

Freeture, a software to drive them all

Matthieu Garnung, Sebastien Celestin, Antoine Lair

LPC2E, University of Orleans, CNRS

European FRIPON meeting Paris Observatory

June 12, 2018

What is freeture ?

- Software for camera management and trigger algorithm
- It is written in C++ 11
- It contains 41 class
- The code is free (GPL) and open-source
- It is available on GitHub

The screenshot displays the GitHub interface for the repository `fripon/freeture`. At the top, there is a navigation bar with links for Features, Business, Explore, Marketplace, and Pricing. Below this, the repository name is shown along with statistics: 14 Watchers, 20 Stars, and 11 Forks. The main content area includes a 'Join GitHub today' banner and a list of commits. The commit list shows the following entries:

Commit	Description	Time
<code>omake</code>	Added parameter for regular capture prefix. Fixing bug from issue #12...	2 years ago
<code>data</code>	Restoring data directory	2 years ago
<code>doc</code>	Update repository.	3 years ago
<code>libs/vl</code>	Add x64 release lib vl	3 years ago
<code>share</code>	Added parameter for regular capture prefix. Fixing bug from issue #12...	2 years ago
<code>src</code>	White-space cleanup, gllignore update	2 years ago
<code>gllignore</code>	White-space cleanup, gllignore update	2 years ago
<code>.travis.yml</code>	Updating .travis.yml build instructions	2 years ago
<code>COPYING</code>	renaming FriponCapture to freeture	4 years ago

Figure: Freeture on GitHub.

How does freeture work ?

Freeture behavior

- 1 Start the main
- 2 Run several threads to manage acquisition, detection (meteors and sprites), and others
- 3 Waiting for a frame from the camera
- 4 Put image into a separate ring buffer for each detection thread

About the detection

Each detection thread manages the processing of frames

Meteor detection is the priority

All others processing threads must implement security in this regard

FRIPON's computer

- Intel NUC
- Intel Core i3-5010U (2c4t) @ 2.10 GHz 3 MB
- Intel HD Graphics 5500 with 23 "cores"
- DDR3 8 GB
- HDD 1 TB for storage
- SSD 128 GB for system



Figure: FRIPON station (FRCE01), Orleans.

Freeture in the future

Freeture v.3

To improve the science return of FRIPON, we need to Freeture

Full frame processing

- Currently, frames are downsized by a factor of 2 to reduce the cost of the image processing
- Hence, the pixel position is less accurate
- Upgrade and/or speed up algorithm for full frame processing

Local database on each station

- Replace the email based multidetection tool
- Need parsing, mailbox, and scripts for processing
- It is down if email server is down
- Using local database gives more flexibility against failures

Automatic masking

- Reduce the workload when adding a new camera to the network
- Allows for a larger detection region and adaption to the local surroundings

Conclusions

- Each camera is driven by a little computer
- On each computer, the software Freeture is set up
- The conception of the software allows the possibility of several detection item
- But upgrade of the software is necessary to improve the science return

Don't hesitate to join us on GitHub