Guide for using the Photo Monitoring Plugins

Overview

This guide explains how to use the photo monitoring plugins written to work with ImageJ and a variant of ImageJ called Fiji. Since Fiji has most of the plugins accessed by the Photo Monitoring plugins bundled with the distribution it is easier to work with Fiji but the plugins should work equally as well in ImageJ if the installation instructions below are followed.

There are two plugins. One is designed to create a file defining image pairs acquired using two different cameras. The other uses the image pair file and geometrically registers one of the images to overlay over the other and also calculates NDVI and related image products if one of the cameras records a near-infrared (NIR) image.

An attempt has been made to create some processing messages that will be output to the ImageJ/Fiji “Log” window. When an error occurs an informative message should be displayed. Please keep in mind that the plugins are still in a beta state so you should maintain a backup copy of all of your images in case the input images are altered.

This guide and the plugins are works in progress and comments are welcome. I work primarily with the Ubuntu operating system so these instructions are based on my experience with Ubuntu. If you notice something that should be altered to accommodate other operating systems please let me (Ned Horning - horning@amnh.org) know.

Installing the plugins

The first step is to download and install ImageJ or Fiji (recommenced) which is a distribution of ImageJ – a powerful image processing package designed to support the biomedical research community. Fiji can be downloaded from http://Fiji.sc/wiki/index.php/Fiji. It is Java based and runs on all computers that support Java. After Fiji is installed you should launch it then click on the menu “Help => Update ImageJ” to get the most recent version of the ij.jar file. If the update does not work you can do the update manually by downloading the most recent version from here: rsb.info.nih.gov/ij/upgrade/ij.jar. Next, replace the old ij.jar file the new one in the “jars” directory which is located under the Fiji root directory. You must use a version that is more recent than 1.46r.

You will also need to get a copy of the plugin which is available on GitHub at https://github.com/nedhorning/PhotoMonitoringPlugin. In addition to the photo monitoring plugin you will also need to download the metadata-extractor-2.6.2.jar and xmpcore.jar files. A copy of these files are available on the GitHub site in the downloads section or you can get it from: http://code.google.com/p/metadata-extractor/downloads/list. You need to download the metadata-extractor-2.6.2.zip file and then unzip that file to get the metadata-extractor-2.6.2.jar and xmpcore.jar files.

These plugins can be accessed through the Fiji or ImageJ Plugins menu by copying the Photo_Monitoring.jar file into the plugins directory which is in the Fiji or ImageJ root directory. After the Photo_Monitoring.jar file has been copied and Fiji/ImageJ has been restarted you

will see a listing labled “PhotoMonitoring” under the Plugins menu. Clicking on that will reveal the plugins described in this guide.

If the plugins are used with ImageJ then the following additional .jar files from the Fiji project need to be also copied to the plugins directory: jars/mpicbg.jar, plugins/mpicbg_.jar, and plugins/bUnwarpJ_.jar. Also, the metadata-extractor-2.6.2.jar and xmpcore.jar files must be copied to the plugins directory.

Using and creating custom NDVI lookup tables

A lookup table (LUT) is used to define which colors will be used to represent different NDVI values when a color NDVI image is output. ImageJ and Fiji have several dozen LUTs but none dedicated specifically for NDVI. A set of LUTs have been created to render NDVI images in color. The custom LUTs can be found on the GitHub site: https://github.com/nedhorning/PhotoMonitoringPlugin. The human readable CSV files used to create the LUTs using the ImageJ “LUT Importer” plugin (http://rsbweb.nih.gov/ij/plugins/lut-importer.html) are located in the main GitHub page and the binary LUTs for direct use with ImageJ or Fiji are in the downloads section.

To install the LUT you need to copy the .lut file (from the downloads section of the GitHub site) to the “luts” directory which is in the Fiji or ImageJ root directory. The next time you start ImageJ or Fiji the LUT should appear in the list of LUTs. To see what the LUT looks like you can use a ImageJ/Fiji utility. Go to Image => Lookup Tables then click on the LUT you'd like to see and a color bar will open. The color on the left side of the bar represents an NDVI value of 0 and the color on the right side represents a value of 255.

If you want to create your own LUT you need to create a CSV file with the first column containing the pixels value (0 to 255) and the next three columns are the intensity values for Red, Green, and Blue respectively. A CSV file can be created using a spreadsheet such as Excel or Calc. The lowest valid value for these three columns is 0 and the highest is 255. You can use the CSV file in the GitHub site as an example. Once the CSV file is finished you need to use the “LUT Importer” plugin to convert the values to a format that can be used by ImageJ/Fiji. After running the “LUT Importer” plugin with the color bar displayed and active you can save the LUT by clicking on Image => Color => Edit LUT... then when the LUT Editor opens click save. Save the LUT in the “luts” directory which is in the Fiji or ImageJ root directory. The LUT will appear in the list of LUTs next time you open ImageJ/Fiji.

Here is a brief description of the different custom LUTs:

- ***ndviClasses_0_1.lut:*** 11 different colors ranging from reddish brown for low NDVI to dark green for high NDVI. This is designed to work with scaling NDVI values from 0 to 1.
- ***ndviClasses_-1_1.lut:*** 11 different colors ranging from reddish brown for low NDVI to dark green for high NDVI. This is designed to work with scaling NDVI values from -1 to 1.
- ***colorRampNDVI.lut:*** A gradient from light beige for low NDVI to very dark green for high NDVI. This will work with scaling ranging from -1 to 1 and 0 to 1 but the results will be very different.

Remove low quality photos

Before you use the plugins it's a good idea to review your photos and remove any that are of low quality or have no image on them. The current version of the plugin is known to freeze when an all black image (like one you would get if you had the lens cap on) is encountered.

Plugin to create file of image pairs

The “Create list” plugin is designed to facilitate the process of matching two digital photographs that were acquired at roughly the same time. The plugin outputs a text file with the path and file names for image pairs (e.g., images acquired from two cameras) that can be input into the “Reg images” plugin. The image matching is done by synchronizing the times stored in the image EXIF DateTimeOriginal tag from each of two cameras. If for some reason the EXIF DateTimeOriginal tag is not set then the files last modified time will be used.

Before running the plugin the images from each camera should be stored in different directories. If this is not convenient please let me know of your preferred method for differentiating between the images acquired from each camera and I will consider adding alternative methods.

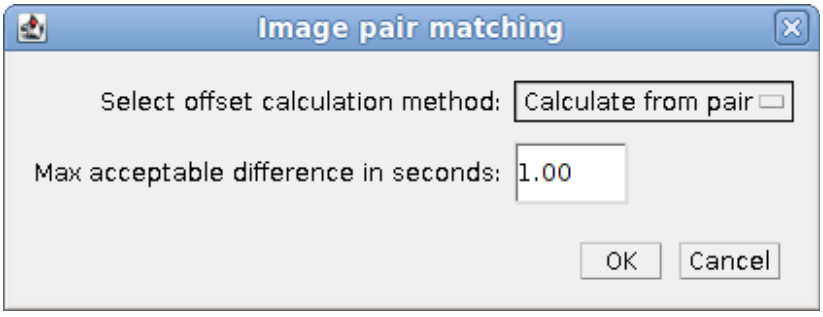
Synchronization of the image pairs is done by calculating or reading a user defined offset between the imaging time recorded by one camera with the image time from another camera and then using the offset to find which images were acquired at roughly the same time.

When the plugin starts a dialog window is opened with the following options:

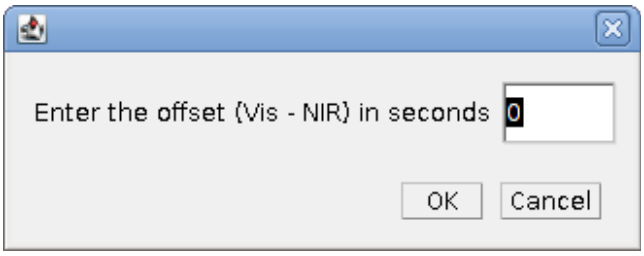
Select offset calculation method: There are three options for calculating the time:

- *Assume synched:* Assumes the camera clocks were synchronized to the same time before photo were acquired so an offset of 0 seconds will be used.
- *Enter offset:* Allows you to enter the offset in seconds. The offset should be calculated using this equation: time of visible image – time of NIR image. It the result is negative it is important to include the negative sign (“-”) before the number of seconds.
- *Calculate from pair:* Uses two images from a pair (one visible one NIR) to calculate the offset. If this option is selected then you will have to specify the visible and NIR images that will be used as the reference pair.

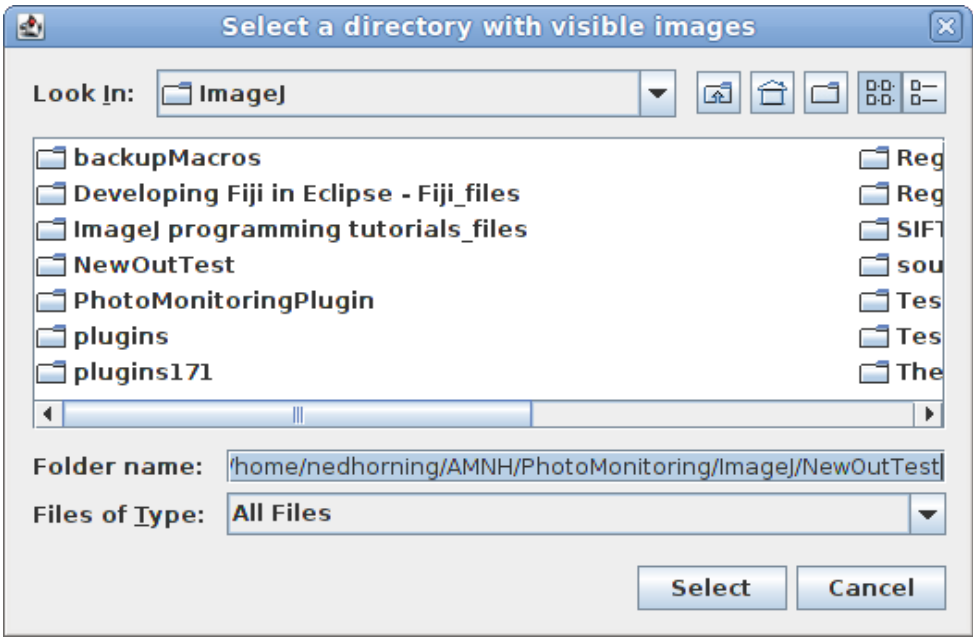
Max acceptable difference in seconds: This is the maximum time difference in seconds, after the calibration has been applied, that is allowable between each image pair. This compensates for slight variations in the camera clock and shutter release mechanism.



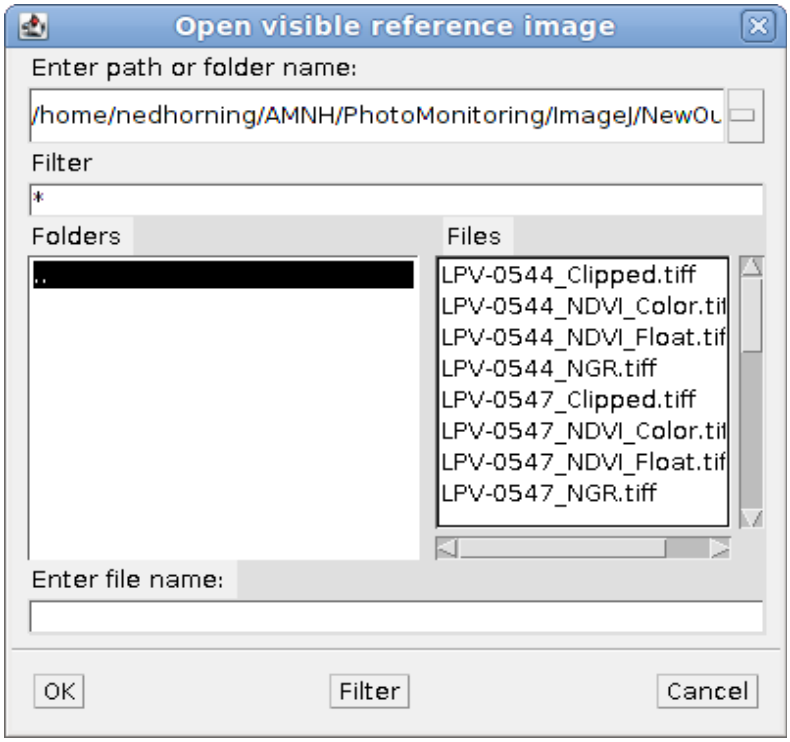
If the “Enter offset” option is selected you will need to enter the offset (calculated as: time of visible image – time of NIR image) in seconds.



If the “Assume synched” or “Enter offset” options are selected you will need to select the location for the visible and NIR images.



If the “Calculate from pair” option is selected then you will need to select the visible image that will be used as a reference to synchronize the times to select image pairs. After clicking “OK” a window will open so you can select the NIR image that will be used as a reference. For both of these windows the assumption is made that all of the visible images that will be processed are in the same directory as the visible reference image and all of the NIR images are in the same directory as the NIR reference image.



Regardless of the other options the last window is where you select the output file location and name. After selecting the directory path and entering the output file name click on “Save” to run the program. It should complete in seconds.

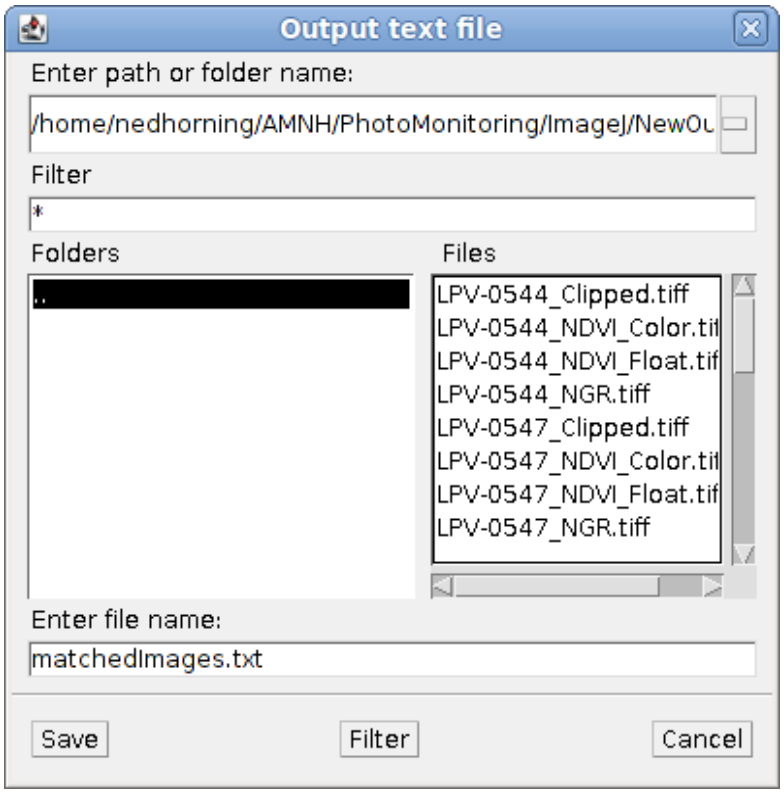


Image registration plugin

The “Reg images” plugin is designed to co-register two images, one using a NIR camera and the other a “normal” visible camera. The plugin will work best if the images were acquired from two cameras mounted with their lenses close to each other, acquired at nearly the same time (so the scene hasn't changed), and it's best if the two cameras have similar characteristics such as image size and resolution.

The plugin can output the following images:
NRG image (false-color image with R=NIR, G=red from visible, and B=green from visible)
NDVI image with a user-selected color table applied
Floating point NDVI image with actual NDVI values (data range -1 to +1)
A visible image clipped to the common area between the registered NIR and visible image
A log file documenting the registration method used for each image pair

The parameters used to run SIFT, Landmark correspondences, and bUnwarpJ are coded into the plugin and at this point can not (at this time) be modified by the parameter window displayed by this plugin.

Instructions

Before running the plugin you will need to prepare a text file that directs the plugin to process specific image pairs. This can be accomplished by running the “Create list” plugin described above or by creating your own file.

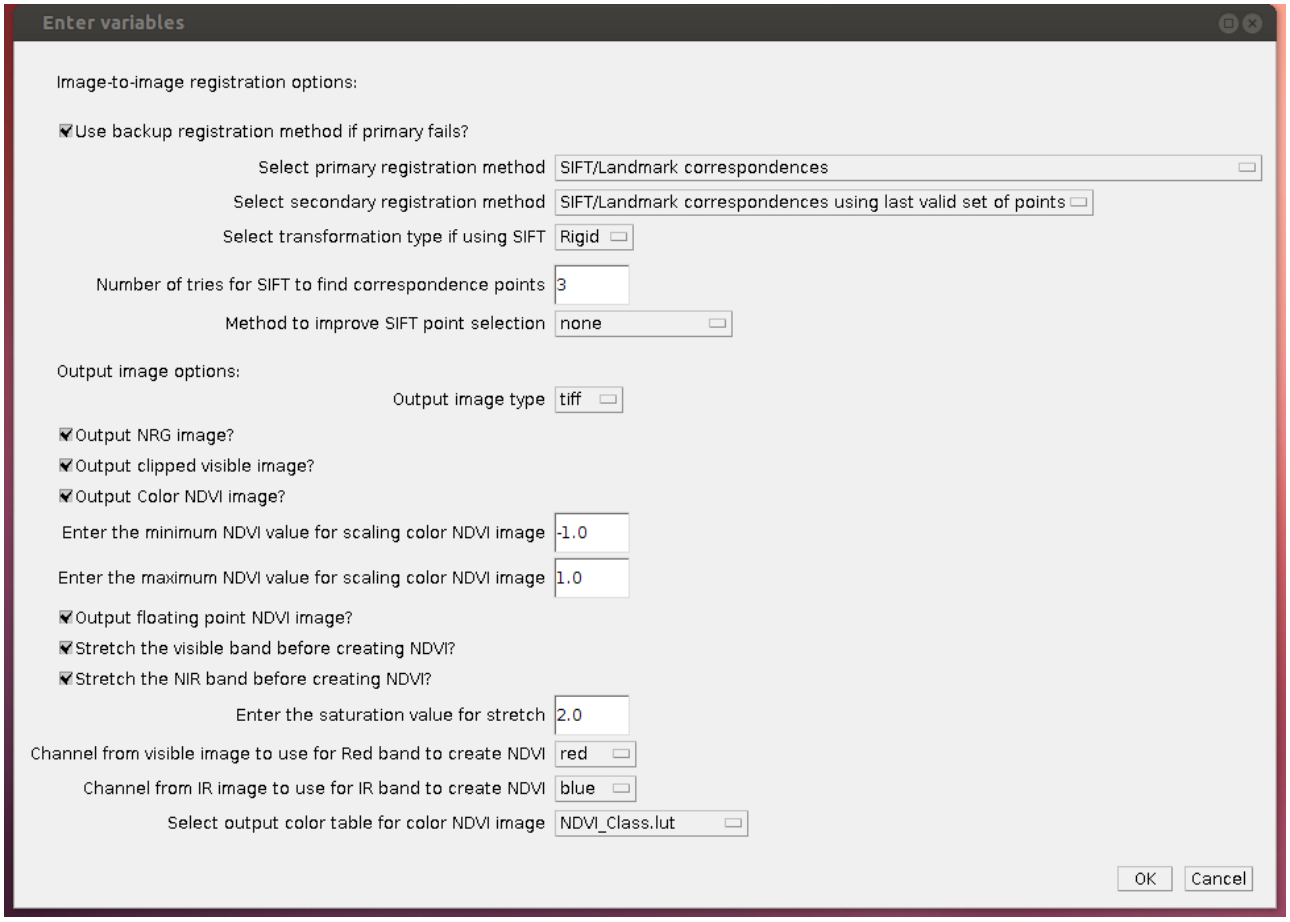
If you create your own file you need to ensure the following format is followed. Each line in the file must have the directory path and file name for the two images that form a pair with a comma separating the two. The first image is the near_IR image (the source image that will be warped) and the second image is the visible image (the target image used as a reference but not altered).
Here is an example:
ned/InputImages/nir/LPV-0001.JPG, ned/InputImages/visible/LPV-1324.JPG
ned/InputImages/nir/LPV-0003.JPG, ned/InputImages/visible/LPV-1325.JPG

The plugin will work with any image format that can be read by ImageJ/Fiji. Your first (test) run should probably include just a couple image pairs to make sure it's working.

To run the plugin go to Plugins => Photo Monitoring => Reg images. When the plugin starts a dialog window opens that lets you specify processing options.

Modify the following options to specify the output that you want to create.

- ***Use backup registration method if primary fails:*** This option allow you to specify if you want to try a backup registration method if the primary method fails.



- ***Select primary registration method:*** This is the first registration method that will be tried for the image pair. Three registration methods are available. The first, “SIFT/Landmark correspondences”, uses a scale-invariant feature transform (SIFT) to select matching points in image pairs. After points are selected a transformation is applied that will modify the NIR image. Unfortunately SIFT often does not always select points that correspond with both images in landscape photos since there are ofter not enough distinct features that can be detected and match. A second registration option, “SIFT/Landmark correspondences using reference points from first valid image pair” will use the reference points from the previous set of correspondence points calculated using SIFT. This option is fast since new correspondence point do not need to be calculated but the results will only be good if there is no relative movement between the two cameras that acquired the image pairs. The third option is “bUnwarpJ” which uses the bUnwarpJ plugin in Fiji. This algorithm is able to register most image pairs and is able to deal with more severe types of image mismatches. The downside is that this algorithm take more time (~4 minutes vs 15 seconds for the SIFT approach) and it can introduce distortions if the corresponding reference points in the image pair are not very accurate and well distributed throughout the image. More information about bUnwarpJ can be found at http://biocomp.cnb.uam.es/~iarganda/bUnwarpJ/.
- ***Select secondary registration method:*** These options, similar to the primary registration options described above, are applied if a backup registration method is requested using the checkbox at the top of the parameter window.
- ***Select transformation type if using SIFT:*** The transformation type option only has an effect if you use the SIFT registration method selected. The *affine* transformation

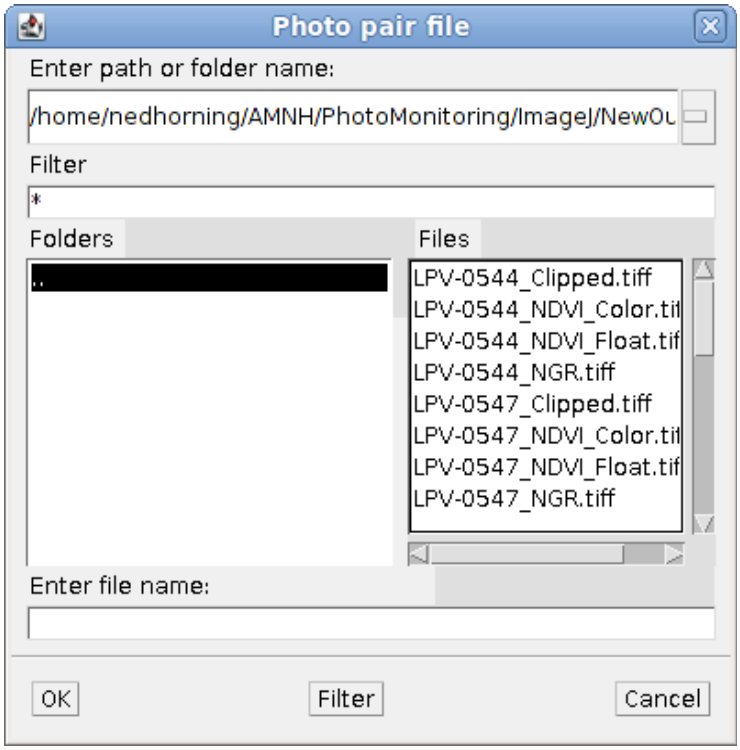
adjusts the image using translation, rotation and scaling and the *rigid* transformation only uses translation and rotation. The rigid transformation sometimes produces a better image-to-image fit when few corresponding points are selected using SIFT.

- ***Number of tries for SIFT to find correspondence points***: This specifies the number of times the SIFT algorithm will run to find corresponding reference points on the two images. The results differ each time the SITF algorithm runs so for difficult to match image pairs running multiple time will sometime result in a match. If you do not want to use multiple runs set this to “1”.
- ***Method to improve point selection:*** This option presents different pre-processing methods that can be used to improve the ability of the SIFT algorithm to find corresponding points. This is experimental and the only option at this time is "target (g-b) source (g+b)" which subtracts the blue channel from the green channel of the target (visible) image and adds the green channel to the blue channel of the source (NIR) image to create input images for the SIFT algorithm. This approach seems to work well for the camera pairs made using the information on the PLOTS web site. Other approaches can be added for camera pairs if this algorithm isn't effective.
- ***Output image type:*** File format for the output images. Remember that if you want to output a floating point NDVI image you will need to select an image format that supports floatig point numbers such as TIFF. Most common graphics image formats such as JPEG do not support floating point numbers and will cause the output image to appear black.
- ***Output NRG image?:*** Output a false-color composite image with the following RGB color channel assignments: R=NIR, G=red from visible, and B=green from visible.
- ***Output clipped visible image?:*** Output a copy of the visible image that is clipped to matched the dimensions of the other output images.
- ***Output* Color NDVI image?:** Output a color NDVI image using the look-up table specified below.
- **Enter the minimum NDVI value for scaling color NDVI images:** When the color NDVI image is created it is necessary to scale from actual floating point NDVI values (possible values range from -1.0 to +1.0) to a range of integers (possible values range from 0 to 255) allowed for the color image. This parameter allows you to set the minimum NDVI value that will be assigned to a value of 0 in the output color image. NDVI values between the minimum and maximum (next parameter) are scaled linearly from 0 to 255.
- **Enter the maximum NDVI value for scaling color NDVI images:** Same principle as the previous variable but value you enter here will be assigned to a value of 255 in the output color image.
- ***Output floating point NDVI image?:*** Output a floating point NDVI image with actual NDVI values that range from -1 to +1.
- **Stretch the visible band before creating NDVI?:** This option will stretch the visible band used to calculate NDVI. This can be used to normalize images if the features in each image are similar. If some images are mostly vegetation and some are mostly urban environments using this technique to normalize images might not work well.
- **Stretch the NIR band before creating NDVI?:** Similar to the previous parameter but for the NIR band used to calculate NDVI.
- **Enter the saturation value for stretch:** The saturation value determines a threshold for extreme values in the image. In other words it defines which of the lowest values will be set to 0 and which of the highest values will be set to 255. If the default value of

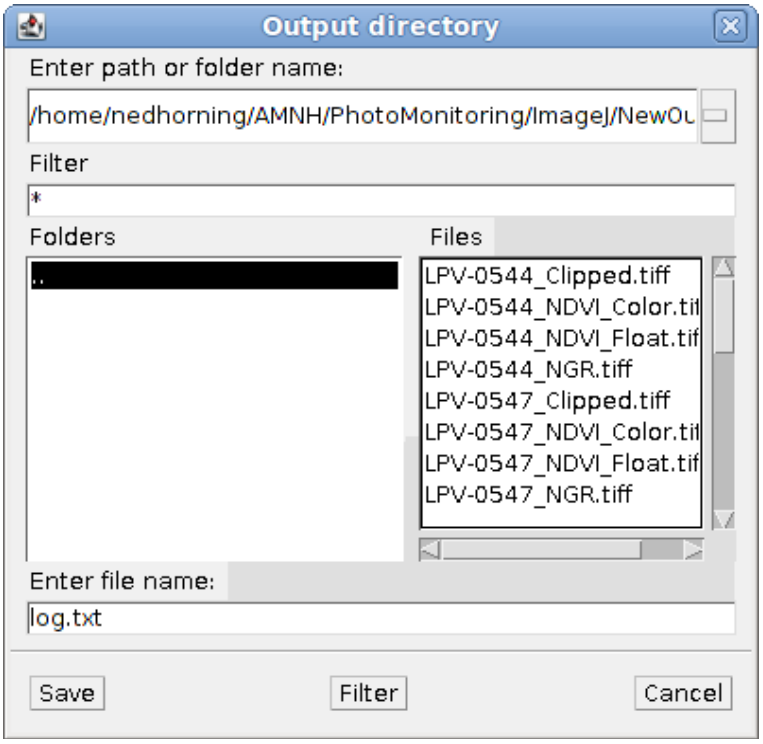
2.0 is used then 1 percent of the lowest values will be set to zero and 1 percent of the highest values will be set to 255 and the remaining values will be linearly scaled to a range of 0 to 255.

- ***Channel from visible image to use for Red band to create NDVI:*** Select the channel number from the visible image that will be used as the red band in the NDVI formula. The default selection seems to work well with most color digital cameras.
- ***Channel from IR image to use for NIR band to create NDVI:*** Select the channel number from the near-infrared image that will be used as the NIR band in the NDVI formula. The default selection seems to work well with the PLOTs nearIR cameras.
- ***Select output color table for color NDVI image:*** The color lookup table to be used if a color NDVI image is being output. You can add new color lookup tables to Fiji/ImageJ by following the instructions in the ImageJ manual.

After you select the options you want click on “OK”. The next window allows you to select the file containing the path and image names of each image pair is located. Select the file (possibly the output from the Create list plugin) and click “OK”.



The last window allows you to select the directory where the images and log file will be output. You can also change the name of the log file that is output when the plugin is run. The log file will be output to the same directory as the output images. After you specifying the directory and log file click on “Save” to start processing.



If the SIFT registration method is selected and an image pair fails to register after 5 tries that pair will be skipped and processing of the remaining image pairs will continue.

Error messages

Here are a few error messages and the recommended solutions. If you are not able to fix your error with this information please contact Ned Horning (horning@amnh.org).

Error:
first line of the exception message is:
java.lang.NoSuchMethodError: ij.process.ColorProcessor.getChannel(I)[B

Explanation and solution:
You are using a version of ij.jar that is older than 1.46r. You need to update the ij.jar file using the instructions in the “Installing the Plugins” sections above.
–

Error:
Plugin Register_Images did not find required class:
com/drew/imaging/ImageProcesssingException

Explanation and solution:
This can due to two possible problems. Check to make sure the metadata-extractor-2.6.2.jar file is in the Fiji “jars” directory as explained in the “Installing the Plugins” sections above.

Another possible reason for this error is that there is a conflict with another plugin that uses some of the classes in the metadata-extractor-2.6.2.jar file. There is an ImageJ plugin called

Exif_Reader installed that has some of the same of the metadata-extractor classes in its jar file and this can cause a conflict with the photo monitoring plugins. To fix this problem you will need to remove the Exif_Reader.jar file from the “plugins” directory.