HDR Image Processing using Robertson and Reinhard Tonemapping in OpenCV

Author: Gayathri C Nair
Date: 29/09/2024

1. Introduction

HDR (High Dynamic Range) imaging is a technique that allows for capturing a greater range of luminosity than traditional digital imaging techniques. The goal of this project is to implement HDR image processing using multiple exposure images and applying Robertson's method for merging these images into an HDR image.

Furthermore, Reinhard tonemapping is used to map the HDR image into a displayable format, allowing it to be viewed on standard monitors. The problem this project aims to solve is creating a seamless HDR image from a series of images captured at different exposures while preserving details in both dark and bright regions.

2. Code Explanation

The code follows a structured approach to load the input images, apply the Robertson merging algorithm, and then tone map the HDR result using Reinhard tonemapping. Below is a detailed explanation of each section of the code:

- **Loading Images:**

```
img_fn = ["img1.png", "img2.png", "img3.png"]

img_list = [cv.imread(fn) for fn in img_fn]
```

Here, we load three images using OpenCV's imread() function. If any image fails to load, the program exits, preventing further errors.

- **Exposure Times:**

```
exposure_times = np.array([8.0, 8.0, 15.0], dtype=np.float32)
```

We specify the exposure times of each image. These values represent the time (in seconds) that each image was exposed during capture.

- **HDR Merging using Robertson's Method:**

```
merge_robertson = cv.createMergeRobertson()

hdr_robertson = merge_robertson.process(img_list, times=exposure_times.copy())
```

This step merges the input images into a single HDR image using Robertson's method, which takes into account the exposure times of the images.

- **Tonemapping with Reinhard:**

```
tonemap_reinhard = cv.createTonemapReinhard(gamma=0.8, intensity=-1.0,
light_adapt=0.7, color_adapt=0.5)

ldr_reinhard = tonemap_reinhard.process(hdr_robertson.copy())
```

Reinhard tonemapping is applied to the HDR image to convert it into a low dynamic range (LDR) image that can be displayed on standard monitors. The parameters allow for control over the gamma, intensity, and color adaptation of the result.

3. Results

In this section, we will compare the input images with the final HDR image processed using Reinhard tonemapping. Below are some key observations:

- **Before and After:**
    - **Input images:** These images were taken at different exposure levels (short, medium, long exposures).
    - **HDR Image:** The final HDR image effectively combines details from all input images, showing good details in both the dark and bright regions.


HYATT

- **Analysis of Parameters:** The Reinhard tonemapping was chosen for its ability to produce natural-looking results. The gamma value of 0.8 helps maintain contrast, while the color adaptation value of 0.5 ensures that colors are balanced and not oversaturated.

4. Testing

To test this code, use a set of images captured at different exposures. Follow the steps below:

1. Place your exposure images in the same folder as the script and name them as img1.png, img2.png, and img3.png.
2. Install the required libraries by running:

pip install opencv-python numpy

3. Run the script:

This will produce an output image (ldr_robertson_reinhard.jpg), which is the tone-mapped HDR image. Verify the results by comparing the output image to the input images.

Expected output: An HDR image that shows enhanced details in dark and bright areas.

5. Instructions (README)

Dependencies:

- OpenCV (4.0+)

- NumPy

Steps to run:

1. Install the required dependencies.
2. Place your images (img1.png, img2.png, img3.png) in the same folder as the script.
3. Modify the exposure times in the script if needed.
4. Run the script to generate the tone-mapped HDR image.

6. Conclusion

This project successfully demonstrates how to combine multiple exposure images into an HDR image using Robertson's merging method and apply Reinhard tonemapping to produce a visually appealing result. The final image preserves details across a wide range of lighting conditions, offering a good balance between dark and bright areas.

In the future, this approach could be enhanced by experimenting with other tonemapping techniques or adjusting the exposure times for better control over the final output.

7. References

- OpenCV Documentation: https://docs.opencv.org/
- Reinhard, Erik, et al. "Photographic tone reproduction for digital images." *ACM Transactions on Graphics* 21.3 (2002): 267-276.