CGRA352 Assignment 3

Light Field Imaging

20 marks

In this project you will work with light field data. The data is simulating a micro-lens array-based light field camera, with the micro-lenses focused on the main lens aperture. Your mission is to write functions that can take the raw sensor data from this camera and process it to create images focused at different distances and with cameras located in different positions than the actual camera.

Core (12 marks)

Light field data loading and processing

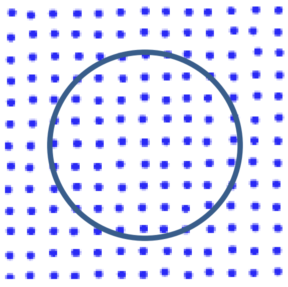
The light field data we have provided **"`LF.zip`"** is a 17x17 2D-array of images, simulating the processed light field data, where each image is captured from different perspective in the 2D UV-plane. The file name of the image is given in the following format:

`out_<row number>_<column number>_<v>_<u>.png`

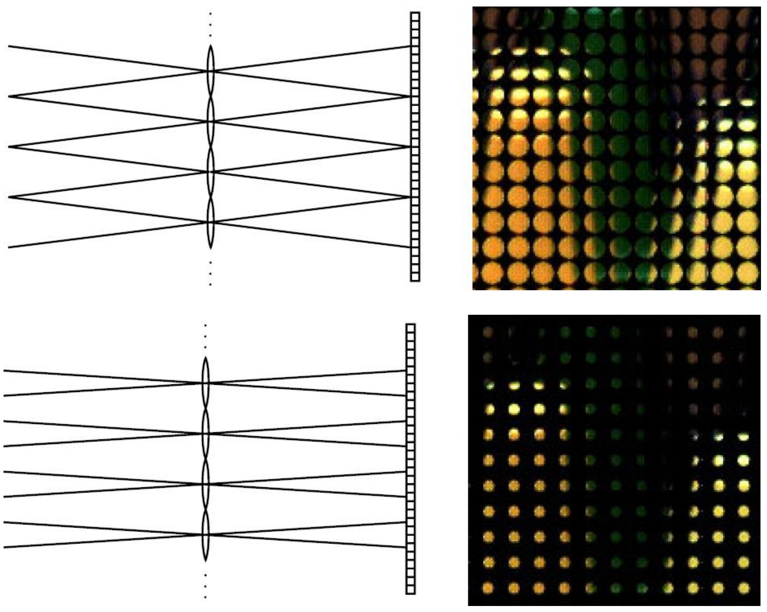
Where the **`row`** and **`column`** number are the position in the camera array, and the **`v`** and **`u`** are the positions on the UV-plane. There is some code provided that will help you load these light field images.

1. **(5 marks) Access the light field**
   Load all the images and write a function or equivalent that can access the value of a point in the 4D light field given the parameters `(row, col, t, s)`. Where `row` and `col` correspond to the array coordinates of the light field images while `t` and `s` are the row and column (respectively) of the pixel in the ST-image. There is no need to consider non-integer coordinates or interpolation for this assignment.
   Print out the results when accessing the 4d point (7, 10, 384, 768).
   The result should be [99, 135, 219]

2. **(7 marks) Generate ST-array**
   Load all the images and reconstruct the pixel values of *raw data* on the sensor of a sub-lens-based light field camera, with different main aperture sizes. Essentially, from your UV-array of ST-images, create an ST-array of UV-images, sampling the original light field using an aperture of a certain radius. The entire result will be too big to save or display so you will only have to do a small range of ST-coordinates.



Output two sensor images for the ST-array of UV-images for the rectangular region of s from 770 to 870 and t from 205 to 305. One with an aperture radius[*] of 75, and the other with a radius of 40, both centered at (-776.880371, 533.057190) on UV-plane.
*The radius means that, for each pixel in the UV-image we are constructing, we only record the value when the UV-coordinate (not the array position) is in the circular region of the aperture. Otherwise we leave it black.



(top) An aperture radius of 75, and (bottom) an aperture radius of 40.
Images are from another light field example.

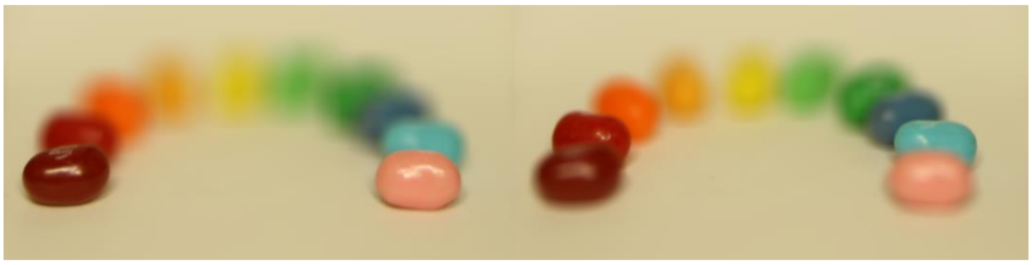
Hint: Your UV array is a 17x17 array of ST-images of size 512x1024. So the full ST-array of UV-images would be a 512x1024 array of 17x17 images. Because you're only doing a **small region of the ST-array**, it should come out to be a 100x100 array of 17x17 images, but displayed as one image (1700x1700) when stitched together.

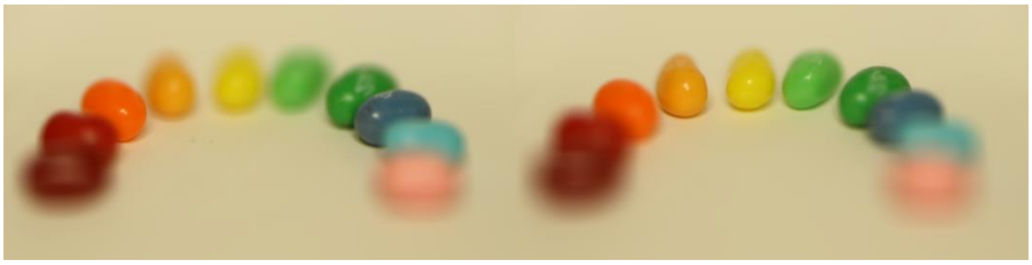
Completion/Challenge (6 marks)

Focal stack

A focal stack is a series of images that are generated by refocusing with a different virtual focal length using light field data. This is done by generating a virtual focal plane some distance away from the real focal plane and for each point, adding up all the light rays from the light-field that converge onto this point on the virtual focal plane. The focal point of the data is not known so you will have to experiment to get to best focal length for each image in the focal stack. **Generate 5 images using different focal lengths, where the focal plane is from far (the farthest object is in focus) to near (the closest object is in focus). The use of an aperture is optional.**

Images with different focal lengths in the focal stack should look similar to this.





The refocusing effects should look like the demo here: http://lightfield.stanford.edu/lfs.html Click "View light field online" for each dataset. Be aware that these are flash files and may not run in some browsers.

Report (2 marks)

You should submit a 1–2 page PDF document reflecting your experience. In this report, you should include:

- Brief introduction of your functions in your programs.
- How to run your program to perform the functions required by the assignment.
- The results of core, completion and challenge (depending on how much you have done).

Practical matters

You are required to submit both your program and any supplementary material including the report. You may complete the assignment in a single or multiple programs, as long as it meets the assignment specification. Please zip your program(s) source and other material into a single compressed file to upload. You are required to show your program can run on the ECS machine OR your own computer.

Helpful code

```cpp
// parse all images
std::cout << "Loading light field ..." << std::endl;
std::vector<cv::String> lf_imgs;
cv::glob(argv[1], lf_imgs);
for (cv::String cv_str : lf_imgs) {
	// get the filepath
	std::string filepath(cv_str);
	size_t pos = filepath.find_last_of("/\\");
	if (pos != std::string::npos) {
		// replace "_" with " "
		std::string filename = filepath.substr(pos + 1);
		pos = 0;
		while ((pos = filename.find("_", pos)) != std::string::npos) {
			filename.replace(pos, 1, " ");
			pos++;
		}
		// parse for values
		std::istringstream ss(filename);
		std::string name;
		int row, col;
		float v, u;
		ss >> name >> row >> col >> v >> u;
		if (ss.good()) {
			// TODO something with the image file "filepath"
			// TODO something with the coordinates: row, col, v, u
			continue;
		}
	}
	// throw error otherwise
	std::cerr << "Filepath error with : " << filepath << std::endl;
	std::cerr << "Expected in the form : [prefix]/[name]_[row]_[col]_[v]_[u][suffix]";
	abort();
}
std::cout << "Finished loading light field" << std::endl;
```