

AVED Processing on the Beowulffish Cluster

D. Oliver, D. Cline

Last updated: 9/6/2005

12/15/05

Overview

These instructions describe processing clips through the AVED software on the Beowulf cluster. This does not include instructions for running the AVED Classifier. In these instructions, in all cases where **username** in bold type is written, replace with your username. In cases where the font is *italic* – these are commands you need to enter in a Linux X-terminal window.

It is assumed the reader is familiar with basic Linux command-line commands like *ls*, *cd*, *mv*, *etc*. If not, ask someone for assistance. You don't need to be an expert Linux user to process on the Beowulf cluster, but you need to know the basics.

Supported file formats

AVED processing only supports two formats: AVI or Quicktime (MOV). In both formats, only 8-bit RGB raster formatted data is support in our current work flow. If your format is different than an AVI or MOV, be sure to put it into this format using a tool like Adobe Premiere, Quicktime Professional, or ImageMagick. Also, the file format needs to be in the format <filename>.mov or <filename>.avi and file names like <filename>.test.mov are not currently supported.

Placing file in scratch space

Processing video consumes a large amount of space so a temporary scratch space is setup on the Beowulf cluster for all AVED processing. Place your AVI's or Quicktime files in scratch for processing. If you do not have a personal scratch directory contact Danelle Cline at dcline@mbari.org or Duane Edgington duane@mbari.org.

Placing files in scratch space can be done through Windows Explorer or through scp (secure copy).

With Explorer, simply launch Windows Explorer from any Windows machine, and surf to [\\beowulffish\scratch\username](#). From here you should be able to drag and drop your Quicktime or AVI clips for processing here.

Alternatively, you can use sftp (secure ftp) to transfer files. This is useful if your files are on a Linux machine and you need to transfer to the Linux scratch space for processing. For example, to copy the file dive1399.mov from the Linux machine Mako to the scratch space on Beowulffish:

```
dcline@mako# scp dive1399.mov dcline@beowulffish:/mnt/scratch/dcline/dive1399.mov
```

Login to Beowulf cluster

Once your file is transferred to the scratch space, the Beowulf cluster can be accessed via ssh on any X-window on Linux, or through a Cygwin X-display.

To login, open an X-window and type:

```
ssh beowulffish -l username
```

When prompted to do so, enter your password. This should be the same password you use on your Windows desktop machine.

Accessing your scratch space

Now, access your personal directory on scratch space:

```
# cd /mnt/scratch/username
```

Your file should be in the scratch space. Check this with the listing command ls:

```
# ls
```

```
dive1399.mov
```

Once the file is in the scratch space, you can process.

Testing clip for correct transcode (optional)

It is first recommended you test this clip to make sure it can be transcoded into a format that can be processed. Transcoding is the process of converting from one format to another. In this case, we need to test if the clip can be converted from AVI or MOV to the PPM frame format required for the AVED software. **This is an optional step**, but if this is the first time you've process this type of video clip, you should try this once to make sure it's correct. After verifying this, you can skip this step for all future clips create like this one. The script *testclip* can be used for this. To use, simply type *testclip <filename>*, e.g.:

```
# testclip dive1399.mov
```

If this is successful, you should see the message *<filename> valid*.

```
dive1399.mov valid
```

Otherwise, if the clip is invalid, you'll get the messages *<filename> invalid* and do not continue with the next step.

Processing with runclip

There are two scripts for processing: *runclip*(non-parallel) or *runclipbeo*(parallel version). The latter is significantly faster but produces slightly different results due to the optimizations for speed in the saliency algorithm. Each of these scripts processes and organizes the results for you.

To view a list of processing options, type *runclipbeo* or *runclip* at the prompt with no arguments.

runclip

The options are as follows:

-f – Enable framegrab mode. Allows direct capture of video from tape deck to PPM. This is currently a “wild” frame grab with no timecode cue support.

-d - Used to deinterlace fast moving video. This option is used for midwater video but has not been used for benthic processing. Do not use this if your video is stored in progressive format and does not need deinterlacing.

-i – Also deinterlaces video but used for slower moving objects. Use this deinterlacer in place of the other if deinterlacing is necessary but video does not contain much motion. Do not use this if your video is stored in progressive format.

-v – Specifies whether the event images should be saved. These are the event-centered images gathered for use by the AVED classifier. These are cropped singleton instances of each event found by the AVED event detection software.

-c – Specifies frame range to be processed. Allows one to process a portion of a quicktime movie, but processing must start at 0. In order to process only the first fifty frames user can type in -c 0-50 followed by other options or the file name.

-j – Crops borders of the original movie files. To use the option type in -j height,width. The height specified will crop this number of lines from the top and bottom while the width will crop the specified number of lines from both sides. Thus, if the original movie is 720 by 480, the option -j 62,8 will reduce the movie to 704 by 356 or $720 - (2*8)$ by $480 - (62*8)$.

This is used to crop out black border artifacts that exist in our video due to the downsampling from HD to SD.

Thus, for benthic and some midwater video with black borders, the specified parameter used for the cropping option is -j 66. This might leave a bit of a border along both sides but allows mpeg movies to be created with the time code intact. This is also necessary to

convert the video aspect ratio format to something compatible with our mpeg 2 encoder which is used in the runclip script to save the results.

-p – Allows user to specify the habitat of the video. The option is followed by specifying either benthic or midwater ie. – p benthic.

An example of processing a clip is as follows:

```
# runclipbeo -d -v -c 0-50 -j 66 -p midwater dive1399.mov
```

This will process the first 51 frames of the video using midwater settings. It will deinterlace the video, save event images and crop height by a total of 132 lines (66 from top and 66 from bottom).

Once processing is complete, a new directory will have been created with the name of the movie file that was processed. Within this directory will be the following:

- The original movie file i.e. *dive1399.mov*
- An xml file which contains all the information gathered from processing i.e. *dive1399.xml*
- An events summary file containing the most important data from each event ie. *dive1399.summary*
- An mpeg movie file which shows the results of processing ie. *dive1399.results.mpeg*
- An *mbarivision_command* file specifying which contains the command used to process with the mbarivision program. (Note - older versions of runclip created a *transvis_command* file. This has been replaced with *mbarivision_command*).
- An mpeg results ie. *dive1399.results.mpeg*.
- A directory called *events* if the –v option was used. This directory stores the event images in ppm format. In the directory, the files would have the name *dive1399_evt00001_000292.ppm*. The name specifies the file name of the movie that was processed, followed by the event number, followed by the frame number.

Overriding options with MBARIVISION_OPTIONS environment variable

The runclip and runclipbeo scripts automatically chose all the options for you to process. These options can be overridden with the environment variable *MBARIVISION_OPTIONS*. There are over 60 different options to modify which can be found by entering:

```
# mbarivision -help
```

You can alternatively save the options to a file if they scroll by too quickly for you to read.

```
# mbarivision -help &>options.txt
```

You can then use your favorite editor to look at these.

For more details on what these options mean, contact Danelle Cline at dcline@mbari.org or Duane Edgington at duane@mbari.org.

For example, to override the option `mbari-cache-size`, use the export command:

```
# export MBARIVISION_OPTIONS='—mbari-cache-size=5'
```

This will override the number of frames used to compute a running average. This is used in the saliency detection and segmentation algorithm.

It's a good idea to check that your environment variable is set correctly before continuing.

```
# echo $MBARIVISION_OPTIONS
```

```
—mbari-cache-size=5
```

Once this is set, `MBARIVISION_OPTIONS` will override those options determined by `runclip/runclipbeo` and be set only for the duration of your session. You can alternatively place the export command in your `~/.bashrc` file which will permanently set these options for all your future processing.

Reprocessing a clip

The `runclip` script creates a directory based on the root name of the original clip. All results are placed into this directory and the original clip is moved to this directory (rather than copied, as it can be a very large file and this saves space).

There are several options to reprocess, depending on what you want to do.

If you don't want to keep the previous results, (1) move the original clip out of the processed directory into a different location, (2) remove the processed directory and finally (3) reprocess. E.g.:

```
# mv dive1399/dive1399.mov .
# rm -rf dive1399
# runclip -d -v -c 0-50 -j 66 -p midwater dive1399.mov
```

Another option that will keep the last processed results and create a separate directory for new results is to (1) move the original clip out of the processed directory into a different location and rename the clip, and then (2) reprocess. E.g.:

```
# mv dive1399/dive1399.mov dive1399-nodeinterlace.mov
# runclip -v -c 0-50 -j 66 -p midwater dive1399-nodeinterlace.mov
```

Commands for viewing frame or playing clips in Linux:

Playing mpeg clips

mpeg_play -dither color <filename.results.mpeg>

Playing quicktime clips

oqtplayer <filename.mov>

Displaying frames with graphics magic, or xv

gm display <frame.ppm> or <frame.dpx> or <frame.png>

xv <frame.ppm>*