User Guide for OPTMISME

OPTIMIsation Stochastique en imagerie MultispectralE, Image-J plugin

Dominique Benielli

Labex Archimède

September 4, 2017

Contents

1 Introduction

1.1 Historical development

The scientific project was at first a french project for young researchers financed by the Gdr ISIS in 2013. It then continued and extended with the financial support of CNRS within the project Imag'In in 2015-2016 under the name OPTIMISME for *OPTIMIsation Stochastique en imagerie MultispectralE* i.e. Stochatic Optimization in Multispectral Imagery.

This project focuses on the restoration of two-photon microscopy data. It is composed of two parts: a theoretical approach and its application to two-photon microscopy data.

In the theoretical part, proximal algorithms have been developed, one of them is based on a Majorization-Minimization procedure. In the applicative part, the developed algorithms were used to restore two-photon microscopy images. Furthermore this approach included an analysis of the noise and the non-stationarity of the PSF.

1.2 Abstract

Modern approaches of inverse problems resolution in the field of multi- or hyper-spectral imaging are based on variational formulations. The goal of this study is to provide a generalization of parallel methods using recent advances of stochastic optimization, for the treatment of massive big data. The algorithms provided solve the multispectral deconvolution problem in two-photon microscopy.

1.3 Scientific issues

The in-vivo observation of a mouse brain by a two-photon microscope leads to direct images at cellular scales, which are degraded by blur and noise and need restoration. This problem yields an inverse problem that was targeted by the OPTIMISME project. The issue of obtaining the PSF of the two-photon microscope was solved by observing fluorescent micro beads (of diameter 0.5 μm) at different depths. The PSF varies with the depth (z coordinate). The OPTIMISME-ImageJ plugin implements the Majorization-Minimization proximal algorithm described in [1] to restore an image that has been blurred by a known PSF.

1.4 ImageJ

ImageJ is an image processing software dedicated to multidimensional scientific images. ImageJ can be easily extended adding plugins and scripts. ImageJ web site:

- imageJ homepage

2 Plugin OPTIMISME

ImageJ gives the possibility to user to develop and install one's own plugin. The following page explain how to install an ImageJ plugin :

- imageJ Plugin Install

2.1 Libraries

Libraries used for the OPTIMISME algorithm are:

- *edu.emory.mathcs.jtransforms* For 3D and 2D FFT and IFFT. On license BSD clause 2. Non contaminant license.
- *org.ejml* a part of ejml library for matrix calculus and pseudoInvers function. On Apache license Non contaminant license.
- *org.apache.commons.math3* a part of apache common library for tricubic interpolation. on Apache license Non contaminant license.

2.2 System requirements

The OPTIMISME plugin requires version 1.48 or higher of ImageJ (or Fiji) and java 8 or higher. (To check your installation click on *Help>About ImageJ..*, the java version must be 1.8.0 or higher).

2.3 Installation

The plugin OPTIMISME is available under the following formats.

2.3.1 Private Distribution, .class

To install it,

- Add the directory OPTIMISME in the plugin directory of ImageJ (or Fiji) (with all the *class files). Then start ImageJ (or Fiji).
- If the plugin item "OPTIMISME" does not appear in the menu : Go to Help>Refresh Menus or (better) restart ImageJ (or Fiji).

2.4 Public Distribution

If you are using Fiji, it is possible to install the plugin through the public distribution using the install plugin menu (*Plugins>Install PlugIn..*).

2.4.1 Private Distribution, .java

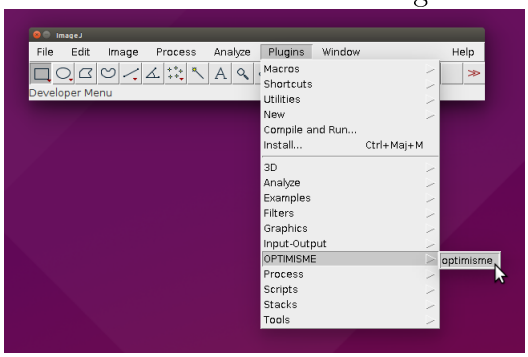
For earlier versions of ImageJ or Java, it may be necessary to compile the plugin. The java files are available in the OPTIMISME folder. To compile them,

- Add the directory OPTIMISME in the plugin directory of ImageJ.
- Rename the file *optimisme.txt* as *optimisme_.java*.
- Start ImageJ.
- Click on Plugins > Compile and Run.
- Select the file *optimisme_.java*.
- If the plugin item "OPTIMISME" does not appear in the menu : Go to Help>Refresh Menus or (better) restart ImageJ.

2.5 Launch

The Plugin is accessible through the "Plugins" menu of ImageJ. The OPTIMISME plugin assumes that at least two images are opened: the image to restored and the PSF. Note that the present version is established with one single PSF.

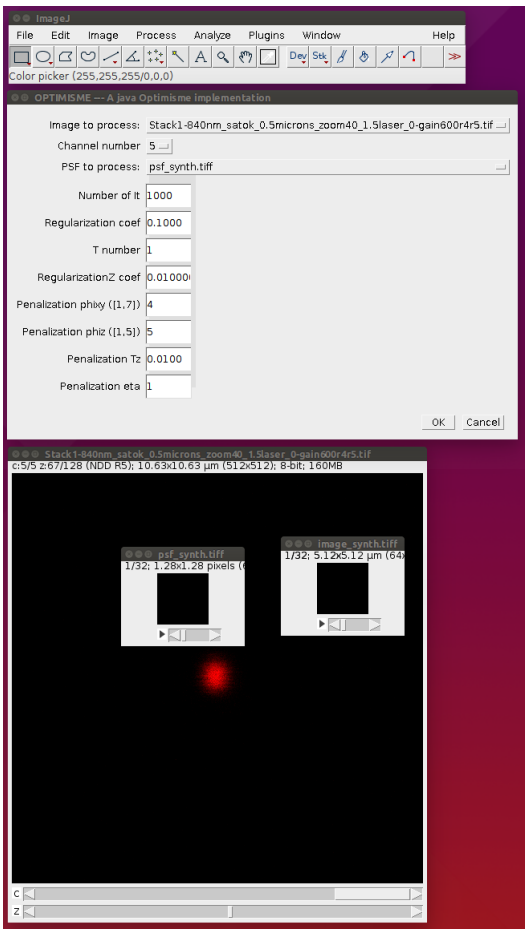
Launch the OPTIMISME Plugin as follows:



In the first window of the plugin OPTIMISME, the user specifies the input image, the channel that needs restoration, and the PSF image. Others parameters of the algorithm described hereafter, may also be specified (see [1] for more details).

2.5.1 Input parameters

In the window of OPTIMISME Plugin, the Input parameters have default values, that can be modified by the user.



- NbIt: number of iteration of the algorithm that are performed.
- regul: regularization coefficient for XY coordinate (space)
- T: smoothing parameter for the XY regularization term
- regulZ: regularization coefficient for the Z coordinate (depth)
- phixy: choice of the XY regularization function
- phiz: choice of the Z regularization function
- TZ: smoothing parameter for the Z regularization term
- eta: regularization parameter to force the solution to be positive
- Channel number: in case the input image contains several channels, the user can select the channel to restore.

2.6 Input Files

Before you launch OPTIMISME, you have to open two images corresponding to 1) the image to be processed and 2) the PSF to be used. OPTIMISME needs as input file two images (image and PSF). ImageJ accepts different formats (tiff format, lsm directly issued from microscope...).

- File>Open open at least two files one for Image and one for the PSF. OPTIMISME assumes 3D files, images with a depth dimension.

2.7 Computations

The algorithm first compares the resolutions of the image and the PSF: if resolutions are different the PSF is re-sampled to the resolution of the image; at the end of this first stage the re-sampled PSF is plotted (it may then be stored by the user for a later use). The plugin then proceeds with the actual restoration algorithm (see [1] for more details), and finishes by plotting of the processed image.

2.8 Results

The OPTIMISME plugin outputs at least one image

- At the end of the computations an image is opened: it is the deconvolved image, the user can save it using the *File>Save as* menu.
- When the input images (image and PSF) do not have the same resolution, the algorithm first re-samples the PSF; then, at the end of this first stage, the program displays the re-sampled PSF with the same resolution as the input image. The user can save it for further runs of the program and use it for all the images having the same resolution using the *File>Save as* menu.

3 Package OPTIMISME

3.1 The java class

The java class of algorithm Majorization/Minimization in ImageJ plugin, and other dependencies

```
/ImageJ/plugins/OPTIMISME/
├── org.jar
├── fftj.jar
├── doc/
│   ├── index.html
│   └── ...
├── optimisme_.java
└── data
```

```
    ├── image_synth.tiff
    ├── psf_synth.tiff
    └── psf_synth_subsampled.tiff
└── optimisme
    ├── MM.java
    ├── PSFPreparator.java
    ├── MMCal.java
    └── test
        ├── AllTests.java
        ├── MMCalTest.java
        ├── MMTest.m
        └── PSFPreparatorTest.m
```

3.2 Test Coverage

Tests files are in the test folder. The coverage by Emma leads to :

optimisme (3 avr. 2017 18:01:58) > OPTIM > OPTIM

OPTIM

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
optimisme		92 %		89 %	93	549	120	1 317	20	133	1	6
default		0 %		0 %	30	30	128	128	4	4	1	1
optimisme.test		99 %		90 %	17	237	53	1 816	6	132	1	4
Total	1 568 of 56 366	97 %	162 of 1 066	85 %	140	816	301	3 261	30	269	3	11

optimisme (3 avr. 2017 18:01:58) > OPTIM > OPTIM > optimisme — Source Files Sessions

optimisme

Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
MM		82 %		60 %	52	104	89	463	10	30	0	1
MMCal		98 %		95 %	27	364	14	643	2	84	0	1
MM.MMParameter		28 %		n/a	1	2	11	24	1	2	0	1
TriCubicInterpolator		0 %		n/a	7	7	6	6	7	7	1	1
MMCal.Squeeze		100 %		95 %	6	67	0	124	0	8	0	1
PSFPreparator		100 %		100 %	0	5	0	57	0	2	0	1
Total	719 of 9 528	92 %	89 of 804	89 %	93	549	120	1 317	20	133	1	6

3.3 Tests

source files	file size	%CPU	%MEM	%TIME
stark1_840 *2	342 (512*512*261)	100	61	763 -stop
stark1_450 *2	225 (512*512*172)	100	61	763 -stop
stark1_840 * psf_synth	167 (512*512*128)	100	7.4	22:20?
stark1_840 * psf_synth	167 (512*512*128	324	81	3:00

References

[1] E. Chouzenoux, L. Lamasse, S. Anthoine, C. Chaux, A. Jaouen, I. Vanzetta, and F. Debarbieux. Approche variationnelle pour la déconvolution rapide de données 3D en microscopie biphotonique. In *Actes du 25e colloque GRETSI*, Lyon, France, September 2015.