Rating images painlessly with exiftool feat. ranger & sxiv

Firmin Martin

2020-12-03

Contents

1 Background **1**

2 Goals **1**

3 Exiftool **2**
3.1 Rating images with exiftool 2
3.2 File types supported by exiftool 2

4 ranger **2**

5 sxiv **5**

6 Bonus **5**
6.1 Rate or view images of specific rating 5
6.2 Migrate XMP rating into image metadata 6

1 Background

I was looking for a way to classifying images by rating them on the fly. My first attempt was using `darktable` as suggested in a thread. Indeed, the auto-advance rating mechanism was quite handy. But it is still too heavy for this sole purpose. In `darktable`, user have to import images before editing metadata. When tens of thousands images are involved, the process of importing images can be quite time-consuming[1] as it creates for each image an XMP file to store metadata. And this also applies to rating, even though I configure it to improve performance (without OpenCL) the latency is counted by seconds. Moreover, XMP are also created for symlink of image. This was not plausible in my use-case[2] as it enforces me to keep multiple metadata files for the same image.

Lots of critics, but clearly `darktable` was not the right tool. It suits better on raw photo post-production as intended. I will present in this post two solutions to remediate the issues aforementioned.

2 Goals

After experiencing `darktable`, I know better what I am seeking:

1. **Edit metadata in the image file itself**. This has two advantages:

[1] Dozen hours for 950.000 images.

[2] Statistical classification of images. For each class, it creates a directory in which each symbolic link is associated to the actual image.

   (a) Keep metadata even if the filename is changed.

   (b) Get rid of XMP files.

2. **Preview and select image without latency**. Namely, preview images and rate them on the fly.
3. **Metadata editing should follow symlink**. To centralize metadata in the same place.
4. **Batch rating**. Rating a whole directory or multiple selected images at once.

3 Exiftool

ExifTool is a free and open-source software program for reading, writing, and manipulating image, audio, video, and PDF metadata.

3.1 Rating images with exiftool

Rating image with exiftool is very simple.

```
exiftool -rating=5 -overwrite_original_in_place <files>
```

The option `-overwrite_original_in_place` overwrite directly the file(s) instead of moving the original one to `filename.ext_original`. Use it wisely at your own risk.

To read back the rating:

```
exiftool -rating <files>
```

... or format yourself the output:

```
exiftool -p '$Rating $Filepath' -f <files>
```

And, of course the symbolic links are followed[3]!

But with exiftool alone, one cannot watch and rate image at the same time. This can be done by combine up exiftool with a file manager having preview ability or an image viewer. Next, I will show how to integrate exiftool capability in the file manager `ranger` and the image viewer `sxiv`.

3.2 File types supported by exiftool

In fact, the version `11.88` of exiftool already supports a large set of file types. Thus, what has been and will be said is not limited to images and rating.

4 ranger

ranger is a free and open-source CLI files manager I'm using for years. It is very handy to select images and preview them[5].

Append the following snippet in `~/.config/ranger/commands.py`. It will add the custom command `rate_image <0-5> <files>`.

[3] Beware, without `-overwrite_original_in_place`, symlink will be removed!

[4] You may notice that you can't write MP3 metadata. `ffmpeg` should be used instead.

[5] As long as you use the right terminal emulator, *e.g.* iTerm, kitty, urxvt, xterm *etc.* From my past experience, Elementary OS's pantheon-terminal doesn't work.

3FR (r)	DR4 (r/w/c)	ITC (r)	ODP (r)	RIFF (r)
3G2 (r/w)	DSS (r)	J2C (r)	ODS (r)	RSRC (r)
3GP (r/w)	DV (r)	JNG (r/w)	ODT (r)	RTF (r)
A (r)	DVB (r/w)	JP2 (r/w)	OFR (r)	RW2 (r/w)
AA (r)	DVR-MS (r)	JPEG (r/w)	OGG (r)	RWL (r/w)
AAE (r)	DYLIB (r)	JSON (r)	OGV (r)	RWZ (r)
AAX (r/w)	EIP (r)	K25 (r)	OPUS (r)	RM (r)
ACR (r)	EPS (r/w)	KDC (r)	ORF (r/w)	SEQ (r)
AFM (r)	EPUB (r)	KEY (r)	OTF (r)	SKETCH (r)
AI (r/w)	ERF (r/w)	LA (r)	PAC (r)	SO (r)
AIFF (r)	EXE (r)	LFP (r)	PAGES (r)	SR2 (r/w)
APE (r)	EXIF (r/w/c)	LNK (r)	PBM (r/w)	SRF (r)
ARQ (r/w)	EXR (r)	LRV (r/w)	PCD (r)	SRW (r/w)
ARW (r/w)	EXV (r/w/c)	M2TS (r)	PCX (r)	SVG (r)
ASF (r)	F4A, F4V (r/w)	M4A, M4V (r/w)	PDB (r)	SWF (r)
AVI (r)	FFF (r/w)	MAX (r)	PDF (r/w)	THM (r/w)
AVIF (r/w)	FITS (r)	MEF (r/w)	PEF (r/w)	TIFF (r/w)
AZW (r)	FLA (r)	MIE (r/w/c)	PFA (r)	TORRENT (r)
BMP (r)	FLAC (r)	MIFF (r)	PFB (r)	TTC (r)
BPG (r)	FLIF (r/w)	MKA (r)	PFM (r)	TTF (r)
BTF (r)	FLV (r)	MKS (r)	PGF (r)	TXT (r)
CHM (r)	FPF (r)	MKV (r)	PGM (r/w)	VCF (r)
COS (r)	FPX (r)	MNG (r/w)	PLIST (r)	VRD (r/w/c)
CR2 (r/w)	GIF (r/w)	MOBI (r)	PICT (r)	VSD (r)
CR3 (r/w)	GPR (r/w)	MODD (r)	PMP (r)	WAV (r)
CRM (r/w)	GZ (r)	MOI (r)	PNG (r/w)	WDP (r/w)
CRW (r/w)	HDP (r/w)	MOS (r/w)	PPM (r/w)	WEBP (r)
CS1 (r/w)	HDR (r)	MOV (r/w)	PPT (r)	WEBM (r)
CSV (r)	HEIC (r/w)	MP3 (r)[4]	PPTX (r)	WMA (r)
DCM (r)	HEIF (r/w)	MP4 (r/w)	PS (r/w)	WMV (r)
DCP (r/w)	HTML (r)	MPC (r)	PSB (r/w)	WTV (r)
DCR (r)	ICC (r/w/c)	MPG (r)	PSD (r/w)	WV (r)
DFONT (r)	ICS (r)	MPO (r/w)	PSP (r)	X3F (r/w)
DIVX (r)	IDML (r)	MQV (r/w)	QTIF (r/w)	XCF (r)
DJVU (r)	IIQ (r/w)	MRW (r/w)	R3D (r)	XLS (r)
DLL (r)	IND (r/w)	MXF (r)	RA (r)	XLSX (r)
DNG (r/w)	INSP (r/w)	NEF (r/w)	RAF (r/w)	XMP (r/w/c)
DOC (r)	INSV (r)	NRW (r/w)	RAM (r)	ZIP (r)
DOCX (r)	INX (r)	NUMBERS (r)	RAR (r)	
DPX (r)	ISO (r)	O (r)	RAW (r/w)	

Table 1: File types supported by exiftool (v11.88) (r = read, w = write, c = create).

```
# ~/.config/ranger/commands.py
class rate_image(Command):
    """:rate_image <0-5> <files>

    Command for rating image with exiftools.
    """

    def execute(self):
        import subprocess
        if self.arg(1):
            rating_score = self.arg(1)
        else:
            self.fm.notify("rate_img: a rating score is required!", bad=True)
            return
        if self.arg(2):
            files = self.arg(2)
        else:
            cwd = self.fm.thisdir
            cf = self.fm.thisfile
            if not cwd or not cf:
                self.fm.notify("Error: no file selected for deletion!", bad=True)
                return
            if len(cwd.marked_items) > 1:
                files = " ".join([f.shell_escaped_basename for f in cwd.marked_items])
                self.fm.mark_files(all=True, val=False)
            else:
                files = cf.shell_escaped_basename
        command = "exiftool -rating=" + rating_score + \
                  " -overwrite_original_in_place " + files
        self.fm.notify("Run command: " + command)
        result = self.fm.execute_command(command, stdout=subprocess.PIPE)
        stdout, stderr = result.communicate()
        if result.returncode == 0:
            # This is a generic function to print text in ranger.
            self.fm.notify("Succeed to rate image " + files + \
                           " with score " + rating_score + ".")
```

It remains to define some key bindings to be granted the full power of `ranger`. Append the following snippet to `~/.config/ranger/rc.conf`.

```
# ~/.config/ranger/rc.conf
map r1 rate_image 1
map r2 rate_image 2
map r3 rate_image 3
map r4 rate_image 4
map r5 rate_image 5
```

The resulting workflow is as follows:

- Select images with `SPC` (single selection) or `v` (reverse selection).
- Press `r` and the rating `1` to `5`.

ranger_rating.gif

5 sxiv

sxiv is a free, open-source, lightweight and scriptable image viewer. Add the following entries in ~/.config/sxiv/exec/

```
# ~/.config/sxiv/exec/key-handler
case "$1" in
# ...
"C-1")      tr '\n' '\0' | xargs -0 exiftool -rating=1 -overwrite_original_in_place "{}" ;;
"C-2")      tr '\n' '\0' | xargs -0 exiftool -rating=2 -overwrite_original_in_place "{}" ;;
"C-3")      tr '\n' '\0' | xargs -0 exiftool -rating=3 -overwrite_original_in_place "{}" ;;
"C-4")      tr '\n' '\0' | xargs -0 exiftool -rating=4 -overwrite_original_in_place "{}" ;;
"C-5")      tr '\n' '\0' | xargs -0 exiftool -rating=5 -overwrite_original_in_place "{}" ;;
esac
```

The resulting workflow is as follows:

- Open images with sxiv.
- Press C-x C-<1..5> to rate the current image.
- Or mark images with m[7], toggle thumbnails mode with RET and press C-x C-<1..5> to rate selected images.

6 Bonus

Some interesting tips are presented here. The main dependencies are zsh and GNU parallel, adapt it to fit your need.

6.1 Rate or view images of specific rating

The long command below search all JPG files (potentially symlink) of rating 3, 4 or 5 and view them with sxiv.

```
# 1. Find recursively all JPG under the current directory
# 2. Use GNU parallel, for each JPG
#    (a) noglob (zsh specific): disable zsh globing to prevent "No such file" error
#    (b) Build the string "$Rating $Filepath"
#    (c) Remove the last line of exiftool output which count the amounts of the files.
#    (d) Keep entry with $Rating in {3, 4, 5}. Remove the first space.
# 3. Use GNU parallel: view at most 250 found images with sxiv at a time.
find -L . -type f -regex ".*\.jpg" |
parallel -L 5000 \
  'noglob exiftool -p '\''$Rating $Filepath'\'' -f -q -q {} | \
   head --lines=-1 | \
   awk '\''$1 ~ /[345]/ {$1=""; print substr($0, 2)}'\'' ' |
parallel -L 250 -q sxiv "{}"
```

Some remarks:

[6] If this file doesn't exist, you can copy the sample one: mkdir -p ~/.config/sxiv/exec/ && cp /usr/share/doc/sxiv/examples/key-handler .

[7] See sxiv/config.def.h for more keybindings to mark images.

- To see unrated images, use - as rating, then you can rate them with `sxiv` as above.
- This one-shot command is fast enough for hundreds of images. Above this amount, you may want to take some time to dump the result in a file as follows.

```
find -L -type f -regex ".*\.jpg$" |
parallel -L 5000 \
 'noglob exiftool -p '\''$Rating $Filename'\'' -f -q -q {} | \
  head --lines=-1 | \
  awk '\''$1 ~ /[^0-]/ {print $0}'\'' ' >> ratingdb.txt
```

And view images with specific rating with this command:

```
awk '$1 ~ /345/ {$1=""; print substr($0, 2)}' ratingdb.txt |
parallel -L 250 -q sxiv "{}"
```

- The best would be writing a python script to maintain an SQLite database, and adapt the the script of ranger and sxiv above to update the database each time a file rating changed.
- `shuf` may be used after `awk` to shuffle images before viewing them.
- Actually, no one will type again and again those lengthy commands. I either use `C-R` in `zsh` with fzf for casual ones or add them as entries in pet, a manager of parametrizable snippet.

6.2 Migrate XMP rating into image metadata

If you migrate from `darktable` or have XMP files with rating, you can try the following commands.

```
zmodload zsh/mapfile
# 1. Find recursively all XMP under the current directory
# 2. Use GNU parallel, for each XMP
#   (a) Extract the rating with grep and awk; put it in $RATE.
#   (b) Append the image path to .xmp-$RATE-rating.db
find . -type f -regex ".*\.xmp" -print0 |
parallel -0  'TMP=$(grep -o "Rating=\"[012345]\"" {} | \
  awk '\''{print gensub(/Rating="([0-5])"/, "\1", "g", $1)}'\''); \
  echo {} >> xmp-$TMP-rating.db'
# 3. For each $RATE:
for i in {1..5}; do
    FNAME="xmp-$i-rating.db"
    # For each file in .xmp-$RATE-rating.db
    for f in "${(f)mapfile[$FNAME]}"; do
        # Use exiftool to rate it with the corresponding $RATE
        exiftool -rating=$i -overwrite_original_in_place "${f%.*}"
    done
done
```