You ‘Otta Be In Pictures!

Making a movie from data using Analysis   April 23, 2002   Dave McCrea
Last updated   January 9, 2009   Gilles Detillieux
Originating Document: T18_makemovie.wpd

Making a movie consists of:
1. In Linux, select a runfile and load into the analysis program.
2. In Linux, use the rawwfplt program to create the individual frames of the movie.
3. In Linux, generate an audio file from one of the WFs in the runfile.
4. Transfer the video and audio files to the Windows platform.
5. In Windows, examine the wave file (audio) using the Soundforge program.
6. In Windows, use the Platypus Animator program to create the movie.

Overview:
A movie can made that displays waveform and trace data in real time. In addition one of the wavefroms can be converted to an audio file and attached to the movie. In this way, audio is synchronized to the video display. A movie consists of a sequence of frames displayed at a particular interval and occupying a certain size of the display. All of these parameters can be manipulated and all must be specified to generate a movie. The rawwfplt program (script) discussed below is setup to make a movie using a default set of frame parameters that seem to work well for our data. Start with these.
We generate an AVI format movie from compressed bitmap frames (GIF). This movie can be played using any standard windows media player providing the CODEC has been installed in Windows. The standard Microsoft Video 1 codec seems to provide as good a compression as we can expect for this sort of data, so as long as the playback system has that codec, you may not need to install anything else. An update for this codec is available from Microsoft, which gives better results than the standard one when compression is cranked up a bit when making the AVI movie.
Recommendation: do NOT rely on an “improved” codec since it may not be installed on a foreign computer used to make a presentation.

1. In Linux, select a runfile and load into the analysis program.

- We recommend using the copyrun command to generate a new runfile from which the movie is created.
- Select the waveforms for display and their order (SWL), the run length (SRS, SRE), scale units and display (SDU, SDS) as appropriate for the movie. The goal idea is to setup the raw waveform display on the screen as you want it to appear in the movie.
- We recommend deleting sections of the runfile that will not be displayed. Set the analysis

range to the part that will be in the movie and trim (delete) the sections before and after using MT. By trimming a run and saving it as a new runfile, you could avoid using the copyrun command. The goal is to create a file in which the movie will span the entire run and allow synchronizing the audio to the video more easily.

- The colours of the waveforms can be changed (SDWP), a list of numbers (1-8) is entered that corresponds to the WF#s NOT the WF list order. If fewer pen numbers are entered, the list repeats (starts over) until a pen is assigned to each wavefrom.
- Pen numbers are: 1 black 2 blue 3 green 4 red
5 purple 6 gold 7 brown 8 gray
- If you want to display a trace in the movie above the waveforms (only one allowed), you must select the trace to be displayed (SLTN). By default in analysis, traces are displayed vertically when SDWM is used to display traces. To change to a horizontal trace display use (SDWL). The vertical dimension of the movie window to be used for trace display can be changed to increase or decrease the trace size (SDWH, in percent, e.g. 50).
- Exit analysis saving the parameter set. This parameter set will be used by the rawwfplt program to create the video frames.

2. In Linux, use the rawwfplt program to create the individual frames of the movie.
   - Options include the number of pixels per frames (eg. 520 x400), the number of frames per second in the movie (e.g. 10) and the length of the horizontal time scale (in seconds) displayed during the movie.
   - rawwfplt is actually a script file that runs two programs. First, using the parameter set you created it, generates a HPGL plot file from the run file for each video frame using the analysis program. Then it uses the hpgl2gif program to convert these HPGL files into compressed bitmap images in the GIF format.

   e.g.
   ```
   rawwfplt -g  -f 10 -l 3 myrun movie000
   ```
   This command generates a series of GIF images from myrun. There will be 10 frames per second and each frame will display a 3 second image of the waveforms. By default the entire run length will be used and the size will be 520x400 pixels. If the run length was 30s, 300 frames will be generated and named as movie000.gif, movie001.gif, movie002.gif ..... movie300.gif. Do NOT use any numbers in the output file name other than the starting frame number (movie000).

```
Usage:  rawwfplt [-g gifopts] [-s start] [-e end] [-f fps] [-l length] runfile [outname]
```

where: -g specifies that you want HPGL plotfiles converted to GIF this can be followed by options to hpgl2gif if desired, and if valid hpgl2gif options are given without -g, the -g is assumed

typical options are -w and -h to specify the width and height of the movie in pixels. Default is 520x400. The recommendation is to use a 1.33:1 ratio with each

dimension divisible by 8. (e.g. 320x240)
-s start specifies start time in run, in seconds (default is 0)
-e end specifies end time (start of last frame), in seconds (default is end of run)
-f fps specifies "frame" rate, in plot files per second of data (default is 10)
-l length specifies length of data per frame, in seconds (default is 1)
runfile specifies a run file name or analysis parameter file (no default, runfile must be specified)
outname   specifies first file name in plot file series (default is "raww0000.plt")
a maximum of 10000 plot files will be generated

Note: the start, end and length arguments to the -s, -e and -l options **must** be specified in seconds, and unit specifiers are not allowed after the numbers. This is unlike the analysis program and some other analysis scripts where time values are generally specified in ms with an optional unit specifier to override the default.

You can use the -help option to either rawwfplt or hpgl2gif to get a usage summary of either of these programs, explaining all available options. The rawwfplt program accepts any option that hpgl2gif accepts, as well as its own set of options. If you are insane, you can use the -p option to change the pen colours to something other than the default ones listed above. Otherwise use the defaults and change pen colours (e. g. colours of the waveforms) in the analysis program.

3. In Linux, generate an audio file from one of the WFs in the runfile.

- An ENG (or any other) WF can be used to generate a Windows standard wave(audio) file. This is accomplished either by a script (wf2wav) that runs the Linux sox program (with certain options set for convenience) or by directly using the sox program.
- the advantage of using the script wf2wav instead of the sox program is that you do not need to figure out the sample rate of the original WF in the runfile. The script extracts this information from the run header.
- The wave file will be the entire run length; this is a very good reason for the creation of a trimmed down copy of the original runfile mentioned above.
- sox has many options, including multiple channels, sound effects, file type conversion, etc. See sox(1) for details (type “man sox” at the command prompt to get detailed sox instructions)

```
sox -V  -c 1 -r 5000 -t .sw -x myrunfile.w03 -r 11025 -v 64 mynoise.wav
```

this runs the sox program to be verbose (-V); convert one channel (-c 1); with an input (data) sample rate of 5000Hz (-r 5000); convert the file from signed word format (-t .sw); using the data contained in the third waveform myrunfile.w03 (-x); converting it into a 11025 Khz wavefile (one of the windows standard rates; -r

11025); with a volume of 64 (range 1-128; -v 64); and create the wave file called mynoise.wav (you must supply the .wav extension)

- wf2wav - convert W.F. file from runfile waveform into RIFF WAVE (.wav) file

```
Usage:   wf2wav [-V] [-v volume] [-r rate] runname.wnn [sox-options] [outfile]
where:   -V sets verbose output for sox
-v volume  sets output gain factor for signal volume (default 128)
-r rate  sets output sample rate in Hertz (default 11025)
```

if no sox options or outfile is specified, the default is to output a .wav file on standard output. This is not pretty site and may screw up your display!

- If the volume is set too high in the conversion, clipping will occur. This sounds very nasty and must be avoided. The sox program with the -V option (upper case V) reports how many events clipped. Adjust the -v option (lower case v) to eliminate clipping.
- A rectified integrated ENG can be used for a wave file but it will sound very “bassy”. One solution is high pass to filter the waveform in analysis (MFH yes; MFC 50) creating a new waveform that is centered around 0 (no DC offset) and contains relatively more high frequency information (which sounds better). Use this WF for the audio file.
- See the Soundforge program discussion below.

4. Transfer the video and audio files to the Windows platform.

- In Linux, create a zip file of all the video frames (*.gif)

```
zip -m movie *.gif
```

The zip file is not necessary but it is faster to transfer the single file than the 300 or so individual frames.

- In Windows, transfer the video frames and the wave file to a directory using FTP and unzip them.

5. In Windows, examine the wave file (audio) using the Soundforge program.

- Soundforge can be used to amplify, filter, compress, normalize, etc. the wavefile. There are many reasons why it is better to manipulate the file in Soundforge than using sox under Linux.
- Note in particular if the wavefile contains a large DC offset. This can be removed with the Process - DC offset menu.
- Note if the very beginning and end of the file has a few large data points. This can result from a DC offset in the wave but must be removed separately (i.e. may not be removed by Process - DC offset). Highlight the offending regions and use the Process-Mute menu to remove the data points while maintaining the length of the waveform.
- A word of advice. You will make a movie and listen to it on your cheap desktop computer speakers using 2-5 watts of audio power. You may not notice anything untoward; clipping or DC offsets may make only small, almost imperceptible clicks and pops. However, when

the movie is played in a large theatre with a professional audio system (100's of watts of audio power), the same artefacts will leap out of the audio track and bite you in the ass!

- Soundforge can also be used to change the audio in an existing movie or to create a stereo audio track. You're on your own for these operations.

6. In Windows, use the Platypus Animator program to create the movie.

- Select the Microsoft Video 1 codec under Options/AVI settings; Compression Quality = 100 (to retain high quality images), and Temporal Quality Ratio between 0.75 and 1.00.
- Avoid the "Quick AVI create..." wizard. Instead, follow the sequence under the File menu to import frames, add audio and create the avi.
- Note: the first frame of the movie will contain only the axes and no data. Use Platypus to copy a suitable frame (e.g. the last frame) and insert it at the beginning. To keep the movie the correct length, delete the original first frame.

OTHER movie handling notes....

- To extract frames from an animated .GIF file, use Jasc Software - Animation Shop 3.
- To extract frames from an .AVI movie, use Platypus, File / Extract Images from AVI...
- To make an animated .GIF file on Linux, from individual .GIF frames, use the "gifsicle" utility (see "man gifsicle").