Inhomogen Isotropic Diffusion Filter 2D

version 1.0en for ImageJ

Developed by Jan Friedrich Ehlenbröker, Alexander Maier, Uwe Mönks, Stefan Schwalowsky and Matthias Tobergte

Tested with ImageJ 1.36b and JRE 1.5.

Abstract

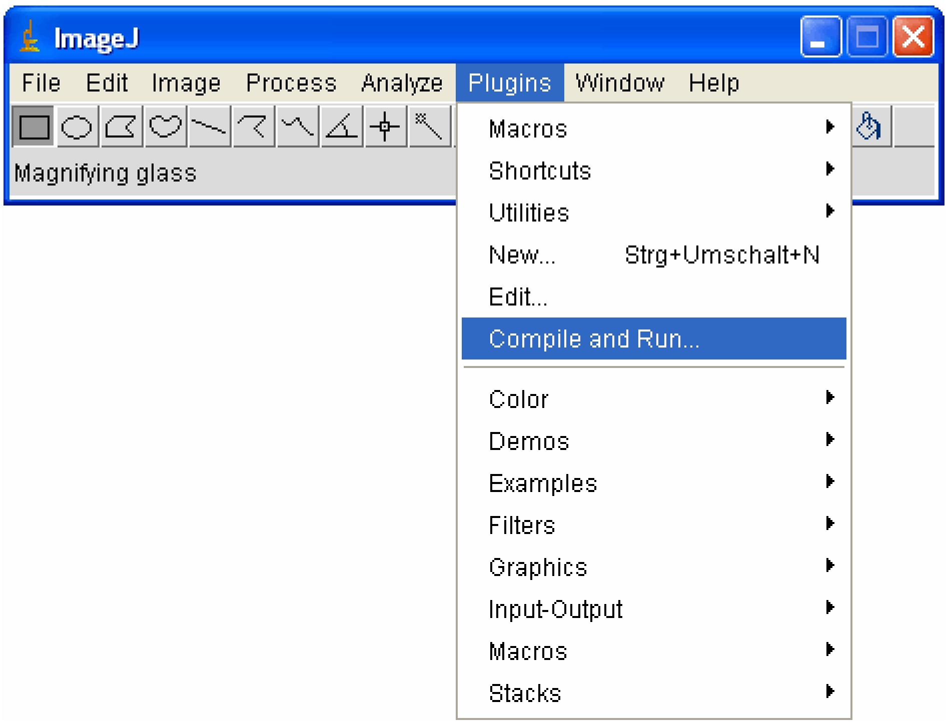
The filter is the result of a project we worked on during UaS Lippe and Höxter's (http://www.fh-luh.de/fb5/en) and Weidmüller's (http://www.weidmueller.com) *Summer Academy* 2006.
The task was to implement the Perona-Malik function to create an inhomogeneous isotropic diffusion filter that processes 8bit pictures in two dimensions to reduce noise in homogeneous areas and keep and even sharpen edges.

This manual gives you installation instructions and explains the filter's GUI.

Installation Instructions

1. To install the *Inhomogen Isotropic Diffusion Filter 2D* copy the `Inhomogen_Isotropic_Diffusion_2D.java` file in ImageJ's plugins folder. You may find it at `C:\Program Files\ImageJ\plugins`.
   It is also possible to create a subfolder and copy the file into this folder, e.g. `C:\Program Files\ImageJ\plugins\Diffusion Filter`. In the following we will use this folder.

2. In the next step we have to compile the filter. Start ImageJ, go to the "Plugins" tab and select "Compile and Run…"

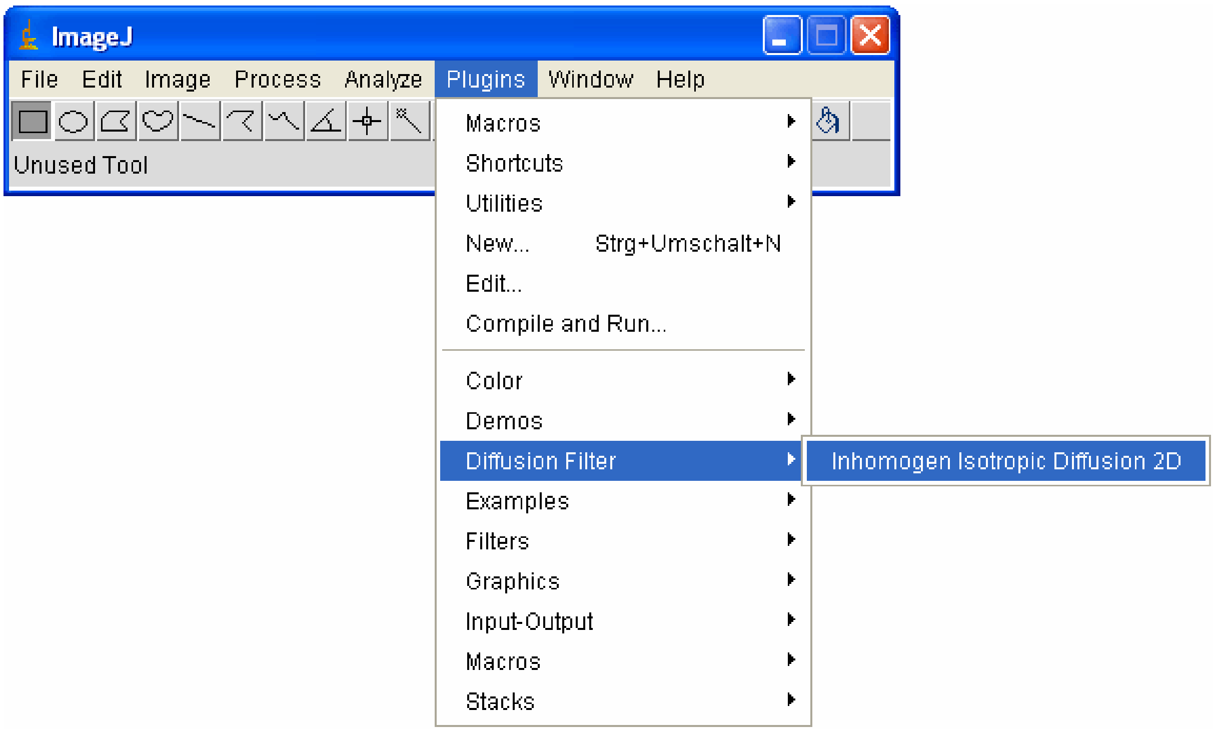


   This opens a dialog where you have to choose the `Inhomogen_Isotropic_Diffusion_2D.java` file you copied the step before. This compiles the filter and tries to process a picture in the next step. If there is no picture opened, you will get an error message that tells you this fact. This also tells you, that compiling was completed successfully and the filter is ready to work.

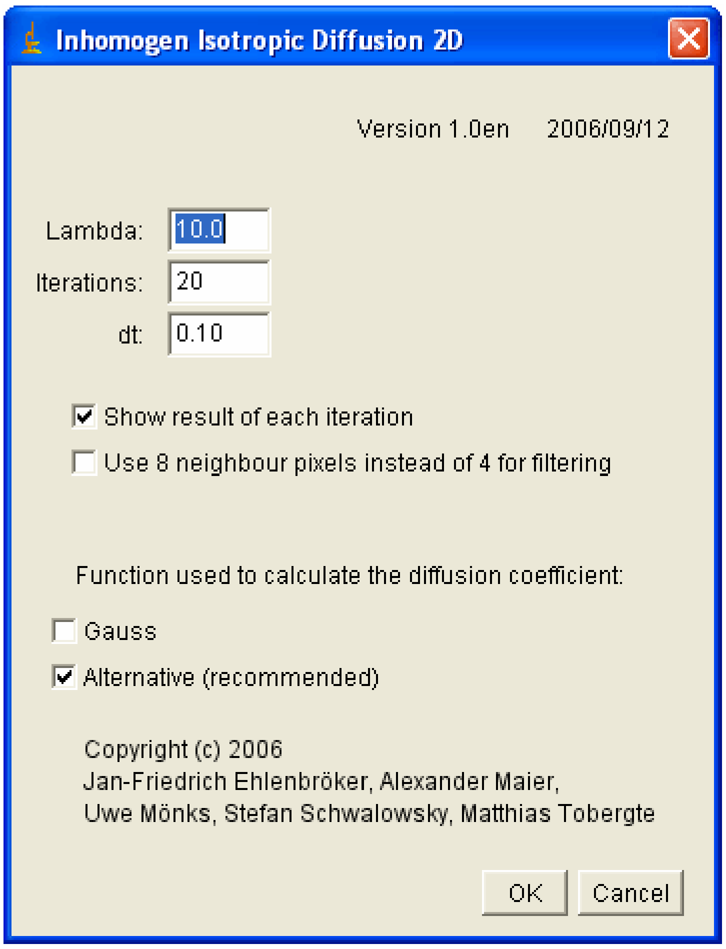
3. In the last installation step please close ImageJ and open it again – installation is finished!

Using the Inhomogen Isotropic Diffusion 2D Filter

This chapter explains the filter's GUI.
To use the filter, open an image in ImageJ. Then choose our filter from the Plugins tab. You can find it in the *Diffusion Filter* folder.



This opens the filter's GUI:



Version 1.0en 2006/09/12

Lambda: 10.0
Iterations: 20
dt: 0.10

[✓] Show result of each iteration
[ ] Use 8 neighbour pixels instead of 4 for filtering

Function used to calculate the diffusion coefficient:

[ ] Gauss
[✓] Alternative (recommended)

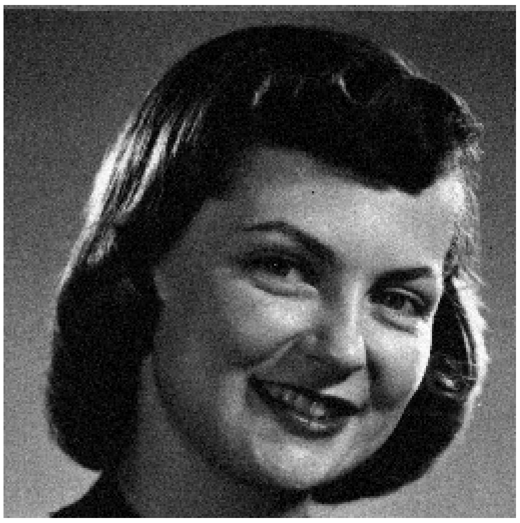
Copyright (c) 2006
Jan-Friedrich Ehlenbröker, Alexander Maier,
Uwe Mönks, Stefan Schwalowsky, Matthias Tobergte

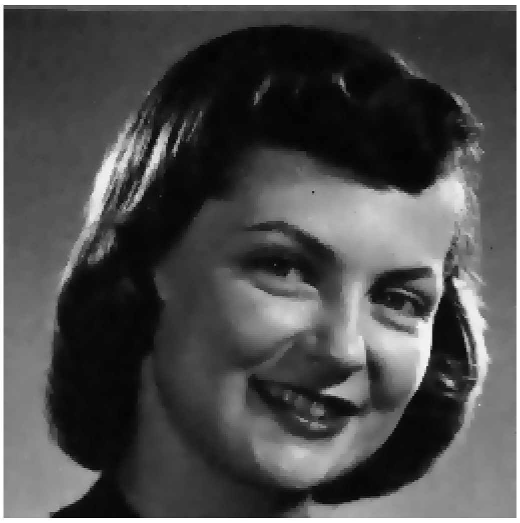
The following table explains all inputs of this dialog. The functions, that can be chosen to calculate the diffusion coefficient, are explained afterwards.

Lambda	Parameter used to manipulate calculation of the diffusion coefficient. This defines which amount of the neighbour pixel's grey value has effect diffuses to the actually calculated pixel.
Iterations	Parameter defining how often the picture is processed.
dt	Parameter used to define the weight of each iteration. This means the calculated grey value is not taken 1:1 but multiplied with this factor. On the one hand you might more iterations, but on the other hand it makes the result look better. **Remember:** Values greater than 0.14 or less than 0 are not valid.
Show result of each iteration	If this box is checked, ImageJ will show you an image stack besides your processed picture. You can scroll through it to see what has changed from one iteration to the next. Therefore it is also possible to use a previously generated image instead of the last one in the row.
Use 8 neighbour pixels instead of 4 for filtering	If this box is checked, the filter will use the grey values of all eight neighbour pixels for calculation. Otherwise the filter will just use the upper, lower, left and right pixels' grey values.

It is possible to choose two different functions for calculating the diffusion coefficient. One is like the Gaussian function called *Gauss* in this case, another is a kind of $1/x^2$ function called *Alternative*. Both of these functions were originally proposed by Perona and Mailk in their article "Scal-Space and Edge Detection Using Anisotropic Diffusion", IEEE Transactions on Pattern Analysis and Machine Intelligence, Washington D.C. 1990, number 12, pages 629 – 639.
We found out, that in most cases the *Alternative* function brings better results, but feel free to try on your own!

Sample Results


Original


Filtered by Inhomogen Isotropic Diffusion 2D

Contact

For issues concerning the filter feel free to contact Uwe Mönks via email:

uwe.moenks(at)gmx.de