Space Details

Key:	BIAUV
Name:	900807_BenthicImagingAUV
Description:	
Creator (Creation Date):	headley (Jan 15, 2009)
Last Modifier (Mod. Date):	headley (Jan 15, 2009)

Available Pages

- Home
  - BI AUV Data Processing
  - BIAUV Image Processing Strategy
  - BIAUV Navigation Processing Strategy
  - BIAUV Software
  - Documentation
  - Image Data Workflow Diagram
  - List of Software Tasks
  - Meeting Notes
  - VARS Loading

This page last changed on Sep 14, 2010 by brian.

Benthic Imaging AUV (BIAUV) Project Home

VARS Image Processing

- Loading Benthic Imaging AUV Data into VARS
- Launch VARS for BIAUV

Visit the links below to see other BIAUV pages

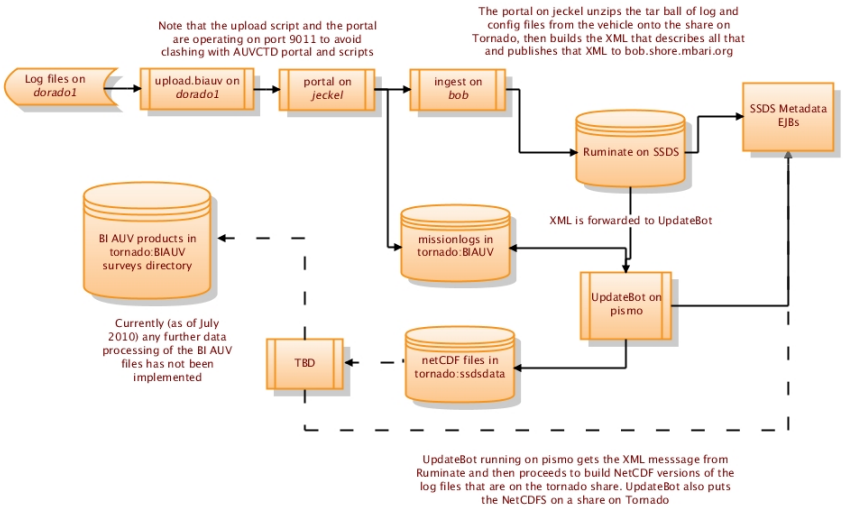
BI AUV Data Processing

This page last changed on Jul 28, 2010 by kgomes.

This page describes the Benthic Imaging AUV data processing. It contains a processing diagram, software needed, and assorted more detailed information.

BI AUV Data processing Diagram

This is a version of the AUV CTD Data Flow Diagram that has been adapted for the Benthic Imager AUV project. It describes the intended processes that are being instantiated for the BI AUV.



BI AUV Data Processing Software

The following table lists and briefly describes the software that is needed for BI AUV.

Application	Purpose	Development Needed (Base)	Development Needed (Images)
Log files	Store data from instruments on vehicle	Verify config files are correct	Camera driver needed, must log metadata
Upload Script	Create message to send to portal and tar up data files	Verify correct message creation and all files are included	Add handling of images?
AUV Portal	Copy log files to Tornado and construct XML and send to SSDS	Make sure all log files are transferred and XML is correct	Add image handling
Ingest	Stores metadata message	No change	No changes
Ruminate	Take XML and store in Metadata database	no changes	no changes
UpdateBot	Convert log files to NetCDF	Verify NetCDF files OK	Should we do anything with images?
		(new)	(new)

Post-process (system TBD)	add metadata to images, calculate best navigation values?		

SSDS AUV Portal Information

The portal runs on jeckel, 134.89.12.45. The distribution is managed out of the ssdsadmin account, which has as its home directory /private/var/automount/ssdsadmin. In that directory there is a dev/ directory, which contains auv_ctd/, which contains the distribution. Normally, you can update the distribution from CVS in this directory. The build is then run and the distributions are copied to the /usr/local/auv_bi. Then the jndi.properties and auvportal.properties files are edited to ensure the portal is configured correctly. See the README file in that directory for more information (also look at the README in the /usr/local/auv_ctd directory.

Normally, the portal should just be running. If you execute a 'ps -aux' hopefully you will see an entry for '/bin/sh portal 9011 &' followed by a java command. If it is not running, you can start it by using the following (on jeckel):

```
cd /usr/local/auv_bi
/bin/sh portal 9011 &
```

Upload script on Benthic Imaging AUV

First you telnet to the dorado machine (IP 134.89.32.106) (see AUV team for username and password) and cd the directory where the script is run
The upload script on the BIAUV is in /home/dorado1/auv-qnx/auv/altex/onboard/scripts.

You can run the script in interactive mode as ./upload.biauv. Alternatively, you can run the script with the mission name (YYYY.MM.DD) and campaign name (AUVBI) on the run line. For example:

```
./upload.biauv 2010.089.01 AUVBI
```

The error message "Bad directory specification (AUVBI2009168), skipping..." is a bug in AuvDataXfer.cc (called as 'adx' from the upload script) which luckily has no impact on the script's behavior. The authorities have been notified.

BIAUV Image Processing Strategy

This page last changed on Jul 30, 2009 by graybeal.

This page captures the image processing status and notional plan as of July 28 2009. It was written by John Graybeal based on discussions with Brian Schlining, and work to data for BIAUV and Rover.

The automated processing shown at the Image Data Workflow Diagram shows the different pieces of the image processing task. This document addresses three tasks: adding metadata to images; putting images into SSDS; and creating an image index for Brian's software. We'll deal with those in reverse order.

Images index for mosaic tool

For driving the mosaicing of images using the mosaic tool, Brian requested an index in the form of an ASCII text file. John created a draft, with the following headers/content (this content is tab separated):

```
Time	Center Latitude	Center Longitude	ImageURL	Altitude(m)	Height(m)
Width(m)
20090413T153133.00	38.922086	119.077900	http://ssds.shore.mbari.org/auvbi/
missionlogs/2009/2009092/All%20images/DSC_3717.jpg	1.212	1.001	0.997
20090413T153135.59	39.036678	118.963317	http://ssds.shore.mbari.org/auvbi/
missionlogs/2009/2009092/All%20images/DSC_3717.jpg	1.212	1.001	0.997
```

Brian responded with the following requests for changes:

- add Heading (whatever form AUV reports, but make clear which it is)
- add Pitch and Roll (for calculating skew if/when needed)
- make sure the longitude sign is correct (we're negative)
- a precision indication would be very helpful ("either an error estimate or number of significant digits in lat and lon")

Regarding the last point, lat/lon precision is easy because it's fixed in the data set and it's either just the number of digits we provide, or (for meaningful precision) can be calculated from the worst case value of precision in the inputs; whichever is fewer is the one to specify in the header. Error estimate (accuracy) is hard, because it's different at every point (depending on how recent the last GPS or similar fix is) and the distribution has a long tail (worst case error is potentially pretty high). I'm going to suggest starting with precision, but being prepared to add accuracy fields if the data is noisy.

I now suggest the following:

- Time (ISO8601)
- Image Center N Latitude (to n places)
- Image Center W Longitude (to n places)
- Altitude (m)
- Heading (geographic)
- Pitch (deg up)
- Roll (deg right)
- Image URL
- Image Height (m)
- Image Width (m)

Note that calculating Image Height and Image Width will require some mathematics to incorporate the Altitude, Pitch, and Roll of the AUV, as well as the optical characteristics of the system. The camera is a fixed focus camera (or used that way at least), so that makes life easier.

The sequence for performing this post-processing vs the BIAUV Navigation Processing Strategy will eventually be important, as the post-processing has the potential to significantly modify the exact location of each image. However, I propose first using the AUV-calculated navigation information for creating this index, because that should provide a smoothly connected AUV path, which will be the most important characteristic for single-track (non-overlapping) AUV path mosaics.

Putting images into SSDS

As of this writing, images are still being conveyed to shore using a PC or Mac equivalent to mount the images volume on the BIAUV. There is ongoing discussion of making the volume mountable in the native OS, allowing direct copy. In this case it will be a trivial matter to put the images on a store (assuming space is available), and then indexing them in SSDS.

Kevin and I have discussed how best to do this. It is probably not ideal to put an entry for each of 1600 images in a single AUV mission in SSDS. (For example, it would make the navigation of data in SSDS even more difficult.) The best idea I can recall from the discussion is an image index document, that is listed as a resource associated with the source camera sensor.

Note that although it is possible to add images to the SSDS database, we are not doing so because (a) we have not found an optimal submission strategy for such large files, (b) it occupies a lot of space in the database, and (c) we don't have good ways to display image data formats in the SSDS interfaces.

Other considerations

It makes sense to do the images for the AUV in the same way they are managed for the Benthic Rover. This implementation is the most general-purpose, and would be an acceptable starting point for the Rover. Shortly thereafter the Rover may wish to catalog every image in SSDS, as they are not taking as many images and have multiple deployments under which the images are taken.

There is some consideration for putting images into either VARS or the Expd-style systems. This can be accomplished readily if the images have been indexed using SSDS, so SSDS submission will be the first step in any case.

Adding metadata to images

The best mechanism for adding metadata to images in most common formats is ExifTool, which is free software that can be run from a Unix command line. It has an immense range of metadata-setting options and Mike and Kent have both used it. And it can be easily incorporated in any scripting language (or programming language).

The more tricky question is deciding what metadata to add. There are 3 possible additions: corrected timestamps, navigation data, and expanded metadata of the style Mike McCann adds to ROV data.

An important reason to add this metadata to images is to make the images self-describing. Often images from the BIAUV will be separated from their original storage environment (in presentations, on the web, sent to colleagues, and so on). As is generally the case, including metadata inside each image will be important. It allows the image to be traced back to its source (institution, platform, and observation date), and all metadata is easily discoverable with most image software (rather than chasing down the image manually in a data system and using that data system to look up the metadata).

Corrected image timestamps

The existing images are expected to contain timestamps to centiseconds, in both the GPS and 'image timestamp' parts of the metadata. There was some uncertainty (last I heard) about whether this mechanism was working correctly. If it is not, the project has expressed the opinion that they can derive the correct timestamp from other logs. Whichever path is chosen, this should be a priority, as otherwise timestamp information may be hard to recover.

Navigation data

Although a process will create a text document that describes each image (see above) and its navigation/attitude information (see BIAUV Navigation Processing Strategy), it will also be valuable to include that same information in the image itself. This should be a minor modification to the image processing software, and can be made a part of the expanded metadata described in the next section.

ROV (CenSeam) expanded metadata

This metadata will include contact information, copyright and rights of reuse, and so on. It can also include annotation information if annotations are created for the images. Software to add expanded metadata is already operating for the ROVs, and should be easily adoptable for this purpose. The specification for this metadata entry is provided at http://marinemetadata.org/community/teams/imagemd-censeam.

BIAUV Navigation Processing Strategy

This page last changed on Jul 30, 2009 by graybeal.

This page captures the navigation processing status and notional plan as of July 2009. It was written by John Graybeal based on meetings with Mike McCann and Rob McEwen.

The automated processing described at BI AUV Data Processing produces a gridded netCDF (.nc) navigation file based on the raw on-board (real-time) navigation. This follows a similar flow to the AUVCTD, and the operational procedures are described at the bottom of the file. They produce netCDF files that are visible to SSDS and accessible on the web.

The first-level strategy discussed with Rob is to perform post-processing in the same way as the AUVCTD, namely to correct the locations based on fixes obtained from GPS. In the case of BIAUV, the fixes occur at the beginning and end of runs, rather than at the top of each yo-yo maneuver, but the effect would still work. Mathematically, the error discovered at any fix is averaged into preceding positions, using linear weighting (vs any previous fix).

The idea for the software per discussions with Mike M is to copy the AUVCTD code, then modify it for this purpose. The copy has been made and lives in the biauv CVS repository at biauv/src/matlab. (omatlab is junk and can be ignored.) There are a number of products that the AUVCTD matlab files create that are not useful for the BIAUV; these parts of the matlab could be commented out.

Note that running this software is a manual process on the AUVCTD, requiring adaptations for each mission, and presumably would be the same for BIAUV.

The BIAUV has 2 additional navigation sources that are not available with the AUV CTD. The first of these is position corrections from sonar on the ship (telemetered acoustically); the second is position corrections from a target beacon (which presumably has been surveyed in to some known accuracy). Each of these could be used to improve the navigation, either in real time or in post processing.

In either case, some thought is required, as neither navigation system is as solid as GPS. Whereas a GPS navigation fix is guaranteed accurate within a meter or less, the fixes from the ship and beacon are subject to considerable noise. You do not want to "improve" the nav by fixing it to a bad/noisy position, especially if you might get multiple noisy positions in a row. (There is no Kalman or other filtering in the AUVCTD nav software.)

So the next steps for integrating these other fixes into the real-time and post-processed navigation will require some mathematical analysis to create the appropriate algorithms for possibly noisy fixes. (That was as far as the project got, last I was aware.)

BIAUV Software

This page last changed on Jan 15, 2009 by headley.

The software effort for biauv consists primarily of control software (Rob McEwen) and camera control and metadata management (k headley).

This space will be used to maintain development notes and communicate with the team on software related issues.

Development Environment

System	Description
mvc-bi	OS: QNX 4.25 Hardware: x86 (HP Kayak) Description: QNX development host on AUV subnet Location: Headley's office (A210)
hummingfish	OS: Fedora Hardware: x86 PC Description: Linux development host (gphoto) Location: Headley's office (A210)
moonjelly	hosts auv-qnx CVS repository
docking AUV	target vehicle for biauv

Information

- See the News Page for recent activity on BIAUV software
- View Attachments on this page for related documents
- Source code is maintained under CVS in the auv-qnx repository:
  - auv-qnx/auv/altex/onboard/camera
  - auv-qnx/auv/altex/onboard/taskIF
  - auv-qnx/auv/altex/onboard/Simulator
- Documentation is maintained under CVS in the biauv repository
  - biauv/doc
  - biauv/doc/reference
  - See Attachments to this space for additional documents
  - use doxygen to build biauv documentation; a doxygen config file is included in this page's attachments and in biauv/doc

Documentation

This page last changed on Jan 15, 2009 by headley.

This page contains shared wiki documents and links to project documentation hosted on Alfresco

BIAUV Alfresco Documentation Space

BIAUV Use Case Doc

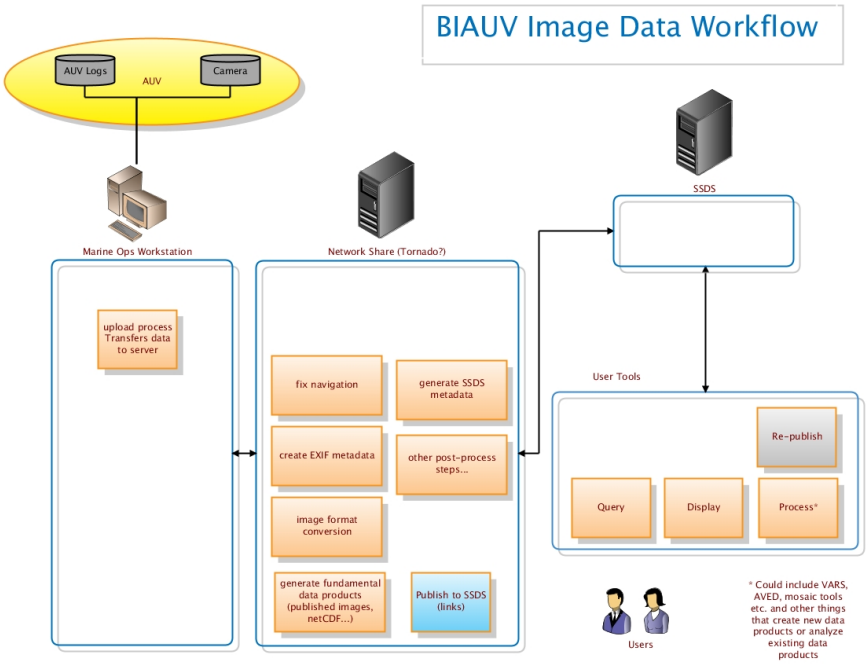
GWIZ Utility Notes

Camera Control Software Class Diagram

Camera Control Software Package Diagram

Image Data Workflow Diagram

This page last changed on Jan 15, 2009 by headley.



List of Software Tasks

This page last changed on Mar 19, 2009 by graybeal.

These are pretty much in sequence, with early tasks in parallel. The numbers are my own time estimates (in hours). They will prove generous in some cases, but my error bars are pretty high. Based on these estimates, we'd get all the raw data and image metadata into SSDS, but I doubt I could get all the way through the navigation post-processing.

The assignments are based on what has been inferred from discussions.

Design/Documentation

- (Mike) finish documenting existing AUV data processing
- (John:4) review existing (HTML) processing description, edit as needed
- (John:4) Create copy of data flow diagram reflecting the below processing changes for BIAUV
- (John:2) Update Brett's diagram of processes at https://oceana.mbari.org/confluence/display/BIAUV/Image+Data+Workflow+Diagram

Data To Shore and SSDS

- (Hans...?) Modify on-board script to copy data to BIAUV share on shore
- (John:6+10) create instance of portal software to get data into SSDS from BIAUV share (install, test)

Image Handling Preparation/Design

- (Hans) Verify access to camera images from on-board computer (👎 it appears only manual access available)
- (John:4) verify GPS time in image Exif data; write command to export key metadata from images
- (John:4) define any critical metadata/data for camera logging (device ID, items logged/ image times, strobe times?)
- (John:8) determine SSDS image storage/reference protocol (aligned with Rover, ideally)
- (John:8) prepare test data set for Brian to work with

Camera Interface Implementation

- (?) Verify camera control operations, write camera driver
- (?) Create camera log with sufficient metadata in header for image post-processing
- (?) implement camera logging
- (?) define process to copy images to shore
- (John:80) add (portal or equivalent) handling of images (probably via separate script)

BIAUV Data Post-Processing

- (John...) post-processing of BIAUV data (new version of auvsdb)
  - (John:24) basic image metadata
  - (John,?: 160?) improved navigation (depends heavily on the implemented on-board navigation and nav resources)
  - (John:12) image metadata improved with improved navigation
  - (?) science products when CTD on board
- (Brian) image tagging ⚠ *review from minutes and Brian*
  - (Brian) grab images with SIMPA
  - (Brian) annotate images
  - (Brian) put images into final location (need to decide what that is)

Meeting Notes

This page last changed on May 20, 2009 by hobson.

VARS Loading

This page last changed on Sep 15, 2010 by brian.

Loading images from a mission into VARS

The software for loading an Benthic Imaging AUV mission is kept on the computer Seaspray. To load an AUV mission into VARS do the following:

1. ssh to Seaspray

   ```
   ssh seaspray
   ```
2. Locate the mission directory that you want to load. The AUVBI directory is mounted on Seaspray at `/mbari/AUVBI`. Note the following:
   - The full path to the mission directory you are going to load. e.g. `/mbari/AUVBI/missionlogs/2009/2009173/2009.173.07`. It should contain the navigation.log and camera.log. If those are missing from the directory then the load will fail.
   - The location of the image directory for that mission. e.g. `/mbari/AUVBI/missionlogs/2009/2009173/Images`
   - The URL mapping of that specific image directory. e.g. `http://ssds.shore.mbari.org/auvbi/missionlogs/2009/2009173/Images`.

   > **Be Careful**
   >
   > Double check that URL in a web browser. If it's correct then you should see a directory listing the images. Don't just copy and paste the example URL above!! It's only for missions from 2009173. Also, case is important in the URL; the auvbi directory is mapped to http://ssds.shore.mbari.org/auvbi **(all lower case)**
3. cd into /opt/vars-imaging/bin

   ```
   cd /opt/vars-imaging/bin
   ```
4. Execute the following, substituting in your info:

   ```
   gsh ../scripts/groovy/load_biauv.groovy <mission directory> <image directory> <image directory URL>
   ```

   Here's an example:

   ```
   gsh ../scripts/groovy/load_biauv.groovy \
   /mbari/AUVBI/missionlogs/2009/2009173/2009.173.08 \
   /mbari/AUVBI/missionlogs/2009/2009173/Images \
   http://ssds.shore.mbari.org/auvbi/missionlogs/2009/2009173/Images
   ```

Notes

It's OK to load the same mission more than once. It just updates the information, it does not create duplicate missions.

To annotated the images:

1. launch VARS from http://seaspray.shore.mbari.org/webstart/vars-images/varsannotation.jnlp.
2. When opening a video archive to annotate, select the **Open Existing** radio button and look for your mission. It will be named something like **`biauv-2009.173.03`**.