Documentation for Implementation of "On Event Based Optical Flow"

Veli Can Erdem

Introduction

This document contains information for the implementation of "On event-based optical flow detection" by Brosch et al. (2015). Link.

The project has 3 code files:

1. filters_demonstrations.ipynb
2. optical_flow.ipynb
3. util.py

filters_demonstrations.ipynb has code for generating filters and corresponding functions. It also has documentation on properties of filters and demonstrative outputs. It is used in optical_flow.ipynb with "%run" command.

optical_flow.ipynb uses filters over the image in order to produce filter outputs, which then is converted to quiver plots, the direction and quantity of output is visualized.

util.py has functions that are used for importing event data inputs.

Folders named slider_far, slider_close and basic_event are dataset of events. slider_far and slider_close are accessible in http://rpg.ifi.uzh.ch/davis_data.html.

Outputs in src folder contains outputs given by the code. Currently, in this folder, filters_4 and filters_32 contain the vectorized outputs of filters, using slider_far data, artificially slowed for maximal selective velocity.

Issues

Spatial Filters

- Spatial even filters and spatial odd filters are generated over a basis of 21x21 square. If the filters need to be extended to a bigger size, the ratio of the filters odd / even change.
- Currently no workaround for spatial even filters cancelling each other. May be solved by normalization.

Outputs

- When spatial filters are extended to a considerable size (i.e. 100x100) they cancel each other and make response at edges arbitrary, making artificial slowing necessary. (This can be solved by dilating temporal response)
- Only spatial odd filters' values are shown in outputs due to even cancelling itself.
- Normalization seems somewhat ambiguous in the paper, making a reimplementation of function necessary.