Università
della
Svizzera
italiana

Faculty
of Informatics

Image and Video Processing (Spring 2024)

Assignment 2: Local Operations

March 21, 2024

1 Linear Motion Blur Filter [7 points]

The Gaussian kernels you have seen in the lecture are radially symmetric, i.e., isotropic. One can make Gaussian filters directional, i.e., anisotropic, by defining standard deviations separately for horizontal and vertical directions. More precisely, given two standard deviations σ_x and σ_y for the horizontal and vertical direction, respectively, the anisotropic Gaussian is defined by:

$$G(x, y) = A \cdot e^{-\left(\frac{x^2}{2\sigma_x^2} + \frac{y^2}{2\sigma_y^2}\right)}, \tag{1}$$

where A is the normalizing coefficient that makes the weights sum up to 1.
Additionally, we can rotate the Gaussian function by angle θ by applying rotation to the coordinates of the kernel. In practice, you define your rotated Gaussian as:

$$G(x, y) = A \cdot e^{-\left(\frac{x_\theta^2}{2\sigma_x^2} + \frac{y_\theta^2}{2\sigma_y^2}\right)}, \tag{2}$$

where

$$x_\theta = x \cdot cos(\theta) - y \cdot sin(\theta) \tag{3}$$

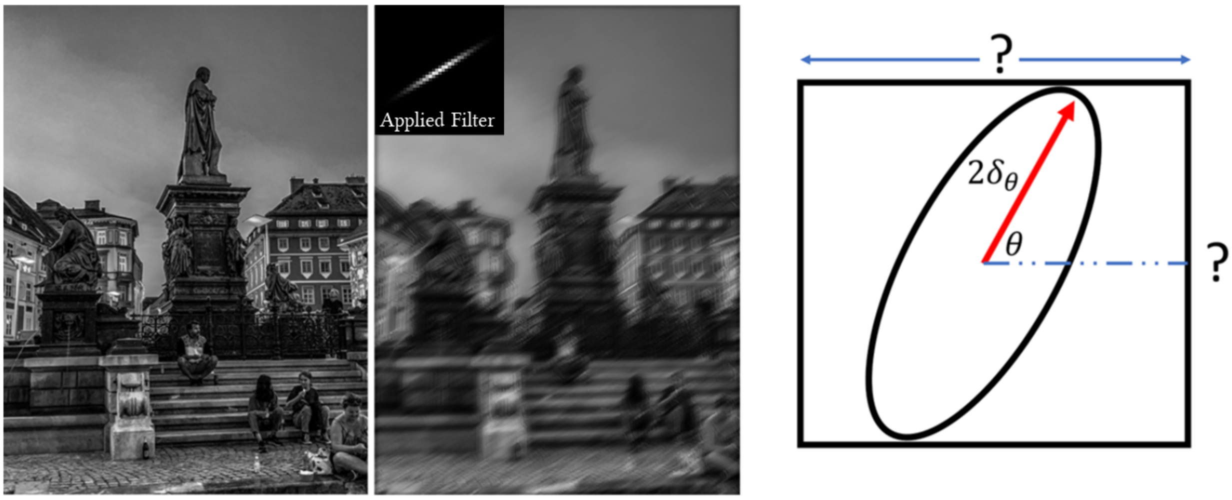
$$y_\theta = x \cdot sin(\theta) + y \cdot cos(\theta). \tag{4}$$

In this exercise, you will simulate a motion blur with the anisotropic Gaussian filter. The idea is to filter an image with a Gaussian filter which is oriented along the motion direction. To this end:

1. Implement a function that computes an anisotropic Gaussian kernel for blurring the image along the motion direction. The function should take as an input the angle of motion (θ), σ_x, and σ_y.
2. Apply this filter to the provided 'graz.png' image to simulate motion blur at an angle of 45 degrees, as shown in Figure 1(b). Experiment with parameters σ_x and σ_y to obtain the results you like. Include the resulting image and the filter in the report, similarly as demonstrated in Figure 1(b). You can use `conv2` function for this exercise.

2 Iterative filtering [6 points]

Applying a Gaussian filter to an image multiple times results in a similar result to applying one bigger Gaussian filter; see Figure 2. Verify empirically to what extent this is true also for a bilateral filter. For the experiments, you can use the implementation of the bilateral filter provided in MATLAB (`imbilatfilt`). Note that the two parameters of this function (`DegreeOfSmoothing` and `SpatialSigma`) correspond to standard deviation parameters of range and domain kernels discussed during the lecture. Present the results and observations of your experiments in the report. Please provide an explanation for any differences between the results of applying a bigger bilateral filter once versus applying multiple times a smaller filter.



(a): Input (b): Output (c): Filter Diagram

Figure 1: Motion Blurring

1x Gaussian filter (standard deviation = 6.0) 10x Gaussian filter (standard deviation = 2.0)

Figure 2: Example of applying iterative filtering. The image on the left presents the result of applying one big Gaussian filter. The image on the right presents the result of applying multiple times a much smaller Gaussian filter. Note the similarity of the results.

3 Image Stylization [7 points]

Design and implement a filtering procedure that performs image stylization visualized in Figure 3. More precisely, use the knowledge from the lectures on linear and non-linear filtering to develop a sequence of filtering steps that turn the image on the left in Figure 3 into the image on the left. You do not need to obtain the same, but somewhat similar results. However, we want you to focus on two aspects of the stylization you are asked to develop. First, it smoothes the image in a particular way. Second, it adds black lines along image discontinuities. These lines are approximately 2-3 pixels wide. In your report, explain the sequence of filtering steps and provide the intermediate results. For the implementation, you can use any MATLAB in-build functions which help you implement the filtering procedure. With the assignment, we provide image *delicate_arch.jpg*, which you should use for this exercise. In-case it is needed, we also provide a sample-script (ColorHandling.m) of how to separate the color from the gray channel; process the gray channel, and then re-colorize the image. Hint: Consider using an iterative filtering procedure for better results.


Input image

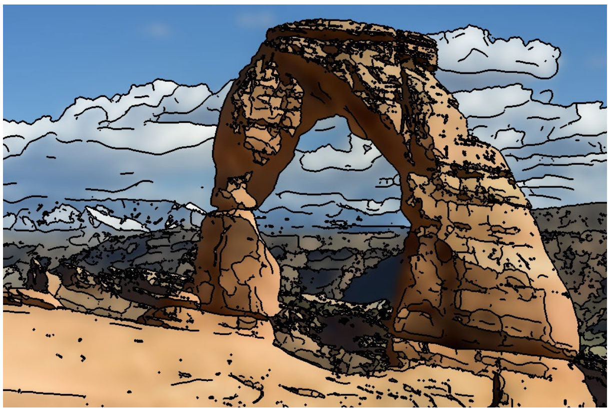

Output image

Figure 3: Stylization example

Submission

You should submit one ZIP-file via iCorsi containing:

- All your code in MATLAB appropriately commented, the processed pictures that you obtained and your own pictures if you did the Bonus, so that we can reproduce your results. **Failure to make your code reproducible will result in deducted points.**
- A complete PDF report detailing your solution and partial results for each exercise (Maximum 7 pages + 3 if you choose to do the Bonuses). **Points will be deducted for reports exceeding the limit.**

Grading will be **solely** based on the provided PDF report so we encourage clarity and detailed answers. We recommend using LaTeX or Overleaf to write the report. Usage of ChatGPT or any other natural language model is strictly prohibited and will be severely punished.

Solutions must be returned on April 10, 2024 via iCorsi3