Final Report:

SmartSonar II

Making seafloor mapping data more tactical through faster processing

Principal Investigator: Thomas B. Reed IV,
Contact Info: reed@oicinc.com

Company Name: Oceanic Imaging Consultants, Inc.
Address: 1144 10th Avenue, Suite 200
Honolulu, HI 96816
Phone: 808-539-3706

Supporting Agency: CEROS
Contract No. MDA972-02-2-0002
Date: September 5, 2008

Table of Contents

SmartSonar2: Executive Summary

Acoustic seafloor mapping systems produce raw sounding and navigation data, which must be cleaned and reduced in post-processing operations to produce final charts. While data from surface ships more often than not contain accurate navigation data from GPS sensors, data from sub-surface manned and un-manned vehicles lack this luxury, and as a consequence often require greater effort in processing. This greater required effort often results in significant delays between acquisition of the data and availability of interpretable final products. To make data from sub-surface platforms more tactical, processing systems should be able to produce final products in no more time than that required to collect the data.

The Naval Oceanographic Office (NAVOCEANO), Stennis Space Center, MS, Code NPL is tasked by CNO N23 (Operational Support) to process and analyze high resolution bathymetric information collected by USN platforms in support of unique and specific initiatives. CNO is reliant upon the data processing staff at NAVOCEANO, Code NPL, for the production of these high-quality bathymetric maps and databases. This processing is one of the primary focus areas of NAVOCEANO, Code NPL, and it is a task at which they are very proficient. The quality of data produced and the timeliness of its delivery to this program office and subsequently back to the Fleet is extremely important.

Oceanic Imaging Consultants (OIC) has been working with NAVOCEANO over the past 12 years under various contracts supporting modernization of NAVOCEANO's hydrographic survey fleet and data processing capabilities. NAVOCEANO has been using OIC's post-processing software since 1996, and has become quite proficient at using it to produce finished products for the fleet, but at a non-trivial cost in time and effort. Anecdotally, data from one day of acquisition can easily take over a week to process.

CEROS (Center of Excellence for Research in Ocean Sciences) funded OIC to design, develop, demonstrate and deliver to NAVOCEANO new processing capabilities which would at least triple the existing processing speed, without a loss in accuracy. OIC has completed this task, and delivered our next-generation post-acquisition processing product in the form of CleanSweep3. This new package presents a global approach to processing of navigation data and the co-registered swath bathymetry and imagery. In trials conducted at NAVOCEANO by trained NAVO operators, CleanSweep3 delivered processing times 3 to 10 times faster than those of the previous generation software. The acceleration in processing was largely due to automation of previously manually executed tasks, including correcting navigation, image processing and swath mosaicking. NAVOCEANO's enthusiastic support for this new product is indicated by its recent contracting of OIC for development of additional features post demonstration of CleanSweep3 this past June. While designed in response to specific NAVOCEANO data processing needs, CleanSweep3 is a general purpose post-processing package offering similar processing speed and ease for all types of seafloor mapping data. By processing data in less time than it takes to acquire it, CleanSweep3 effectively promotes post-processing from a task which must be done post-mission, to an activity which can be executed by ships cadre on mission, while in transit between areas or during vehicle surface intervals.

Introduction

Seafloor mapping systems such as sidescan and swath-bathymetric sonars produce acoustic imagery and raw sounding data. While significant processing can occur in real-time, the production of properly geo-coded mosaics and bathymetric maps from this raw data is largely a post-acquisition processing task. Traditionally, there have been two approaches to post-processing: batch and interactive.

Batch Processing

"Batch" post-processing systems would essentially re-play the raw data, apply default processing and filtering procedures, and produce monolithic results, as shown below in Figure 1. By "monolithic" we mean that the result of the processing would combine all passes of the survey data into one result, with no ability to intelligently pick and chose between different passes over the same area. All data were treated equally, and assumed to require no "manual" correction of navigation, gain nor image balance. Batch processing can be very fast, but the results are not known until completion. If the resulting product was un-satisfactory, the processing steps must be repeated in their entirety.

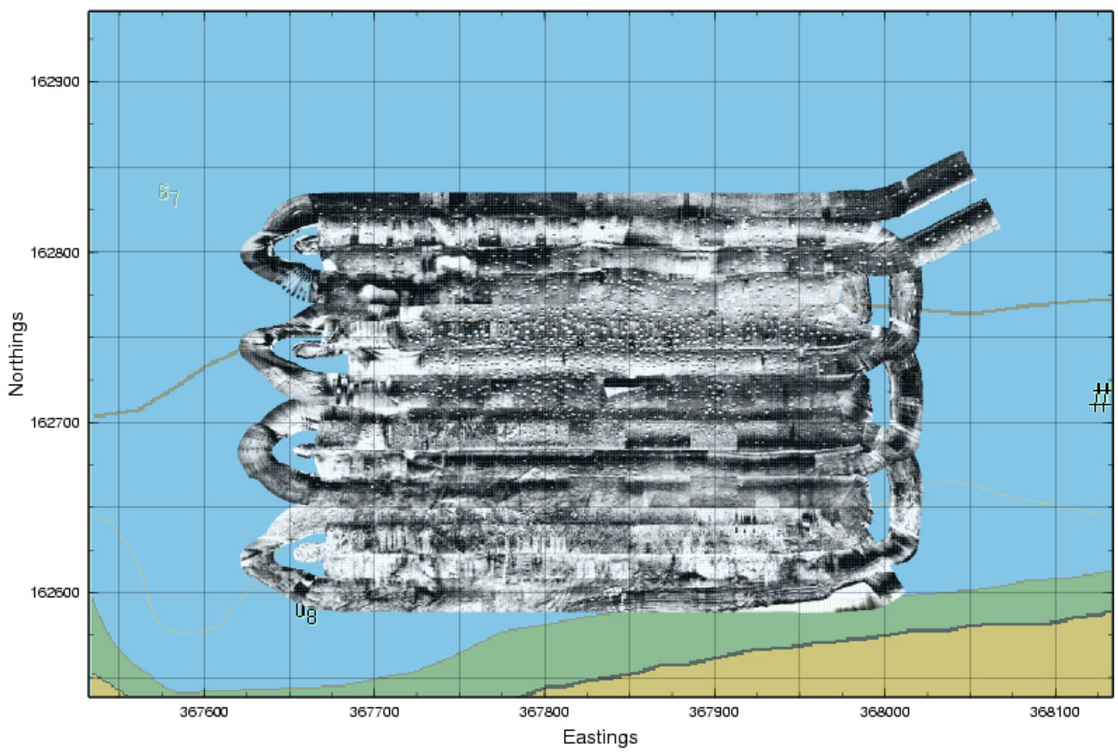


Figure 1. Typical sidescan mosaic resulting from "Batch" processing. Due to the lack of operator oversight, features in adjacent tracks do not match up well. Also, data logged during turns, which is often less than optimal, was included, possibly over-printing good data.

Interactive Processing

In contrast to the “turn-key” nature of batch processing systems, “interactive” processing systems proceeded in a more peristaltic fashion. A common interactive approach was to break the survey up into segments corresponding to the survey “legs”, so that the data from each individual leg can be processed and mosaicked into separate “swaths”. Depending on the processing package, in one way or another, the user could then collage the swaths to produce a final output.

Typically, the data was read to determine the overall coverage of the area, and displayed to the user in a “coverage map” as seen in Figure 2. To process the data in an “interactive” scheme, the user selected a portion of the data corresponding to one “leg”, as shown in Figure 3.

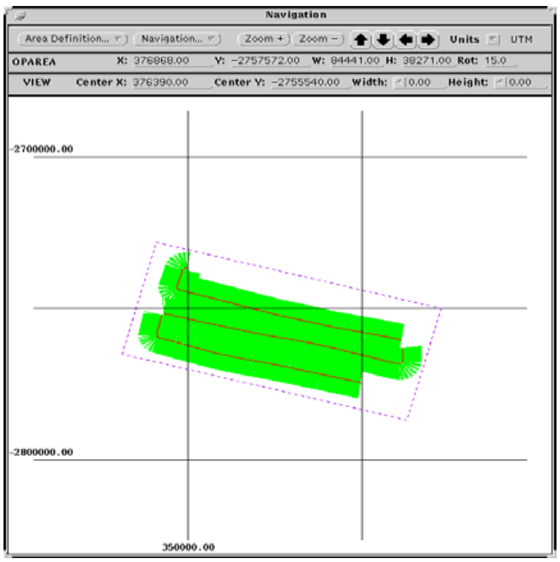


Figure 2. Typical "coverage map" showing navigation track (thin red line) and data coverage (green).

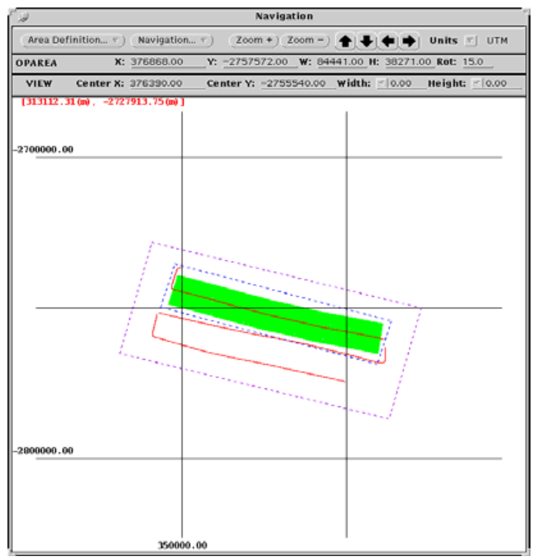


Figure 3. Manual selection of the data for one "leg".

The data for the selected “leg” would then be passed to a navigation and attitude processing tool, as seen in Figure 4a. The ‘cleaned’ navigation and attitude data could then be used for processing and geo-coding the sonar data. The data would then be passed to another editor, as seen in Figure 4b, to clean up data noise, fix gain changes and suppress artifacts.

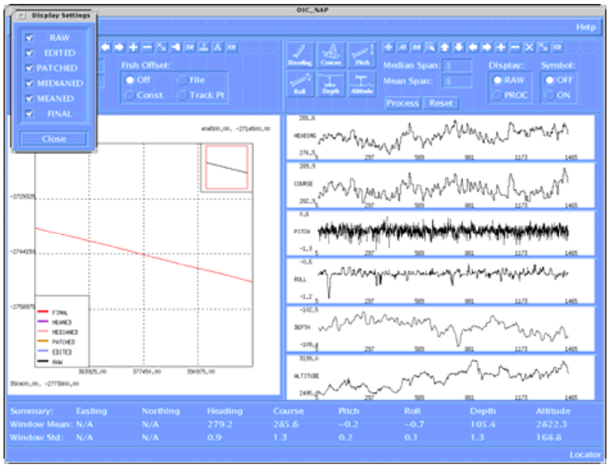


Figure 4a. OIC_NAP, for meta-data processing

Figure 4b. Typical sonar data editor.

After cleaning the meta-data (navigation and attitude) and the sonar data, the two were combined in a geo-coding process to produce a “single-track swath”. Each swath is a geo-coded image independent from all other swaths, but once created, can be “collaged” into a final mosaic, to produce an overall picture. An example of the processing of the three “legs” of data represented by the coverage map seen in Figure 2 can be seen below in Figure 5.

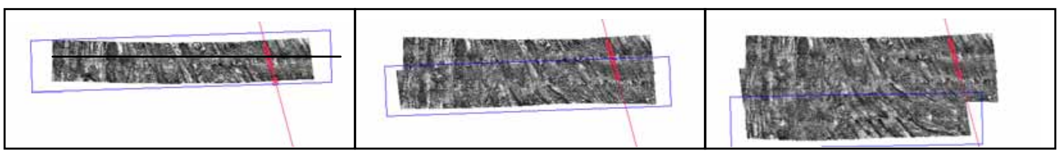


Figure 5. Progressive building of a mosaic from three independent swaths.

Presuming all meta-data are correct, and there are no errors in navigation, nor offsets in heading, pitch, roll or other data, the features in the collaged swaths should line up, building a seamless “mosaic” of the seabed. More often than not this was not the case, and the user had to manually deal with the resulting misalignment of swaths.

One approach which we called InterNAV, deals with the mismatch by declaring one swath to be correctly located, and then finding features in that swath which show up (but possibly at a different place) in adjacent, partially overlapping swaths (Figure 6). This technique worked as long as three conditions were met: this first swath actually was properly located; common features existed in both the first swath and the overlapping swaths; and the features could be easily and correctly detected in both swaths. If however, the first swath is poorly located, or if the wrong features are matched, this approach can cause more harm than good, and will consume non-trivial time.

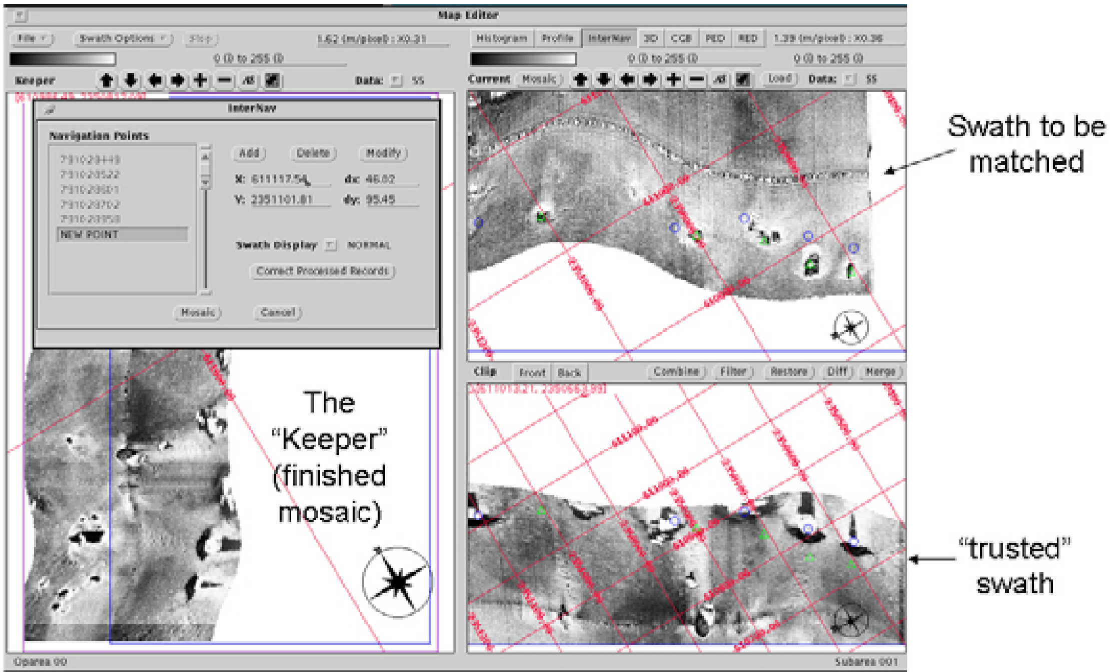


Figure 6. An example of manual navigation adjustment, called InterNAV. Features in the new swath are matched to corresponding features in the "trusted" swath. The real issue is, which swath do you trust?

A better approach would honor not just one, but all known points (such as surface navigation fixes for an AUV survey), and then try to best fit all features. Regrettably, the interactive approach of processing one line at a time does not permit this.

Background for this Work

The Naval Oceanographic Office (NAVOCEANO), Stennis Space Center, MS, Code NPL is tasked by CNO N23 (Operational Support) to process and analyze high resolution bathymetric information collected by USN platforms in support of unique and specific initiatives. Swath bathymetric data is collected to produce charts for purposes of safe navigation. While traditional hydrographic surveys rely upon continuous GPS position fixes for survey navigation, much of the work sponsored by N23 utilizes submerged platforms, which do not have the luxury of continuous navigation fixes. As such, they rely on dead reckoning, utilizing Doppler Speed-logs and heading sensors. Between surface GPS fixes, non-trivial navigation errors can accumulate.

Previous efforts at NAVOCEANO to process these data sets and correct the navigation errors relied largely on manual detection and matching of features in the sidescan imagery collected with the swath bathymetry. This navigation matching consumed a significant amount of time, and due to its iterative nature and requisite re-processing, could significantly delay production of final bathymetric products.

Oceanic Imaging Consultants (OIC) has been working with NAVOCEANO over the past 12 years under various contracts supporting modernization of NAVOCEANO's hydrographic survey fleet and data processing capabilities. CEROS previously funded OIC under SmartSonar1 to work on automation of real-time sonar tuning, reducing operator loading and improving data quality. Our SmartSonar2 proposal to CEROS was to re-engineer the processing of the swath bathymetry data to take advantage of all the navigation data present, as well as any information that could be inferred by matching features in the sidescan and bathymetry to either known features, or the same feature in overlapping swaths. Our goal was to automate the routine processing steps, reduce processing time and improve adjustment of navigation, so that the staff at NAVOCEANO could produce equal or better quality bathymetry data in less time. By re-engineering the post-processing approach, and developing a blended "batch-interactive" approach, OIC aimed to improve processing time by an order of magnitude.

In this CEROS funded SmartSonar2 effort, OIC has developed, demonstrated and delivered our next-generation post-acquisition processing product in the form of CleanSweep3. This new package presents a comprehensive, global approach to processing of navigation data and the co-registered swath bathymetry and imagery. In trials conducted at NAVOCEANO by trained NAVO operators, CleanSweep3 delivered processing times 3 to 10 times faster than those of the previous generation software, which typically relied on an interactive approach to processing. The acceleration in processing was largely due to automation of previously manually executed tasks, including correcting navigation, image processing and swath mosaicking.

While this proposal speaks directly to the needs of NAVOCEANO in processing this specific data, the work developed here actually answers a much larger need, providing faster and more accurate processing for any survey involving data from submersibles, deep-towed sonars, ROVs or AUVs. By processing data in less time than it takes to acquire it, CleanSweep3 effectively promotes post-processing from a task which must be done post-mission, to an activity which can be executed by ships cadre on mission, while in transit between areas or during vehicle surface intervals.

Methods

In the SmartSonar2 project, our methods were guided by two simple goals:

- **Reduce processing time by a factor of three or more**
- **Retain bathymetric accuracy of 95% of soundings within 2% of reference depths**

Our approach to achieving these goals was three-fold:

1) design and implement a new, comprehensive, global approach to navigation processing
2) design and build a new "navigator" capable of "re-navigating" the data based on both intermittent GPS fixes, and user-supplied anchors or match-points
3) automate routine processing tasks, to allow creation of "first-cut" batch mosaics with a minimum of effort.

OICSwath was developed in 1996 to answer a need for more control than available via conventional batch processing. It utilized the interactive model of processing, allowing the user to break the data up into "legs", hand-tool each leg, and interactively collage swaths into mosaics. It supported detailed processing down to the level of the individual ping, automated and manual editing tools, and our InterNAV tool, as shown above in Figure 6, for track-to-track feature matching and navigation adjustment. CleanSweep2, released in 2005, was largely a port of OICSwath from UNIX to Windows. While CleanSweep2 offered an improved design and interface, it still operated in the same interactive model of processing each leg of the survey separately. While effectively offering limitless control in the ability of the operator to fine tune the data and make navigation adjustments, the interactivity these packages offered were both their strength and their limitation. They REQUIRED interactivity and consumed significant amounts of operator time.

In CleanSweep3, we envisioned a departure from this, to produce a more "batch-interactive" tool, which could make use of all the information in the survey, so that less effort was required of the user, while still retaining interactive control over data processing and mosaicking. This required development of a new back-end database, a new Graphical User Interface, a new navigation model, automation of pre-processing tasks, a new InterNAV algorithm, new bathymetry processing tools and an improved mosaicking strategy. We detail these developments below, and conclude with a description of the use of the new CleanSweep3 at NAVOCEANO.

The New CleanSweep3 Database and Processing model

Traditional survey work consists of running a number of parallel, slightly overlapping survey "legs", also referred to as "swaths". Winds, currents, environmental conditions and operators contribute to each "leg" being somewhat unique in terms of the processing required, so it seemed reasonable to process the survey data on a "leg by leg" basis. In previous incantations of CleanSweep and OICSwath, the focus was on selection of data for one "leg" of the survey, i.e. the section of data between two turns. This was predicated on the general belief that survey data during turns was useless, and often the turns were conducted outside the area of interest. Unfortunately, this "leg" based processing imposed the restriction that any model for navigation processing was restricted to the data within a leg, and thus could not take advantage of data past the turns.

If, for instance, a well-known, well positioned bottom feature were seen in one swath, one could easily position that swath to show the feature in the correct place, but no method existed to allow neighboring swaths to automatically inherit this position re-adjustment. Similarly, if a roll offset were determined based on one set of data, no method allowed all the roll data for the survey to be equally adjusted. The operator had to adjust each line individually. The "global" approach to processing both navigation and sonar data developed in CleanSweep3 obviated these limitations.

We have designed a new "track"-based model, wherein a track is simply defined as a time-continuous section of data. Ideally, a track would span a survey (i.e. contain numerous, parallel "legs" and their intervening turns) and be able to make use of all information in a survey. In this way, position fixes at either end of a survey could contribute to the entire model, not just the model for the leg in which they lie. The process flow for the new Global navigation model is summarized by the following diagram:

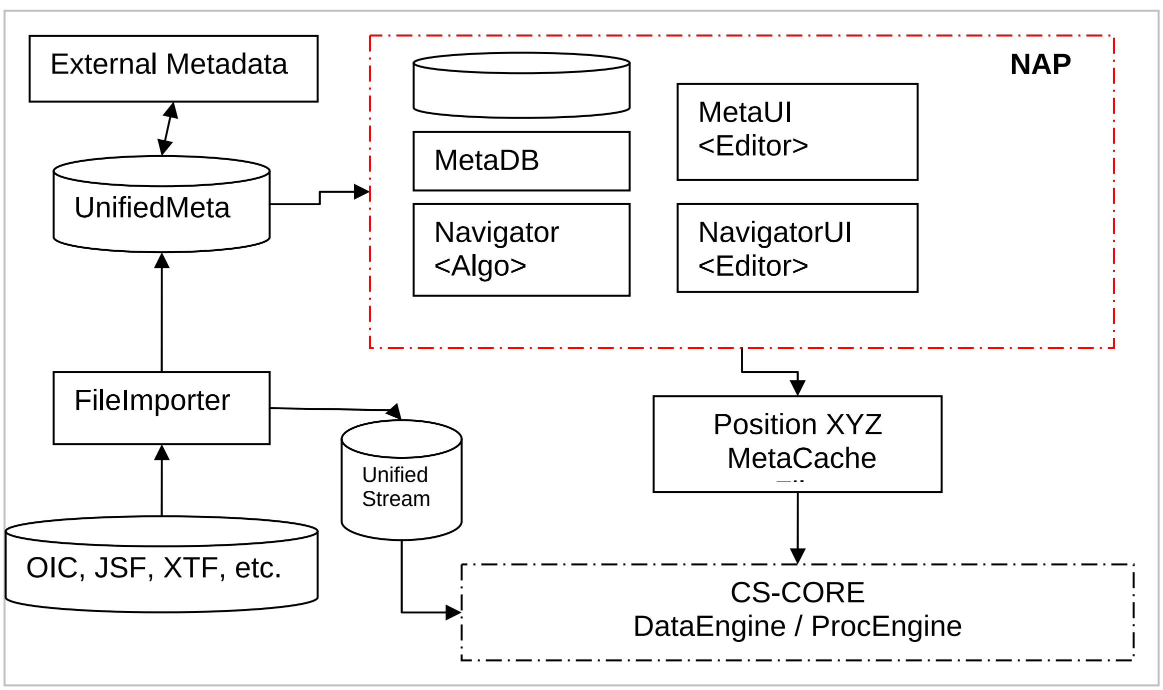


Figure 7. Schematic of new OIC CleanSweep3 processing.

In the figure above, “NAP” (red box) refers to Navigation and Attitude Processing. By separating the processing of the navigation and attitude data (which we refer to as “meta-data”) from the CleanSweep (CS) core processing, we allow a global approach to the navigation model unconstrained by the processing of the survey data.

The chief philosophical difference between CleanSweep3 and its predecessors is in the granularity at which data are processed. In leg-based processing, all legs stand on their own. Every leg must be examined, edited and processed, even if it has no errors, and would not require any special attention. In CleanSweep3, the time-continuous Track is the unit. Any adjustment within a track affects all data, so all data in a track must be accessible. As all data are accessible, it’s easy for the user to see the effect of adjustment of the meta-data (say roll) on the acoustic data processing, and of the feature-matching on the navigation. As all tracks require common processing due to systematic errors during acquisition, the operator can define default processing based on a few looks at the data, and mosaic everything. Only the regions where default processing did not produce acceptable results need be inspected further. If the majority of data are good, this “batch-interactive” approach will produce an enormous savings in time and operator effort.

CleanSweep3 Graphical User Interface (GUI)

The CleanSweep3 Graphical User Interface provides new tools for display, filtering and editing sensor, navigation and attitude data (meta-data). The old processing focused on the “leg” as the granular unit of processing, so one only had to keep track of one leg at a time. In contrast to this, CleanSweep3 breaks the survey into “tracks”. A “track” is defined for CleanSweep3 as a time-continuous segment of data. It can span minutes or days, one leg or an entire survey, as long as the data are continuous in time. The software automatically reads the raw data, and defines all tracks. Figure8 below shows the new CleanSweep3 interface, with about 2 hours of survey data from a survey of the Honolulu Harbor Kapalama Turning Basin, showing numerous “legs” contained in three “Tracks”, as indicated in the database description panel at the left of the GUI.

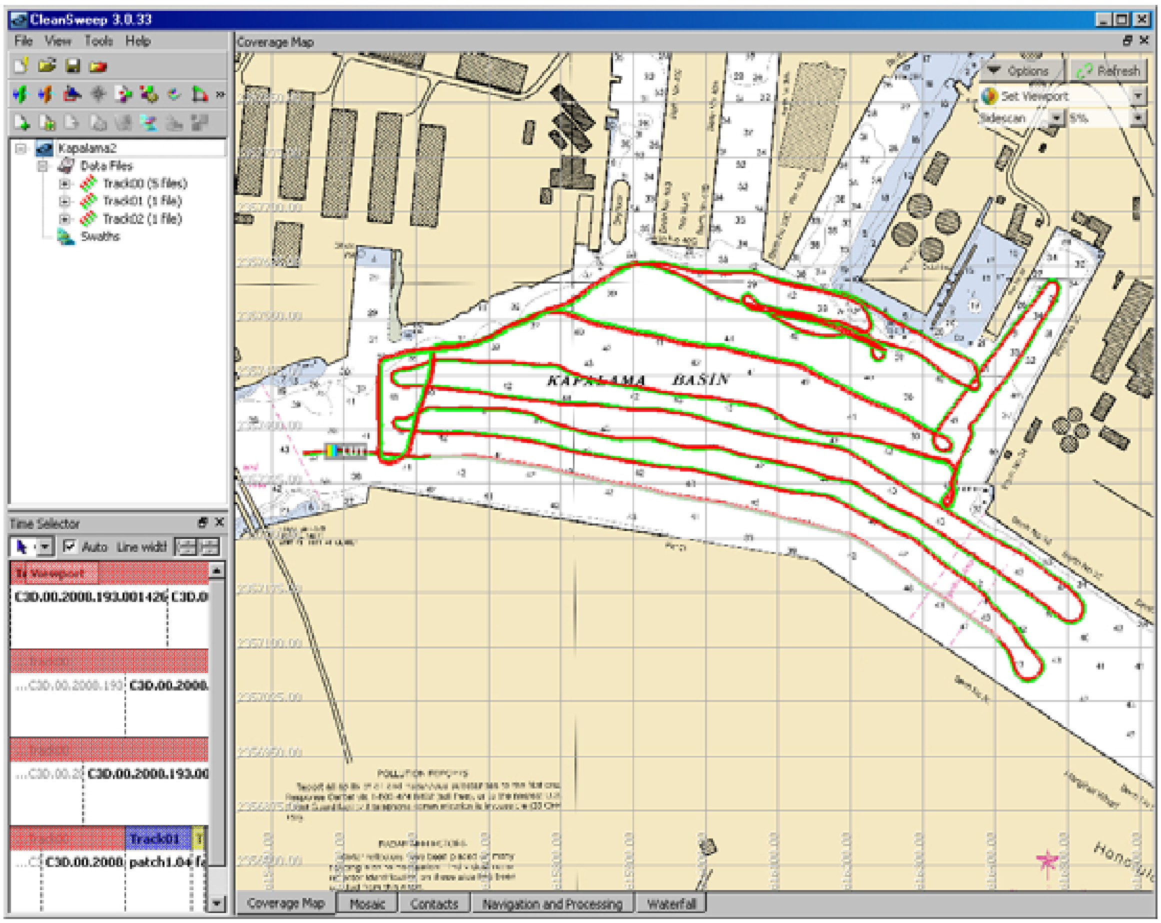
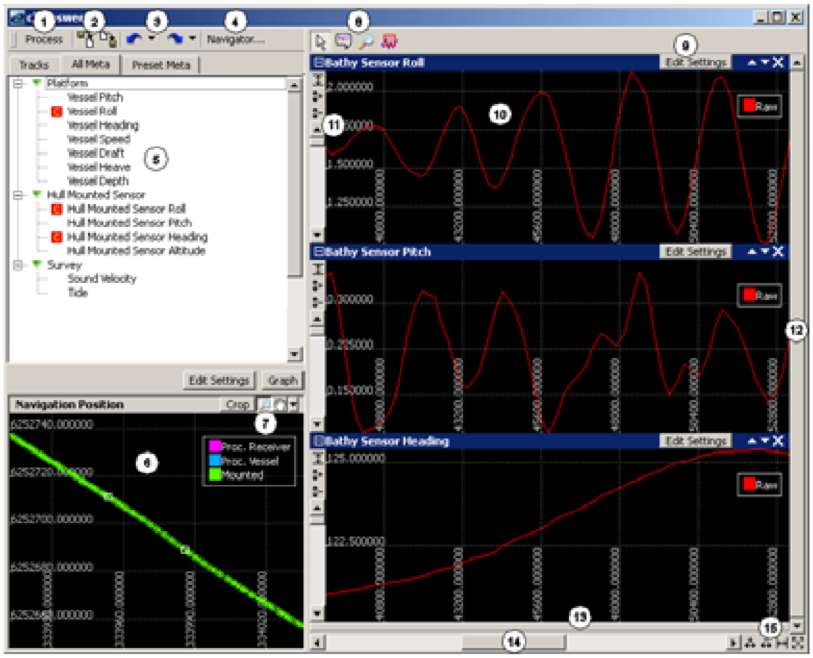


Figure 8. The new CleanSweep3 track display, showing tracks (3), the files that compose them (lower left) and a graphical trace of the tracks over a background chart.

Figure 9 below shows the new CleanSweep3 navigation and meta-data editor (MetaUI), with sections as indicated by the legend to the right.



1. Process button
2. Import/Export Meta Data
3. Undo/Redo buttons
4. Navigator button
5. Tracks/Meta Time-series
6. Navigation Position Map
7. Navigation Position Tools
8. Meta-Data Graph Tools
9. Meta-Data Edit Button
10. Meta-Data Graph(s)
11. Graph Vertical Zoom/Scroll
12. Graph Display Scrollbar
13. Graph Display Resize bar
14. Meta Data Time Scrollbar
15. All Graph Horizontal Zoom

Figure 9. New CleanSweep3 Meta-data editor.

The MetaUI Display replaces the meta-data processing portion of the Navigation and Attitude Processing (NAP) dialog in CleanSweep v1&2. Previously the NAP dialog would consist of a static set of processing nodes, displaying a static set of graphs, which would require the user to conform to the layout. The new MetaUI interface provides the user with a customizable method of viewing and editing the meta-data. The new interface allows greater flexibility in data display, as well as greater control over data editing, import and export.

While a default overview display is provided for navigation review and editing (zoom-able, expandable window in the lower left of the GUI) all other windows for any other component (pitch, roll, heading, cable-out, etc.) can be called up independently, with configurable view, zoom, filters and processing. All displayed windows are linked, so clicking on any meta-data trace (say, roll) displays both the value of that trace, as well as the value of all other displayed traces, plus the location of that sample on the navigation track.

The New Global Navigation Model

The Navigator

CleanSweep3 contains a new engine for re-calculation of sensor position, given all offsets, biases, raw navigation data and user input. We refer to this as "The Navigator" (Figure 10). The need for this tool is based on the simple assumption that rarely do we actually record the position of the sensor, but rather the position of the navigation system antennae on the survey vessel. From this, in real-time, one infers the position of the sensor based on dead-reckoning in the case of a submersible, or a layback calculation for a towed sensor. In post-processing it therefore makes sense to re-estimate sensor position based on all possible inputs, after any necessary cleaning and biasing, plus any user input. There are some systems which do provide real-time sensor position estimates, so the Navigator does offer an option to use the existing sensor position (with available editing, filtering and smoothing tools). In general, however, the real-time sensor position is only an approximation, and we can do a better job re-creating it with properly treated raw data inputs.

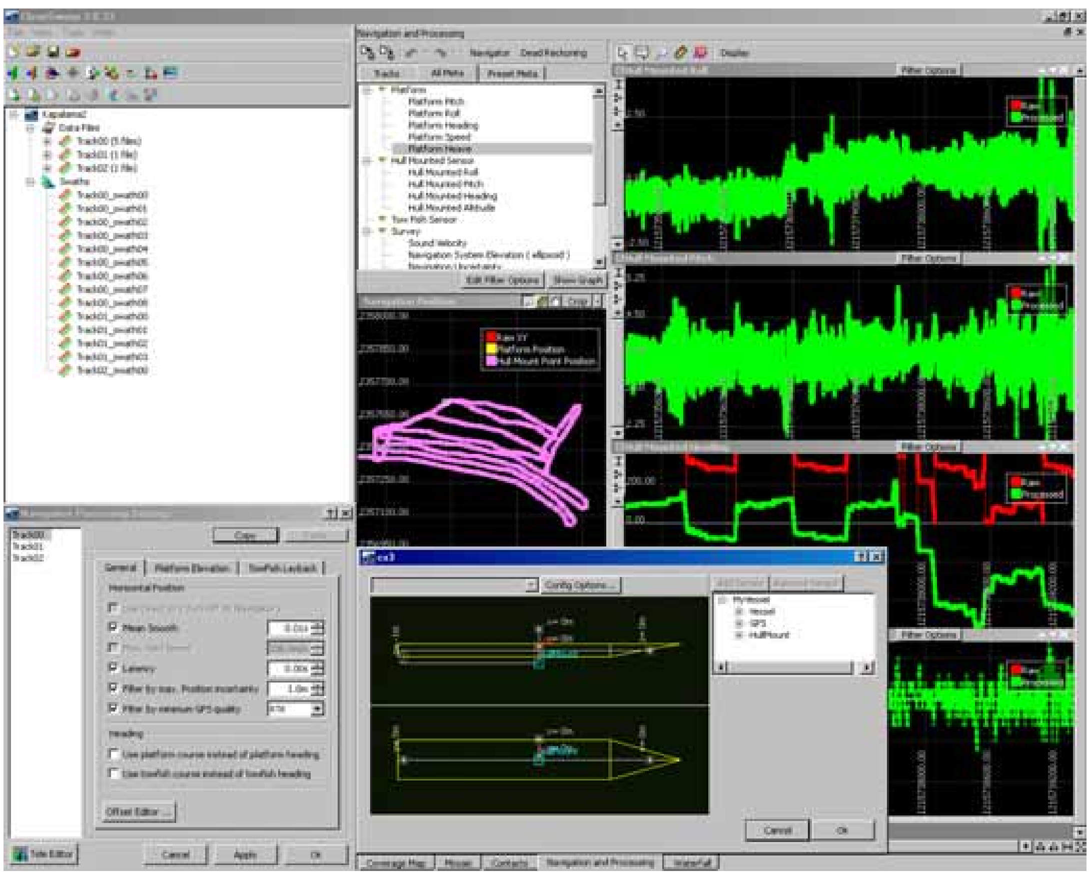


Figure 10. The new CleanSweep3 Navigator, with options.

While re-calculation of sensor position from cleaned navigation data is not new, the CS3 Navigator does support two novel features: a **Re-NAV** module, and **InterNAV2**.

The “Re-NAV” module supports automatic detection of offsets in sensor position estimates due to surface GPS fixes. These fixes, and the associated offsets, occur when a sub-surface survey platform surfaces, and re-acquires satellite coverage. Inevitably, the platform will have drifted from the real-time estimated position, and the Navigator will detect the offset. The “Re-NAV” engine will then calculate the offset between the real-time position estimate just prior to the fix, and the actual position at the fix, and “back-correct” the real-time data between the current fix and the previous fix, effectively removing un-accounted for drift. We document an application of this technique in the section which follows.

InterNAV2, on the other hand, provides an improved approach to using features visible in the sonar data to improve the overall navigation solution, by both constraining the navigation to absolute references, and enforcing co-registration of matching features in overlapping swaths. Similar to Re-NAV’s back-correction of position based on GPS fixes, InterNAV2 will automatically propagate navigation corrections based on feature matching made in one location both ahead of and behind the corrected point, so that the entire navigation solution improves. This was previously only approximated by iterative manual re-adjustment of all swaths. The new approach both improves the solution and reduces operator loading and time.

Introduction to ReNAV

With data collected from surface ships, one almost always has the luxury of continuous GPS position data. Even with towed sensors, we know where the towing vehicle is on the surface, and can well approximate the sensor position. On the other hand, with data from AUV’s or submersibles, we do not have GPS data continuously, and are more often than not reliant on bottom-lock speed logs and heading sensors to calculate a position estimate using dead reckoning from the last surface GPS fix. As errors in heading and speed tend to be small, this works fairly well. Nonetheless, errors do accumulate with time, so uncertainty in position grows with time. Policy may dictate that the vehicle must surface to re-acquire a GPS fix after position uncertainty exceeds some threshold, or they may just decide to live with the uncertainty. Either way, one starts with a GPS fix at the beginning of the survey, and a GPS fix at the end of the survey, or when the platform surfaces. By comparing the estimated position with the actual position at the time of the fix, one has a measurement of accumulated error. All position estimates calculated between the two fixes will contain an uncertainty, and in general that uncertainty in position (and the actual error in position) will grow with time since the last fix.

Previous approaches to dealing with this accumulated error involved selecting the swaths containing data which included a surface GPS, and using these as fiducials, then manually finding features in these swaths and matching them to similar features in overlapping swaths. More often than not, swaths between the swaths containing GPS fixes would not meet exactly, and the process would have to be repeated to minimize error by multiple adjustments. This laborious process consumed a significant portion of operator time at

NAVOCEANO and other processing groups. We theorized that since we had all the raw data upon which the real-time dead reckoning was based, we could take any GPS fix after the first one, and run the algorithm backwards, thus reducing the accumulated total error by half.

Our initial efforts focused on replicating the real-time Dead-Reckoning algorithm used during the collection of data for NAVOCEANO. We replicated the routine, and verified that we could replicate their results, but soon realized that we achieved the same results by simply detecting any GPS fixes after the first one (and their associated navigation jumps) and applying the accumulated error with an *inverse lever rule* back to the previous surface GPS fix. In this way, the position for the ping just preceding the second GPS fix was adjusted the most (as it had accumulated the most error) while the position of the ping just after the first GPS fix was adjusted the least. As this method produced identical results as running the DR routines backwards, but had no requirements of proprietary code nor data, we implemented our approach as a general solution to the problem of automatic offset detection and re-navigation (i.e. ReNAV).

ReNAV Position Accuracy Testing

We wished to validate that the new CleanSweep3 processing was not only faster, but accurate, in both a relative and an absolute sense. Matching features between overlapping tracks allows validation of relative accuracy. We wanted to verify that the new “Re-NAV” algorithms produced results which were accurate in an absolute sense as well.

Survey Design

In order to test the accuracy of ReNAV position calculations, we needed a set of data emulating that which would be collected by an AUV or other submerged vessel (i.e. one based on bottom-lock, dead-reckoning navigation with intermittent navigation fixes), collected simultaneously with actual GPS data. Since we could not do this from a real AUV (no GPS), we instrumented our survey launch with a Doppler Velocity Log (DVL), a heading sensor and a sidescan mounted on the bottom of the boat, plus two GPS units. The first GPS unit was used to initialize the DR (Doppler-based dead-reckoning navigation), while the second would be logged by a separate computer, giving continuous actual position. To emulate “diving” of the “AUV”, we simply submerged the first GPS in a beverage cooler full of water. To emulate “surfacing” to acquire a fix, we simply pulled the GPS head out of the cooler, and allowed the software to get a GPS fix.

We laid out a survey pattern as shown by the green line in Figure 11, and after initializing the DR navigation system with a good GPS fix, proceeded to attempt to run the survey pattern, intermittently pulling the GPS out of the cooler to get a new “Fix” and to update our dead reckoning. Again, the navigation in this track is derived from an initial GPS fix, combined with continuous output from a Doppler Velocity Log and a compass, as would be available on a typical REMUS AUV. This navigation data was fed to the sidescan logging computer, along with the heading, speed & altitude from the DVL. We refer to this as “DR Nav”.

At the same time, we logged actual GPS position data from a separate GPS to a separate but synchronized computer. The actual navigation track provided by that data is shown in Figure 12, to the right. Both GPS systems showed the same initial position, consistent with the planned survey route. The actual GPS data (in addition to the observations of the vessel operator) indicated that we were reasonably within course for the first leg, but had started to drift, and had drifted significantly by the end of the third long leg. At that point, we "surfaced" the DR nav GPS, so it could get a fix.

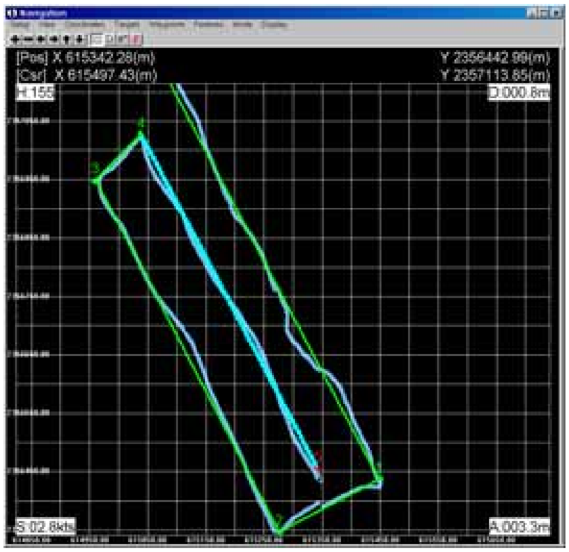


Figure11. Planned track lines in green, and Dead-reckoned (DR) navigation for test survey in blue.

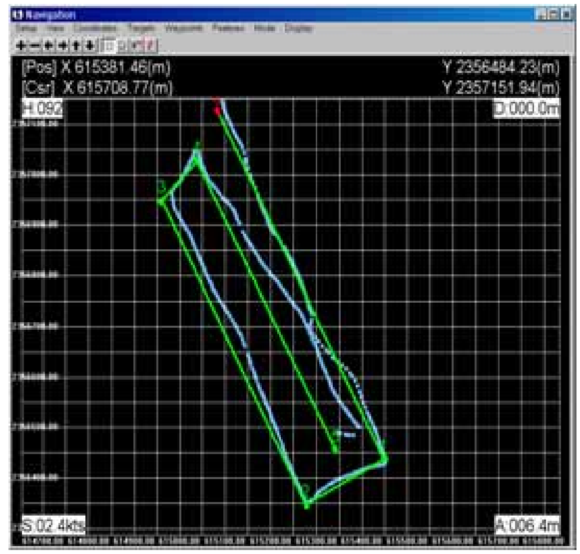


Figure 12. Planned track lines (green), and actual GPS vessel position (blue).

ReNAV Processing using CleanSweep3

We loaded the sidescan data containing the DR navigation data into CleanSweep3. The results of applying the ReNAV program within CS3 are shown in Figure 13 below. The yellow track shows the raw DR navigation position. The ReNAV track, created by the new CS3 algorithm, is shown in blue. The offset shows that the ReNAV algorithm has detected the offsets at the "surfacings" and applied them backwards to the data, successfully correcting for drift.

The "ReNAV" module built in to the new Cleansweep3 navigator automatically detects navigation "resets" (steps in position, with minimal step in time), and applies a reverse dead-reckoning algorithm, using the offset from the "reset", applying it backwards until the previous fix. The logic behind this is that the error in position during dead reckoning is due to cumulative growth in largely speed and heading errors. Linearly distributing the error backwards over time is the minimum cost approach to attempting to determine the actual path in the presence of offsets. Figure 14, shown below, compares the result of applying this algorithm to the emulated AUV data, to the actual GPS position data. While the solution is perfect at the offsets, the algorithm also does a reasonable job over the rest of the data, evenly distributing the error accumulated from the re-set at the bottom of the first line, to the end of the third line.

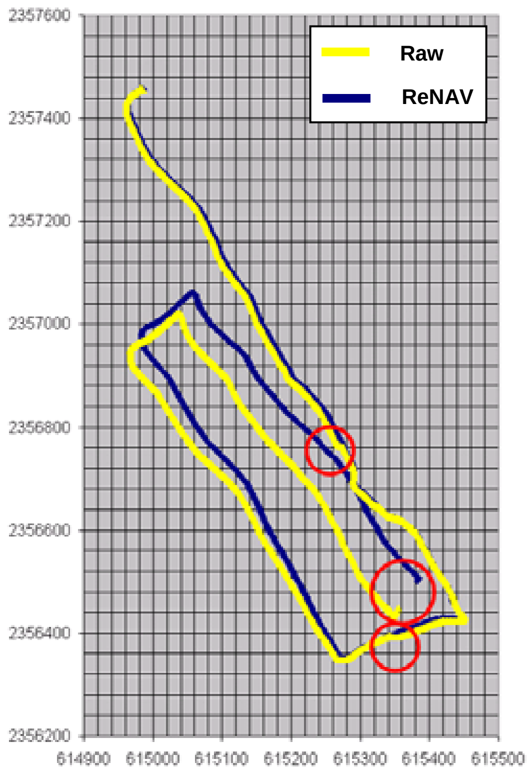


Figure 13. Raw DR navigation compared to nav processed using CleanSweep3 ReNAV.

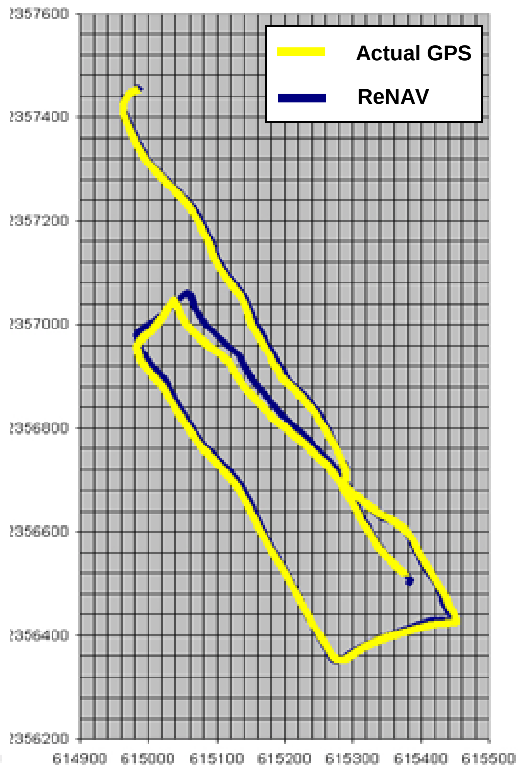


Figure 14. CleanSweep3 ReNAV (blue)

To compare the effectiveness and speed of the new CleanSweep3 to the old approach, we loaded the AUV emulation data into CleanSweep2. It took us 16 minutes to produce the mosaic of the three lines, shown in Figure 15, below. As one can see from both Figure 15, and the detail view in Figure 16, the default processing has produced a mosaic using the logged DR navigation data.

The results of this processing included offsets within the tracks (circled in Figure 15), as well as offsets between the tracks. The arc-like feature seen in the east-most track is clearly the same as the similar feature in the middle track – but which one is in the right position? In Cleansweep2 one would have been required to choose one as "correct" and move the other feature to match the position of the same feature in the "trusted" track. The ReNAV feature of CleanSweep3 obviates this compromise, and eliminates this manual effort.

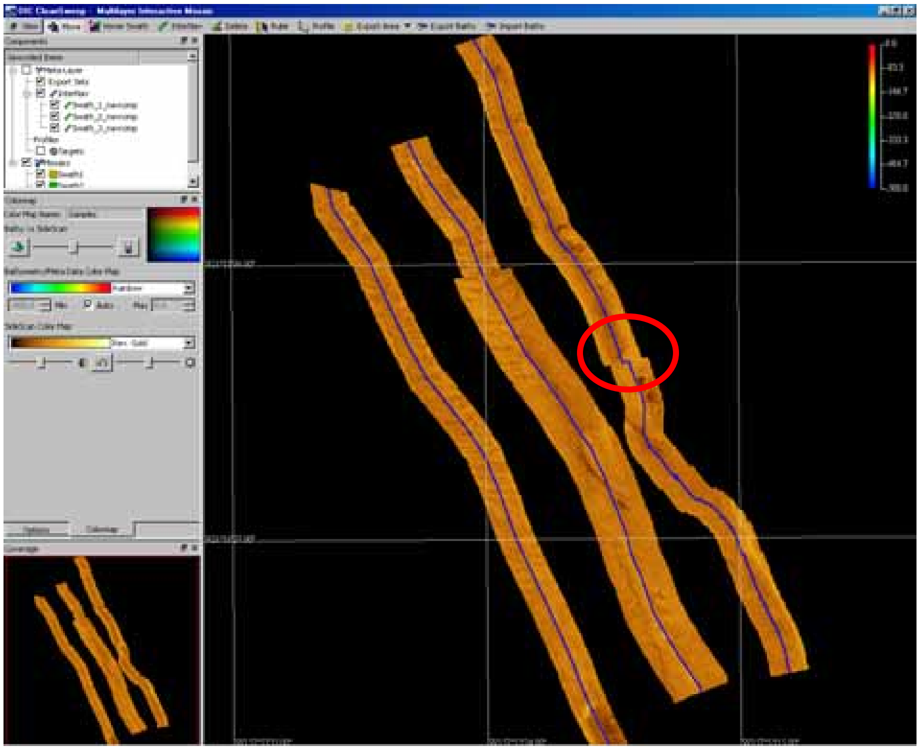


Figure 15. Mosaic of emulated AUV data using CleanSweep2. Note the navigation offset in Track 1 due to GPS "resets" (circled).

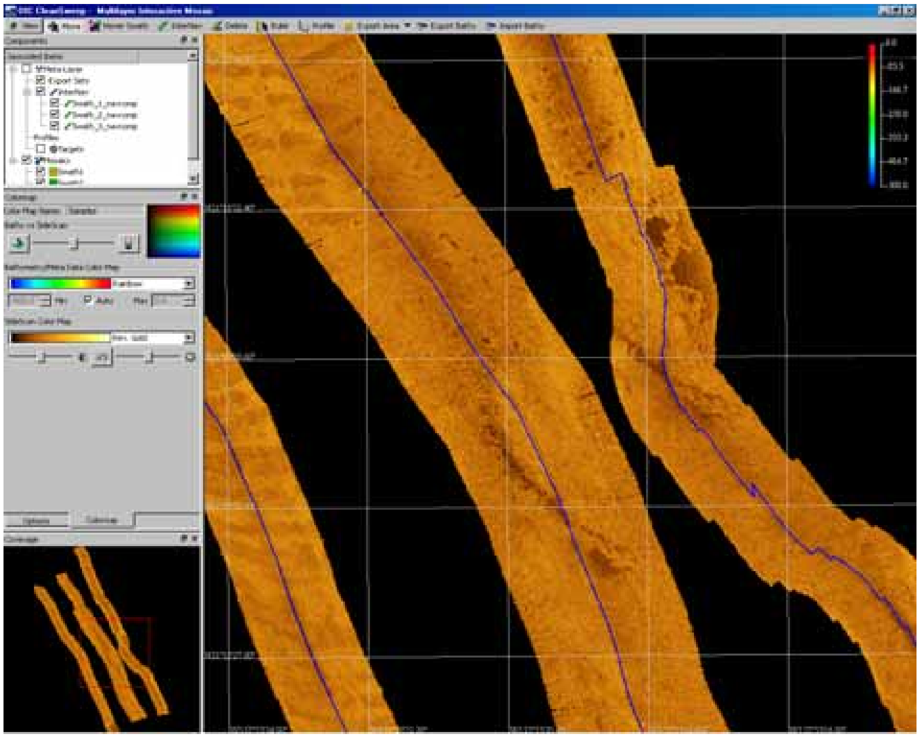


Figure 16. Detail view of the center portion of the CleanSweep2 generated mosaic shown in Figure 16. Note the likely match-up between the arcuate features seen in the eastern and middle tracks.

Figures 17, 18 and 19 below show the same data loaded into CleanSweep3. Figure 17 shows the data in CS3 coverage map, while Figure 18 shows both the original navigation data in red, and the result of application of the CS3 ReNAV algorithm in pink. The navigation offsets due to GPS "fixes" have been automatically detected, and the offsets applied to the original data to produce the adjusted track. The "re-navigated" data was used to automatically produce the mosaic shown in Figure 19. Details of the section of the mosaic where the two swaths overlap follow in Fig. 20.

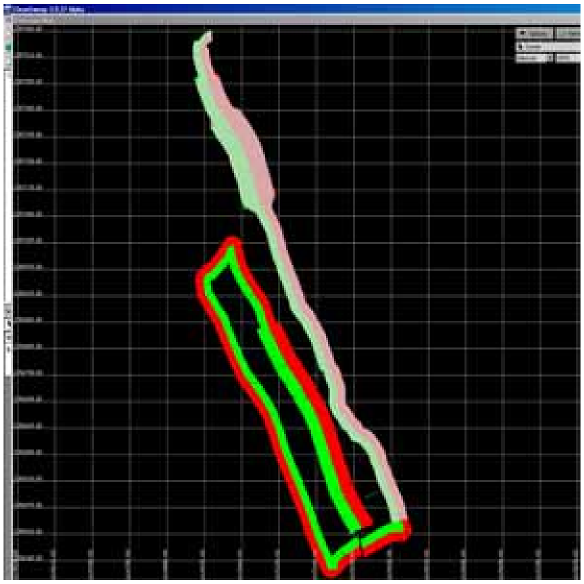


Figure 17. Test data in the CS3 coverage map.

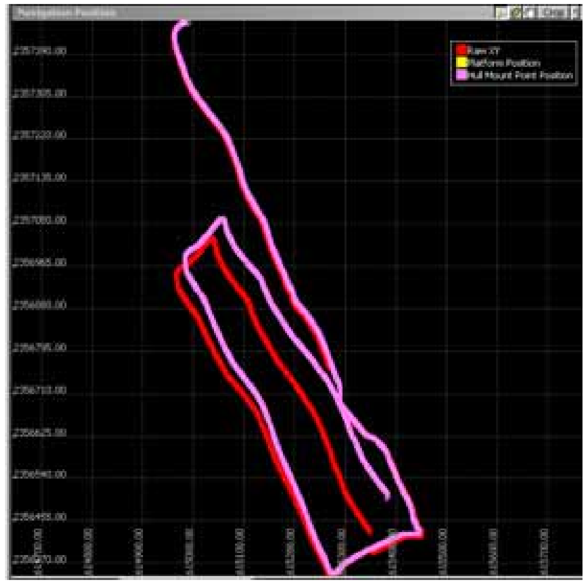


Figure 18. Test data after CS3 ReNAV

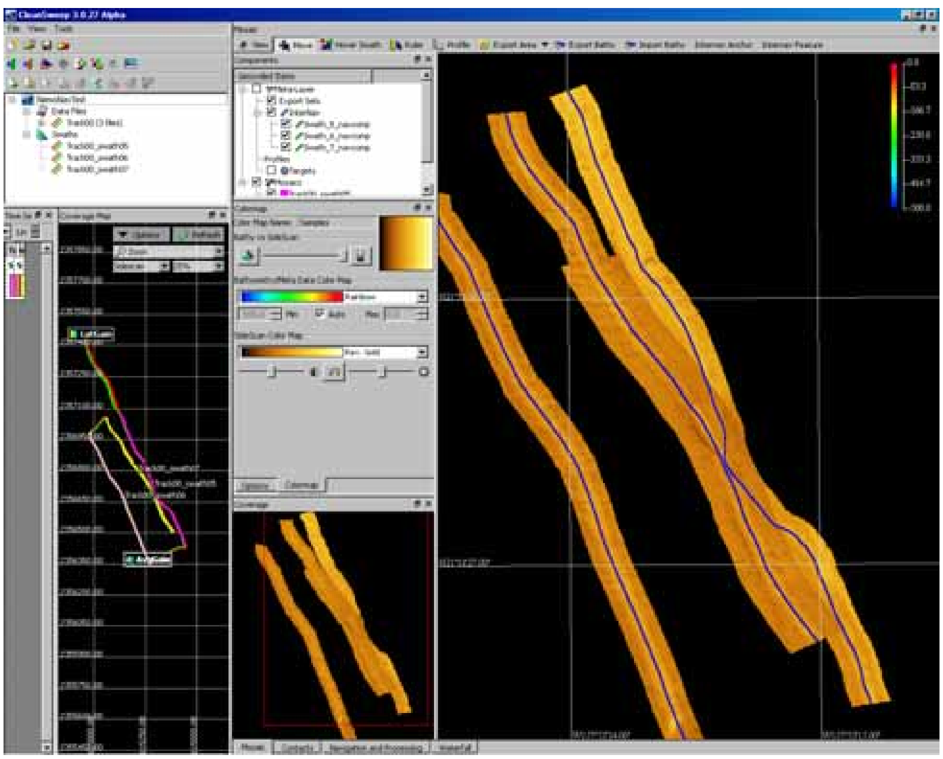


Figure 19. Test data mosaicked in CS3 after automatic ReNAV

Figures 20 and 21 display detailed views of the same arc-like feature in the first and third swaths, which now overlap, compliments of the CS3 Re-nav processing.

Until now, data processors would have had to choose which swath to move, and then iteratively adjusted all swaths. The ReNAV utility should prove an enormous time savings in processing. Loading, re-navigating and automatically mosaicking the data in CleanSweep3 took just under 5 minutes, and required no operator intervention. This compares favorably with the time of 16 minutes to process the same data using CleanSweep2, considering that additional time would be required in CleanSweep2 to manually adjust swath position, and re-mosaic each line.

While the co-registration of features seen in the re-navigated swaths is not perfect, it is now within the tolerance of the accuracy of the GPS. Furthermore, the navigational offset associated with each GPS "reset" was linearly re-distributed to all preceding navigation estimates back to the previous fix or anchor point. This results in are-navigation track that is smooth and continuous.

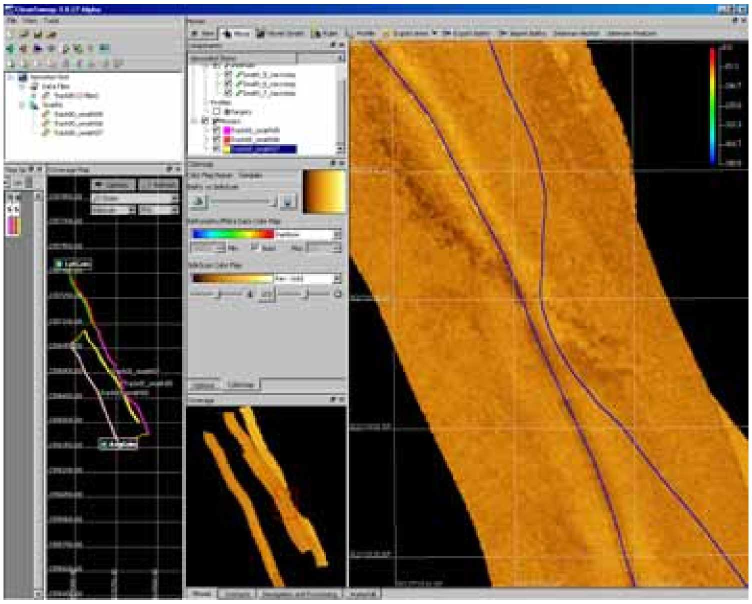


Figure 20. Detailed view of Track3, after ReNAV

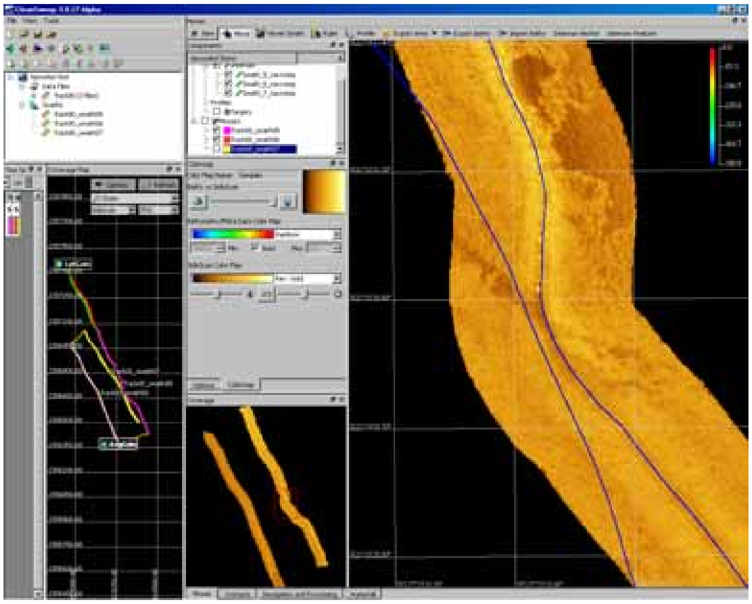


Figure 21. Detailed view of Track1, after ReNAV

ReNAV Error Analysis

Figure 14 above shows a plot of the actual GPS position superimposed on the re-navigated track. While the two tracks match exactly at the locations of the GPS "resets", there is some small offset further away from the resets. As the real-time dead reckoning is based both on estimates of speed and heading, errors in either or both will contribute to errors in position estimation.

Errors in speed from a Doppler Velocity Log will occur largely during turns or when the system is over non-reflective seabeds (seagrass, porous sands, etc.), or when due to platform roll one or more beams may no longer properly sample the bottom.

Errors in heading will largely depend on the device measuring the heading. Magnetic compasses may exhibit constant deviation and errors which change as a function of heading (variation), as well as errors due to the environment (passing a large, ferrous object such as an oil platform or a wreck, as an example). Gyroscopic sensors are immune to the above issues but still can provide only finite accuracy.

Given the uncertainty associated with the factors contributing to the errors in the original dead reckoning, additional attempts to further improve the re-navigation would probably yield minimal improvement. On the other hand, additional fixes, especially during turns when speed logs and compasses tend to fare worse, would significantly improve the final result.

Introduction to InterNAV2

The second new tool in the Navigator is referred to as **InterNAV2**. It's based on our existing InterNAV module, which allowed users to match a feature in a poorly navigated swath to a feature in a well navigated swath, and adjust the navigation accordingly. This has the weakness of requiring as many well navigated swaths, as poorly navigated swaths. It is also quite laborious, requiring numerous manual adjustments. InterNAV2 obviates these weaknesses.

During hydrographic surveys, we collect information about the seafloor (sonar data), and we associate it with the estimated position of the sonar at the time of collection. This allows us to create a geo-coded map representing an approximation of the seafloor. There are many sources of errors associated with the sensor position (GPS error, dead reckoning drift, etc). The intent of InterNAV2 is to give the user the ability to improve the position estimation precision and accuracy during post-acquisition processing.

Given a display of the data, a user recognizes some feature on the seafloor for which the geographical position is known a-priori, and can use this information to improve the sensor position accuracy at the time *t* this feature was observed. As an example, if we pass a navigational marker (channel marker, buoy, etc.) we will see the acoustic image of the marker's anchor or base in the sidescan image, and see the position of the marker on the chart. Given this, in CleanSweep3 the user can create an *Anchor Point* on the mosaic. It is simply an arrow that starts at the observed position of the feature in the sidescan or bathymetry data, and ends at the known true position of the feature on the chart, or background map. CleanSweep3 can also create these Anchor Points automatically, by detecting the offsets associated with GPS fixes at the surface. What this means to the data processor is that not only does the Navigator detect offsets associated with GPS fixes and back-correct the drift, it tacks the mosaic to the location of the trusted GPS fix, so that any future manual adjustment of other swaths has to take this absolute fix location into account.

Whether defined by the user, or automatically, the Anchor Point defines the error in our estimated position at time *t* of the observation. We can use this information to re-position the sensor at time *t*. Assuming that the error in position builds up with time, and that the position of the sensor is continuous in time and does not contain jumps, CleanSweep3 will propagate this error to re-position to the data before time *t*. If the Anchor Point is a GPS fix, the data at and after the fix is at the correct position (until further drift accumulates) so the error is only propagated backwards. If the Anchor Point was created by the user matching a known feature (navigation marker) to its image in the sonar data, the error is propagated both forward and backward. This global propagation of error is the basis of the new InterNAV2 algorithm.

In addition, during a survey one swath of data may overlap with adjacent swaths. In that case, a feature seen in one swath may be seen in adjacent, overlapping swaths. Due to positioning errors, the different observations of the same feature may not land at the same point on the mosaic of the swaths. If we don't know the true position of such a feature, we can not create an *Anchor Point*. Instead, we create a *Feature Point* for each observation of the feature, and we link them together. This creates a new constraint, which explicitly says that all these observations must land at the same position. This implies that the position of the sensor, and hence the vessel to which it is attached, must be modified to accommodate for all these constraints.

This gives us a network of nodes. Each node is called an InterNAV Point (either Anchor or Feature). Each link is an approximate, relative distance between nodes. In the network, the Anchor Points are fixed. The Feature Points may move. The links between the nodes have their direction and length approximately known. This gives us a set of absolute and relative constraints. We can use this network to find the optimal position of the feature points that respects the given absolute and relative constraints.

Simultaneous Localization and Mapping (SLAM) Algorithm

SLAM is a field of study that is mostly related to robotics. It has also been studied to a certain extent in oceanography to solve problems similar to the one at hand. CleanSweep3 implements a derived version of a SLAM algorithm to optimize the navigation based on the constraints given by InterNAV2. The solution is based on the paper "Fast, On-Line Learning of Globally Consistent Maps", by Tom Duckett, Stephen Marsland and Jonathan Shapiro, Autonomous Robots 2002. Starting from a topologically connected set of places, the algorithm assigns a location to each place that is consistent with the constraints given as input.

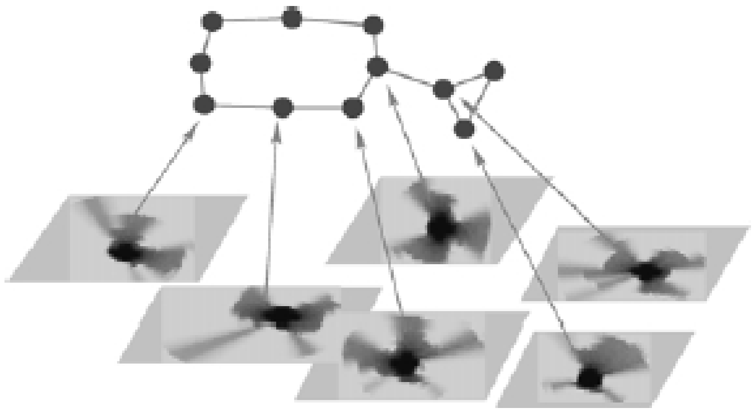


Figure 22. A network of topologically connected places.

In the final solution, the links between the nodes may have been extended or compressed. The energy associated with each link is proportional to this deformation. The goal of the algorithm is to minimize the global energy of the network, while respecting the given constraints. The navigation is modified so that the Anchor Points do not move, and the Feature Points associated with one feature are all mosaicked at the same geographic location. The algorithm also ensures the minimum energy for the network. That is, it ensures that the deformation of the original navigation is minimal.

The algorithm is detailed in Figure 23. In the algorithm, the variance for the link from node i to node j is proportional to the distance in time that separates the observations of the two nodes.

1. For each node i, do:
   a) For each of the neighbors j of node i, i.e., the places that are topologically connected to i, obtain an estimate r'_{ji} of the coordinates of node i using

   $$r'_{ji} = r_j + D_{ji}, \quad (28)$$

   where $r_j = (x_j, y_j)^T$ refers to the coordinates of node j, and $D_{ji} = d_{ji}\binom{\cos\theta_{ji}}{\sin\theta_{ji}}$.

   b) Combine the position estimates r'_{ji} for all j to produce new coordinates r'_i for node i using

   $$r'_i = \left(\sum_j{}' v_{ji}^{-1}\right)^{-1} \sum_j{}' \left(v_{ji}^{-1} r'_{ji}\right), \quad (29)$$

   where $\sum'_j$ refers to the sum over the neighbours of node i, and v_{ji} is the variance for the link from node i to node j.
2. Repeat from step 1 untill the change in energy falls below some pre-defined threshold, or some other stopping criterion is reached.

Figure 23. SLAM Algorithm

Demonstration of the InterNAV Utility

The combination of the Navigator, ReNAV and the SLAM-based InterNAV2 algorithm gives CleanSweep3 users a unique tool with which to rapidly and easily achieve significant improvements in the navigation and geo-coding of seafloor mapping data from any platform, surface or submerged. We demonstrate below an application of the InterNAV tool.

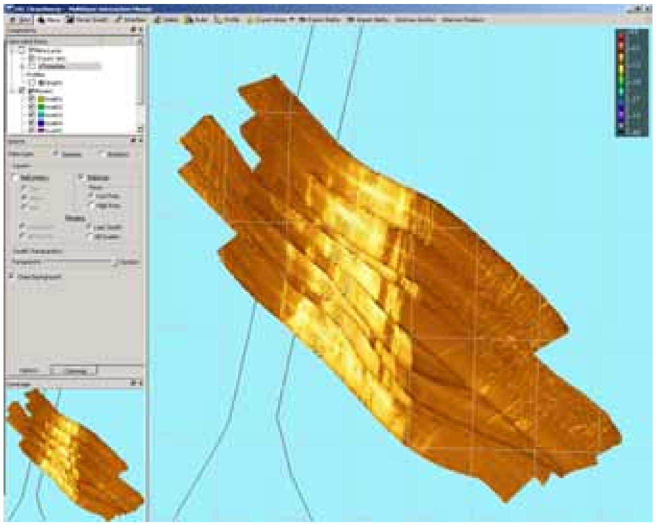


Figure 24. Raw Deepscan 360 sonar imagery from Loch Ryan, Scotland.

The Deepscan 360 is a high-frequency (360 KHz) interferometric sidescan and swath bathymetry sonar. It produces backscatter and angle of arrival information at 60 kHz sample rate and ranges up to 150 meters. It is typical of the types of sonar systems used by both surface and submerged platforms. The data shown in this example (Figure 24) were gathered with the system pole-mounted on a surface craft, in Loch Ryan, Scotland. Due to operator error, the data were collected with a large and variable latency (approximately 11 seconds). As a result, a preliminary mosaic of the data, shown in Figure 24 to the right, exhibits non-trivial track to track offsets. Due to errors in selecting the datum for the navigation system, there is also an absolute navigational offset. The light colored sand region should be centered on the demarked navigation channel.

Figures 25 and 26 below show detail views of just two swaths from the center section of the data shown in Figure 24. To resolve the navigational offset between the two swaths, we created a "feature point" matching the depression we see at the tip of the pink arrow in swath 2, Figure 25, to the same feature seen in Swath 1. Re-mosaicking with this single "feature-point" constraint produces the image seen in Figure 26.

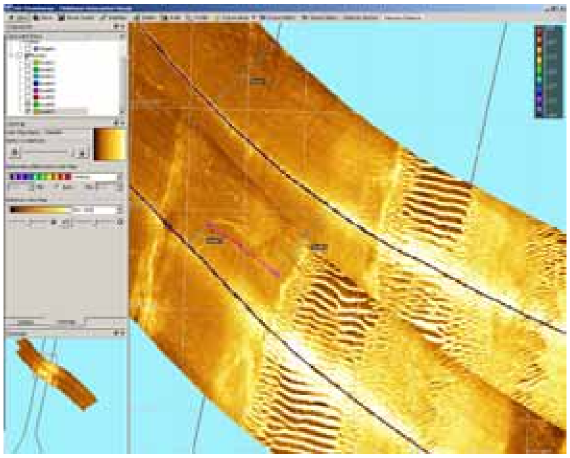


Figure 25. Zoom of two original swaths from Figure 24.

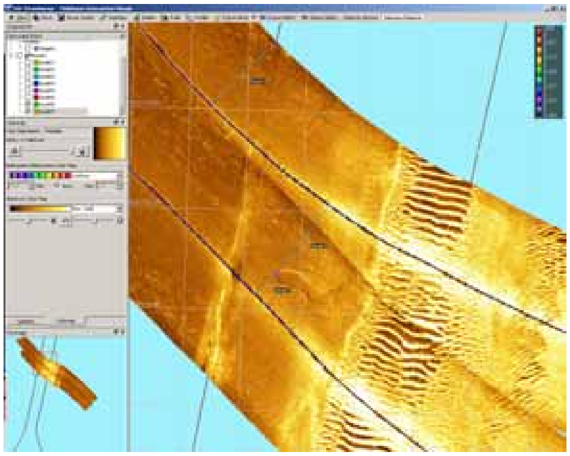


Figure 26. The same swaths after InterNAV2.

After replicating this for subsequent swaths, we produce the mosaic seen to the right, in Figure 27. The relative navigational errors have been reduced, with significant improvement in track to track alignment of features. Note that in this case we had to adjust all tracks. Had the data been collected through the turns, so that it formed a continuous track, we would have had to adjust only one pair of tracks to have all tracks inherit the offset.

To resolve the absolute navigational offset due to the Datum error, we can create a single anchor point, moving the center of the sand channel to align with its charted position. All relative feature-point matches are retained, but they inherit the absolute offset of the anchor point (Figure 28).

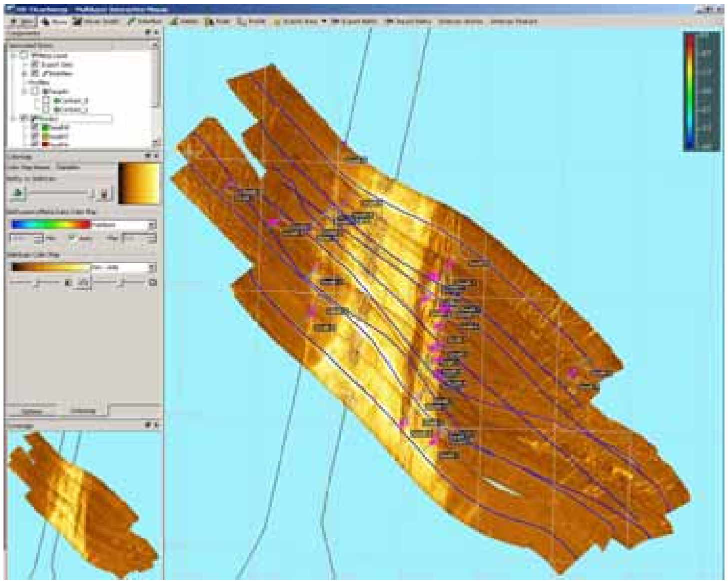


Figure 27. Mosaic of Deepscan data, showing Swath tag at Feature points

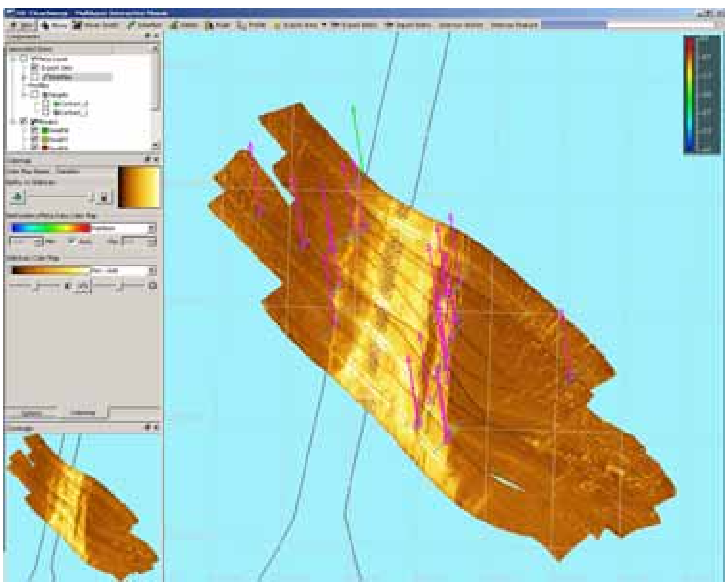


Figure 28. Mosaic ready to be adjusted by dragging the "anchor point".

Automated Feature Matching

The InterNAV2 technique described above is still hostage to the user's detecting and defining matching features in overlapping swaths. While the ReNAV and SLAM-based InterNAV do save the user a remarkable amount of time, the user may still have to spend a good bit of time manually determining the best points at which to match features in the overlapping swaths. To aid this, we have developed a prototype "feature-matching" tool, which, when given two images of the same feature on the bottom, will determine how they best fit together. The tool then reports the offsets to the InterNAV2 algorithm as a "feature-match" at the location of the targets given, just as if the user had created it manually, as was done above in Figures 25 and 26.

The feature-matching tool works best with isolated targets, as seen below in Figure 29. The four images show a single isolated target on the bottom, seen from four different passes. In the last pass, we are looking at the target from the opposite direction. As can be seen from this example, the nature both of the target, and the shadow it casts changes with look direction and aspect angle. An operator, whether human or algorithmic, attempting to deal with this target must take this aspect-angle variability into account in determining the best match point.

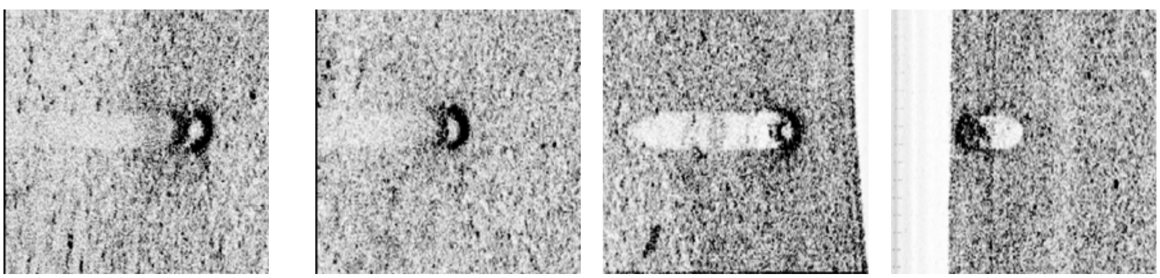


Figure 29. Sonar images of the same target on the seabed seen from different aspects.

While the above images all look reasonably similar, automated attempts to match them would fail due to both noise, and the variability of shadow and target appearance with aspect angle. We developed a strategy to defeat this by noise cleaning, classification, erosion and correlation.

Noise Cleaning

Figure 30 shows a synthetic target created for the purpose of developing and testing the

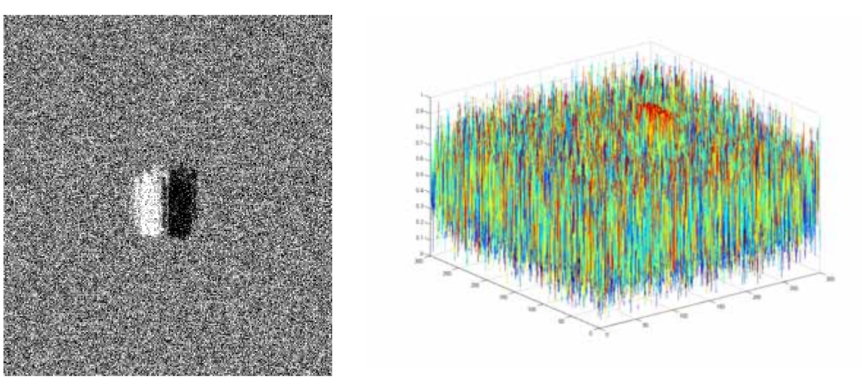


Figure 30. Synthetic sidescan image of target in grey-scale and 3-D views.

Feature matching algorithms. We used synthetic images in the development so we could test the strength of the filtering and classification routines under known signal to noise ratios. Following the development section, we show examples with actual sonar data.

A 5x5-averaging convolution is passed over the data to attempt to smooth noise spikes and reduce noise (Figure 31). If the signal to noise ratio of the image is very low (low signal, high noise), then this will be substantially less successful, but it still tends to improve feature extraction and matching. Convolution filters provided by fspecial() in MatLab, such as average, Gaussian, and pillbox filters are available, or a completely custom convolution kernel may be used.

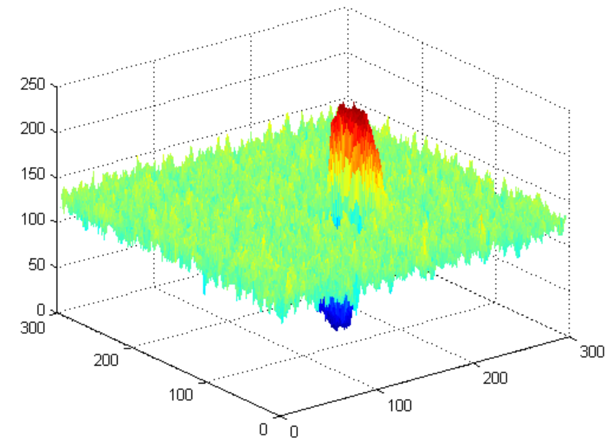


Figure 31. 3-D view of the same data after filtering

C-means Clustering / Thresholding

After noise cleaning, a fuzzy C-Means clustering algorithm is run on each image. For each point in the image we assign a weight of 'belonging' to 3 possible clusters, corresponding to target (high amplitude), shadow (low amplitude) and background (in between). The algorithm utilizes fcm() as implemented in MatLab. We then select the highest intensity cluster and threshold the image at the cutoff between the high intensity cluster and the middle cluster. This extracted cluster (hopefully) represents the target(s), and probably some noise peaks as well. Therefore our target extraction method assumes that the target is of high intensity and is higher than the background. The thresholding routine converts the classified image into a binary image, with the targets white (1) and everything else black (0).

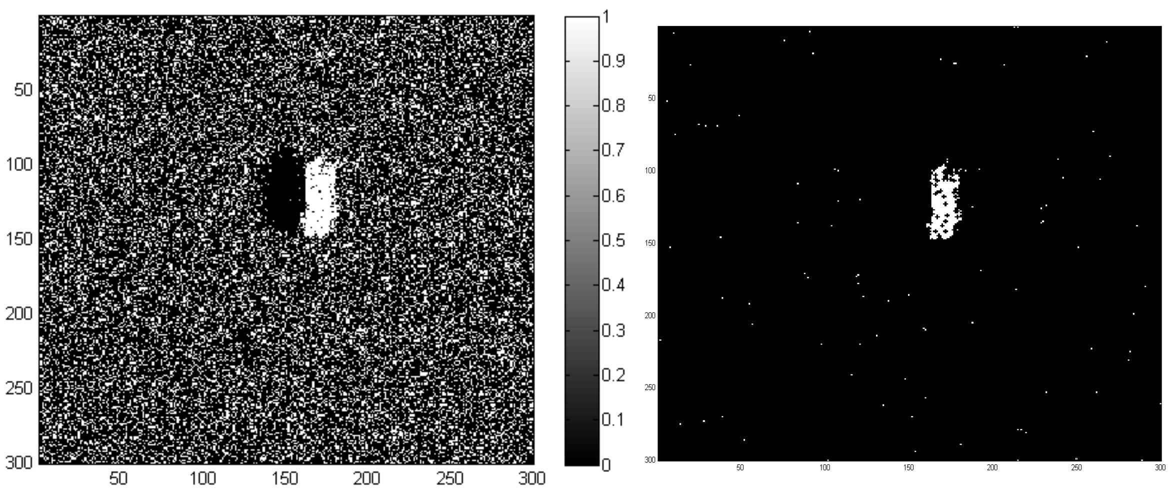


Figure 32. Classified image (left) and binary thresholded image (right).

Correlation Output

Once thresholding is completed, the target grids are sent to a cross-correlation algorithm. Cross-correlation is a statistical method of measuring the similarity of two signals – in this case a pair of 2-dimensional matrices (grids) – which are then 'slid' over each other. We utilized normxcorr2() as implemented in MatLab. While not currently implemented, support for multiple solutions could be added, likely giving the user a choice via a GUI. Multiple peak solutions may occur if multiple similar targets exist in a scene or if substantial noise bypassed the de-noising algorithms

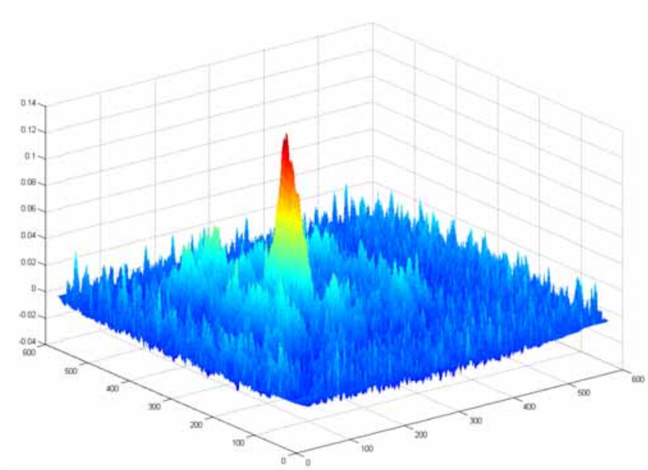


Figure 33. 3-D view of results of cross-corelating two binary images.

Once we have computed the correlation matrix output, we may determine its peak, and then convert this index location to pixel coordinates in the original image, and then to real-world geo-referenced coordinates (Northings & Eastings). In the case of synthetic data, artificial pixel sizes and X/Y positions are used. Using this offset information, we may now co-register this simple point target.

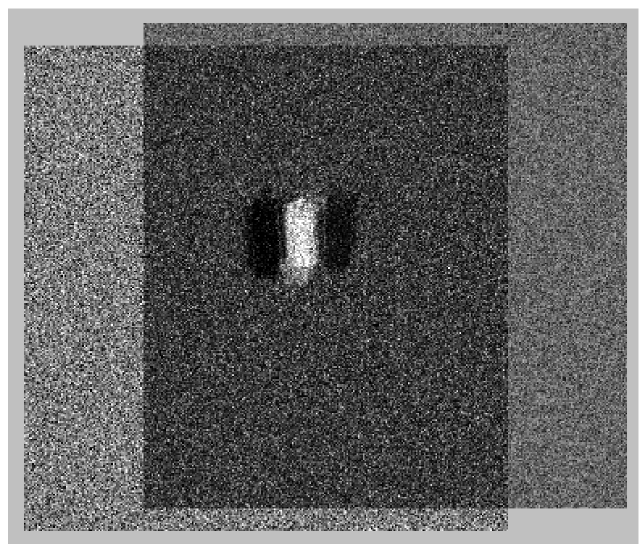


Figure 34. Montage of two opposing look images, automatically co-registered.

This correlation method was chosen as it is an established method for performing image comparison and has many existing implementations in various image processing software routine packages. Another benefit of the cross-correlation coefficient method is that since it is based on raw image intensities, future enhancements or changes to the convolution filters utilized in the feature extraction section of the algorithm could be more easily accommodated. In other words feature extraction and correlation are not 'tightly coupled'.

Below we include some examples using real sonar data. In each case, a single image is shown on the left, followed by a montage image of the first image and a second image from an opposite look direction. In each case, the algorithm has determined the proper offset between the two images so that the target object is best aligned. We allow the bottom image in each case to “shine-through” the top image, to show that the targets are in fact aligned. In actual mosaicking, only one layer would be shown, to minimize confusion caused by shadows going to either side of the target. At this time, this technique does not replace the detection of features, but may serve as an aid to ease the burden of determining their optimum co-registration.

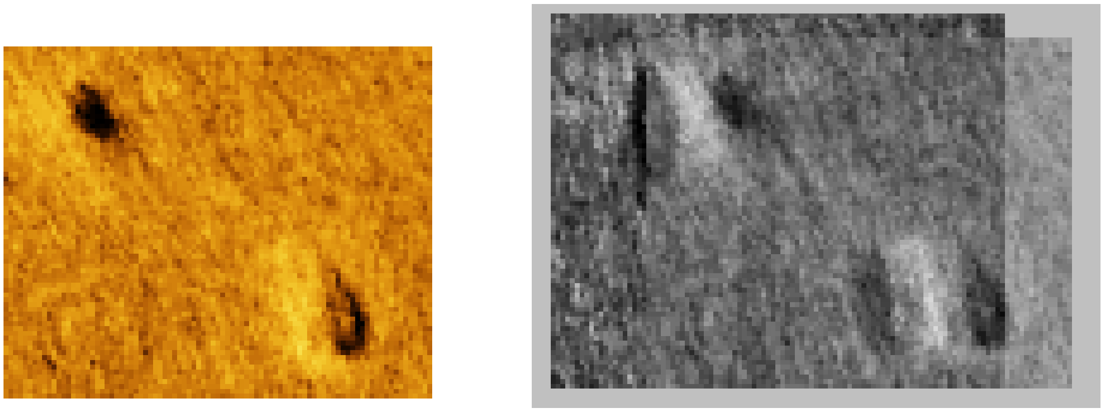


Figure 35. Complex image containing two isolated point targets.

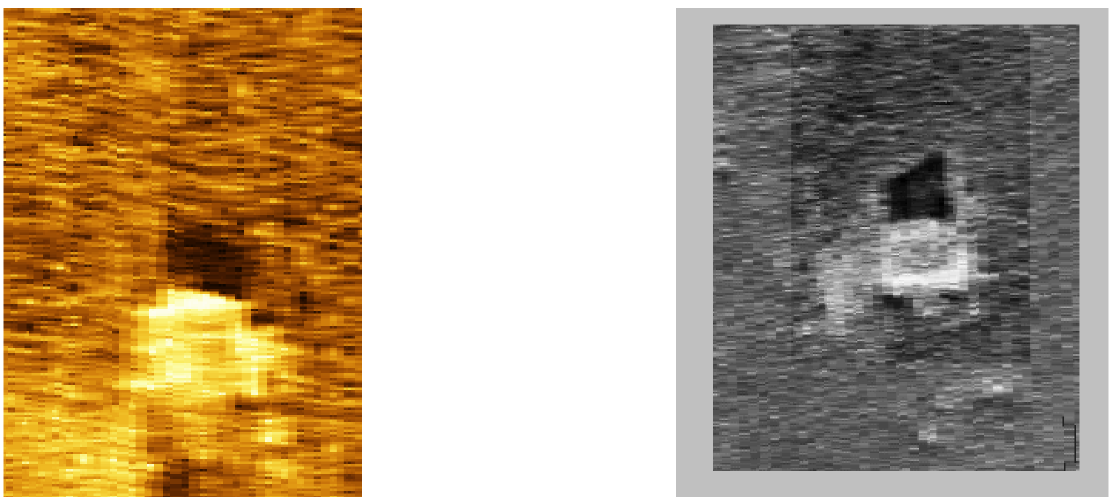


Figure 36. "Box-like" target seen in the same look direction on two reciprocal passes.

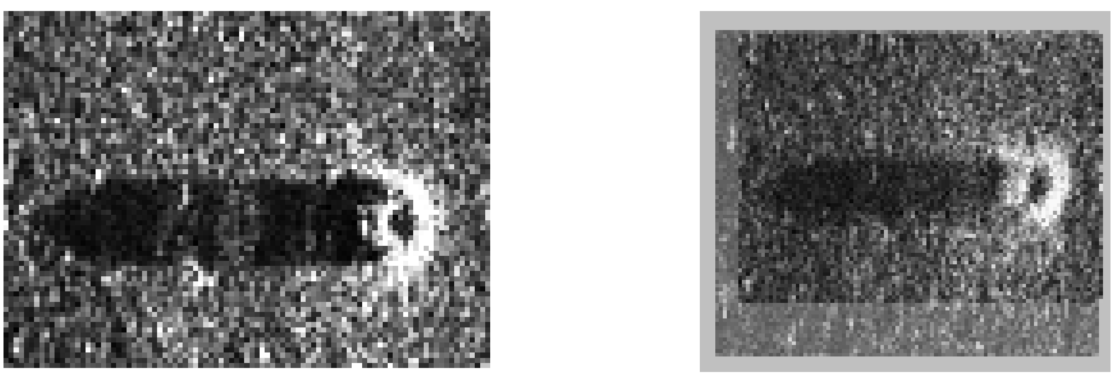


Figure 37. Our original "mine-like" target seen from two parallel passes.

Pre-Processing, AutoSwath & Batch Mosaicking

In CleanSweep2 and its UNIX predecessor OICSwath, the user had to manually select and load the data for each line, defining both the span of data in time, and the swath coverage in the mosaic. The user also had to define processing for each line, and mosaic the lines individually. While this allowed tight control over processing quality, it became repetitious, and consumed a significant amount of time in simple mechanical activities. In CleanSweep3, several new features obviate much of the manual work, allowing the user to much more rapidly see the results of their processing.

A new "AutoSwath" feature automatically divides the tracks up into swaths, defining their limits based on rate of turn, and displays the result to the user, as seen below in Figure 38. As the data gathered during turns are generally distorted due to platform motion, and are also outside the survey area of interest, this allows the operator to focus on processing the sonar data within the survey area, while still retaining all navigation data as continuous time series. In trials, customers found this an enormous time saver, as they heretofore had been required to create each swath manually.

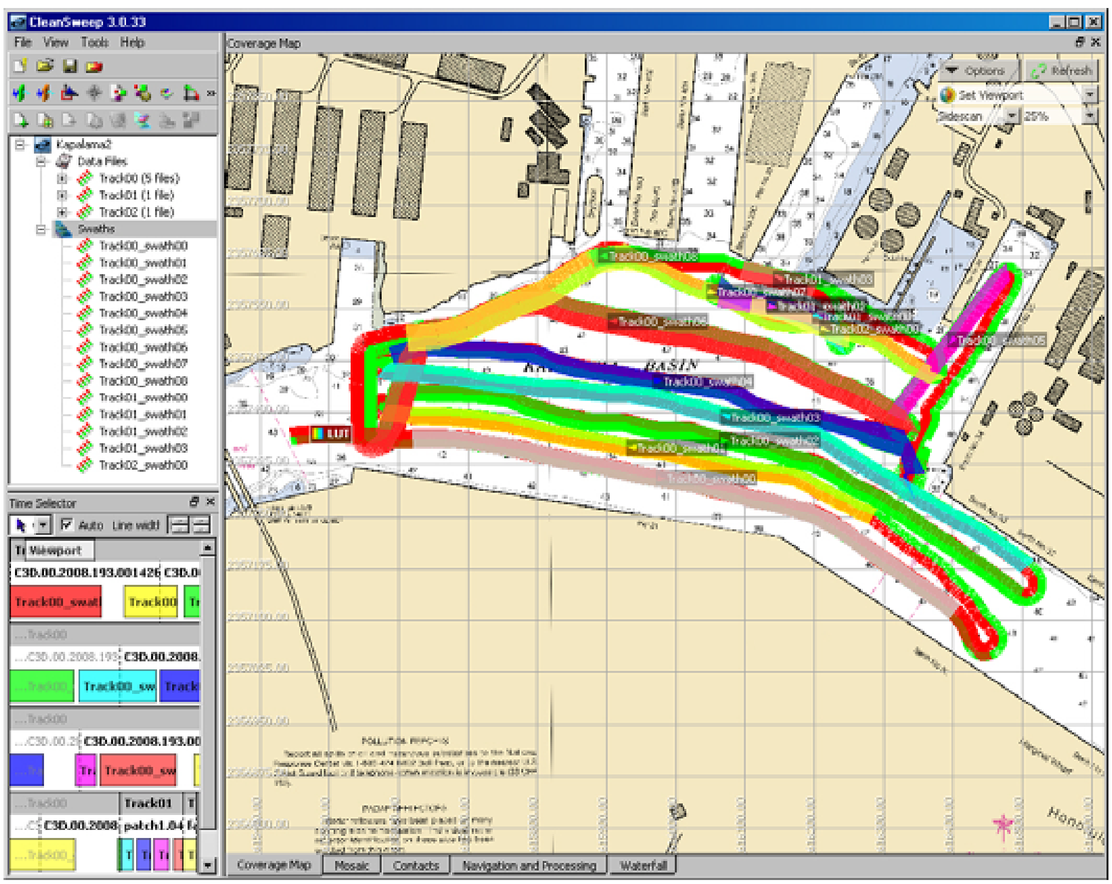


Figure 38. CleanSweep3, showing automatic division of "Tracks" into "Swaths".

Default Processing

To further ease operator burden, CleanSweep3 also offers new default processing of both sidescan and bathymetry, as well as batch mosaicking. The chief benefit of these tools is that an operator can identify a data source, load the data, see automatically derived default processing applied to the raw data to produce a first cut at processing, and create a draft mosaic, all without any effort at editing, creating swaths or defining mosaic bounds. All these operations are data driven, and automatic. After viewing the draft mosaic, the user can decide either that the default settings were adequate, and proceed to creating final output, or that some or all tracks need local tuning. This has the effect of now offering near-real-time production of final products and interpretable data from seafloor mapping data which previously was send back to a post-processing lab. This provides a more tactical solution, allowing commanders in the field the opportunity to both collect and evaluate the data while they are on site, with minimal effort.

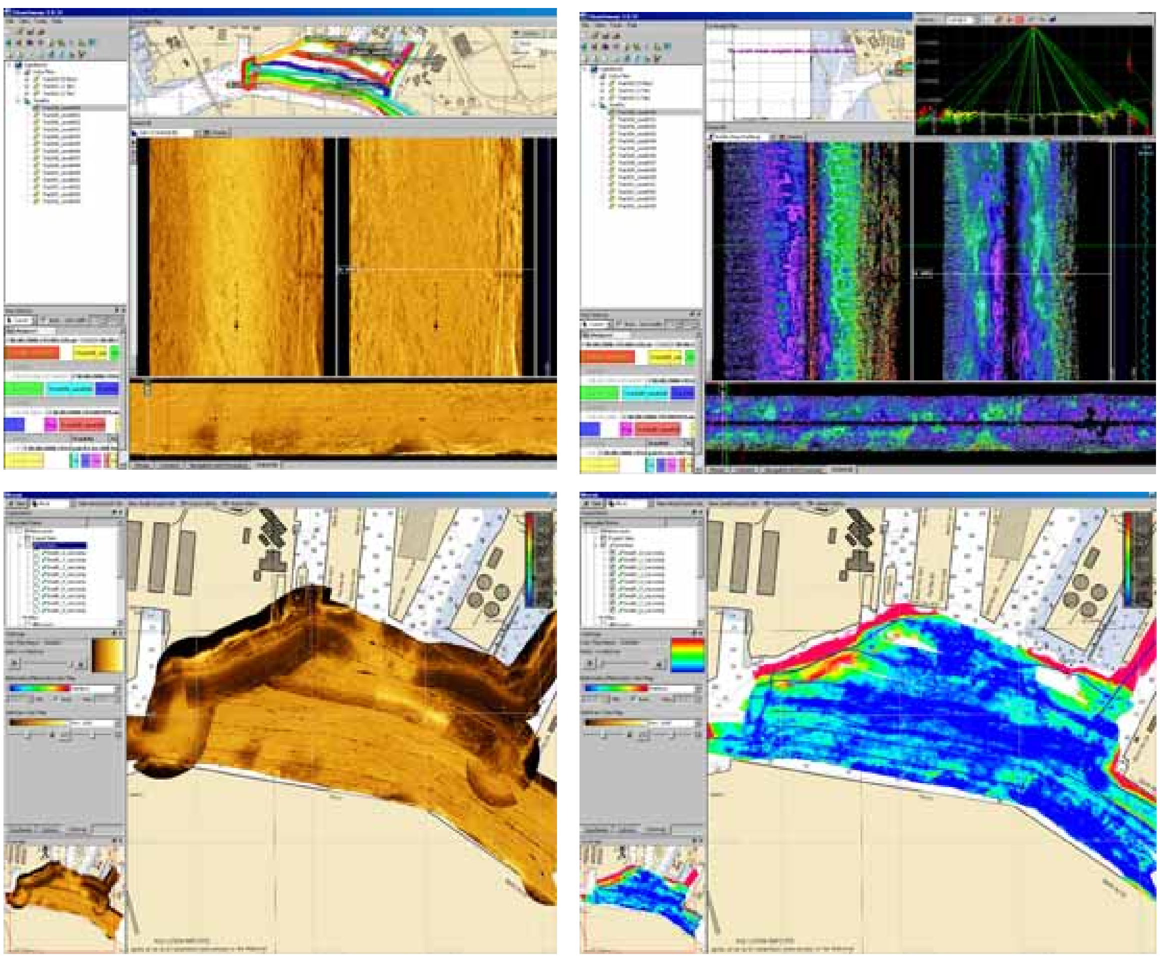


Figure 39. CleanSweep3 showing draft processing and mosaicking results for sidescan and bathymetry. Draft mosaics of 2 hours worth of data were created at 0.5 meter resolution in 5 minutes.

New Bathymetry Processing Tools

Display

CleanSweep3 implements all of the previous tools for bathymetry processing, and delivers a number of new tools as well. The standard view of the bathymetry processing interface is shown in Figure 36 below, with raw data as delivered by the sonar on the left, and processed data on the right.

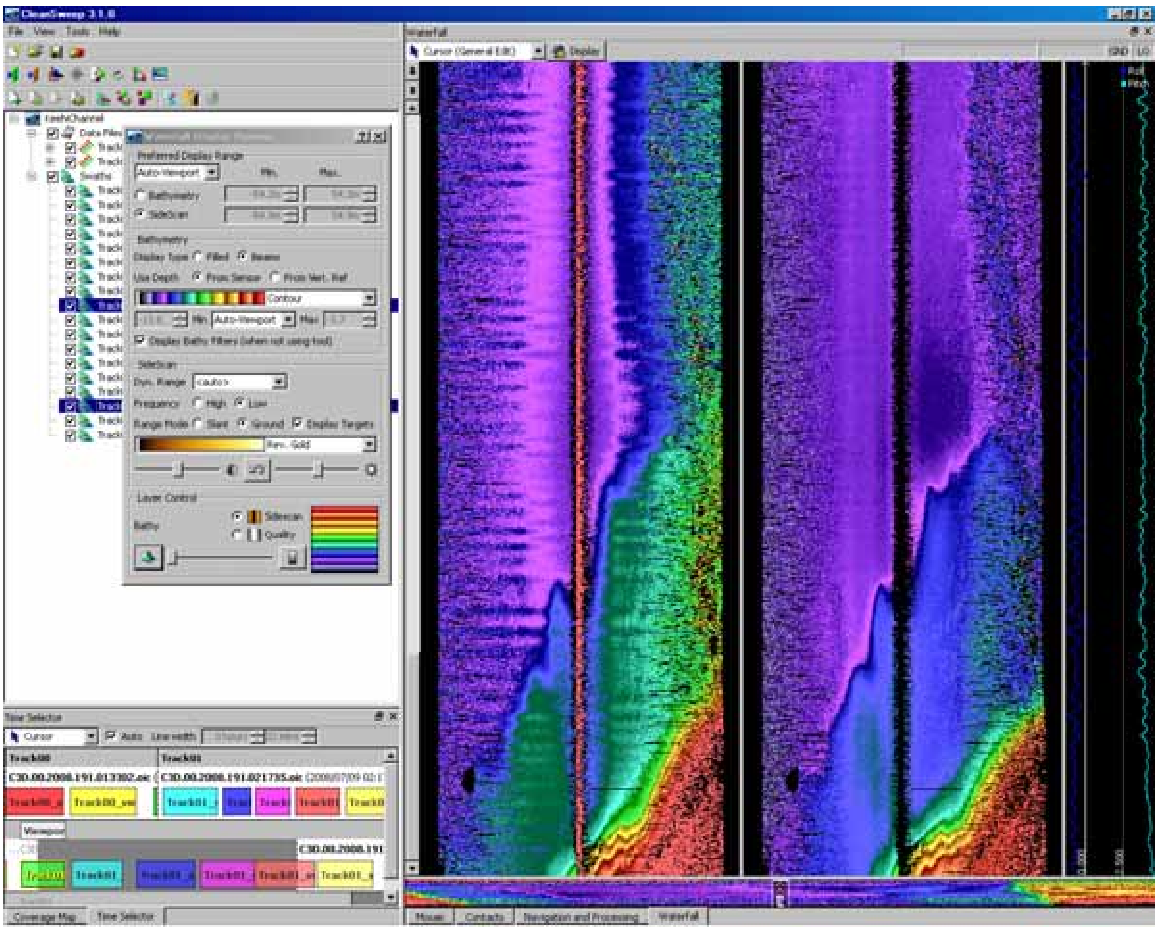


Figure 40. The CleanSweep3 bathymetry processing interface.

New display features include auto-scaling of the colormap to the range of data for the line, the viewport or just the data in the window, an improved choice of color palettes, and the ability to interactively blend the sidescan in with the bathymetry. The standard bathymetry processing, including application of heave, pitch and roll corrections, is now automatic, using the data previously treated in the Navigator. As the user declares the nature of the survey (surface ship, submerged platform or deeptow) on creation of the processing project, CleanSweep3 uses the correct components by default to process the data, again automating the effort and unburdening the operator.

Editing

CleanSweep3 offers both automatic and manual tools for rejection of outliers. Automatic filters include limits on depth, slant-range, ground-range and angle, as well as absolute deviation from a best fit line. Manual editing can now be performed both in a 2-D "ping-profile" editor, as seen in Figure 41, or in our new Swath3D interactive "point-cloud" editor, as seen in Figure 42.

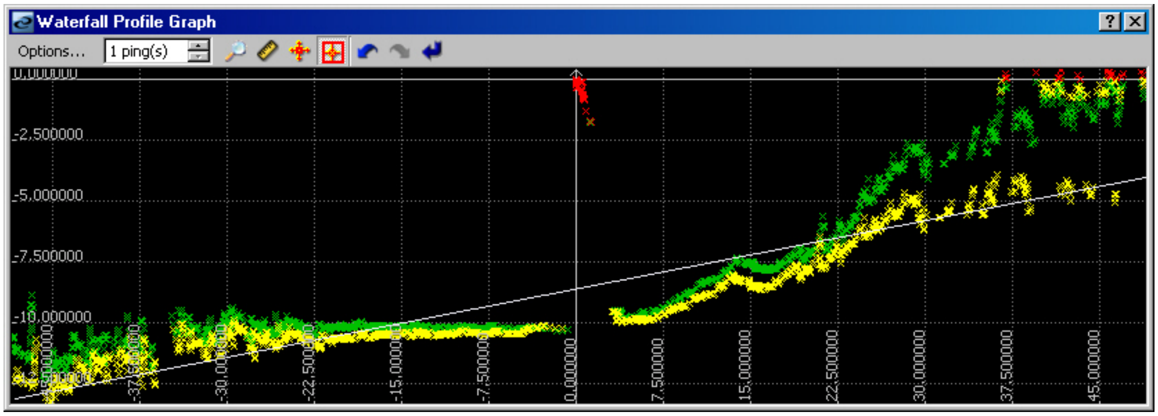


Figure 41. CleanSweep3 Ping Profile tool for 2-D bathymetry editing.

The new Ping Profile tool shows both raw and processed data, as well as indicating in red the points which have been deleted by your filtering or editing operations. The user can load up to 100 pings at a time in the profile editor, to greatly facilitate editing of noisy data.

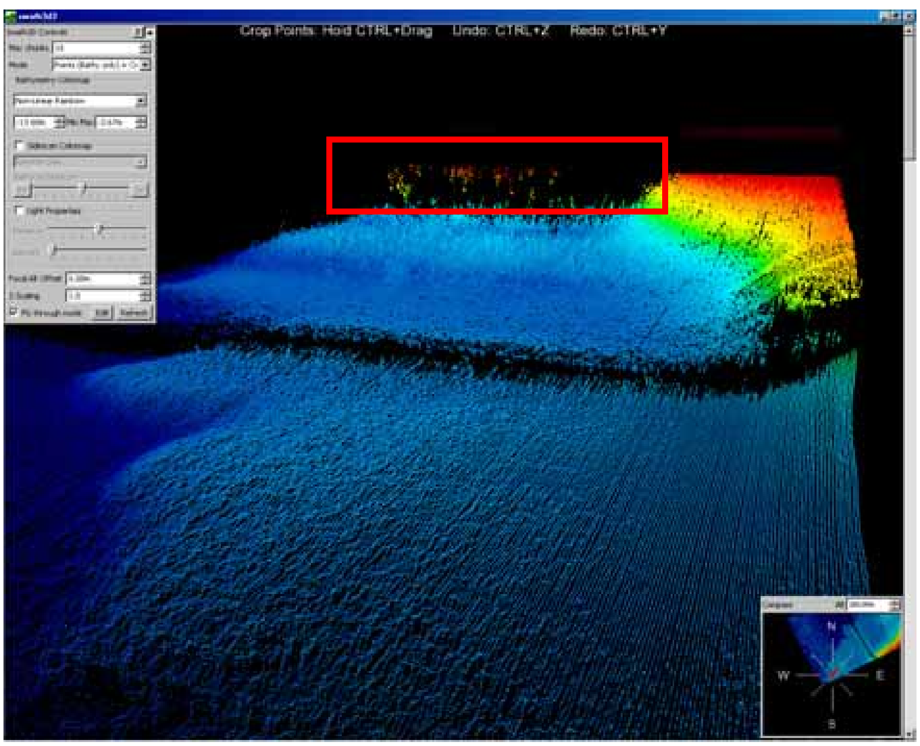


Figure 42. CleanSweep3's new Swath3D bathymetry editing tool.

For further ease in manual data editing CleanSweep3 now offers Swath3D, our 3-D data visualization and editing tool, as seen in Figure 42. Swath3D allows a three dimensional view of the data in the current swath, supporting mesh, sidescan drape and point-cloud views. In Point-cloud mode, the user can drag a box around a cloud of data to delete points from the volume.

Roll From Slope

While filters and editing tools are useful for cleaning the noisy data from the good, some data sets suffer more from distortions to the good data which no amount of filtering or editing could fix. As an example, if the roll data for a survey is either missing, noisy or in some non-linear fashion not representative of the actual platform motion, then straightforward application of this corrupted roll data to the raw sonar data will fail to compensate properly for vehicle motion, resulting in possibly further corrupted bathymetry. One might assume that with proper control of survey quality data sets with corrupted or missing attitude data would be rare, missions can not always dictate the sea state in which they survey, and quite often vehicle motion exceeds the linear response rate of the available sensors. For some survey groups, fully a third of their data some years have been collected in conditions which exceed the capabilities of the sensor. A utility to properly correct data in the absence of good roll data would be quite useful. In response to this need, we have developed such a tool for CleanSweep3. We refer to this as Roll-From-Slope.

The problem is to process raw sonar data (range and angle measurements) without relying on the roll data. Clearly, if the platform is motionless, this is trivial. If, on the other hand, the platform was rolling while collecting this raw data, the raw data will reflect both the slope and curvature of the seabed, and the platform motion. Consider a flat seabed being surveyed from a platform which rolls back and forth with a period of 4 seconds and an amplitude of 5 degrees. In the absence of roll data, processing the raw data would produce a seabed which alternately sloped five degrees left, then five degrees right, much as seen in Figure 43 below. If the bottom were truly flat this problem would be trivial to solve. As luck would have it, we rarely get to survey flat bottoms, so to solve for roll one must continuously estimate the real topographic slope, and extract the component of apparent slope attributable to vehicle motion. The Roll-From-Slope tool does just this.

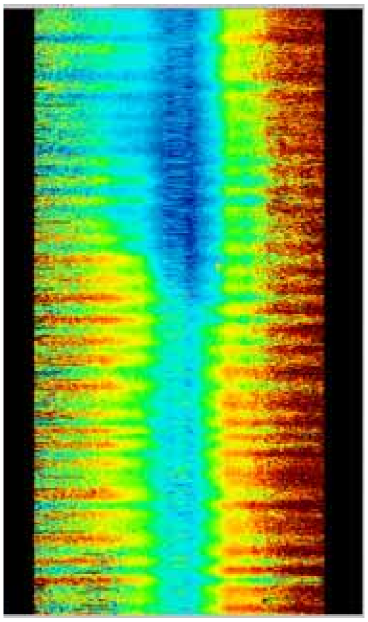


Figure 43. Raw Data, without roll correction.

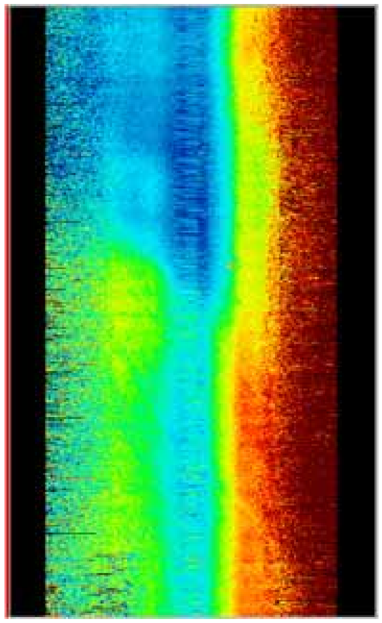


Figure 44. Processed with the actual roll data.

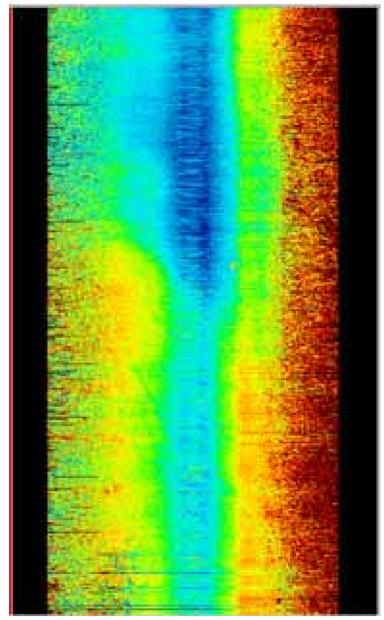


Figure 45. Processed with "roll-from-slope" algorithm.

Figure 44 shows the raw data from Figure 43 processed with the actual roll data. Figure 45 shows the raw data with the actual roll data deleted, and then processed using the new "Roll-From-Slope" algorithm. Clearly, it is better to use good roll data when it is available. However, if roll data are not available, this technique may allow operators to salvage otherwise lost survey data.

Beam Angle Correction

Bathymetric systems do not measure depth. Bathymetric systems measure angle and range. Even a fathometer or "single-beam" echo-sounder measures the echo travel-time (an estimate of range) and assumes the angle to be zero, i.e. straight down. From angle and range, one calculates, after correction for attitude, heave and other offsets, the distance athwartships and depth. In general, it is easier to measure travel time, and hence range, than angle. Furthermore, if the transducer mount angle is poorly known, or if the system does not perfectly measure the arrival angle, or if the sound velocity profile is unknown and the echo arrives after significant refraction, the simple reduction of range and angle to distance athwartships and depth will produce erroneous results.

Traditionally, one conducts a "patch-test" maneuver to determine mount-angle offsets, and one acquires sound velocity data to correct for refraction. Similarly, one hopes that there is no bias in the bathymetric sensor resulting in less than perfectly reporting angles. If, however, mount angles cannot be properly measured, or sound velocity measurements cannot be taken, then the resulting bathymetry data will suffer, usually leading to track-to-track miss-match, such as seen in Figure 46.

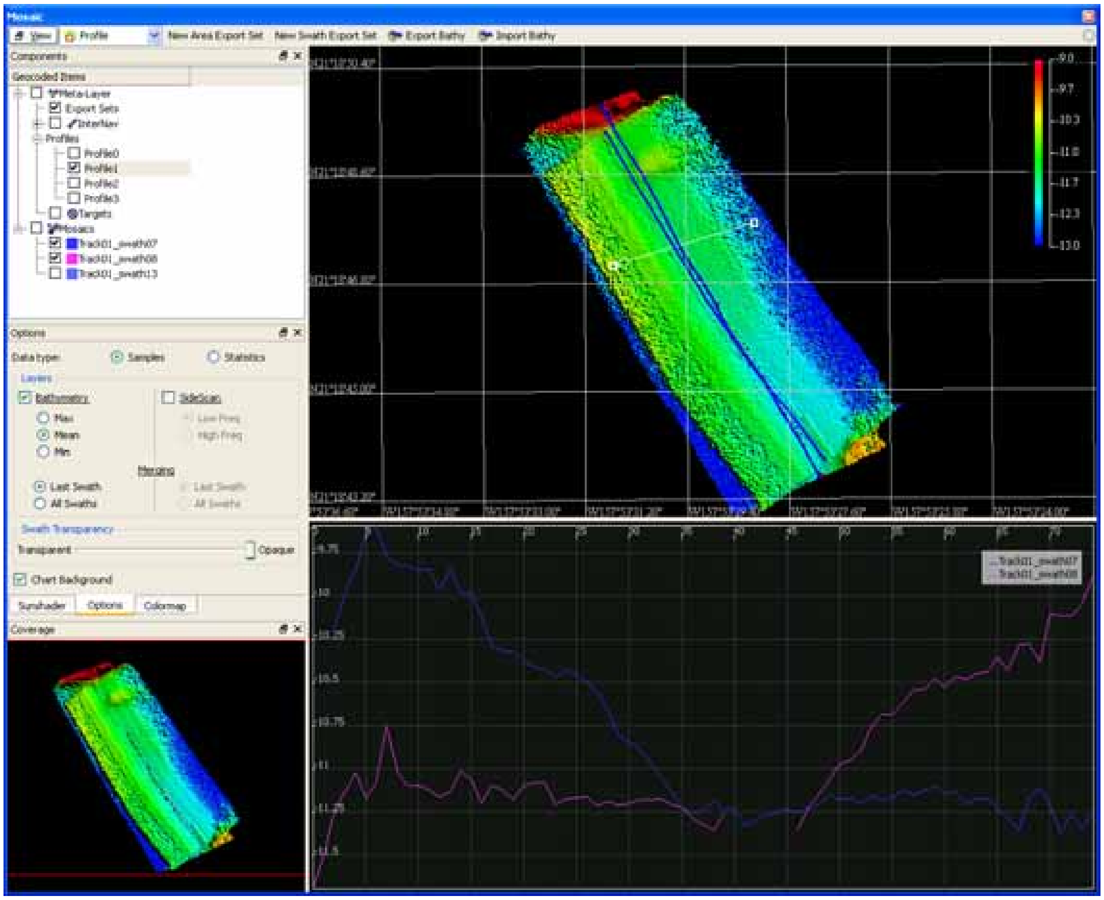


Figure 46. In this example, parallel overlapping tracks were run in opposite directions over flat seafloor. While there is good agreement on the port side, the starboard channel suffers from an improperly measured mount-angle.

In Figure 46 (above), two tracks were run parallel to each other in opposite directions over a flat seafloor. The profile at the bottom of the screen-capture shows that for both channels, one of the channels captured the flat seafloor correctly, while the other is offset by an angle due to an incorrectly measured mount angle.

In CleanSweep3, a new calibration tool, the Beam Angle Correction tool, uses the nadir data from one trackline, and compares it to a swath/ping of overlapping data run at a perpendicular orientation (Figure 47). Assuming that the nadir data provides an accurate measure of depth, it is possible to measure the beam-by-beam error for the cross-track/tie-line.

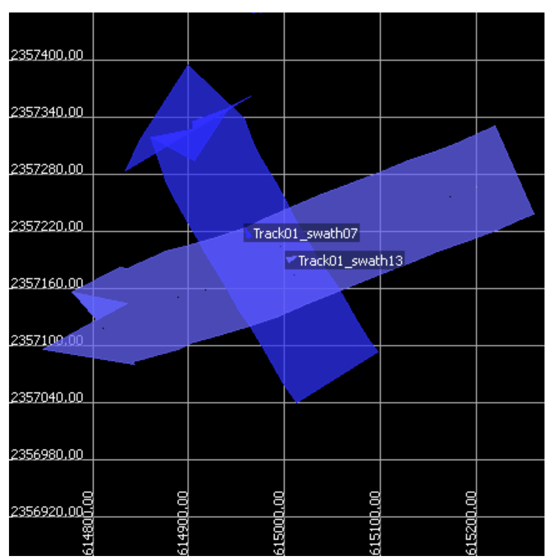


Figure 47. Example of trackline orientation that allow for the calculation of beam angle corrections.

These corrections can then be applied back to the data in order to make angle corrections on a beam-by-beam basis (Figure 48). The application of these corrections should result in accurate beam angle adjustments, thus accurate depths.

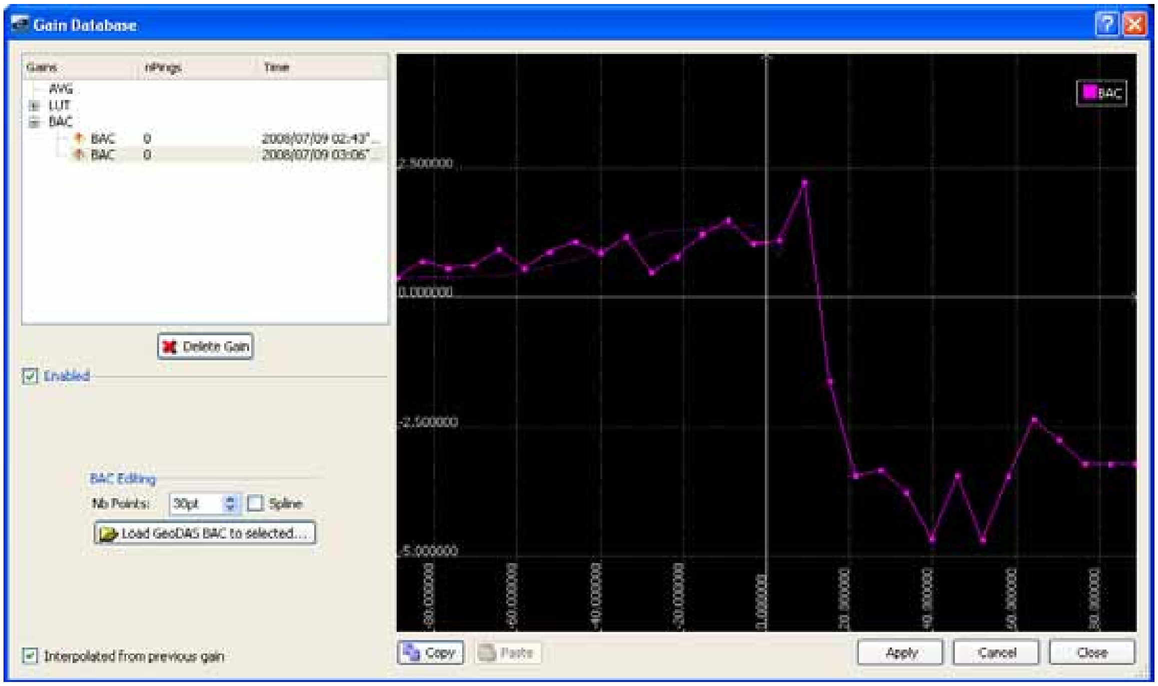


Figure 48. Using the data from perpendicularly overlapping swaths, CleanSweep3 can calculate beam-by-beam angle corrections.

This correction effectively solves the problem of both unknown or incorrect mount angles and unknown sound velocity profiles in one step. The value of this tool is its contribution to the production of high-quality data in situations where data acquisition procedures would otherwise not allow.

Roll Patch Test

With the release of CleanSweep3, OIC has also implemented a Roll Patch Test tool allowing for accurate calibration of roll bias within a dataset. The Roll Patch Test requires a set of parallel track lines run in opposite direction over flat seafloor.

Figure 49 (below) shows an example of parallel track lines with a roll bias apparent in the swath profile comparison. Using the new Roll Patch Test tool, users can accurately measure the bias.

The Roll Patch Test produces individual grids for the overlapping areas of each swath. The program will then automatically calculate the roll bias value (in degrees) that produces the best fit between the two surfaces (Figure 50). The resulting value is applied to the attitude data for the project, thus ensuring data accuracy by properly correcting for roll bias.

Figure 49. Example of parallel track lines with a roll bias apparent in the swath profile comparison.

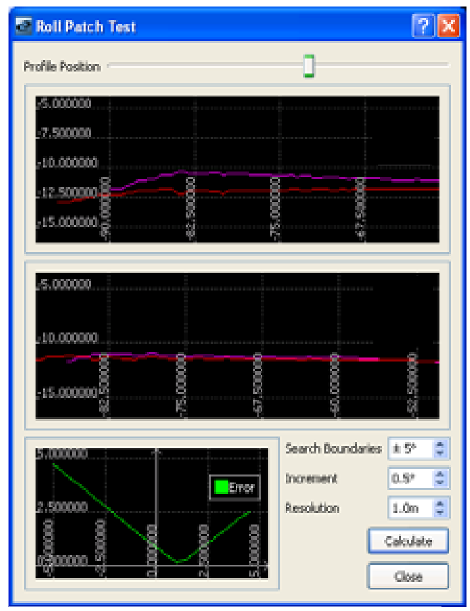


Figure 50. The Roll Patch Test provides a calibration tool for correcting systematic roll bias using CleanSweep3.

Discussion

This effort has produced and delivered a post-acquisition processing package which allows a significant improvement in NAVOCEANO's ability to process swath bathymetry and imagery data. It has met and exceeded the proposed metric of accelerating processing by a factor of 3, while retaining bathymetric accuracy as required by the proposal.

We have demonstrated the new CleanSweep3 software to the staff at NAVOCEANO, and to members of the Seaview Working Group. Response to the new CleanSweep3 has been enthusiastically positive. In particular, the NAVOCEANO operators were favorably impressed with the improvements offered by SmartSonar2. To quote Eddie Cranford, NAVOCEANO Special Projects:

"Bottom Line.

> This test of CleanSweep 3.0 revealed a greater than 3 to 1 reduction in processing time with an equivalent overall quality of processed data, when compared to present CleanSweep Version 2.5. The opinion of NPL analysts is that even greater time savings will be observed with normal mission data, barring unforeseen or unexpected issues with the larger data sets that were not observed with the relatively small NGI2 test data set. "

References:

T. Duckett, S. Marsland, and J. Shapiro, "Fast, on-line learning of globally consistent maps," Autonomous Robots, vol. 12, no. 3, pp. 287 -- 300, 2002.

Appendix A: Report From NAVOCEANO 10 July, 2008

Attached please find Eddie Cranford's updated report on CS 3.0 based on a dedicated comparison test with CS 2.5. Please feel free to forward to interested parties or let me know if you would like us to forward. If there are any questions, please feel free to contact me or Eddie.

Thanks.

Mike Sandler
Special Projects Cell, Code NPL
Naval Oceanographic Office
228-688-4560
UNCLAS: michael.sandler@navy.mil
SIPR: sandlerm@navo.navy.smil.mil

This report describes the comparison of CleanSweep 3.0 processed NGI2 data when compared with CleanSweep 2.5 processed data. Eight swaths of data containing ~ 3 Gb of raw data were processed with both versions of CleanSweep. The results for time comparisons for the different phases of processing are shown as well as statistical comparison of the two exported bathy grids.

I. Time comparisons:

	CS 2.5	CS 3.0	
A. Data load:	3 min	5 min	
B. Data setup:	25 min	25 min	
C. Processing:	3.0 hrs	30 min	(including internav & mosaicking)
Total:	3.5 hrs	1 hr	

** This time comparison was atypical in one important consideration. Normal processing with CS 2.5 requires the swaths to be reactivated, reprocessed and remosaiced in a repeated manner. These processing steps will increase the above ratio (3.5 to 1), to an amount depending on the specific data set issues and problems, i.e. a large dataset with minimal problems should experience a greater than 3.5 to 1 decrease in production time using Cleansweep 3.0.

II. Data comparison:

The exported bathy llz files were gridded at ~ 5m resolution for comparison purposes. The CS 3.0 compared 100% within +/- 2% of the CS 2.5 bathy grid.

III. Other issues & concerns:

A. There remains the question of loading larger data sets with the present version of 3.0.

We have experienced problems loading data sets larger than ~10Gb of raw data. We typically work with data sets consisting of 20Gb or more and have not experienced the same data load issues in CS 2.5.

B. The ping profiler internal crop tool did not work properly in CS 3.0 as in CS 2.5, although individual pings could be deleted.

C. The bottom tracking feature did not work properly in CS 3.0 as in CS 2.5.

D. Renav using anchor points and feature match was successful, however the renav feature using gps fixes had not yet been implemented in the CS 3.0 version that NPL has to work with.

IV. Summary & Conclusions:

This test of Cleansweep 3.0 revealed a greater than 3 to 1 reduction in processing time with an equivalent overall quality of processed data, when compared to present Cleansweep Version 2.5. The opinion of NPL analysts is that even greater time savings will be observed with normal mission data, barring unforeseen or unexpected issues with the larger data sets that were not observed with the relatively small NGI2 test data set.