Documentation to COMP0028 (Course Work 2)

1 Basic section

In basic section, I followed the procedure from moodle and the program runs as follows:

- Load the image using the provided code.
- Compute the distance matrix.
- Generate the graph based on the distance matrix.
- Find the shortest path based on user's input.
- Render the path input a image sequence as a result.

1.1 Distance Matrix Computation

I referred to several different TAs during the lab section to find out how to compute the distance matrix correctly. I get two different approaches, one is to use SSD another one is to use L2. They both achieve the similar looks if I visualised them, to meet the coursework specification based on Lab 5, my program stick with the L2 norm approach which is based on the following equation:

$$m = \sqrt{\mathrm{diff}_R^2 + \mathrm{diff}_G^2 + \mathrm{diff}_B^2}$$

2 Advanced section

2.1 Better Path Computation

My approach to this section is to use a different way to compute the distance matrix, which includes the use of optical flow. Since the optical flow is a form of pixel displacement, so we can just do the addition when computing the distance matrix:

$$m = \sqrt{\mathrm{diff}_R^2 + \mathrm{diff}_G^2 + \mathrm{diff}_B^2 + vx^2 + vy^2}$$

Where vx,vy stands for the optical flow along horizontal direction and vertical direction.

Figure 1 and 2 visualised the differenc between the optical flow version and the normal version distance matrix. The computation of such matrix is named advanced1

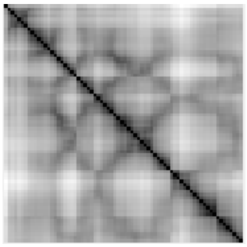

Figure 1: Distance Matrix without optical flow

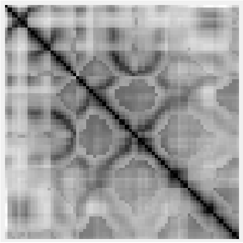

Figure 2: Distance Matrix with optical flow

2.2 Produce my own data

In this section, I use Canon 5D mark iii camera with EF 24-105mm f1/4L lense plus Speedlite EX600II-RT flashlight as my gear to caputer the movement of a lens cap with text 'Leica' inscribed.

The photos was taken on a pure dark area. The cap was placed on a suputer-low-reflection mat in order to maintain minimum background variation. I use bounce flahsh to create a soft light source.

All photos are taken on a tripod with the follwing specs: f/6.3, 1/40 shutter speed, ISO 100. Here is the one sample photo:

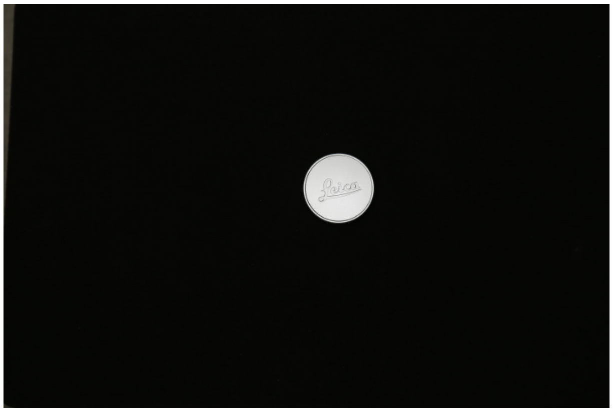

Figure 3: Data sample

And the full data set:

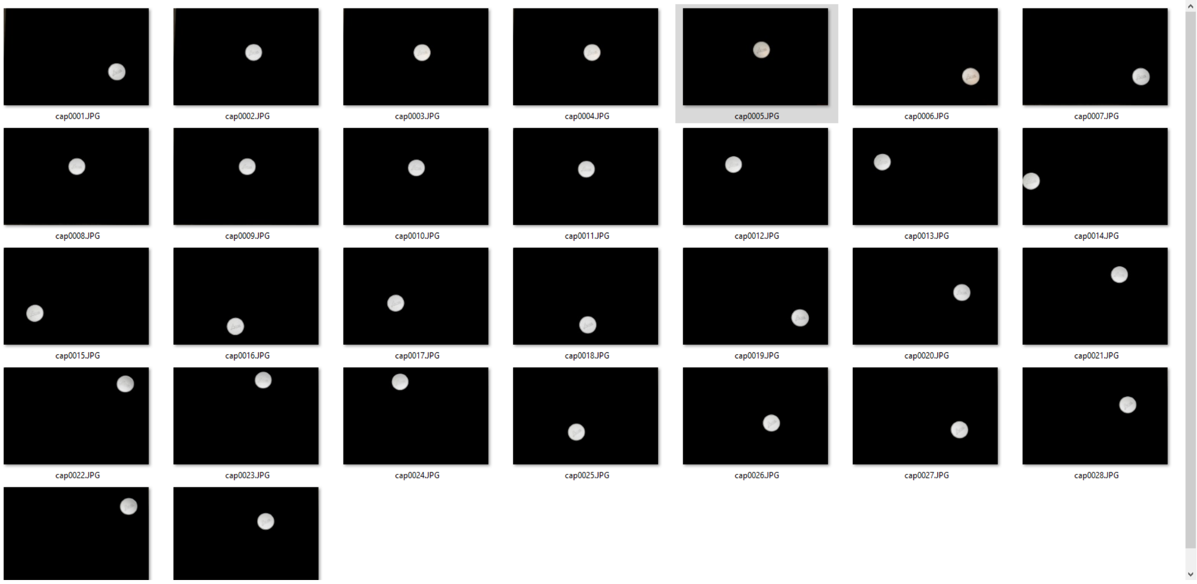

Figure 4: Whole data set

After loading the data, I use the provided function coarse2finetwoframes to get the optical flow. The implementatoin is in the computeOpticalFlow file.

Due to the upload limitaion, I have not upload the flow file, but the program will compute it once it finds out the file is missing.

All data are collected by myself JIALIN YU and I'm not sharing it to others.