**User Guide**

Fiji plug-in: VSNR\_Macro\_2D and VSNR\_Macro\_3D

Font Benjamin / Weiss Pierre

July XX, 2015



**Contents**

1. **Acknowledgment………………………………………………3**
2. **Introduction…………………………………………………....3**
3. **How to use theses plug-ins…………………………………….4**

* **Installation…………………………………………………..4**
* **VSNR\_Macro\_2D…………………………………………..4**
* **VSNR\_Macro\_3D…………………………………………..9**
* **Use in Fiji macros…………………………………………10**

1. **Practical examples.…….……………………………………11**

* **Gabor filter………………………………………………...11**
* **Dirac filter…………………………………………………12**

1. **Known limitations of theses plug-ins...……………………..13**
2. **Macro examples………...……………………………………14**
3. **Acknowledgment**

Even thought I have developed theses plug-ins (as an intern student), many people contributed to their creation.

Theses plug-ins are based on the codes of the VSNR\_V2, developed by Pierre Weiss, Léo Mouly (an intern student of Toulouse University in computer science) and Omar Dounia (an intern student of the Toulouse University in mathematical engineering).

The mathematical algorithms were developed by Jérôme Fehrenbach and Pierre Weiss (from the PRIMO team of ITAV).

Many people at ITAV-USR3505 contributed to improve the plug-in with their feedback and help, particulary Mathieu Vigneau, Lise Teyssedre, Guillaume De Brito.

1. **Introduction**

This manual is written to explain the main features and the possible usages of the VSNR\_Macro\_2D and VSNR\_Macro\_3D plug-ins. Theses plug-ins are a new implementation of the VSNR denoising algorithm. A Fiji plug-in already exists but is not usable in macros (this plug-in is available here: www.math.univ-toulouse.fr/~weiss/PageCodes.html). This is why the PRIMO Team has decided to create them, in order to denoise images using Fiji macros, which open many possibilities in processing large image data volumes.

1. **How to use theses plug-ins**
2. **Installation**

The installation of theses plug-ins is the same as any other Fiji plug-in: just copy/paste the *VSNR\_Macro\_2D.jar* and *VSNR\_Macro\_3D.jar* files into the plug-in directory of Fiji.

You can also use the function *Install Plug-in…* in the *Plug-ins* tab of Fiji.

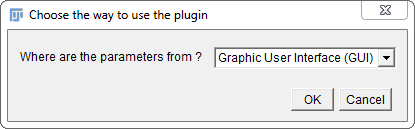
Each plug-in will appear in *Plug-in → Process*.

1. **VSNR\_Macro\_2D**

The plug-in VSNR\_Macro\_2D is used to denoise simple 2D images. You can use this plug-in to tests some parameters before use the VSNR\_Macro\_3D on large data.

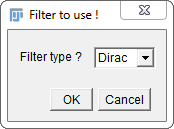
The plug-in is simple to use:

1. Open an image to denoise and launch the plug-in.
2. You have to choose among two way to use the plug-in :



* Using the Graphic User Interface (GUI): the classic way to use it, it ask you the different parameters using a series of pop-up windows.
* Using the Text File: you can read the parameters from a text file (.txt), this is the simplest way to use the plug-in, once you have found the good parameters. If you choose this, you’ll have to select the text file on your computer. The plug-in will denoise the image and display the result and the filter used on the screen. In order to make easy the creation of the text file, we give you some models in the example part of this guide. The text file is also generated when you use the Graphic User Interface.

1. If you chose the GUI, you’ll have to choose the filter to use.

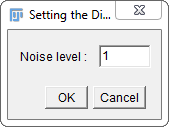


There are two filters available:

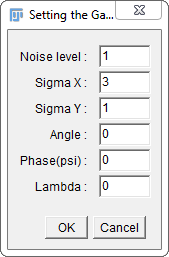
* The Dirac filter: this filter is used to remove the standard white noise (might be Gaussian, Poisson, impulse …).
* The Gabor filter: this filter is used to remove the ray noise.

1. You can now enter the filter parameters :

* If you choose the Dirac filter, the only parameter is the Noise level.



* If you choose the Gabor filter, there are more parameters.



The Noise level: same function as the Dirac filter.

The Sigma X: length of X axis.

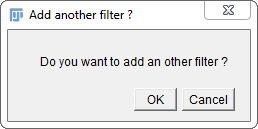
The Sigma Y: length of Y axis.

The Angle: orientation of the rays to remove.

The Phase: may be used to shift the frequency axis.

The Lambda: frequency (the Gabor function might oscillate more or less).

1. The next Window asks if you want to add another filter in order to remove a different noise of your image. If you click on OK, you will return to the step 3. This can be useful if your image have rays of different orientations.

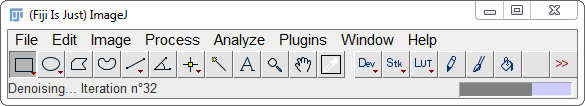


1. Once all your filters are set, you must choose the algorithm to use :



* TV: this algorithm requires a number of iteration that you decide of. It is long but the results are pretty good. Contrarily to some standard algorithms in image processing, the solution will not change if many iterations are performed. It is important to make enough iterations to find a solution sufficiently close to what you are looking for, but making too many will not really change the solution.
* H1: this algorithm is really faster than the TV, but the results it provides are less good. It doesn’t require entering the number of iterations.

1. You now have to wait the results of the denoising. A progress-bar on the Fiji interface shows the progression of the algorithm.



The plug-in will display the filter used and the denoised image in two different windows.

The log window will also print the text of the text file corresponding to the parameters entered. In that way, the next denoising will be easier and the parameters can be saved if they are good.

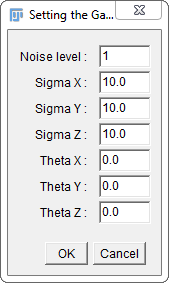
1. **VSNR\_Macro\_3D**

The plug-in VSNR\_Macro\_3D is used to denoise stacks. You can use this plug-in once you have found the good parameters. This plug-in can be very useful to denoise large data volumes. Its functionnement is pretty the same as the VSNR\_Macro\_2D, except that the parameters are not the same (this is due to the 3D nature of the images).

Only the step 4 (choice of the filter’s parameters) is different from the VSNR\_Macro\_2D.

1. You can now enter the filter parameters :

* If you choose the Dirac filter, the only parameter is the Noise level.
* If you choose the Gabor filter, there are more parameters:



The Noise level: same function as the Dirac filter.

Sigma X: length of X axis.

Sigma Y: length of Y axis.

Sigma Z: length of Z axis.

Theta X: angle of X axis.

Theta Y: angle of Y axis.

Theta Z: angle of Z axis.

If you want more information about theses parameters, you can consult the VSNR\_V2 Reference Manual written by Pierre Weiss which is available at this adress: www.math.univ-toulouse.fr/~weiss/Codes/VSNR/Documentation\_VSNR\_V2\_Fiji.pdf.

1. **Use in Fiji macro**

Theses plug-ins are mostly useful when they are used in Fiji macro.  
Indeed, the last version of the plug-in was not made to be integrated in macro, which is not quite practical. The new implementation of the VSNR can now be used automatically thanks to the macros.

Some examples of macros are given in the package of the plug-ins.

The first one use the VSNR\_Macro\_2D and denoise all the images of a chosen directory and then save them into another directory.

The second one is made for the stacks and uses the VSNR\_Macro\_3D. It divides the stacks of a chosen directory in substack of a chosen length and denoise them before save them into another directory.

These macros are the simplest way to use these plugins, but we can imagine a lot of applications for them. Examples are given at the end of this guide.

**4) Practical example**

**a) Gabor filter**

**C:\Users\benjamin.font\Desktop\teststackBenjamin.tif**

C:\Users\benjamin.font\Desktop\Denoised Imageh1.tifC:\Users\benjamin.font\Desktop\Denoised Imagetv.tif

**Top : original image - Left : Denoised image using TV - Right : Denoised image using H1**

*Denoising\_Algorithm: H1*

*Iteration\_Number: 50.0*

*\*\*\**

*Filter\_Type: Gabor*

*Noise\_Level: 0.15*

*Sigma\_X: 3.0*

*Sigma\_Y: 1.0*

*Angle: 0.0*

*Phase(psi): 0.0*

*Lambda: 0.0*

*\*\*\**

*Denoising\_Algorithm: TV*

*Iteration\_Number: 5.0*

*\*\*\**

*Filter\_Type: Gabor*

*Noise\_Level: 1*

*Sigma\_X: 5.0*

*Sigma\_Y: 1.0*

*Angle: 0.0*

*Phase(psi): 0.0*

*Lambda: 0.0*

*\*\*\**

**b) Dirac filter**



C:\Users\benjamin.font\Desktop\Denoised Imagefishh1.tifC:\Users\benjamin.font\Desktop\Denoised Image.tif

**Top : original image - Left : Denoised image using TV - Right : Denoised image using H1**

*Denoising\_Algorithm: TV*

*Iteration\_Number: 50.0*

*\*\*\**

*Filter\_Type: Dirac*

*Noise\_Level: 1.0*

*\*\*\**

*Denoising\_Algorithm: H1*

*Iteration\_Number: 50.0*

*\*\*\**

*Filter\_Type: Dirac*

*Noise\_Level: 0.4*

*\*\*\**

**5) Known limitation of theses plugins**

**Some of the denoising results are black ?**

If your results are black, don’t panic, it is normal. You just have to adjust the contrast of the image using Fiji and your denoised image will appear.

**The processing is very long !**

Indeed, the processing time of the VSNR is very long when used on large stacks. One way to solve this problem is to use a macro which will automatically denoise many images, even when you are not in front of the computer.

**I have problems of “out of memory” when I want to denoise large stacks !**

You might encounter memory troubles with very large images. One way to solve this problem is to use a macro that divide your stack in substacks and denoise each one of them in succession.

**6) Macro examples**

This macro opens all the images in a chosen directory, denoises them using the parameters of a text file and save them in another chosen directory.

You must put the good path of the parameters. To get easily the different paths, you can use the Fiji macro recorder.

*//= = = = = = = = = = = = = = = = = = = = = = = = = = = =//*

*// //*

*// HOW TO USE VSNR\_PLUGIN\_2D ON A COMPLETE DIRECTORY //*

*// //*

*//= = = = = = = = = = = = = = = = = = = = = = = = = = = =//*

*// this variable is the input folder*

*input = "C:\\Users\\benjamin.font\\Desktop\\StackLise\\";*

*// this variable is the output folder*

*output = "C:\\Users\\benjamin.font\\Desktop\\DenoisedStackLise\\";*

*list = getFileList(input);*

*setBatchMode(true);*

*for(i = 0;i<list.length;i++){*

*denoiseImage(input, output, list[i]);*

*}*

*function denoiseImage(input, output,filename){*

*open(input+filename);*

*run("VSNR\_Macro\_2D", "where=[Text File (.txt)] choose=[C:\\Users\\benjamin.font\\Desktop\\log.txt]");*

*close();*

*saveAs("Tiff", output+filename+"\_denoised");*

*close();*

*close();*

*}*

This second macro opens all the stacks in a chosen directory, denoises them using the parameters of a text file and saves them in another chosen directory. The macro divides the stacks in many substacks of a chosen length. So you have series of substacks saved for each stack denoised.

*//= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = //*

*// //*

*// HOW TO USE THE VSNR\_MACRO\_3D PLUGIN BY MAKING SUBSTACKs //*

*// //*

*//= = = = = = = = = = = = = = = = = = = = = = = = = = = = = = //*

*// this variable is the input folder*

*input = "C:\\Users\\benjamin.font\\Desktop\\Testmacro3D\\";*

*// this variable is the output folder*

*output = "C:\\Users\\benjamin.font\\Desktop\\DenoisedTestmacro3D\\";*

*// this variable is the text file*

*textfile = "C:\\Users\\benjamin.font\\Desktop\\text.txt";*

*// this variable is the limiter of the size of each substack to denoise for a stack (must be greater than 1)*

*max = 3;*

*// beginning of the algorithm*

*list = getFileList(input);*

*for(i = 0;i<list.length;i++){*

*denoiseStack(input, output, list[i],max,textfile);*

*close();*

*}*

*// this function denoise a stack using the parameters of the text file*

*// input : the path of the source folder*

*// output : the path of the destination folder*

*// filename : the name of the file*

*// maxsize : the maximum size of the substack to denoise*

*// textfile : the path the text file to use*

*function denoiseStack(input,output,filename,maxsize,textfile){*

*open(input+filename);*

*size = nSlices;*

*begin = 1;*

*end = 0;*

*rest = size%maxsize;*

*nb = (size-rest) / maxsize;*

*for(i = 0 ; i<nb;i++){*

*// we determine the part of the stack to denoise*

*begin = end+1;*

*last = nb-1;*

*if(i!=nb){*

*if(i==last){*

*end = size;*

*}*

*else{*

*end = begin+maxsize-1;*

*}*

*}*

*// we make a substack and we denoise it*

*run("Make Substack...", " slices="+begin+"-"+end);*

*run("VSNR\_Macro\_3D", "where=[Text File (.txt)] choose=["+textfile+"]");*

*selectWindow("Denoised Image");*

*saveAs("Tiff", output+filename+"\_denoised"+i);*

*close();*

*close();*

*close();*

*}*

*}*