Exam in SSY097

March 18, 2017

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.

Grades

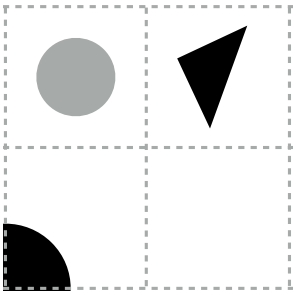
≥ 8 **points** Grade: 3

≥ 11 **points** Grade: 4

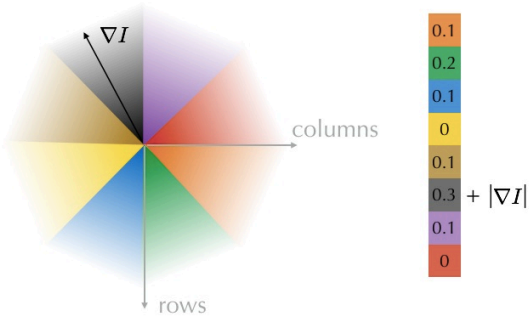
≥ 14 **points** Grade: 5

1 SIFT, 3 points

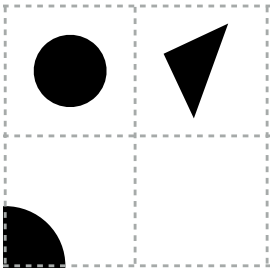
(a) A SIFT-like descriptor (as the one in Lab 1) was computed for the following image Q. The regions used are indicated with grey dashed lines so these lines are not part of the actual image[1]. Note that unlike for original SIFT only four regions are used. Describe (for example, using vector bouquets) how the SIFT descriptor will look for this patch. The structure of the gradient histograms is indicated below for reference.



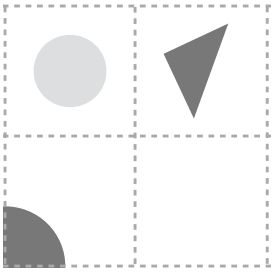
Q



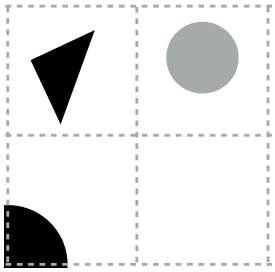
(b) Which of the following three images, A, B and C will produce the most similar SIFT-like descriptor. Motivate your answer carefully.



A



B



C

(c) In this example four regions were used but in original SIFT 16 regions were used. What is the effect of using too few regions? What is the effect of using too many?

[1] You can ignore any strange effects at the image border.

2 Statistical learning, 3 points

(a) Given input values (x, y) the output probability, p, is computed as[2]

$$p = \frac{1}{1 + e^{-s}} \quad \text{with} \quad s = w_1 x + e^{w_2 y}. \tag{1}$$

To find suitable values for w_1 and w_2, we will use statistical learning. More precisely we will try to minimize the negative log likelihood loss over a small training set with positive examples:

$$(x_1, y_1), \quad (x_2, y_2), \quad \ldots, \quad (x_n, y_n) \tag{2}$$

and negative examples:

$$(x_{n+1}, y_{n+1}), \quad (x_{n+2}, y_{n+2}), \quad \ldots, \quad (x_m, y_m). \tag{3}$$

Derive the update rule for stochastic gradient descent in this case. Try to make the formulas simple.

(b) Consider five layers from a convolutional neural network listed below,

1. Convolutional layer with 10 filters.
2. Rectified linear unit.
3. Max pooling with 2×2 regions.
4. Convolutional layer with 20 filters.
5. Rectified linear unit.

Spatially all the filters are 3×3. If you prefer, here are the same layers as defined in `MatConvNet`:

```
net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{f*randn(3,3,1,10, 'single')/sqrt(3*3), zeros(1, 10, 'single')}}, ...
                           'stride', 1, ...
                           'pad', 0) ;

net.layers{end+1} = struct('type', 'relu') ;

net.layers{end+1} = struct('type', 'pool', ...
                           'method', 'max', ...
                           'pool', [2 2], ...
                           'stride', 2, ...
                           'pad', 0) ;

net.layers{end+1} = struct('type', 'conv', ...
                           'weights', {{f*randn(3,3,10,20, 'single')/sqrt(3*3*10), zeros(1,20,'single')}}, ...
                           'stride', 1, ...
                           'pad', 0) ;

net.layers{end+1} = struct('type', 'relu') ;
```

How many weights are there in all these layers in total?

(c) If we input a 100×100 image to the network in b, the output from the last layer is roughly $50 \times 50 \times 20$. Roughly, how many weights would a fully-connected layer with this many inputs and outputs have?

[2] Normally, the input is an image or and image patch, but here it is just two numbers (x, y).

3 Image registration, 3 points

Let's say we want to warp a source image to a target image. Let $\tilde{u}_1, \ldots, \tilde{u}_n$ be points in the source image and $u_1, \ldots, u_n$ points in the target image, where

$$\tilde{u}_k = \begin{pmatrix} \tilde{x}_k \\ \tilde{y}_k \end{pmatrix} \quad \text{and} \quad u_k = \begin{pmatrix} x_k \\ y_k \end{pmatrix} \tag{4}$$

In order to align the images we use Ransac to estimate an *affine* transformation.

(a) Explain how to construct minimal solver for this problem. Show how to use the u_j's and $\tilde{u}_j$'s to get a system on $M\theta = b$ form, (where θ are all the unknowns) and explain how it can be solved in Matlab. Be sure to define all variables that you use in your explanation.

(b) A similarity transformation consists of rotation, scaling and translation. Consider a case with 100 correspondences/measurements and 50% outliers. On average, how much longer would it take to find a good affine transformation with RANSAC than it would with a similarity transformation? Assume that both transformations are appropriate for the data. Motivate your answer.

4 Triangulation, 3 points

Given a set of Sift points with pixel coordinates

$$u_i = \begin{pmatrix} x_i \\ y_i \end{pmatrix} \tag{5}$$

and corresponding camera matrices P_i, we want to triangulate a 3D point U using Ransac.

(a) Explain how to construct a *minimal* solver for this problem. Show how to get a system $M\theta = b$ form, (where θ are all the unknowns) and explain how it can be solved in Matlab. Be sure to define all variables that you use in your explanation. (There are multiple correct solvers. Choose one.)

(b) Given the camera matrix

$$P = \begin{pmatrix} 100 & -100 & 0 & 20 \\ 100 & 100 & 0 & -20 \\ 0 & 0 & 2 & 2 \end{pmatrix}, \tag{6}$$

check if the 3D point

$$U = \begin{pmatrix} -2 \\ -1 \\ -2 \end{pmatrix}, \tag{7}$$

is consistent with the image point

$$u = \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} 42 \\ 156 \end{pmatrix} \tag{8}$$

at an outlier threshold of 5 pixels. (The same way that you did in the lab.)

5 Object detection, 3 points

(a) To detect objects, we can use a CNN or a linear classifier in a sliding window. But sometimes you also want to detect the size of the object. Suggest a good way of doing this.

(b) Below is part of the output from a sliding window detector. There is one strict local maximum in this output. Explain how we can find the *true* maximum with sub-pixel accuracy. Setup the equation that will solve the problem. The equation should be on the form $M\theta = b$ where M and b are numerical.

$$\begin{bmatrix} 1 & 1 & 2 & 1 & 2 & 1 \\ 1 & 1 & 6 & 6 & 4 & 1 \\ 1 & 1 & 6 & 12 & 8 & 1 \\ 1 & 1 & 4 & 10 & 8 & 1 \\ 1 & 2 & 2 & 1 & 1 & 1 \end{bmatrix}. \tag{9}$$

6 Camera geometry, 3 points*

Consider the triangulation problem as in Lab 4. We have used Ransac + Gauss-Newton (on the inliers) to estimate the best 3D point, U. You can assume that Gauss-Newton has converged.

Let $\bar{r}$ be a vector with the reprojection residuals (of the inliers) and J be the Jacobian matrix of $\bar{r}$. Explain carefully why and how the smallest eigenvalue of $Q = J^T J$ is related to the uncertainty in our estimated U.