CMSC 733, Computer Processing of Pictorial Information
Project 4: Neuromorphic Vision!
Due on: 11:59:59PM on Tuesday, April 25 2017

Prof. Yiannis Aloimonos,
Nitin J. Sanket

April 13, 2017

In all the projects we have been dealing with images/frames which contain a lot of redundant information. Studies have shown that animals do not use input as frames but only percieve changes which makes more sense. A lot of smart people developed sensors based on the animal vision system and they are collectively called Neuromorphic Sensors. One particular sensor is called the Dynamic Vision Sensor or DVS made by iniLabs. DVS is a sensor in which each pixel (called event) transmits information in an asynchronous manner only when the intensity at that particular value has changed.

In this project, we will compute visual flow using events.

1 Data

The data file is provided as a mat file (`/Data/Data.mat`). The image shown in Fig. 1 is stuck on the wall and the DVS camera is moved such that the movement is only along the camera Z direction (as time increases the camera moves towards the image).

The data files contain the following:
`I` is the image of size 340×340 pixels. `x`, `y` are the location of the event at time `ts`, `pol` is the polarity of the event at (x, y, ts). The polarity can be ± 1.

2 Event-Based Visual Flow - 100Pts

Implement the algorithm given in Ref. [1] to compute the Visual Flow in the given data. Visualize the data as a 3D plot of (x, y, ts) and events of different polarity as different colors, visual flow using `quiver` and the fitted planes using `meshgrid`. Also talk about any interesting observations.

Figure 1: Image used in the dataset.

3 Starter Code

There is no starter code given for this project.

4 Submission Guidelines

There will be no Test Set for this project. Please make your report extremely detailed with as much visualization as possible. Describe all the steps (anything that is not obvious) and any other observations in your report. Your report should be in the IEEE double column format and should be typeset in LaTeXand **NOT** contain any code, feel free to include algorithms and math. Use the format provided in the `Draft` folder. Include outputs after each stage and talk about non-obvious obervations. You should also include a detailed `README` file explaining how to run your code.

Submit your codes (`.m` files) with the naming convention `YourDirectoryID_P4.zip` onto ELMS/Canvas (**Please compress it to .zip and no other format**). If your e-mail ID is `ABCD@terpmail.umd.edu` or `ABCD@umd.edu` your Directory ID will be `ABCD`.
To summarize, you need to submit these things and in the following strcture: A zip file with the name `YourDirectoryID_P4.zip` onto ELMS/Canvas. A main folder with the name `YourDirectoryID_P4` and the following sub-folders:

- `Code` with all your code. Please include instructions on how to run your code in the `README` file.
- `Report` with your report in pdf format (typeset in LaTeXdouble-column IEEE format given to you in `Draft` folder).

If your submission does not comply with the above guidelines, you'll be given **ZERO** credit.

Also, please mention about all the extra credit you have done in your report.

Also make a presentation (need not be slides, you can use your report or the whiteboard or just talk) if you want to present it in the class **Note: Every student should present AT LEAST once in the class (you can volunteer to present for more than one project) and can choose to do for any of the Project except the last one. Good presentations will receive upto 10 bonus points.**

5 Collaboration Policy

You are restricted to discuss the ideas with at most one more group. But the code you turn-in should be your own and if you **DO USE** (try not it and it is not permitted) other external codes/codes from other students - do cite them. For other honor code refer to the CMSC733 Fall 2016 website.

DON'T FORGET TO HAVE FUN AND PLAY AROUND WITH IMAGES!.

References

[1] Ryad Benosman, Charles Clercq, Xavier Lagorce, Sio-Hoi Ieng, and Chiara Bartolozzi. Event-based visual flow. *IEEE Transactions on Neural Networks and Learning Systems*, 25(2):407–417, 2014.