Assignment 01

Implement Image Processing Pipeline with *rawpy*

Late submission policy:

For the whole semester, you totally have **5 exempt days** for late submission days. After expert days running out, you totally have **5 slip days**, 20% of the grades will deducted per day for the relative assignment.

Download Data:

data.rar

Part 0- Setting up your machine and environment

This assignment is required to use of python, define your python file name as **main.py**.

1. For this assignment, we require you to use the latest **conda** to manage your developing environment. It will help you to achieve a great experience for coding.

2. Install rawpy · PyPI and opencv-python · PyPI in conda.

```
pip install rawpy
pip install opencv-python
```

3. Implement the given blank functions using "rawpy". Read online document Params class rawpy 0.18.0a1 documentation

```
classrawpy.Params(self, demosaic_algorithm=None, half_size=False,
four_color_rgb=False, dcb_iterations=0, dcb_enhance=False,
fbdd_noise_reduction=FBDDNoiseReductionMode.Off, noise_thr=None,
median_filter_passes=0, use_camera_wb=False, use_auto_wb=False, user_wb=None,
output_color=ColorSpace.sRGB, output_bps=8, user_flip=None, user_black=None,
user_sat=None, no_auto_bright=False, auto_bright_thr=None, adjust_maximum_thr=0.75,
bright=1.0, highlight_mode=HighlightMode.Clip, exp_shift=None,
exp_preserve_highlights=0.0, no_auto_scale=False, gamma=None,
chromatic_aberration=None, bad_pixels_path=None)¶
```

Try your best to implement a better visual-looking result with a faster speed and less memory consumption.

Part 1 - Implement ISP using rawpy

Note: Here you do not need to fill the corresponding functions, just **comments** the functions and implement this steps using rawpy's functions.

BTW, pay attention to the **metadata**, it's just the example metadata and you need to read and replace the metadata using rawpy instead of directly using the given one.

Step 1. Bad Pixel Correction (10pts)

Implement it using **rawpy**, and save the corresponding result using **opencv** in .jpg format. Show and explain your result in your report.

Step 2.'Black Level Compensation' (10pts)

Implement it using **rawpy**, and save the corresponding result using **opencv** in .jpg format. Show and explain your result in your report.

- **user_black** (*int*) – custom black level

Example: read RawPy

```
XX.postprocess(param=rawpy.Paramers) Disable all other unrelated parameters and **only enable** parameters this and previous steps. You can use the same way to implement the following steps.
```

Step 3.'lens shading correction (Skipped)

Step 4. Anti Aliasing Filter (Skipped)

Step 5. Auto White Balance and Gain Control (10pts)

Implement it using **rawpy**, and save the corresponding result using **opencv** in .jpg format. Show and explain your result in your report.

- **use_auto_wb** (*bool*) – whether to try automatically calculating the white balance

Step 6. Test the following functions (10pts)

- **no_auto_scale** (*bool*) – Whether to disable pixel value scaling
- **no_auto_bright** (*bool*) – whether to disable automatic increase of brightness

Implement it using **rawpy**, and save the corresponding result using **opencv** in .jpg format. Show and explain your result in your report. Test different algorithms and compare the results.

Step 7. Color Space Conversion (10pts)

For now, you have got the no-auto-scale and no_auto_bright **RGB** image.

Implement color space conversion using **opencv**, and save the corresponding result using **opencv** in .jpg format. Show and explain your result in your report.

```
RGB2YUV()
UV2RGB()
```

Step 8. Test Gamma Correction (10pts)

- **gamma** (*tuple*) – pair (power,slope), default is (2.222, 4.5) for rec. BT.709

Also do the gamma directly in RGB space using BT.709's standard. Compare the difference of your results and the gamma provided by rawpy.

Submission

To Grade submission for this homework. It would help if you tried committing and tagging your code as specified in the given code structure. We highly recommend that you finish the homework to prevent any issues when you start on the following projects for this class.

Your grades = correct code (70%) + good report (30%)

Code submission (60%)

The code is required with a simple run, and then the TAs can see the results of each steps (name you saved results as *partxx-stepxx.jpg*)

Report (40%)

For each function you implemented, show the results crops , describe what you have done, and explain your results.

Write your report using **Markdown** (like MarkText, a free markdown editor) and export it to PDF format. Markdown is easy to use, just learn to use.