Processing of Biological Data

Prof. Dr. Volkhard Helms | Saarland University
Winter Semester 2018/19 | Chair for Computational Biology

Exercise Sheet 3

Due: 13.12.2018 23:59

Submission

- You are advised to work in groups of two people. If necessary, we will suggest teammates.
- Submit your solutions on paper in Room 3.01, E2 1 or better send an email with a single PDF attachment to nicolas.kuenzel@bioinformatik.uni-saarland.de. Late submissions will not be considered. **In any case, hand in all source code via mail. Please also include your output. Otherwise you will loose points.**
- Do not forget to mention your names/matriculation numbers.
- You are free to use any programming language to solve the problems. The usage of libraries that allow you to circumvent implementing the algorithms asked for will not grant you points.

Exercise 3.1: Denoising of images using a diffusion filter (60 points)

In the first part of the assignment you will derive and implement a diffusion filter which you then apply on a given density matrix. This exercise should give you an understanding of the underlying process and mathematics of image denoising. The data is given in **noise.csv** in the supplement.

(a) First, derive the discrete form of the diffusion equation

$$\frac{\partial \rho(x,y,t)}{\partial t} = -\nabla(-D\nabla\rho(x,y,t)) \tag{1}$$

(using finite differences) in its simplest form (homogeneous diffusion, meaning that the diffusion coefficient is independent of position $\rightarrow$ use $D = 1$) in 2D with a maximum error of $\mathcal{O}(\Delta t)$ in time and $\mathcal{O}(\Delta x^2)$ and $\mathcal{O}(\Delta y^2)$ in space (show why the error has this size). You are allowed to apply equal spacing in x and y. (10 points)

(b) Explain why the diffusion equation is suitable to denoise images and why this is needed when performing automated image analysis. Does is matter how long the diffusion is applied and if yes why? What happens when it is applied for a large number of time steps? (10 points)

(c) Implement the discrete homogeneous, two-dimensional diffusion equation

$$\rho(x,y,t+\Delta t) = k\cdot(\rho(x+h,y,t)+\rho(x-h,y,t)+\rho(x,y+h,t)+\rho(x,y-h,t))+(1-4k)\cdot\rho(x,y,t) \tag{2}$$

with $k = \frac{c\Delta t}{h^2}$ and equal spacing in x and y ($\Delta x = \Delta y = h$) and apply it to the data given in **noise.csv**. Set your coordinate spacing to 1 and the time spacing to 0.1. Since the formula can't be applied directly at the edges of the array, you can either use

$$\begin{aligned}\rho(0-h,y,t) &= \rho(0,y,t)\\ \rho(x,0-h,t) &= \rho(x,0,t)\\ \rho(x_{\max}+h,y,t) &= \rho(x_{\max},y,t)\\ \rho(x,y_{\max}+h,t) &= \rho(x,y_{\max},t)\end{aligned} \tag{3}$$

or keep the values at the borders constant.

Play with the number of time steps and find a value for which you obtain a reasonably good result. Plot the data after applying the diffusion with this number of time steps. Also plot the data before applying the diffusion in order to have a good comparison between the two states. You can also plot the result for different numbers of time steps if you like (not more than three though). (30 points)

(d) Set the value of Δt to 0.25 and 0.4 and decrease the number of time steps accordingly. How does the result change and why? (10 points)

Exercise 3.2: Laplace filter (40 points)

In the second part of the assignment you will study another filter. This so-called Laplace filter is based on the Laplacian

$$\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \tag{4}$$

which calculates the second partial derivative of the function with respect to each independent variable. Thus a discrete approximation of the second derivative $\frac{\partial^2 I}{\partial x^2}$ can be obtained by convolving an image given as a matrix $f(x, y)$ with the kernel $\begin{pmatrix} 1 & -2 & 1 \end{pmatrix}$ which is nothing else than the matrix formulation of the following formula:

$$\frac{\partial^2 f(x,y)}{\partial x^2} = \frac{f(x+\Delta x, y) - 2f(x,y) + f(x-\Delta x, y)}{\Delta x^2} \ . \tag{5}$$

In image processing, a kernel (also called mask or convolution matrix) is a small matrix that is used for blurring, sharpening etc. These effects arise when performing a convolution between a specific kernel and an image.

(a) Use the given kernel to derive a 3x3 kernel that can be used to compute a discrete approximation to the 2D Laplacian. For this add the kernels for the second derivative in x and y together. Apply the derived kernel to a center pixel of the following image (via convolution):

$$\begin{pmatrix} 10 & 0 & -2 \\ -1 & -2 & 5 \\ 9 & 4 & 1 \end{pmatrix} \tag{6}$$

(10 points)

(b) Implement the application of the Laplace filter onto a given data matrix (**modified_gaussian.csv**). The filtering process works via convolution of the original data with the filter matrix. Plot the original data and the filtered data using a surface plot (maybe multiply the convolution result by -1 in order to have a better look at the result). What effect does the Laplace filter have? How can this be used in order to improve the understanding of your data? You can also apply the filter onto the image **dna.jpg** (takes quite some time to compute) in order to see its effect. (25 points)

(c) Why is it important in practice to convolve an image with a Gaussian (or diffusion) filter before convolving with a Laplacian filter? You can test this on the data in Exercise 3.1 (use a gray color scheme). If you do this please hand in a plot for this as well, which will make it easier for you to explain. (5 points)