

facebook

Surround 360 Run

24.04.16

(https://docs.google.com/presentation/d/1Ji6pteoc8ca9dmvOFFtJtB87eUW2krZ9Wp1Jm6CnAsM/edit?usp=drive_link)

Sample Data 테스트

[Sample Data 테스트]

From Surround360/README.md

Sample Data

We provide a sample dataset for those who are interested in testing the rendering software without first building a camera.

* "Palace of Fine Arts Take 1" - 2 frames - (337.4MB)

* Raw data: http://surround360.hacktv.xyz/sample/sample_dataset.zip

* Sample result: https://s3-us-west-2.amazonaws.com/surround360/sample/sample_result.zip

* NOTE: The Surround 360 hardware and camera control software records to a RAW binary format which needs to be unpacked to get to the individual images. In this smaller dataset, the 'unpack' and 'arrange' steps of the pipeline have already been run (see run_all.py), so you do not need to run them again.

→ 단순히 raw 와 isp_out 이미지 확인용 파일들임

* "Facebook Building 20" - 190 frames - (21.15GB)

* Binary file 1: [\(10.76GB\)](https://s3-us-west-2.amazonaws.com/surround360/github_samples/test/0.bin)

* Binary file 2: [\(9.56GB\)](https://s3-us-west-2.amazonaws.com/surround360/github_samples/test/1.bin)

* Sample result:

[\(823.6MB\)](https://s3-us-west-2.amazonaws.com/surround360/github_samples/test/render.zip)

* NOTE: the render directory contains calibrated config files and two rendered frames

* The file NOTES.txt contains sample commands to process the binaries

→ ~/Desktop/test 폴더를 생성 후, 0.bin, 1.bin, render/ 를 위치시킨 후, 아래 명령어를 실행

```
python ~/cubox/Surround360/surround360_render/scripts/run_all.py \
--data_dir ~/Desktop/test --dest_dir ~/Desktop/test/render \
--start_frame 0 --frame_count 30 \
--quality 6k --steps_unpack --steps_render --steps_ffmpeg \
--enable_top --enable_bottom --enable_pole_removal \
--cubemap_format video --cubemap_height 0 --cubemap_width 0 --verbose --save_raw \
--save_debug_images
```

print "Terminating process: " 에서 에러가 발생하면 print("Terminating") 식으로 소스를 수정
ValueError: can't have unbuffered text I/O 에러가 발생하므로
open(dest_dir + "/runtimes.txt", 'w', 0)에서 마지막 0 인자를 제거

Sample Data (1)

"Palace of Fine Arts Take 1" - 2 frames - (337.4MB)

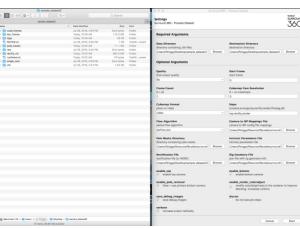
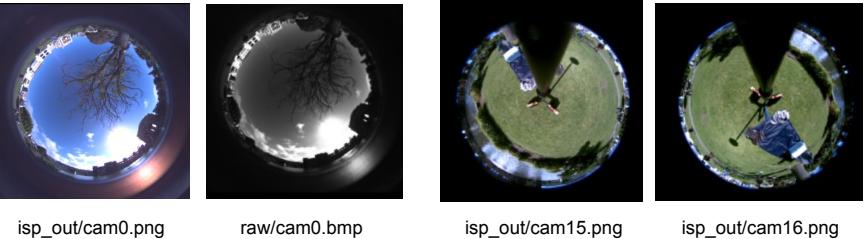
→ 단순히 raw 와 isp_out 이미지 확인용임

[sample_dataset.zip]

```
cube_frames
eqr_frames
logs
pole_masks
raw
single_cam
vid
```

```
000000
  flow
  flow_images
  isp_out
    cam0.png
    cam1.png
    ...
    cam16.png
  projections
  raw
    cam0.bmp
    cam1.bmp
    ...
    cam16.bmp
```

1st frame



eqr_000000.png
(sample_result.zip)

gui_config.png
(sample_result.zip)

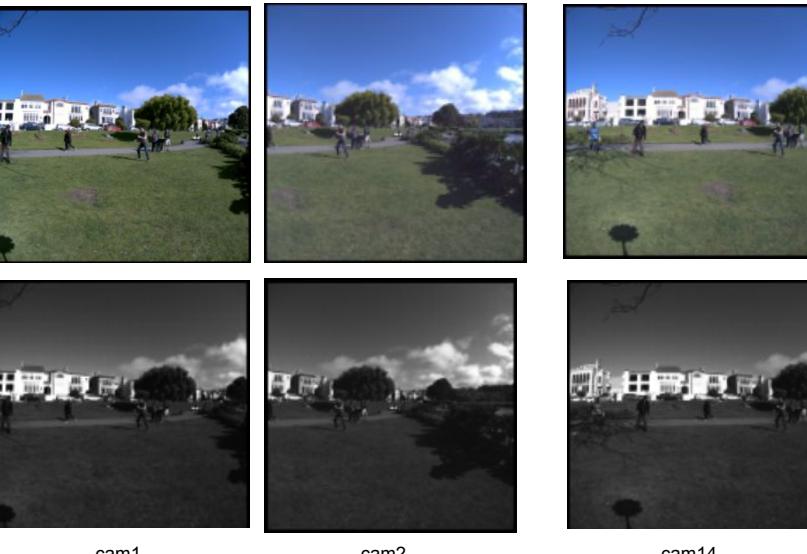
// surround360_sample/sample_dataset/NOTES.txt

This sample dataset provides **2 frames of RAW data** captured with the Surround360 rig. This directory structure reflects the state after the "unpack", "arrange", and "isp" steps of the pipeline have already been run. This means you do not need to run these steps again. We also provide the RAW files which are the inputs to the ISP, which you may use as input to run the "isp" step of the pipeline as well.

NOTE: we adjusted the black levels that are defined in batch_process_isp.py before running the ISP, which changes the brightness of the images. If you run the "isp" step without adjusting black levels, the contents of your /isp_out folders will differ from the examples we provide here. To get started quickly, we recommend skipping the ISP step and using the outputs in this example dataset.

NOTE: it is strongly recommended that you read CALIBRATION.md before attempting to process this data, especially the "Rectification" section. If you don't compute rectification parameters for this camera array and instead use the defaults, the results will not provide comfortable perception of 3D in stereo.

→ ISP 의 입력으로 사용된 RAW 이미지(e.g., raw/cam0.bmp)와 unpack, arrange, isp 과정을 거쳐 생성된 이미지(e.g., isp_out/cam0.png)만 존재함. 따로 돌려볼게 없는 데이터임



// Surround360/NOTES.txt

NOTE: The Surround 360 hardware and camera control software records to a RAW binary format which needs to be unpacked to get to the individual images. In this smaller dataset, the 'unpack' and **'arrange'** steps of the pipeline have already been run (see run_all.py), so you do not need to run them again.

Sample Data (2)

"Facebook Building 20" - 190 frames - (21.15GB)

[~/Desktop/test]

- 0.bin (10.76GB)

- 1.bin (9.56GB)

- render (823.6MB)

- config

- isp

- 15636976.json

- 16237674.json

- ...

- 16283829.json

- camera_rig.json

- eqr_frames

- eqr_000000.png

- eqr_000001.png

- pole_masks

- cam15.png

- cam16.png

- raw

- cam0

- 000000.png

- 000001.png

- cam1

- cam2

- ...

- cam16

NOTE: the render directory contains

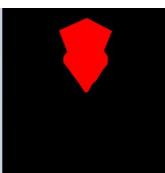
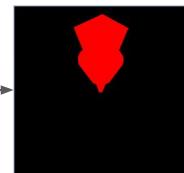
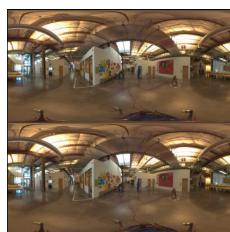
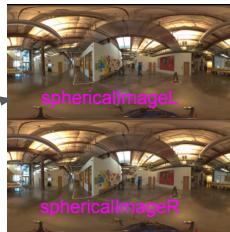
calibrated config files and two

rendered frames

The file NOTES.txt contains sample

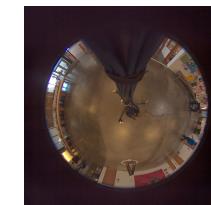
commands to process the binaries

render.zip



cam15.png

cam16.png



cam0/000000, 1.png
(2048 x 2048)

cam1/000000, 1.png

cam2/000000, 1.png

cam14/000000, 1.png

cam15/000000, 1.png

cam16/000000, 1.png

run_all.py 실행시 raw/ 가 생성되며, rgb/15636976 폴더도 생성된다.
(기존의 rgb/cam0 등의 폴더는 비교용으로 들어 있는 것으로 보임)

~/Desktop/test/render/eqr_000000_stereo_comparison.gif 와
test_259927_6k_TB.mp4 등의 동영상 파일도 생성된다.

두번째 실행시에는 raw/ 폴더를 지우고 진행해야 한다.
(raw/ 폴더가 이미 존재시 에러 발생함)

Sample Data 테스트

[Tip]

vscode 는 code_1.64.2-1644445741_amd64 로 설치하도록 한다.

code_1.88.0-1712152114_amd64.deb 로 설치할 경우,

vscode 에서 폴더 오픈시 죽어 버리며 /usr/share/code/code . 식으로 폴더를 열어도 Python 실행을 시키면(디버깅 포함) 죽어버리고 있다.

vscode 에서 python 디버깅을 위해 최초에 한번 vscode Extensions 탭에서 Python extension 설치가 필요함. 추후 C++ 디버깅을 위해 C/C++ 확장도 설치

카메라 초기 설정

Configuring Synchronized Capture with Multiple Cameras - FLIRs

<https://www.flir.eu/support-center/iis/machine-vision/application-note/configuring-synchronized-capture-with-multiple-cameras/>

사이트 로그인 계정은 skc0833@cubox.aero / s12~!

What is Synchronized Capture?

다수의 카메라가 동시에 exposing 을 시작하는 것을 의미함(동시는 수 microseconds 이내를 의미)

primary camera 의 strobe 를 이용해 1대의 primary camera 로 다수의 secondary

카메라에서 동시에 캡쳐가 가능함(이미지 캡쳐 시작시에 strobe 가 발생함)

external hardware trigger 를 사용하는 방법도 존재하지만 여기서 다루지는 않음
(이경우, pull-up resistor 는 사용되지 않으므로 pull-up resistor 설정은 무시됨)

Configuring Synchronized Capture

Step 1—Connect the Cameras

첫번째로 카메라 간의 physical link 를 설정해줘야 한다.

GPIO Trigger Cable 연결(Grasshopper3 (GS3) 카메라)

--> primary camera 빨간선(3번, output) + secondary camera 초록선(4번, input),

primary 갈색선(5번, ground) + secondary 갈색선(5번, ground)

Step 2—Configure the Cameras (불필요)

SpinView 사용시 ==>

1) primary 카메라에 대해서 output line 설정

Features 탭 / Grasshopper3 GS3-U3-41C6C / Digital IO Control 을 선택 후

Line Selection 에서 Line 2(0 base 로 3번 핀?) 를 선택(default Line 0) 하면

Line Mode 를 Output 로 선택(default Input),

Line Source 를 ExposureActive 로 선택(default empty)

이 화면에 다시 들어올 경우, Line 0 가 표시되고 있지만 Line 2 로 바꿔보면 설정돼 있음

(Optional) user set 에 저장:

Features 탭 / Grasshopper3 GS3-U3-41C6C / User Set Selector 에서 User Set 1 or

User Set 2 로 저장이 가능함

--> User Set Save/Load 항목의 Execute 버튼으로 저장 후, 로드

좌상단 카메라 리스트에서 선택시, Features 탭, GPIO 탭의 설정값이 변경되는 듯함

디폴트 설정을 secondary 카메라로 설정이 편리해 보임(User Set 2)

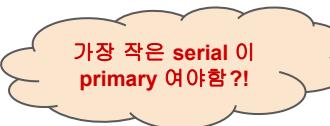


DIAGRAM	COLOR	PIN	FUNCTION	DESCRIPTION
	Black	1	IO	Opto-isolated input (default Trigger in)
	White	2	O1	Opto-isolated output
	Red	3	IO2	Input/Output/serial transmit (TX)
	Green	4	IO3	Input/Output/serial receive (RX)
	Brown	5		
	Blue	6		
	Orange	7		
	Yellow	8		

2) secondary 카메라에 대해서 trigger source 를 설정

GPIO 탭에서 Trigger Source 콤보에서 Line 3(0 base 로 4번 핀?) 를 선택

Trigger Mode 콤보에서 On 선택

Trigger Overlap 콤보에서 Read Out 선택

primary 카메라의 경우, Trigger Mode 콤보만 Off 선택 &

Features 탭 / Grasshopper3 GS3-U3-41C6C / Digital IO Control 는 Line 2 를 Input 으로 설정

꼭, Features 탭 / Grasshopper3 GS3-U3-41C6C / User Set Control 화면에서 User Set Selector 를 User Set 2 를 선택하고? User Set Save 버튼을 클릭해줘야 디바이스에 기록된다? User Set Default 도 User Set 2 를 선택

그래야 카메라쪽 usb 를 뗐다 다시 꽂아도 설정이 유지된다!

preview 재생을 꺼줘야 User Set Save 버튼이 활성화됨에 주의!!!

surround360_camera_ctl_ui 테스트(카메라 촬영)

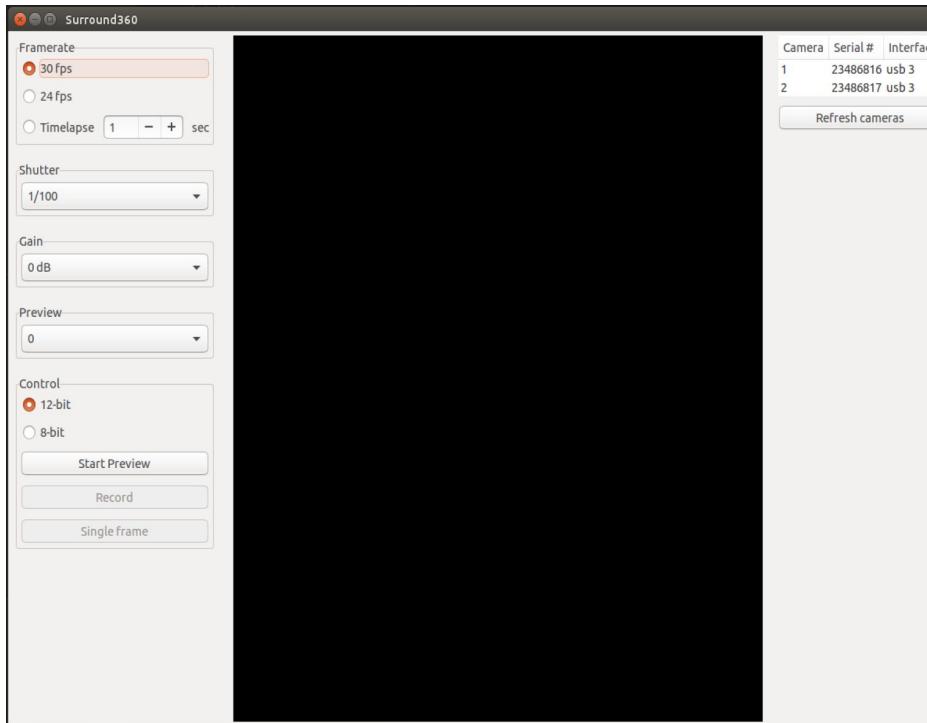
vscode에서 최초 실행시 "type": "cppdbg"로 설정했을 경우, Configured debug type 'cppdgg' is not supported. 에러가 발생함.

Install cppdbg Extension 버튼을 클릭하면 무반응

다시 C/C++, C/C++ Extension 확장팩을 제거 & 재설치해줘야 해결됨(그냥 설치만 해서는 해결 X)

또는 "type": "gdb"로 설정해도 실행은되나 중단점이 찍히질 않게 됨(보통 cppdbg로 설정하는듯)

CameraControlUI 프로그램 종료시에 crash 되고 있으나, 동작에 문제는 안됨(원인미상)



빌드는 surround360_camera_ctl_ui 경로에서

```
cmake -DCMAKE_BUILD_TYPE=Release -DHALIDE_DIR=$HOME/cubox/Halide/cmake_build  
make -j32 // 디버깅을 위해서는 -DCMAKE_BUILD_TYPE=Debug로 빌드
```

```
// /home/cubox/cubox/Surround360/surround360_camera_ctl_ui/.vscode/launch.json
```

```
{
```

```
    // Use IntelliSense to learn about possible attributes.
```

```
    // Hover to view descriptions of existing attributes.
```

```
    // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
```

```
"version": "0.2.0",
```

```
"configurations": [
```

```
{
```

```
    "name": "CameraControlUI",
```

```
    "type": "cppdbg",
```

```
    "request": "launch",
```

```
    "justMyCode": false,
```

```
    "stopAtEntry": false,
```

```
    "externalConsole": false,
```

```
    "cwd": "/home/cubox/cubox/Surround360/surround360_camera_ctl_ui",
```

```
    "program": "bin/CameraControlUI",
```

```
    "args": [
```

```
    ],
```

```
    "MIMode": "gdb",
```

```
    "miDebuggerPath": "gdb",
```

```
    "setupCommands": [
```

```
{
```

```
    "description": "Enable pretty-printing for gdb",
```

```
    "text": "-enable-pretty-printing",
```

```
    "ignoreFailures": true
```

```
}
```

```
]
```

```
}
```

```
]
```

```
}
```

Unpacker (bin to tiff, png) 테스트

```
// /home/cubox/cubox/Surround360/surround360_render/.vscode/launch.json
{
  "version": "0.2.0",
  "configurations": [
    {
      "name": "Unpacker (bin to tiff, png)",
      "type": "cppdbg",
      "request": "launch",
      "stopAtEntry": false,
      "externalConsole": false,
      "cwd": "/home/cubox/cubox/Surround360/surround360_render",
      "program": "/home/cubox/cubox/Surround360/surround360_render/bin/Unpacker",
      "args": [
        // copied from
        /home/cubox/cubox/Surround360/surround360_render/res/config/isp/cmiosis_sunex.json
        // "--isp_dir", "/home/cubox/cubox/Surround360/cubox/color_calibration/isp",
        "--isp_dir", "NOT_EXISTING_DIR", // if --isp_dir 가 존재하지 않는 경로이면 tiff
        파일까지만 생성한다(png 대비 생성 속도가 훨씬 빠름)
        "--bin_list", "/home/cubox/Desktop/0.bin,/home/cubox/Desktop/1.bin",
        // bin_list is saved to output_raw_dir/23486816/000000.tiff and
        unpacker@output_dir/cam0/000000.png (renamed from 23486816 to cam0 dir)
        // needs existing unpacker@output_dir, unpacker@output_raw_dir directory
        "--output_raw_dir",
        "/home/cubox/Desktop/Surround360_output/unpacker@output_raw_dir",
        "--output_dir", "/home/cubox/Desktop/Surround360_output/unpacker@output_dir",
        ],
        "MIMode": "gdb",
        "miDebuggerPath": "gdb",
        "setupCommands": [
          {
            "description": "Enable pretty-printing for gdb",
            "text": "-enable-pretty-printing",
            "ignoreFailures": true
          }
        ]
      ]
    }
  ]
}
```

빌드는 surround360_render 경로에서
cmake -DCMAKE_BUILD_TYPE=Release -DHALIDE_DIR=\$HOME/cubox/Halide/cmake_build
make -j32 // 디버깅을 위해서는 -DCMAKE_BUILD_TYPE=Debug 로 빌드
(이럴게 해줘야 breakpoint 에 걸리게 된다)

Calibration

```
// /home/cubox/cubox/Surround360/surround360_render/.vscode/launch.json
{
  "name": "[calibration] 0_color_calibrate_all.py",
  "type": "python",    "request": "launch",    "justMyCode": false,
  "cwd": "/home/cubox/cubox/Surround360",
  "program": "surround360_render/scripts/color_calibrate_all.py",
  "args": [
    // TODO: color patch not all found so below error occurs and no
    ~/Desktop/color_calibration/isp/xxx.json created
    // E0416 14:10:04.798012 24733 TestColorCalibration.cpp:136] Number of patches found
    (0) different than expected (24)
    // [Errno 2] No such file or directory:
    '/home/cubox/Desktop/color_calibration/output/23486816/intercept_x.txt'

    // Do capture and Unpacker to
    ~/Desktop/Surround360_output/unpacker@output_raw_dir/23486816/000000.tiff
    // and rename 23486816/000000.tiff to ~/Desktop/color_calibration/charts/23486816.tiff
    // color_calibrate_all.py read all .tiff files from data_dir/charts
    // and make output directory(includes debug infos), isp directory(includes color calibrated
    json(e.g, 23486816.json))

    // INPUT:
    // <data_dir>/charts/23486816.tiff
    // OUTPUT:
    // isp/23486816.json
    "--data_dir", "~/Desktop/color_calibration",
    // Default "--output_data_dir", "~/Desktop/color_calibration/output",
    "--illuminant", "D50",
    "--num_squares_w", "6",
    "--num_squares_h", "4",
    "--min_area_chart_perc", "0.5",
    "--min_area_chart_perc", "0.5",
    "--max_area_chart_perc", "40.0",
    "--black_level_hole",
    // "--save_debug_images",
  ],
}
```

// Macbeth Colorchecker Download
https://upload.wikimedia.org/wikipedia/commons/4/4f/Color_Checker.pdf

24개의 패치를 못 찾을 경우, 아래 에러와 함께 isp 파일이 생성되지 않게 된다.
Colorchecker 를 모니터 가득, 모니터 높이는 바닥에 낮춰서 카메라 가까이에 밀착시켜
촬영해야 어렵게 성공한다!

E0416 13:03:25.103821 19280 TestColorCalibration.cpp:136] Number of patches found (21)
different than expected (24)

Calibration

soft ISP 설정을 위한 필드값을 포함하고 있음

(1) Color Calibration --> 카메라별로 1장씩만 촬영!!!

MacBeth ColorChecker 와 SpyderCUBE 디바이스(optional ?!)가 있는 장면을 카메라별로 1장만 캡쳐후,
~/Desktop/color_calibration/charts/<serial_number>.tiff 에 저장후 돌리면 isp/ 하위에 json 이 생성됨
(e.g. ~/Desktop/color_calibration/charts/23486816.tiff)

~/Surround360/cubox/color_calibration/charts/23486825.tiff

(See surround360_render/res/example_data/color_calibration_input, ~_output.png)

python scripts/color_calibrate_all.py --data_dir ~/Desktop/color_calibration --black_level_hole

Copy to ~/Desktop/render/config/isp/<serial_number>.json,
and then run run_all.py

```
// ~/work/Surround360/surround360_render/.vscode/launch.json
{
  "name": "color_calibrate_all.py",
  "program": "~/surround360_render/scripts/color_calibrate_all.py",
  "args": [
    // read tiff from data_dir/charts, make output, isp directory
    "-data_dir", "./Desktop/color_calibration",
    // "-data_dir", "/home/skc0833/work/Surround360/cubox/color_calibration"
  ],
}

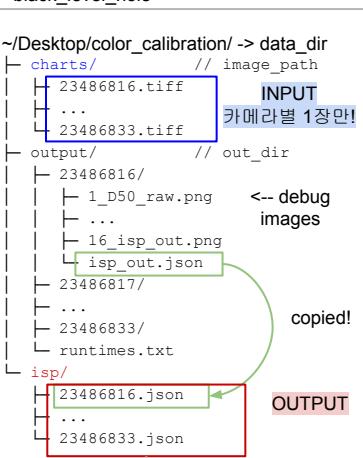
// Called bin/TestColorCalibration with
--image_path = ~/Desktop/color_calibration/charts/23486816.tiff
--isp_passthrough_path = /res/config/isp/passthrough.json
```

MacBeth ColorChecker



1_D50_raw.png

16_isp_out.png



Vignetting Calibration 을 위해
01 color calibrated json 파일들을
~/Desktop/vignetting_calibration/isp/
하위에 복사해줘야 함!

```
{
  "CamerasIsp" : {
    "serial" : 0,
    "name" : "PointGrey Grasshopper",
    "bitsPerPixel" : 16,
    "compandingLut" : [[0.0, 0.0, 0.0],
                        [0.6, 0.6, 0.0],
                        [0.7, 0.7, 0.0],
                        [1.0, 1.0, 0.0]],
    "blackLevel" : [1285.0, 1285.0, 1285.0],
    "vignetteRollOffH" : [[1.1, 1.1, 1.1],
                          [1.0, 1.0, 1.0],
                          [1.0, 1.0, 1.0],
                          [1.0, 1.0, 1.0],
                          [1.1, 1.1, 1.1]],
    "vignetteRollOffV" : [[1.1, 1.1, 1.1],
                          [1.0, 1.0, 1.0],
                          [1.0, 1.0, 1.0],
                          [1.0, 1.0, 1.0],
                          [1.1, 1.1, 1.1]],
    "whiteBalanceGain" : [1.1, 1.0, 1.65],
    "stuckPixelThreshold" : 5,
    "stuckPixelDarknessThreshold" : 0.11,
    "stuckPixelRadius" : 0,
    "denoise" : 0.6,
    "denoiseRadius" : 2,
    "ccm" : [[1.02169, -0.05711, 0.03543],
              [0.16789, 1.13419, -0.30208],
              [-0.15726, -0.07864, 1.2359]],
    "sharpening" : [0.5, 0.5, 0.5],
    "saturation" : 1.2,
    "contrast" : 1.0, // no contrast gain !!!
    "lowKeyBoost" : [-0.2, -0.2, -0.2],
    "highKeyBoost" : [0.2, 0.2, 0.2],
    "gamma" : [0.4545, 0.4545, 0.4545],
    "bayerPattern" : "GBRG"
  }
}
```

res\config\isp\cmosis_sunex.json
(Side camera)

```
{
  "CamerasIsp" : {
    "serial" : 0,
    "name" : "PointGrey Grasshopper",
    "bitsPerPixel" : 16,
    "compandingLut" : [[0.0, 0.0, 0.0],
                        [0.6, 0.6, 0.0],
                        [1.0, 1.0, 0.0]],
    "blackLevel" : [1542.0, 1542.0, 1542.0],
    "vignetteRollOffH" : [[1.3, 1.3, 1.3],
                          [1.1, 1.1, 1.1],
                          [1.0, 1.0, 1.0],
                          [1.1, 1.1, 1.1],
                          [1.3, 1.3, 1.3]],
    "vignetteRollOffV" : [[1.3, 1.3, 1.3],
                          [1.1, 1.1, 1.1],
                          [1.0, 1.0, 1.0],
                          [1.1, 1.1, 1.1],
                          [1.3, 1.3, 1.3]],
    "whiteBalanceGain" : [1.1, 1.0, 1.65],
    "stuckPixelThreshold" : 5,
    "stuckPixelDarknessThreshold" : 0.11,
    "stuckPixelRadius" : 0,
    "denoise" : 0.8,
    "denoiseRadius" : 4,
    "ccm" : [[1.02169, -0.05711, 0.03543],
              [0.16789, 1.13419, -0.30208],
              [-0.15726, -0.07864, 1.2359]],
    "sharpening" : [0.5, 0.5, 0.5],
    "saturation" : 1.2,
    "contrast" : 1.0,
    "lowKeyBoost" : [-0.2, -0.2, -0.2],
    "highKeyBoost" : [0.2, 0.2, 0.2],
    "gamma" : [0.4545, 0.4545, 0.4545],
    "bayerPattern" : "GBRG"
  }
}
```

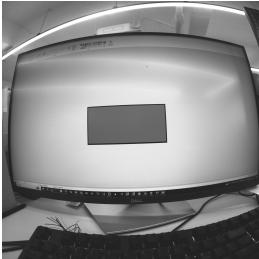
res\config\isp\cmosis_fujinon.json
(Top/Bottom camera)

Calibration

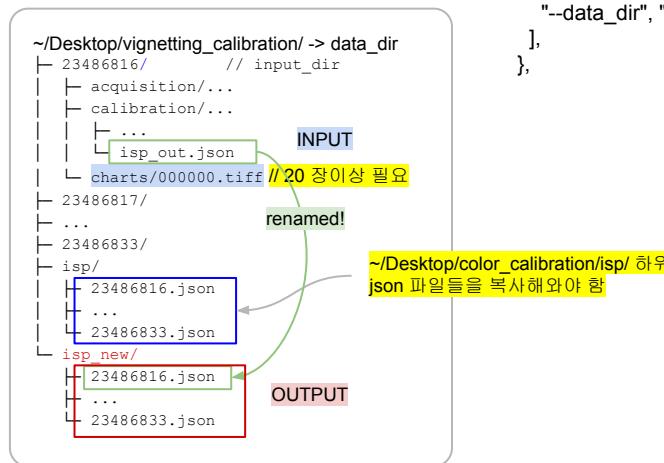
(2) Vignetting Calibration --> 카메라마다 20장 이상 촬영이 추천됨!!!
python scripts/vignetting_calibrate.py --data_dir ~/Desktop/vignetting_calibration \
--num_cams 17 --save_debug_images

```
{  
  "name": "vignetting_calibrate.py",  
  "program": "surround360_render/scripts/vignetting_calibrate.py",  
  "args": [  
    // data_dir should have serial_number folders!  
    // "--data_dir", "~/work/Surround360/cubox/vignetting_calibration",  
    "-data_dir", "~/Desktop/vignetting_calibration",  
    "--save_debug_images",  
  ],  
}  
  
// bin/TestVignettingDataAcquisition  
-> INPUT_DIR=~vignetting_calibration/23486816  
OUTPUT_DIR=~23486816/acquisition
```

```
// bin/TestVignettingCalibration  
-> OUTPUT_DIR=~23486816/calibration  
DATA_PATH=~23486816/acquisition/data.json
```



charts/000000.tiff



Copy to
~/Desktop/test/render/config/isp/23486816.json
and run run_all.py ???

```
// /home/cubox/cubox/Surround360/surround360_render/.vscode/launch.json  
{  
  "name": "[calibration] 1_vignetting_calibrate.py",  
  "type": "python", "request": "launch", "justMyCode": false,  
  "cwd": "/home/cubox/cubox/Surround360",  
  "program": "surround360_render/scripts/vignetting_calibrate.py",  
  "args": [  
    // INPUT:  
    // <data_dir>/23486816/charts/000000.tiff  
    // <data_dir>/isp/23486816.json(from color_calibrate_all.py)  
    // OUTPUT:  
    // <data_dir>/isp_new/23486816.json  
    // isp/<serial>.json files from ~/Surround360/cubox/color_calibration/isp ?  
    "-data_dir", "~/Desktop/vignetting_calibration",  
  ],  
},
```

Calibration

(3) Geometric Calibration

```
python scripts/geometric_calibration.py \
--data_dir ~/Desktop/geometric_calibration \
--colmap_dir /usr/local/bin \
--rig_json $PWD/res/config/camera_rig.json \
--output_json ~/Desktop/geometric_calibration/camera_rig.json \
--save_debug_images
copy!
```

* This generates a new JSON file, camera_rig.json, to be used when rendering by just copying it to the output directory, e.g. ~/Desktop/render/config/camera_rig.json. It also generates debug images under ~/Desktop/geometric_calibration showing the accuracy of the calibration process.

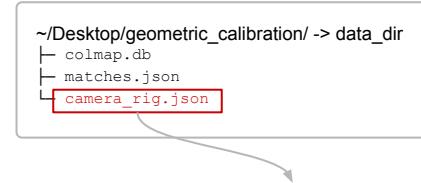
~/Desktop/geometric_calibration/ 하위에서 이미지 파일들을 찾아서 8 bit 이미지로 변환 & 저장
--> e.g. ~/Desktop/geometric_calibration/cam[0-16]/000000.png

```
COLMAP_DIR = "/usr/local/bin"
COLMAP_EXTRACT_TEMPLATE = {COLMAP_DIR}/feature_extractor
--General.image_path "{IMAGE_PATH}"          // 저장된 8 bit 이미지 파일들 디렉토리(data_dir)
--General.database_path "{COLMAP_DB_PATH}"      // ~/Desktop/geometric_calibration/colmap.db
```

```
COLMAP_MATCH_TEMPLATE = {COLMAP_DIR}/exhaustive_matcher
--General.database_path "{COLMAP_DB_PATH}"
```

```
{SURROUND360_RENDER_DIR}/bin/GeometricCalibration
--json "{RIG_JSON}"           // $PWD/res/config/camera_rig.json
--output_json "{OUTPUT_JSON}" // ~/Desktop/geometric_calibration/camera_rig.json
--matches "{MATCHES_JSON}"   // ~/Desktop/geometric_calibration/matches_json (from colmap.db)
--pass_count {PASS_COUNT}    // 10
--log_dir "{LOG_DIR}"        // ~/Desktop/geometric_calibration/logs
--logbuflevel -1
--stderthreshold 0
{FLAGS_EXTRA}                // --save_debug_images
-->
```

~/Desktop/geometric_calibration/debug 도 생성됨
refine(cameras, keypointMap, overlaps, pass, debugDir);
Camera::saveRig(FLAGS_output_json, cameras); **camera_rig.json** 를 업데이트!



Geometric Calibration 까진 마친 최종 camera_rig.json 를
~/Desktop/test/render/config/camera_rig.json 에 복사 후
run_all.py 를 실행!!!