Assignment 1

Convex Optimization
University of Bern

Denoising

An image taken by a camera can contain noise. This means that every pixel in the image can be written as a sum of a noise-free value plus additive noise.

$$g(x, y) = u(x, y) + n(x, y), \tag{1}$$

where g is the noisy image, u is the noise-free image and n is the noise. The pixel coordinates are denoted by x and y. The objective of denoising is to recover the noise-free image.

In order to solve this problem we need to make some assumptions or prior knowledge on how a noise-free image "looks like". A common choice is to assume that the solution must have a low total variation. This is based on the fact that natural images are well-approximated by piece-wise constant functions, which have low total variation.

We obtain the solution by minimizing the following functional

$$\tilde{u} = \arg\min_u \frac{\lambda}{2}\|u - g\|^2 + \|\nabla u\|, \tag{2}$$

where $\|u - g\|^2$ is a least squares penalty on deviating from the input image g, $\|\nabla u\|$ is the total variation, and $\lambda > 0$ is a parameter.

Your task is to implement the gradient descent method that solves the problem in (2) and answer the questions below.

1. Briefly describe the problem.
2. Describe the reasons and motivations behind this problem.
3. Let $E(u) = \frac{\lambda}{2}\|u - g\|^2 + \|\nabla u\|$ be the objective function. Write the finite difference approximation of E, and derive the gradient $\nabla_u E$.
4. Implement a gradient descent method that solves denoising. The function specification is the following:

```
function [ u ] = denoising_YourName( g, lambda )
% input: g: single grayscale image, the noisy image
%        lambda: parameter
% output: u: denoised image
```

   You have to substitute "YourName" with your name and surname in capital letters. Add comments in your code to explain the most important parts of your algorithm.

5. Show images obtained by very high, very low and a reasonable λ (you find a reasonable λ by looking at the solution). Explain what the effect of λ is.
6. Because we obtain the noisy images by adding the noise ourselves, we can compare the solution to the original image. Find the optimal λ that minimizes the sum of squared distances (SSD)

$$SSD(u) = \sum_x \sum_y (\tilde{u}(x,y) - u(x,y))^2,$$

   where u is the ground truth image and $\tilde{u}$ is our solution. Display the SSD vs. λ graph.

What to hand in:

- Your personal written and commented **Matlab code** (zip file). The code should run without errors or warnings and should not crash, otherwise there will be a penalty in the final assignment mark.
- A **PDF report** (see template for details).

Demosaicing

RGB CCD sensors do not record the full color information of the scene, but they capture different color information for each pixel according to a Bayer color filter. This results in an image where in each pixel we know only one of the RGB components (fig.1.). Demosaicing algorithms aims to recover the missing data information.

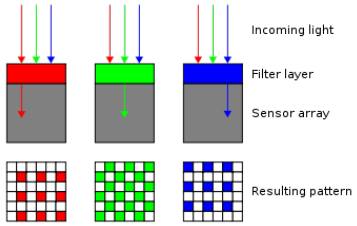

Fig. 1: Bayer filter

We get the solution by minimizing the following three functional

$$\tilde{u}_R = \arg\min_{u_R} \frac{\lambda}{2}\|u_R - g\|^2_{\Omega_R} + \|\nabla u_R\|, \tag{3}$$

$$\tilde{u}_G = \arg\min_{u_G} \frac{\lambda}{2}\|u_G - g\|^2_{\Omega_G} + \|\nabla u_G\|, \tag{4}$$

$$\tilde{u}_B = \arg\min_{u_B} \frac{\lambda}{2}\|u_B - g\|^2_{\Omega_B} + \|\nabla u_B\|, \tag{5}$$

where the first term penalizes the deviation from the input, the second term is a penalty on total variation and λ is a parameter. The exact meaning of the first term is

$$\|u_C - g\|^2_{\Omega_C} = \sum_x \sum_y \Omega_C(x,y)\|u_C(x,y) - g(x,y)\|^2, \tag{6}$$

where C denotes the three different color channels. Ω_C is defined such that $\Omega_C(x,y) = 1$ if the pixel value at (x,y) is valid and $\Omega)_C(x,y) = 0$ when the data is missing.

Your task is to implement the gradient descent method that solves the problems (3), (4) and(5) and answer the questions below.

1. Briefly describe the problem.
2. Describe the reasons and motivations behind this problem.
3. Let $E(u) = \frac{\lambda}{2}\|u - g\|_\Omega^2 + \|\nabla u\|$ be the objective function. Write the finite difference approximation of E, and derive the gradient $\nabla_u E$.
4. Implement a gradient descent method that solves demosaicing. The function specification is the following

```
function [ u ] = demosaicing_YourName( g, lambda )
% input: g: single bayer-filtered image
%        lambda: parameter
% output: u: demosaiced image
```

   You have to substitute "YourName" with your name and surname in capital letters. Add comments in your code to explain the most important parts of your algorithm.

5. Show images obtained by very high, very low and a reasonable λ (you find a reasonable λ by looking at the solution). Explain what the effect of λ is.
6. Because we synthetically generated the Bayer filtered input image, we can compare the solution to the original image. Find the optimal λ that minimizes the sum of squared distances (SSD)

$$SSD(u) = \sum_x \sum_y (\tilde{u}(x,y) - u(x,y))^2,$$

   where u is the ground truth image and $\tilde{u}$ is our solution. Display the SSD vs. λ graph.

What to hand in:

- Your personal written and commented **Matlab code** (zip file). The code should run without errors or warnings and should not crash, otherwise there will be a penalty at the final assignment mark.
- A **PDF report** (see template for details).

Inpainting

Inpainting aims to recover undefined or invalid pieces of an input image g. The mask denoted by Ω specifies the missing data. The pixel value at (x, y) is valid or invalid, when $\Omega(x, y) = 1$ or $\Omega(x, y) = 0$ (see fig.2.). These are the only values (0 or 1) that Ω can have.

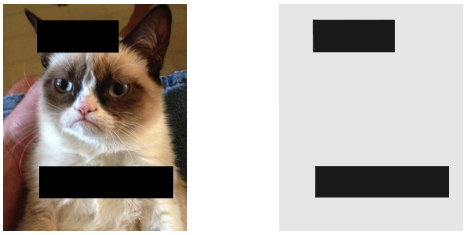

Fig. 2: left: input image, right: mask

In order to recover the missing data, we optimize the function

$$\tilde{u} = \arg\min_u \frac{\lambda}{2}\|u - g\|_\Omega^2 + \|\nabla u\|, \tag{7}$$

where the first term penalizes the deviation from the input, the second term is the total variation penalty and λ is a parameter. The exact meaning of the first term is

$$\|u - g\|_\Omega^2 = \sum_x \sum_y \Omega(x, y)\|u(x, y) - g(x, y)\|^2. \tag{8}$$

Your task is to implement the gradient descent method that solves the problem in (7) and answer the questions below.

1. Briefly describe the problem.

2. Describe the reasons and motivations behind this problem.

3. Let $E(u) = \frac{\lambda}{2}\|u - g\|_\Omega^2 + \|\nabla u\|$ the objective function. Write the finite difference approximation of E, and derive the gradient $\nabla_u E$.

4. Implement a gradient descent method that solves inpainting. The function specification is the following:

```
function [u] = inpainting_YourName(g,omega,lambda)
% input: g: single gray scaled image
%           omega: mask
%           lambda: parameter
% output: u: inpainted image
```

   You have to substitute "YourName" with your name and surname in capital letters. Add comments in your code to explain the most important parts of your algorithm.

5. Show images obtained by very high, very low and a reasonable λ (you find a reasonable λ by looking at the solution). Explain what the effect of λ is.

6. Because we synthetically generate the input image, we can compare the solution to the original image. Find the optimal λ that minimizes the sum of squared distances (SSD)

$$SSD(u) = \sum_x \sum_y (\tilde{u}(x,y) - u(x,y))^2,$$

   where u is the ground truth image and $\tilde{u}$ is our solution. Display the SSD vs. λ graph.

What to hand in:

- Your personal written and commented **Matlab code** (zip file). The code should run without errors or warnings and should not crash, otherwise there will be a penalty at the final assignment mark.
- A **PDF report** (see template for details).

Superresolution

Given an input image, the goal of superresolution is to compute a higher resolution image. This can be modeled by

$$g(x, y) = (u * k)(\alpha x, \alpha y) \tag{9}$$

where g is the low resolution input image, u is the high resolution image we want to recover, k is an averaging filter that models the down-sampling and α is the down-sampling factor. In the discrete this model can be written as

$$\mathbf{g} = D\mathbf{u} \tag{10}$$

where $\mathbf{g}$ and $\mathbf{u}$ are column vectors obtained by the lexicographic juxtaposition of the columns of the two-dimensional matrices g and u, and D is a downsampling operator. If the high-resolution image is a $M \times N$ image, the low resolution image would be a $\frac{M}{\alpha} \times \frac{N}{\alpha}$ image and D a $\frac{MN}{\alpha^2} \times MN$ matrix. The super resolution problem can be solved by minimizing the functional

$$\tilde{\mathbf{u}} = \arg\min_{\mathbf{u}} \frac{\lambda}{2} \|D\mathbf{u} - \mathbf{g}\|^2 + \|\nabla \mathbf{u}\|, \tag{11}$$

The final image $\tilde{u}$ can be easily obtained from the column vector $\tilde{\mathbf{u}}$. Notice that the term $\|\nabla \mathbf{u}\|$ is equivalent to the two dimensional TV regularization $\|\nabla u\|$ applied on the two dimensional matrix u, *i.e.*, if $\mathbf{u}$ is the column vector equivalent to the matrix u, then we have $\|\nabla \mathbf{u}\| = \|\nabla u\|$.

Your task is to implement the gradient descent method that solves the problem in (11) and answer the questions below.

1. Briefly describe the problem.

2. Describe the reasons and motivations behind this problem.

3. Let $E(\mathbf{u}) = \frac{\lambda}{2}\|D\mathbf{u} - \mathbf{g}\|^2 + \|\nabla\mathbf{u}\|$ the objective function. Write the finite difference approximation of E, and derive the gradient $\nabla_u E$.

4. Implement a gradient descent method that solves super resolution. The function specification is the following:

```
function [u] = superresolution_YourName(g,D,lambda)
% input: g: single M X N gray scaled image
%           D: downsampling ooperator
%           lambda: parameter
% output: u: superresolved image
```

   You have to substitute "YourName" with your name and surname in capital letters. Add comments in your code to explain the most important parts of your algorithm.

5. Show images obtained by very high, very low and a reasonable λ (you find a reasonable λ by looking at the solution). Explain what the effect of λ is.

6. Since we synthetically generate the low-resolution image, we can compare the solution to the original image. Find the optimal λ that minimizes the sum of squared distances (SSD)

$$SSD(u) = \sum_x \sum_y (\tilde{u}(x,y) - u(x,y))^2,$$

   where u is the ground truth image and $\tilde{u}$ is our solution. Display the SSD vs. λ graph.

What to hand in:

- Your personal written and commented **Matlab code** (zip file). The code should run without errors or warnings and should not crash, otherwise there will be a penalty at the final assignment mark.
- A **PDF report** (see template for details).