aflak

Advanced Framework for Learning Astrophysical Knowledge

VISUAL PROGRAMMING ENVIRONMENT ESPECIALLY TUNED FOR MULTI-SPECTRAL ASTROPHYSICAL OBSERVATIONS

Malik Olivier Boussejra,¹ Shunya Takekawa,² Rikuo Uchiki,¹ Kazuya Matsubayashi,³ Yuriko Takeshima,⁴ Makoto Uemura,⁵ and Issei Fujishiro¹

¹ Keio University
² Nobeyama Radio Observatory ³ Kyoto University ⁴ Tokyo University of Technology ⁵ Hiroshima University

http://aflak.jp

malik@boussejra.com
Rust is a systems programming language that runs blazingly fast, prevents segfaults, and guarantees thread safety.

https://www.rust-lang.org/
A MODULAR STRUCTURE

Integrating together game technology for astronomy!

- cake: Computational task runner
- Node Editor
- Node primitives
- ImGui window
- Plotting
- ImGui windows
- pthread
- OpenGL

http://aflak.jp
- New nodes can be added from a list of nodes
- Nodes can be wired together
- When an output node is created, a corresponding output window opens
IMPORT / EXPORT OF NODE PROGRAM

- Export and import with *ad hoc* format

**Question**

- Any kind of VO standard for this kind of information?

http://aflak.jp
DOUBLE-FEEDBACK BETWEEN NODE EDITOR’S VARIABLES AND VISUALIZATION OUTPUT

Node A’s value is equal to the position of this vertical line
SAVE OUTPUT AS FITS FILE CONTAINING END-TO-END LINEAGE OF HOW THE DATA WAS CREATED

Question

Any kind of “standard” procedure for recording data provenance?
select ROI from pixel by pixel
CASE STUDY: EXTRACTION OF EQUIVALENT WIDTH
USE CASES

• Analysis that requires gradual and manual fiddling with many parameters

  aflak aims at putting the astronomer in the loop

• Denoising, preprocessing?
PRIMITIVES

- A set of algorithms applying transformations on datasets

- New nodes can be created
  - by combination of existing nodes (macro)
  - by direct implementation (currently Rust only, but C or Python solutions are explored)
FITS LIBRARY IN RUST: FITRS

https://github.com/malikolivier/fitrs
FUTURE WORK

• Batch processing
• Full macro support (Sit back! Writing the code as we speak, will be release very soon!)
• Full WCS support. Currently only partial and most probably buggy support is implemented.
• VO standards for communication with Aladin / Topcat
• Node primitives implementable in languages other than Rust (e.g. Python, C)
• Have more primitives included by default

http://aflak.jp
ONE (TWO) – COMMAND INSTALL!

$ curl https://sh.rustup.rs -sSf | sh
$ cargo install --git https://github.com/aflak-vis/aflak aflak

• Currently supported OSes:
  • Debian 9.X
  • Ubuntu 18.04
  • macOS (partial)
  • Crashes on Windows

• Run on normal laptop. RAM requirements depend on the open datasets. 4GB or more is advised.

• Bug report / Feature requests / Comment / Anything
  https://github.com/aflak-vis/aflak/issues/