User Guide

Ruizhi Cao

June 22, 2018

This is an open-source code* for the reconstruction of the structured illumination microscopy (SIM). The inverse matrix based phase estimation algorithm is the default algorithm to estimate the phases of the illumination pattern.

1. Reading files

To start, replace the `filepath, filename, fileformat` with your own file's path, name and format in the main function "SIM_main_v2.m". Please add the "code" folder to the search path of your MATLAB. This prototype code requires square images. The images should be zero-padded if the acquired image is not square.

Please make sure that the parameters applied in the code match your system's parameters (numerical aperture, fluorescence emission wavelength, pixel size, etc.).

2. Parameter estimation and visualization

The modulation frequency vector $\mathbf{k}_0$ is estimated via the phase-only correlation method. A high-pass filter `fmask` is introduced to exclude the low frequency information from the peak searching algorithm. The cutoff frequency of the high-pass filter is k_c`*mask_factor`, where k_c is the cutoff frequency of the object transfer function (OTF).

By default, the correlation results using the phase-only correlation method are displayed after the phases are corrected. If you want to see the results using the default phases $\left(0, \frac{2\pi}{3}, \frac{4\pi}{3}\right)$, please set `show_initial_result_flag` to 1.

An example of the phase-only correlation result is shown in Fig. 1.

The recommended value of `mask_factor` is 0.8 for TIRF-SIM and 0.6 for SIM. If the peak is inside the dashed green circle, please decrease the value of `mask_factor`.

There are three options using different phase estimation methods in the source code:

1) The inverse matrix based phase estimation method (default)

2) The cross-correlation based iteration phase optimization method [1]
   Uncomment `%inv_phase=-cc_phase;` to use this method in the phase optimization. Please download the minFunc toolbox if you want to use this method. minFunc is available here [2]:
   http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html

* Support for MATLAB 2015a and later version. The main function uses the image processing toolbox in MATLAB. Other functions in this project do not rely on the toolbox.

And you can also use the built-in function `fminsearch` for your own implementation.

3) Single-step phase estimation method based on auto-correlation [3]
Uncomment `%inv_phase=auto_phase;` to use this method in the phase optimization. This algorithm provides a reliable phase estimation result in most cases and it is also the fastest algorithm among the above three.

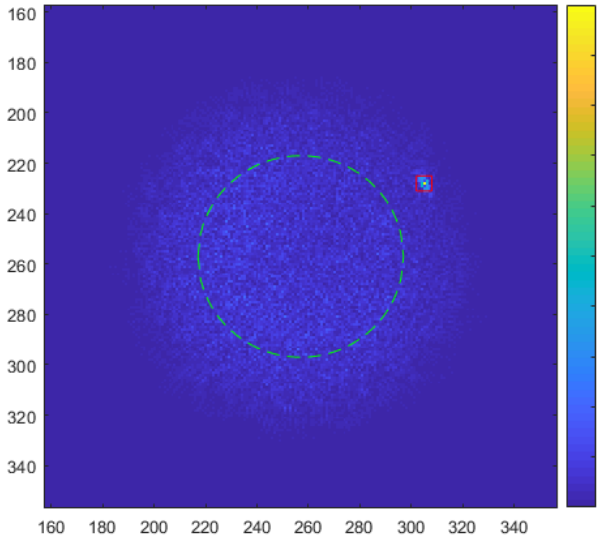

Fig. 1. The phase-only correlation result of a certain data set. The green dashed circle denotes the cutoff frequency of the high-pass filter utilized in the code. The component within the dashed circle is excluded when searching the peak. The red square points out the peak that the algorithm finds.

3. Save the result

If you want to save the reconstruct result and the efficient widefield result, please set `save_flag` to 1. By default, the results are normalized and saved as an 8-bit image associated with the built-in "`hot`" colormap. The file format is the same as the input format. Refer to the documents of the built-in function "`colormap`" to change the colormap associated with the images.

The results are normalized by the maximum in the corresponding image. Figure 2 shows an example of the result.

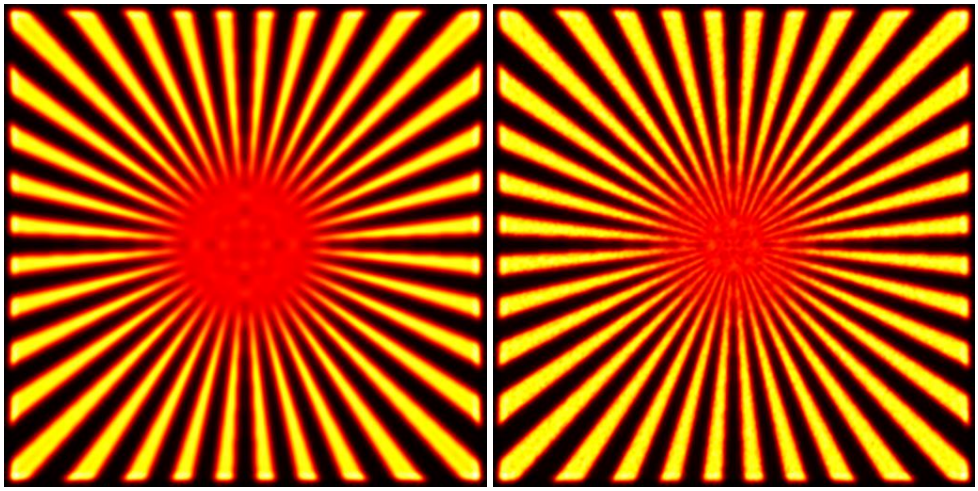

Fig. 2. The widefield (left) and the SIM result (right) using the inverse matrix based phase estimation method. The data are available in the "data" folder.

4. Functions and variables

1) General functions:

[shift_im] = exact_shift(im,relative_pixel,size_flag);

This function is similar to the built-in function "circshift" while it supports sub-pixel shifting. The first input argument is the image that you want to shift. The second input argument tells the program how to shift the input image. If size_flag=1, then the output and the input matrixes is of the same size. Otherwise, the image is zero-padded until the length of each dimension is odd.

As the image is zero-padded before it is shifted in this function, part of the information might be lost in the process. Please make sure that you do not suffer from information loss when you use the function. An example of the information loss is shown in Fig. 3.

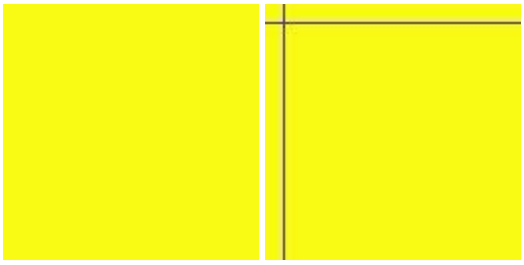

Fig. 3. The input image (left) and the output image (right) when using the function exact_shift. The black line in the right is a resultant of the zero-padding. The input 128*128 matrix is shifted by [10, 10] in this case.

[noiseimage] = quasi_wnr(OTF,noiseimage,noise_factor);

A low-pass filter that is similar to the wiener filter.

2) Special functions:

i. Modulation frequency vector estimation

[shiftvalue,background]
=frequency_est_tirf_v2(ft_im,suppress_noise_factor,fmask,if_show_factor,cutoff);

This function is based on phase-only correlation. It finds the peak outside the user defined high-pass filter.

[precise_shift,test_ini_phase]=precise_frequency_tirf(noiseimagef,shiftvalue,search_range);

Search the modulation frequency vector in the local determined by the above function. This function applies Fourier transform in order to obtain a sub-pixel modulation frequency vector.

ii. Cross-correlation based iterative phase optimization (See ref. 1 for details)

[ans_phase] = crosscorrelation_phase_est_SIM(noiseimagef,precise_shift,sigma,OTF);

The iterative algorithm for the phase estimation. Refer to the function and ref. 1 for detail.

[cost] = cost_func_CC(tensor_d,cal_phase);

Cost function in the iteration. See Eq. 10 of ref. 1.

iii. Single-step phase estimation method based on auto-correlation
This is not a standalone function but the majority of the function can be found in the code of the main function behind `%% auto-correlation based algorithm`

iv. Inverse matrix based phase estimation algorithm

[my_phase] = separation_matrix_correction_v2(noiseimagef,precise_shift,OTF);

[my_phase] = separation_matrix_correction_v3(noiseimagef,precise_shift,OTF);

This function calculate the phases of the illumination pattern by solving equations in this form: $\psi_{peak} = f\left(\varphi_1^{ill}, \varphi_2^{ill}, \varphi_3^{ill}\right)$, where φ_i^{ill} is the actual phases to be estimated and ψ_{peak} differs when using different inverse matrix. Version 3 is faster but Version 2 is easier to understand.

solve_trigonometric_linear_equation_var3

solve_trigonometric_linear_equation_var2

Functions aim to solve the above equation sets.

3) Variables
The following table lists some important variables and the content that they stored.

variable name	content
noiseimage	The raw captured image with pre-processing step (eg. filtering)
noiseimagef	The 2D Fourier transform of the data
separated_FT	Reweighed/Separated frequency component
ft_true	Different frequency components that have been shifted to their correct position
inv_phase	The estimated phases using the inverse matrix method
cc_phase	The estimated phases using the cross-correlation method
auto_phase	The estimated phases using the auto-correlation method
shiftvalue	The estimated modulation frequency vector
precise_shift	The estimated modulation frequency vector with a sub-pixel precision
modulation_depth	modulation depth for each pattern orientation

Reference

[1] K. Wicker, O. Mandula, G. Best, R. Fiolka, and R. Heintzmann, "Phase optimisation for structured illumination microscopy," Opt. Express **21**, 2032-2049 (2013).

[2] M. Schmidt. minFunc unconstrained differentiable multivariate optimization in Matlab. http://www.cs.ubc.ca/~schmidtm/Software/minFunc.html, 2005.

[3] K. Wicker, "Non-iterative determination of pattern phase in structured illumination microscopy using auto-correlations in Fourier space," Opt. Express **21**, 24692-24701 (2013).