Image Reconstruction with Burst of Low-Light Photography

TONG YANG, WANZI ZHOU, CHEN QIAN
tongy@stanford.edu, wanziz@stanford.edu, cqian23@stanford.edu

Introduction

One of the most common problems that we encounter when we shoot photos with smartphones is that there is not enough light, leading to a lot of noise and low dynamic range in the image. Traditional solutions using analog /digital gain would either amplify the noise or motion blur.
Inspired by Google's paper published in 2016[1] which tries to solve the problem by implementing a newly proposed computational photography pipeline called HDR plus, we want to start from their work and develop a competitively efficient pipeline with good quality.

Problem Statement

We used the online HDR+ dataset where the input is a **bursts of underexposed raw images** of the same scene (much noise with each input but no detail is overexposed) and perform all steps of the pipeline on **Bayer raw patterns**.
Our goal is first try to reproduce the hdr+ pipeline and then by adjusting pipeline processes and parameters in multiple ways we derive something new, different and even better. Due to limited time, we spared the steps of aligning frames and started from frame merging.

References

[1] S. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy. Burst photography for high dynamic range and low-light imaging on mobile cam- eras. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia 2016)*, 2016.

Models and Algorithms

- Tile Split
  We split the Bayer raw input images into square tiles of width 16/32 with certain overlaps, and processed on the tiles of input images.
- Merge Frames
  Take frame 0 as reference. For each given reference tile, assemble a set of tiles across the burst, and compute their respective 2D DFTs as $T_z(w)$
  - ➢ Baseline average mering method
    We merge the frame by taking average for each frequency coefficients
    $$\tilde{T}_0(w) = \frac{1}{N}\sum_{z=0}^{N-1} T_z(w)$$
  - ➢ Frame merging with shrinkage operator
    Here we incorporates a filter that lets us control the contribution of different frames, which increases robustness and reduces misalignment problem
    $$\tilde{T}_0(w) = \frac{1}{N}\sum_{z=0}^{N-1} T_z(w) + A_z(w)[T_0(w) - T_z(w)] \quad A_z(w) = \frac{|T_0(w) - T_z(w)|^2}{|T_0(w) - T_z(w)|^2 + c\sigma^2}$$
    With 10-bit raw input images, we merge to 12 bits to preserve the precision gained from merging.
- Chromatic Denoise
  To reduce red and green splotches in dark areas of low light images, we converted the RGB image after demosaicing to YCbCr and applied Bilateral filter to the chromatic channels.
- Tone Mapping
  To fit the image into luminance range of display, we implemented local tone mapping by applying a non-linear bilateral filter to the intensity while preserving the detail information. .

Results and Analysis

- Merging Method Comparison
  Below shows the result of merging frames of baseline merging method, we find significant improvement on diminishing the ghost effect with the second method.


- Image Reconstruction
  The input low light condition images have mean value **0.1436**, after the pipeline we are able to reconstruct image that shows clear details of both the dark and bright region, and the mean pixel values is **0.5080**.

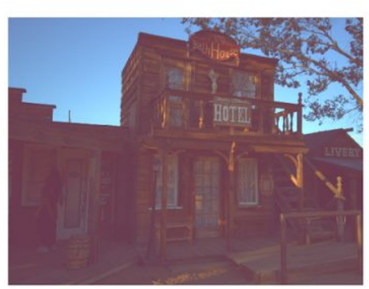
Conclusion & Future Work

We conclude that our pipeline is performing equally good as the hdr+ algorithm when not performing frame aligning. Using the merging algorithm with shrinkage factor effectively removes the ghost blur due to misalignment. The details are well recovered in hdr scenes (women in lower left corner). The tonemapped color is natural and well saturated (blue sky, brown house, green leaves).
Future work: We did not perform frame aligning, thus the edges are not sharp enough (eg. edges of the house). There are some artifacts in the blue sky (some parts are more green than blue), we think this needs further parameter adjustment in demosaic and denoise, and probably include steps such as hue specific color adjustment. We should also try to apply our algorithm in actual smart phones so that its computational efficiency can be better measured.

Pipeline and Result

Tile Split ⇨ Merge Tiles ⇨ Stack Tiles ⇨ Black-Level subtraction ⇨ White Balancing ⇨ Demosaic ⇨ Chromatic Denoise ⇨ Tone Mapping ⇨ Dehazing ⇨ Global Tone Adjustment ⇨ Sharpening


(Scaled by 2)


(Scaled by 4)