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

Before FIM

Young Hare
Albrecht Dürer, 1502 (from Wikipedia)

- I was a user of Gerd Hoffmann's `fbi` (Linux framebuffer imageviewer)
- I wanted VIM-styled *arrow keys* `h` `j` `k` `l`

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

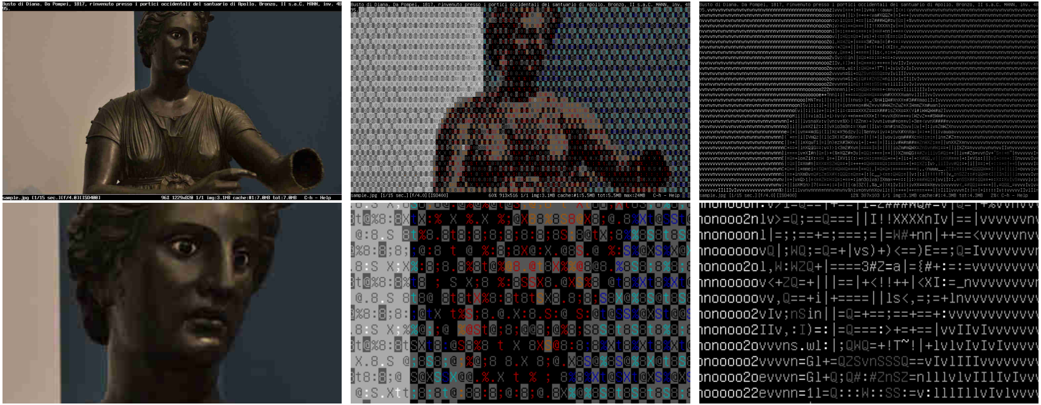
not here at least

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`:



```
Busto di Diana. Da Pompei, 1817, rinvenuto presso i portici occidentali del santuario di Apollo. Bronzo, II s.a.C. MANN, inv. 48
95.
```

```
sample.jpg [1/15 sec.][f/4.0][ISO400]          96% 1229x820 1/1 img:3.1MB cache:#1:7.0MB tot:7.0MB   C-h - Help
```

```
sample.jpg [1/15 sec.][f/4.0][ISO400]       60% 913x516 1/1 img:3.1MB cache:#1:5.5MB tot:5.5MB max:24MB   C-h - Help
```

```
sample.jpg [1/15 sec.][f/4.0][ISO400]        12% 307x103 1/1 img:3.1MB cache:#1:4.3MB tot:4.3MB   2%< C-h - Help
```

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*

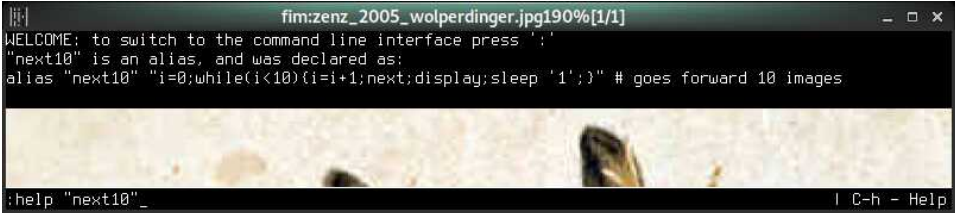
actions with `alias` and `bind`

- `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'



```
WELCOME: to switch to the command line interface press ':'
"next10" is an alias, and was declared as:
alias "next10" "i=0;while(i<10){i=i+1;next;display;sleep '1';}" # goes forward 10 images
```

```
:help "next10"_                                                                 | C-h - Help
```

special language shortcuts for `goto`

- `:` activate interactive command line
- `:` `3` *goto 3*
- `:` `^` `goto '^'` *first image*
- `:` `$` `goto '$'` *last image*

special language shortcuts for `scale`

- `:` `-10%` *scale '-10%'*
- `:` `+10%` …
- `:` `*0.9` …
- `:` `*1.1` …

repeat an action

- : `2 magnify`
- : `360 { rotate 1; center; }`

range syntax

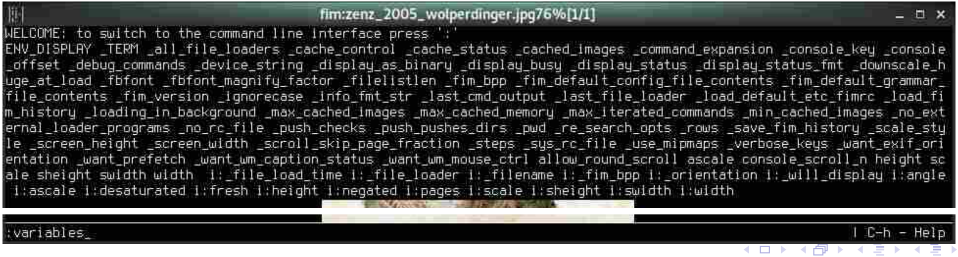
- repeat action on files in position interval
- : `2,4 echo ’{}’` *remember* `find -exec echo {} ’;’` ?

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*
  - `_filelistlen` *file list length*
  - ...
- configured by user:
  - `_cache_control` *cache behaviour*
  - `_want_wm_caption_status` *window manager caption format*
  - ...
- special, e.g. `random` *mutable (pseudo-random integer)*



```
WELCOME: to switch to the command line interface press ':'
ENV_DISPLAY _TERM _all_file_loaders _cache_control _cache_status _cached_images _command_expansion _console_key _console
_offset _debug_commands _device_string _display_as_binary _display_busy _display_status _display_status_fmt _downscale_h
uge_at_load _fbfont _fbfont_magnify_factor _filelistlen _fim_bpp _fim_default_config_file_contents _fim_default_grammar_
file_contents _fim_version _ignorecase _info_fmt_str _last_cmd_output _last_file_loader _load_default_etc_fimrc _load_fi
m_history _loading_in_background _max_cached_images _max_cached_memory _max_iterated_commands _min_cached_images _no_ext
ernal_loader_programs _no_rc_file _push_checks _push_pushes_dirs _pwd _re_search_opts _rows _save_fim_history _scale_sty
le _screen_height _screen_width _scroll_skip_page_fraction _steps _sys_rc_file _use_mipmaps _verbose_keys _want_exif_ori
entation _want_prefetch _want_wm_caption_status _want_wm_mouse_ctrl allow_round_scroll ascale console_scroll_n height sc
ale sheight swidth width  i:_file_load_time i:_file_loader i:_filename i:_fim_bpp i:_orientation i:_will_display i:angle
 i:ascale i:desaturated i:fresh i:height i:negated i:pages i:scale i:sheight i:swidth i:width

:variables_                                                                                              | C-h - Help
```

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

```
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

range syntax + `system`

Warning: danger zone ⚠

: `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

internals and parser

- `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.