Tools for DSO camera calibration

In order to run the DSO code, we need 3 calibration files for the **global shutter camera** that we are using:

1) Calib file, which is a geometric camera calibration file. It has 4 options: **(these options can be found in the following github repo READ ME section with more explanation https://github.com/JakobEngel/dso)**
   - **Calibration File for Pre-Rectified Images**

```
Pinhole fx fy cx cy 0
in_width in_height
"crop" / "full" / "none" / "fx fy cx cy 0"
out_width out_height
```

   - **Calibration File for FOV camera model:**

```
FOV fx fy cx cy omega
in_width in_height
"crop" / "full" / "fx fy cx cy 0"
out_width out_height
```

   - **Calibration File for Radio-Tangential camera model**

```
RadTan fx fy cx cy k1 k2 r1 r2
in_width in_height
"crop" / "full" / "fx fy cx cy 0"
out_width out_height
```

   - **Calibration File for Equidistant camera model**

```
EquiDistant fx fy cx cy k1 k2 r1 r2
in_width in_height
"crop" / "full" / "fx fy cx cy 0"
out_width out_height
```

2) Gamma calibration file pcalib.txt, containing a single row with 256 values.
3) Vignette calibration file, is a monochrome 16bit or 8bit image containing the vignette as pixelwise attenuation factors.

In order to get calibrations files 2 and 3, we will use this github repo:

https://github.com/tum-vision/mono_dataset_code

Where we record several images from a camera and give them as input to the responseCalib() for pcalib.txt and vignetteCalib for vignette.png.

Installation of the Mono Dataset Code

Following the steps in the READ ME section of the link:

1) Already installed Eigen and OpenCV.
2) Already installed Ziplib.
3) Install aruco marker detection (for vignette calibration):

   It can be found in mono_dataset_code thirdparty folder.

   I unzipped it, then:
   cd aruco-1.3.0

   Mkdir build

   Cd build

   Cmake ..

   Make

   Sudo make install

Though I needed to add :

#include <opencv2/imgproc/imgproc.hpp>

In the aruco_text_board_gl_mask.cpp and aruco_test_board_gl.cpp Cpp files.

Afterwards, in the mono_dataset_code directory I performed:

Mkdir build

Cd build

Cmake ..

Make

Video to frames and timestamps

Now, in order to use the 2 functions needed for calibration, I needed to give them as input a folder which contains images folder which contains all the video frames, and times.txt file which contains the time stamp of each frame and its exposure.

In order to achieve that efficiently, I have written my own python code that takes the following arguments:

- --video: The video I want to convert.
- --imageFolder: a path to empty images folder (will contain the frames)
- --timeStampsFile: a path to the times.txt file, if there is none it will creat one.

And output the following:

- Puts all the frames in the images folder
- Creates time stamps with **constant exposure** 1 for each frame.

This code can be found in:
Desktop/DSO/Mono_Dataset_Code/VideoToImageConverter

Response Calibration

After I used this code and acquired the images folder and the times.txt file, I navigated to the DSO/Mono_Dataset_Code/mono_dataset_code/build/bin folder and performed the following command:

./responseCalib <path to a folder that contains the images folder and times.txt file>

(The folder also needs to contain camera.txt file with the resolution set to the resolution of the images).

And it successfully created pcalib.txt file, which can be found in the following folder:
DSO/Mono_Dataset_Code/mono_dataset_code/build/bin/PhotoCalibResult

** Important notice: We need to verify the values in the pcalib.txt file are monotonically increasing (values from 0 to 255).

Vignette Calibration

After receiving a Segmentation Fault when running the ./vignetteCalib function, I decided to build the files again by using the "make" command (even though make was successful the first time) and it for some reason it is working now.

To use vignetteCalib we need to give it as input a path to folder which contains the following:

- Images folder (or zipped images folder).
- times.txt file which again contains the timestamp for each frame and its exposure time.
- Camera.txt, which is camera calibration (we can choose 4 options). Keep in mind we need to change the image resolution in it according the resolution of our frames.
- pcalib.txt, which is the file we receive from the response calibration.

Finally, we navigate to Desktop/DSO/Mono_Dataset_Code/mono_dataset_code/build/bin/vignetteCalibResult and get the vignette.png from it. Which is the vignette calibration for the DSO algorithm.