



SVCam-ImgCorrTool

Version 0.1

Software manual

Purpose of this document

This document explains how to use the SVCam-ImgCorrTool. The software is a software tool to adjust flatness, shading and defect pixels with some SVCam cameras from SVS-Vistek. Please note, correction of shading and defect pixel is dependant on the camera model (please refer to data sheet of your camera model to see the camera's features). The aim is to make the camera to deliver a flat image in your final application despite lens shading or pixel defects..

Targeted audience

Experienced users in machine vision. You should have knowledge in using command line operation (see -> Examples).

Content

1	General	4
1.1	Camera driver	4
1.2	General idea	4
1.3	Procedures	4
1.3.1	Create a shading map and upload it to the camera	4
1.4	Activate user defined shading map	5
2	Parameter description	6
3	Examples	9
3.1	Example – create and upload shading map	9
3.2	Example – create defpix map and mark into image.....	9
3.3	Example – upload defectpixel map using GenTL	9
3.4	Example – upload defect pixel map using clser.....	10
4	Appendix	11
4.1	Useful batch files	11
4.1.1	Shading-drop.bat	11
4.1.2	Shadingmap-create.bat.....	11
4.2	Cameras to connect with CLSER driver	11

1 General

1.1 Camera driver

With the new software kit SVCamKit 2.5.0 including SVCapture 2.5.0, ConvCam5 is obsolete. You might use SVCapture to modify the adjustments in the GenICam tree instead.

Please note, depending on camera model and firmware version you need to connect with the cti driver file of SDK 2.5.0 or with the --clser protocol. SVCapture 2.5.0 provides a cti file to access cameras with GigE or Camera Link interface. (see 4.1 for determining whether you connect via CLSER or CTI driver)

When uploading a map into the camera, please make sure SVCapture or ConvCam has closed its connection to the camera.

1.2 General idea

Svcam_imgcorrtool.exe is a windows tool to calibrate the camera for a flat image.

Svcam_imgcorrtool.exe uses images or image-maps to identify and modify final image output. The correction inside the camera has several benefits compared to post processing in the final image.

Basically, a raw (uncorrected) image is taken. Svcam_imgcorrtool.exe will do a research of the raw image and generate a map of coefficients („map“) how to adjust the raw image in the camera to have a flat image as result. The map will run the sensor with different gains depending on the content of the shading map and its source image respectively. This map can be loaded into the camera. The user defined shading map in the camera has to be activated then.

1.3 Procedures

You might want to look to 3. Examples on page 9 for examples how to use the tool. Please look as well to 4.1 for useful batch files.

1.3.1 Create a shading map and upload it to the camera

Task

The shading tool should correct the shading, generated by lens shading or a non-homogenous lighting situation

Prerequisites

Your software has a batch file called 'shading-drop.bat'. It does the job of receiving the homogenous image, create the shading map and upload it to the camera.

Procedure

(1) Take an image with a homogenous background, no object. The background should have an uniform colour / brightness.

(2) Save the image as 8 bit monochrome image (original)

you might drop this original image now on Shading-drop.bat (see 4.1.1) and you are finished

or

(3) Create from this original image a shading map

(4) Upload the shading map into the camera

(5) Activate the shading map in the camera

All these steps can be done separately or with one command as in example 3.1.

1.4 Activate user defined shading map

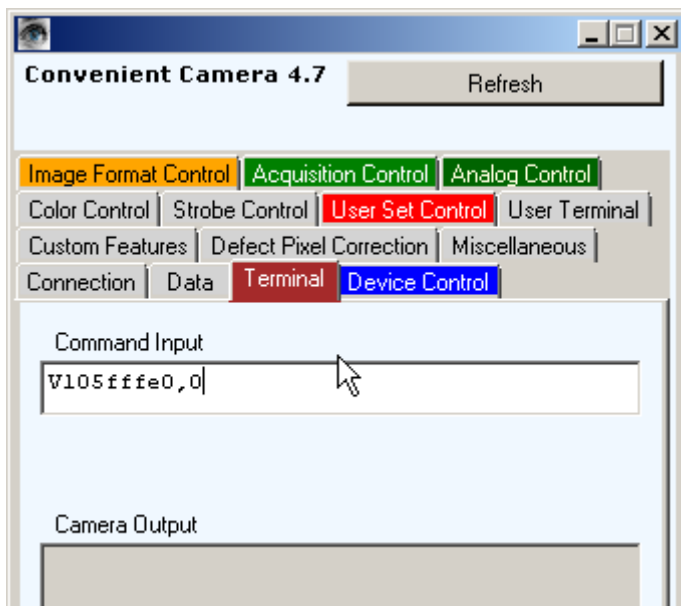
With SVCapture, you might use the GenICam tree to activate / deactivate shading correction.

With ConvCam4, use command input for activation / deactivation. Commands to use:

Shading disable / enable ConvCam4 command

Disable V105ffe0,0

Enable V105ffe0,1



2 Parameter description

Svcam_imgcorrtool.exe works as a commandline program. Parameters as explained below might be added as required. The length of parameters in your commandline may exceed the size of your commandline window. Do not put any newline into your commandline, just keep going and continue adding parameters on the next line(s)

--load_img <filename> [<filename>]

Load image(s) from file(s). Multiple images are averaged into one image. This image is used for creating defpix and shading maps.

Note: This version only supports 8-bit grayscale bitmaps.

--create_defpix_map [<threshold> [<width>]]

Calculate defect pixel map. Threshold parameter defines the threshold in graylevels whether a pixel is detected as defect inside an averaged area of size width x width. The created map is concatenated the tools internal map.

Defaults:

Threshold → 16

Width → 128

--clear_defpix_map

Clear the internal defect pixel map.

--load_defpix_map <filename>

Load a defect pixel map from file and concatenate to the tools internal map.

--save_defpix_map <filename>

Save a defect pixel map to file.

--apply_defpix_map/mark_defpix_map <filename-in> <filename-out>

Apply or mark a given image with the tools internal map and save the corrected image to a file.

Note: This version only supports 8-bit grayscale bitmaps.

--download_defpix_map [<mapindex>]

Download a defect pixel map from camera to the tools internal map. The download data is concatenated.

Mapindex 0 – factory map

Mapindex 1 – customer map

Map index 1 is default.

--upload_defpix_map [<mapindex>]

Upload defect pixel map to camera and save. Map index 1 is default.

Mapindex 0 – factory map

Mapindex 1 – customer map

Note: Only mapindex 1 is valid thus factory map is write protected.

--create_shading_map

Calculate shading map. The shading map is multiplicative.

--clear_shading_map

Clear map.

--load_shading_map <filename>

Load shading map from file.

--save_shading_map <filename>

Save shading map to file.

--apply_shading_map <filename-in> <filename-out>

Apply shading map to image give in filename-in and save the corrected image to filename-out.

Note: This version only supports 8-bit grayscale bitmaps.

--upload_shading_map [<mapindex>]

Upload shading map to camera. Default map is at map index 0. The camera supports up to 3 maps.

--cti <filename>

Open GenICam Transport layer and Enumerate all interfaces.

--clser

Open clserxxx.dll as transport layer, enumerate all interfaces and use SVS-Vistek ASCII Protocol.

--if <interfacename>

Select interface and enumerate all available devices.

--connect [<devicename>]

Connect to device. If multiple devices are available the first or give device is getting connected.

3 Examples

If you connect to your camera (using latest driver with latest GenICam producer (-> cti file), the output of the connect might look as follows:

```
open_cti('c:\temp\sv_cl_tl_x64.cti')
NumInterfaces 4
Interface 0 - 'COM_Port#COM1'
Interface 1 - 'COM_Port#COM2'
Interface 2 - 'COM_Port#COM3'
Interface 3 - 'EURESYS#Euresys Grablink Full#0'
TLOpenInterface('EURESYS#Euresys Grablink Full#0') -> 0
NumDevices 1
Device 0 - 'SVS-Vistek#CameraLink#hr120MCL#1.1.0#99991'
IFOpenDevice('SVS-Vistek#CameraLink#hr120MCL#1.1.0#99991')
```

Please note, when opening the camera it is necessary to use the interface exactly with the name as shown (here interface3: 'EURESYS#Euresys Grablink Full#0') and printed. This string shows your grabber card, the connected camera, firmware version and serial number.

For correct connection to the camera please look on 1.1 Camera driver. In the examples as below, replace the -- if parameter with the correct -clser parameter if applicable.

3.1 Example – create and upload shading map

```
Svcam_imgcorrtool.exe --load_img "c:\temp\120_1.bmp" --
    create_shading_map
--save_shading_map "c:\temp\shading.map" --cti c:\temp\sv_cl_tl_x64.cti
--if "EURESYS#Euresys Grablink Full#0" --connect --upload_shading_map
```

3.2 Example – create defpix map and mark into image

```
Svcam_imgcorrtool.exe --load_img "c:\temp\120_1.bmp" --create_defpix_map
    12 --save_defpix_map c:\temp\120_1.map --mark_defpix_map
    "c:\temp\120_1.bmp" "c:\temp\120_1 mark.bmp"
```

3.3 Example – upload defectpixel map using GenTL

```
Svcam_imgcorrtool.exe --cti c:\temp\sv_cl_tl_x64.cti --if
    "EURESYS#Euresys Grablink Full#0" --connect --load_defpix_map
    c:\temp\test2.map --upload_defpix_map
```

3.4 Example – upload defect pixel map using clser

```
Svcam_imgcorrtool.exe --clser --if "EURESYS#Euresys Grablink Full#Port0"  
--connect --load_defpix_map c:\temp\test2.map --upload_defpix_map
```

4 Appendix

4.1 Useful batch files

All batch files below are useful and made for your convenience, removing the need for too many params on the command line. You might need to modify them regarding the camera connection (see 0). By dropping an image file from the explorer on the batch file, the appropriate action will be taken

4.1.1 Shadingmap-create.bat

Creates a shading map from a dropped image and saves it as shmap.map.

```
@echo filename %1
SVCam_ImgCorrTool.exe --load_img %1 --create_shading_map
--save_shading_map shmap.map
pause
```

4.1.2 Shading-drop.bat

Uploads the shading map to the camera. In the explorer drop the shading reference monochrome image on the batch file.

```
@echo filename %1
SVCam_ImgCorrTool.exe --load_shading_map %1 --clser --if
"EURESYS#Euresys Grablink Full#Port0" --connect --upload_shading_map
pause
```

4.2 Cameras to connect with CLSER driver

All cameras with ConvCam need to be connected with the CLSER driver. Cameras where you can use SVCapture, you might use the --if parameter and connect via cti file. Note, support of defect pixel correction or shading maps is dependant of your camera model.

model	driver
EXO CL	cti
EVO CL	clser
HR (not hr25000 and hr120)	clser
HR (25000 and 120)	cti