ASSIGNMENT – 3: **HDR Image Processing Using Multiple Exposure Fusion**

**INTRODUCTION**

This report provides an overview of an HDR (High Dynamic Range) image processing implementation using the OpenCV library in Python. The project merges multiple exposure images into HDR images using three popular methods: **Debevec**, **Robertson**, and **Fusion Mertens**. Additionally, the tone-mapped outputs are displayed for visualization and saved as images.

**PURPOSE OF THE APPLICATION**

The purpose of this application is to create visually enhanced HDR images by merging multiple exposures of the same scene. It utilizes camera response function (CRF) estimation and exposure merging techniques to achieve realistic HDR outputs, ensuring better contrast and details in images with challenging lighting conditions. Furthermore, the Fusion Mertens method enables HDR-like results without requiring CRF estimation or exposure time inputs, providing a simpler and effective alternative.

**KEY FEATURES**

* **HDR Processing with OpenCV:**Efficiently merges exposure images using the Debevec, Robertson, and Fusion Mertens methods.
* **CRF Estimation:**Estimation of camera response functions using calibration techniques**.**
* **Fusion Mertens Method:**Produces HDR-like results by blending images without relying on CRF or exposure times.
* **Tone Mapping:**Converts HDR images into low dynamic range images (LDR) for visualization.
* **Visualization Tools:**Plots CRF curves and displays processed images with Matplotlib.

**LIBRARIES USED**

The application uses several libraries to implement its functionalities efficiently:

* **OpenCV**  
  Core library for HDR merging, CRF estimation, and image processing.
* **NumPy**  
  Handles array operations and exposure time storage.
* **Matplotlib**  
  Visualizes CRF curves and displays results.

**WORKFLOW**

**Image Loading**

* Loads multiple images captured with varying exposure times into a list.
* Ensures all images are correctly loaded; otherwise, exits the program with an error.

**Exposure Time Specification**

* Defines the exposure times corresponding to each input image in seconds.
* Normalizes these times for accurate HDR computation.

**CRF Estimation (Debevec Method)**

* Uses the createCalibrateDebevec method to estimate the camera's response curve.
* Processes the input images and exposure times to create a CRF that maps pixel values to radiance.

**CRF Estimation (Robertson Method)**

* Uses the createCalibrateRobertson method for camera response function estimation.
* Generates a CRF iteratively, minimizing discrepancies between observed and predicted radiance values.

**HDR Image Merging (Debevec Method)**

* Applies the createMergeDebevec method to combine the input images and their CRF.
* Produces an HDR image that accurately represents the scene's dynamic range.

**HDR Image Merging (Robertson Method)**

* Uses the createMergeRobertson method with the estimated CRF for HDR merging.
* Accounts for input image alignment and compensates for any noise introduced during exposure variations.

**HDR Image Merging (Fusion Mertens Method)**

* Utilizes the createMergeMertens method to blend multiple exposures.
* Bypasses CRF estimation and exposure time requirements, directly creating HDR-like images.
* Focuses on enhancing details, preserving contrast, and avoiding artifacts introduced by CRF calibration.

**Tone Mapping (Debevec Result)**

* Applies tone mapping to the HDR image using the createTonemap function with a gamma correction factor (e.g., 2.2).
* Converts the HDR result into an 8-bit LDR image for display.

**Tone Mapping (Robertson Result)**

* Similar to the Debevec result, applies gamma-based tone mapping to produce an LDR image.
* Ensures consistent brightness and contrast for visualization.

**Tone Mapping (Fusion Mertens Result)**

* Directly outputs an 8-bit blended image, eliminating the need for additional tone mapping.
* Provides visually pleasing results with enhanced clarity and color vibrance.

**CRF Visualization**

* Plots the CRF curves for both Debevec and Robertson methods.
* Visualizes channel-wise mappings (Red, Green, Blue) or grayscale response based on the input data.

**Result Saving and Display**

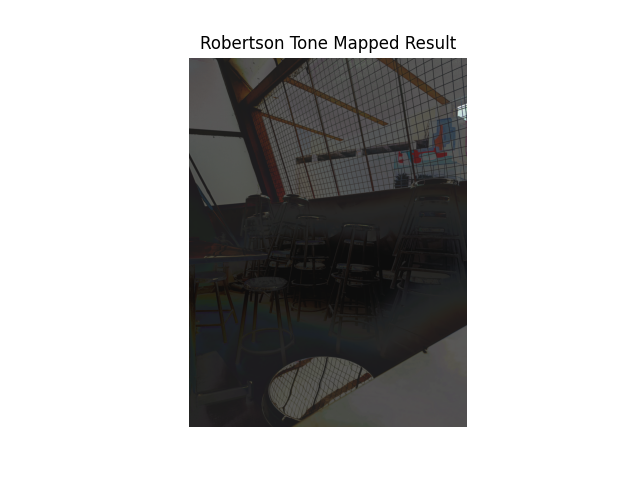
* Saves the tone-mapped images (ldr\_debevec.jpg and ldr\_robertson.jpg) in an 8-bit format for use.
* Displays the processed images using Matplotlib, showcasing the differences in tone mapping between methods.

**OBSERVATIONS**

* **Debevec (OUTPUT)**



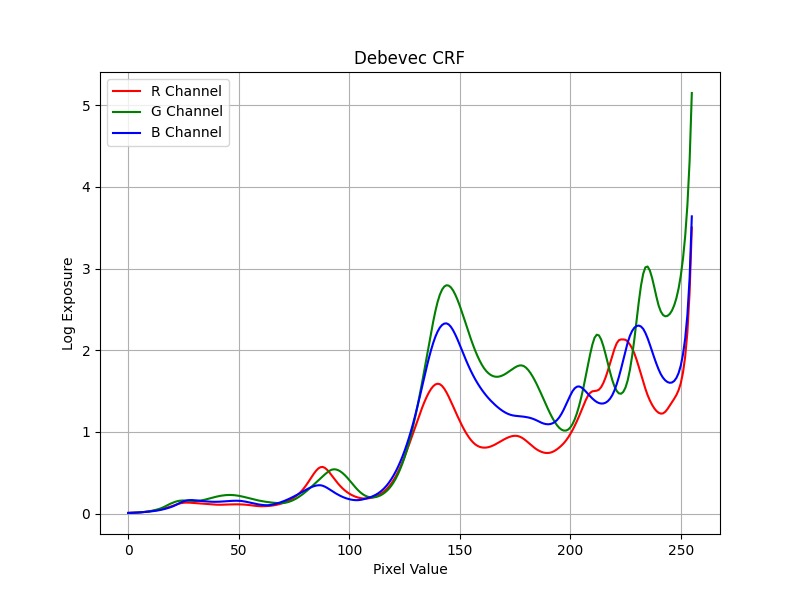
* **Robertson (OUTPUT)**



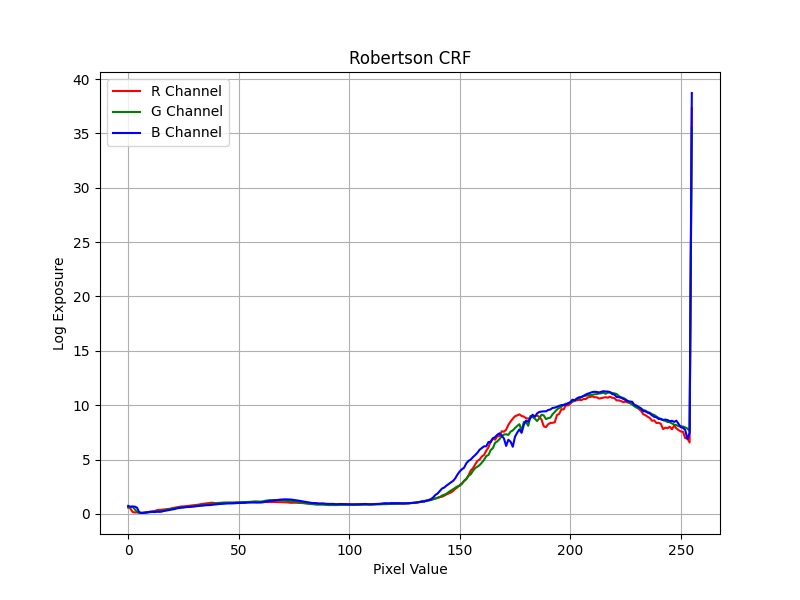
* **Fusion-Mertens (OUTPUT)**



* **Debevec-CRF**



* **Robertson-CRF**



**CONCLUSION**

The project demonstrates the effective use of OpenCV for HDR image processing, combining multiple exposure images using **Debevec**, **Robertson**, and **Fusion Mertens** methods. The Fusion Mertens technique is particularly beneficial for producing HDR-like results in a simpler and faster manner without CRF estimation. The inclusion of tone mapping and visualization tools ensures that the results are intuitive and visually appealing.