Computer Vision II - Homework Assignment 1

Stefan Roth, Stephan Richter, Anne Wannenwetsch
Visual Inference Lab, TU Darmstadt

May 10, 2017

This homework is due on May 25th, 2017 at 9:00.
Please read the instructions carefully!

General remarks

Your grade will depend on the correctness of your answer, but also on a clear presentation of your results and good writing style. It is your responsibility to find a way to *explain clearly how* you solved the problems. Note, that you will get grades for the solution, not for the result. If you get stuck, try to explain why and describe the problems you encountered – you can get partial credit even if you did not complete the task. So please hand in enough information for us to understand what you did, what you tried, and how it worked!

Every student has to submit his / her own original solution. We encourage interaction about class-related topics both within and outside of class. However, you should not share solutions with your classmates, and *everything you hand in must be your own work*. You are also not allowed to just copy material from the web. You are required to *acknowledge any source of information that you used to solve the homework* (i.e. books other than the course books, papers, web sites, etc.). Acknowledgments will *not* affect your grade. Not acknowledging a source that you have used, is a clear violation of academic ethics. Note that the university as well as the department is very serious about plagiarism. For more details please see `https://www.informatik.tu-darmstadt.de/de/studierende/studium/plagiarismus/` and `http://plagiarism.org`.

Programming exercises

For the programming exercises you will be asked to hand in Julia code. Please make sure that the code also works on v0.5.1., which is what we will use for grading. We have Julia access available for everyone who needs it. In order for us to be able to grade the programming assignments properly, stick to the function names that we provide in the assignments and comment your code in sufficient detail so that it will be easily clear to 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 be sure to include 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 handin should contain a PDF file (plain text is ok, too) with any textual answers that may be required. You do not have to include images of your results. Your code should show 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 writeup and your code to the corresponding moodle area: `https://moodle.tu-darmstadt.de/course/view.php?id=8744`. If *and only if* you experience problems with uploading your solution, you may also email it to `cv2staff@visinf.tu-darmstadt.de`

You are supposed to send all your solution files as a single `.zip` or `.tar.gz` file. **Please note that we cannot accept file formats other than the ones specified!**

Late Handins

We will accept late handins, 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, for some serious (say medical) reason, cannot make the deadline, you need to contact us *before* the deadline. We might waive the late penalty in such a case.

Code Interviews

After your submission, it may be possible that you are required to give a code interview. In such a case you need to be able to explain your written solution as well as your submitted code to us.

Problem 1 - Probabilities and Statistics - 8 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. The joint distribution of two variables x and y is given by its density function $p(x, y)$. What remains if we marginalize the joint distribution with respect to variable x? How can we compute the density function $f(\cdot)$ of the resulting distribution using $p(x, y)$?

2 points

3. We have three boxes with two drawers each. We randomly pick one of the boxes, randomly open one of the drawers and find a gold coin. It is known that each drawer contains a silver or a gold coin and that the three boxes have the combination gold/gold, silver/gold and silver/silver. Use Bayes rule to compute the posterior probability that the second drawer of the chosen box also contains a gold coin.

2 points

4. We assume that variables x and y are independent given a third variable z, i.e.

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

Show that we can factorize $p(x, y \mid z)$ as $p(x, y \mid z) = p(x \mid z) \cdot p(y \mid z)$.

3 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. We will have a look at the simple Markov Random Field (MRF) model for binocular stereo that we encountered in lecture 2

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

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

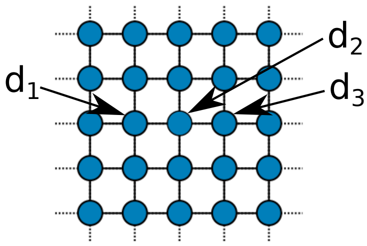

Figure 1: Markov Random Field prior over disparity values

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. Why is it unfavorable to model the likelihood function as a Gaussian?

   1 points

3. What assumptions does the pairwise MRF prior over the disparity values encode? Given the disparity values d_1, d_2 and d_3 in figure 1, are the values d_1 and d_3 independent?

   2 points

4. Can you think of different compatibility function (or "potential function") 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

Problem 3 - Getting to know Julia - 1 point

In this first programming problem you will get to know Julia.

Tasks:

- Use the image `a1p3.png` from the ZIP file.
- In `Problem3.jl`, write a Julia script that does the following:
  1. Load the image using the `Images` package, convert it to grayscale, and then convert it to 64-bit floating point format, which we will use internally for all our problems.
  2. Display the image using the `ImageView` package.
  3. Display the image using the `PyPlot` package with an appropriate color map (i.e. so that we see a graylevel image on screen), add a title and remove the axes.
  4. Compute and display (either at the Julian 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 `apply_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 `compute_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 `compute_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 `compute_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 function `load_images()`, load the images `i0.png` and `i1.png` from the Tsukuba dataset, load the ground truth disparity map $\mathbf{d}_{\mathrm{GT}}$ contained in the file `gt.png` and convert the values to doubles.

     1 points

  2. Compute and display the likelihood under the Gaussian likelihood model with $\mu = 0$ and $\sigma = 1$ 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 $\mathbf{d}_{\mathrm{GT}}$.

     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 10% (and also 30%) of all pixels. To do so, write a function `add_noise(I, p)` that randomly chooses p% of the pixel positions and replaces the corresponding brightness with values chosen uniformly from the range $[0, 1]$. We will call the modified image $\mathbf{I}^1_{\text{Noise}}$. Compute and display the likelihood $p(\mathbf{I}^1_{\text{Noise}}|\mathbf{I}^0, \mathbf{d})$ and the negative log-likelihood $-\log p(\mathbf{I}^1_{\text{Noise}}|\mathbf{I}^0, \mathbf{d})$.

   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 $\hat{p}(\mathbf{I}^1_{\text{Noise}}|\mathbf{I}^0, \mathbf{d})$ for the noisy and original images. Use $\mu = 0$ and $s = 1$.

   1 points

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

   1 points