UNIVERSITÄT TÜBINGEN
Lehrstuhl für Computergrafik
Prof. Dr.-Ing. Hendrik P.A. Lensch
Jieen Chen
28.06.2018

Computational Photography
Assignment 8

Submission deadline for the exercises: Thursday, 5. July 2018 before 11:59.

Instructions: Upload the source code to your solution **(no images please, just the plain code)** in the ILIAS system at:
ILIAS Computational Photography SoSe2018

8.1 Demosaicing with Total Variation Regularizer (50 points)

In the previous exercises, we implemented demosaicing by interpolation as well as two data fitting approaches: gradient descent and conjugate gradient. However, the reconstructed color images show strong color checkerboard patterns. From our experience, we know that this is clearly an reconstruction artifact and should not show up in the image. Therefore we would like to bring our knowledge of images somehow into the reconstruction. The images that contain the unwanted checkerboard pattern have strong image gradients [Hint 1], therefore punishing the image gradients during reconstruction should lead to better results. In the lecture, you learned about the Total Varaiation (TV) regularizer, which is the integral of the absolute gradient of the signal. It is formulated as

$$\|\nabla x\|_1 \tag{1}$$

where x is the reconstructed image, ∇ is gradient operator and $\|.\|_1$ is the $L1$ norm. In this exercise, you will use the total variation regularizer in the gradient descent approach to improve upon the demosaicing result from the last exercise.
The objective function with TV is

$$E(x) = \|b - Ax\|^2 + \lambda_{TV}\|\nabla x\|_1 \tag{2}$$

where λ_{TV} is the weight parameter for the TV.

(a) Original Image (b) Result without TV Regularizer (c) Result with TV Regularizer

Figure 1: Demosaicing

a) Implement function `tv_grad.m` that returns the gradient of the TV. Please refer to the lecture slide and use the column vector form of the gradient.

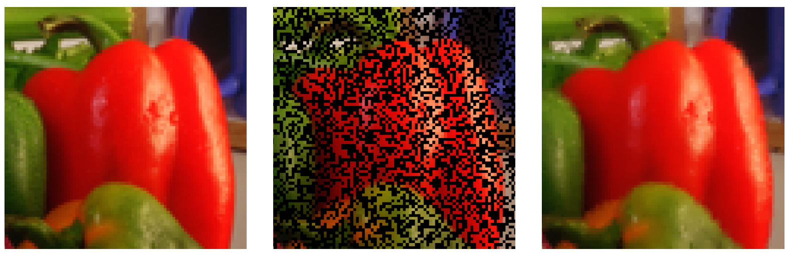
b) Add the gradient of the TV regularizer to the gradient descent step.

8.2 Inpainting with Total Variation and L2 Gradient Regularizer (50 points)

In photography or cinema, deterioration of images occurs due to camera or film artifacts, so that pixels or blocks of pixels can be lost. The reconstruction of the lost part of an image or video is called inpainting. Even if the deterioration operator is known, the solution is not unique, therefore prior knowledge about the image must be added to the reconstruction, such as its smoothness. We can again use regularizers that punish the gradient to achieve that. Besides the TV, a regularizer with the gradient in L2 norm can be formulated as,

$$\|\nabla x\|_2^2 \tag{3}$$

In this exercise, you will use both regularizers to inpaint an image with lost pixels.
We simulate the deterioration by designing the image formation matrix A. It randomly produce dark (dead) pixels, such as shown in Figure 2(b). The matrix A is provided in the code.



(a) Original Image (b) Deteriorated Image (c) Inpainted Result

Figure 2: Inpainting

a) Use the objective function from Equation 2 to solve the inpainting problem. Use the same implementation as in the exercise above.

b) Analytically compute the derivative of the $L2$ regularizer from Equation 3.

c) By using $L2$ regularizer, we change the objective function to

$$E(x) = \|b - Ax\|^2 + \lambda_{L2}\|\nabla x\|_2^2 \tag{4}$$

Implement function `l2_grad.m` to compute the gradient of the $L2$ regularizer.

d) Add the gradient of the $L2$ regularizer to the gradient descent step.

e) From the results, you will observe both regularizers produce different artifacts by their own. To improve the result, it is better to combine both regularizers. This new objective function is then,

$$E(x) = \|b - Ax\|^2 + \lambda((1-\beta)\|\nabla x\|_2^2 + \beta\|\nabla x\|_1) \tag{5}$$

where λ is the regularization weight, β is the blending weight.

f) Add the gradient of both regularizers with their weights to the gradient descent step.

g) Compare and briefly discuss the results of the three approaches. Feel free to experiment with the weights.

Hints:

1) The image gradient ∇x is not the same as the gradient in the gradient decent step. The pixels of the image gradient contain differences of neighboring pixels from the original image. Horizontal and vertical directions are both considered.
Please refer to the link for more details:
https://en.wikipedia.org/wiki/Image_gradient.
The two gradient images can be computed as matrix vector products. The matrix is the gradient operator ∇ and the vector x is the original image reshaped to a column vector. Use the provided function `grad_operator(img_size)` to generate the gradient operator matrices for both directions.

2) Please refer to the slides named `least_squares_gradient.pdf` that present the analytical solution of least squares gradient. (Answer to Exercise 5.1)