**USAGE of PIX2NVS**

Features

PIX2NVS is a software that generates neuromorphic vision streams (also called address-based events) from conventional pixel-domain video frames encoded and wrapped in any standard format container (e.g. MP4. MKV, AVI, FLV, M4V).

Usage of executable PIX2NVS

Prerequisites

* Download the PIX2NVS software [here](https://github.com/PIX2NVS/PIX2NVS/tree/master/Executable%20Files), which is an executable file. There are two types for Windows and Linux and choose one suitable for your PC.
* PIX2NVS is developed based on FFMPEG library, so users should download the both ffmpeg and ffprobe bins from the website [here](https://ffmpeg.org/download.html) . Make sure you download the right bins for your computer since the bins for different operating systems (Windows, Unix) and architecture (64-bit or 32-bit) are different.

Procedures

1. Put the PIX2NVS, ffmpeg and ffprobe into a workspace folder.

Make sure that change the name of ffmpeg and ffprobe to ‘ffmpeg’ and ‘ffprobe’, respectively.

1. Create a folder named ‘input’ in this workspace and put the videos that you want to convert to NVS to the ‘input’ folder.
2. Open your computer terminal, change the shell’s working directory to your workspace folder. More details to see the command [cd](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/3/html/Step_by_Step_Guide/s1-navigating-cd.html) .
3. Run the executable PIX2NVS files by typing the code ‘PIX2NVS’ in the terminal, then the software will convert the videos to neuromorphic streams by using the default parameter. Default parameters are the much better parameters.

If you want to explore more by using different parameters, by attaching some arguments to the ‘PIX2NVS’ code. Arguments are list as following:

|  |  |  |
| --- | --- | --- |
| Parameter | Meaning | Vaules |
| R | Methods to update reference frame | 1,2,(0.9,1) |
| N | The number of events generated from one pixel | {1,2,3,4,5} |
| F | Values of threshold whether generates event | (0.2,0.9) |
| A | Changing coefficient of adapted threshold | (0,0.05) |
| C | Size of comparison blocks | {1 , 2 , 4} |

e.g. code ‘PIX2NVS R 1 N 2 F 0.5 C 1’ means using previous frame as reference, maximum number of events from one pixel is 2, the threshold is 0.5 and using pixel-to-pixel check, more details about the parameters to see the paper. Note that you can choose the number of arguments.

5. Generated events are stored in the txt files in the ‘Events’ folder. There is one event in each row, ordered as x, y, timestamp and polarity.

Usage of PIX2NVS code

Prerequisites

* Download the PIX2NVS code [here](https://github.com/PIX2NVS/PIX2NVS/tree/master/SourceCode).
* Download the both ffmpeg and ffprobe bins from the website [here](https://ffmpeg.org/download.html) . Make sure you download the right bins for your computer since the bins for different operating systems (Windows, Unix) and architecture (64-bit or 32-bit) are different.

Procedures

1. Put the PIX2NVS, ffmpeg and ffprobe into a workspace folder.

Make sure that change the name of ffmpeg and ffprobe to ‘ffmpeg’ and ‘ffprobe’, respectively.

1. In the “all\_defines.h” file, comment out the code ‘#define MSWIN’ at the beginning if you use this code on the Linux system.
2. Use Visual Studio, Eclipse to create a project and import these code files, then build and compile project. (For Linux user, you can directly use ‘gcc’ to compile these files in your terminal.)
3. Create a folder named ‘input’ and put the videos that you want to convert to NVS to the ‘input’ folder.
4. Run the code and the neuromorphic streams are stored in the txt files in the ‘Events’ folder.

Author & License

This project is licensed under the MIT License.

If you use our software, please cite our paper:

[1] Yin Bi, Yiannis Andreopoulos, ‘PIX2NVS: Parameterized Conversion of Pixel-domain Video Frames to Neuromorphic Vision Streams’ [C], IEEE International Conference on Image Processing (ICIP), Sept. 17-20, 2017, Beijing, China

Acknowledgement

This work is funded by the EPSRC project (EP/P02243X/1, IOSIRE Project).