Computer Vision II - Homework Assignment 1

Stefan Roth, Xiang Chen, Jochen Gast
Visual Inference Lab, TU Darmstadt

May 1, 2019

This homework is due on May 15th, 2019 at 11:00.
Please read the instructions carefully!

General remarks

Your grade does not only depend on the correctness of your answer but also on clear presentation of your results and good writing style. It is your responsibility to find a way to *explain clearly how* you solve the problems. Note that we will assess your complete solution and not exclusively the results you present to us. If you get stuck, try to explain why and describe the problems you encounter – you can get partial credit even if you have not completed the task. Hence, please hand in enough information so that we can understand what you have done, what you have tried, and how your final solution works.

Every group has to submit its own original solution. We encourage interaction about class-related topics both within and outside of class. However, you are not allowed to share solutions with your classmates, and *everything you hand in must be your own work*. Also, you are not allowed to just copy material from the web. You are required to *acknowledge any source of information you use to solve the homework* (i. e. books other than the course books, papers, websites, etc). Acknowledgments will *not* affect your grade. Not acknowledging a source you rely on is a clear violation of academic ethics. Note that both the university and the department are very serious about plagiarism. For more details, see the department guidelines about plagiarism at `https://www.informatik.tu-darmstadt.de/studium_fb20/im_studium/studienbuero/plagiarismus/index.de.jsp` and `http://plagiarism.org`.

Programming exercises

For the programming exercises you will be asked to hand in Julia code. Please make sure that your code complies with **Julia v1.1.0**. In order for us to be able to grade the programming assignments properly, stick to the function names and type annotations that we provide in the assignments. Additionally, comment your code in sufficient detail so that it will be easy to understand for us what each part of your code does. Sufficient detail does not mean that you should comment every line of code (that defeats the purpose), nor does it mean that you should comment 20 lines of code using only a single sentence. Your Julia code should display your results so that we can judge if your code works from the results alone. Of course, we will still look at the code. If your code displays results in multiple stages, please insert appropriate `sleep` commands between the stages so that we can step through the code. Group plots that semantically belong together in a single figure using subplots and don't forget to put proper titles and other annotations on the plots. Please be sure to name each file according to the naming scheme included with each problem. This also makes it easier for us to grade your submission. And finally, please make sure that you included your name and email in the code.

Files you need

All the data you will need for the problems will be made available in Moodle.

What to hand in

Your hand-in should contain a PDF file (a plain text file is ok, too) with any textual answers that may be required. You must not include images of your results; your code should display these instead. For the programming parts, please hand in all documented `.jl` scripts and functions that your solution requires. Make sure your code actually works and that all your results are displayed properly!

Handing in

Please upload your solution files as a single `.zip` or `.tar.gz` file to the corresponding Moodle area at `https://moodle.tu-darmstadt.de/course/view.php?id=15293`. **Please note that we will not accept file formats other than the ones specified!** Your archive should include your write-up (`.pdf` or `.txt`) as well as your code (`.jl` scripts). If *and only if* you have problems with your upload, you may send it to `cv2staff@visinf.tu-darmstadt.de`

Late Handins

We will accept late hand-ins, but we will deduct 20% of the total reachable points for every day that you are late. Note that even 15 minutes late will be counted as being one day late! After the exercise has been discussed in class, you can no longer hand in. If you are not able to make the deadline, e. g. due to medical reasons, you need to contact us *before* the deadline. We might waive the late penalty in such a case.

Code Interviews

After your submission, we may invite you to give a code interview. In the interview you need to be able to explain your written solution as well as your submitted code to us.

Problem 1 - Probabilities and Statistics - 13 points

1. Give three real-world examples of a joint distribution $p(x, y)$ where x and y are

x	y
discrete	discrete
continuous	discrete
continuous	continuous

1 points

2. Show that the following relation is true:

$$p(v, w, x, y, z) = p(x, y, z)\, p(v \mid w, x, y, z)\, p(w \mid x, y, z)$$

2 points

3. The joint probability $p(w, x, y, z)$ over four variables factorizes as

$$p(w, x, y, z) = p(w)\, p(z \mid x, w)\, p(y \mid z)\, p(x).$$

Show that x is independent of w by showing that $p(x, w) = p(x)p(w)$.

3 points

4. For manipulating expectations, we have – among others – the following four relations

$$\begin{aligned} \mathrm{E}[c] &= c, \\ \mathrm{E}[cf(x)] &= c\,\mathrm{E}[f(x)], \\ \mathrm{E}[f(x) + g(x)] &= \mathrm{E}[f(x)] + \mathrm{E}[g(x)], \\ \mathrm{E}[f(x)g(y)] &= \mathrm{E}[f(x)]\,\mathrm{E}[g(y)], \quad \text{if } x, y \text{ independent}, \end{aligned}$$

where c is a constant, and $\mathrm{E}[\cdot]$ denotes the expectation operator.

Use the relations from above to prove the following relationship between the second moment around zero and the second moment about the mean (variance):

$$\mathrm{E}[(x - \mu)^2] = \mathrm{E}[x^2] - \mathrm{E}[x]\mathrm{E}[x].$$

3 points

5. On a game show, a contestant is told the following rules:

There are three doors, labeled 1, 2, and 3. A single prize has been hidden behind one of them. You get to select one door. Initially your chosen door will *not* be opened. Instead, the game show host will open one of the other two doors, and *he will do so in such a way as not to reveal the prize.* For example, if you first choose door 1, he will then open one of doors 2 and 3, and it is guaranteed that he will choose the one which does not reveal the prize.

At this point, you will be given a fresh choice of door: you can either stick with your first choice, or you can switch to the other closed door. All the doors will then be opened and you will receive whatever is behind your final choice of door.

Imagine that the contestant chooses door 1 first; then the game show host opens door 3, revealing nothing behind the door, as promised. Should the contestant (a) stick with door 1, or (b) switch to door 2, or (c) does it make no difference? Base your choice on the probabilities of the outcomes. You may assume that initially, the prize is equally likely to be behind any of the three doors.

Hint: Use Bayes rule.

4 points

Problem 2 - Modeling - 6 points

When we create models in computer vision, we are encoding prior assumptions about the real world in a mathematical framework. Here, we will have a look at the simple Markov Random Field (MRF) model for binocular stereo that we encountered in the lecture:

$$\begin{aligned} p(\mathbf{d} \,|\, \mathbf{I}^0, \mathbf{I}^1) &\propto p(\mathbf{I}^1|\mathbf{I}^0, \mathbf{d}) p(\mathbf{d}|\mathbf{I}^0) \\ p(\mathbf{I}^1|\mathbf{I}^0, \mathbf{d}) &= \prod_{i,j} f(I^0_{i,j} - I^1_{i,j-d_{i,j}}) \\ p(\mathbf{d}|\mathbf{I}^0) &= \prod_{i,j} f_H(d_{i,j}, d_{(i+1),j}) f_V(d_{i,j}, d_{i,(j+1)}). \end{aligned}$$

For the sake of simplicity, we assume for the moment that the disparity $\mathbf{d}$ is independent of $\mathbf{I}^0$. To recap the assumptions that we put in the above model, please briefly answer the following questions **in your own words**:

1. How can we justify to factor the likelihood $p(\mathbf{I}^1|\mathbf{I}^0, \mathbf{d})$ into individual terms for each pixel?

   1 points

2. What may be the reason to choose a robust likelihood function (e. g. Laplacian, Student-t) over the Gaussian likelihood? What distinguishes a Gaussian from a robust likelihood such as a Laplacian?

   1 points

3. What assumptions does the pairwise MRF prior over the disparity values encode? Are the disparity values d_1 and d_3 in Figure 1 independent?

   2 points

4. Can you think of different compatibility functions (or "potential functions") other than the delta function used in the Pott's model?

   1 points

5. In window-based stereo we always compare a whole region around a single pixel with the corresponding region in the second image in order to measure, how good a certain disparity value for that pixel fits to the images. Why can we get away in our model with comparing only single pixel values in the factors of the likelihood function?

   1 points

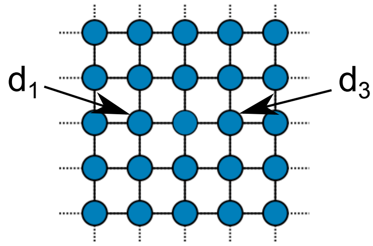

Figure 1: Markov Random Field prior over disparity values.

Problem 3 - Getting to know Julia - 1 point

In this first programming problem you will get to know Julia and the PyPlot package.

Tasks:

- Use the image `a1p3.png` from the ZIP file.
- In `problem3.jl`, write a Julia script that does the following:
  1. Read the image using the `PyPlot` package and convert it to grayscale. Make sure that you obtain an image in 64-bit floating point precision which has values within the valid color range $[0, 1]$.
  2. Display the image using the `PyPlot` package with an appropriate color map, i. e. so that we see a graylevel image on screen. Also add a title and remove the axes.
  3. Compute and display (either at the Julia prompt or as annotation of the image display) the minimum, maximum, and mean pixel value.

Problem 4 - Stereo Likelihood - 10 Points

In this problem you are going to implement a simple likelihood model for stereo. The `Distributions` package should be useful.

Tasks:

- Write a function `shift_disparity(I, d)` that takes an image $\mathbf{I}$ and a disparity map $\mathbf{d}$ and shifts each pixel $I_{i,j}$ by the disparity $d_{i,j}$ to get the displaced image $\mathbf{I}^d$.

  1 points

- Write a function `gaussian_lh(I0, I1d, mu, sigma)` that computes the Gaussian likelihood
$$p(\mathbf{I}^1|\mathbf{I}^0, \mathbf{d}) = \prod_{i,j} \mathcal{N}(I^0_{i,j} - I^1_{i,j-d_{i,j}} \mid \mu, \sigma). \tag{1}$$

  1 points

- Write a function `gaussian_nllh(I0, I1d, mu, sigma)` that computes the negative log of the Gaussian likelihood (Eq. 1). Do not just take the negative log of the previous result but rather implement everything in the log domain.

1 points

- Write a function `laplacian_nllh(I0, I1d, mu, s)` that computes the negative log of the Laplacian likelihood

$$\hat{p}(\mathbf{I}^1|\mathbf{I}^0,\mathbf{d}) = \prod_{i,j} \frac{1}{2s} \exp\left\{-\frac{|I^0_{i,j} - I^1_{i,j-d_{i,j}} - \mu|}{s}\right\}. \tag{2}$$

1 points

- Write a Julia script `problem4.jl` that does the following:

1. Using the function `load_data()`, load the images `i0.png` and `i1.png` from the Tsukuba dataset (make sure that the values are 64-bit floating points in the range $[0,1]$) and convert them to grayscale; additionally load the ground truth disparity map $\mathbf{d}_{\mathrm{gt}}$ contained in the file `gt.png`, convert the values to 64 bit floating point and multiply them by a factor 255. The resulting disparities should be (floating point) integer values in the range $[0,16]$. If you get an error of the form "InexactError: Int64" the scaling of your disparities is off; these should be 64 bit floating points corresponding to integer values, e. g. 1.0, 2.0, 3.0, ...

1 points

2. Compute and display the likelihood under the Gaussian likelihood model with $\mu = 0$ and $\sigma = 1.2$ for the ground truth disparity $\mathbf{d}_{\mathrm{gt}}$. Note that the ground truth is not defined along the image borders. Therefore, write and use the function `crop(I0,I1d,gt)` to crop $\mathbf{I}^0$, $\mathbf{I}^d$, and $\mathbf{d}_{\mathrm{gt}}$ to the (central) region where ground truth disparities are greater than 0.

1 points

3. Now compute and display the value of the negative log-likelihood. What is the reason for computing the log instead of working with the actual probability densities?

1 points

4. Take the second input image (corresponding to `i1.png`) and artificially generate pixels for which the brightness constancy is violated at 12% (and also 25%) of all pixels. To do so, write a function `make_noise(I, p)` that randomly chooses p% of the pixel positions and replaces the corresponding brightness with values chosen uniformly from the range $[0.1, 0.9]$. We will call the modified image $\mathbf{I}^1_{\mathrm{noise}}$. Compute and display the likelihood $p(\mathbf{I}^1_{\mathrm{noise}}|\mathbf{I}^0,\mathbf{d})$ and the negative log-likelihood $-\log p(\mathbf{I}^1_{\mathrm{noise}}|\mathbf{I}^0,\mathbf{d})$.

1 points

5. Compute, display and compare the negative log of the Laplacian likelihood $\hat{p}(\mathbf{I}^1|\mathbf{I}^0,\mathbf{d})$ and $\hat{p}(\mathbf{I}^1_{\mathrm{noise}}|\mathbf{I}^0,\mathbf{d})$ for the noisy and original images. Use $\mu = 0$ and $s = 1.2$.

1 points

6. Discuss your findings regarding outlier robustness based on the Gaussian and the Laplacian likelihood models.

1 points