Bogazici University, Electric Electronic Eng.
H. Işıl Bozma

Date: 8.05.2024
Due date: 17.05.2024 t:19:00

EE 576 - Project 5

The project aims to familiarize you with OpenCV image reading, showing and interaction.
Your code should be as **modular** as possible. This means that

> All the parameter settings and method-class definitions should be in a header file (*.h or *.hpp files)
>
> The code should not have any numbers directly - rather you should use parameter values as defined in the header file.
>
> The methods-functions need to be defined in a separate *.cpp file
>
> The main method should be minimal as possible
>
> The code should not contain any references to your local directory setup.

1. Consider the time-varying dataset and the corresponding mask images with mask indicating moving parts of the respective images. Make a directory named `Data` under your directory where your sources are. You are asked to write C/C++ code that will do the following:

1. Use code from previous projects as to open an empty image of size $(2N'_1 \times 2N'_2)$ on the screen where $N'_i \leq N_i$, $i = 1, 2$. Hence you are able to display images in a 2×2 grid arrangement. In case the image dimensions are large, pls resize images so that all the images in the grid are visible. Pls then consider the directory of the time-varying images. The top two cells should display any two consecutive images with indexes n and $n + 1$ as given by the user where n is a legitimate number wrt to the dataset;
2. Apply optical flow algorithm (using OpenCV API if you prefer) on these two images and then use the masks provided as to determine an average flow for each masked region of the image. Show the resulting average flow vector on each masked region in the third cell of the 2×2 grid arrangement.
3. Now consider masked regions one-by-one and apply a tracking algorithm (using OpenCV API if you prefer) to each. Show the resulting average flow on each masked regions.
4. Compute the norm of difference btw the two results and output this.
5. Repeat this as long as a legitimate index value is input. Otherwise, stop running the program. Compute average normed difference.

Write a very short report (1-2) in latex that discusses your algorithms for each part, sample results for each of the two parts as well as their comparative performance as measured by difference norm. Pls provide code and your report's tex, original figure files and pdf files. If you are using Overleaf, you may provide the link instead.