SkippyMovie

(Hopefully Redundant Instructions…)

SkippyMovie is my attempt at producing a no frills, easy to use Time Lapse animation utility.

To do this I make calls to the free FFMPEG utility (https://www.ffmpeg.org/). FFMPEG is designed as a Command Line utility, and does not offer an interactive GUI front end, which is where SkippyMovie steps in to provide a GUI that will assist the user by providing selection of the desired files and construction of the necessary calls to FFMPEG to create the output movie file.

Note Bene:

SkippyMovie does not convert raw data files. It can read the EXIF data from raw files and create indexed copies of the raw files (see below), but it cannot digest raw files and return an image. For that you need either the native software that came with your camera, or third party programs that use DCRAW or LibRAW.

SkippyMovie will handle BMP, GIF, JPG, TIFF and FITS files. Convert your raw files into TIFF or FITS and SkippyMovie is your friend.

FILE SORTING and INDEXING

Creating a Time Lapse movie clearly requires that the frames be included in the correct chronological sequence. Determining the correct chronological sequence may not be straight forward! Some imaging software will incorporate a timestamp in the filename, others in EXIF metadata, and FITS has its own metadata header fields. However, filenames with an embedded timestamp may not be correctly sorted by your PC's Operating System, leading to out of sequence frames. Sorting by file creation or modification date may also give false results depending on how the files were copied from one device to another, from one drive to another. The original creation/modification date from your Observatory laptop may be lost along the way.

Adding to that issue is FFMPEG's one critical demand of input files – that their filenames include a unique, sorted numerical index starting at zero, for example :-

Milky_Way_20170327T213345_0000.tif

Milky_Way_20170327T213547_0001.tif

..

Milky_Way_20170328T050252_9999.tif

This isn't so unreasonable, as we should know the correct order of our frames – the problem is interrogating either the filename, the EXIF data or the FITS header to retrieve the timestamp, and to then sort the timestamps and append the ascending numerical index to the filename.

The choice of 4 digits in that index is mine, on the assumption that the capacity to handle 10,000 frames is more than enough for any single animation. I may stand corrected.

Renaming your frames to include that numerical index may be difficult for some users, depending on their level of computer literacy and what software they have on hand. SkippyMovie includes a File Sorter utility to make this task easy. The File Sorter also provides methods to ensure that the files are indexed in chronological order, which is a potentially tricky issue depending on how the files incorporate a time stamp.

Template Editing

Some imaging software incorporates the timestamp in the filename, and allow the user considerable flexibility in the format of the filename, allowing temperature and other variables to also appear according to some pre-defined template. How is a 3rd piece of software like SkippyMovie to know just which numbers in the filename pertain to the timestamp? Easy, by editing a template of a selected filename to indicate where the year, month, day, hour, minutes and seconds fields are.

A typical filename, in this case from Backyard EOS might look like :-

LIGHT_45s_1600ISO_f4_+24c_20170129-02h25m24s401ms.jpg

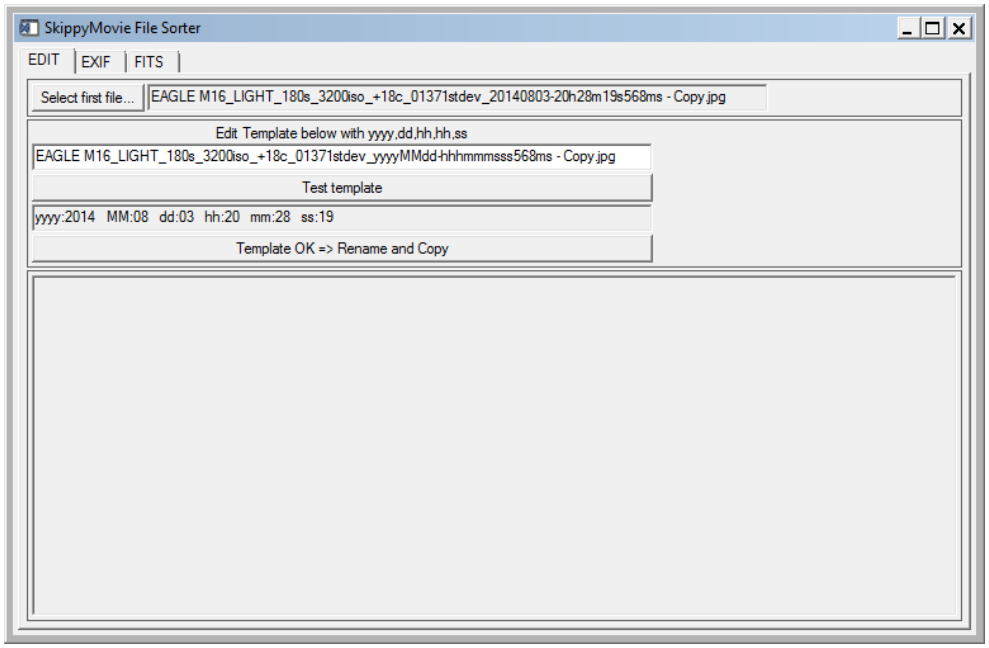
The edited template would look like:-

LIGHT_45s_1600ISO_f4_+24c_yyyyMMdd-hhhmmmsss401ms.jpg

Which the program then uses to extract the timestamp from all the frame filenames.

Note the “test template” button, which takes the user defined template to extract the matching characters from the filename and presents the results for inspection.

Happy with the results? Then press “Template OK => Rename and Copy” which will prompt for selection of the Input files, and for the output folder.



EXIF data

Some filenames might not contain a timestamp at all, but carry the timestamp info in EXIF data. This should be an easy, straightforward procedure to extract the timestamp from the EXIF data.

FITS Headers

FITS files have a range of standard and optional metadata fields, including "DATE-OBS" for the timestamp. One would hope that any FITS file contains the DATE-OBS field. If not, then all hope is not lost as the Template Editing mode is still be available to you.

In all three cases, the recovered timestamps are converted to Julian Day values, and then sorted to produce a master index of chronologically sorted files which is used to append the index number to renamed files in a nominated output folder.

If your files do not have embedded timestamps, or a timestamp in the EXIF or FITS metadata, then you are wholly reliant on the sort order of the Operating System. Good Luck…

Image Processing

Now that your files are correctly sorted, it is worth noting that SkippyMovie performs absolutely no quality assessment of your images. It's strictly a case of Garbage In, Garbage Out.

Deflicker Filter

On the assumption that deflickering the movie is a Good Thing, the use of FFMPEG's Deflicker Filter is hard-wired into SkippyMovie.

Video Codec

I have chosen to hardwire the video codec to H264, which is a very common codec, and offers good quality and compact file size. I can't imagine that any video player out there will have trouble playing a H264 file.

24bit input

You will need to pre-process your 16 bit images into useful, visible 8 bit RGB images ready for inclusion in a movie stream.

Audio Tracks

SkippyMovie can incorporate either mp3 or WAV audio files into the output movie. The audio track will be automatically truncated to the length of the output movie. Some WAV files produce a blank video – I do not know why. Mp3 files seem to function AOK.

Contact

The author, Andrew Cool, can be contacted at andrew@cool.id.au

SkippyMovie is written in Interactive Data Language (IDL), the language of professional astronomy.