Frame to Event-Based (F2EB)

Description

The software presented in this repository is a tool written in c# language to convert Frame based images to neuromorphic vision sensor ones. Also the tool provides a mechanism to convert Videos to event based scenes. The output AEDAT file generated by the tool can be visualize in other application as JAER.

Table of contents

- Description
- Getting started
- Usage
- Contributing
- Credits
- License

Getting started

Prerequisites

F2EB requires Microsoft .NET Framework 4.6.1 or greater to be executed. The .NET Framework 4.6.1 and later versions are not supported on Windows XP, but on Windows Vista, Windows 7 and later versions of Windows.

In addition to Microsoft .NET Framework 4.6.1 or greater, the F2EB Visual Studio project requires Microsoft Visual Studio to compile the code (F2EB was programmed using Visual Studio Community 2017).

Installation

To use F2EB, first you need to download the latest release. This can be done by clicking on the "releases" button on the home page or just by cloning/downloading the repository.

In release page you will find different version, but we suggest to download the latest build.

When you download the folder, you have to decompress it and click F2EB.exe to run the application. Remember that if you dont install .NET Framework 4.6.1, the software wont work. If problem persists, please contact with me.

Usage

Double-click on the F2EB.exe file to run it.A form as next image shows will be open.

In this windows we can select an image to be converted clicking on **Browse Image**. Image will be loaded and displayed as next picture. Image is resized to 128x128 due, the idea is convert the image as a DVS sensor image.

When Image is loaded, you can apply different filter as Sobel,Laplace or Gaussian, you have to select in **Filter** tab which filter and then click on **apply**, next picture shown an example for all filters.

After apply filter(optional), you can select the **Conversion method** in conversion method tab and **convert** the image clicking convert. Currently three methods to convert frames are implemented:

- **Scan**: Produces a number of events in function of Grayscale value of each pixel. This methods analyze all pixels and check its greyscale, producing a maximum of 255 events per pixel due greyscale is coded using 8 bits
- **Random**: This method consists in a random number coded in 22 bits, where 14 most significants bits corresponds to the position of the pixel and the other 8 bits to the grayscale. If greyscale of the corresponding pixel is lower than the random greyscale generated, an event is produced. The method finished when the number of numbers generated is equal to the number of pixels in the image. The distribution of the events in this method is completely random.
- **Bitwise**: This method consists in a counter of 22 bits that is inverted in each iteration. As Random 14 bits are for position of the pixel and 8 to the greyscale.However distribution of events is uniform

When conversion finish a pop up window will be displayed.The Result Aedat can be displayed in JAER.Two example of Lena photo converted and displayed in JAER are shown in next picture.First image correspond to SCAN method and second one is Random method.

Apart of image conversion the tool is able to process a video and convert it using one method cited before. In **Video Capture time**, we specified time in microseconds that will be converted,then a conversion method must be selected and finally click on **Browse Video** to select the video to be converted.It will generate a pop up message when it finish, an AEDAT of the Video will be created.It is important to highlight that depends on the time specified it could take time until finish due the number of frames to convert.

Finally the last field called **Time between events** specified the time, in microseconds, since one event and the next that will be fired.

Contributing

New functionalities or improvements to the existing project are welcome. To contribute to this project please follow these guidelines:

1. Search previous suggestions before making a new one, as yours may be a duplicate.
2. Fork the project.
3. Create a branch.
4. Commit your changes to improve the project.
5. Push this branch to your GitHub project.
6. Open a Pull Request.
7. Discuss, and optionally continue committing.
8. Wait untill the project owner merges or closes the Pull Request.

If it is a new feature request (e.g., a new functionality), post an issue to discuss this new feature before you start coding. If the project owner approves it, assign the issue to yourself and then do the steps above.

Thank you for contributing in the F2EB project!

Credits

The authors of the F2EBI' original idea are: Ricardo Tapiador-Morales, Angel F. Jimenez-Fernandez, Alejandro Linares-Barranco and Gabriel Jimenez-Moreno.

The authors would like to thank and give credit to:

- Delbrück, T., "jAER Open Source Project," 2007. [Online]. Available: http://sourceforge.net/p/jaer/wiki/Home/.
- R. Berner et al., "A 5 Meps $100 USB2.0 Address-Event Monitor-Sequencer Interface," 2007 IEEE ISCAS, 2007.
- Linares-Barranco, A., Jimenez-Moreno, G., Linares-Barranco, B., & Civit-Balcells, A. (2006). On algorithmic rate-coded AER generation. IEEE Transactions on Neural Networks, 17(3), 771–788. https://doi.org/10.1109/TNN.2006.872253
- F. Gomez-Rodriguez, R. Paz, L. Miro, A. Linares-Barranco, G. Jimenez, A. Civit, 126 Two hardware implementations of the exhaustive synthetic aer generation method, in: 127 J. Cabestany, A. Prieto, F. Sandoval (Eds.), Computational Intelligence and Bioinspired 128 Systems, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005, pp. 534–540.
- J. P. Dominguez-Morales, et al., Navis: Neuromorphic auditory visualizer tool, Neuro115 computing 237 (2017) 418 – 422. doi:https://doi.org/10.1016/j.neucom.2016.12.046

License

This project is licensed under the GPL License - see the LICENSE.md file for details.

Copyright © 2018 Ricardo Tapiador-Morales
ricardo@atc.us.es

License GPL v3