

Sparse deconvolution

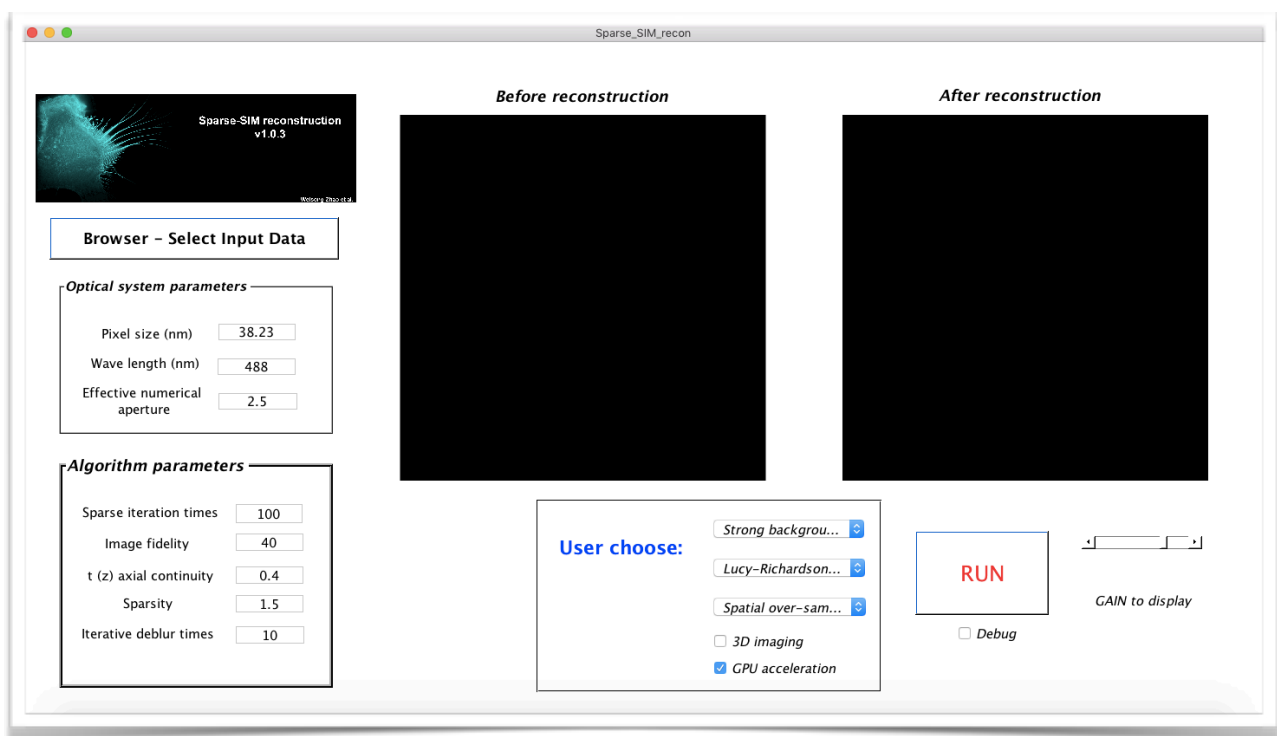
User Manual: Version(1.0.3)

Weisong Zhao - 2021/7/10

It is a part of the publication:

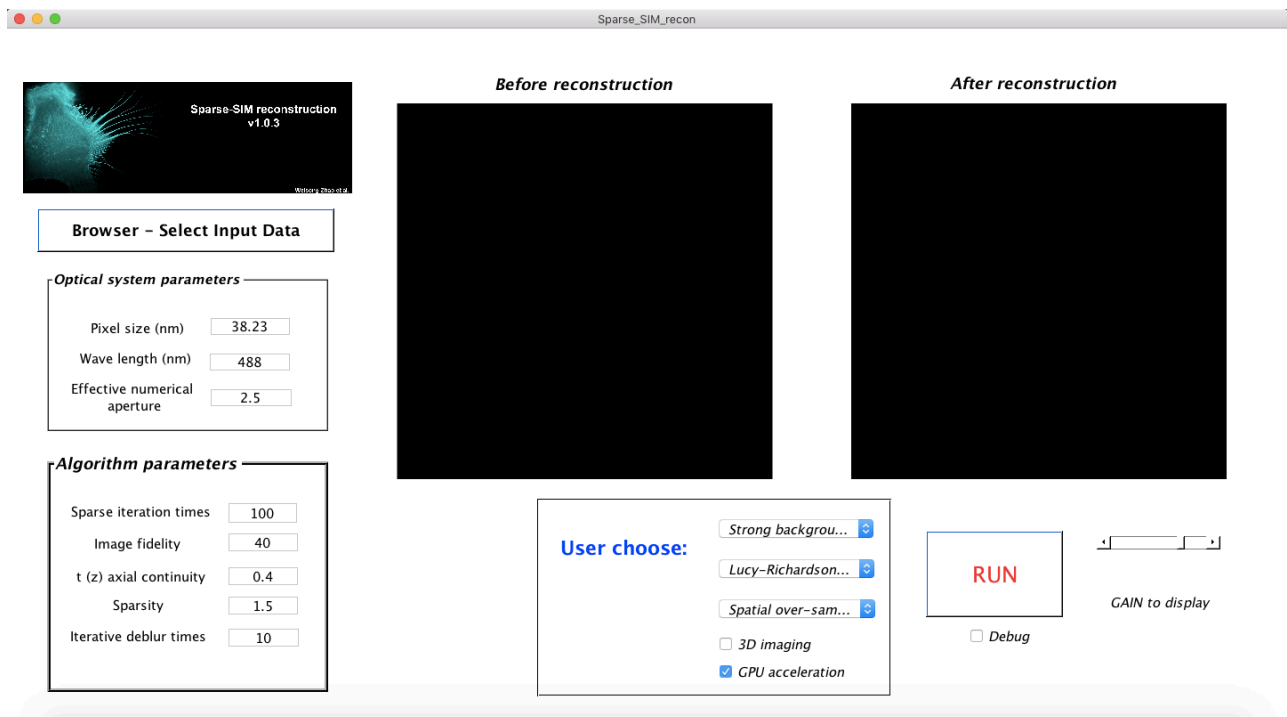
Improving live-cell super-resolution fluorescence microscopy with sparse deconvolution

The Sparse deconvolution is an universal post-processing framework for fluorescent (or intensity-based) image restoration, including xy (2D), xy-t (2D along t axis), and xy-z (3D) images. It is based on the natural priori knowledge of forward fluorescent imaging model: sparsity and continuity along xy-t (z) axes.



Details

- ❖ **Input data structure:** It should be the **.tif** data.
- ❖ This software is not designed to be compatible with various data structures for now, the update version of source code in [GitHub](https://weisongzhao.github.io/Sparse-SIM/) e.g. <https://weisongzhao.github.io/Sparse-SIM/> maybe compatible with the various data.
- ❖ **Easy to use:**
- ❖ If you are using Win10-64bit or MacOS-64bit systems, the SparseSIM_web_installer.exe, and SparseSIM_web_installer.app are provided in the link <https://github.com/WeisongZhao/Sparse-SIM/releases/tag/v1.0.3>. Execute the SparseSIM_web_installer.exe or SparseSIM_web_installer.app and follow the instructions. The APP will be installed at /Sparse_SIM/application/Sparse_SIM.exe (.app) as default. Or you can download the Matlab runtime v9.3 (for 2017b, [Win10](#), or [MacOS](#)) at first, and execute the for Matlab users/Sparse_SIM.exe (.app). Additionally, if you are a Matlab 2017b user, (great!) you can execute the for Matlab users/Sparse_SIM.exe or Sparse_SIM.app directly (without downloading these things through internet).
- ❖ Of course, you can use the source code directly: for Win10 is src_win/install.m and for MacOS or Unix-like users, please just run the src_unix/install.m and the GUI will pop up:



Parameters of GUI

- **Browser:** Select your data to reconstruct;
- **Pixel size:** The pixel size of input images;
- **Wave length:** The emission wavelength of the fluorescence probes.
- **Effective numerical aperture:** Effective NA of images after sparse reconstruction;
- **Sparse iteration times:** The iteration times of Sparsity reconstruction;
- **Image fidelity:** The parameter for images fidelity (distance between the images after and before reconstruction);
- **t (z)-axial continuity:** The t or z axial continuity of input video or volume;
- **Sparsity:** The sparsity of the data;
- **Iterative deconvolution times:** The iteration times of post deconvolution;
- **GPU acceleration:** Whether or not to compute on CUDA-GPU;
- **3D imaging:** The input images are whether or not 3D volume;
- **GAIN:** The brightness of the image can be altered via this control bar;
- **Iterative deconvolution:** The iterative deconvolution methods chosen by user.

Three options: accelerated **LR**, **LW** methods, and no deconvolution. Interestingly, even with Nesterov momentum acceleration, the **LW** method may be much slower than the vector extrapolation version **LR** method. So, the common iteration times of **LR** is 5~15, and **LW** is 30~50;

- **Over/up sampling:** The over-sampling methods (written as up-sampling in main text) chosen by user to achieve better image quality.

Three options: **spatial**, and **Fourier** over-sampling, and no over-sampling. As the **Fourier** over-sampling is sensitive to noise, the low SNR data is recommended to be processed with **spatial** over-sampling method;

- **Background:** The background estimation level chosen by user.

Six options: **Strong background (HI)**, **Weak background (HI)**, **Strong background (LI)**, **With background (LI)**, **Weak background (LI)** and **No background** are provided. **HI** is the High-dose illumination, and **LI** means low-dose illumination, which denote to two different types of background.

For the strong fluorescence background originated from out-of-focus emission and cellular auto-fluorescence (usually contained in high SNR data)

is regarded as **HI**. The low, and stable noise-like background (commonly exhibited in low SNR data) is seen as **LI**.

Furthermore:

- The maximum intensity projection (MIP) is used to display the 3D volumetric data in the Sparse-SIM UI. Users can adjust parameter values of the **Image fidelity**, **t (z)-axial continuity**, **Sparsity**, **Iterative deconvolution times**, **Background (including 6 options)**, and **Over sampling (including 3 options)** to enhance the image quality.
- The **debug** mode (choose first 5 frames from the input image stacks) is offered to help users to select the parameters more quickly.

Parameters

	<i>2D-SIM Actin Fig4a</i>	<i>2D-SIM LysoView Fig4m</i>	<i>TIRF-SIM Caveolae Fig4j</i>	<i>SD-SIM sCMOS Actin Ex. Fig15</i>	<i>SD-SIM sCMOS CCP Ex. Fig15</i>	<i>SD-SIM EMCCD Peroxisome Fig6e</i>	<i>SD-SIM EMCCD Tubulin Fig6e</i>	<i>SD-SIM EMCCD Lysosome Fig6e</i>
<i>Pixel size</i>	32.5	32.5	32.5	38.23	38.23	94	94	94
<i>Wave-length</i>	488	488	488	488	561	405	488	561
<i>Effective NA</i>	3	3	5	2.5	2.5	2.5	2.5	2.5
<i>Sparse iteration</i>	100	100	100	100	100	300	300	300
<i>Image fidelity</i>	150	30	1000	40	60	40	25	25
<i>t (z) axial continuity</i>	1	0.1	0.1	0.4	0.6	0.1	0.1	0.1
<i>Sparsity</i>	10	2.5	90	2	5	0.5	0.45	0.4
<i>Deblur times</i>	30	5	50	10	15	10	7	7
<i>Background</i>	No	Strong(HI)	No	Weak(HI)	Strong(HI)	No	No	Strong(LI)
<i>Deblur type</i>	LR	LR	LW	LR	LR	LR	LR	LR
<i>Over-sample</i>	No	No	Fourier	No	No	Spatial	Spatial	Spatial
<i>If 3D imaging</i>	No	No	No	No	No	No	No	No

We offered 8 datasets public online, and used the SD-SIM/Ex.Fig15 sCMOS/2colorActin.tif as an example in this manual.

The reconstructed video is then saved in the folder Data/SHReconstructed/****_**_**_2colorActin_Video/2colorActin_reconstructed.tif.

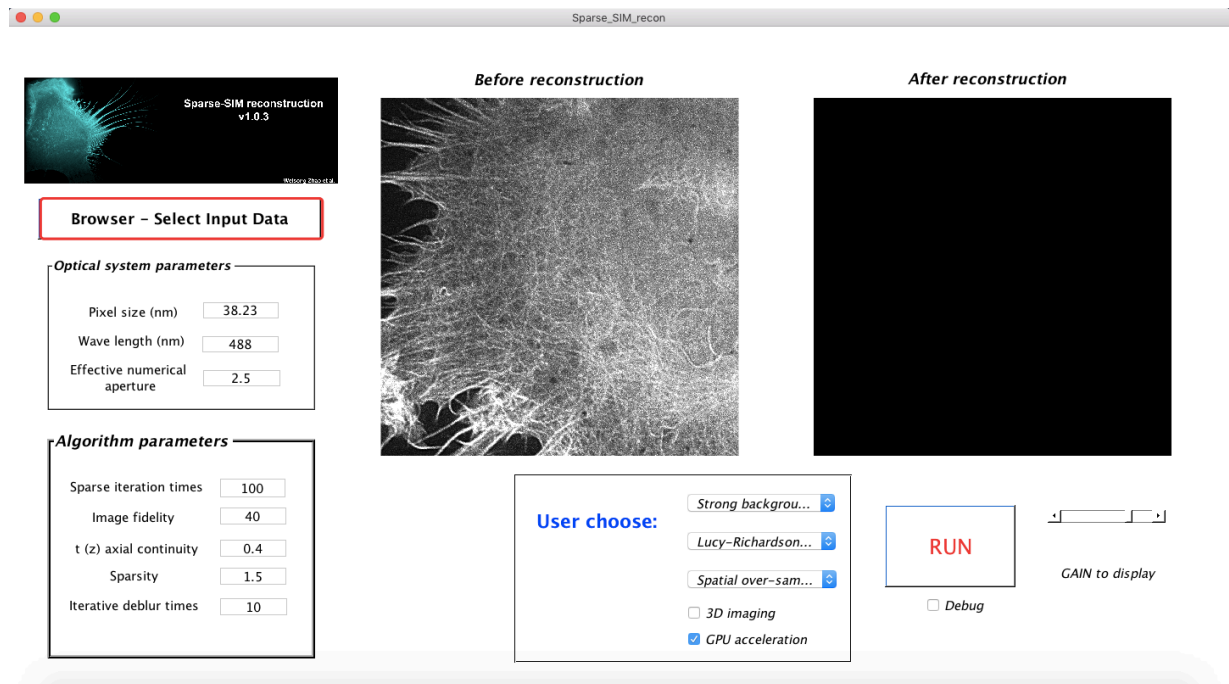
The ****_**_**_ is determined by the date you analyze the data eg. (2019_01_19_19: 2019 January 19th 19 o'clock).

Usually,

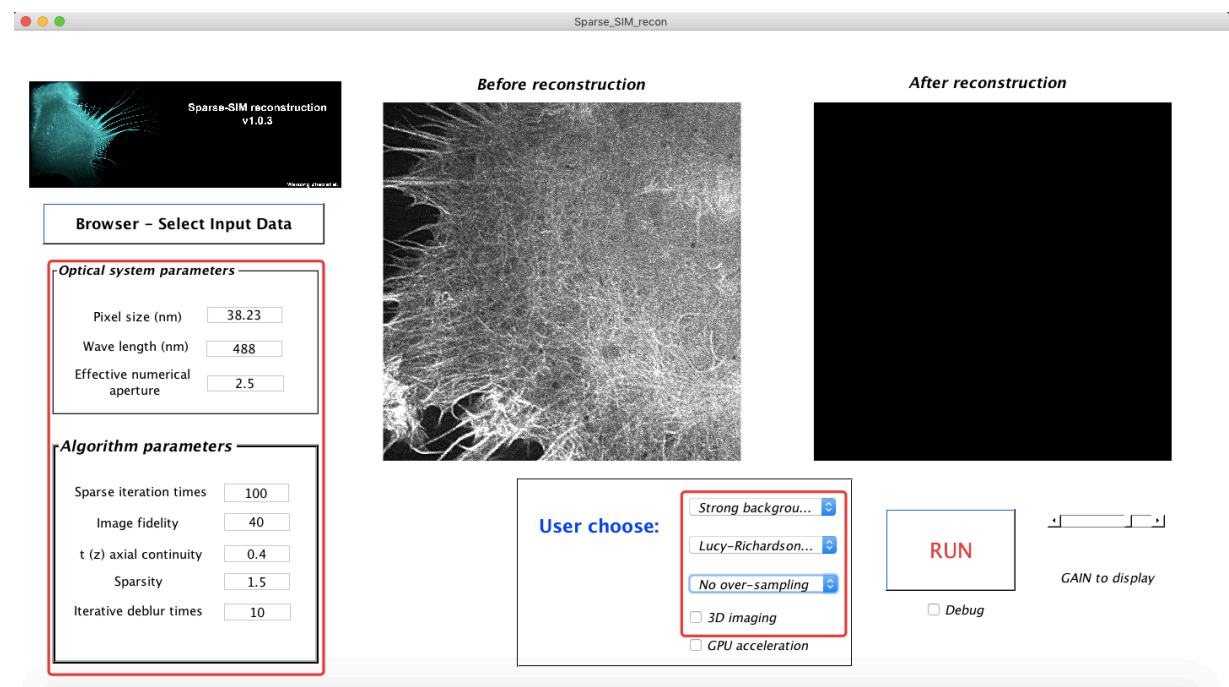
- **Iterative deconvolution times** for **LW**, and **LR** should be 5-15, and 30-50, respectively. **Sparse iteration times** should be 100. If the **Spatial over-sampling** operation is selected, the **Sparse iteration times** should be 200 or 300 and the **Image fidelity**, and **Sparsity** should be small enough to avoid 'zero artifacts'.
- **Sparsity** and **t (z)-axial continuity** should be smaller than one tenth of **Image fidelity**.
- **Image fidelity** should be larger than 10.

Example: SD-SIM/Ex.Fig15_sCMOS/2colorActin.tif

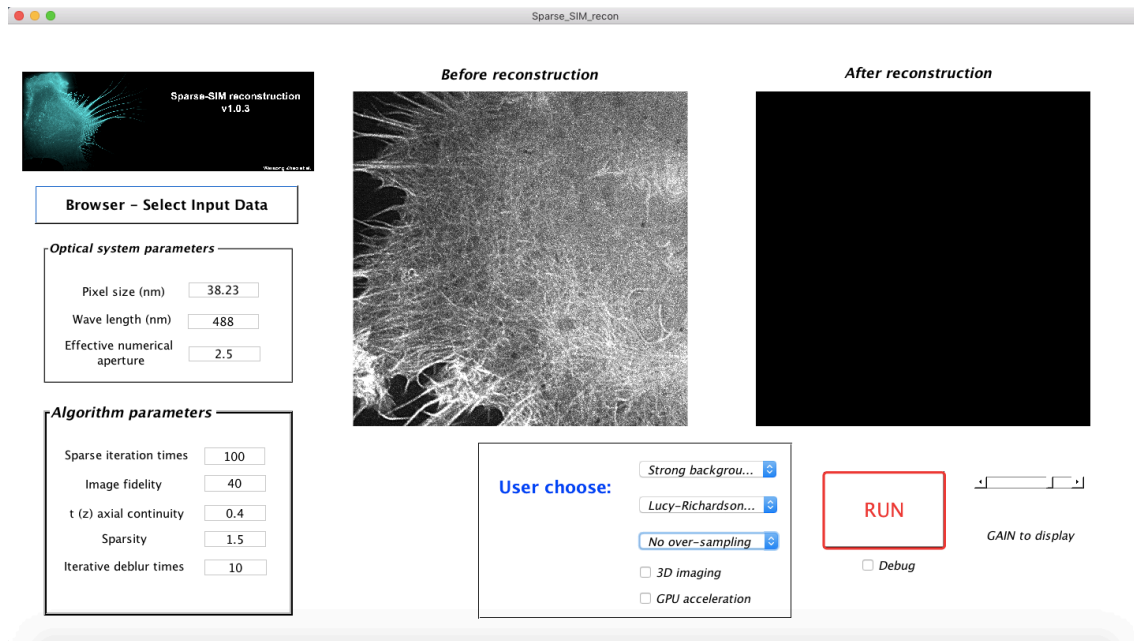
Step1. Choose the offered data.



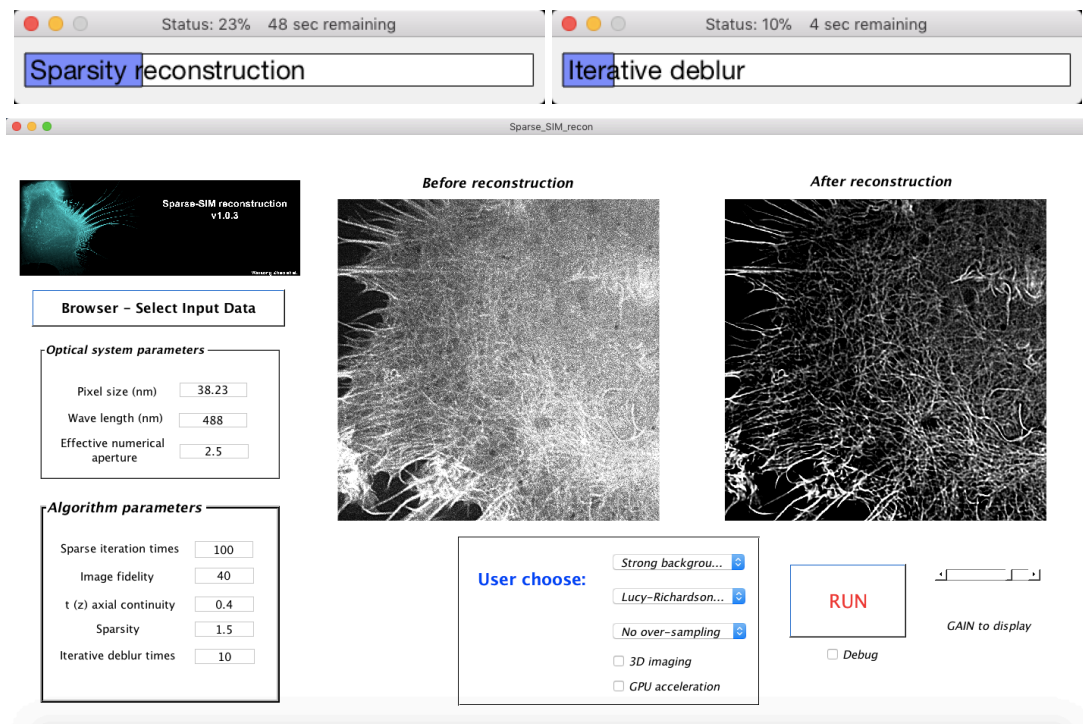
Step2. Set the parameters.



Step3. RUN



Step4. END



If the GPU memory is large enough to hold all variables, GPU processing is recommended for 60~100 fold improvement of execution speed. To achieve more effective GPU acceleration, the CUDA C code (.cu) for GPU acceleration is generated, and compiled to the binary form .mex. Specifically, the program that consumes the most computing resources are converted into CUDA mex:

back_diff_cuda.mexw64, and forward_diff_cuda.mexw64

These CUDA mex files are compiled in our local working stations. As they are related to the type of GPU (e.g. TITIAN RTX), the form of operating system (e.g. Windows 10), and the version of CUDA (e.g. 10.0), we did not apply these mex files in the released version of Sparse-SIM UI.

This software has been tested on:

- MATLAB R2017b on (Win 10: 128 GB and NVIDIA Titan Xp: 12GB; CUDA 9.1);
- MATLAB R2019b on (Win 10: 128 GB and NVIDIA Titan RTX: 24GB; CUDA 10.0);
- MATLAB R2019b on (Win 10: 16GB and NVIDIA GTX1050Ti: 4GB, CUDA 10.2);
- MATLAB R2015b on (CentOS 7: 64GB and Tesla K40 :12GB, CUDA 9.0);
- MATLAB R2018b on (Ubuntu 18.04: 16GB and NVIDIA TITAN Xp: 12GB, CUDA 10.1);
- MATLAB R2017b on (MacOS 10: 8GB without GPU).

/src_unix is the source code for Unix-like systems (including MacOS).

/src_win is the source code for Windows systems.

<https://github.com/WeisongZhao/Sparse-SIM/releases/download/v1.0.3/DATA.zip> holds all the provided example data.

<https://github.com/WeisongZhao/Sparse-SIM/releases/tag/v1.0.3> holds the binary executable files for MacOS, and Win10 systems.

<https://doi.org/10.5281/zenodo.5079744> holds several template datasets for parameter tuning

License:

This software and corresponding methods can only be used for **non-commercial use**, and they are under Open Data Commons Open Database License v1.0.