Flat Field Corrector

Users Guide

RHMG Software Tools Library

7/19/2011

Authors: Bill and Scott Werba
Developer: Scott Werba
Document ID: 3000-001
Revision: v1.0.0
Publication Date: July 19, 2011
Publication Status: Final

1.0   Introduction:

The **Flat Field Corrector** application is a program that will correct images due to lens distortions and products of uneven light gathering in an image sensor. These subtle aberrations are very apparent when an imaging system is used under long exposure conditions such as in astronomy and some medical applications.

2.0   Installation:

Installation of this application is quite straight forward, with some caveats. This program was written in C# using .Net Framework 2.0. It is a 32bit application. If you wish to just use the **Setup.exe** file as the install program, you must have .NET Framework 2.0 also installed on your computer. If not, the *Manifest files* (included in the install folder) and *FlatFieldCorrector* folder must be present along with the **setup.exe** file in the install directory and folder of your choice. The **flatfield.EXE** file will show up in the startup menu as "*Flat Field Corrector*" with a default icon. You are ready to use.

3.0   Initialization and Some Theory

To correct anomalous images like long exposures may present, there are two image assets to produce before a flat field correction can take place. These assets will characterize the current state of your image sensor and lens combination.

The first image asset to produce is a very underexposed or "dark current" image. It is required to get the imager noise metric. Dark currents represent a main source of noise in an image sensor. Both CMOS and CCD image sensors have this inherent problematic property. Characterizing the sensor noise will help compensate for potential noise patterns the sensor may imprint on an image. To get a "dark current" image, put your lens cap on your camera (close the camera shutter to ensure complete darkness) then take a picture. Name the image "***dark reference***.bmp".

The second image to produce is the actual "flat field" reference image. This image will reflect the actual measure of the illumination uniformity the surface of the image sensor will see. Normally, one would set the camera exposure to the usual brightness level you would set before a picture is taken, but is not the most efficient. During our testing and for better results, take an image at a slightly lower exposure level just so you can see some brightness gradient changes without clipping any white pixels. This can be subtle to detect, but is possible to take. In our demo, the image results are very obvious to see for purposes of this description. Name this image "***lens reference***.bmp". This image must be monochromatic, produced by a white diffused light source. Our '*lens reference*' images were done using a white back light over diffusing material. Alternatively, white balance your camera on a whitish wall that is widely and evenly lit. Adjust the exposure level down until you see lighting gradients if they exist. Move the camera around slightly to check for a difference between uneven lighting on the wall and actual gradients from the imager. If the gradient areas do not move with respect to the moving camera position, then take a picture and bump the exposure time a bit more,. Keep the bumped exposure setting for the rest of the images you plan to capture. This is your '*lens reference*'.

Choose the same frame resolution and pixel density you plan to use for all images taken. Camera settings should have the same exposure times and acquisition settings when creating the reference images as well as the actual images to correct. Save these reference images in a folder of your choice.

The actual images to be taken should use the same exposure setting as the '*lens reference*' image. They won't be optimal, but can be nicely compensated using the interface processing controls. For better pixel depth, capture the *lens reference* and input image at a longer exposure setting without incurring white clipping. Depth of correction will not be as pronounced.

4.0   User Interface

Using the Flat Field corrector application is quite easy. See **Figure 1.0** for reference.

Import the Lens Reference and Dark Reference images in the *Setup* input area shown. The images themselves will show on the image column to the right of the interface. The top image is the flat field image and the bottom is the dark current image.
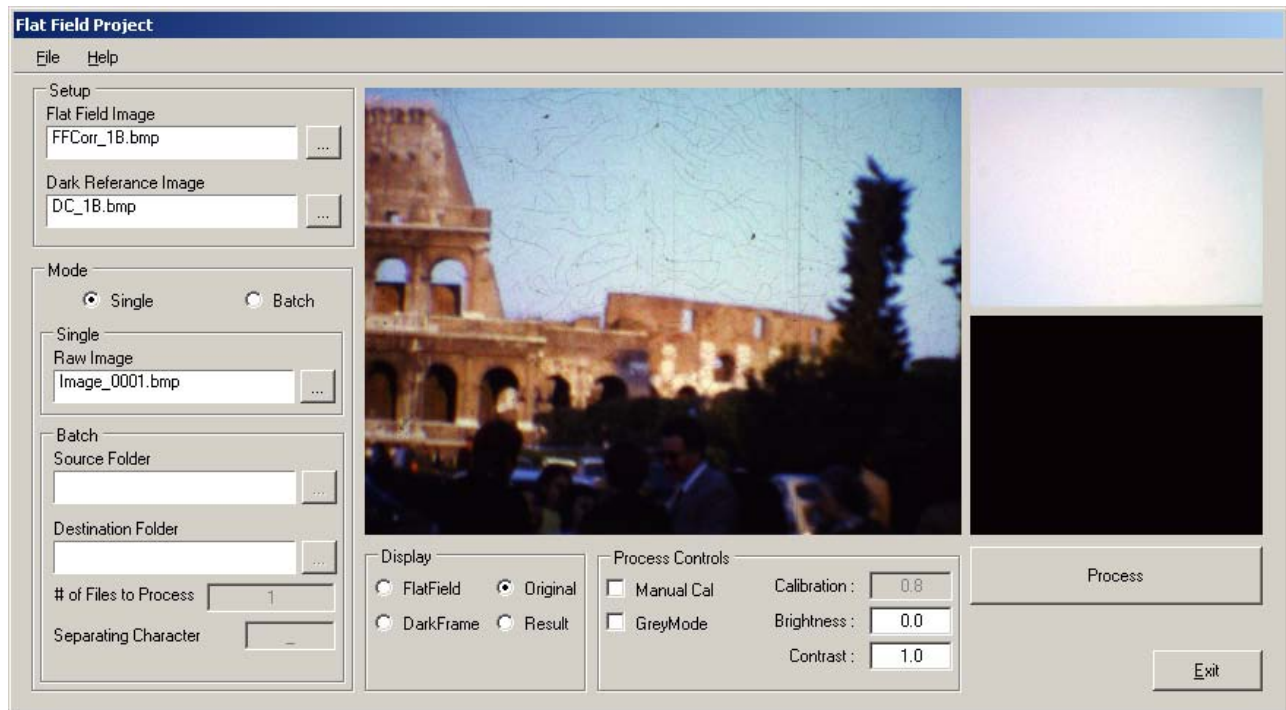

Figure 1.0   User Interface

The *Mode* section configures the program to run in single or batch processing modes. When *Single* mode is selected, images to be corrected are input here, processed one image at a time. When *Batch* mode is selected, an image sequence can be processed. In either case, input images will be processed against the two reference images to correct for any collective pixel aberrations. This application is multi-threaded, so when choosing batch mode, having a multi-core processing computer would be beneficial. While running, the progress bar may show to be stalled or slowing. This is normal. The application is still running and doing its thing.

If choosing *Batch* mode, select the first image in the sequence (*Raw Image*) to be processed, followed by selecting an image source folder and then a destination folder where the processed images will be written. Count the number of images to process from the source folder and enter that number into the "*Files to Process*" input field.

Batch based images must follow a file naming syntax. First section of defining sequential file names is in choosing an image name followed by a special character (separator), then a numerical sequence of any length, followed by the image type, in this case only 24bit uncompressed bitmaps are supported.

Put your choice of separator in the "*Separator Character*" input field on the interface. The special separator character field only accepts only one character. Choices -_, * ,&, #, %.

The *Display* section controls what is displayed in the main screen in the center of the interface. Select the '*Result*' option to see the processed output.

The *Process Controls* section controls how the program processes the images. All user input images are processed using RGB colour space. The reference images are converted to YUV colour space and only the Y component is used for calculations.

Manual when checked, will force the program to use the *Gain Boost* value in its calculations. This option can be used to extend the brightness of an image if needed, particularly when an image is dark to start with. Over extending this number may clip your whites, so be cautious in its use. The range of the gain number is from 0 to 1.5. When *Manual* is unchecked, the gain values are calculated and normalized internally.

Brightness and Contrast adjustments can be used to improve the overall depth of the corrected image. These controls are available at all times.

Once all files have been loaded or referenced and controls set, click the "*Process*" button to begin the correction process. The *Process* button will only be enabled when all inputs and selections have been made.

5.0   Specifications

Image input	24 bit uncompressed bitmaps (BMP), any frame resolution
Image Output	conforms to input
Max batch files	unlimited
Separator Character	suggested --- &, _, *, $, #
Gain Boost	0 to 1.5, steps of .01
Contrast	0 to 10, in steps of 1
Brightness	+/- 128 in steps of 1

6.0   Minimum System Requirements

Windows XPpro 32bit, Vista, Windows 7 (32bit or 64bit)
.NET Framework 2.0 installed
CPU: single core Pentium 4 (batch processing will be slow)
RAM: 512MB

7.0   Recommended System Requirements

CPU: Core 2 Duo (allows for multi-threaded operation for faster batch processing)
2.4GHZ or higher
RAM: 1024MB

8.0   Support and Contact

No technical support will be available for this product at this time. The supplied documentation will describe as much detail as needed for a great percentage of users.

9.0   License

There is no software license attached to this product or have open source restrictions. You can do what you want with it.