Image Processing and Pattern Recognition

Assignment 2 - Non-Local Means

October 31, 2016

Deadline: November 14, 2016 at 23:59h.
Submission: Upload report and implementation zipped (*[surname].zip*) to the TeachCenter using the "My Files" extension.

1 Goal

Image denoising is one of the most fundamental problems in computer vision. The task is to recover the true image U from a noisy observation F. Due to the image formation process and other external effects, the true image U is corrupted by noise, i.e. $F = U + N$, where $N \sim \mathcal{N}(0, \sigma)$ is zero-mean gaussian noise of variance σ. An image denoising method tries to separate the observation F into the image and noise components. It is a classical example of an *inverse problem*, where we assume an underlying (often physical) process and we try to reason about unknown parameters (true image U and noise N) based on the result of the process (noisy observation F).

2 Method

Let U, F be the discrete image functions of the true image and the noisy observation. We denote by U_p the value of the image at pixel $p = (i, j), \quad 1 \leq i \leq M,\ 1 \leq j \leq N$, where N, M are the width and height of the image respectively. Grayscale images are given as intensity values in $\mathbb{R}$, whereas the value of a pixel in a color image is a vector in $\mathbb{R}^3$. In the following, we are describe the Non-Local Means (NLM) method [1] for grayscale images, everything applies *mutatis mutandis* to color images.
The basic idea of the Non-Local Means algorithm is to exploit the *self-similarity* of the image. By recovering the value of a pixel from a weighted average of pixels *that look similar*, we can get a substantial improvement in denoising performance. Due to its simplicity and ease of implementation, NLM is among the most popular denoising approaches. On the downside, the method is very computing intensive.
The value of a pixel p in the NLM method is given by

$$U_p = \frac{\sum_{q \in Q} w_{p,q} F_q}{\sum_{q \in Q} w_{p,q}}, \tag{1}$$

where Q is a neighborhood around p and $w_{p,q}$ is the weight (*i.e.* similarity between pixel p and q).

Connection to Gaussian Filtering Eq. (1) can be seen as a generalization of gaussian filtering. In the gaussian filter, the weights $w_{p,q}$ are given by a gauss function and depend only on the spatial distance between p and q, *i.e.* pixels far away from the center pixel p have a lower weight. The gauss filter does not take into account the image content, therefore it smoothes over image edges.

Connection to Bilateral Filtering In a similar way, eq. (1) is related to bilateral filtering. The bilateral filter improves upon simple gaussian filtering in that it takes into account the image content. The coefficients of the gauss kernel are downweighted if the pixel value is very different from the center pixel. This means that the weight $w_{p,q}$ now depends both on the spatial distance between p and q, and on the similarity of the pixel values F_p and F_q. For this reason, the bilateral filter is capable of preserving edges in the image.

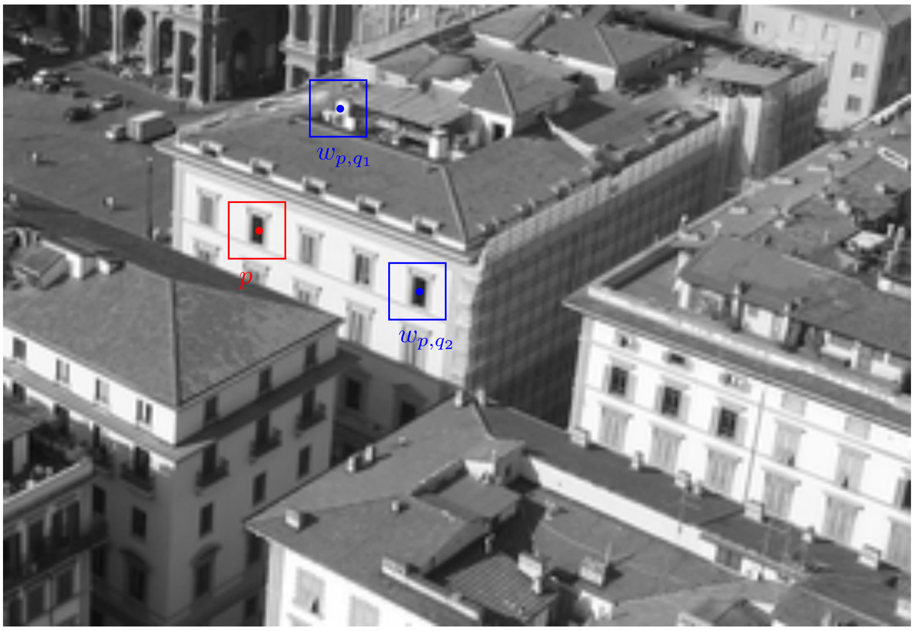

Figure 1: Non-Local Means image denoising. The value of the pixel p (red dot) is computed as a weighted average of pixels in the neighborhood. The weight depends on the similarity of patches, *e.g.* the patch around q_2 has a higher similarity to p than q_1. In the bilateral filter, the patch size is reduced to 1, *i.e.* the similarity is computed between single pixels only.

Non-Local Means Filtering The NLM method generalizes the bilateral filtering, it uses a more robust way to compute the similarity between the pixels. In the bilateral filter, the similarity is computed from the single pixel values F_p and F_q. However, because F is corrupted by noise, this might give wrong similarity values. The simple idea of the NLM method is to use *patches* around p and q to compute the similarity. The weight function for NLM reads

$$w_{p,q} = e^{-\frac{s_{p,q}}{h^2}}, \tag{2}$$

where s is a similarity measure between the pixel p and q defined as

$$s_{p,q} = \frac{1}{(2t+1)^2} \sum_{k=(-t,-t)}^{(t,t)} (F_{p+k} - F_{q+k})^2 \tag{3}$$

and h is a filtering parameter. The similarity is computed as the quadratic grayvalue distance between square patches of radius t centered on the pixels p and q respectively.
Since the search range (Q in eq. (1)) also includes the center pixel p itself, clearly the weight $w_{p,p}$ is always 1. In order to prevent giving excessive weight to the center pixel, a better strategy is to exclude the center pixel from the search range, and use the maximum weight of the remaining pixels as weight for the center pixel. For this purpose, we define a neighborhood $\widetilde{Q} = Q \setminus p$ around p that excludes p and modify eq. (1) to

$$U_p = \frac{\max_{q\in\widetilde{Q}}\{w_{p,q}\}F_p + \sum_{q\in\widetilde{Q}} w_{p,q}F_q}{\max_{q\in\widetilde{Q}}\{w_{p,q}\} + \sum_{q\in\widetilde{Q}} w_{p,q}} \tag{4}$$

3 Framework

We provide a framework with the basic structure of the NLM method. Your task is to implement the steps needed to calculate the filtered value U_p. This includes getting the patches from the image, compute the similarity and the weights (eq. (2) and (3)) and finally compute the filtered value (eq. (4)).

Tasks:

- Implement the NLM method
- In order to have a competing denoising method, also compute a denoised image using a simple gaussian filter
- Compare the *method noise* N_M of the gauss denoising and the NLM method. The method noise is the difference between the original image and the denoised image $N_M = U_{\text{true}} - U$. Ideally the method noise should resemble a gaussian noise of given variance. Compare the method noise of the gauss denoising and the NLM denoising. What do you find?
- To assess the quality of the denoised image, compute the Peak-Signal-to-Noise-Ratio (PSNR) between the original and the denoised image. The PSNR is a logarithmic error measure of the difference between two images based on the mean-squared-error (MSE). For two images F, G it is defined as

$$\mathrm{PSNR}(F,G) = 10\log_{10}\left(\frac{\max\{F\}}{\mathrm{MSE}(F,G)}\right)\mathrm{dB}, \tag{5}$$

  where $\mathrm{MSE}(F,G) = \frac{1}{NM}\sum_{i,j}(F_{i,j} - G_{i,j})^2$.
- Compare the PSNR of the noisy image and the denoised image using the gaussian filter and the NLM method. Try to find parameters for the gaussian filter and the NLM method that give good PSNR!

Experiment with your own images! Since the NLM method is based on the self similarity of the image, try to find an image where you think the NLM algorithm will do a good job (*i.e.* a lot of self similarity), and one where you think it will fail. What do you get in terms of PSNR? Is the assumption that the image with a lot of self similarity gives better denoising performance justified? Include your findings in the report!

Parameters The NLM method has 3 main parameters:

- Radius of the neighborhood Q to search for similar pixels (framework: `search_radius`)
- Radius of the patch t to compute similarities between pixels (framework: `patch_radius`)
- Filtering parameter h for computing the weights (framework: `sigma_weight`)

References

[1] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR 2005, pages 60–65, Washington, DC, USA, 2005. IEEE Computer Society.