Inside the FIM (Fbi IMproved) Scriptable Image Viewer

About a Small Command Language Powering an Image Viewer

Michele Martone

Declarative and Minimalistic Computing Devroom at FOSDEM
4 February 2023
I was a user of Gerd Hoffmann’s fbi (Linux framebuffer imageviewer)

I wanted VIM-styled arrow keys △ □ ◇ ▽
FIM: Fbi IMproved

Wolperdinger
Rainer Zenz’s edit on Albrecht Dürer’s Young Hare (from Wikipedia)

fim began as a fork of fbi, inspired by
- the VIM text editor
- the Mutt mail user agent
- shell languages
a Unix tool 🛠️

- one task: viewing images
- command line-oriented
- configuration files /etc/fimrc, $HOME/.fimrc
- regexes, stdin, stdout, scripting
- highly interoperable

there’s more than one way to do it (like in Perl and C++)

**Warning:** a viewer is *not* an editor

not here at least
respects slower hardware and unusual setups

- image caching (spare repeated load)
- background prefetch (anticipate load)
- minimal user interface
- all video modes are optional:
  - X11 (libsdl)
  - Linux Framebuffer
  - ASCII Art (libcaca/aalib)
- friendly to ssh and screen
an universal image viewer

▶ consistent look and feel
▶ several graphical modes
▶ `-o fb` / `-o sdl` vs `-o ca` vs `-o aa`

`fb` / `sdl` vs `ca` vs `aa`:
basic invocation

- $ fim file.jpg detect type by magic number
- $ fim dir/ consider graphics-like filenames
- $ fim -R dir/ recurse
- $ fim -B dir/ background load
keyboard-oriented interactive use

The scaling and "move around" keys you expect:
+,-,Left,n,p,r,[PageDown],...
bind keys with actions in the FIM language

- + magnify
- - reduce
- Left pan_left
- n next
- p prev
- r rotate90
- Del list 'pop'; reload
- PageDown if(_display_console==0){next;}else{scd;}
commands "live" in the : command line mode

▶ : activates command line mode
▶ Tab triggers autocompletion
▶ same syntax as configuration files and scripts
FIM's **language elements**

- **commands**
- **aliases** (customisable)
- **variables** (customisable)
- **if** and **while** blocks
- shortcut expressions and statements
how do I use fim

- daily
  - interactive usage
  - organize pictures collections
- occasionally
  - use special functionality
  - use internal command line
- rarely
  - change configuration file
  - create shell aliases/functions with it
this talk’s focus

FIM’s language

- intro-like
- quite exhaustive
- for interactive usage topics, see my other FOSDEM talk
most important commands

- help: the entire documentation (as in man fimrc)
- goto: navigate files list
- scale: change image scale
- pan: on-screen image alignment
- list: edit files list
- limit: temporarily shorten files lists
actions with **alias** and **bind**

- alias "pan_left" "pan 'left'"  *associate action to alias*
- alias "next" "goto '+1'"
- alias "next10" "i=0;while(i<10){i=i+1;next;display;sleep '1';}"  
  "goes forward 10 images"
- bind "N" "next10"  *associate keypress to alias*

*Easy things should be easy and hard things should be possible*  
(Larry Wall about Perl)
forgot what a **key** or an **alias** do?

- `help 'N'
- `help 'next10'`
special language shortcuts for \textbf{goto}

- \textsuperscript{1} activate interactive command line
- \textsuperscript{2} \texttt{goto 3}
- \textsuperscript{3} \texttt{goto '^	ext{}'} \textit{first image}
- \textsuperscript{4} \texttt{goto '$	ext{}$'} \textit{last image}
special language shortcuts for `scale`

- `-10%`
- `+10%`
- `*0.9`
- `*1.1`
repeat an action

- 2 magnify
- 360 { rotate 1; center; }
range syntax

- repeat action on files in position interval
- 2-4 echo '{}'

`remember find -exec echo {} ';'`
variables and expressions

- typing: dynamic, weak
- operators: arithmetic, comparison, string concatenation, ...
- examples
  - `a = 1+2` 3, integer
  - `a = '1+2'` '1+2', string
  - `a = '1'.2` 12
  - `a = '1'.2+3` 15
  - `stdout a` variable
  - `stdout 'a'` string constant
  - `stdout "a"` string constant
variables

- set by FIM:
  - `_cached_images` count of currently in memory
  - `_filelist_len` file list length
  - ...

- configured by user:
  - `_cache_control` cache behaviour
  - `_wantwm_caption_status` window manager caption format
  - ...

- special, e.g. `random` mutable (pseudo-random integer)
image variables

- i:filename: current filename
- i:swidth: scaled width
- i:width: original width
- i:EXIF_DateTime: from EXIF's DateTime tag
- ...

list current ones with [*]
autocommands

```vim
alias "gitinfo" "system 'git' 'log' '-1' i:_.filename";
autocmd "PreRedisplay" ".*" "gitinfo";
```

- perform action on matching
  - filename (.* means any)
  - event (PreRedisplay, PostRedisplay, ... see `man fimrc`)
- key internals rely on this
  - VIM-inspired

/cogs
key internals rely on this
VIM-inspired
range syntax + **system**

Warning: danger zone 🔄

```bash
[:2,4] system 'cp' '{}' './selection/' } as i:filename
```
scripts in/out

--write-scriptout  get script from interactive session, with timing

--execute-script  replay a script

if(i:width==0) quit 1; else quit 0;  exit status control
pipes galore

$ fim --script-from-stdin < script.fim  read a script

$ fim -i < image.jpg  read one image file from stdin

$ fim * | fim - | fim - > selection.txt  
mark with "Enter" to refine list for printout
fim v.0.6 is around 42 KLoC C++
bison parser + flex lexer
grammar (syntax, operators, ...) in man fimrc
language wishlist

► clearer variable identifiers
► parse error reporting
► escaping for reusable output
► less quoting
► better autocompletion
► glue languages
the end

▶ on the web: http://www.nongnu.org/fbi-improved
▶ in the distros:
  ▶ $ sudo apt-get install fim  *Debian & family*
  ▶ check out your distro, be it NixOS, Gentoo, arch, ...
▶ see `man fim` for options and `man fimrc` for language
▶ enjoy fim! (feedback welcome!)

Reminder

For more of fim’s interactive usage, check out my other FOSDEM’23 talk.