

Notes for Kevin Gomes discussion:

Thursday, December 08, 2005

D.Cline

Last updated: Tuesday, December 20, 2005

Overview

Print out and discuss Jerome's work. Go over his presentation, discuss profiles: benthic, midwater, and transect versus non-transect. Papers at

<\\Tornado\ProjectLibrary\900260.AVED\Project.Documents\Jerome>

Administrivia

Login to nanomia and kgomes/aved/root.sudo administrivia notes. kgomes has sudo privileges, and needs to do matlab work as user aved, and all else as kgomes. Also have an account on beowulffish with mirror copy of nanomia ~/aved directory.

Matlab Overview

Run matlab as user aved. (Note may have conflicts as Andrew Chase will also be using the license.)

```
#ssh -X nanomia -u aved
```

```
>>cd Jerome/matlab
```

```
>>matlab &
```

Once in matlab run

```
% adaptiveThresholding baxter000552.ppm benthic
```

```
% imtool baxter000552.ppm and baxter000552_segmented.ppm
```

Sourcenaavigator

Sourcenaavigator is a simple IDE to navigate and compile code. Walk through test code using sourcenaavigator. Duane is amenable to buying commercial IDE so suggest asking him about this.

```
#ssh -X nanomia -l kgomes
```

```
>> snavigator
```

Discuss what this would replace in mbarivision code

```
>> emacs aved/mbarivision.C and aved/Segmentation.C
```

Image processing libraries

There are some image processing calls in the saliency toolkit. To start with, check-out Image_FilterOps.C/H, Image_Transforms.C/H in the ~/aved/saliency/src3/ directory. Duane also has some ideas on open source libraries. Matlab usually has text references to the algorithms used in the matlab code so you can research what algorithm was used.

Discuss test code and added switches

I've already added the switches to support the new algorithms to mbarivision. This involved:

- Adding-mbari-segmentation-algorithm switch for segmentation algorithms to mbarivision.C and ModelOptionsDefs.C:. Valid options are: BackgroundCanny or HomomorphicCanny and are case sensitive.

- Adding switch for canny segmentation dilate structure element to mbarivision.C, segmentation.C, and ModelOptionsDefs.C. Valid options are Benthic or Midwater and are case sensitive.
- Added functions for BackgroundCanny and HomomorphicCanny to Segmentation.C to fill in.

I also wrote some test code for you to get you started. See the file test-segmentation.C

```
>>emacs test-segmentation.C
```

Build with make test-segmentation

```
>>make test-segmentation
```

The stubs for the algorithms are in Segmentation.C. Put the algorithms here. Usage of test-segmentation is as follows:

To test frames 500-503 in the directory ~/originalppms/Baxter with the algorithm BackgroundCanny and structure element for benthic:

```
>>cd ~/originalppms/Baxter (Note, on beowulffish, processing is done in the scratch directory, not your home directory, so copy these files to the /mnt/scratch/kgomes directory and work from there)
>>test-segmentation -input-frames=500-501@0 ---mbari-segment-algorithm=BackgroundCanny --mbari-segment-algorithm-se-type=Benthic Baxter
```

To verify the segment algorithm is identical to the matlab algorithm Jerome created, I suggest you compare the segmented frames output from the matlab code and from the test-segmentation program. The test-segmentation program will output a file appended with __segmented.pgm. Use this file to compare against Jerome's matlab generated file appended with _segmented.ppm. Note that the PPM and PGM formats are slightly different, so to compare, you will have to convert the PGM to PPM or vice versa, then do a "diff" between them to compare bit-by-bit. The graphics magic "gm convert" command should provide the tools to convert to PPM/PGM formats.

Go through gcvs file check-in to aved/mbarivision

```
>>gcvs
```

Suggest creating branch for segmentation work, then merging with mainline when done.

Debugging

DDD debugger installed. Again, Duane is also amenable to buying commercial IDE with debugger if speeds up development. DDD usage to debug mbarivision:

```
>> ddd mbarivision
```

(Note - need to include debugging symbols to debug with gdb. Debugging symbols are not built by default. To add debugging symbols, add the -g option to the Makefile CPPFLAGS in mbarivision/Makefile). Also can create a .gdbinit file to automatically load breakpoints, and user arguments for example. See ddd help for info on this, or google for .gdbinit which is part of the gdb debugger.

Discuss project user doc:

This is where all user documents are:

[\\Tornado\ProjectLibrary\900260.AVED\Project.Documents\Project.UserManual](#)

Discuss runclip usage. To print help, simply type

>>runclip

Or look at “AVED Processing on the Beowulffish Cluster” in

<\\Tornado\ProjectLibrary\900260.AVED\Project.Documents\Project.UserManual:>

Which includes details on runclip. Note **this document may not be up to date as runclip and our work flow is constantly changing**. Please do update the word document if you make changes to runclip or any part of the work flow, and regenerate the PDF.

When done with segmentation algorithms, discuss release of mbarivision:

Do testing first on nanomia, then check-out and test on beowulffish. On beowulffish, mbarivision is installed in /home/aved/bin as the user aved so you'll need to rerun configure to install mbarivision to /home/aved, then **install the new version on beowulffish, install as user aved, e.g.:**

```
>>cd ~/aved/saliency
>>./configure --prefix=/home/aved
>> su aved -c 'make install'.
```

Also do a CVS update of the aved/scripts module as changes have been made to libruncip and runclip to incorporate the switches for segmentation algorithms. Then copy the scripts to the /home/aved/scripts directory

Ask Karen to do a CVS update of the aved/scripts in addition to the aved/mbarivision software. When all is working, ask Karen propagate changes to the AVED machines and send out an email summarizing the changes to the AVED team.

In aved/mbarivision directory

1. edit version.H and increment subversion e.g. 0.2.0 to 0.3.0 .
2. make clean
3. make
4. su aved -c 'make install' (need aved password here)
5. check-in all aved/mbarivision code using gcvs or other CVS based tool.
6. Use cvs or gcvs to create a tag on module with the format:
 - a. release-<year>-<release version with underscores>name, e.g. release-2006-00_03_00.

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>