

Smoothing Filters

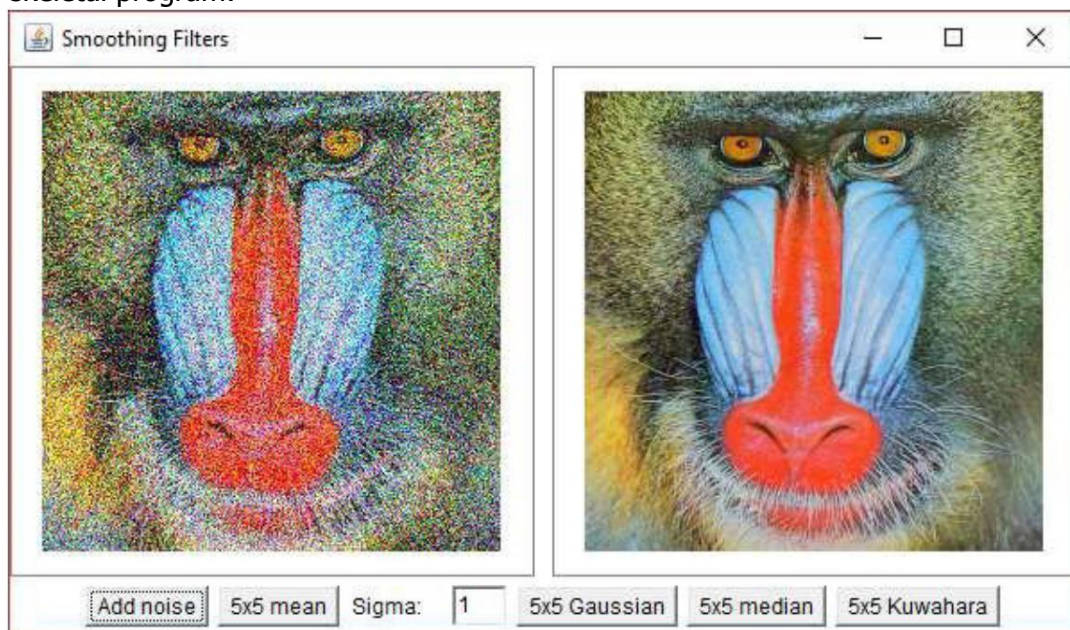
COMP 3301 — Assignment 2 Due: October 15, 2020 (Thursday) 11:59 PM

Objectives:

In this assignment you will implement four different image smoothing filters discussed in class. The goal is to give you a further understanding of image smoothing operation, as well as that you get familiar with image handling in Java.

Your Task:

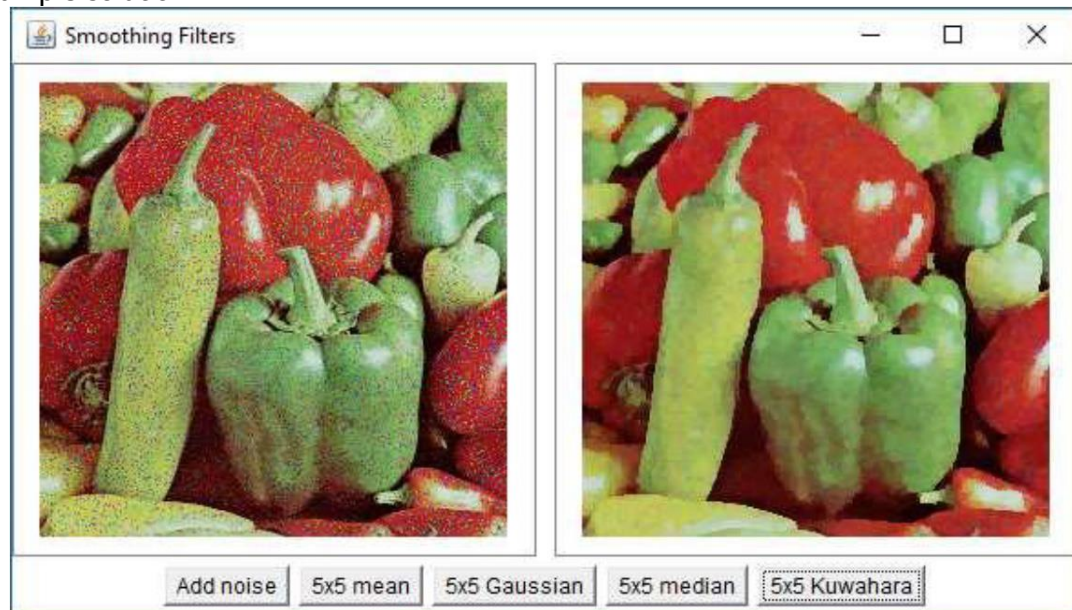
The four image smoothing filters you need to implement are 5×5 mean filter, 5×5 Gaussian filter with user-specified σ parameter, 5×5 median filter, and 5×5 Kuwahara filter. Making use of the most efficient implementation based on the property of the filters is required. Be careful when handling pixels near the boundaries of the image. If a pixel is outside of the image, the closest pixel that is in the image should be used. Output of the skeletal program:



Getting Started:

A skeletal program is supplied to get you started, which you are required to use as the basis of your implementation. To run the skeletal program, you need to put the testing images in the same folder. The program opens a window that contains two panels and displays the test image in both panels. As a demo, the program adds noise to the input image when the first button is pressed.

Output of a sample solution:



Grading:

Your program will be tested and graded using a standard Java environment. The grade will be based on your program's functionality (whether or not it works under different settings), as well as the efficiency of your implementation. The weights for different components are as follows:

- Result of 5x5 mean filtering 20%
- Result of 5x5 Gaussian filtering 20%
- Result of 5x5 median filtering 20%
- Result of 5x5 Kuwahara filtering 30%
- Updating the image on the for further filtering 10%

To do all these filters operations, the use of two separate image buffers will be necessary. For the last item (10%), add a button with a label "Update source". When pressed, the image on the left (the source) will become a copy of the image on the right (the result), and the filters will be applied on it, as if it was the new source image.

These are some foreseeable deductions on grades:

- If an optimization for a separable filter is missing, 5% will be deducted for each missing optimization
- If an optimization for an incremental filter is missing, 5% will be deducted for each missing optimization
- If the pixels at the border are not handled properly, 5% will be deducted for each filtering method
- If .java helper files are missing to compile your solution, 10% will be deducted
- If the documentation mentioned below is missing or incomplete, anywhere from 10% to 20% will be deducted
- If essential source files (.java) are missing, anywhere from 50% to 100% will be deducted, depending on how much is missing.

What and How to Hand in:

You are handing in the source of your program, as well as any other files required for running your program. Your source code must contain sufficient internal documentation to facilitate grading. This includes your names, student numbers, a brief description of what the programs do, and a listing of known bugs or things that are missing, if any, at the top of the file, and/or in a text file on the submission comments field/window. Send in your

source program through the Direct2Learn's Dropbox as a single .zip file. The submitted file must have the following naming convention: A2_GroupX_Lastname1Lastname2Lastname3.zip. No late submission is allowed.

Verifying your submission:

Since late submissions are not allowed, you should be careful to make sure each of your submissions has been properly done. Once you have placed your assignment into the Dropbox, click on the link that indicates our submission has been done, download it, open it, revise the contents, test that the program you have submitted actually is the one intended (not the source files, for example), and if it does not correspond to what you want the TA to mark, you must resubmit again, until you are satisfied. The TA will only mark the last submission with the timestamp that is prior to the indicated deadline.