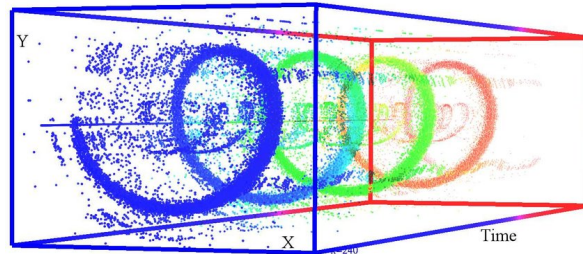


Stereo Vision based on Event Camera

Motivation



Event based cameras have been proved to have relative advantages over traditional cameras in the following aspects:

1. High temporal resolution
2. On-sensor pre-processing
3. High Dynamic Range: capable of handling extreme light or dark situations

The only shortcoming is the lack of supported algorithms, and the direct way to solve this shortcoming is to implement applications on this technology.

Stereo vision is often used to judge the distance between objects and the camera. And the advantages of our event-based cameras can well cope with various scene needs. Therefore, the transition from frame-based stereo vision algorithm to event-based stereo algorithm is our main task in this project.

Code Structure

This project is used to reproduce the results published in

[Kogler et al, Address-Event Based Stereo Vision with Bio-Inspired Silicon Retina Imagers, Advances in Theory and Applications of Stereo Vision \(2011\), pp. 165-188.](#)

We skipped the calibration process, since the images we used were already been calibrated.

The project implements the following functions:

- In the [CameraBuffer](#) class, the event procedures are being simulated.

This class includes two methods, one is to prepare data, it puts the collected data into memory.

The other is to find the corresponding event. Both of these two methods have implementations corresponding to two different data structures, one is ordinary Array, the other is a dictionary, also called the hashmap.

- In the [Util](#) class, loading method and matching costs functions are been implemented.

The loading method is mainly used to adapt to different formats of data.

Three common linear and nonlinear methods are provided in the cost calculation method.

- The [OutputController](#) class is the most important class in the entire project, which implements the very important WMI data structure in the event data processing

process.

In addition, some related methods such as refreshing WMI, applying filters, evaluating cost values, etc., are all implemented in this class.

- The data we used were provided by Professor Marianne Maertens and Professor Guillermo Gallego Bonet.

Please refer to this [link](#) for details and contact information.

Problems found during the implementation

- Due to the limited time, we only implemented the offline algorithm.
The efficiency of the algorithm for online calculations has yet to be considered.
The main reason for the inefficiency of the calculation in the offline version is that the event search interval is too large.
Since we have not limited the size of the cache, simulation During the process, all events are loaded and are taken into consideration.
This situation does not happen in practice because the cache size is limited.
- The choice of filter.
At the beginning of the algorithm, because the available data is very small, if the filter is applied at this time, it will cause wrong results.
The solution is that the filter should not be applied to the pixels where no event has occurred.
- The formulation of maximum disparity parameters. During the experiment, we found that setting the maximum disparity value incorrectly will lead to very wrong event matching results, because our algorithm only penalizes the time difference, but lacks the penalty for the relative distance of the event.
This problem has been improved in [subsequent papers](#).

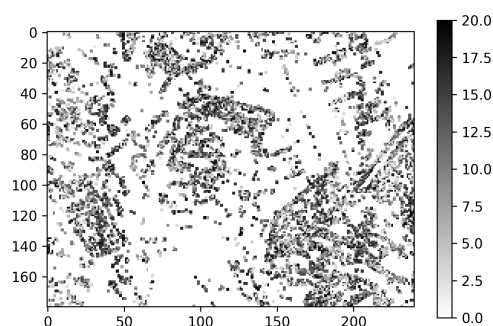
Usage

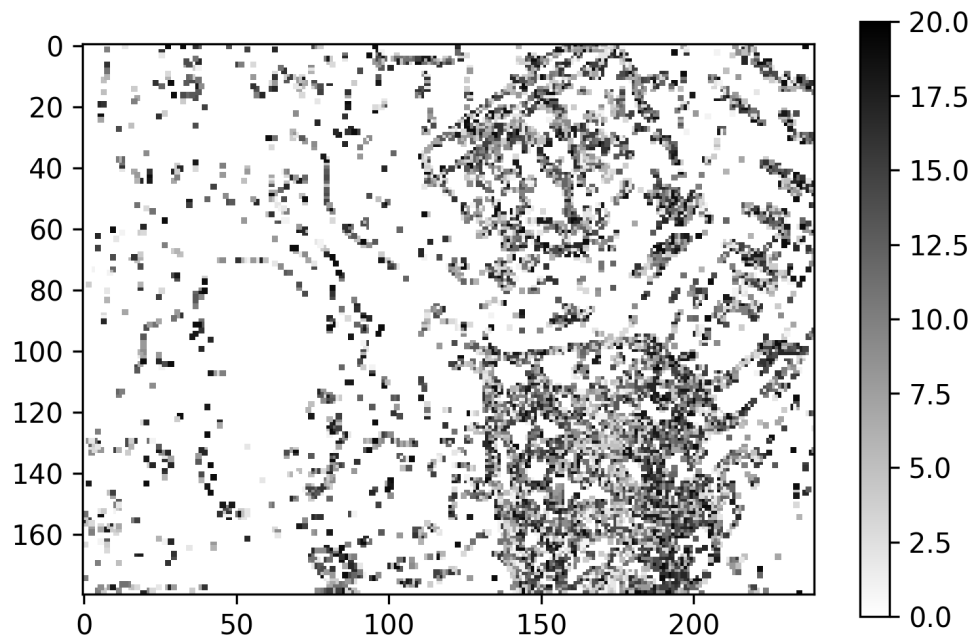
See `template.py`

Result

The result of our algorithm is a disparity map, which is shown below. The maximum disparity in the example is 20.

The detailed results are in the directory "result".





Reference

[Kogler et al., Address-Event Based Stereo Vision with Bio-Inspired Silicon Retina Imagers, Advances in Theory and Applications of Stereo Vision \(2011\), pp. 165-188.](#)

[Firouzi, M. & Conradt, J., Asynchronous Event-based Cooperative Stereo Matching Using Neuromorphic Silicon Retinas, Neural Processing Letters, 43\(2\):311-326, Apr. 2016.](#)