Low-Light Light Field (LF) Restoration

Snehal Singh Tomar
Aqil K H
Subhankar Chakraborty

Problem Statement

- A LF camera offers unique advantages such as post-capture refocusing & aperture control, but low-light conditions severely limit these capabilities
- Restoring LFs captured in low-light is not possible with single-frame low-light enhancement techniques designed for smartphones and DSLR cameras
- The goal of this project is thus to design a new deep-learning framework to restore low-light LFs

Proposed Approach

- The project shall be executed in the following stages:
    - Decoding raw LFs captured using lenslet based plenoptic cameras
    - Restoring decoded LFs using a deep neural network
    - Eliminating the decoding process altogether and training the neural network to restore raw LFs captured in low light

Understanding the Paper[1]

The Seven Broad steps required to be performed in order to decode raw LFs into their 4D representations as mentioned in [1] are:

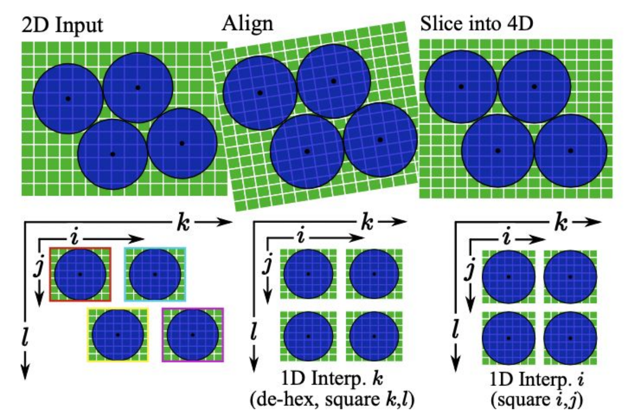
1. Demosaicing and Vignetting Correction
2. Aligning Sub-aperture image centres to integer locations on the sensor grid
3. Slicing each of the lenslet images separately. This slicing happens in the outer/spatial coordinates(x,y) and not in the inner/angular coordinates(u, v).
4. Converting Hexagonally sampled data to a rectilinear grid by interpolating along x.

[1] D. G. Dansereau, O. Pizarro, and S. B. Williams, “Decoding, calibration and rectification for lenselet-based plenoptic cameras,” in Computer Vision and Pattern Recognition (CVPR), 2013, pp. 1027–1034.

5. Correcting for rectangular pixels by interpolating along u.

6. Masking off pixels that lie outside the hexagonal lenslet image

7. Final conversion from (x,y,u,v) to (u,v,x,y) and interpreting the LF as an array of images captured from different perspectives which form on a specific sized sub-grid of the sensor plane.



Explanation of the steps mentioned above. Figure taken from [1].

[1] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in Computer Vision and Pattern Recognition (CVPR), 2013, pp. 1027–1034.

Progress Made

- The initial focus is on translating the relevant sections of MATLAB's light field toolbox to Python to seamlessly integrate it with a DL model
- Most of the relevant code is present in LFDecodeLensletImageSimple.m.
- An optimized version of the aforementioned code is present in LFDecodeLensletImageDirect.m
- Most of LFDecodeLensletImageSimple.m has been converted to Python except for a few functions which are MATLAB specific (*affine* and *imtransform*)
- The Python versions of these files can be found here

Bottlenecks Faced

- Initial difficulty in understanding the paper due to unfamiliarity with LFs
- Mapping the relevant sections of the LF toolbox to convert to Python
- Code conversion is time consuming
- Few MATLAB specific functions which have no direct substitute in Python
- Converting the future editions of the LF toolbox to Python

Plans for next two weeks

- Benchmark the Python code against the existing MATLAB implementation
- Look at future versions of the author’s implementation and try to include that in the Python code if it gives significant improvements
- Try a Python implementation of another open-source LF decoding algorithm[2]
- Train a neural network to perform restoration on RAW low-light fields decoded and saved as JPEG images
- Repeat the above step on the decoded LFs directly without saving them as JPEG or PNG images

[2] C. Hahne, “Plenopticam,” https://github.com/hahnec/plenopticam, 2020

Individual Contributions

- Snehal : Converting the “SliceXYImage” function in “LFDecodeLensletImageSimple.m”, the file containing the naive implementation of converting a 2D lenslet image to a 4D LF
- Subhankar : Converting the rest of the code in the file “LFDecodeLensletImageSimple.m”
- Aqil : Converting parts of “LFDecodeLensletImageDirect.m” different from “LFDecodeLensletImageSimple.m”

Questions?