

Helpdesk XIMEA

Portal > Knowledgebase > Camera Features > Flat Field Correction

Flat Field Correction

Support SK - 2026-01-26 - in Camera Features

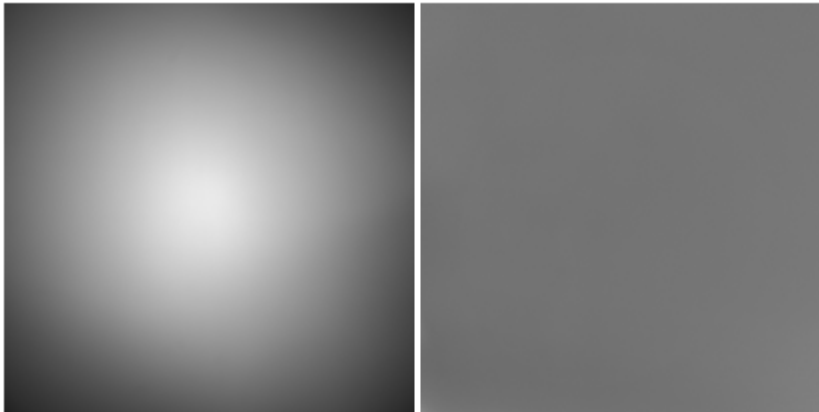
https://www.ximea.com/support/wiki/allprod/flat_field_correction

Flat Field Correction¹

[Flat field correction](#) (FFC or shading correction) is a method to remove non-uniformities in the image caused by different sensitivities of the pixels and by distortions caused by optics. For XIMEA cameras this correction is applied in the image processing part of the [image data flow diagram](#) (it is performed on the host computer thus depending on the image size and CPU performance enabling it may cause increased processing time).

In order for the xiapi to calculate the gain and offset coefficients for each individual pixel, it is necessary to load **calibration images** (1 dark image and 1 mid-saturated image) before applying FFC in **RAW8** or **RAW16** format depending on desired output bit depth. To obtain optimal results, camera should be in the **same setup** (lens, device output bit depth, gain, ROI, downsampling, Zero ROT, light conditions...) during acquisition of calibration images and while using FFC.

Compare of camera output with shading (left) and after FFC enabled (right):



Acquisition of calibration images¹

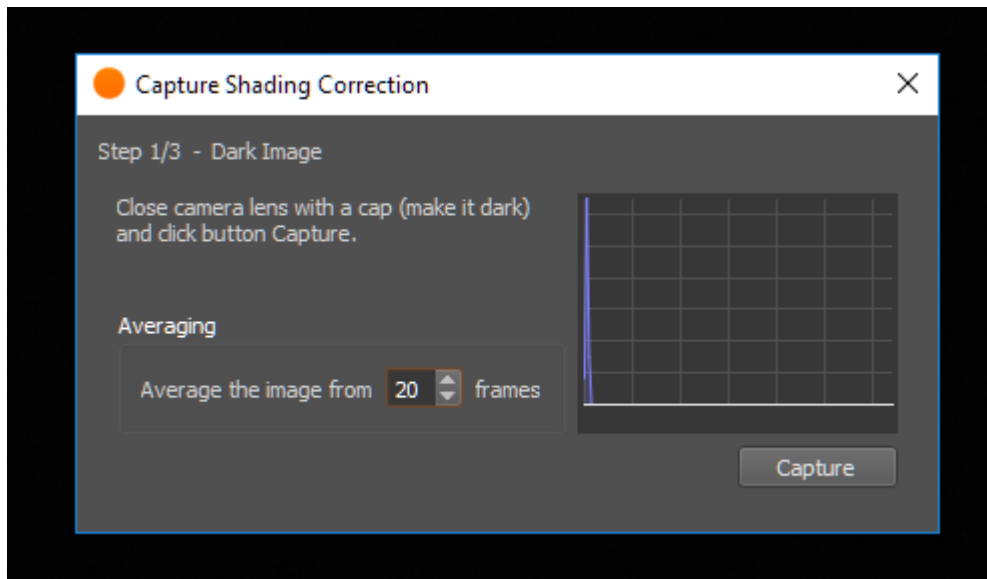
The easiest way to acquire calibration images is by using **CamTool** guide:



Dark Image

Close camera lens with a cap (make it dark) and click button Capture.

Average the image from set number of frames to suppress dynamic noise.

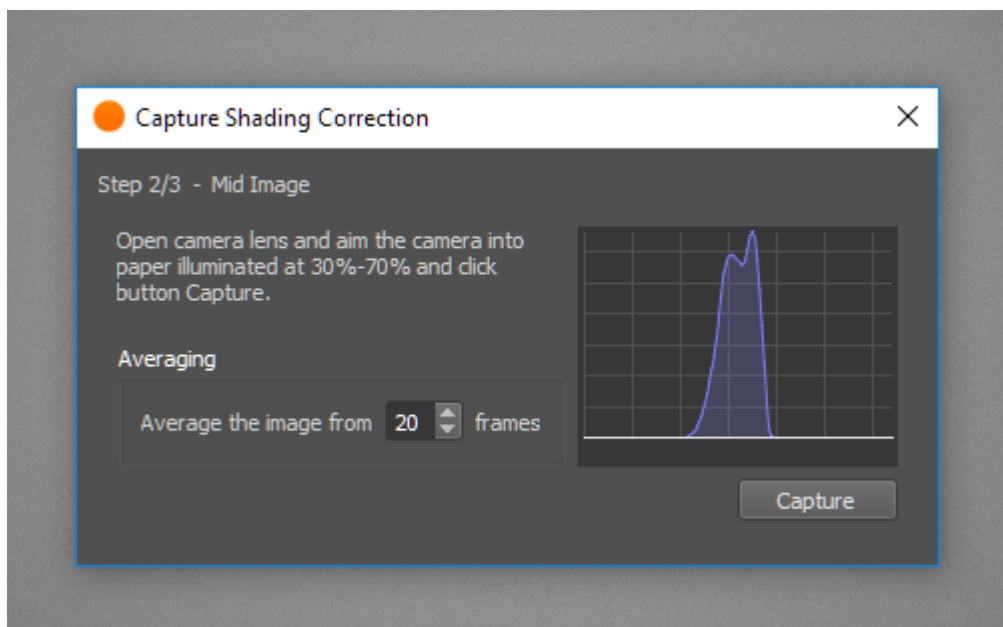


Mid-saturated image

Open camera lens and aim the camera into paper illuminated at 30-70% and click button Capture.

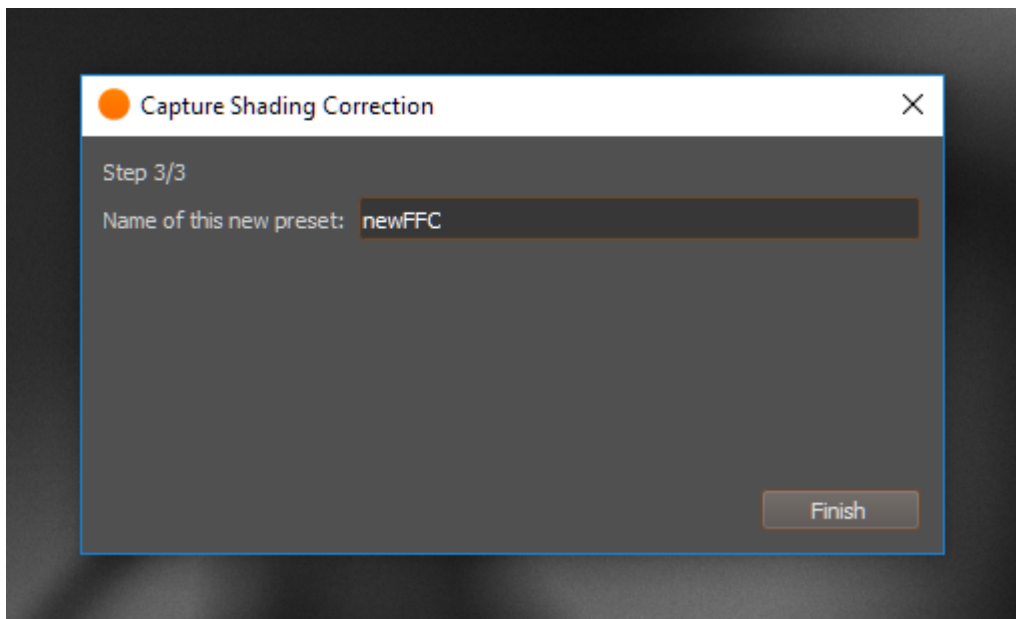
Average the image from set number of frames.

Image should be focused out of captured plane to reduce dust or other details in the scene.

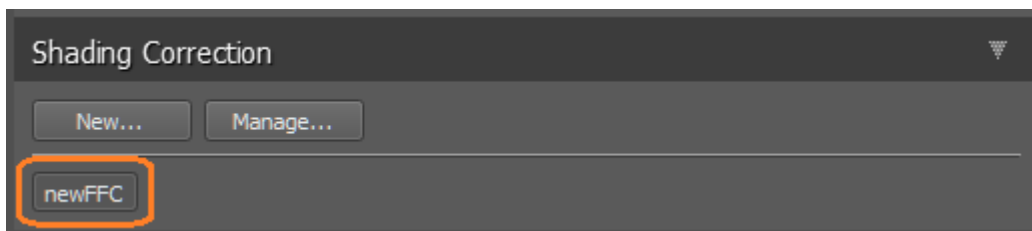


Save TIFF files

- Save the new preset how it be displayed in CamTool

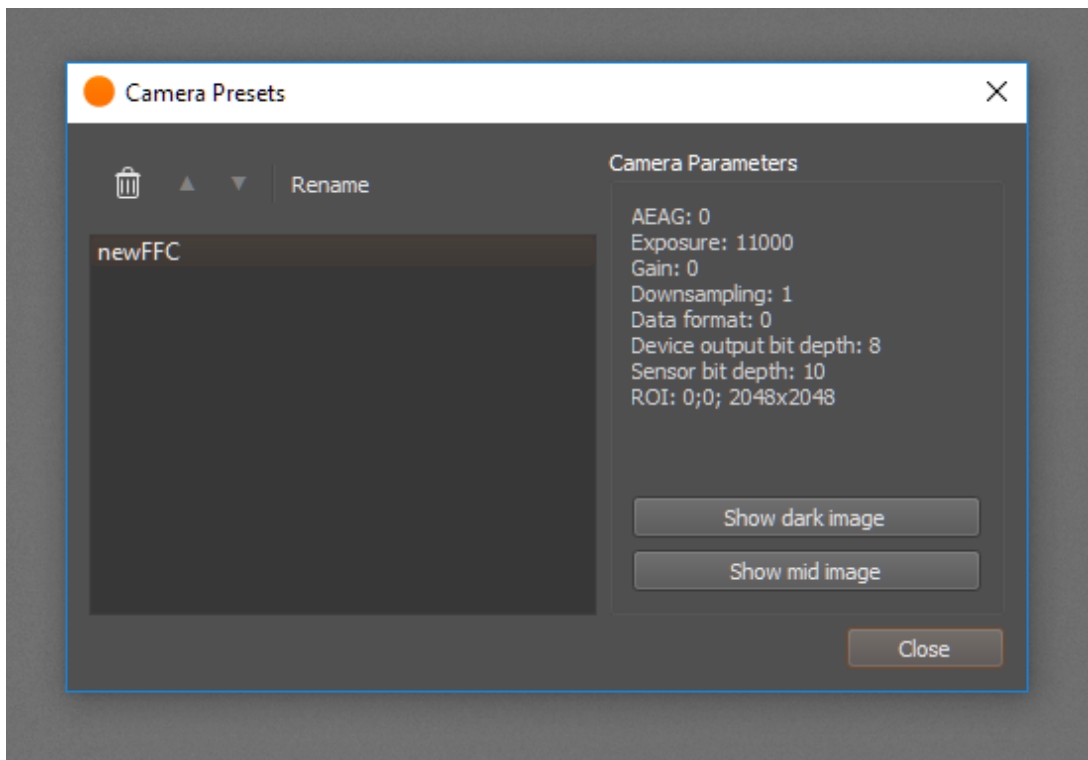


- To verify calibration, FFC can be also enabled in CamTool by clicking on created preset

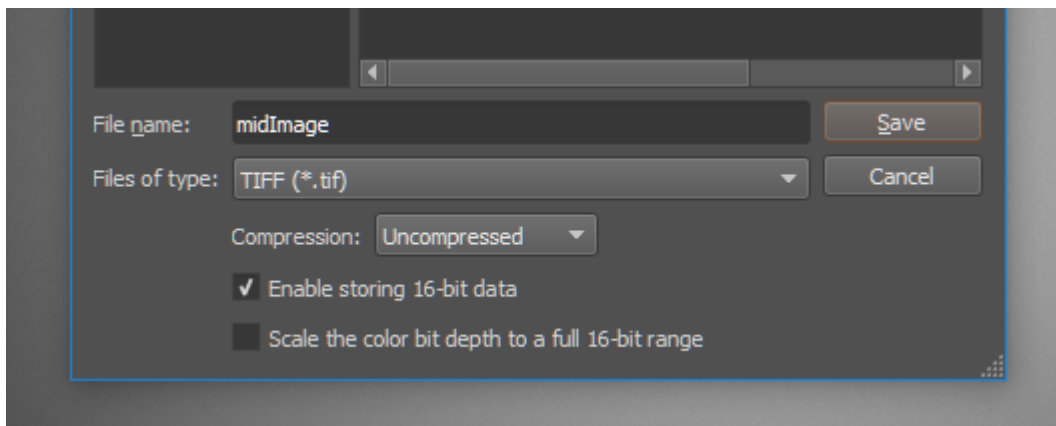


In case of any issue, please double check that all dependent camera parameters (device output bit depth, ROI, downsampling, Zero ROT, exposure, gain...) are in the same setup as during acquisition of calibration images

- To see calibration images and be able to save them, click on *Manage*, then choose preset from list.
Camera parameters for that particular preset will be displayed.
Click on show dark/mid image



- When images are shown in CamTool it is possible to **save** them by clicking on save icon in top toolbar.
Images should be saved in **uncompressed TIFF format** to use in API.



Otherwise, calibration images from CamTool are stored in hidden AppData folder. It can be opened in Windows *Run* application by command:

```
%LOCALAPPDATA%\xiCamTool\shading
```

Calibration images might be also acquired in any other tool as long as the output is in TIFF format and with the same camera setup.

Applying FFC in xiAPI

xiAPI command sequence:

1. set all dependent camera parameters (device output bit depth, ROI, downsampling, Zero ROT, exposure, gain...) to be in the same setup as during acquisition of calibration images
2. load dark image - [XI_PRM_FFC_DARK_FIELD_FILE_NAME](#)
size - size of the file name - `strlen(file_name)`
3. load mid-saturated image - [XI_PRM_FFC_FLAT_FIELD_FILE_NAME](#)
size - size of the file name - `strlen(file_name)`
(place both image files into project folder)
Note: Use the same image file for this parameter as for [XI_PRM_FFC_DARK_FIELD_FILE_NAME](#) for dark-field correction only. Processing will subtract the dark image only while using the unity (1.00) gain for correction.
4. enable [XI_PRM_FFC](#)

Sample code:

```
xiSetParamInt(0, XI_PRM_NEW_PROCESS_CHAIN_ENABLE, XI_ON); // MU, MQ, MD  
camera families
```

```
xiOpenDevice(0, &xiH);
```

```
// set dependent camera params to same values as during calibration
```

```
xiSetParamString(xiH, XI_PRM_FFC_FLAT_FIELD_FILE_NAME,  
"midImage.tif", strlen("midImage.tif"));
```

```
xiSetParamString(xiH, XI_PRM_FFC_DARK_FIELD_FILE_NAME,  
"darkImage.tif", strlen("darkImage.tif"));
```

```
xiSetParamInt(xiH, XI_PRM_FFC, 1);
```

In [FFCdemoWithOpenCV.cpp](#) is FFC demonstrated in OpenCV+xiAPI example. FFC might be enabled or disabled by pressing any key while program is running.