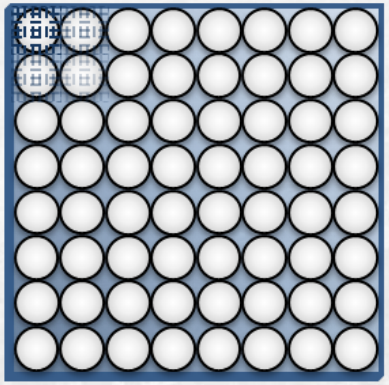


LFIT v2

LIGHT FIELD IMAGING TOOLKIT V2

An Introduction to the Light Field Imaging Toolkit (v2.23)

Presented by: Jeffrey Bolan & Elise Munz

Primary developers: Jeffrey Bolan and Kyle Johnson

Many thanks to Dr. Brian Thurow, Tim Fahringer, Paul Anglin, Dominic Hildebrandt, and Chelsea Thomason for extensive discussions and various contributions that made this toolkit possible.

Table of Contents

Basic overview of LFITv2 capabilities

INTRODUCTION

Primary Features

- The Light Field Imaging Toolkit (LFIT) facilitates quick processing of plenoptic images
  - Perspective Shifts
  - Refocusing
  - Generation of Perspective Sweep Animations
  - Generation of Refocusing Animations
  - Focal Stack Export

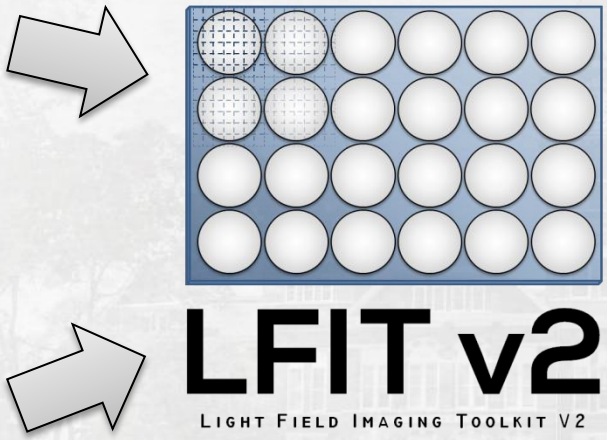
Toolkit Overview

- Two primary ways to use the toolkit
  - Single Image (GUI) mode
    - User interface provides options to load and process a single plenoptic image at a time
    - User can then adjust parameters in the interface before exporting perspective shifts, refocused images, etc.
  - Batch Processing (script) mode
    - User edits a script file
      - Folder of input images is selected for automatic sequential processing
      - Output parameters for perspective shifts, refocusing, etc. are defined in the script beforehand
      - Following calibration, program requires no user input and runs automatically through all the images in the folder.

Example Workflow

100 calibration images

raw experimental images



- Perspective shifts
  - Variety of image formats
- Refocused images
  - Variety of image formats
- Perspective sweep animation
  - GIF or video file
- Refocusing animation
  - GIF or video file
- Focal stack
  - Output a series of refocused images

• One also needs to determine the magnification of the images. An easy way to do this is by capturing an image of a ruler.

• If using the Auburn Light Field Analyzer (ALFA), a white image is needed as well. However, LFIT v2 does not require a white image at this time.

Using the Single Image (GUI) mode for processing plenoptic data

QUICK START GUIDE

Installation

- Obtain latest LFITv2 zip file and extract the contents
- Option #1:
  - Run the SETUP.m file to automatically uninstall any old versions of the toolkit and install the new toolkit
- Option #2:
  - Delete any existing LFITv2 folders inside the MATLAB user directory*
  - Copy the LFITv2 folder from the zip file to the MATLAB user directory*

*MATLAB user directory is typically C:\Users\<username>\Documents\MATLAB. To check, type userpath at the MATLAB command line.

Program File Structure

LFITv2_22_main.m

-Main program that invokes the GUI interface(s)
-Also contains the batch processing script section (user-editable)

-The user is encouraged to save new versions of this script when creating batch files (see later section). Think of this initial version as a template. The script name does not matter for this file.

LFITv2

-Contains all the function files that compose the toolkit
-Placed in MATLAB user directory

-To modify functions, user may create a copy of LFITv2 in the same folder as the main program. The program will use the functions from the local /LFITv2/ folder instead.

toolkitpathv2.m

-Required function that locates LFITv2 folder

SETUP.m

-Optional script used to install the locally extracted version of LFITv2

samplePath.txt

-Optional: An example custom perspective sweep path

Required GUI components

LFITv2_GUI_SinglePanel.fig
LFITv2_GUI_Prerun.fig

LFITv2_GUI_SinglePanel.m
LFITv2_GUI_Prerun.m

LFITv2_Documentation.pdf

header.png
refocusRef.png

Optional GUI components

lastrun.cfg

lastGUI.gcfg

- These are config files saved after each run to remember last used settings.

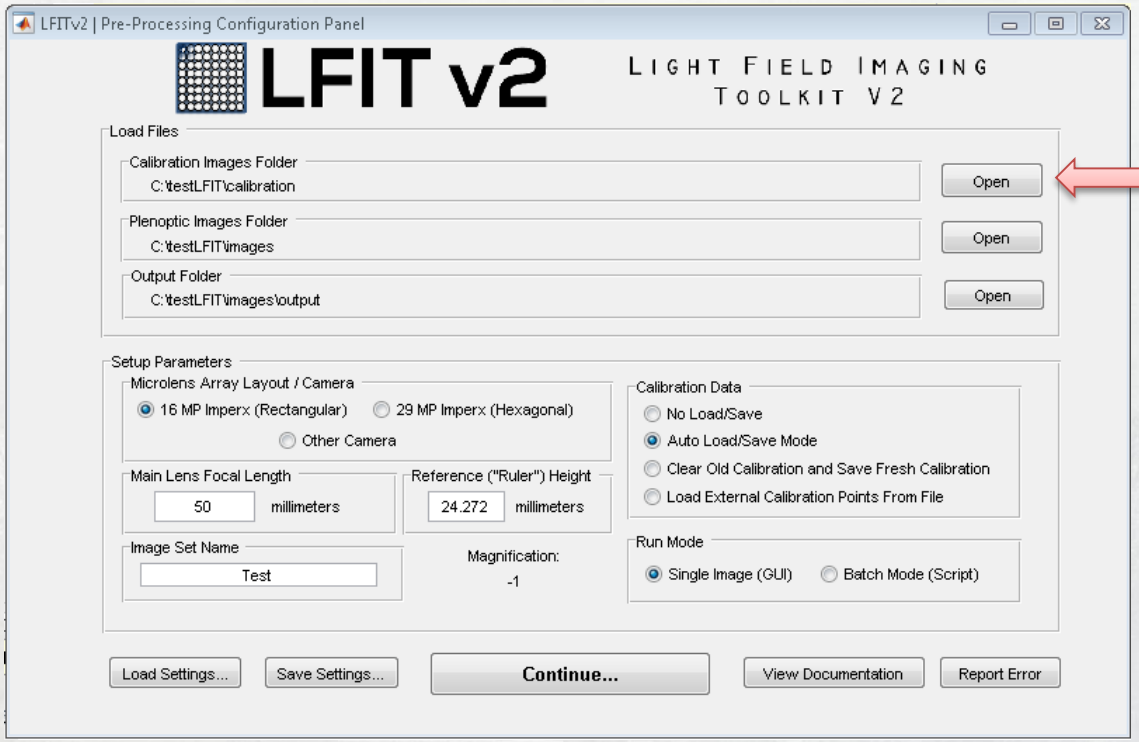
All files shown above should be located in the same folder, except for the LFITv2 function folder which should be located in MATLAB's userpath.

Running LFITv2

- User should have the calibration images (to be averaged) in one folder and the image(s) to be processed in a separate folder
- Open the main program file (nominally LFITv2_22_main.m) in MATLAB.
  - If running in Batch Mode, user will need to edit this file to configure batch processing (see later section).
- Run the script to open the pre-processing configuration panel.
  - This interface panel is used to set up file paths and other basic parameters

Pre-Processing Configuration Panel

For a given camera setup in an experiment, the aperture should be stopped all the way down (ie f/16 or f/22) and typically 100 TIFF images taken of a white surface. These images record the center of the each microlens—a key piece of information used in LFITv2 as well as ALFA.



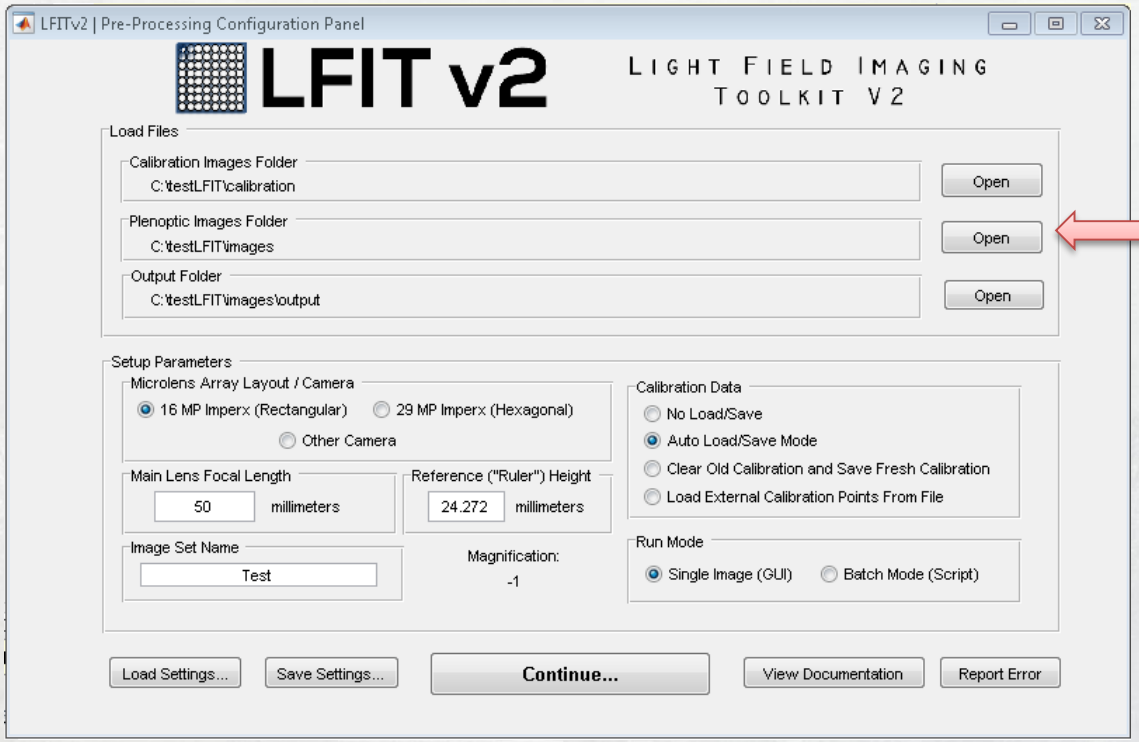
Select the calibration folder from the pop-up dialog.

- #1) User opens the folder of calibration TIFF images.

Pre-Processing Configuration Panel

This folder parameter is more important for the batch processing mode, as batch mode will cycle through all images in the folder.

For the single image GUI mode, the user can select any image (within or without this particular folder).



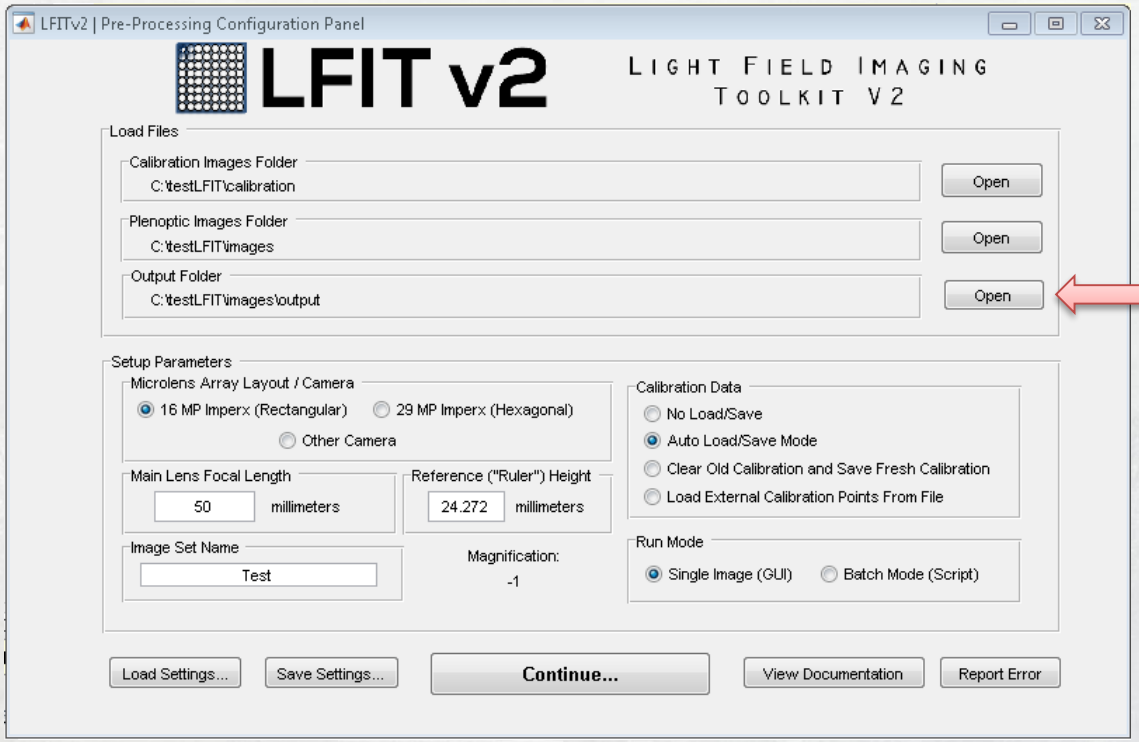
Select the experimental images folder from the pop-up dialog.

- #2) User opens the folder of the raw plenoptic image(s) to be processed.

Pre-Processing Configuration Panel

Typically, the user will want to create a new folder to contain the outputted images. This can be done from the Open prompt in the GUI.

If unsure what to choose, a common place to output exported images is to a subfolder within the plenoptic images directory (ie /Output/).

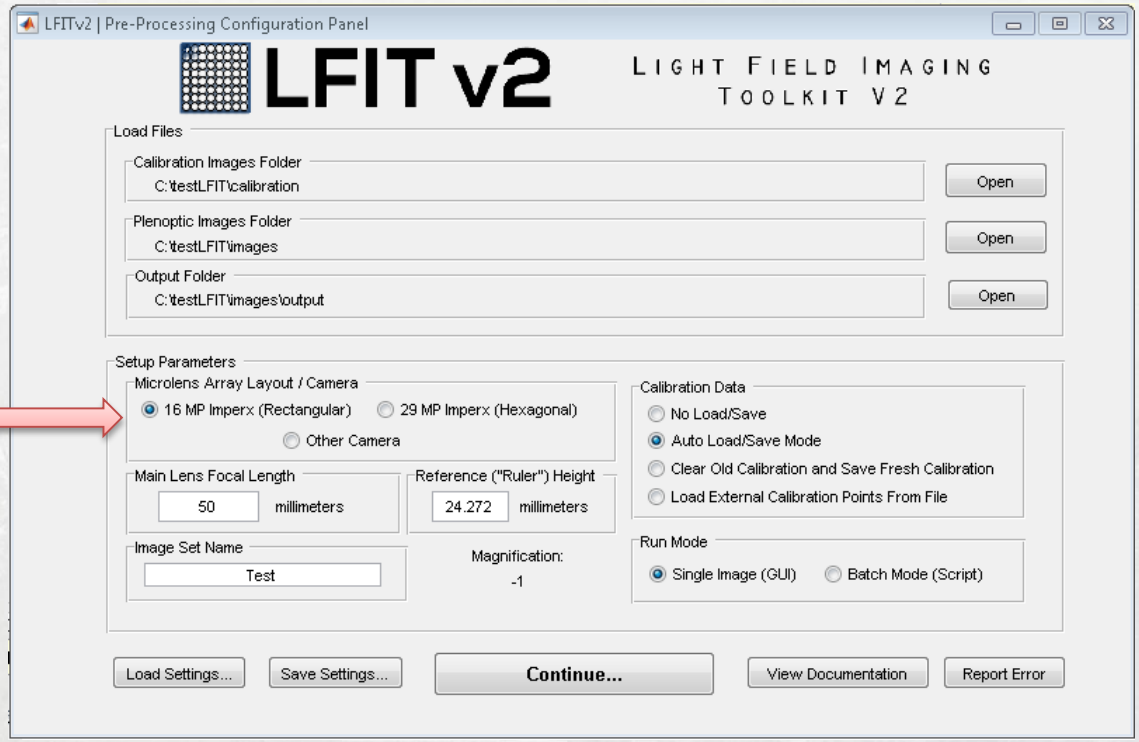


Select the desired output folder for any exported images.

- #3) User selects a folder for exported images to be outputted to.

Pre-Processing Configuration Panel

Select the camera used in the experiment.



Select whichever camera was used to capture the plenoptic images to be processed.

The 29 MP Imperx has a hexagonally arranged microlens array.

The 16 MP Imperx has a rectilinear grid of microlenses.

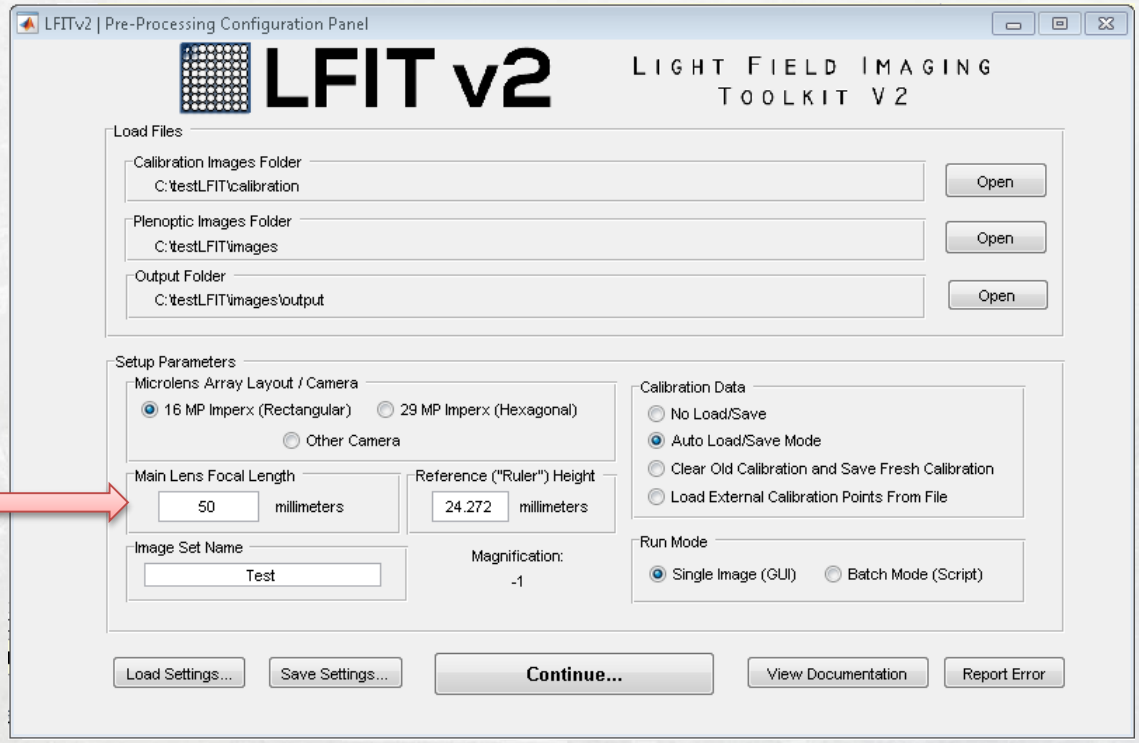
The other camera option will prompt the user to input parameters of the camera used.

- #4) User selects the camera used to capture the plenoptic images.

Pre-Processing Configuration Panel

The main lens focal length entered here is just the focal length of the main lens. Do not include extension tubes.

Enter the focal length (in millimeters) of the main lens.



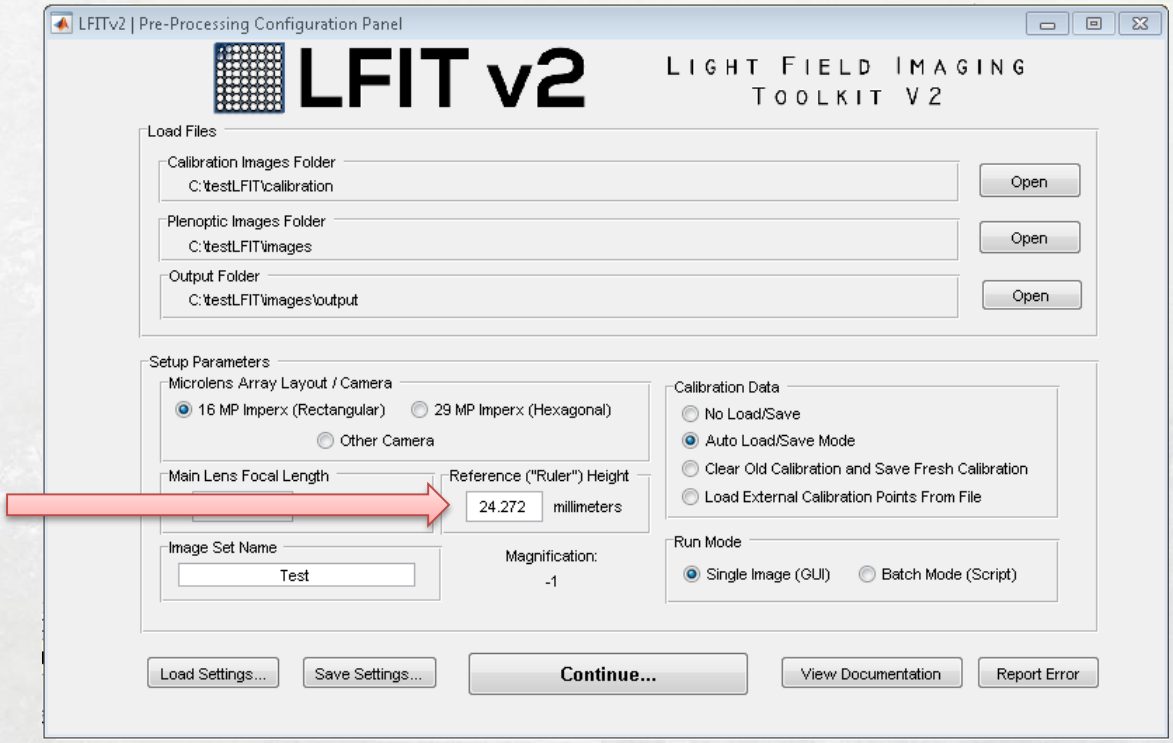
Typical focal lengths might be 50 mm, 100 mm, 250 mm, etc. This will be indicated on the lens itself.

Prime lenses have a fixed focal length and are recommended for experiments. Telescopic lenses are discouraged as it is often unclear what the exact focal length of the lens is when set at some intermediate zoom.

- #5) User enters the focal length of the main lens.

Pre-Processing Configuration Panel

Enter the height (in mm) of a ruler placed at the nominal focal plane of the given experimental setup.

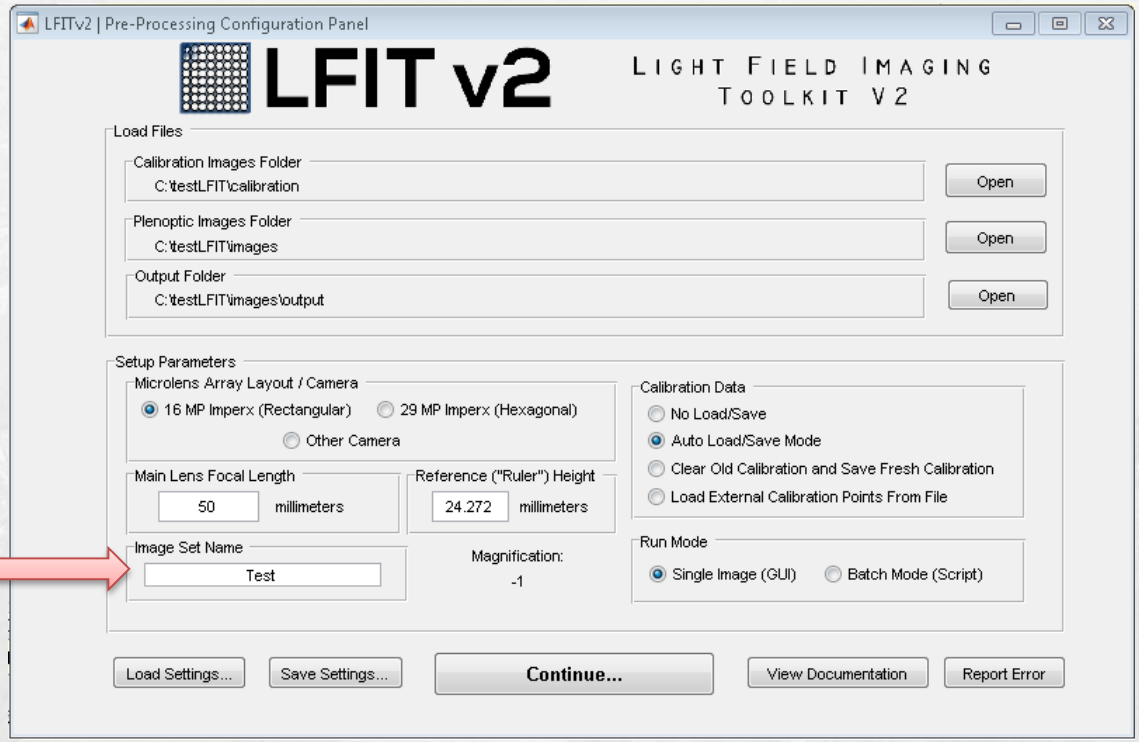


For a given experimental setup, place a ruler at the nominal focal plane of the camera setup. Ideally it will fill the image frame. If it doesn't, the user will need to figure out how tall it would be if it did fill the entire frame. Record the height in mm and enter it here.

The magnification text below will update depending on the value you enter.

- #6) User enters the reference ("ruler") height to determine the magnification.

Pre-Processing Configuration Panel



Enter a name for the given experimental image set.

This Image Set Name is used as a prefix on exported files, so keep it of reasonable length.

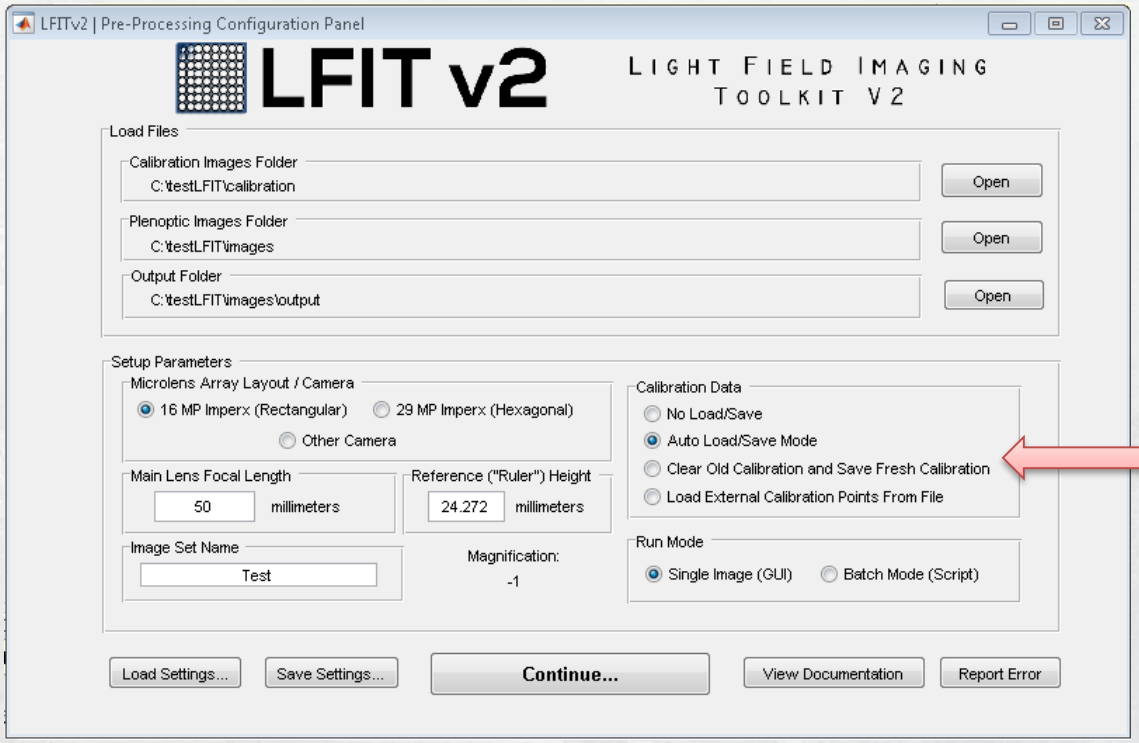
Also, this name is used to label calibration data saved if using the calibration save options (next slide).

- #7) User enters a name for the set of plenoptic images to be processed.

Pre-Processing Configuration Panel

Auto Load/Save Mode will attempt to load calibration data in the Calibration Images folder defined above that is labeled with the Image Set Name. Otherwise, it recomputes the calibration and saves it with the Image Set Name in the Calibration Images folder.

Clear Old/Save Fresh will attempt to delete saved calibration data associated with the Image Set Name in the Calibration Images folder. It then recomputes and saves calibration data as above.



No Load/Save always recomputes the calibration data every time LFIT is run.

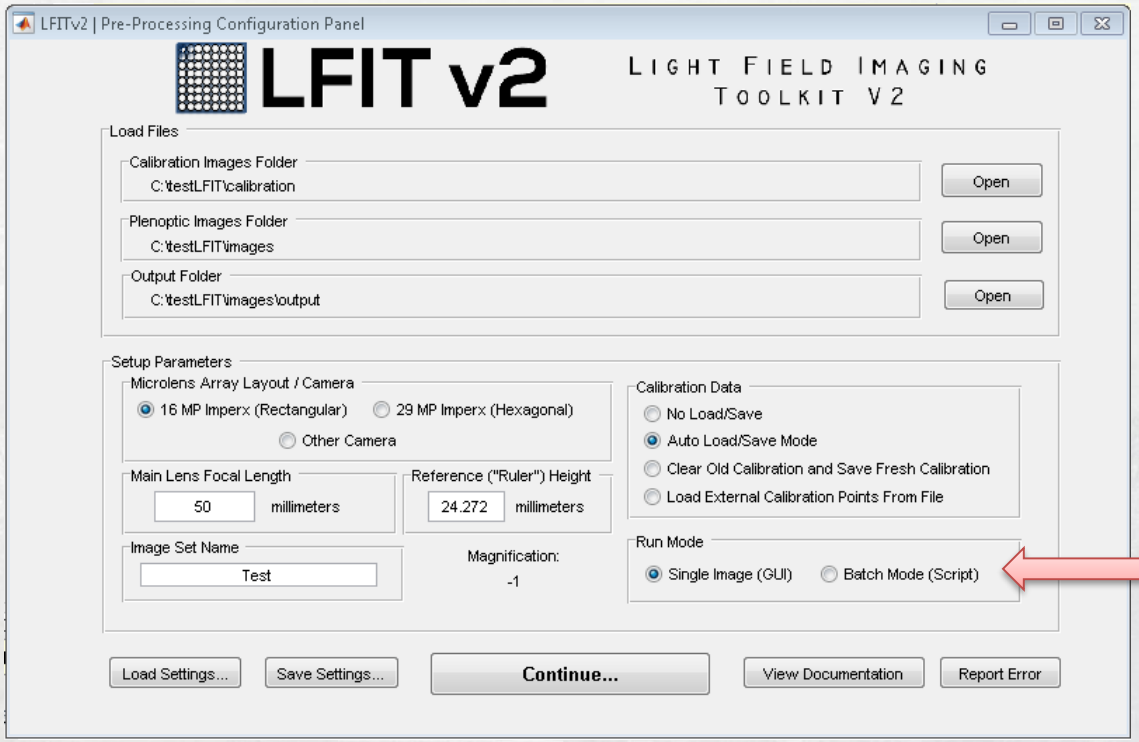
Select a calibration option from the button group.

- #8) User selects a calibration data save/load mode option.

Pre-Processing Configuration Panel

Single Image (GUI) mode presents a graphic user interface with buttons and checkboxes etc. to facilitate quick processing of individual plenoptic images.

Batch (Script) mode executes the script found in the bottom portion of the LFITv2_00_main.m (or equivalent) file. This allows for processing of many plenoptic images in an automatic, prescribed fashion. See later section on Batch Mode.



No Load/Save always recomputes the calibration data every time LFIT is run.

Select a run mode option from the button group.

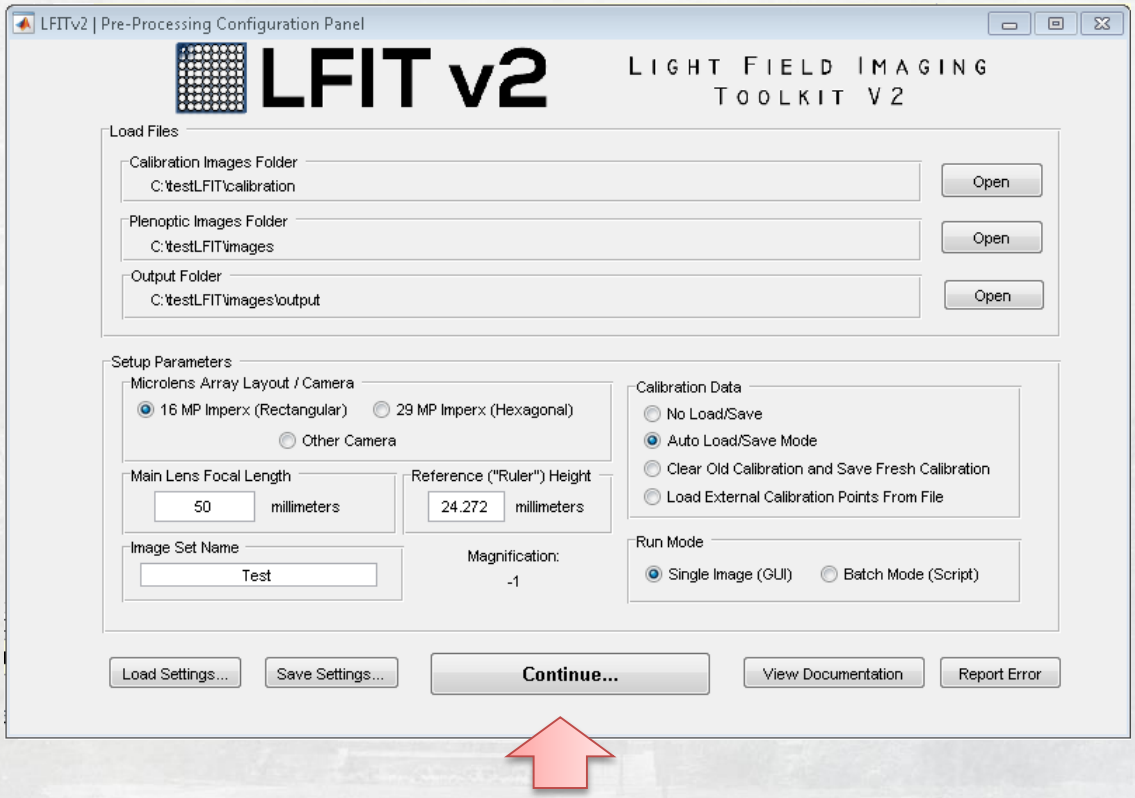
- #9) User selects a run mode for the program.

Pre-Processing Configuration Panel

Load Settings and Save Settings permit loading and saving of the parameters defined in this interface panel.

Note that the panel will automatically 'remember' the settings from the previous run.

(The program attempts to load the previous run; if no saved run can be loaded, it simply fills the panel out with defaults.)



View Documentation opens a help file in the PDF viewer installed on the local machine.

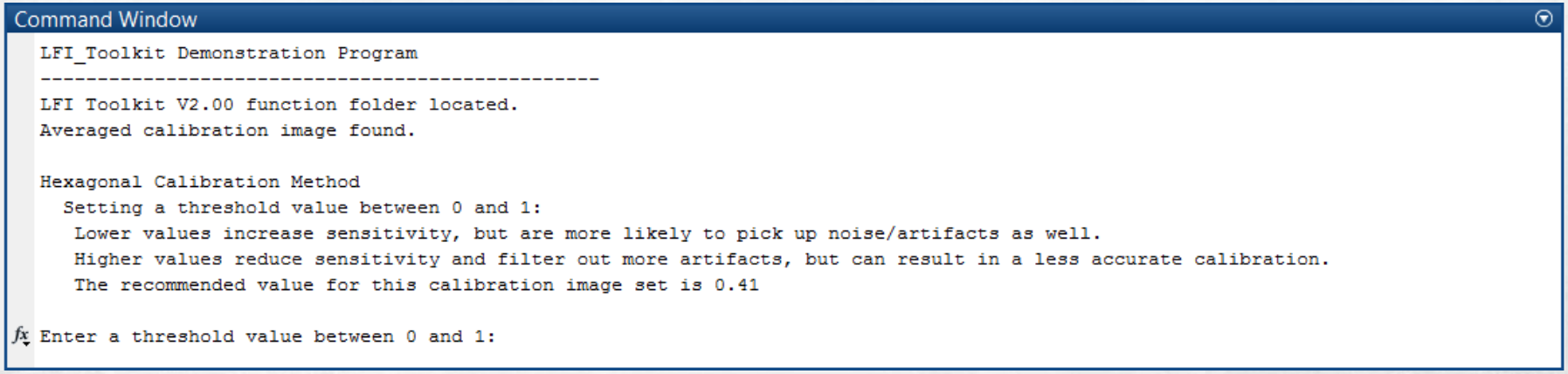
Report Error opens the default email client with a new message addressed to the LFIT developers.

- #9) User presses Continue... to begin.

Single Image (GUI) Mode

- The remainder of this Quick Start Guide follows the case of Single Image (GUI) mode with the 29 MP Imperx (hexagonal) camera.
  - There are some slight differences for the 16 MP Imperx (rectangular) camera process, but overall the steps are the same.
  - See later information on Batch Mode processing.
- Also, now begin watching the MATLAB command line output.
  - Warnings, prompts for user input, and estimated wait times during processing steps all may be displayed.

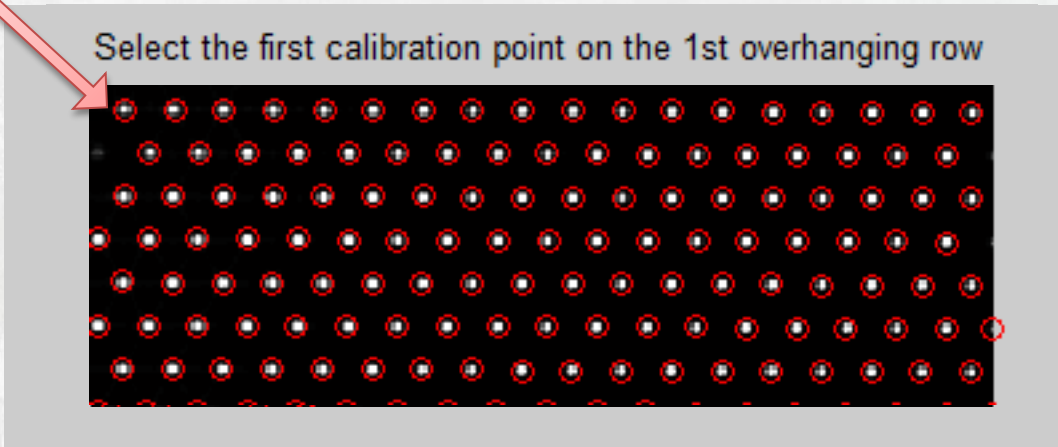
Hexagonal Calibration



- If no calibration data was loaded, the command line output should look similar to above.
- The user is now prompted for a calibration threshold value.
  - A recommended value is dynamically calculated for the averaged calibration image in the above prompt, but the user may experiment with different values if there is difficultly in getting an accurate calibration.
  - Type the desired value and press <Enter>.

Hexagonal Calibration

#1



The initial point selection is not critical.

This calibration algorithm assumes that the first point is an 'overhanging' or 'outset' row, regardless of whether or not the row is inset/indented or not.

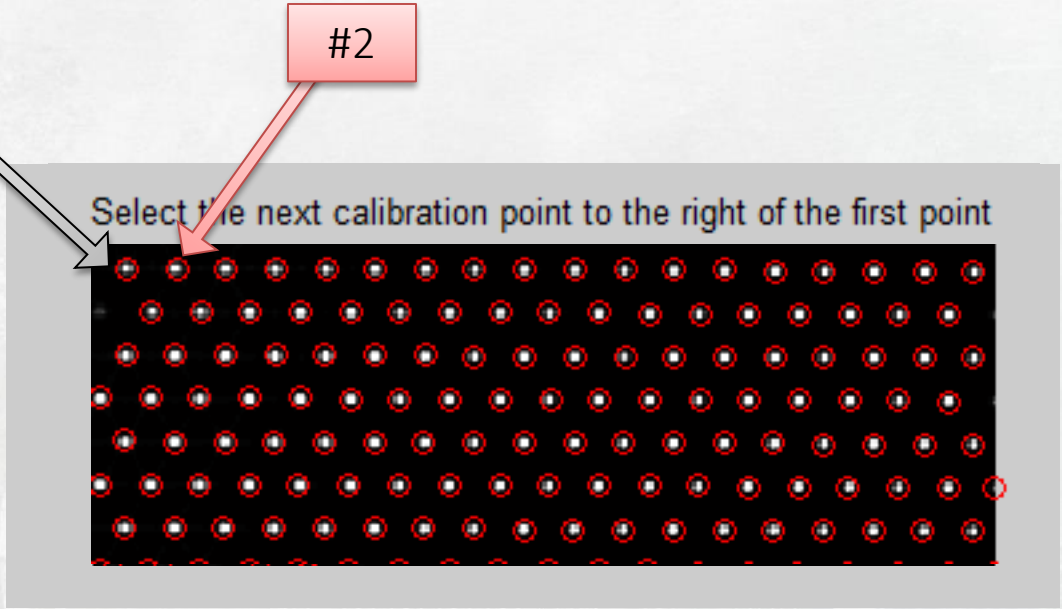
If needed, the algorithm crops out a few rows at the top and/or columns from the left to prevent the selection of microlenses on disappearing rows.

While it's acceptable if a handful of microlenses are not circled, ideally nearly all the microlenses will be circled (as at left).

If an appreciable portion of the microlenses did not get circled or if many extra red circles appear, press <Ctrl>+<c> to end program execution. Rerun the program and try a different sensitivity.

- #1) Click an initial microlens center point in the displayed array of circled points.

Hexagonal Calibration



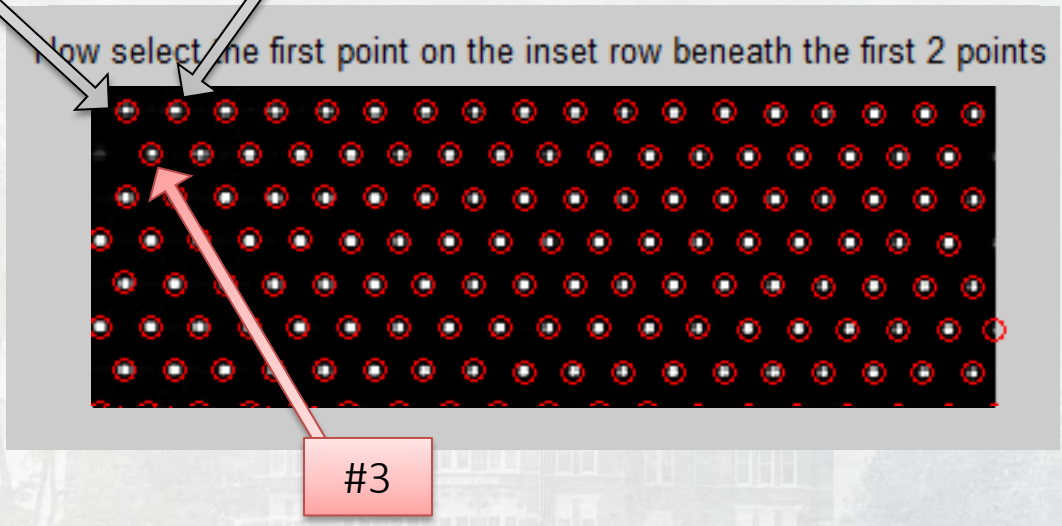
The second point MUST be immediately to the right of the first point selection, on the same row.

The literal click does not have to be 100% accurate. The algorithm looks at the (x,y) positions of the user's clicks then matches the clicked (x,y) points to the exact (x,y) locations of the microlenses as identified by the red circles.

This is why it is critical that the initial microlenses the user plans on selecting are circled. If they are not, the program will match the user's clicks to incorrect microlenses.

- #2) Click the microlens center point immediately to the right of the first selection.

Hexagonal Calibration



The third point MUST be on the row immediately beneath the initial row, and located directly between the first two points.

The literal click does not have to be 100% accurate. The algorithm looks at the (x,y) positions of the user's clicks then matches the clicked (x,y) points to the exact (x,y) locations of the microlenses as identified by the red circles.

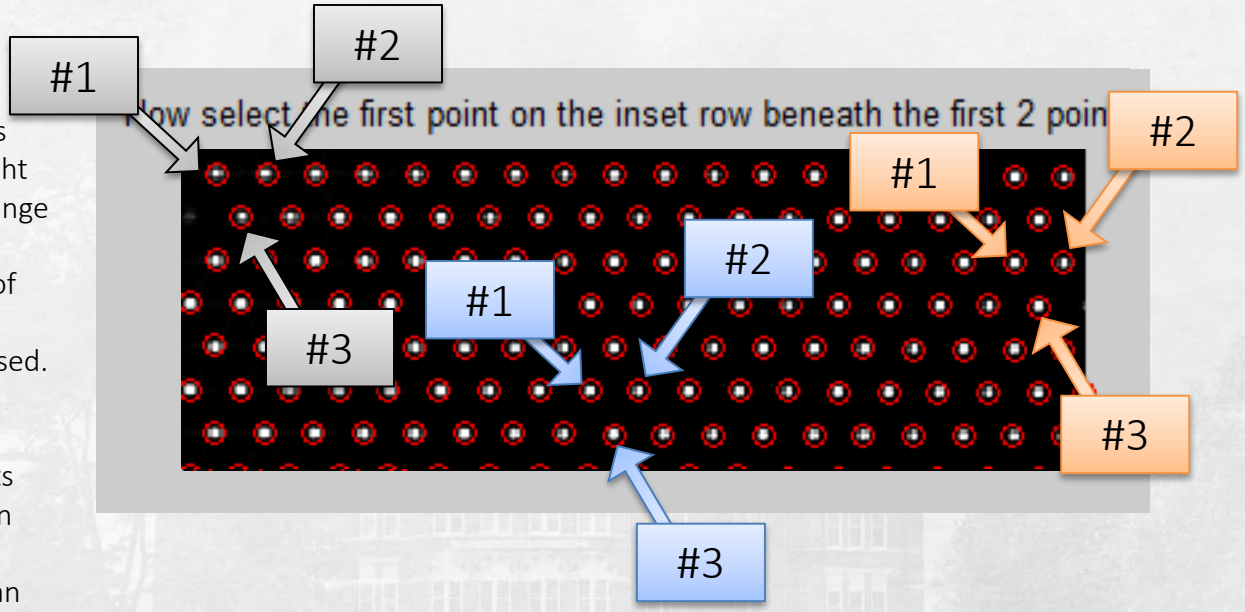
This is why it is critical that the initial microlenses the user plans on selecting are circled. If they are not, the program will match the user's clicks to incorrect microlenses.

- #3) Click the microlens center point on the next row that is directly between the first two selected microlenses.

Hexagonal Calibration Aside

Note that if one starts significantly to the right as in the blue and orange cases at right, an appreciable amount of the image will be cropped out and unused.

It's generally best to select the three points near the top left, in an area of well-circled microlenses (that is, an area without significant image artifacts or noise).

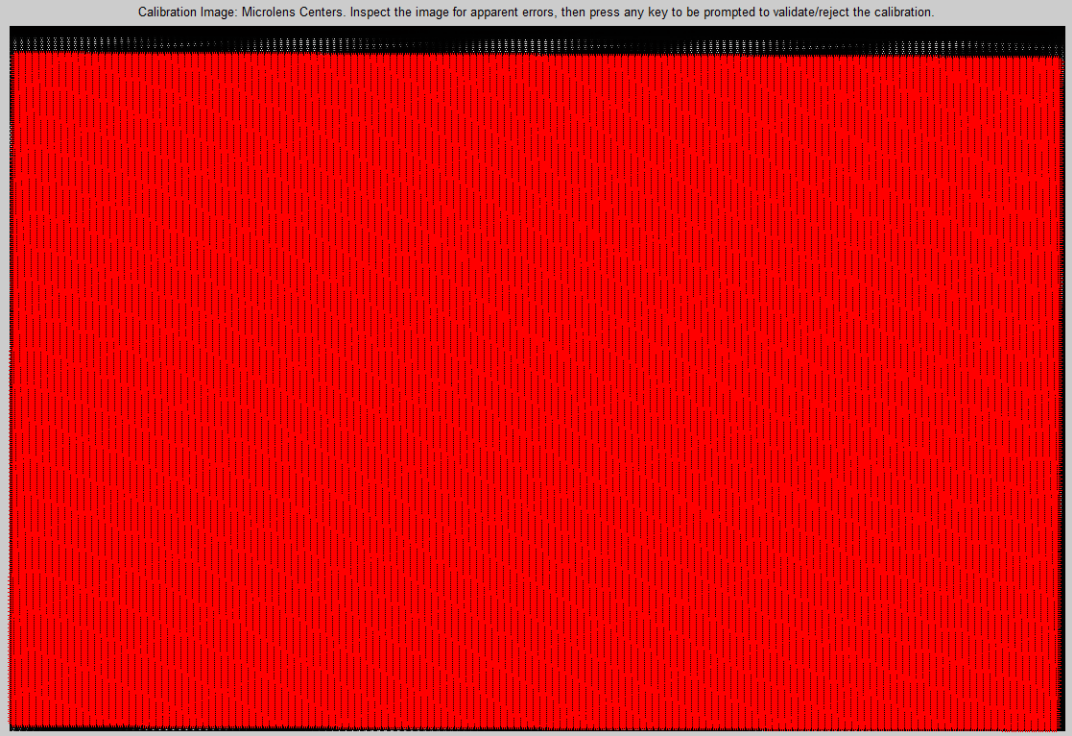


- Note that the starting point can vary (as shown by the 3 colors representing 3 different valid cases). What matters is that the exact pattern is followed as shown above.

Hexagonal Calibration

This is an example of a good calibration, excepting the fact that the top is excessively cropped.

This is due to our selecting of an initial point some distance from the top of the image to avoid some artifacts caused by a microlens array that was not completely centered over the image sensor.



The overall pattern looks consistent/uniform from this zoomed out perspective.

The main thing to watch for would be skipped rows or columns; these errors would manifest themselves with an obvious break in the overall pattern or with a distinct repeating pattern. It may be necessary to zoom in to see these abnormalities, however.

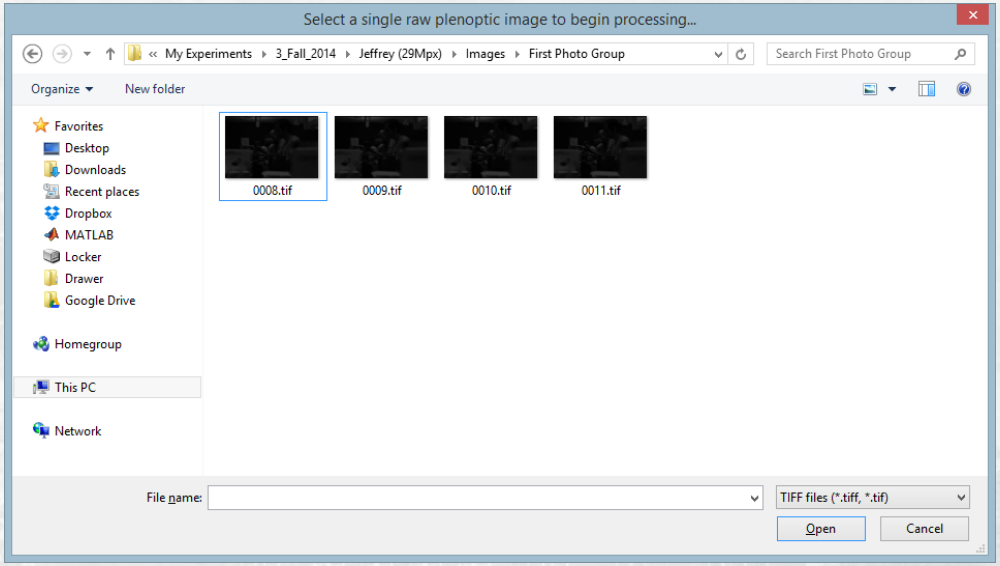
Such errors would also be apparent in outputted images, particularly in a ruler image.

- Following completion of the calibration algorithm (observe command line output for estimated wait time), inspect the resulting calibration for obvious errors, then press any key.
- If the calibration looks good, type Y at the command line. Otherwise type N. Then press <Enter>.

Image Selection

The dialog opens by default in the chosen plenoptic images directory previously defined in the pre-processing interface at the beginning of the program.

After selecting an image, the program interpolates the image data behind each microlens (i.e., the microimage data) onto a uniform (u,v) grid. This hexagonally arranged data is then resampled onto a rectilinear grid.

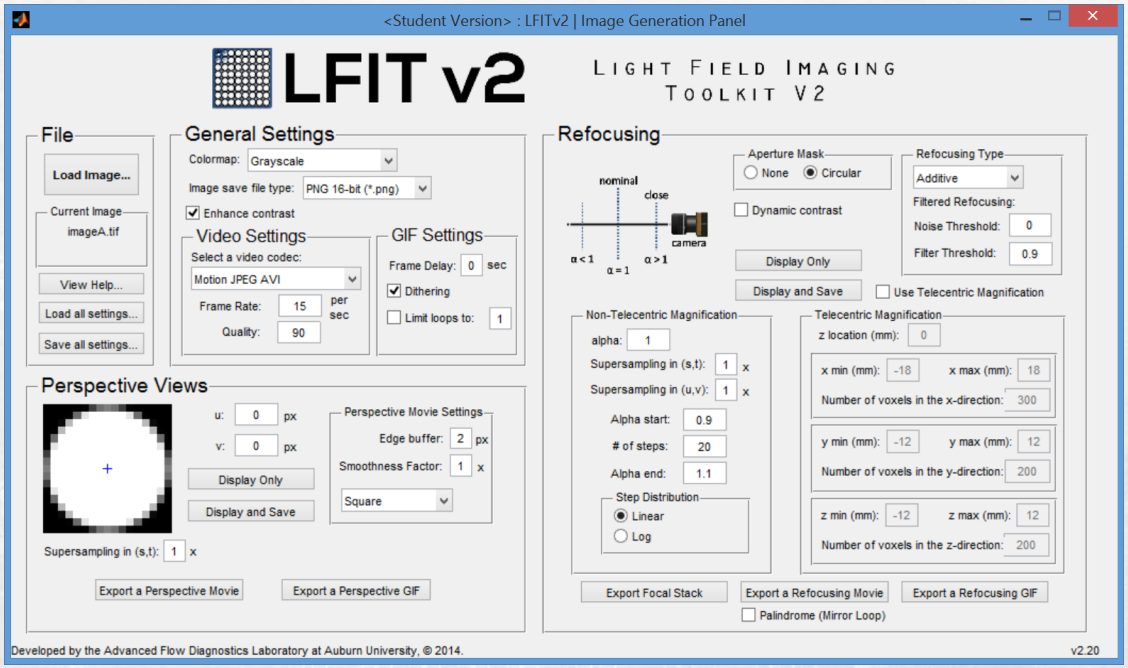


If the user selects an image from a different folder than defined at the start, the program will immediately prompt the user for a new output folder.

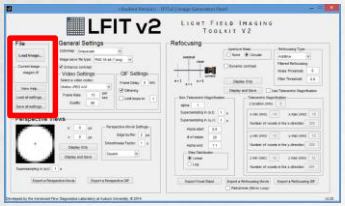
If the user cancels this dialog, the program automatically creates an output folder (/Output/) in the same directory as the selected image. The exact path is then shown in the command line output.

- Select a single raw plenoptic image to process from the dialog, then wait for the interpolation and resampling processes to complete.
  - Observe the command line output for estimated wait times.

Image Generation Panel



- The main image generation interface panel opens after the interpolation and resampling is complete.
- While it may look intimidating, this guide will break it down section-by-section to explain the contents.
  - Also note that every box in the interface has mouse over ("tooltip") text that provides help.



File Load/Save

Press to load and process a new raw plenoptic image.

Note: This image must still be associated with the same calibration set as used earlier.

Watch the command line output for estimated wait time as it will take some time to process a new image. Refrain from adjusting the interface settings while waiting.

File

Load Image...

Current Image

0008.tif

Displays the currently loaded image file name.

Click to open the help file in the local PDF viewer

View Help...

Loads all settings from a file.

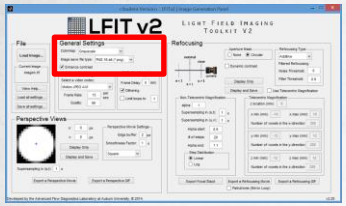
Note that the settings are automatically loaded from the previous run when the interface window initially opens. If this auto load fails, defaults are used instead.

Load all settings...

Saves all settings in the currently displayed interface to a file.
Note that the currently loaded/processed image data is NOT saved via this option. Also note that the settings are automatically saved when closing the interface figure.

Save all settings...

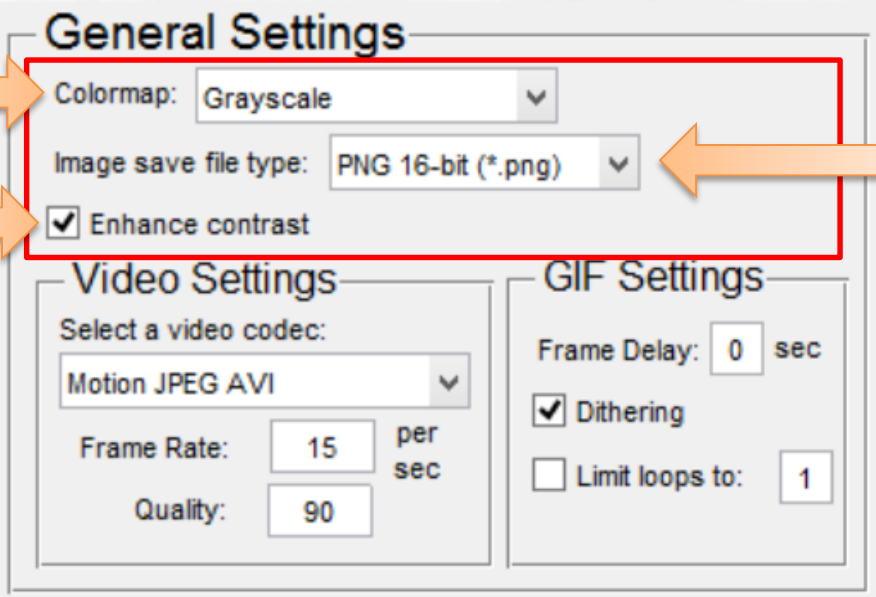
- This section allows for loading/processing a new plenoptic image, viewing help, and saving/loading settings files (only necessary if you would like to load a new image).



General Settings

Built-in MATLAB colormaps

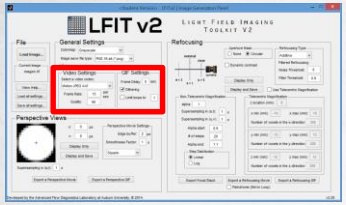
Check to apply imadjust function
(saturates top and bottom 1% of pixel intensities to increase contrast)



Select a file type to be used in perspective shifts/refocused exports:
BMP
PNG
JPEG
16-bit PNG
16-bit TIFF

(Grayscale colormap recommended when using 16-bit options.)

- This section allows for the selection of image settings that apply to all exported files.



Movie Settings

Video Compression Codec:

MATLAB R2010a or earlier:
Uncompressed, MSVC, RLE, Cinepak

MATLAB R2010b or newer:
Uncompressed, Motion JPEG AVI, Lossless Motion JPEG 2000, Compressed Motion JPEG 2000,MP4 (H.264)

Integer number.
Typical frame rates:
15=minimum for motion,
24=cinema,
29.97=NTSC,
30=typical,
60=display refresh rate.

Integer between 0 and 100

Check to limit the number of loops the GIF completes, otherwise it will continue looping indefinitely.

Integer number.

Indicates how many loops of the GIF should animate before stopping, if the left check box is checked.

Integer number.

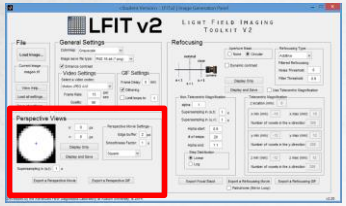
Typically 0, but can be a non-zero value to create pauses between discrete frames.

If checked, program will dither if necessary to achieve better color resolution at the expense of spatial resolution.

Otherwise each color in the original image is mapped to the closest color in the new map.

From Wikipedia: "Dither is an intentionally applied form of noise used to randomize quantization error, preventing large-scale patterns such as color banding in images."

- This section allows for the selection of video settings that apply to all exported files.



Perspective Views

Plots the currently entered (u,v) coordinates on a simulated microimage of a circular aperture.

Value (in pixels) between:
Rectangular: -8 and +8
Hexagonal: -7 and +7
(decimal values are supported)

Perspective Views

u: p

v: 0 p

Perspective Movie Settings

Edge buffer: 2 px

Smoothness Factor:

Display Only

Display and Save

Supersampling in (s,t): 1 x

Export a Perspective Movie

Export a Perspective GIF

Computes and displays a perspective shifted image in a new figure.

Computes, displays, and save perspective shifted image in previously defined output folder.

Integer value of 1 or greater
1= no supersampling
2 = 2x supersampling
4 = 4x supersampling, etc…

Also, the 29 MP hexagonal images are already supersampled in (s,t) during the resampling process.

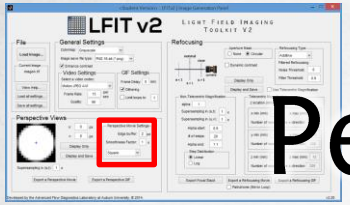
Computes, displays frame-by-frame, and saves a perspective movie in previously defined output folder.

Watch command line output for estimated wait time.

Computes, displays frame-by-frame, and saves a perspective GIF in previously defined output folder.

Watch command line output for estimated wait time.

- This section permits the generation of perspective shifts of the loaded image.



Perspective Movie Settings

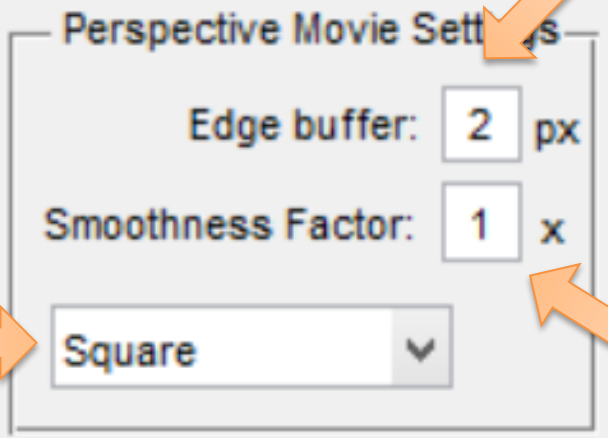
Controls the path of the perspective sweep animation.

Square = Follows a box path in a clockwise direction about (u,v) = (0,0). *Note that without an edge buffer, this path will go off the circular aperture.*

Circle = Follows a clockwise circular path with radius equal to the microimage radius minus any edge buffer.
Note that the circular path evaluates many non-integer (u,v) values which take slightly longer to compute than integer values as in Square or Cross.

Cross = Moves in a cross (or plus + sign) path across the aperture plane.
Order: Left-Right-Center-Bottom-Top-Center-Left

Path from File... = Will prompt user for a text file containing a list of [u v] coordinates upon clicking to export either a movie or GIF.
See samplePath.txt. The format is:
u value ‹space› v value
with line breaks to indicate new points.

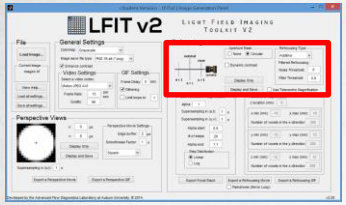


Value between:
Rectangular: 0 and +7
Hexagonal: 0 and +6
This is a buffer from the edges of the microimages when moving about in the aperture plane during perspective sweep movie generation. The data tends to degrade near the edges, so this is a way to buffer it out. 2 is a typical choice.

Integer value of 1 or greater
1= default smoothness
2 = 2x smoothness
4 = 4x smoothness

This makes the step size between evaluated (u,v) values finer/smaller, increasing the total number of frames. This does not apply to Path from File...

- This section sets the parameters for perspective movie exports (video and GIF).



Refocusing

None: The full 15x15 or 17x17 grid of extracted image data behind each microlens is used in refocusing. (May pick up adjacent microlens data.)

Circular: A circular mask is applied such that the corners of the microimages are masked out. Circular should be selected in most cases when using a hexagonal camera array. (This prevents picking up data from neighboring microlenses in the tightly packed hexagonal array.)

Selects additive, filtered, or multiplicative refocusing

Check to normalize contrast limits on a per slice (per frame) basis (Otherwise, output intensity limits will be set by the max/min intensities of the entire refocused stack as in focal stack generation.)

Refocusing

Aperture Mask
None Circular

Refocusing Type
Additive

Filtered Refocusing:
Noise Threshold: 0
Filter Threshold: 0.9

Dynamic contrast

Display Only

Display and Save

Use Constant Magnification

nominal close camera α < 1 α = 1 α > 1

Selects the noise threshold for filtered refocusing

Displays a handy quick reference figure for the relationship between the depth parameter alpha and refocused focal planes.

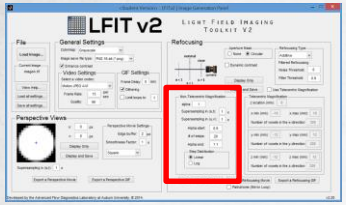
Selects the filter threshold for filtered refocusing

Computes and displays a refocused image in a new figure.

Computes, displays, and saves a refocused image in previously defined output folder.

Enables telecentric magnification settings

- This section permits the selection of refocusing settings and generation of refocused views of the loaded image.



Legacy Refocusing

Value greater than 0
(decimal values supported)
α>1 focuses closer
α=1 focuses at nominal focal plane
α<1 focuses further

Value greater than 0
(decimal values supported)
α>1 focuses closer
α=1 focuses at nominal focal plane
α<1 focuses further

Value greater than 0
(decimal values supported)
α>1 focuses closer
α=1 focuses at nominal focal plane
α<1 focuses further

Legacy Magnification
alpha: 1
Supersampling in (s,t): 1 x
Supersampling in (u,v): 1 x
Alpha start: 0.9
# of steps: 20
Alpha end: 1.1
Step Distribution
Linear
Log

Integer value of 1 or greater
1= no supersampling
2 = 2x supersampling
4 = 4x supersampling, etc…
NOTE: (s,t) supersampling increases resolution. (u,v) supersampling significantly increases processing time, but can significantly improve results for very large or small values of alpha.

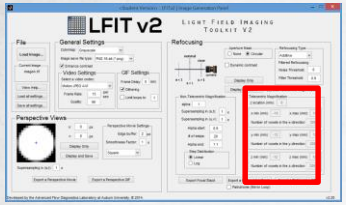
Also, the 29 MP hexagonal images are already supersampled in (s,t) during the resampling process.

Integer number of steps/frames in the refocused movie

Linear = Linearly spaced values of alpha.

Log = Slightly increases the number of alpha values below 1 and decreases the number of alpha values present above 1 via log spacing.
Follows formula:
=logspace(log10(alphaStart),log10(alphaEnd),steps)

- This section permits the selection of non telecentric refocusing settings.



Telecentric Refocusing

Selects the z-location for single image display/save

Constant Magnification

z location (mm): 0

x min (mm): -18 x max (mm): 18

Number of voxels in the x-direction: 300

y min (mm): -12 y max (mm): 12

Number of voxels in the y-direction: 200

z min (mm): -12 z max (mm): 12

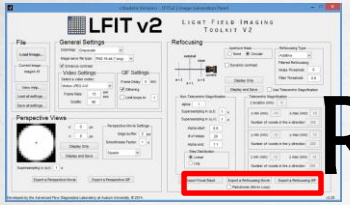
Number of voxels in the z-direction: 200

Selects minimum and maximum x values of the volume

Selects the number of voxels in the x direction

Selects y and z values similarly

- This section permits the selection of telecentric refocusing settings.



Refocusing Movie Export

Computes, displays frame-by-frame *(when complete)*, and saves a refocusing movie in previously defined output folder.

Watch command line output for estimated wait time.

Computes, displays frame-by-frame *(when complete)*, and saves a series of refocused images in previously defined output folder.

Watch command line output for estimated wait time.

Export Focal Stack | Export a Refocusing Movie | Export a Refocusing GIF

☐ Palindrome (Mirror Loop)

Computes, displays frame-by-frame *(when complete)*, and saves a refocusing GIF in previously defined output folder.

Watch command line output for estimated wait time.

Palindrome will mirror the alpha range, doubling the number of steps.

For the example values here:
When the ending alpha is reached at step 20, the alpha values are decreased back to the initial value for a total of 40 frames.

- This section permits the export of refocused movies.

For processing entire folders of data in an automated fashion

BATCH MODE

Batch Mode

- To automate LFIT, a batch mode setting can be enabled in the initial interface panel.
- LFIT will automatically process every raw plenoptic image in the designated folder according to the batch mode portion of the main program script.
- Open the main program file and save it under a new name.
  - Use the original as a template, and save it under different names as needed when making new batch processing section edits.

Batch Mode

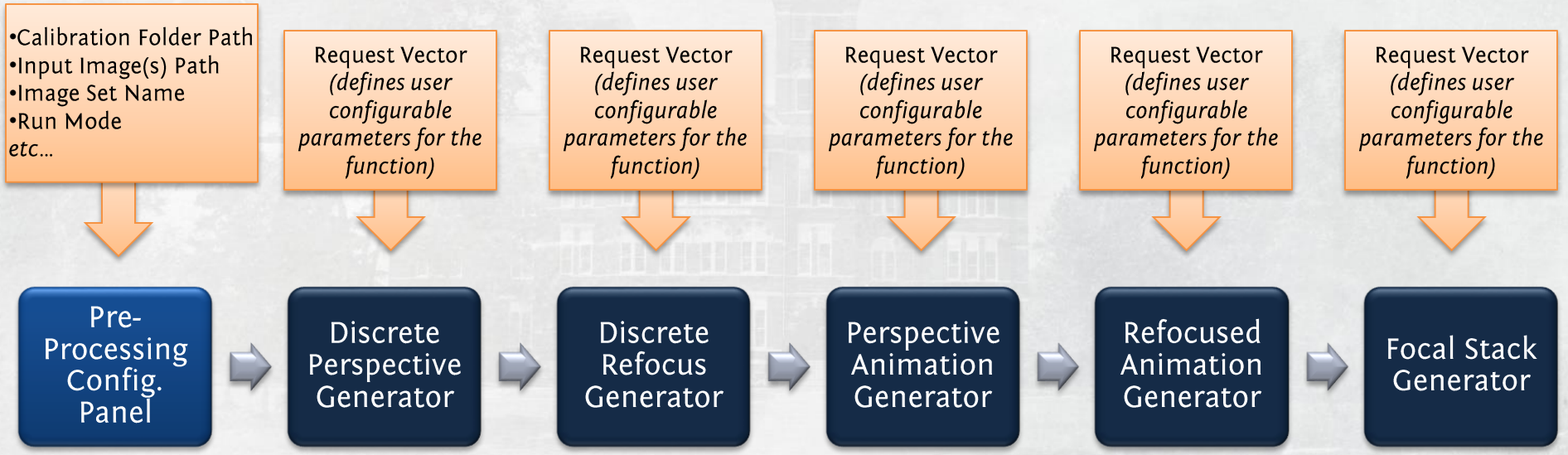
```
%%%%%%%%----------------------------------%%%%%%%%
%%%%%%%%---USER EDITS BEGIN BELOW HERE---%%%%%%%%
%%%%%%%%----------------------------------%%%%%%%%
```

- Scroll down in the file until the above section is found (around line 115)
- Between the above and the below are a series of request vectors and generation function entries.
- By editing the request vectors and commenting out unneeded generation functions, the user can set up a custom batch processing script.

```
%%%%%%%%----------------------------------%%%%%%%%
%%%%%%%%---USER EDITS END ABOVE HERE---%%%%%%%%
%%%%%%%%----------------------------------%%%%%%%%
```

Batch Mode

- Below is a graphical overview of the different generation functions.



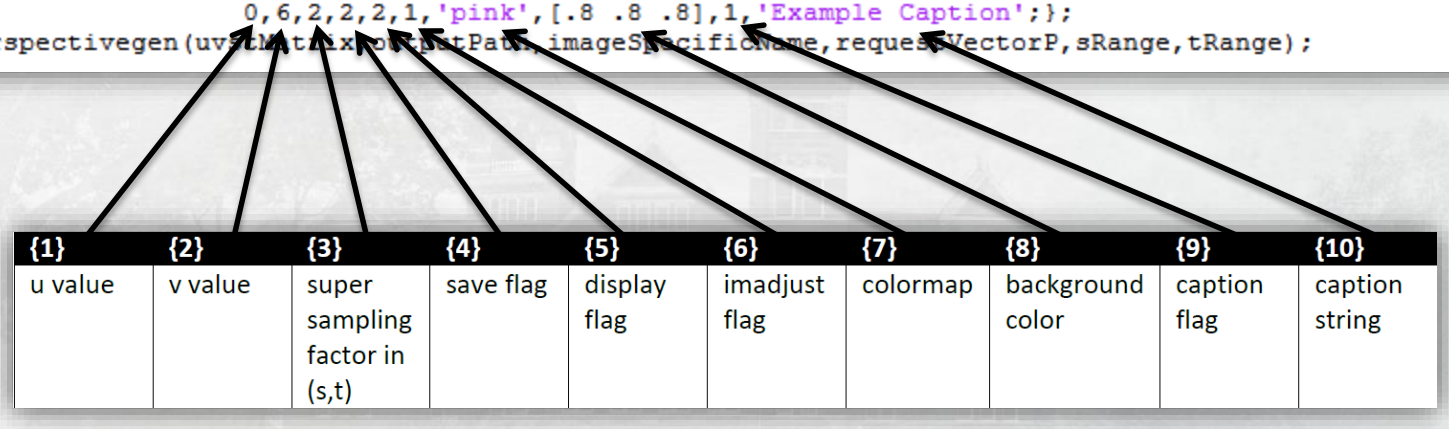
Setting Generator Parameters

```
%--------------------------------------
% Perspective Shift
%--------------------------------------
% (See comprehensive requestVector documentation in perspectiveGenerator.m)
%[u,v,SS_ST,saveFlag,displayFlag,imadjustFlag,colormap,backgroundColor,captionFlag,'A caption string'];
requestVectorP = {0,0,1,1,2,1,'jet',[.8 .8 .8],0,'No caption';
                  0,-6,4,4,2,1,'gray',[.8 .8 .8],0,'No caption';
                   0,6,2,2,2,1,'pink',[.8 .8 .8],1,'Example Caption';};
perspectivegen(uvstMatrix,outputPath,imageSpecificName,requestVectorP,sRange,tRange);
```

- The generator functions are all controlled by requestVector variables.
- The requestVector is a cell array that defines the parameters for the generator function.
- Each line in the requestVector corresponds to another output configuration (that is exported image) for the given input plenoptic image.
  - The sample requestVector above will generate 3 different output images from a single input image.
  - Each image in the folder will be processed in the three configurations in this example above; that is to say, there will be 3 exported images for every raw plenoptic image in the defined folder.

Setting Generator Parameters

```
%--------------------------------------
% Perspective Shift
%--------------------------------------
% (See comprehensive requestVector documentation in perspectiveGenerator.m)
%[u,v,SS_ST,saveFlag,displayFlag,imadjustFlag,colormap,backgroundColor,captionFlag,'A caption string'];
requestVectorP = {0,0,1,1,2,1,'jet',[.8 .8 .8],0,'No caption';
                  0,-6,4,4,2,1,'gray',[.8 .8 .8],0,'No caption';
                   0,6,2,2,2,1,'pink',[.8 .8 .8],1,'Example Caption';};
perspectivegen(uvSTMatrixOutputPath,imageSpecificName,requestVectorP,sRange,tRange);
```



{1}	{2}	{3}	{4}	{5}	{6}	{7}	{8}	{9}	{10}
u value	v value	super sampling factor in (s,t)	save flag	display flag	imadjust flag	colormap	background color	caption flag	caption string

- Each comma separated variable in a requestVector corresponds to a specific parameter.
  - The above example is for the perspective generator function
    - Other generator functions have slightly different requestVector layouts

Discrete Perspective Generator

(perspectivegen.m)

{1}	{2}	{3}	{4}	{5}	{6}	{7}	{8}	{9}	{10}
u value	v value	super sampling factor in (s,t)	save flag	display flag	imadjust flag	colormap	background color	caption flag	caption string

Value between:
Rectangular: -8 and +8
Hexagonal: -7 and +7
(decimal values supported)

Value between:
Rectangular: -8 and +8
Hexagonal: -7 and +7
(decimal values supported)

Integer value of 1 or greater
1= no supersampling
2 = 2x supersampling
4 = 4x supersampling, etc…

Pick integer from below:
- 0 for no saving
- 1 for saving a BMP
- 2 for saving a PNG
- 3 for saving a JPEG
- 4 for a 16-bit PNG
- 5 for a 16-bit TIFF

(Options 4 and 5 do not support captions)
('gray' colormap recommended for 4 and 5)

Pick integer from below:
- 0 for no display
- 1 to display with a pause
- 2 to display without a pause

Pick integer from below:
- 0 for raw output
- 1 is to apply imadjust function

(saturates top and bottom 1% of pixel intensities to increase contrast)

MATLAB colormap name (e.g., 'gray' or 'jet' or 'pink')
(include apostrophes to designate as a string)

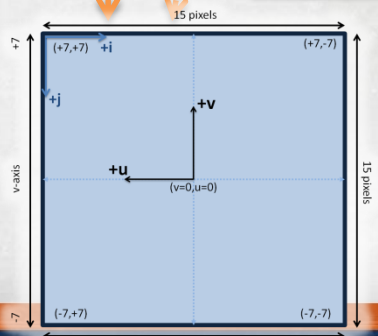
RGB vector (e.g., [.8 .8 .8] to get gray) or built in color (e.g. 'white')

Pick integer from below:
- 0 for no caption
- 1 for caption string only
- 2 for caption string + (u,v) coordinates appended

(Note options 1 and 2 will prevent any 16-bit output from being saved if saveFlag is set to 4 or 5.)

String defining the figure title (caption) if caption flag is 1 or 2

(include apostrophes to designate as a string)



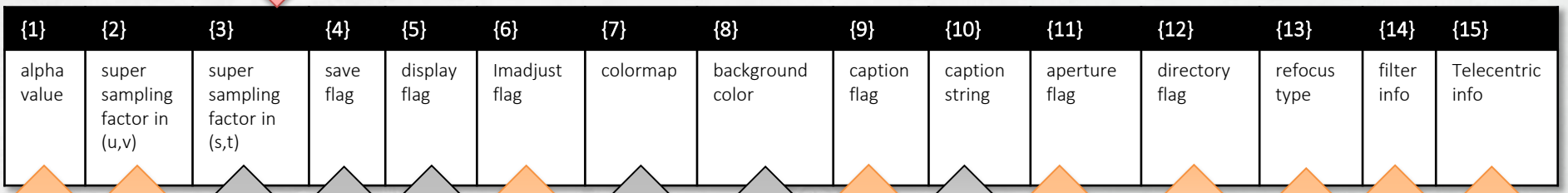
(u,v) coordinate system (hexagonal camera)

AUBURN UNIVERSITY

Discrete Refocus Generator

(genrefocus.m)

Caution: For the refocus generator, every line must have the **same** (s,t) supersampling factor!



Value greater than 0
(decimal values supported)
α>1 focuses closer
α=1 focuses at nominal focal plane
α<1 focuses further

Integer value of 1 or greater
1= no supersampling
2 = 2x supersampling
4 = 4x supersampling, etc…

Pick integer from below:
•0 for no saving
•1 for saving a BMP
•2 for saving a PNG
•3 for saving a JPEG
•4 for a 16-bit PNG
•5 for a 16-bit TIFF
(Options 4 and 5 do not support captions)
('gray' colormap recommended for 4 and 5)

Pick integer from below:
•0 for no display
•1 to display with a pause
•2 to display without a pause

Pick integer from below:
•0 for quasi-raw output
(basic contrast stretching without saturation)
•1 is to apply imadjust function
(saturates top and bottom 1% of pixel intensities to increase contrast)

MATLAB colormap name (e.g., 'gray' or 'jet' or 'pink')
(include apostrophes to designate as a string)

RGB vector (e.g., [.8 .8 .8] to get gray) or built in color (e.g. 'white')

Pick integer from below:
•0 for no caption
•1 for caption string only
•2 for caption string + alpha value appended
(Note options 1 and 2 will prevent any 16-bit output from being saved if saveFlag is set to 4 or 5.)

String defining the figure title (caption) if caption flag is 1 or 2
(include apostrophes to designate as a string)

Pick integer from below:
•0 for full aperture (all microimage data)
•1 to enforce circular aperture *(masks out corner data; important for hexagonal array!)*

Pick integer from below:
•0 to save refocused images on a per-raw plenoptic image basis (default)
•1 to save images on a per-alpha basis (as in PSF generation)

Pick integer from below:
•1 for additive
•2 for multiplicative
•3 for filtered

See Slide 50

See Slide 50

Perspective Animation Generator

(animateperspective.m)

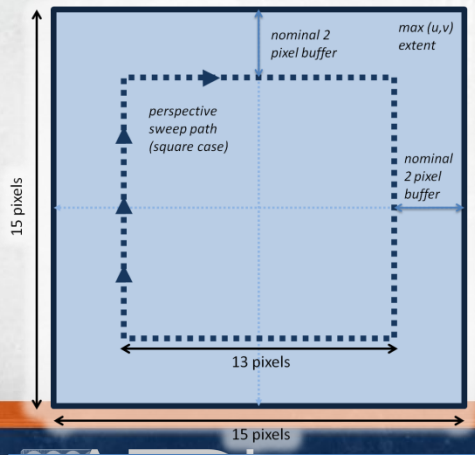
Pick integer from below:
- 0 for no saving
- 1 for saving a GIF
- 2 for saving a AVI
- 3 for saving a MP4

(Option 3 requires MATLAB 2010b or newer)

{1}	{2}	{3}	{4} [save flag]			{5}	{6}	{7}	{8}	{9}	{10}	{11}
edge buffer	super sampling factor in (u,v)	super sampling factor in (s,t)	type Flag	0	0	display flag	imadjust flag	colormap	background color	caption flag	caption string	travel vector index
			arg1	arg2	arg3							

Value between 1 and 7 (rect)
Value between 1 and 6 (hexa)
(decimal values supported)

Integer value of 1 or greater
1= no supersampling
2 = 2x supersampling
4 = 4x supersampling, etc…
(u,v supersampling evaluates non-integer u,v values for a finer/slower movie sweep)



Example of sweep path for edge buffer of 2 with a square path for hexagonal camera

For GIFs:
integer # of loops [inf for unlimited]

For AVI and MP4: fps [integer frames per second]

For GIFs:
delay time between frames (e.g. 0)

For AVI and MP4:
quality [integer between 0 and 100]

For GIFs:
dithering [0 = none; 1 = yes (recommended)]

For AVI: (compression)
MATLAB R2010a or earlier:
0=uncompressed, 1=MSVC, 2=RLE, 3=Cinepak
MATLAB R2010b or newer:
0=uncompressed, 1=Motion JPEG, 2=Lossless Motion JPEG 2000, 3=Compressed Motion JPEG 2000

For MP4: 0 [does nothing]

RGB vector (e.g., [.8 .8 .8] to get gray) or built in color (e.g. 'white')

String defining the figure title (caption) if caption flag is 1 or 2
(include apostrophes to designate as a string)

MATLAB colormap name (e.g., 'gray' or 'jet' or 'pink')
(include apostrophes to designate as a string)

Pick integer from below:
- 0 for raw output
- 1 is to apply imadjust function

(saturates top and bottom 1% of pixel intensities to increase contrast)

Pick integer from below:
- 0 for no display
- 1 to display with a pause
- 2 to display without a pause

Pick integer from below:
- 0 for no caption
- 1 for caption string only
- 2 for caption string + (u,v) coordinates appended

(Note options 1 and 2 will prevent any 16-bit output from being saved if saveFlag is set to 4 or 5.)

Pick integer from below:
- 1 for clockwise square/box path
- 2 for clockwise circle path
- 3 for cross/plus sign (+) path
- 4 for load path from file…

Warning: Option 4 will halt program execution to prompt user for path

AUBURN UNIVERSITY

Refocus Animation Generator

(animaterefocus.m)

Pick integer from below:
- 0 for no saving
- 1 for saving a GIF
- 2 for saving a AVI
- 3 for saving a MP4

(Option 3 requires MATLAB 2010b or newer)

{1} [alpha input]		{2}	{3}	{4}[save flag]			{5}	{6}	{7}	{8}	{9}	{10}	{11}	{12}	{13}	{14}
space type	# steps	super sampling factor in (u,v)	super sampling factor in (s,t)	typeFlag	0	0	display flag	Imadjust flag	color map	background color	caption flag	caption string	aperture flag	refocus type	filter info	Telecentric info
α start	α end			arg1	arg2	arg3										
palindrome	0															

Integer value of 1 or greater
1= no supersampling
2 = 2x supersampling
4 = 4x supersampling, etc…

For GIFs: integer # of loops [inf for unlimited]

For AVI and MP4: fps [integer frames per second]

Pick integer from below:
- 0 for no display
- 1 to display with a pause
- 2 to display without a pause

RGB vector (e.g., [.8 .8 .8] to get gray) or built in color (e.g. 'white')

See Slide 50

{1} [alpha input]	
space type [0 for linear, 1 for log]	number of steps [integer]
alphaStart [starting alpha value]	alphaEnd [ending alpha value]
palindrome [0 for no loop mirroring, 1 to make the loop reverse and continue at the end of the range]	0 [does nothing]

For GIFs: delay time between frames (e.g. 0)

For AVI and MP4: quality [integer between 0 and 100]

Pick integer from below:
- 0 for quasi-raw output (basic contrast stretching without saturation ***on a slice-by-slice basis***)
- 1 is to apply imadjust function (saturates top and bottom 1% of pixel intensities to increase contrast)
- 2 for stack-normalized contrast (contrast limits set according to max and min intensities in entire focal stack)

String defining the figure title (caption) if caption flag is 1 or 2

(include apostrophes to designate as a string)

See Slide 50

For GIFs:
dithering [0 = none; 1 = yes (recommended)]

For AVI: (compression)
MATLAB R2010a or earlier:
0=uncompressed, 1=MSVC, 2=RLE, 3=Cinepak
MATLAB R2010b or newer:
0=uncompressed, 1=Motion JPEG, 2=Lossless Motion JPEG 2000, 3=Compressed Motion JPEG 2000

For MP4: 0 [does nothing]

MATLAB colormap name (e.g., 'gray' or 'jet' or 'pink')

(include apostrophes to designate as a string)

Pick integer from below:
- 0 for full aperture (all microimage data)
- 1 to enforce circular aperture *(masks out corner data; important for hexagonal array!)*

Pick integer from below:
- 0 for no caption
- 1 for caption string only
- 2 for caption string + alpha value appended

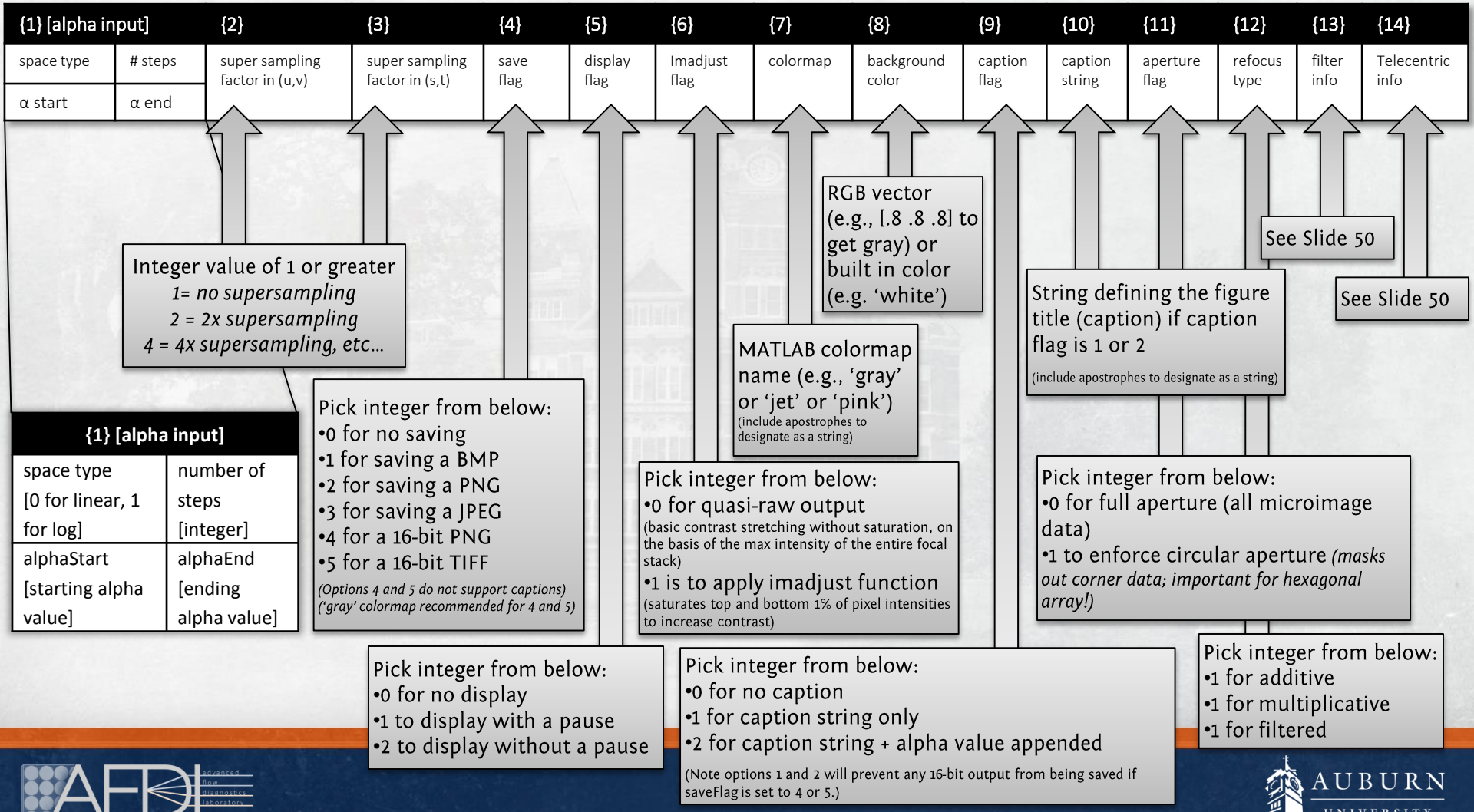
(Note options 1 and 2 will prevent any 16-bit output from being saved if saveFlag is set to 4 or 5.)

Pick integer from below:
- 1 for additive
- 2 for multiplicative
- 3 for filtered

Focal Stack Generator

(genfocalstack.m)

Optional Output Argument:
[focalStack] = genfocalstack(…)
exports the focal stack to the main workspace.



{1} [alpha input]	
space type [0 for linear, 1 for log]	number of steps [integer]
alphaStart [starting alpha value]	alphaEnd [ending alpha value]

Extra Request Vector Elements

Telecentric Info

{1}	{2}	{3}	{4}	{5}	{6}	{7}	{8}	{9}	{10}	{11}	{12}	{13}
telecentric flag	x min	x max	y min	y max	z min	z max	voxX	voxY	voxZ	focLen Main	magnification	z location

Pick an integer from below:
- 0 for non-telecentric
- 1 for telecentric

Volume boundaries in mm

Number of volume elements in each direction

Main lens focal length

Magnification

Single plane z location in mm

Filter Info

{1}	{2}
noise threshold	filter threshold

Filter intensity threshold

Threshold below which intensity will be disregarded as noise

Questions?

Contact:

Elise Munz

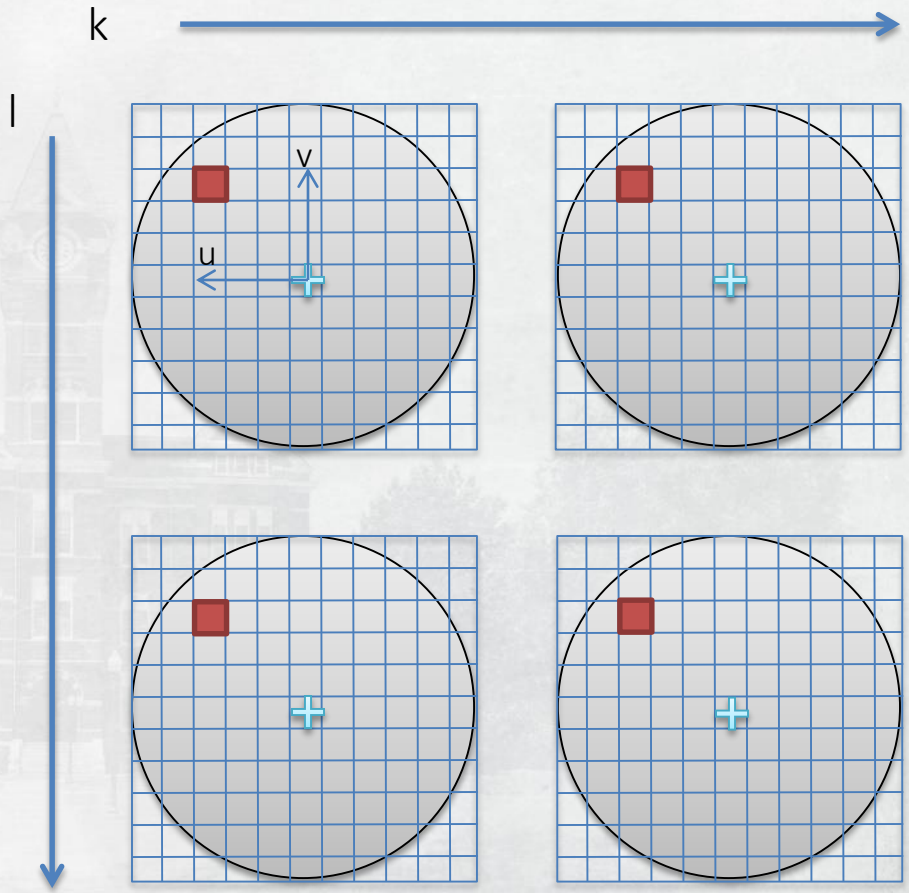
(edm0003@auburn.edu)

Other relevant LFIT materials

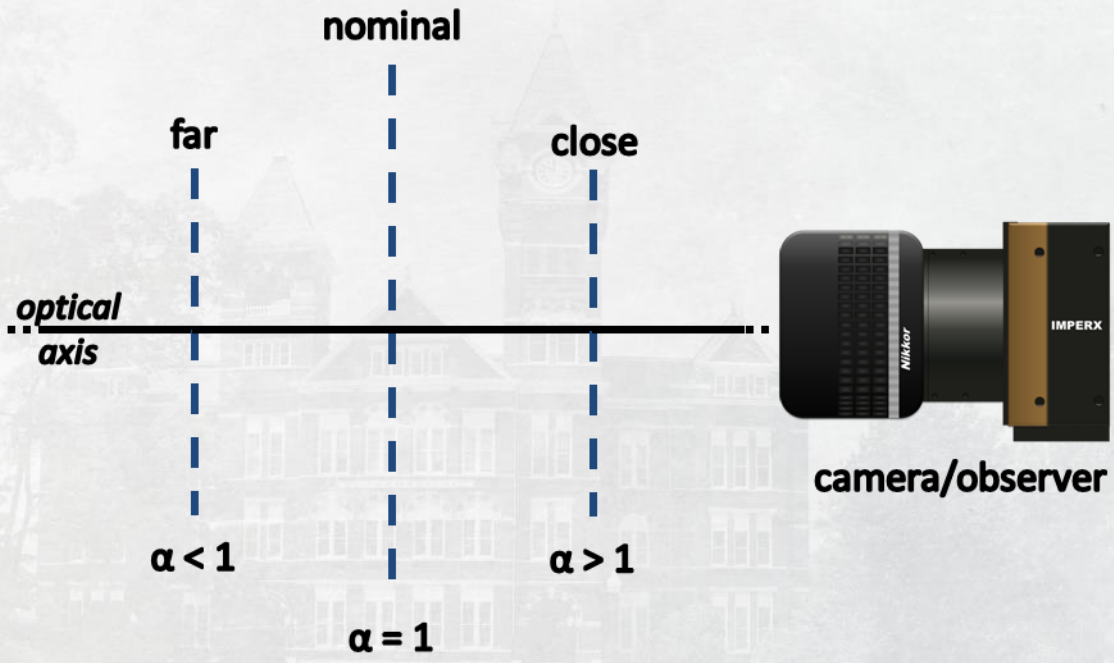
APPENDIX

Perspective Shifts

- Sub-aperture imaging
  - Hold (v,u) constant and vary (k, l) values
  - Analogous to extracting the same pixel beneath each microlens



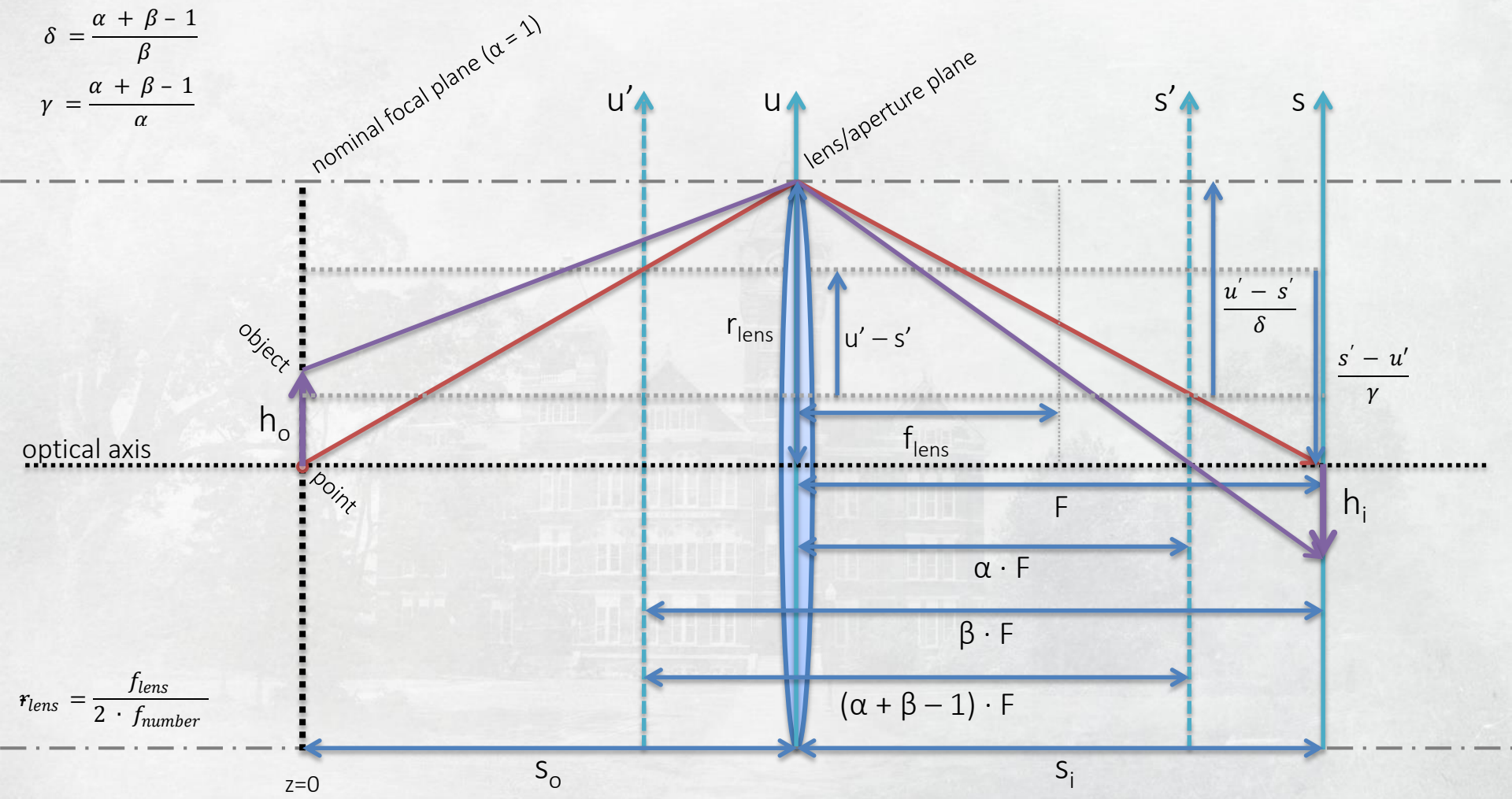
Alpha: Depth Parameter



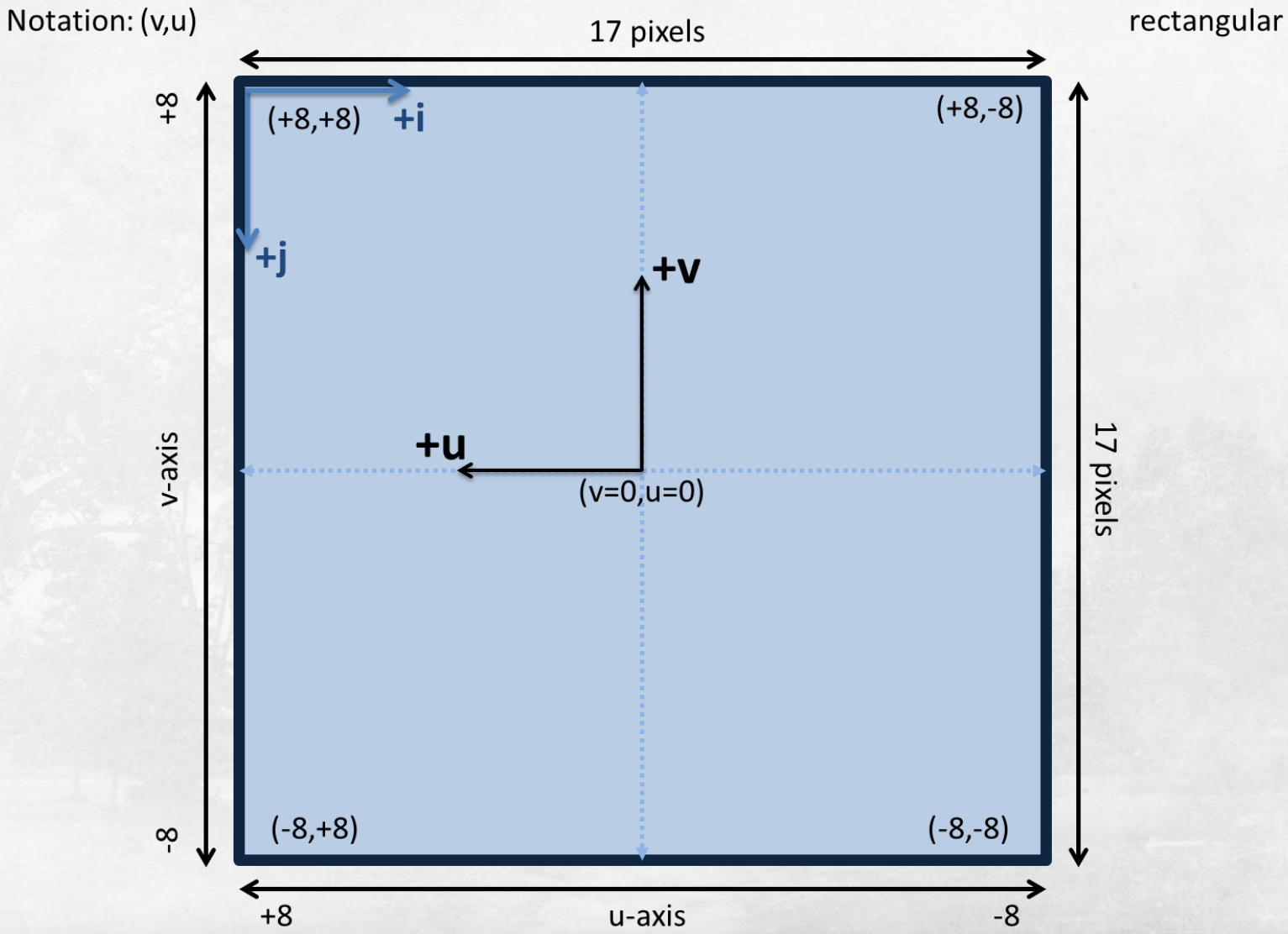
Geometric Relationships

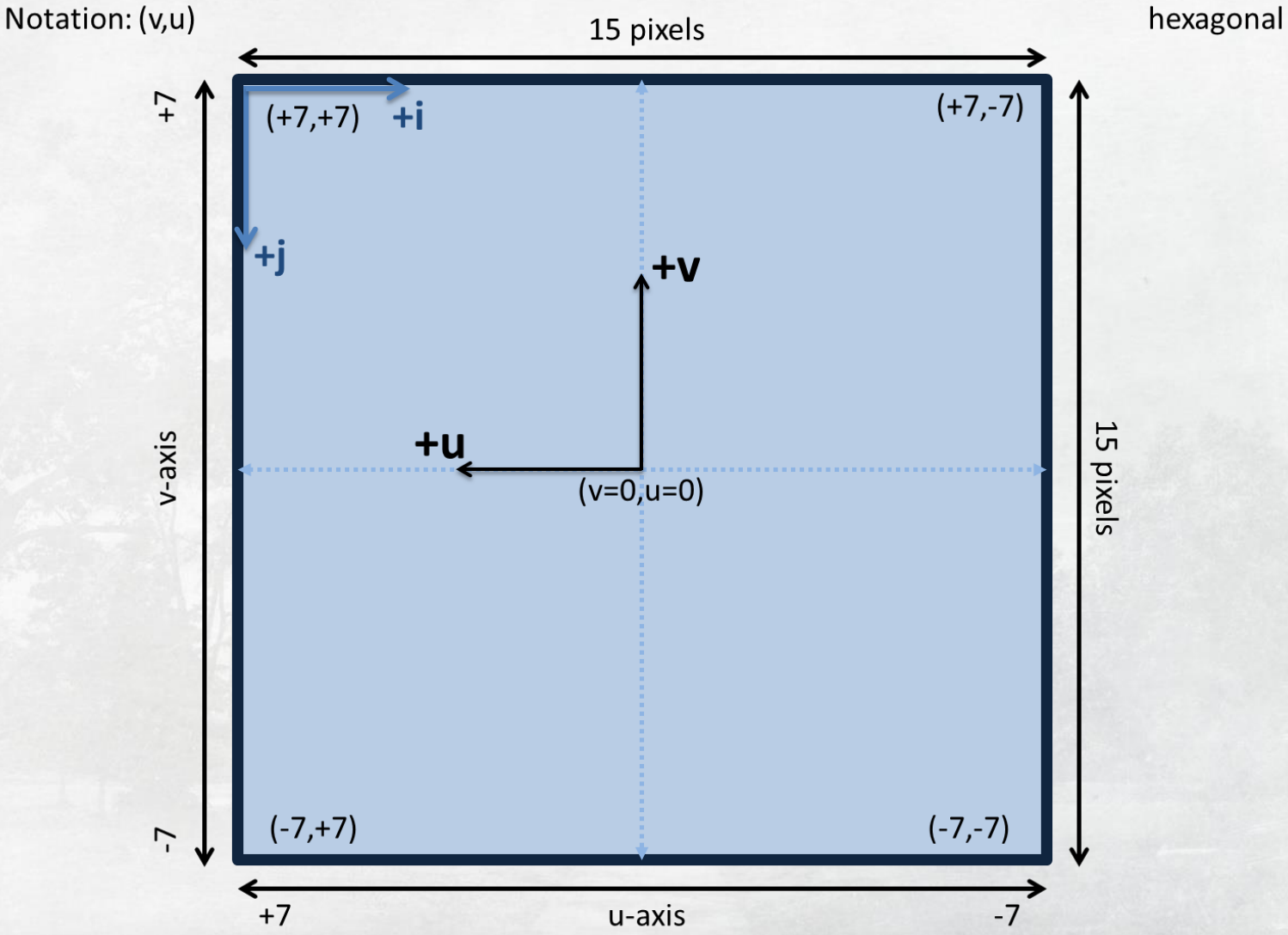
$$\delta = \frac{\alpha + \beta - 1}{\beta}$$

$$\gamma = \frac{\alpha + \beta - 1}{\alpha}$$

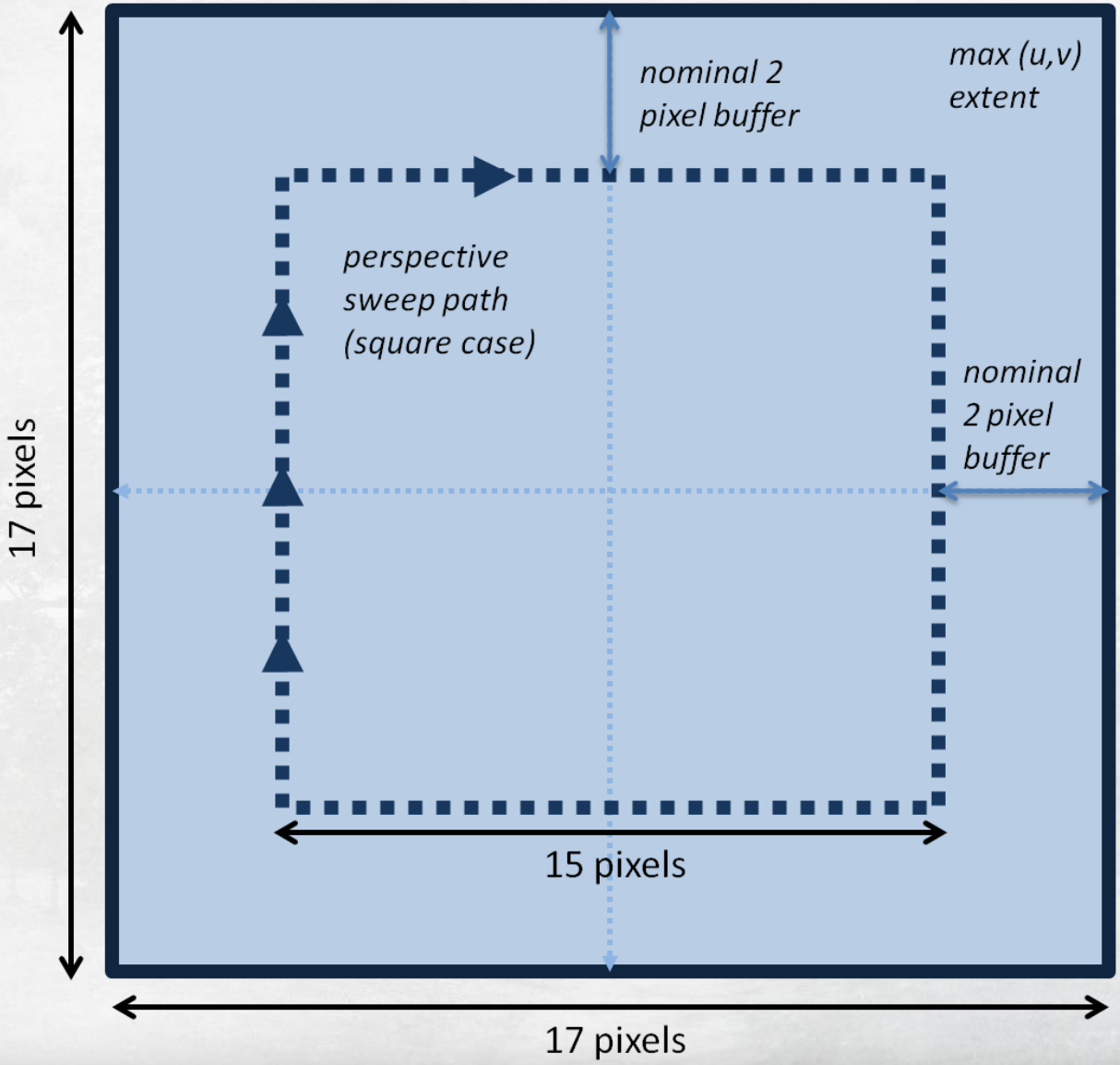


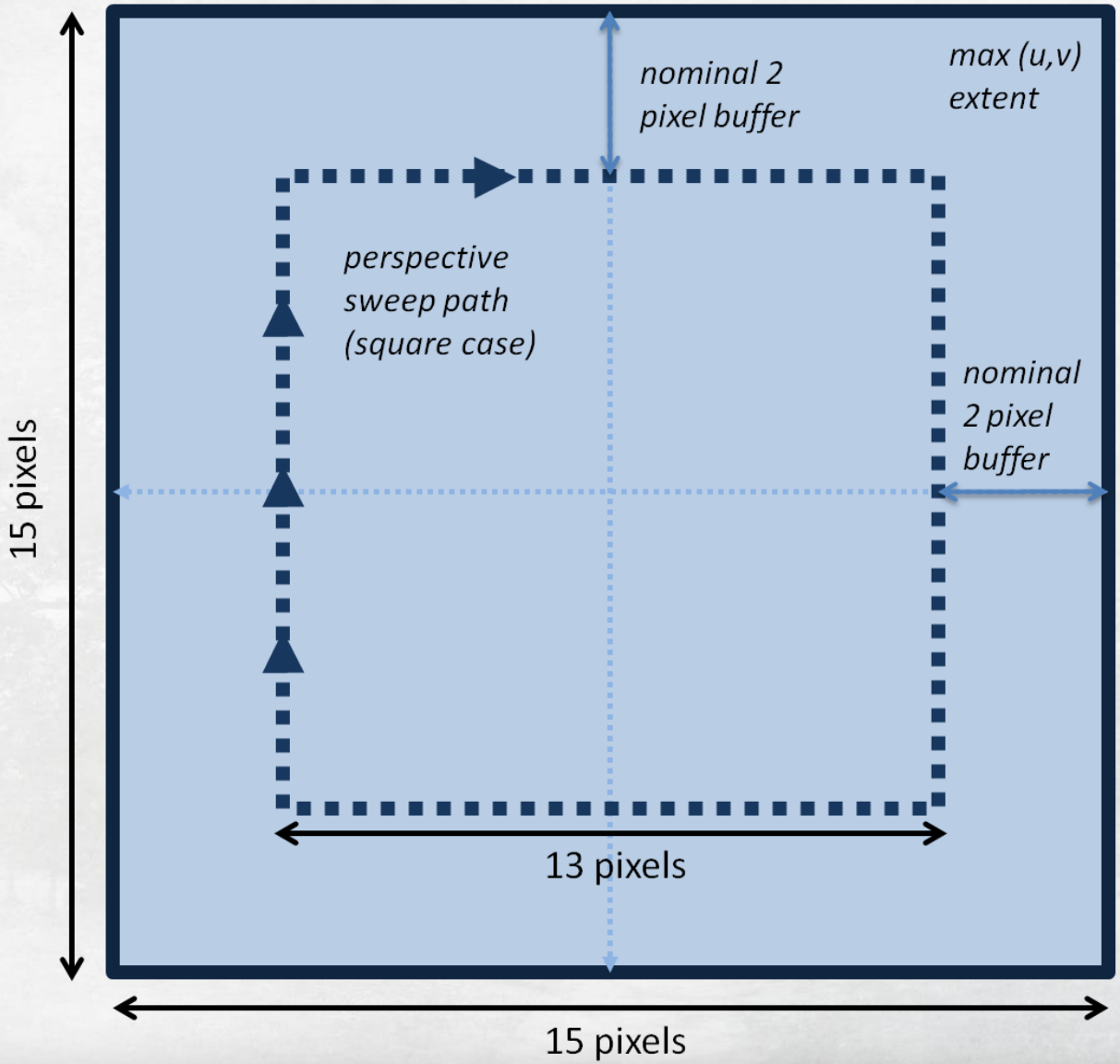
$$r_{lens} = \frac{f_{lens}}{2 \cdot f_{number}}$$

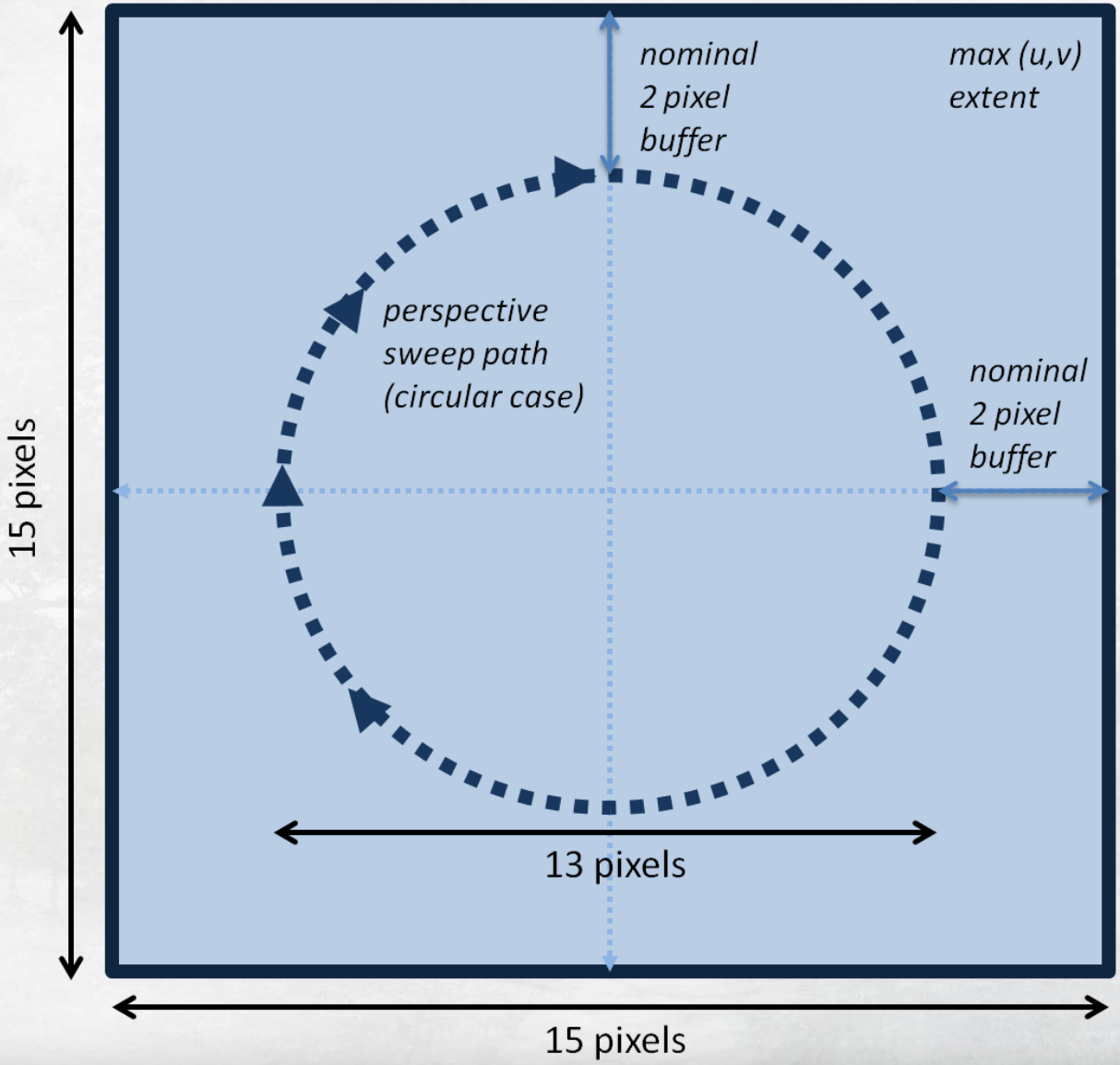

Notation: (v,u)
17 pixels
rectangular
+8
(+8,+8)
+i
(+8,-8)
+j
+v
+u
(v=0,u=0)
v-axis
17 pixels
-8
(-8,+8)
(-8,-8)
+8
u-axis
-8

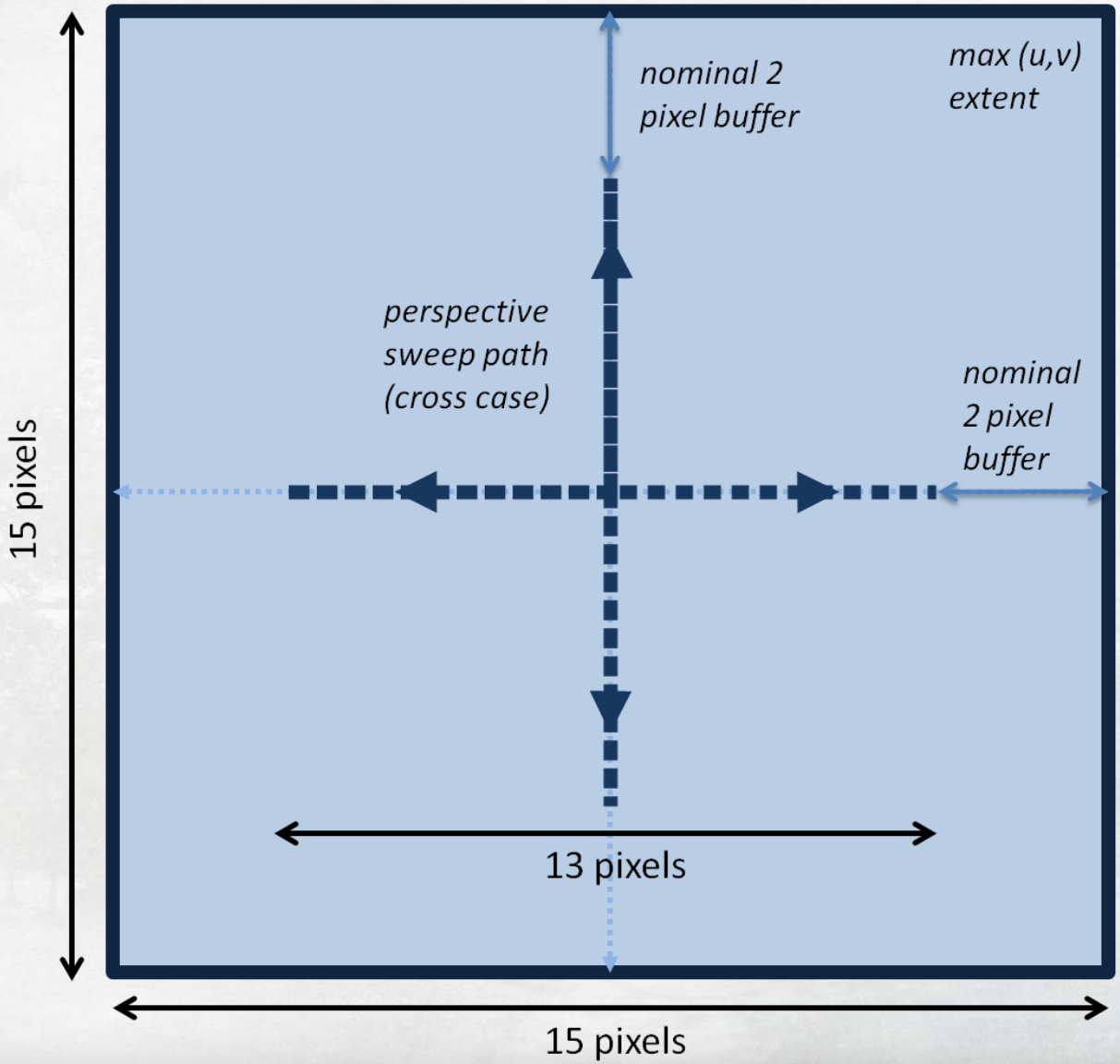

Notation: (v,u)
15 pixels
hexagonal
+7
(+7,+7)
+i
(+7,-7)
+j
+v
+u
v-axis
(v=0,u=0)
15 pixels
-7
(-7,+7)
(-7,-7)
+7
u-axis
-7

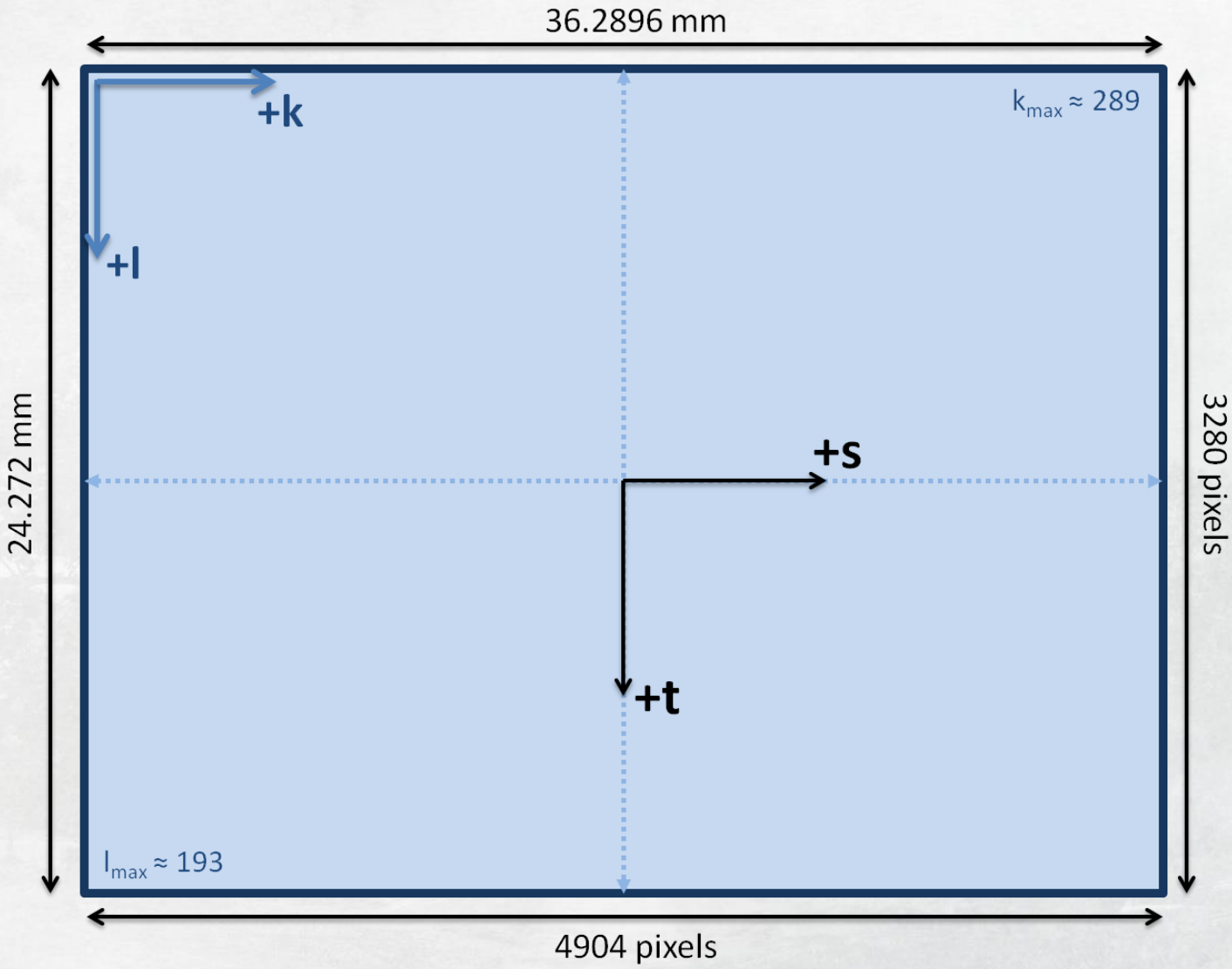
rectangular

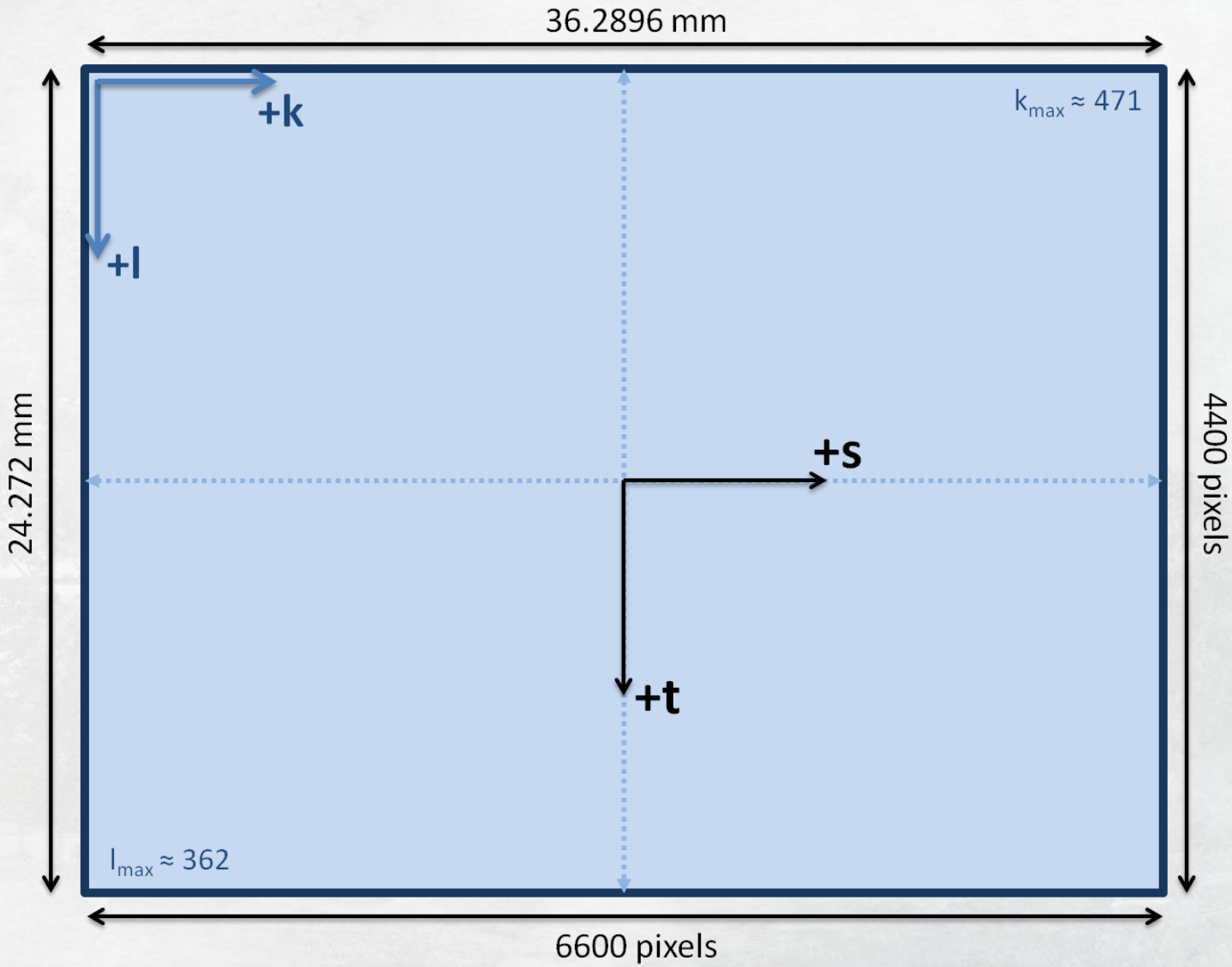



nominal 2
pixel buffer
max (u,v)
extent
perspective
sweep path
(square case)
nominal
2 pixel
buffer
15 pixels
13 pixels
15 pixels


nominal
2 pixel
buffer
max (u,v)
extent
perspective
sweep path
(circular case)
nominal
2 pixel
buffer
15 pixels
13 pixels
15 pixels

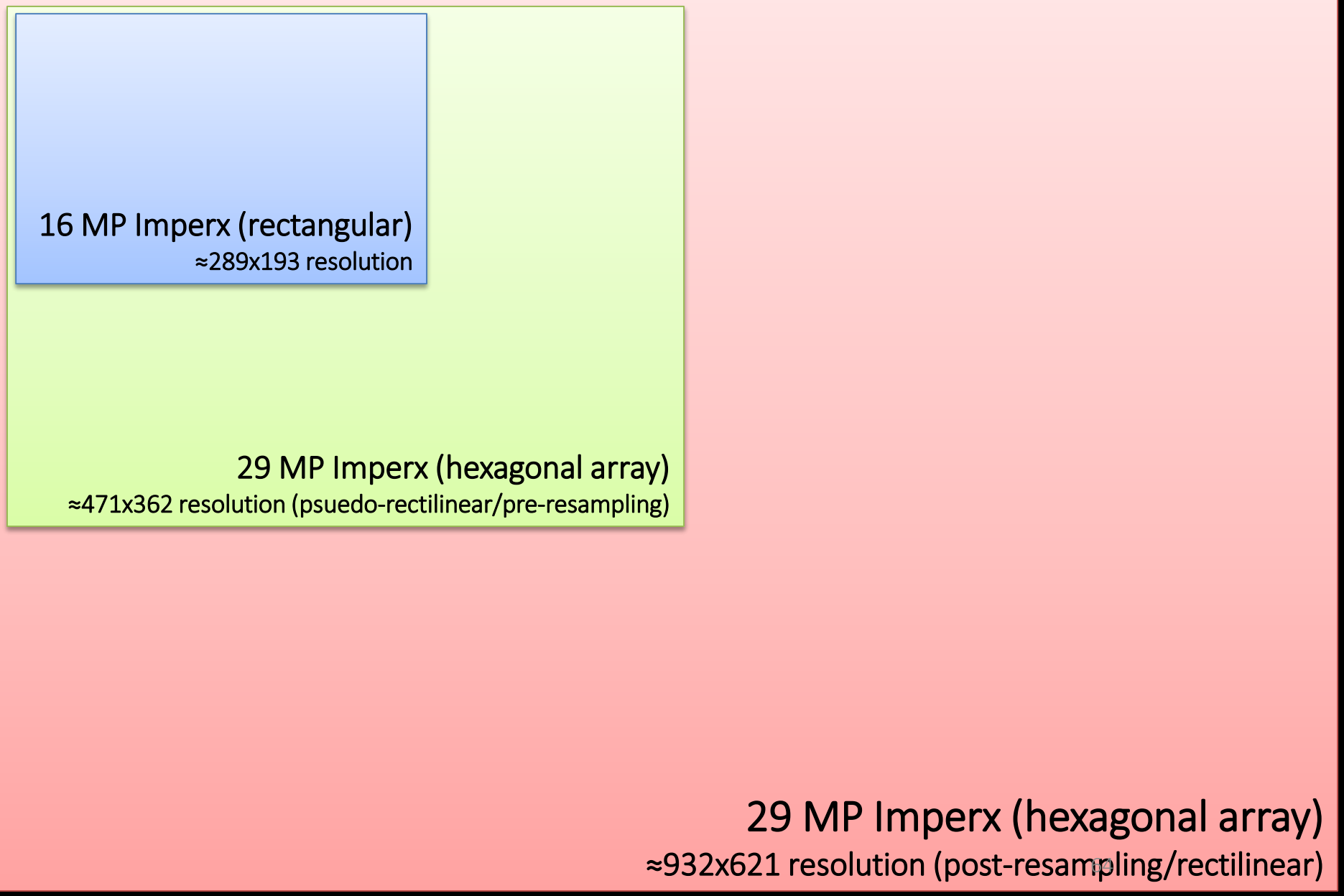

nominal 2
pixel buffer
max (u,v)
extent
perspective
sweep path
(cross case)
nominal
2 pixel
buffer
15 pixels
13 pixels
15 pixels

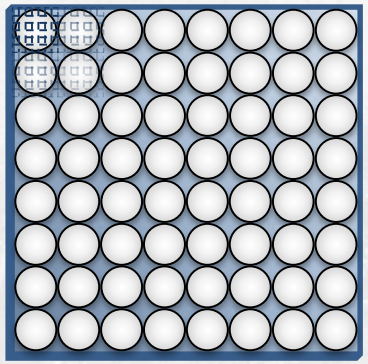

36.2896 mm
+k
+l
$k_{max} \approx 289$
+s
+t
24.272 mm
3280 pixels
$l_{max} \approx 193$
4904 pixels



l_{max} and k_{max} here are pre-resampling.

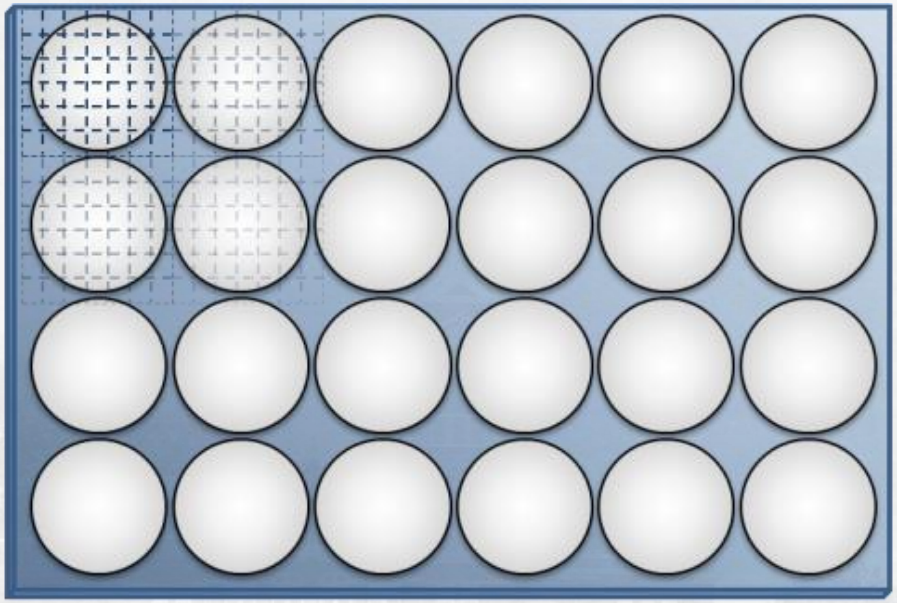
1:1 Resolution Comparison





LFIT v2

Light Field Imaging Toolkit V2



LFIT v2

LIGHT FIELD IMAGING TOOLKIT V2


AFDL
advanced
flow
diagnostics
laboratory