, $(5,4)$ and $(6,5)$. When we refine these positions, we only want to move them perpendicular to the edge (otherwise they just move closer to each other.) Explain how to refine an edge point to find the edge with sub-pixel precision. You don't have to compute any numerical values, just give the general formula. You can assume that we can approximate derivatives such as I_x' or M_{xx}'', without showing how.

6 Registration, 3 points*

Using SIFT we have found 100 matches between two images of sizes 1000×1000. To align the images we use Ransac with an affine transformation model (as in Lab 3). However, in this case all matches are incorrect and thus completely random. (This would, for example, be the case if the two images are showing completely different things.)

Note: To solve the following problems you need to make reasonable assumptions and approximations.

(a) What is the probability that the first Ransac iteration yields 8 inliers using an error threshold of 2 pixels. Doing very crude approximations such as $13 \approx 10$ is fine.[2]

(b) What is *roughly* the probability that after 1000 iterations, the best solution has at least 8 inliers using an error threshold of 2 pixels. Answer with a formula if it is hard to get a numerical approximation.

[2] *Hint: Recall from statistics that if you toss a coin n times and p is the probability of heads, then the number of times you get heads follows a binomial distribution. For the problem above, you can use the Poisson approximation for the binomial function*

$$P(x = k) \approx \frac{\lambda^k e^{-\lambda}}{k!} \tag{7}$$

where $\lambda = np$.