

Overview

The objective is to factor out all the matlab code for your EITS changes, build the new mbarivision, test it, then check-in a new release.

Sourcenavigator

Sourcenavigator is a simple IDE to navigate and compile code. You don't have to use this, but it has a nice front-end to jump-to gcc build errors and fix them.

Image processing libraries

There are some image processing calls in the saliency toolkit. To start with , see Image_FilterOps.C/H, Image_Transforms.C/H in the ~/aved/saliency/src3/ directory. Matlab usually has text references to the algorithms used in the matlab code so you can research what algorithm was used.

Test code

I added test code for Kevin Gomes, who was asked to convert your Matlab segmentation algorithms, but he didn't do the work. See his directory on Nanomia at /home/kgomes/. Here are the notes I gave to Kevin on test code I wrote to get him started. I think this will be helpful to you:

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.
- Test code
 - >>emacs test-segmentation.C
 - Build with make test-segmentation
 - >>make test-segmentation
 - The stubs for the Jerome 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 that 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.

Check-in to aved/mbarivision using gcvs

```
>>gcvs
```

Suggest creating branch for segmentation work, then merging with mainline when done. Also check-in any changes to libruncip, and be sure to add help in runclip for any new command switches.

When done with converting Matlab code, add new release version to mbarivision, then have Danelle test your new release on Beowulffish.

Test mbarivision first on nanomia, then ****important****contact Danelle to check-out the new release, and test on beowulffish.

To add a new release, in aved/mbarivision directory

1. edit version.H and increment subversion e.g. 0.2.0 to 0.3.0 .
2. make clean
3. check-in all aved/mbarivision code using gcvs or other CVS based tool.
4. 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.