OpenSIM: Documentation

Amit Lal*, Chunyan Shan†, and Peng Xi‡

January 23, 2016
version 1.0

Contents

*Department of Biomedical Engineering, College of Engineering, Peking University, Beijing 100871, China.
†School of Life Sciences, Peking University, Beijing 100871, China.
‡Department of Biomedical Engineering, College of Engineering, Peking University, Beijing 100871, China. E-mail: xipeng@pku.edu.cn

1 File Description

1. The OpenSIM matlab codes may be downloaded from:
   https://github.com/LanMai/OpenSIM
   The site also accompanies OpenSIM mannual providing necessary instructions for using these codes.

2. OpenSIM codes are provided in four sets (folders) – SIMbasic, SIMexpt, TIRFbasic and TIRFexpt, Fig. 1. Files contained in SIMbasic are suited to perform SIM reconstruction on simulated raw SIM images, while those contained in SIMexpt are suited for performing SIM reconstruction on experimental raw SIM images. Files contained in TIRFbasic and TIRFexpt are TIRF-SIM counterparts of SIMbasic and SIMexpt, respectively.

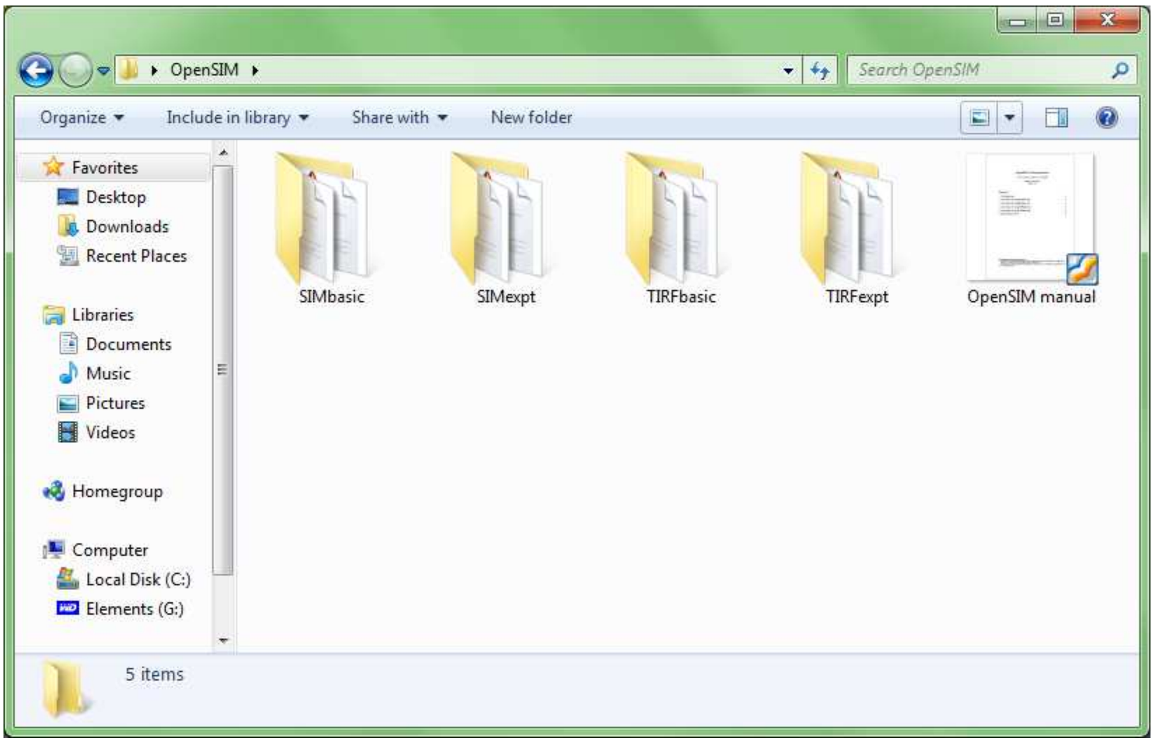


Figure 1: OpenSIM code sets: SIMbasic, SIMexpt, TIRFbasic and TIRFexpt.

3. SIMbasic contains 23 Matlab files, listed in Table 1, and one synthetic image file `testpat.tiff`, see Fig. 2. [`testpat.tiff` is identical with `testpat.1k.tiff` mentioned in manuscript (section VI-A), but is renamed to avoid two dots '.' in its name.]

4. SIMexpt contains 23 Matlab files, plus two '.tif' files: `OTF.tif` and `sim01z4.tif`. `OTF.tif` is the experimentally determined system OTF, while `sim01z4.tif` is experimental raw SIM data file, see Fig. 3. Nomenclature of Matlab files contained in SIMexpt are identical with that listed in Table 1, with the following exceptions:

   (a) `SIMimagesF.m` and `PsfOtf.m` are absent,

   (b) Contains two new files

   - `OTFpost.m`: flattens out the rippling higher frequencies outside OTF support, and

- BgNormalization0.m: Carries out intensity normalization of raw SIM images

(c) Files SIMo.m and PCMfilteringF.m in SIMexpt differ from their counterparts in SIMbasic, a heuristically designed notch filter (Eq. 25 of main manuscript) is incorporated to suppress residual peaks resulting from inaccuracies in background fluorescence removal.

5. TIRFbasic contains 21 Matlab files, and one synthetic image file testpat.tiff. Nomenclature of Matlab files contained in TIRFbasic are identical with that listed in Table 1, with the following exceptions:

   (a) ApproxFreqDuplex.m, IlluminationFreqF.m, IlluminationPhaseF.m, PatternPhaseOpt.m and PhaseKai2opt.m are absent,

   (b) Contains following new files

   - Kai2Opt.m: Optimizes the relative phases ϕ_2' and ϕ_3',
   - IlluminationFreqTIRF.m: Estimates illumination frequency vector $\mathbf{p}_\theta$, and
   - Ifreq2opt.m: Used for subpixel optimization of $\mathbf{p}_\theta$ within IlluminationFreqTIRF.m.

6. TIRFexpt contains 21 Matlab files, plus two '.tif' files: OTF.tif and sim01z4.tif. OTF.tif is the experimentally determined system OTF, while sim01z4.tif is experimental raw SIM data file, see Fig. 3. Nomenclature of Matlab files contained in TIRFexpt are identical with that listed in Table 1, with the following exceptions:

   (a) SIMimagesF.m, PsfOtf.m, ApproxFreqDuplex.m, IlluminationFreqF.m, IlluminationPhaseF.m, PatternPhaseOpt.m and PhaseKai2opt.m are absent,

   (b) Contains following new files

   - OTFpost.m: flattens out the rippling higher frequencies outside OTF support, and
   - BgNormalization0.m: Carries out intensity normalization of raw SIM images
   - Kai2Opt.m: Optimizes the relative phases ϕ_2' and ϕ_3',
   - IlluminationFreqTIRF.m: Estimates illumination frequency vector $\mathbf{p}_\theta$, and
   - Ifreq2opt.m: Used for subpixel optimization of $\mathbf{p}_\theta$ within IlluminationFreqTIRF.m.

   (c) Files SIMo.m and PCMfilteringF.m in TIRFexpt differ from their counterparts in TIRFbasic, a heuristically designed notch filter (Eq. 25 of main manuscript) is incorporated to suppress residual peaks resulting from inaccuracies in background fluorescence removal.

7. SIMo.m is the main script file in all SIMbasic, SIMexpt, TIRFbasic and TIRFexpt. Rest of the *.m files in each folder, are *function* files that are used by SIMo.m for its operation.

2 Instructions for using SIMbasic files

1. Open the file SIMo.m contained in folder SIMbasic using Matlab. First few lines of SIMo.m are shown in Fig. 4.

2. Referring to Fig. 4, the parameter values that a user may choose to fiddle with are as follows:

   (a) scale (on line 12): altering this parameter changes width of PSF (and hence, corresponding OTF $\tilde{H}(\mathbf{k})$ support).

Table 1: SIMbasic files

	File name	Function
1	`SIMo.m`	Main script file for simulated raw SIM images and then carrying out SIM reconstruction
2	`PsfOtf.m`	Generates PSF, and corresponding OTF $\tilde{H}(\mathbf{k})$, using Bessel function
3	`SIMimagesF.m`	Generates simulated raw SIM images $D_{\theta,\phi}(\mathbf{r})$ [$\theta = \theta_1, \theta_2, \theta_3$ and $\phi = \phi_1, \phi_2, \phi_3$]
4	`PCMseparateF.m`	Performs steps 3-10 of SIM-RA
5	`IlluminationFreqF.m`	Estimates illumination frequency vector $\mathbf{p}_\theta$ (step 4 of SIM-RA)
6	`ApproxFreqDuplex.m`	Provides initial guess for $\mathbf{p}_\theta$ determination
7	`PhaseKai2opt.m`	Performs the iterative optimization for $\mathbf{p}_\theta$ determination
8	`OTFedgeF.m`	Computes OTF cut-off frequency
9	`IlluminationPhaseF.m`	Estimates illumination phase shift ϕ (step 8 of SIM-RA)
10	`PatternPhaseOpt.m`	Performs the iterative optimization for ϕ determination
11	`SeparatedComponents2D.m`	Computes noisy estimates of $\tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k})$, $\tilde{S}(\mathbf{k} - \mathbf{p}_\theta)\tilde{H}(\mathbf{k})$ and $\tilde{S}(\mathbf{k} + \mathbf{p}_\theta)\tilde{H}(\mathbf{k})$ (step 10 of SIM-RA)
12	`OBJpowerPara.m`	Determines object power parameters $\mathcal{A}$ and α (step 12 of SIM-RA)
13	`OBJparaOpt.m`	Performs the iterative optimization for determination of $\mathcal{A}$ and α
14	`PCMfilteringF.m`	Performs steps 14-17 of SIM-RA
15	`ModulationFactor.m`	Determines modulation factor m (step 14 of SIM-RA)
16	`WoFilterCenter.m`	Wiener filters noisy $\tilde{S}(\mathbf{k})\tilde{H}(\mathbf{k})$ to obtain $\tilde{S}_u(\mathbf{k})$ (step 15 of SIM-RA)
17	`WoFilterSideLobe.m`	Wiener filters noisy $\tilde{S}(\mathbf{k} - \mathbf{p}_\theta)\tilde{H}(\mathbf{k})$ and $\tilde{S}(\mathbf{k} + \mathbf{p}_\theta)\tilde{H}(\mathbf{k})$ to obtain $\tilde{S}_u(\mathbf{k} - \mathbf{p}_\theta)$ and $\tilde{S}_u(\mathbf{k} - \mathbf{p}_\theta)$, respectively (step 15 of SIM-RA)
18	`OTFdoubling.m`	Embeds system OTF in doubled range of frequency; this allows accommodation of higher frequencies in reconstructed images
19	`MergingHeptaletsF.m`	Merges all nine frequency components using generalized Wiener Filter (step 19 of SIM-RA)
20	`TripletSNR0.m`	Computes average signal powers of $\tilde{S}_u(\mathbf{k})$, $\tilde{S}_s(\mathbf{k} - \mathbf{p}_\theta)$ and $\tilde{S}_s(\mathbf{k} + \mathbf{p}_\theta)$
21	`SIMplot.m`	Plots the reconstructed SIM images
22	`ApodizationFunction.m`	Masks the higher residual frequencies of $\tilde{D}_{\mathrm{SIM}}(\mathbf{k})$, i.e. computes $\tilde{D}_{\mathrm{SIM}}(\mathbf{k})\tilde{A}(\mathbf{k})$ [section VII-E]
23	`OTFmaskShifted.m`	Used in computation of apodization mask $\tilde{A}(\mathbf{k})$

(b) `testpat.tiff` in line 25 represents the synthetic image used for simulation to generate raw SIM images. If one wishes to use a different test object for simulation, then `testpat.tiff` may be replaced by the name of the corresponding test object filename.

(c) Note that original size of synthetic image `testpat.tiff` is 1024×1024 pixels, see Fig. 2. However, for simulation, only its central 512×512 pixels are selected (see Fig. 4 of manuscript). This selection is done in line 26 of `SIMo.m`. Thus, the values within brackets in line 26 may be altered if desired (specially, when test object is changed).

(d) Parameters corresponding to structured illumination pattern – illumination spatial frequency and modulation factor – are specified in lines 33 and 34, respectively.

(e) Additive Gaussian noise level in simulated raw SIM images may be set in line 35.

(f) Whether the diffraction limited image acquisition of specimen is to be *computationally* effected in spatial domain by convolving it with system PSF or in Fourier domain by multiplying FT of illuminated specimen with system OTF is decided by value of parameter `UsePSF` in line 36. For effecting it by the former method, set `UsePSF`=1 or else set `UsePSF`=0.

3. The default values of all the parameters described above is set so as to reproduce the results presented in Fig. 4 of manuscript. That is, by simply opening `SIMo.m` contained in folder SIMbasic and executing it directly in Matlab (by pressing function key 'F5') should produce the results presented in Fig. 4 of manuscript. Execution time is ≈1-2min. The reconstructed images are displayed in usual figure windows by Matlab and may be saved in any desired file format permitted by Matlab.

4. All Matlab files are generously commented to enable the user to follow the proceedings of the code with ease.

3 Instructions for using **SIMexpt** files

1. Open file `SIMo.m` contained in folder SIMexpt using Matlab. First few lines of `SIMo.m` are shown in Fig. 5.

2. Referring to Fig. 5, the parameter values that a user may choose to fiddle with are as follows:

   (a) The filename `OTF.tif` in line 6, which contains experimentally determined system OTF information.

   (b) Experimental raw SIM data file name `sim01z4.tif` (in line 10) may be replaced to carry out SIM reconstruction of another experimentally obtained raw SIM data.

   (c) Extracting 9 raw SIM images from experimental data file differs with acquisition/data storage system which accompanies the experimental set-up. Thus, it may be necessary to modify lines 11-29 for a different experimental data file.

   (d) The shape and size of structural element used for background fluorescence removal is set to 'disk' and 20, respectively (line 40). Both shape and size of the structural element may be varied, to obtain satisfactory background fluorescence removal.

3. The default values of all the parameters described above is set so as to reproduce the results presented in Fig. 6 of manuscript. That is, by simply opening `SIMo.m` contained in folder SIMexpt and executing it directly in Matlab (by pressing the function key 'F5')

should produce results presented in Fig. 6 of manuscript. Execution time is ≈1-2min. The reconstructed images are displayed in usual figure windows by Matlab and may be saved in any desired file format permitted by Matlab.

4. All Matlab files are generously commented to enable the user to follow the proceedings of the code with ease.

4 Instructions for using **TIRFbasic** files

1. Open the file `SIMo.m` contained in folder TIRFbasic using Matlab. First few lines of `SIMo.m` are shown in Fig. 6.

2. Referring to Fig. 6, the parameter values that a user may choose to fiddle with are essentially identical to those described in section 2 for file `SIMo.m` of `SIMbasic`. One noteworthy flexibility here is that the illumination spatial frequency in line 33 may now be greater than OTF cut-off frequency.

3. There are some additional parameters defining the tentative search region for illumination spatial frequency vector that needs to be set (lines 62 to 65). The code, searches for an estimate for *true* illumination spatial frequency vector in a circular region of pixel radius 20 with center defined by the parameters set here. For instance, for the first structured illumination orientation, the code searches for an estimate for *true* illumination spatial frequency vector in a circular region of pixel radius 20 with center $[\mathtt{k2o}\cos(\mathtt{thetaA}), \mathtt{k2o}\sin(\mathtt{thetaA})]$.

4. The default values of all the parameters described above is set so as to reproduce TIRF-SIM reconstruction image for the case when illumination spatial frequency is $\approx 1.2 * \mathrm{OTF}_{\text{cut-off}}$. Execution time is ≈1-2min. The reconstructed images are displayed in usual figure windows by Matlab and may be saved in any desired file format permitted by Matlab.

5. All Matlab files are generously commented to enable the user to follow the proceedings of the code with ease.

5 Instructions for using **TIRFexpt** files

1. Open the file `SIMo.m` contained in folder TIRFexpt using Matlab. First few lines of `SIMo.m` are shown in Fig. 7.

2. Referring to Fig. 7, the parameter values that a user may choose to fiddle with are essentially identical to those described in section 3 for file `SIMo.m` of `SIMexpt`. However, there are some additional parameters defining the tentative search region for illumination spatial frequency vector that needs to be set (lines 53 to 56). The code, searches for an estimate for *true* illumination spatial frequency vector in a circular region of pixel radius 20 with center defined by the parameters set here. For instance, for the first structured illumination orientation, the code searches for an estimate for *true* illumination spatial frequency vector in a circular region of pixel radius 20 with center $[\mathtt{k2o}\cos(\mathtt{thetaA}), \mathtt{k2o}\sin(\mathtt{thetaA})]$.

3. The default values of all the parameters described above is set so as to reproduce the results presented in Fig. 6 of manuscript. That is, by simply opening `SIMo.m` contained in folder SIMexpt and executing it directly in Matlab (by pressing the function key 'F5') should produce results presented in Fig. 6 of manuscript. Execution time is ≈1-2min. The reconstructed images are displayed in usual figure windows by Matlab and may be saved in any desired file format permitted by Matlab.

4. All Matlab files are generously commented to enable the user to follow the proceedings of the code with ease.

6 System Requirements

Successful operation of OpenSIM codes are tested on Windows 7 operating system with Matlab (R2010a) accompanied with image processing toolbox.

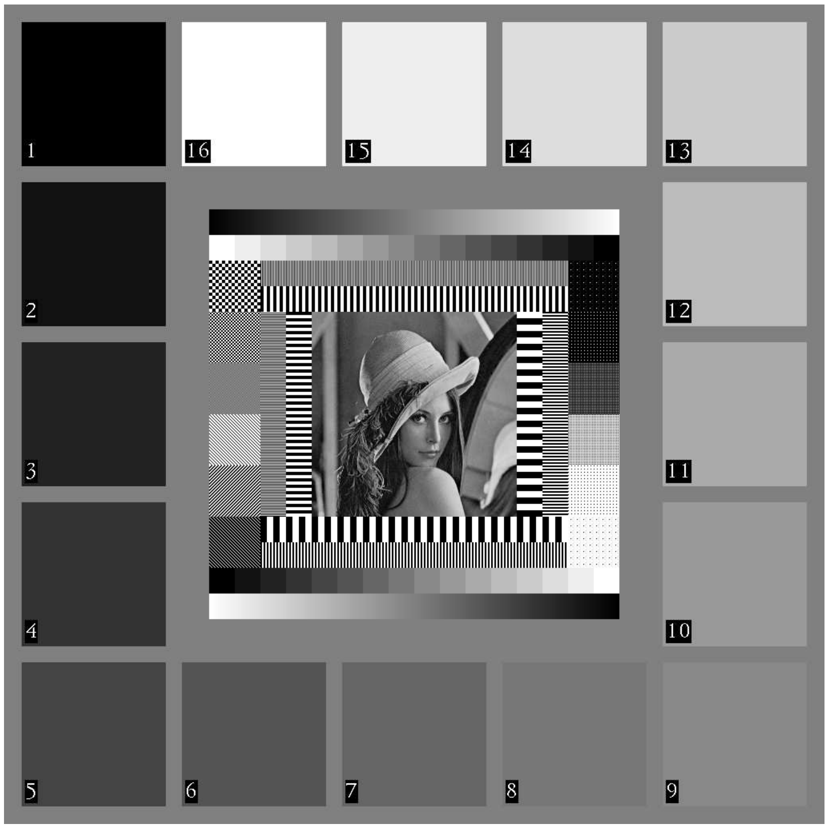


Figure 2: Synthetic image file `testpat.tiff`.

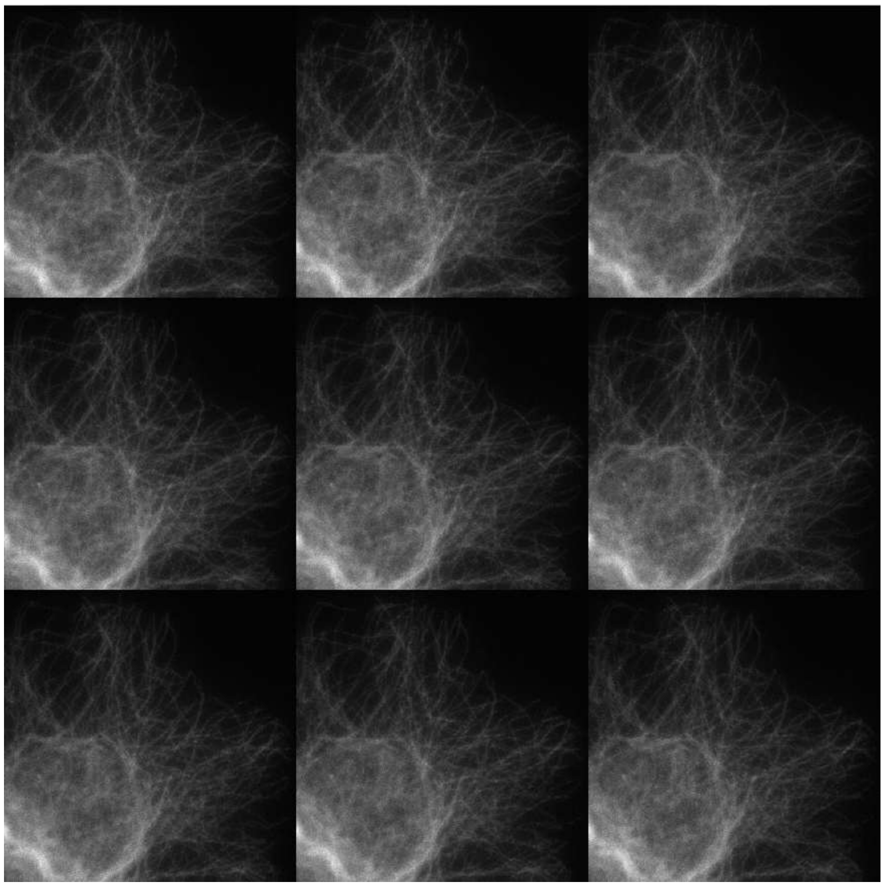


Figure 3: Experimental raw SIM data file `sim01z4.tif`.

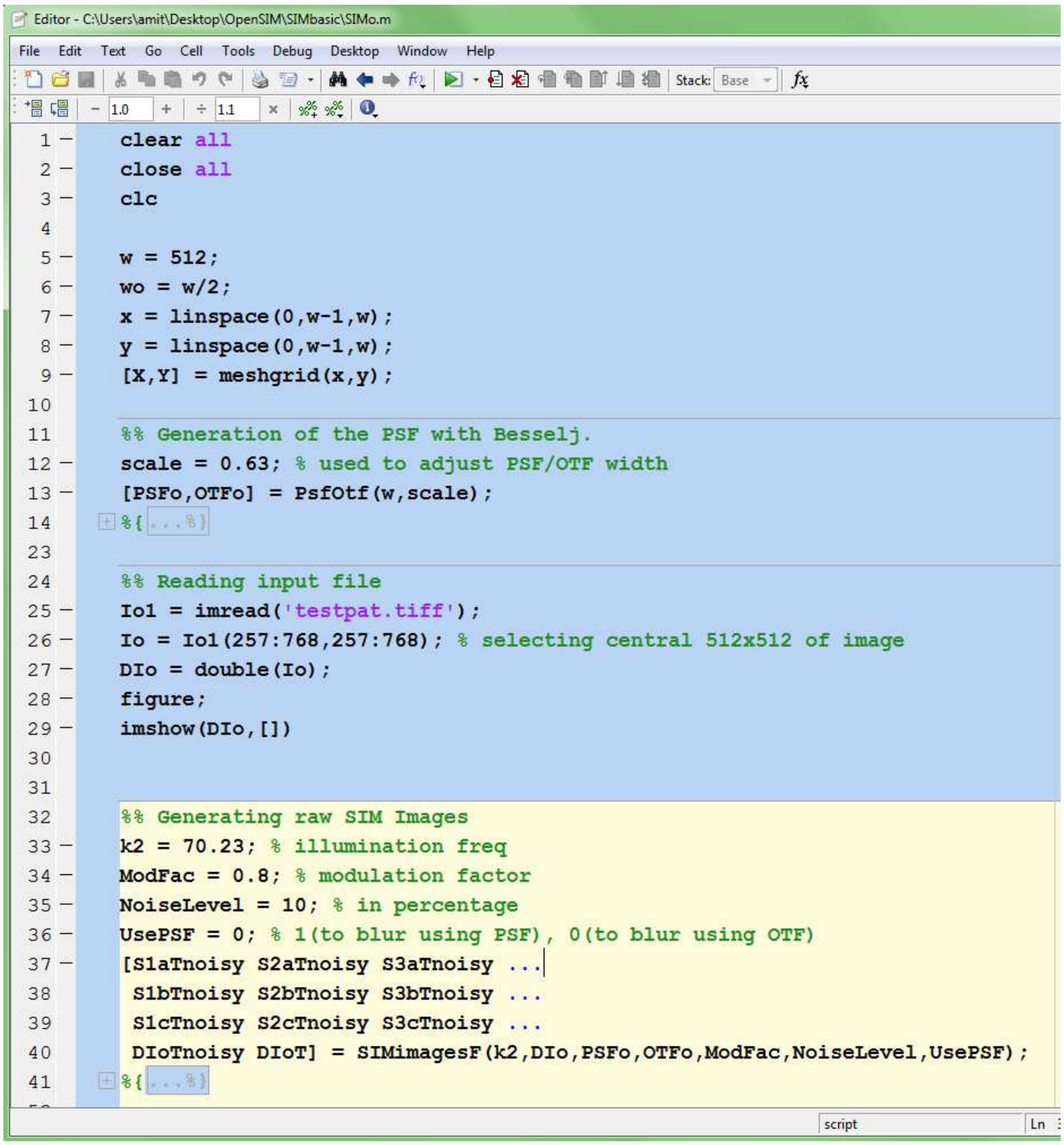
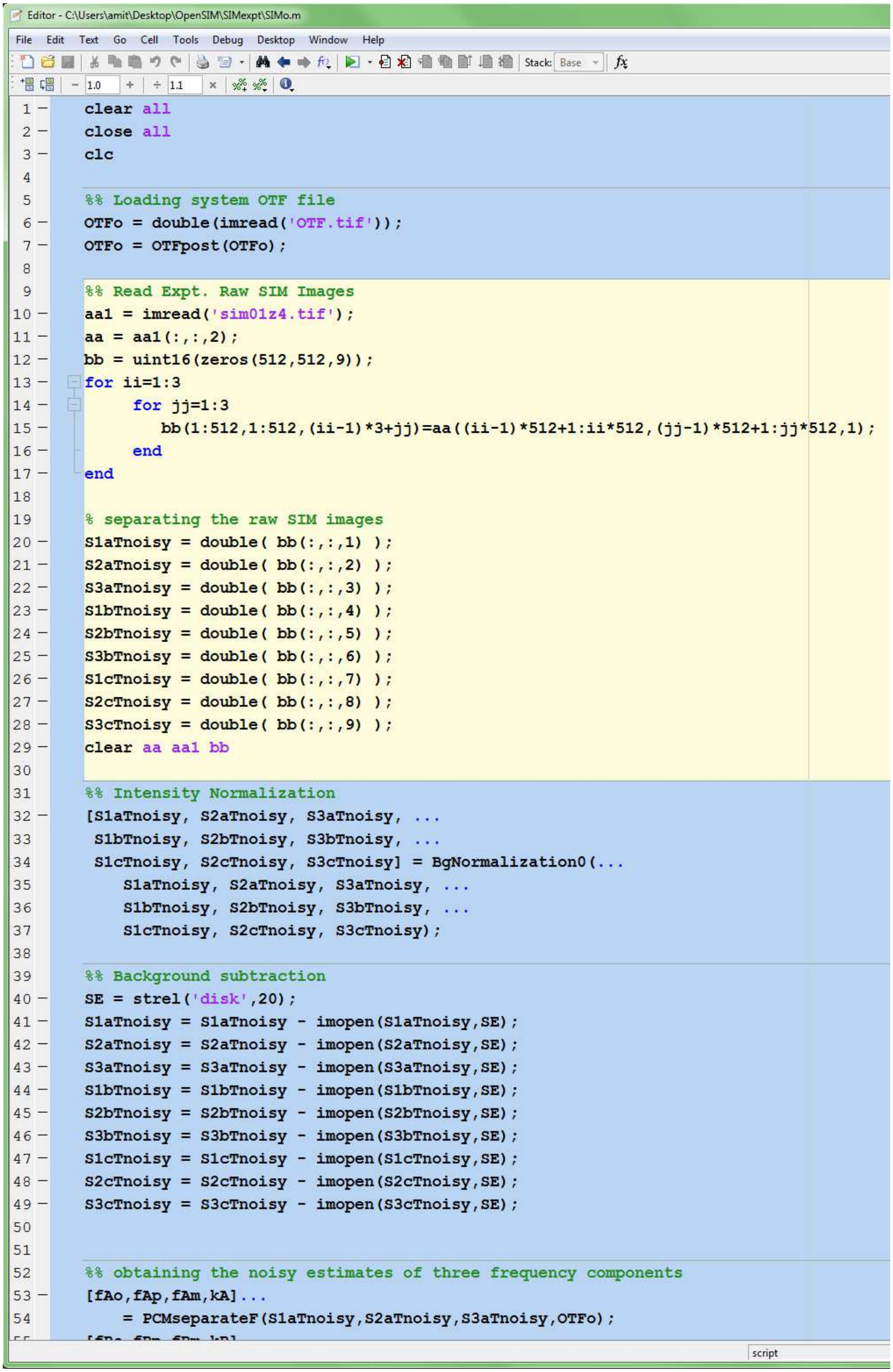


Figure 4: First few lines of file `SIMo.m` in SIMbasic.



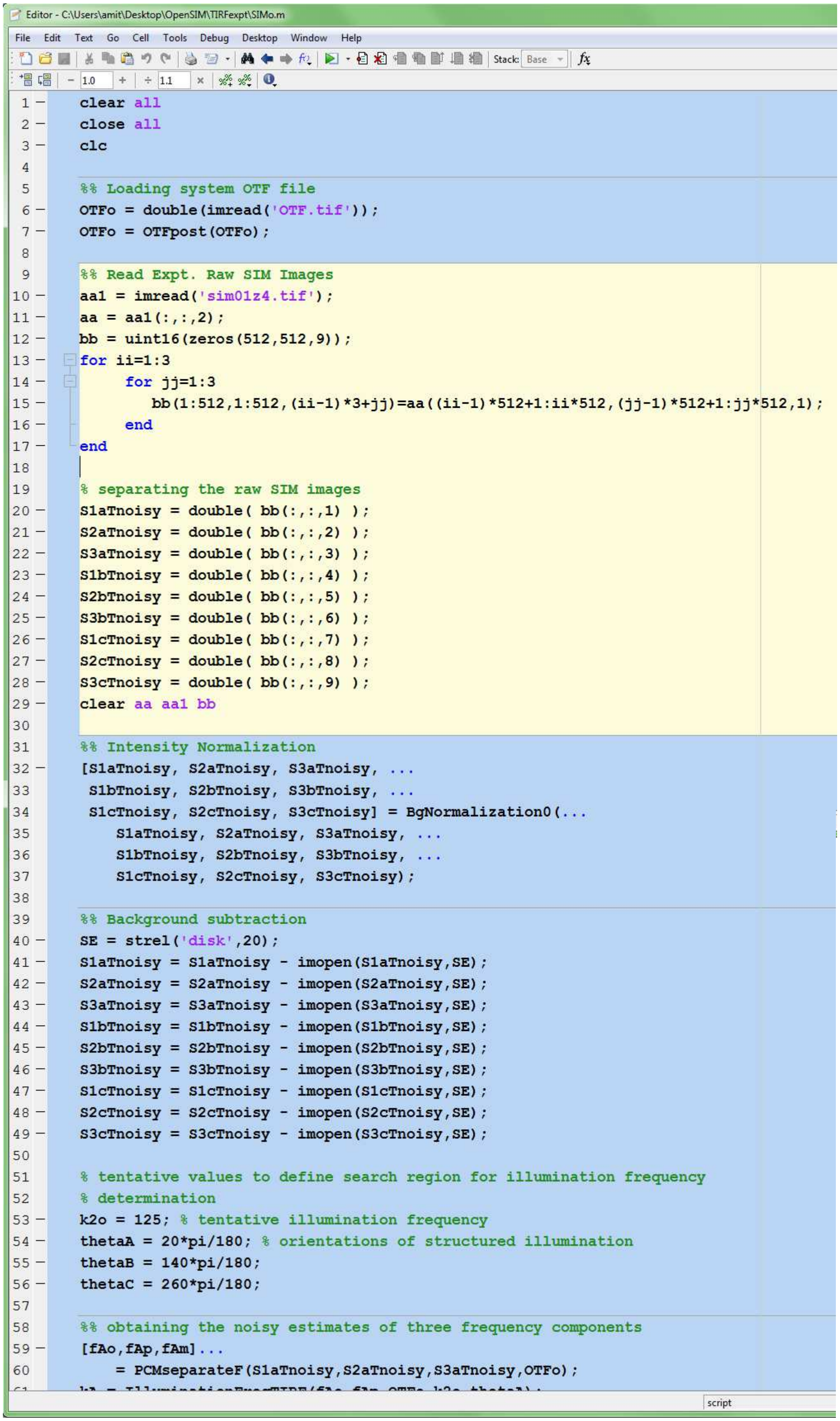

```
clear all
close all
clc

%% Loading system OTF file
OTFo = double(imread('OTF.tif'));
OTFo = OTFpost(OTFo);

%% Read Expt. Raw SIM Images
aa1 = imread('sim01z4.tif');
aa = aa1(:,:,2);
bb = uint16(zeros(512,512,9));
for ii=1:3
    for jj=1:3
        bb(1:512,1:512,(ii-1)*3+jj)=aa((ii-1)*512+1:ii*512,(jj-1)*512+1:jj*512,1);
    end
end

% separating the raw SIM images
S1aTnoisy = double( bb(:,:,1) );
S2aTnoisy = double( bb(:,:,2) );
S3aTnoisy = double( bb(:,:,3) );
S1bTnoisy = double( bb(:,:,4) );
S2bTnoisy = double( bb(:,:,5) );
S3bTnoisy = double( bb(:,:,6) );
S1cTnoisy = double( bb(:,:,7) );
S2cTnoisy = double( bb(:,:,8) );
S3cTnoisy = double( bb(:,:,9) );
clear aa aa1 bb

%% Intensity Normalization
[S1aTnoisy, S2aTnoisy, S3aTnoisy, ...
 S1bTnoisy, S2bTnoisy, S3bTnoisy, ...
 S1cTnoisy, S2cTnoisy, S3cTnoisy] = BgNormalization0(...
    S1aTnoisy, S2aTnoisy, S3aTnoisy, ...
    S1bTnoisy, S2bTnoisy, S3bTnoisy, ...
    S1cTnoisy, S2cTnoisy, S3cTnoisy);

%% Background subtraction
SE = strel('disk',20);
S1aTnoisy = S1aTnoisy - imopen(S1aTnoisy,SE);
S2aTnoisy = S2aTnoisy - imopen(S2aTnoisy,SE);
S3aTnoisy = S3aTnoisy - imopen(S3aTnoisy,SE);
S1bTnoisy = S1bTnoisy - imopen(S1bTnoisy,SE);
S2bTnoisy = S2bTnoisy - imopen(S2bTnoisy,SE);
S3bTnoisy = S3bTnoisy - imopen(S3bTnoisy,SE);
S1cTnoisy = S1cTnoisy - imopen(S1cTnoisy,SE);
S2cTnoisy = S2cTnoisy - imopen(S2cTnoisy,SE);
S3cTnoisy = S3cTnoisy - imopen(S3cTnoisy,SE);


%% obtaining the noisy estimates of three frequency components
[fAo,fAp,fAm,kA]...
    = PCMseparateF(S1aTnoisy,S2aTnoisy,S3aTnoisy,OTFo);
```

Figure 5: First few lines of file `SIMo.m` in SIMexpt.

Editor - C:\Users\amit\Desktop\OpenSIM\TIRFbasic\SIMo.m

```matlab
clear all
close all
clc

w = 512;
wo = w/2;
x = linspace(0,w-1,w);
y = linspace(0,w-1,w);
[X,Y] = meshgrid(x,y);

%% Generation of the PSF with Besselj.
scale = 0.63; % used to adjust PSF/OTF width
[PSFo,OTFo] = PsfOtf(w,scale);
%{...%}

%% Reading input file
Io1 = imread('testpat.tiff');
Io = Io1(257:768,257:768); % selecting central 512x512 of image
DIo = double(Io);
figure;
imshow(DIo,[])


%% Generating raw SIM Images
k2 = 120.23; % illumination freq
ModFac = 0.8; % modulation factor
NoiseLevel = 10; % in percentage
UsePSF = 0; % 1(to blur using PSF), 0(to blur using OTF)
[S1aTnoisy S2aTnoisy S3aTnoisy ...
 S1bTnoisy S2bTnoisy S3bTnoisy ...
 S1cTnoisy S2cTnoisy S3cTnoisy ...
 DIoTnoisy DIoT] = SIMimagesF(k2,DIo,PSFo,OTFo,ModFac,NoiseLevel,UsePSF);
%{...%}

% tentative values to define search region for illumination frequency
% determination
k2o = 125; % tentative illumination frequency
thetaA = 0*pi/3; % orientations of structured illumination
thetaB = 1*pi/3;
thetaC = 2*pi/3;

%% obtaining the noisy estimates of three frequency components
[fAo,fAp,fAm]...
    = PCMseparateF(S1aTnoisy,S2aTnoisy,S3aTnoisy,OTFo);
kA = IlluminationFreqTIRF(fAo,fAp,OTFo,k2o,thetaA);
[fBo,fBp,fBm]...
    = PCMseparateF(S1bTnoisy,S2bTnoisy,S3bTnoisy,OTFo);
kB = IlluminationFreqTIRF(fBo,fBp,OTFo,k2o,thetaB);
```

Figure 6: First few lines of file SIMo.m in TIRFbasic.



```
clear all
close all
clc

%% Loading system OTF file
OTFo = double(imread('OTF.tif'));
OTFo = OTFpost(OTFo);

%% Read Expt. Raw SIM Images
aa1 = imread('sim01z4.tif');
aa = aa1(:,:,2);
bb = uint16(zeros(512,512,9));
for ii=1:3
    for jj=1:3
        bb(1:512,1:512,(ii-1)*3+jj)=aa((ii-1)*512+1:ii*512,(jj-1)*512+1:jj*512,1);
    end
end

% separating the raw SIM images
S1aTnoisy = double( bb(:,:,1) );
S2aTnoisy = double( bb(:,:,2) );
S3aTnoisy = double( bb(:,:,3) );
S1bTnoisy = double( bb(:,:,4) );
S2bTnoisy = double( bb(:,:,5) );
S3bTnoisy = double( bb(:,:,6) );
S1cTnoisy = double( bb(:,:,7) );
S2cTnoisy = double( bb(:,:,8) );
S3cTnoisy = double( bb(:,:,9) );
clear aa aa1 bb

%% Intensity Normalization
[S1aTnoisy, S2aTnoisy, S3aTnoisy, ...
 S1bTnoisy, S2bTnoisy, S3bTnoisy, ...
 S1cTnoisy, S2cTnoisy, S3cTnoisy] = BgNormalization0(...
    S1aTnoisy, S2aTnoisy, S3aTnoisy, ...
    S1bTnoisy, S2bTnoisy, S3bTnoisy, ...
    S1cTnoisy, S2cTnoisy, S3cTnoisy);

%% Background subtraction
SE = strel('disk',20);
S1aTnoisy = S1aTnoisy - imopen(S1aTnoisy,SE);
S2aTnoisy = S2aTnoisy - imopen(S2aTnoisy,SE);
S3aTnoisy = S3aTnoisy - imopen(S3aTnoisy,SE);
S1bTnoisy = S1bTnoisy - imopen(S1bTnoisy,SE);
S2bTnoisy = S2bTnoisy - imopen(S2bTnoisy,SE);
S3bTnoisy = S3bTnoisy - imopen(S3bTnoisy,SE);
S1cTnoisy = S1cTnoisy - imopen(S1cTnoisy,SE);
S2cTnoisy = S2cTnoisy - imopen(S2cTnoisy,SE);
S3cTnoisy = S3cTnoisy - imopen(S3cTnoisy,SE);

% tentative values to define search region for illumination frequency
% determination
k2o = 125; % tentative illumination frequency
thetaA = 20*pi/180; % orientations of structured illumination
thetaB = 140*pi/180;
thetaC = 260*pi/180;

%% obtaining the noisy estimates of three frequency components
[fAo,fAp,fAm]...
    = PCMseparateF(S1aTnoisy,S2aTnoisy,S3aTnoisy,OTFo);
```

Figure 7: First few lines of file SIMo.m in TIRFexpt.