

User's Guide for using MC-ISM software

Wei Ren^{1,2}, Meiling Guan^{1,2}, Qianxi Liang^{1,2}, Peng Xi^{1,2}

¹ Department of Biomedical Engineering, College of Future Technology, Peking University, Beijing 100871, China

² National Biomedical Imaging Center, Peking University, Beijing 100871, China

This is a comprehensive user's guide for the MC-ISM, including MC-ISM reconstruction program (chapter 1- chapter 4), codes for hardware testing (chapter 5), and hardware implementation and control (chapter 6). All content is open source on <https://github.com/Chauncey-Leung/MC-ISM>. MC-ISM reconstruction program was implemented as a friendly Graphical User Interface (GUI) of MATLAB R2022a. Hardware control is implemented through LabVIEW 2017. The computer we used is Dell Precision 5820 Tower, equipped with the operating system Windows 10 Pro, an Intel(R) Core(TM) i9-10920X CPU @ 3.50GHz processor and 256 GB of installed RAM.

Content

1. Overview of the Reconstruction Program	2
2. Single Image Reconstruction.....	3
3. Batch Reconstruction.....	7
4. Reconstruction Tips and Parameters Trade-off.....	10
5. Measurement of the Galvanometer Step Uniformity	16
6. Guide for Hardware Implementation.....	18

1. Overview of the Reconstruction Program

This reconstruction program is used to perform super-resolution reconstruction of raw data acquired by the MC-ISM optical system. The raw data are fluorescence images acquired by multifocal illumination excitation, the number of which is equal to the number of scanning steps. The raw data should be saved in a folder named **pinhole_ALN_raw**.

If this is the first time of your executing the software, please install Matlab 2022a according to <https://www.mathworks.com/help/install/ug/install-products-with-internet-connection.html>.

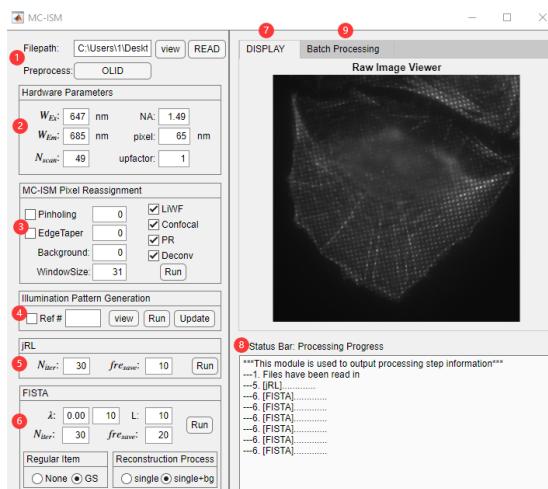
After the program is downloaded, run **MCISMXi.mlapp** to open the GUI for the MC-ISM reconstruction program. The program interface is as follows and consists of nine parts:

① Filepath & Preprocess: Determine the path where the raw data is located and perform optical lock-in detection (OLID) preprocessing.

② Hardware Parameters: User-defined system parameters.

③ MC-ISM Pixel Reassignment: User settings for PR reconstruction parameters

④ Illumination Pattern Generation: Generating illumination patterns for a multi-image deconvolution reconstruction.



⑤ jRL: User settings for jRL reconstruction parameters.

⑥ FISTA: User settings for FISTA-GS reconstruction parameters.

⑦ Display: A raw image viewer.

⑧ Status Bar: An area used to display and record the reconstruction process.

⑨ Batch Process: used for batch processing.

The step-by-step guide on running a MC-ISM reconstruction is provided in the following chapters.

2. Single Image Reconstruction

In this section, we will describe how to use the program to perform MC-ISM reconstruction, including the traditional ISM reconstruction algorithm based on pixel reassignment and frame reduction reconstruction algorithms based on multi-image deconvolution.

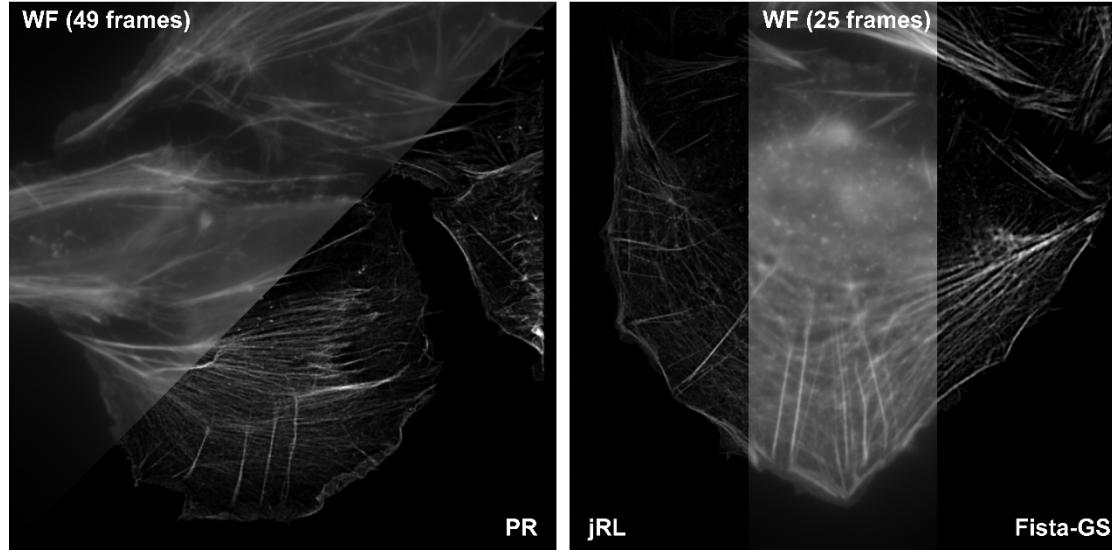


Fig 1. Imaging and reconstruction results of actin labeled by Phalloidin-Atto 647N.

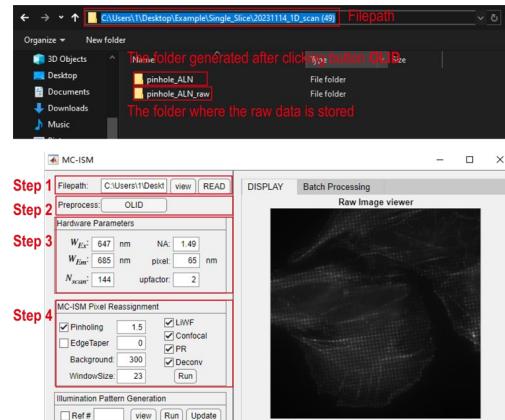
2.1 Pixel Reassignment reconstruction

Step 1: Load raw data

Ensure that the raw data is saved in a folder named **pinhole_ALN_raw**. Subsequently, copy the path of the **pinhole_ALN_raw** folder to the Filepath editing field in the GUI interface, or click the **view Button** to navigate and load the path of the **pinhole_ALN_raw** folder. For example, the sample data file path here is:

[C:\Users\1\Desktop\Example\Single_Slice\20231114_1D_scan_\(49\).](C:\Users\1\Desktop\Example\Single_Slice\20231114_1D_scan_(49).)

You can click the **READ Button** to verify if the path is valid. If valid, the first frame image of the raw data will be displayed under Raw Image Viewer on the right side.



Step 2: OLID preprocessing

Clicking the **OLID Button**, a folder named **pinhole_ALN** will be generated in the path where the **pinhole_ALN_raw** folder is located. This folder will store the images that have been preprocessed using OLID. Subsequent operations will be carried out on the images in the **pinhole_ALN** folder.

Step 3: Set Hardware Parameter

Six parameters need to be set.

W_{Ex} and W_{Em} are the excitation and emission wavelength, respectively. These two parameters are related to generating the digital pinhole and simulating the point spread function.

N_{scan} is the number of scans, which corresponds to the number of original frames needed to reconstruct a single image, *i.e.*, the number of images in the pinhole_ALN folder.

NA is the numerical aperture of the objective.

pixel represents the equivalent size of the detector pixel on the sample plane. For example, if the system magnification is 100 \times and the physical size of the detector pixel is 6.5 μm , then pixel should be set to 65 nm.

upfactor is the upsampling factor applied to the raw images. A higher upsampling factor can provide more accurate PR results, but it also introduces greater memory overhead. We limit its values to 1, 2, 4, or 8.

Step 4: Set Pixel Reassignment Parameter

The checkboxes for LiWF, confocal, PR and Deconv are for whether or not to save the results of the sum averaging of OLID images, confocal, PR and PR+deconvolution calculations, respectively.

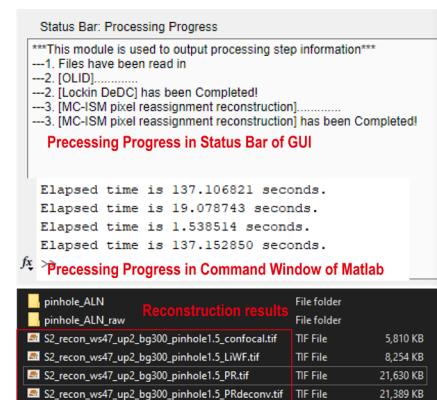
The checkbox for Pinholing determines whether digital pinhole filtering is applied to the sub-image or not, and a multiple of the PSF diameter is entered in the edit field on the back side, usually choosing 1.2~2.

The checkbox for EdgeTaper determines whether the sub-image retains its edges or not, and the percentage of the sub-image's strongest gray value that is discarded is entered in the Edit Field on the back side, *e.g.*, to discard 30% of the strongest light intensity, fill in 0.3;

Background is the threshold value when spatial localization is performed. Center points with intensities below this value are considered as background and are removed from the set of center points.

WindowSize is the size of the sub-image, which needs to include the entire spot and not cover more than one spot.

Finally, click the **Run Button** to obtain the reconstruction results. The reconstruction results will be saved in the same location as the pinhole_ALN folder. The PR parameters set are labeled in the image names.



2.2 Multi-image deconvolution reconstruction

In the MC-ISM system, increasing the step size can reduce the number of steps, thus improving the image acquisition speed. However, excessively long step sizes can lead to a decrease in the sampling rate. Since PR reconstruction stretches the spot distance to twice its original size, this can result in the appearance of gaps in the PR reconstruction results. In such cases, we can approach the problem from the perspective of image fusion. We can consider the acquired original images as results observed from different perspectives and utilize multi-image deconvolution to achieve reconstruction. This approach helps in mitigating the effects of gaps or artifacts that may arise due to stretching of the spot distance during PR reconstruction.

Step 1 ~ Step 3 are the same as PR reconstruction.

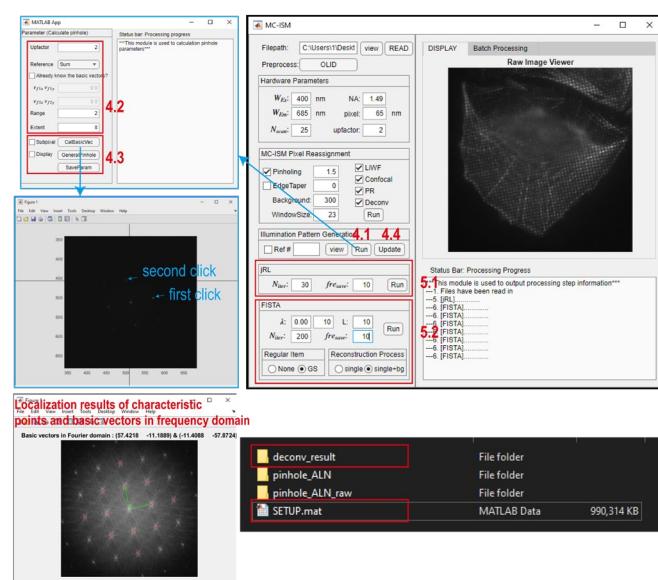
Step 4: Illumination Pattern Generation.

·4.1: Clicking the **Run Button** will pop up a sub-window for generating spatial illumination patterns. This window is divided into two parts: the left side is used for setting parameters, and the right side is used to display the processing progress.

·4.2: Set parameters for generating the illumination pattern. The accuracy of the illumination pattern directly affects the quality of multi-image deconvolution. Since the spatial distribution of the illumination pattern is a two-dimensional array, the periodicity of the spatial illumination pattern can be determined through the localization of frequency domain feature points. Subsequently, the spatial illumination pattern can be obtained through further processing.

Upfactor represents the upsampling factor, which can be directly passed from the upfactor Edit Field in the main window.

Reference indicates the index of the reference image (default is "Sum", indicating a summation of fast Fourier transform of all OLID images). By performing a fast Fourier transform on this reference image, the frequency spectrum is obtained, and basic vectors in the frequency domain are determined by localizing the feature points of the spectrum.



Range represents the highest order of feature points considered in frequency spectrum.

Extent represents the size of the sub-image to be cropped when performing sub-pixel localization of the spectral feature points.

If you already know the rough basic vectors in the frequency domain, you can check "**Already know the basic vectors?**", and then fill in the basic vectors in v_{f1x} , v_{f1y} , v_{f2x} , and v_{f2y} .

·4.3: Click the **CalBasicVec Button** to calculate the basic vectors. Note that if "Already know the basic vectors?" was not checked in 4.2, clicking the button will pop up the spectrum of the reference image, and the user will need to manually click twice to select the feature points in orthogonal directions . After the operation is complete, if **Display** is checked, a figure will pop up showing the localization results, where the red crosses are feature points used to fit the basic vectors, and the green arrows represent the calculated basic vectors.

Next, click the **GeneralPinhole Button** to generate the illumination pattern. Check **Subpixel** to ensure the center of spots in illumination pattern is subpixel level.

Finally, click **SaveParam Button**, and the program will save the relevant parameters in **SETUP.mat** under **Filepath**.

·4.4: Click the **Update Button** in the main window to close the sub-window.

Step 5: Run Deconvolution

·5.1 jRL reconstruction: ***N_{iter}*** and ***Fre_{save}*** denote the total number of iterations and the interval between iterations of the saved data for jRL, respectively. Click **Run Button** to start jRL reconstruction.

·5.2 FISTA-GS reconstruction: ***N_{iter}*** and ***Fre_{save}*** are the same as jRL reconstruction. **λ** and **L** are the hyperparameters and iteration parameters of the FISTA process, respectively.

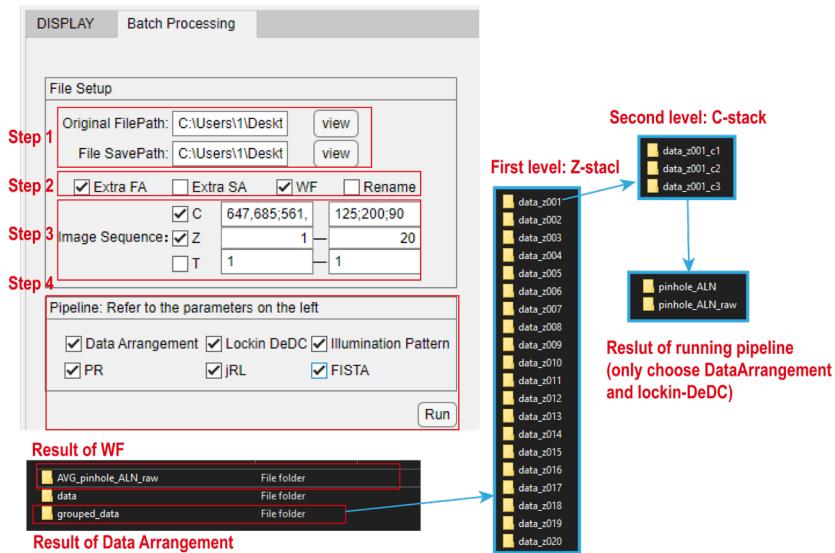
None and **GS** represent the regularization term used in FISTA reconstruction. None indicates no regularization term, while GS indicates the use of group sparsity regularization.

single and **single+bg** represent whether to consider the background from the object separately in the modeling process. The detailed meanings of these parameters can be referred to in the Supporting Information of the paper. Typically, the effects of GS and single+bg are better than None and single. The parameters λ and L can be started with default values and then fine-tuned according to the specific requirements of the reconstruction process.

The results of deconvolution are saved in folder **deconv_result** under **Filepath**.

3. Batch Reconstruction

MC-ISM supports multi-z, multi-c, and multi-t acquisitions, enabling three-dimensional, multi-color, and long-term imaging. The raw data can be substantial, possibly reaching hundreds of gigabytes. Therefore, we have developed batch reconstruction to automate the processing of data efficiently.



Step 1: Determine the path of the raw data and the path for saving the grouped data.

The raw data should be centrally stored in a folder named **data**. For example, if the raw data is stored in <C:\Users\1\Desktop\Example\Batch\data>, then you should enter the path of the folder **data** into the **Original FilePath Edit Field**, such as <C:\Users\1\Desktop\Example\Batch>.

Before performing batch reconstruction, the data needs to be grouped. Therefore, you need to specify the path where the grouped data will be saved. Create a folder named **group_data** in the same directory as the **data** folder. Enter the path, such as C:\Users\1\Desktop\Example\Batch\grouped_data, in the **File SavePath Edit Field**.

In practice, the **data** folder and the **group_data** folder do not have to be in the same directory. Since the grouping operation does not delete the original data, it is essentially a sequential copy of the original data, effectively doubling the data size. This could potentially lead to a situation where the data cannot store in the same storage device. Therefore, the **data** folder and the **group_data** folder can be placed on different directory (or drives) to avoid storage capacity limitations.

Step 2: Determine the grouping parameters.

The MC-ISM program can handle both one-dimensional scanning data and two-dimensional scanning data. For hardware systems with two-dimensional scanning, there are two galvanometers scanning along the fast axis and slow axis. In some cases, there may be instability in the initial movement of the galvanometer, requiring the mirrors to make a redundant scan. However, during the redundant scan, the camera still captures images based on trigger pulses. Therefore, in the grouping process, we need to discard the photos captured during the redundant scan.

Extra FA represents the fast axis redundant scan, while **Extra SA** represents the slow axis redundant scan. Figure 2 illustrates the grouping scenarios under different settings of Extra FA and Extra SA.

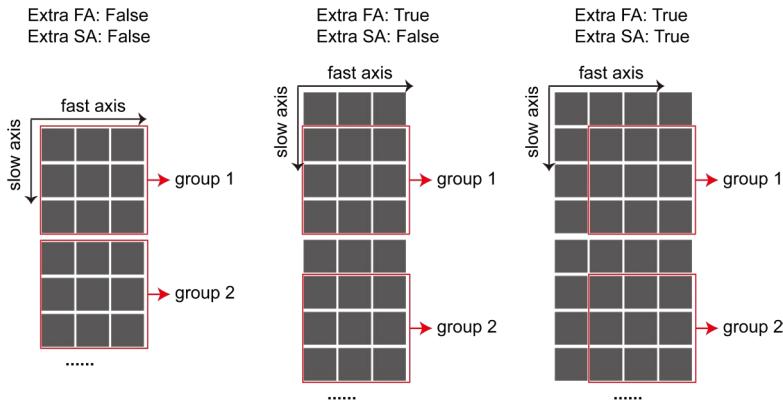


Fig 2. Schematic of different grouping settings. Each gray square represents an image captured by the camera, with galvanometers scanning in a two-dimensional single-directional manner. Assuming a 3×3 step numbers for the two-dimensional scan.

If the **WF** is checked, after grouping, a folder named **AVG_pinhole_ALN_raw** will be generated in the path where the **grouped_data** folder is located. This folder is used to save the wide-field images that have been averaged for each group.

If the **Rename** option is checked, the images will be renamed before grouping. For example, using the ORCA-Flash4.0 V3 Hamamatsu camera, it automatically names the captured images as *Imagexxx_00001.tif*, where the number after the underscore represents the image index for a single capture. Note that the number after the underscore is a 5-digit number. If the data collection exceeds 99,999 images, the 100,000th image will be named *Imagexxx_100000.tif*. Due to the sorting method in MATLAB following string comparison rules, *Imagexxx_100000.tif* should logically be placed at the end. However, according to string comparison rules, any the first number after the underscore greater than or equal to 1 will be placed after *Imagexxx_100000.tif*, causing the grouping order to be disrupted. Therefore, the **Rename** checkbox is used to rename images from *Imagexxx_00001.tif* to *Imagexxx_000001.tif* up to *Imagexxx_99999.tif* to *Imagexxx_099999.tif*. The best practice is to make these modifications in the hardware control software before capturing the images. If this renaming rule does not meet the user's requirements, the user can open the **Batch1_FileArra.m** file and make adaptive modifications to the **data_rename(base_data_path)** function.

Step 3: Determine the Image Sequence.

There are at most two levels of the **grouped_data** files: **Z(Z-stack)/T(long term)-C(channel)**. The first level folder is the **Z-stack** or **T-stack**, the second level folder is the **C-stack**. They depend on the checkbox of **C**, **Z** and **T**.

The two Edit Fields following C are used to label the excitation wavelength, emission wavelength, and background for each channel. For example, when performing three-color imaging on Mouse kidney sections (FluoCells #3, F24630, ThermoFisher), with excitation wavelengths of 647 nm, 561 nm, and 405 nm, and emission wavelengths of 685 nm, 585 nm, and 425 nm, and background for PR reconstruction of 25, 200, and 90 respectively, you should enter the following information:

In the first Edit Field: 647,685;561,585;405,425

In the second Edit Field: 25;200;90

The two Edit Fields following Z are used to indicate the starting and ending layers for grouping.

The two Edit Fields following T are used to indicate the starting and ending time points for grouping.

If multi-Z or multi-T is performed and **C** is not checked, it defaults to single-color imaging. The excitation wavelength, emission wavelength, and background threshold values are taken from the parameters specified in the Hardware Parameters section on the left side of the window.

Step 4: Run Batch Reconstruction Pipeline.

The checkbox for Data Arrangement determines whether grouping will be performed. **The checkboxes for Lockin DeDC, Illumination Pattern, PR, jRL, and FISTA** determine whether OLID preprocessing, illumination pattern generation, PR reconstruction, jRL reconstruction, and FISTA reconstruction will be performed during batch processing. All parameters required for reconstruction, except for excitation wavelength, emission wavelength, and background, are taken from the parameters specified in the left-side window.

When generating illumination patterns in batches, it is necessary to manually generate the illumination pattern for the first group of data through single-image reconstruction, *i.e.*, generating **SETUP.mat** in the corresponding path. Subsequently, other groups will reference the basic vectors determined for the first group to calculate their own feature point positions. Additionally, considering that with multi-layer scanning, as the sample depth increases, the image signal-to-background ratio decreases, and the spot may become invisible. In such cases, it may be necessary to reference the illumination pattern from other layers. To do this, click the **view Button** in the Illumination Pattern Generation section on the left, select the desired **SETUP.mat**, and check the **Ref Edit Field**.

Furthermore, the different steps in the pipeline have a sequential relationship, as shown in Figure 3. For example, if Illumination Pattern has not been run before attempting to run jRL directly, the program will throw an error. However, if Data Arrangement and Lockin DeDC have already been run, Illumination Pattern does not need to be run before jRL can be executed.

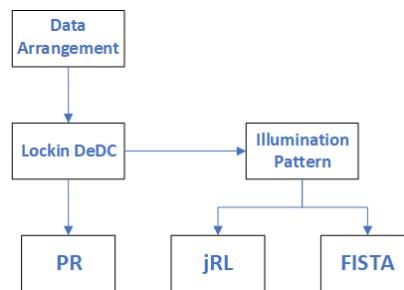


Fig 3. The sequential relationship of different steps in the pipeline.

Finally, click **Run Button** to start the batch reconstruction pipeline.

4. Reconstruction Tips and Parameters Trade-off

① Center Positioning in Spatial Domain

Open the file `Level11_spatial_PR.m`, uncomment lines 83 to 85, and add a breakpoint at line 86. When performing PR reconstruction, the program will stop at the breakpoint and display the spatial domain center positioning results. Based on these results, we can determine the accuracy of the **Background** parameter.

The principle of center positioning in spatial domain is to take the maximum grayscale value in the eight-neighborhood. Before comparing pixel grayscale values, the image needs to be preprocessed. We first apply median filtering to the OLID image, followed by Gaussian filtering, to minimize the impact of noise on the positioning and to ensure that the grayscale distribution of the spot is close to Gaussian. This way, the center point is close to the geometric center of the spot. The size of the Gaussian filter standard deviation is defined in line 19 of `Level11_spatial_PR.m`, for example:

```
sigma = 3
```

If the spatial domain fixation effect is not satisfactory, try adjusting the `sigma` value.

We also encourage users to try other preprocessing methods, such as wavelet filters, lowered Gaussian filters, and morphological operations, etc. **After adjustments are complete, remember to re-comment lines 83 to 85 and remove the breakpoint.**

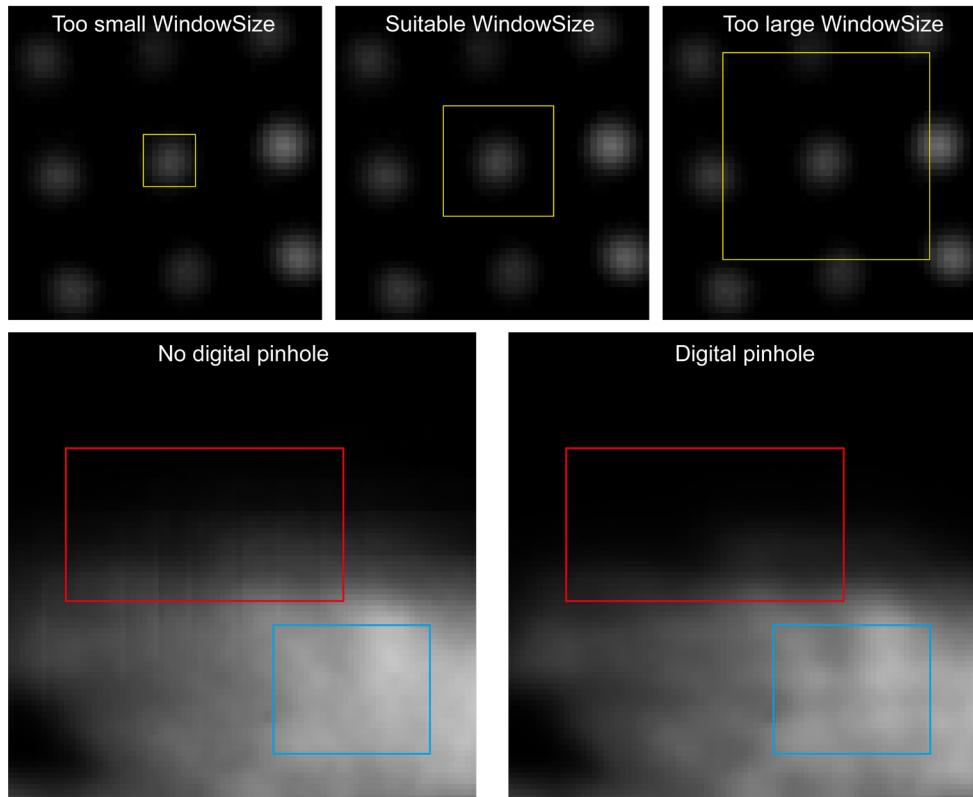


② WindowSize and Pinholing

WindowSize and **Pinholing** are two crucial parameters for PR reconstruction. During PR reconstruction, a subimage with a side length of WindowSize is extracted centered at the spot center coordinates. Firstly, WindowSize needs to be an odd number. Secondly, WindowSize should not be too large, so as to include more than one spot, nor too small to contain a single spot.

After extracting the subimage with WindowSize, you can choose whether to add a digital pinhole

to the subimage. The center of the digital pinhole is the subpixel coordinate of the spot in the subimage, and its size is controlled by the value in the edit field after Pinholing. The pinhole can further remove out-of-focus signals and make the edges of the subimage gradually attenuate to zero, reducing edge artifacts in the image. As shown in the red boxed area in the image below, artifacts are effectively removed after adding a digital pinhole. However, in regions where the signal itself is very strong and the spot fills the subimage area, adding a digital pinhole may attenuate some in-focus information. If the pinhole is too small, grid artifacts may appear, as shown by the slight artifacts in the blue boxed area in the image below. Generally, Pinholing is set to 1.2 to 2.



③ Upfactor

Upfactor is the upsampling factor applied to the raw images. A higher upsampling factor can provide more accurate PR results, but it also introduces greater memory overhead. We limit its values to 1, 2, 4, or 8. Generally, an Upfactor of 2 is sufficient to avoid PR reconstruction artifacts caused by pixilation.

④ Excitation Wavelength and Emission Wavelength

The parameters W_{Ex} and W_{Em} define the excitation wavelength λ_{ex} and emission wavelength λ_{em} used during reconstruction, which in turn determine the size of the excitation PSF and the detection PSF:

$$\sigma_{ex/em} = 0.61 \frac{\lambda_{ex/em}}{\text{NA}} \frac{upf}{2.355ps}$$

$$\text{PSF}_{\text{ex/em}}(x, y) = \frac{1}{\sigma_{\text{ex/em}} \sqrt{2\pi}} \exp\left(-\frac{1}{2} \frac{x^2 + y^2}{\sigma_{\text{ex/em}}^2}\right),$$

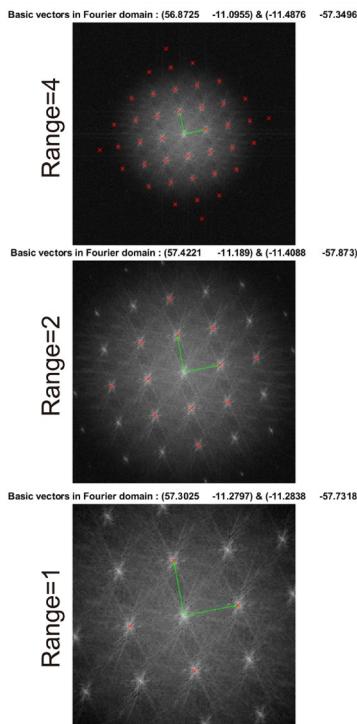
where upf denotes the upsampling factor, ps denotes the pixel size, and NA stands for the numerical aperture of the objective lens. For PR reconstruction, the requirements for \mathbf{W}_{Ex} and \mathbf{W}_{Em} are not actually stringent because appropriate scaling can be achieved through **Pinholing**. However, for deconvolution algorithms, the size of the pinhole can significantly affect the correctness of the reconstruction and may not necessarily match the wavelengths actually used in the system. \mathbf{W}_{Ex} directly influences the size of the generated illumination pattern spot, and its impact is more obvious than that of \mathbf{W}_{Em} . Therefore, after generating **SETUP.mat** in Section 2.2 Step 4.4, **SETUP.mat** can be loaded into Matlab to observe whether the illumination pattern and the original image match by using the following codes:

```
figure; imshow(SETUP.imgStack(:,:,:1),[]);
figure; imshow(SETUP.illumination(:,:,:1),[]);
figure; imshow(SETUP.imgStack(:,:,:1) .* SETUP.illumination(:,:,:1),[]);
```

⑤ Range and Extent

Range and **Extent** are two important parameters for generating illumination patterns. Assuming the rough basic vectors are v_{f1} and v_{f2} , feature points will be in the neighborhood of $v_{m,n}$ and $v_{m,n}$ satisfies the following relation:

$$v_{m,n} = [m \ n] \begin{bmatrix} v_{f1} \\ v_{f2} \end{bmatrix} \quad m, n \in \mathbb{Z} \quad m + n \leq \text{range}$$



Subsequently, the program will extract a subimage centered at $v_{m,n}$ with a side length of $2 \times \text{Extent} + 1$, and find the subpixel coordinates of the maximum value in the subimage as the feature point coordinates. Finally, all the feature point coordinates and the corresponding m and n are subjected to a least-squares fitting to obtain the corrected basic vectors.

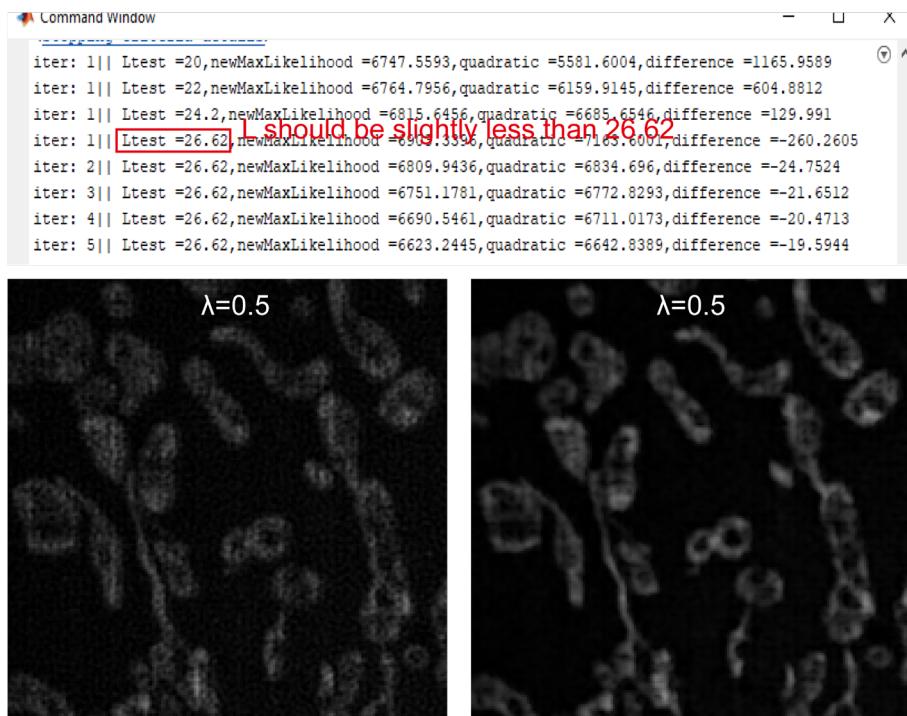
Obviously, the larger the Range, the more feature points are collected, and the more accurate the corrected basic vectors obtained from the fitting should be. However, when the image signal-to-noise ratio is too low, the higher-order feature points are not obvious and may not appear as local maxima in the spectrum. Therefore, if the Range is too large, it may collect incorrect higher-order feature point coordinates, leading to larger least-squares fitting errors. Extent determines the size of the subimage. Clearly, Extent should not be larger than the distance between adjacent feature points, and it should not be smaller than the error between the feature point position determined by the rough basic vectors and the actual feature point

position.

⑥ Hyperparameter Settings for FISTA-GS Deconvolution

Click on **Run Button** for FISTA and observe the output in the Matlab Command Window. Generally, the "difference" initially becomes greater than 0, then gradually decreases until it becomes negative, as the number of iterations increases. The actual setting for **L** should be slightly less than the value Ltest when the number of iterations begins to increase. As the iterations proceed, the "difference" gradually approaches 0.

The regularization coefficient **λ** is used to measure the impact of the regularization term; the larger its value, the sparser the resulting reconstructed image will be. Users can start with the default parameter and adjust it as needed.



⑦ Adjusting the Degree of OLID DC Suppression

The principle of OLID is that the in-focus signal, influenced by the illumination patterns, exhibits Alternating Current (AC) characteristics, whereas the out-of-focus signal remains constant, exhibiting Direct Current (DC) characteristics. Therefore, defocusing can be removed from the raw image by suppressing the DC component. The most thorough method of defocusing is to set the zero-frequency component to zero when performing Fourier transformation pixel by pixel along the scanning direction. Although this results in the loss of some in-focus signals, our experiments have shown that this method provides excellent optical sectioning effects and image fidelity.

However, when using multi-image deconvolution for frame reduction reconstruction, there may be issues. Due to the reduced sampling rate, the overlap of the spot during stepping in the raw image decreases, and the information at the junctions is reduced. If the above method is still used for OLID, grid artifacts may appear with the increasing iterations of FISTA-GS. The solution is to open `func/f1_preprocessLockin.m`, comment lines 20 to 30,

```

fft_img = fft(rawimg,[],3);
fft_img2 = fft_img(:,:,2:end);
fft_img0 = sum(fft_img2,3);
fft_img(:,:,1) = zeros(height, width);
img_pre = ifft(fft_img,[],3);
for ImgCount = 1:n
    temp = img_pre(:,:,ImgCount);
    temp(temp<0) = 0;
    temp = abs(temp);
    imwrite(uint16(temp),strcat(filepathsave,num2str(ImgCount,'%03d'),'.tif'));
end

```

and uncomment lines 33 to 38,

```

minRawimg = min(rawimg, [], 3);
rawimg = rawimg - repmat(minRawimg, 1, 1, n);
for ImgCount = 1:n
    temp = rawimg(:,:,ImgCount);
    imwrite(uint16(temp),strcat(filepathsave,num2str(ImgCount, '%03d'),'.tif'));
end

```

This way, the method of DC suppression is not directly setting the zero frequency to zero but instead lowering the pixel values in the scanning direction so that the minimum grayscale value is reduced to zero. This adjusted OLID processing results in images with more noise, making them less suitable for PR reconstruction but potentially more beneficial for deconvolution reconstruction. Of course, users can also choose an intermediate method to suppress the DC component to a more appropriate degree.

⑧ Regarding the scaling of grayscale values when saving images

The MC-ISM reconstruction program reconstructs two-dimensional images and saves them as 16-bit .tif format images. Before storage, the maximum grayscale value is scaled to $2^{16}-1$. Taking PR as an example, open `Level1_spatial_PR.m`, lines 210 to 221 implement the saving of OLID wide-field, confocal, PR, and PR+deconv results.

```

if isLiWF
    imwrite(uint16(2^16*LiWF_image./max(LiWF_image(:))),filename_LiWF);
end

if isconfocal
    imwrite(uint16(2^16*confocal_image./max(confocal_image(:))),filename_confocal);
end

if isISM
    imwrite(uint16(2^16*ISM_image./max(ISM_image(:))),filename_ISM);
end

if isDeconv
    imwrite(uint16(2^16*I_deconv./max(I_deconv(:))),filename_ISM_deconv);
end

```

When performing three-dimensional reconstruction, since it is reconstructed layer by layer, each layer is divided by its own maximum. Therefore, the relative changes in the strength of grayscale

values between different layers will be lost. The solution is to modify `max()` to a fixed value, which can be the maximum pixel value of the entire three-dimensional stack.

5. Measurement of the Galvanometer Step Uniformity

We have extracted some functionalities from the MC-ISM reconstruction code to be used for hardware testing, with the most critical aspect being the assessment of the uniformity of the galvanometer step distance during scanning. The conventional method involves performing a multi-focus scan on a fluorescent plate, followed by overlaying the obtained multi-focus images to observe uniformity of intensity. This method is quick but relatively crude, lacking precision in determining the specific movement of the galvanometer at each step. Therefore, we directly perform subpixel-level localization of the spots' center in the raw images and then quantify the movement of the galvanometer by measuring the distance between the center position of spots in adjacent frames. The relevant code can be found in the folder named `Lattice_uniformity`. Here is how to use it:

Step 1: Data Acquisition

Use MC-ISM to image the fluorescent plate and save the files in the `pinhole_ALN_raw` folder.

Step 2: Set the Parameters.

Launch Matlab 2022a and open the file `batch_avergeStep.m`. Lines 1 to 14 of the code encompass the setting of 6 parameter.

file_path: Path where the raw images are located (up to the level of `pinhole_ALN_raw`).

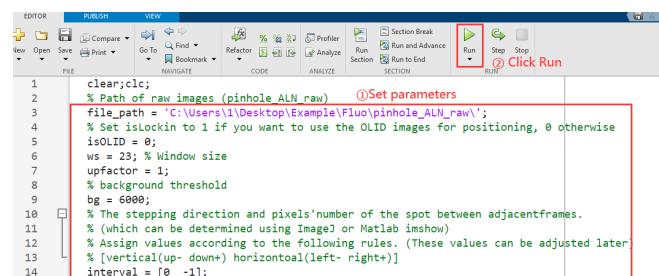
isOLID: Indicates whether to use OLID images for center position localization. If set to 1, it will automatically perform OLID on the raw images and save the processed images in the `pinhole_ALN` folder.

ws, **upfactor** and **bg** are window size, upsampling factor and background threshold. These three parameters are no different from those in MC-ISM reconstruction program.

interval: Represents the step distance and direction of spots between adjacent frames, specified as a 1×2 vector following the [vertical (up-down+), horizontal (left-right+)] rule. For example, if the spot moves approximately 3 pixels to the left, `interval = [0 -3]`; if it moves approximately 7 pixels down, `interval = [7 0]`. This parameter can be roughly determined using ImageJ, and the code will prompt the user to adjust it as needed. After setting the parameters, click **Run**.

Step 3: Adjusting Interval

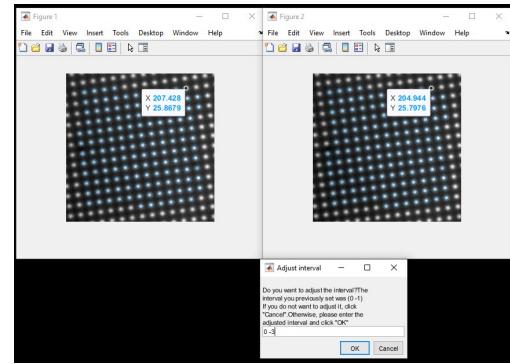
After completing the sub-pixel level center positioning, the program will display two figures showing the adjacent raw images and their center positioning. A dialog box will pop up asking if you want to adjust the **interval**. The two figures will show lattice images of the fluorescent plate



```
1 clear;clc;
2 % Path of raw images (pinhole_ALN_raw)
3 file_path = 'C:\Users\l1\Desktop\Example\Fluo\pinhole_ALN_raw\''; ①Set parameters
4 % Set isLockin to 1 if you want to use the OLID images for positioning, 0 otherwise
5 isOLID = 0;
6 ws = 23; % Window size
7 upfactor = 1;
8 % background threshold
9 bg = 6000;
10 % The stepping direction and pixels'number of the spot between adjacent frames.
11 % (which can be determined using ImageJ or Matlab imshow)
12 % Assign values according to the following rules. (These values can be adjusted later)
13 % [vertical(up- down+), horizontal(left- right+)]
14 interval = [0 -1];
```

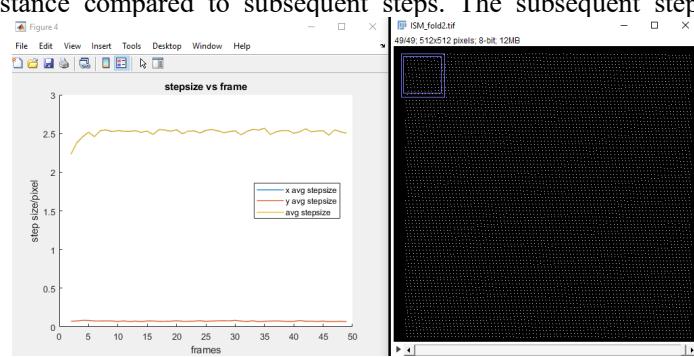
excited by multi-focus illumination patterns, with the subpixel centers of the spots highlighted by blue circles. By clicking on the blue circles on the same spot in both images, you can obtain their coordinates and compare them to determine the step distance between adjacent frames.

It is evident that the spot's step distance is approximately 3 pixels to the left, which is significantly different from the previously set interval of $[0 - 1]$. The dialog box also prompts us with "The interval you previously set was (0 - 1). If you do not want to adjust it, click 'Cancel'. Otherwise, please enter the adjusted interval and click 'OK'." We input " $0 - 3$ " in the edit field and click **OK** to continue the program.



Step 4: Obtaining Results

Finally, a figure will pop up displaying the average step distance of the galvanometer at each step. From the line graph, it is apparent that under the same voltage conditions, the first step of the mirror has a significantly smaller step distance compared to subsequent steps. The subsequent step distances fluctuate slightly, less than 0.2 pixels, indicating a relatively stable motion of the galvanometer. Additionally, a three-dimensional image stack named **ISM_fold2.tif** will be generated in the raw data path, allowing you to view the center positioning results.



6. Guide for Hardware Implementation

6.1 Light Path Adjustment Steps

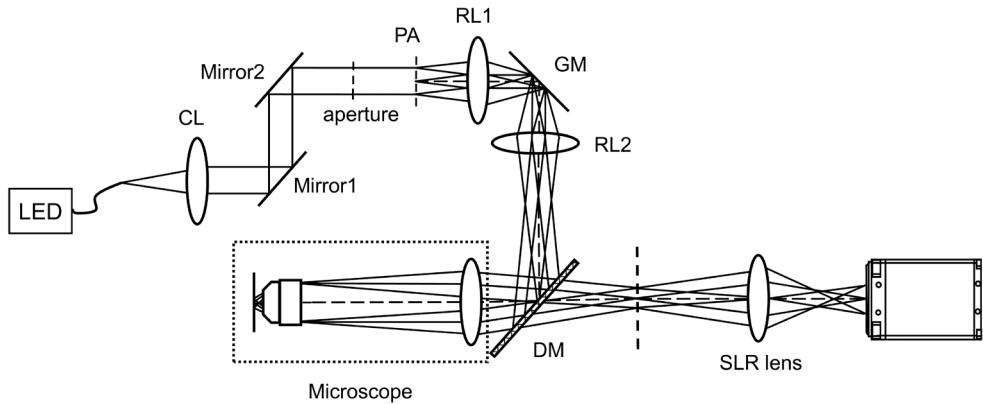


Fig 4. Optical Path Diagram.

Step 1: LED Collimation.

1. In the cage system, position the adapter of the liquid light guide (LLG) and the collimating lens.
2. Adjust the distance between the lenses back and forth to make the exiting LED beam as collimated as possible (perfect collimation is difficult to achieve with an LED).
3. Adjust the pitch knob of the mounting to make the beam spot as circular as possible.

Step 2: Coarse Adjustment of the Galvanometer.

1. Use two mirrors (Mirror 1 and 2) to coarsely align the LED to vertically enter the galvanometer system.
2. Place a cage alignment plate (CPA1, Thorlabs) in front of the galvanometer system (cage), and adjust the mirror 1 to center the beam on the alignment plate.
3. Replace the alignment plate with a mirror (screw on the lens mount), and adjust the mirror 2 so that the beam passes through the aperture and aligns with the returning beam.

Step 3: Calibration of the Galvanometer.

1. Set the galvanometer to zero voltage.
2. Extend the cage system and place an alignment plate behind RL2.
3. Let the X and Y galvanometer scan the same angle range (that is, the same voltage) to ensure that the beam formed by scanning is on the center of the alignment board.
4. Continue scanning individually in the X and Y directions for verification.

Step 4: Adjustment of Relay (Scan) Lenses

1. Reserve the distance from the microscope to RL2, accounting for 85 mm (the distance from the

microscope side port to the sample conjugate plane) plus 75 mm (the focal length of RL2).

2. Position the RL1 and RL2 symmetrically around the center of the galvanometer, each 70mm from the galvanometer (5mm inside).
3. After adding the RL1, adjust the XY translation and pitch knob to align the beam spot with the original optical axis (use the alignment plate).
4. Adjust the RL2 in the same manner.

Step 5: Checking Parallel Light

1. After installing the lenses (RL1 and RL2), adjust the position of the RL1 and RL2 to ensure parallel light output.
2. Maintain symmetrical distances between the two lenses.
3. Repeat Step 4.

Step 6: Vertical Incidence into Microscope

1. Use the objective rear focal plane reflector (half reflection and half transmission) that comes with the Nikon microscope and screw it onto the objective interface.
2. Position DM appropriately to ensure the outgoing beam spot is centered at the objective rear focal plane reflector.
3. Adjust the DM so that the returning beam passes through the aperture center.

Step 7: Checking Conjugate Conditions

1. Verify that the parallel beam converges at the objective rear focal plane reflector after passing through the tube lens.
2. Check that the scan stationary point coincides with the tube lens convergence point.

Step 8: Adjusting Convergence Point

1. Adjust the position of the RL2 so that the point of convergence after the tube lens is centered at the objective rear focal plane reflector.
2. Repeat Step 4.

Step 9: Adjusting the Detection Optical Path Axis and Conjugate Relationship

1. Use a plant sample slice under a microscope bright field light source to preliminarily find a conjugate image of the sample.
2. Try to keep the SLR lens and camera physically connected, and insert an emission filter between them to prevent ambient light and stray light from entering.
3. Move the SLR lens and camera at the same time to find a clear image of the plant sample, and try to ensure that the SLR lens is perpendicular to the emission beam.
4. Place the pinhole array 75 mm in front of the RL1, move it back and forth, and find a clear

image of the pinhole array on the camera.

6.2 Acquisition Steps

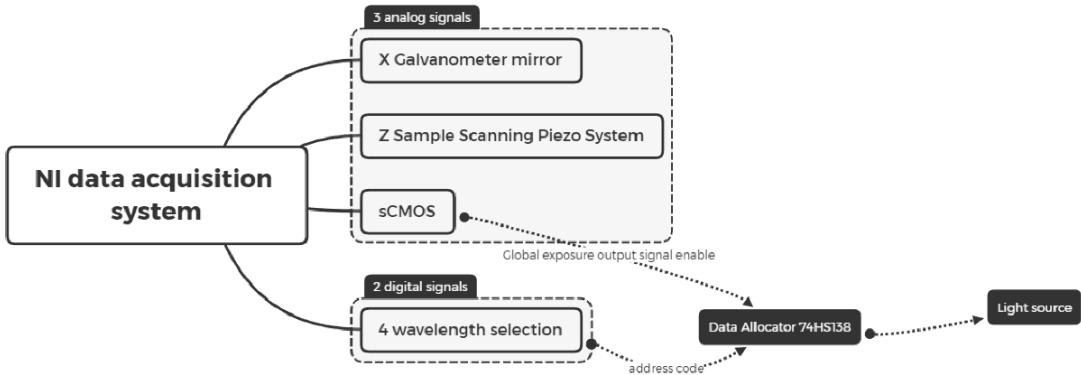


Fig 5. Hardware Control Diagram.

1. Connect the NI data acquisition system, galvanometer, camera, and light source according to the provided hardware control diagram as shown in Fig 5.
2. Set the parameters you want to collect according to the example and explanation shown in **“Section 6.3 Hardware Parameter Settings”**. The voltages in the X and Y directions will vary depending on the pinhole size, spacing, and position of the conjugate plane of the optical path (For 1D scanning, it is also related to the angle of the pinhole array deflection). Adjust the voltages according to the method given in **“Section 5 Measurement of the Galvanometer Step Uniformity”** until there are no artifacts.
3. The LabVIEW and camera parameters listed in the Fig 6 to Fig 9 are those that should be set when collecting a set of data (one reconstructed image). In this configuration, 49 scans are performed in the X direction, each image is exposed for 5 ms, the image size is 1024×1024 pixels, and it is a single-color single-slice acquisition. With these parameters, an imaging speed of 2 frames per second can be achieved.

Table 1 List of components used in the MC-ISM.

Component	Manufacturer	Notes	Link	Replaceable	Quantity	Price
ACL2520U-A	Thorlabs	Collimating lens	https://www.thorlabs.com/nextgroupage9.cfm?objectgroup_id=3835&pn=ACL2520 U#4361		1	¥ 295
CFS1/M	Thorlabs	For mounting and manually switching pinhole arrays.	https://www.thorlabs.com/nextgroupage9.cfm?objectgroup_ID=10262	ELL9K (Automatic switching)	1	¥ 1,585
Pinhole Array	Juzhike optoelectronics	For producing multi-focal points	http://www.jzkgd.com/		1	¥ 1,000
AC254-075-A-ML	Thorlabs	Relay lenses (scan lens) 1 and 2	https://www.thorlabs.com/nextgroupage9.cfm?objectgroup_id=2696		2	¥ 1,015
KC1XY/M	Thorlabs	Lens Mount (cage-compatible)	https://www.thorlabs.com/nextgroupage9.cfm?objectgroup_id=185	CXY1A (Thorlabs)	3	¥ 1,678
S-8107	Sunny Technology	For 1D or 2D small angle scanning	http://www.sunny-technology.com/pro_con.aspx?type=25&id=794	GVS202+ GPS011-EC (Thorlabs)	1	¥ 5,000
GCM001	Thorlabs	Galvo Mount (cage-compatible)	https://www.thorlabs.com/nextgroupage9.cfm?objectgroup_id=3770	GCM102/M (Thorlabs)	1	¥ 1,442
ER3-P4	Thorlabs	Cage Assembly Rod	https://www.thorlabs.com/nextgroupage9.cfm?objectgroup_ID=4125		2	¥ 236
89402 Multi LED set	Chroma	Multi-band filter set (Ex, DM, Em)	https://www.chroma.com/ducts/set/89402-et-391-32-479-33-554-24-638-31-multi-led-set		1	\$1550.00
#58-874	Edmund optics	Kinematic dichroic mirror mount	https://www.edmundoptics.com/f/kinematic-filter-mounts/13406/			¥ 1,040
105mm F2.8 DG DN MACRO Art	SIGMA	single lens reflex (SLR) lens	https://www.sigmalglobal.com/en/lenses/a020_105_28/		1	¥ 2800
						Total: ￥ 31,858

6.3 Hardware Parameter Settings

In this section we show the LabVIEW parameter settings, and the settings of the camera software (HCImageLive) parameters that match the LabVIEW. Additionally, we have also created **Table 2**, which lists in detail the meanings of various parameters on the LabVIEW front panel. For the meanings of camera software parameters, you can refer to the official documentation of the Hamamatsu camera <https://hcimage.com/support/>.

Table 2 Explanation of parameters in the LabVIEW front panel.

Parameter name	Parameter properties	Notes
X size	Adjustable	The number of scanning steps of the galvo along the fast axis.
Y size	Adjustable	The number of scanning steps of the galvo along the slow axis.
Amplitude X	Adjustable	The total step voltage of the galvo in the fast axis direction determines the scanning step of the galvo along the fast axis.
Amplitude Y	Adjustable	The total step voltage of the galvo in the slow axis direction determines the scanning step of the galvo along the slow axis.
Scan range(um)	Adjustable	The depth is collected in the axial (z-axis) direction.
Step size(um)	Adjustable	The step distance collected in the axial (z-axis) direction.
Multicolor	Adjustable	The number of laser wavelengths used for multicolor imaging.
Exposure time	Adjustable	Exposure time.
Vertical	Adjustable	The number of rows of the acquired image. This parameter determines the camera readout time.
Point precision us/point	Adjustable	Sampling accuracy, generally 10us/point.
Slope	Adjustable	The ratio of the galvo movement time to the entire readout time. This parameter is generally 30%.
Dwell time	Adjustable	The ratio of the galvo stabilization time to the entire readout time. This parameter is generally 70%.
Z calibration	Adjustable	The piezo scanning stage is calibrated so that its initial axial (z-axis) position is at a desired position.
First	Adjustable	When the galvo steps along the fast axis, the voltage of the first step is a multiple of the voltages of the other steps.
Num of cycle	Adjustable	Cycle times for multicolor and axial scanning.
Num of captured cycle	Adjustable	The number of cycles of multicolor and axial scans acquired.
Time consumed	Automatic calculation	Time to complete the entire scan.
Camera frame rate	Automatic calculation	When the camera uses external triggering, the number of frames collected per second. This parameter needs to strictly match the camera software (HCImageLive).
Number of camera frame	Automatic calculation	The total number of images captured by the camera per cycle. This parameter needs to be filled in the camera software (HCImageLive).
Acquisition rate	Automatic calculation	Rate calculated based on the acquisition time per cycle.

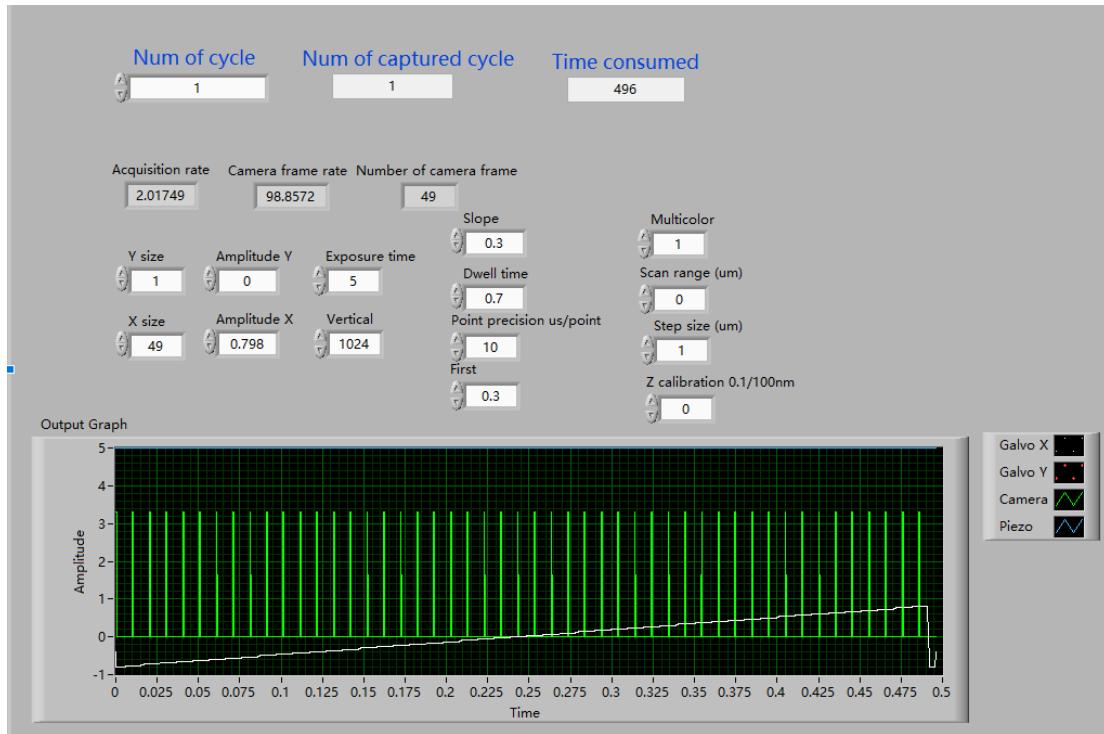


Fig 6. LabVIEW control code front panel.

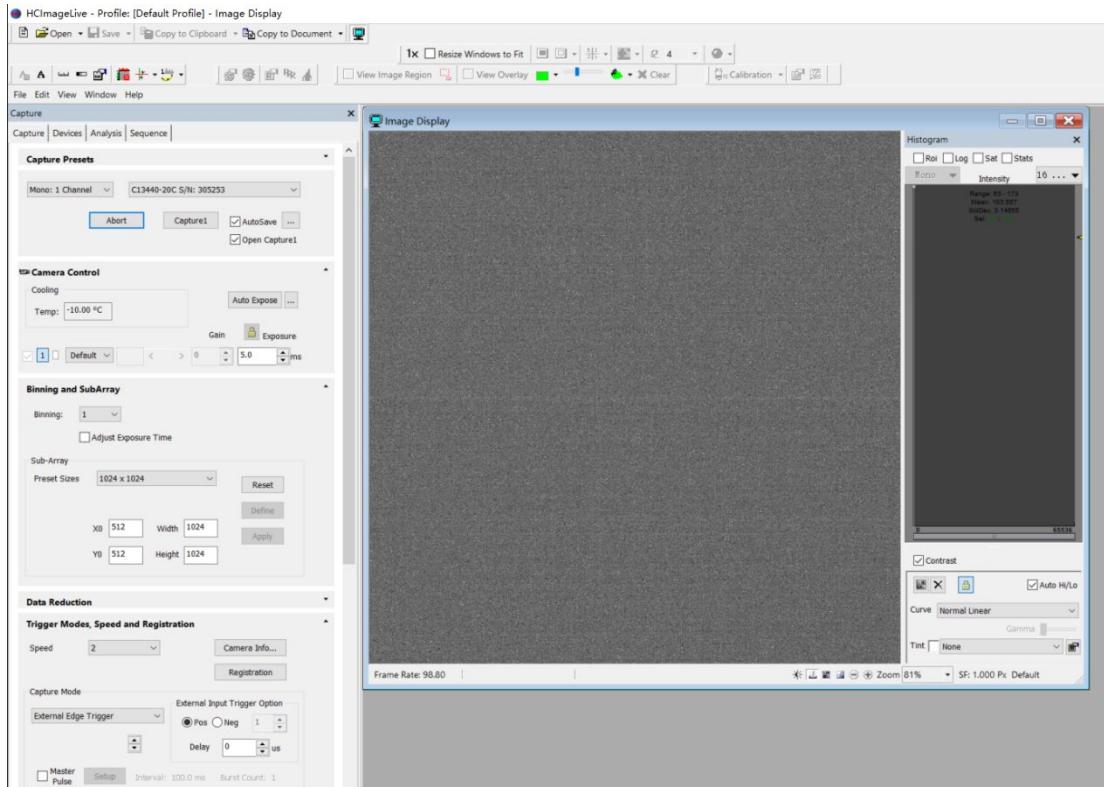


Fig 7. Screenshot 1 of the camera settings that match the LabVIEW control parameters.

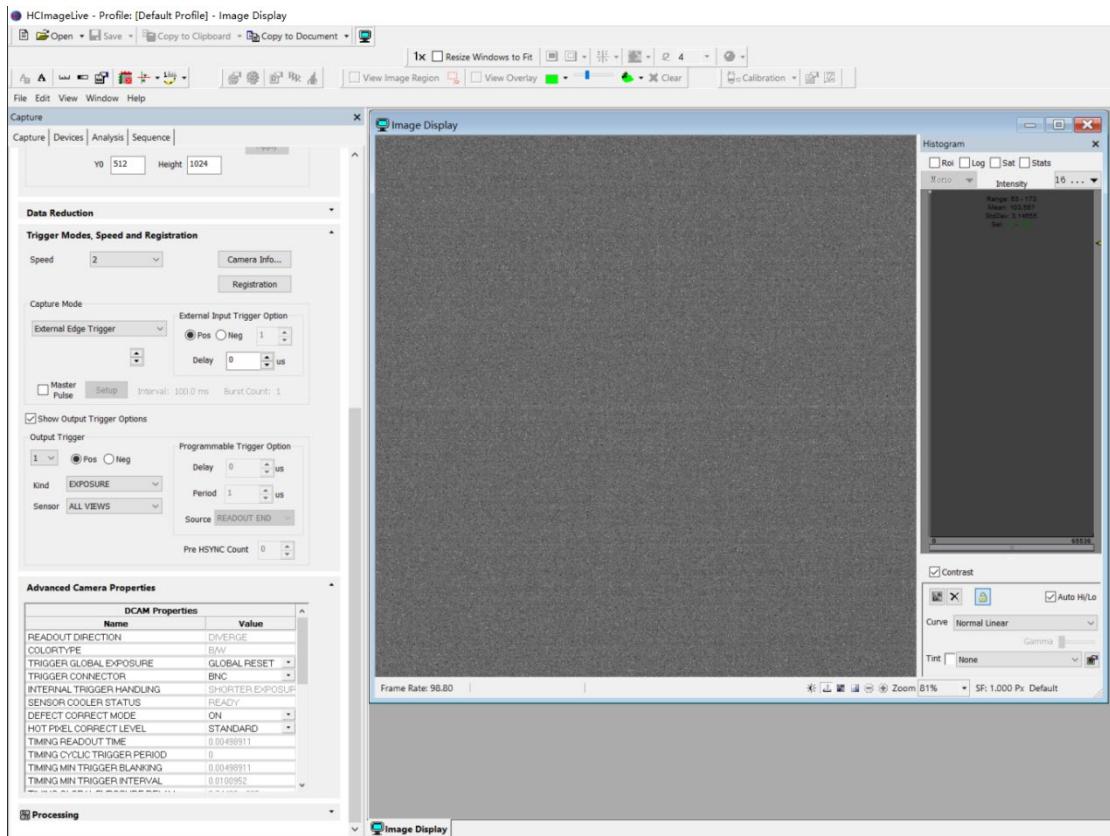


Fig 8. Screenshot 2 of the camera settings that match the LabVIEW control parameters.

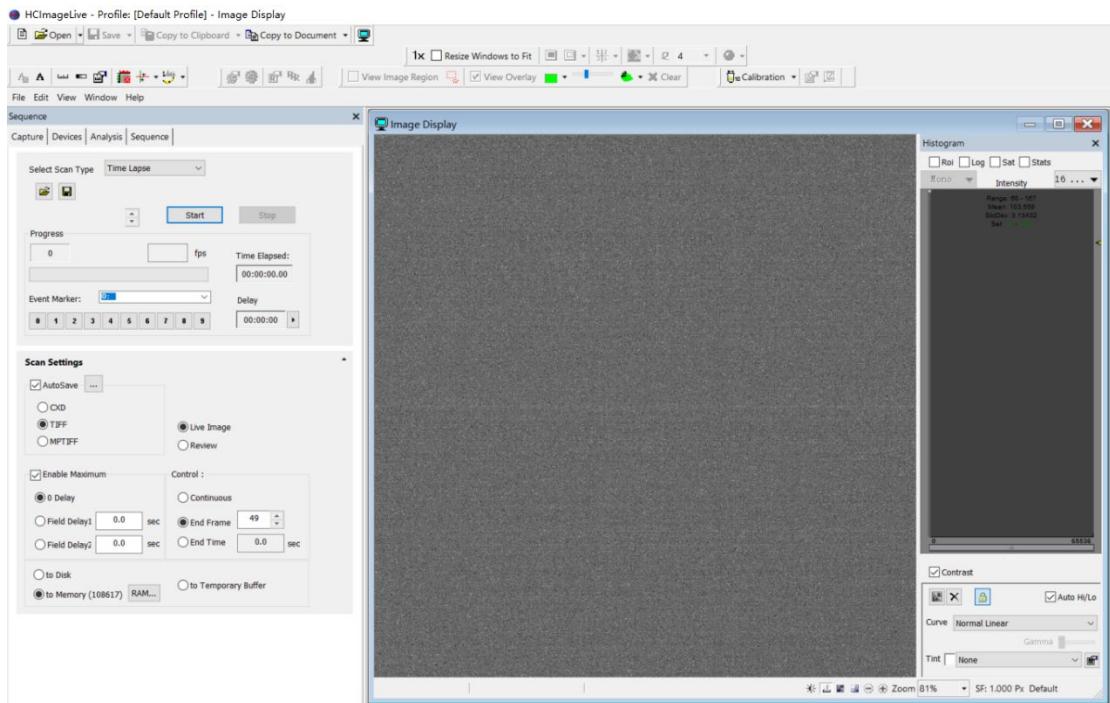


Fig 9. Screenshot 3 of the camera settings that match the LabVIEW control parameters.