

Sonar Explorer: A New Tool for Visualization of Fish Schools from 3D Sonar Data

Jean-Paul Balabanian¹ Ivan Viola¹ Egil Ona³ Ruben Patel³ and Eduard Gröller^{2,1}

¹Department of Informatics, University of Bergen, Norway

²Institute of Computer Graphics and Algorithms, Vienna University of Technology, Austria

³Institute of Marine Research, Norway

Abstract

We present a novel framework for analysis and visualization of fish schools in 3D sonar surveys. The 3D sonar technology is new and there have not been applications to visualize the data in 3D. We have created an application called Sonar Explorer that satisfies the requirements of domain scientists. Sonar Explorer provides easy and intuitive semi-automatic fish school tracking and survey map generation. The overall pipeline is described and all pipeline stages relevant for visualization are highlighted. We present techniques to deal with 3D sonar data specifics: highly anisotropic volume data aligned on a curvilinear grid. Domain scientists provide initial impressions on interaction and outlook.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation

1. Introduction

Accurate estimates of fish stocks are necessary for stock assessment and a sustainable fishery. With proper monitoring, the risk for over-fishing and potential recruitment failure is reduced. Modern assessment methods need data on present stock level and distribution, which calls for new *surveillance* technology. In a modern, new ecosystem approach, studies of fish behavior and relations between animals in the water column might be studied by exploiting new technology. Similarly, the fishing industry needs to fish their quota correctly, targeting exactly the size and school volumes they can cope with during the catch process, without harming unwanted species and size groups. New sea surveillance technology may therefore also help sustainable harvesting of the stocks. Advances in underwater acoustic methods using scanning sonar seems to be a promising alternative, where now sequences of 3D information is available for visualization and measurement at realistic ranges to cover entire fish schools. Since the systems are also delivering data with output, i.e. calibrated amplitude, they can deliver fairly accurate measures of biomass. The new multi-beam sonar MS70 is a horizontally observing sonar. It yields very high spatial resolution when operating all 500 beams, covering the frequency band 75-112 kHz [ABGL06]. The sonar has un-

dergone sea trials from research vessel "G. O. Sars" from December 2005 to December 2006, including detailed calibrations [ODK*06]. One of the most challenging tasks, how-

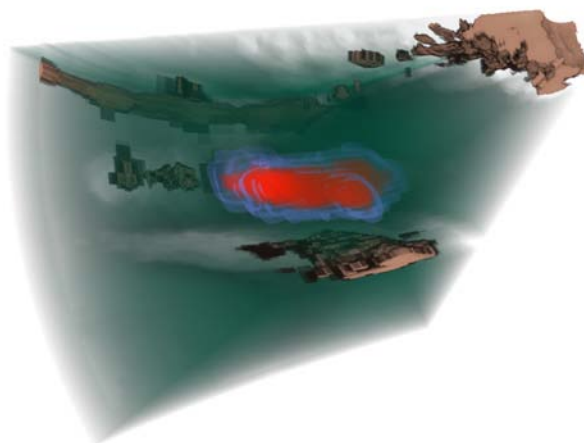


Figure 1: Volume rendering of 3D sonar data.

ever, is to capture, process and display the data collected by the 800-transducer elements in real time. In standard survey mode, the receiver boards may deliver data at a rate of 2GB per ping (i.e., a one time-step three-dimensional dataset), with a ping rate of 1Hz. Nevertheless, these rates would only be stored in short sequences for research purposes. Normally, these data are transmitted internally within the sonar system to six front-end computers for beamforming and data reduction. Typical data rates delivered to the operator station of the sonar to be stored are 1-2 GB/hour, depending on the range sampled and the sonar pulse repetition frequency (PRF).

At the operator station, two selectable cross sections through the water column can be presented in real time. One cross section is a horizontal slice from a *25-beam-fan* which shows data from the transducer face to a maximum range of 500 m. The corresponding vertical slice is from a *20-beam-fan* which covers 45 degrees from the surface downwards. Position data from DGPS and accurate motion sensors are delivered to the receiver unit for automatic compensation for vessel roll and heave. The digital range resolution delivered is dependent on the pulse duration used. It is typically 0.38m for the most often used pulse duration of 2ms. As the sonar scans the water column, the data matrix may be used to reconstruct entire fish schools for one ping in 3D. Alternatively if successive pings cover the same school a reconstruction over time in 4D is possible [ABGL06].

Efficient surveying, however, necessitates that derivatives of the data are analyzed and displayed in near-real time. As realized in the Sonar Explorer, several tasks are readily solved simultaneously, such as a true geographical representation of the vessel, and an effective observation of the detected schools.

The contribution of our paper is the introduction of a framework for visual analysis of fish schools in 3D fishery survey data. This framework satisfies specific needs of domain scientists and provides mechanisms for semi-automatic survey reporting. We provide background information on the data characteristics, present the visual analysis pipeline, and describe how existing visualization methods have to be altered in order to handle specific properties of 3D fishery survey data, i.e.,:

- curvilinear adaption of the GPU ray-caster
- addressing scheme for anisotropic volumes to overcome graphics hardware limitations
- tracking of fish schools in the temporal domain

The paper is organized as follows: Section 2 describes previous work related to our framework. Section 3 describes the individual steps of the pipeline that generates the 3D output. Section 4 highlights pipeline steps relevant to visual analysis and interaction and provides technical details of these steps. Section 5 describe the resulting survey map used for fish school distribution analysis. An outlook to future possibilities provided by domain scientists are discussed

in Section 6. Finally we draw conclusions and summarize the paper in Section 7.

2. Related work

Our work intends to assist fishery industry and marine research in estimating stocks for sustainable fishery [Gun93]. The estimation is based on measurements by different acoustic scanning devices [MS93]. Sonar Explorer performs visual analysis on 3D surveys obtained by MS70 where the measurements are taken over several hundred time-steps. Software provided by the hardware vendor [sim06] has very limited functionality allowing only slicing of the curvilinear dataset in two orthogonal directions. There have been only very few attempts to visualize features of 3D surveys until now. Applied methods have been 3D volume rendering and multi-planar reconstruction of time-varying data [ABGL06,ODK*06]. The results did not offer any interaction possibilities, as these precomputed animations were intended for presentation purposes. In contrast to these early results, the functionality of the Sonar Explorer is intended for visual analysis and feature (fish school) identification.

One time-step of the entire survey, denoted as a ping, is a volumetric dataset aligned on a curvilinear lattice. Direct volume rendering of 3D curvilinear grid data has been the subject of research for many years [Gar90, WCA*90]. Our curvilinear volume data corresponds to a *conical* cutout of a sphere, therefore the rendering is significantly simplified as opposed to handling general curvilinear grids. This property allows us to use a modified version of a standard GPU ray-caster [RGW*03, KW03] with addressing using spherical coordinates.

Our framework performs visual analysis of three-dimensional time-varying scalar data. Frameworks for visual analysis of data with similar characteristics such as SimVis [DGH03, DSH05], provide, up to some extent, similar functionality. The difference is driven by a special handling of the underlying data as they originate from different science domains.

Our aim was to present the results of the visual analysis as clear as possible. This aim was achieved by incorporating focus+context methods [Hau05] where the goal is the possibility to analyze individual pings and see the distribution of all features in a linked survey map.

3. Overall Processing Pipeline

The 3D sonar MS70 is the first device that allows three-dimensional screenings of sea resources over time. The overall goal is to study and precisely quantify fish resources. Also the three-dimensional distribution of fish schools in the sea according to properties such as sea temperature or season is of high interest. The entire pipeline starts with the **data acquisition**. The research vessel performs 3D sonar measurements co-registered with DGPS position and UTC time

in the scope of a *survey*. A survey consists of several 3D measurements of interesting areas over time. These measurements are denoted as *observations* and each observation consists of a set of single time-step 3D datasets denoted as a *ping*.

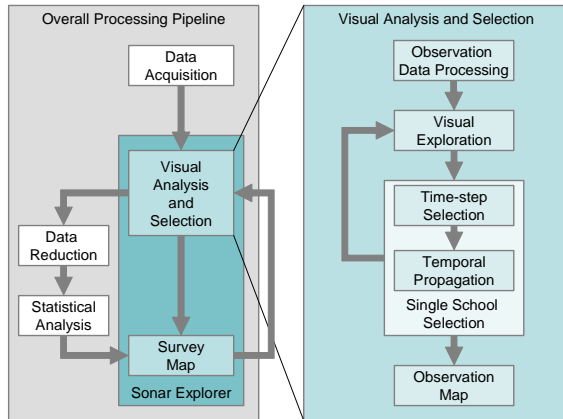


Figure 2: The overall processing pipeline with the Sonar Explorer pipeline in detail.

The **Sonar Explorer** allows to **visually analyze** entire survey data in order to semi-automatically generate survey maps of a sea region where the scanned fish schools are clearly depicted. This is an iterative process. The user analyzes every observation of the survey individually. For each observation all time-steps are explored. If a school is present, we provide a robust and easy way to **select** the school volume. After all fish schools of an observation are selected, this information is exported to the **survey map** and the visual analysis of the next observation is performed. Processing all observations results in a survey map of all identified fish schools.

Our system supports export of the segmented features from our application to allow processing beyond visualization. Exporting only selections can significantly help to **reduce the data** size of the entire survey. After storing the relevant data only, noise artifacts, entire pings or even all pings of an observation can be removed from the survey. In case when there is no interesting data in an observation, only the vessel path with time-stamps will be stored. The export is also important for further information extraction from the school data. The reduced size of the survey data enables processing of the data by **statistical analysis** packages to retrieve more information about the survey. Such information can be then included into the generated survey map. The overall processing pipeline is depicted in Figure 2.

4. Sonar Explorer

Sonar Explorer provides tools that enable a user to visualize 4D sonar data. The visualization tools aid the user in

looking for fish schools and selecting these for immediate overview and later analysis. The right part of Figure 2 shows the pipeline architecture followed in the application.

The acquired data for a single observation is first loaded into memory, processed for noise, and localized according to DGPS coordinates. This is performed in the **observation data processing** stage. After being processed, the data is available for **visual exploration**. It is at this stage that the user can search for fish schools. When a fish school has been located it is possible to perform **single school selection**. This process is divided into two parts. The first part is the **manual selection** of a single school in one time-step. The other part is to let this selection automatically be **propagated** in the temporal domain to neighboring time-steps. At the final stage the user can search for other schools or investigate the results in the **observation map**. We now take a closer look at the considerations that have been made during the design of our framework.

4.1. Dataset Characteristics

The data that the application has been designed for is generated by a sonar device. The device takes 3D images at a specified interval while the research vessel is moving. This generates a large 4D dataset. Due to the properties of the sonar device the data is given on a curvilinear grid. More specifically the covered volume is a cone in spherical coordinates. Since many existing volume rendering technologies are designed for volumes given on a rectilinear grid we just had to adapt the standard raycasting technique to our type of grid. A typical hardware accelerated technique uses proxy geometry to find startpoints and endpoints of all visible rays. The usual proxy geometry is a cube. This does not work efficiently with our data so a more suitable proxy geometry was needed. Additionally due to a limitation of current graphics hardware concerning 3D texture sizes, we had to use a more sophisticated storage scheme as a work-around.

The MS70 is a sonar transducer consisting of 25x20 beams. These beams are positioned in a grid and they are pointing in a 60 degree horizontal angle and 45 degree vertical angle. Each horizontal array is called a fan and transmits a signal at a specified frequency and then listens for the reflected signal. The reflected value is measured in decibel (dB). The number of samples along a ray is configurable. Our data has 1319 samples per ray and with a sampling distance of 0.38 m this equals to a beam length of approximately 500 meters. The data file has a ping interval of about 5-6 seconds. In addition to the volume data other devices on the vessel also provide information such as DGPS coordinates, UTC timing, and the dynamics of the vessel including heading, pitch, heave, and roll. This data is introduced into the data stream at a datagram rate of 10 Hz. Due to the nature of the sonar device, the returned signal strength is reduced with geometrical spreading and absorption. To compensate for this a time varied gain function [SIM96] could be ap-

plied. Noise is also an issue with the given data. We remove the fan closest to the surface because this fan contains noise from sea surface waves and air bubbles. We also remove the first 15 meters of slices parallel to the transducer since the noise here is produced by the sonar device itself. Other noise types that occur are: high intensity walls and high intensities along a beam sometimes occurring when loosing data packets in the data stream during the trial survey. These artifacts are strongly suppressed by applying median filters. These four noise types are illustrated in Figure 3.

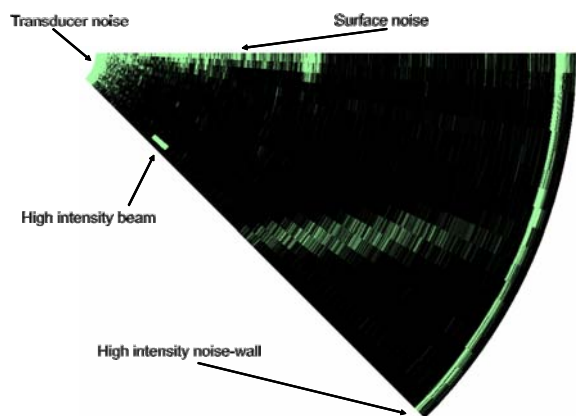


Figure 3: Examples of typical noise in 3D sonar data: transducer noise, water surface noise, high intensity beam, and high intensity noise-wall.

The sonar cone’s center is located at the transducer. The curvilinearity of the volume can be represented using spherical coordinates. The following equations describe the conversion from spherical coordinates (θ, ϕ, ρ) to Cartesian coordinates (x, y, z) :

$$\begin{aligned} x &= \rho \sin \phi \cos \theta \\ y &= \rho \sin \phi \sin \theta \\ z &= \rho \cos \phi \end{aligned}$$

These equations describe the conversion back to spherical coordinates:

$$\begin{aligned} \rho &= \sqrt{x^2 + y^2 + z^2} \\ \theta &= \tan^{-1} \left(\frac{y}{x} \right) \\ \phi &= \cos^{-1} \left(\frac{z}{\rho} \right) \end{aligned}$$

In computer memory the voxel values are positioned on a regular grid. The grid is interpreted as having coordinates in spherical coordinate space. This simplifies the conversion between a position in spherical coordinates and the corresponding position in Cartesian coordinates. The raycaster

that we have adapted processes the start and end ray positions in spherical coordinates. To traverse the ray through the volume from the start point to the end point, the position is converted to Cartesian coordinates before calculating the next ray position. After the new position has been located the position is converted back to spherical coordinates before the sample value is retrieved. This is done because a linear ray in Cartesian coordinates corresponds to a curved ray in spherical coordinates.

The volume is highly anisotropic. A technical problem that we encountered is due to limitations of NVidia’s implementation of 3D textures. NVidia has limited the dimension of these textures to 512^3 . Our volume data contains more than 512 samples in one direction. This problem is solved by folding the volume in the RGBA components. This technique works by putting the first 512 samples in the red component then putting the next 512 values starting with the 512th sample in the green component. This is continued in the same way for the blue and alpha components. With this approach we can have a volume with dimensions of up to $512 \times 512 \times 2045$. The reason we do not have 2048 in the last dimension is linear interpolation. To achieve linear interpolation in hardware, we need to repeat the last samples in one fold as the first samples in the next one. A GLSL shader takes care of returning the correct value from the appropriate component during raycasting and texturing. In general this technique can be used to handle anisotropic volumes with different configurations. Figure 4 illustrates two different configurations that might occur and the color component that could be used as a mapping.

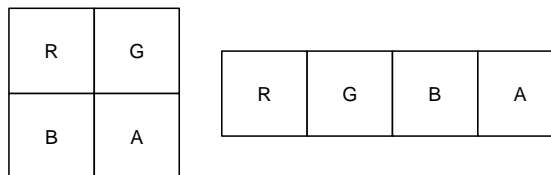


Figure 4: Two possible texture folding configurations. The R, G, B, and A values denote the color and alpha components used for storing.

4.2. Visual Analysis

The visualization part of the application is the most important interaction tool. It aids the user in finding schools, enables him to extract the features that are interesting, and gives the user a context in which the information makes sense. In Figure 5 the control panel that provides most of the interaction tools is highlighted in blue. For exploring an observation to find schools we provide 2D slicing and 3D volume rendering. These visualizations correspond to the widgets enclosed by the yellow rectangle in Figure 5. The two viewports on the left show axis aligned slices from the top

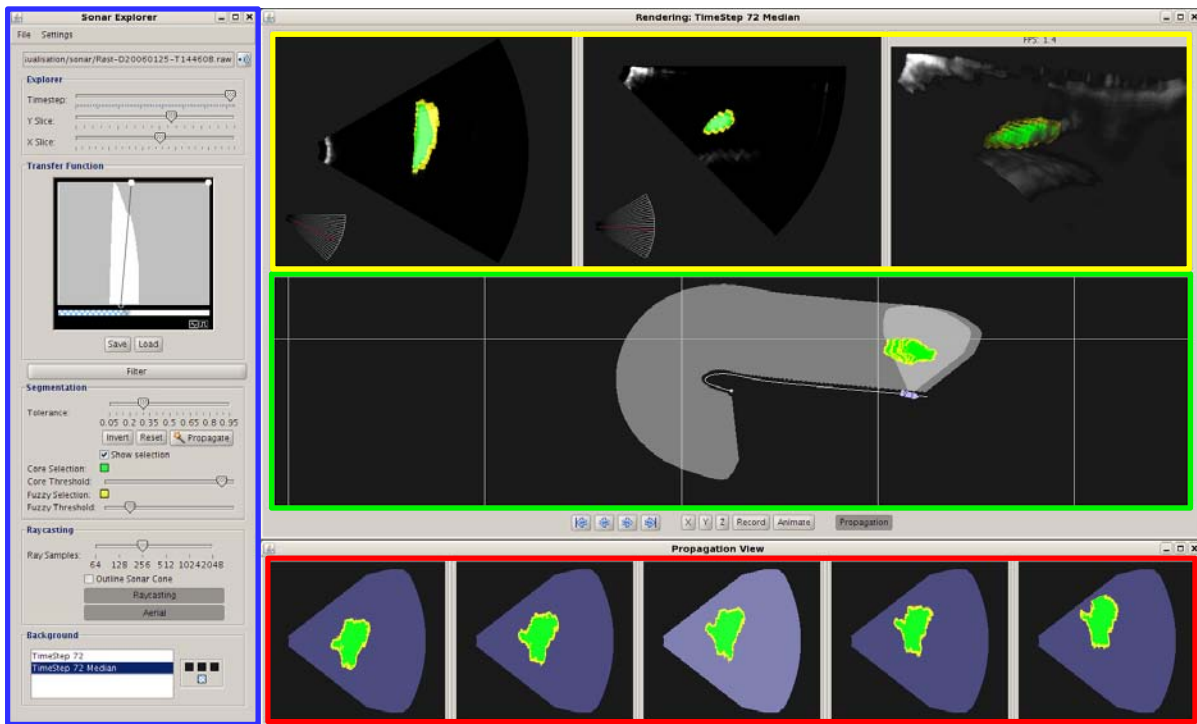


Figure 5: Screenshot of the Sonar Explorer application: Blue: The main control panel of the Sonar Explorer application. Yellow: 2D (top and side) and 3D visualizations of a single ping. Green: Observation map. Aerial overview of observation path (several pings). Red: Propagation view. Orthographic projection of segmentation masks (several pings).

and side of the sonar volume. The viewport on the right displays a direct volume rendering of the data. Both the slices and the volume rendering are dependent on the chosen transfer function. To give the user some context we provide an overview. The overview displays the path traveled during acquisition of the observation, where the current volume is located in this path, and a highlighting of any segmented data. The green part of Figure 5 shows the top-down aerial view displaying the overview in the so-called observation map. The application also provides a way of showing the results from neighboring time-steps. The red part of Figure 5 shows the propagation view which displays the current volume as well as the two previous ones and the two following ones.

The 2D slices provide an elementary way of exploring the data. These views are the ones that represent the data with highest accuracy since they are displayed without any filtering. The two viewports to the left in the yellow part of Figure 5 display axis-aligned slices from the top and from the side. Any given transfer function will change the output of this view. The slicing view allows the user to select data that are fish schools. The user utilizes the mouse to click a point in the slice and the coordinate of this point are used as a seed point in a region growing algorithm. After a segmen-

tation mask is created for the selection, it is blended with the displayed data to highlight the selected area.

The right viewport in the yellow part of Figure 5 displays the direct volume rendering of the sonar data. The volume rendering is an OpenGL GPU based raycaster adapted to a curvilinear grid. The algorithm achieves a speedup by rendering a proxy geometry that defines which pixels contain data that need to be raytraced. Default raycasting algorithms use a cube as proxy geometry but we have modeled the sonar cone and are using this as our proxy geometry. Analogous to the 2D slice view, the 3D rendered view is also using a given transfer function configurable from the control panel shown in the blue part of Figure 5. The transfer function defines which densities are transparent and which provide a visible contribution. Figure 1 illustrates the results that are possible by only adjusting the transfer function. Due to the high anisotropy of the volume some rendering artifacts occur. This can be resolved by increasing the number of samples along a ray but penalizes the performance.

The observation map is an aerial overview of the current observation. This view shows the vessel, vessel path, the convex hull of all ping locations, highlighting of the current ping location, and an orthographic projection of the seg-

mented schools. The projection uses the same basic raycasting algorithm supporting curvilinear grids but the projection is now orthographic. The raycasting in this view performs Maximum Intensity Projection (MIP) of the segmentation mask of the current volume.

To let the user see how the fish schools move over time, we provide the propagation view. The propagation view displays the projection of the selection mask of the current volume and the selection masks of adjacent time-steps.

If segmentation data is available the red, green, and yellow parts of Figure 5 will highlight this fact. The segmentation mask contains float values that represent the confidence of the segmented voxels being part of a fish school. In all the view-ports in Figure 5 one can see the highlighting as the green and yellow blobs. The green center is the core area and the outer yellow area is called the fuzzy area. These areas can be changed by modifying two threshold values. To highlight the boundary of the feature in the 3D view, we use first hit rendering to display the fuzzy area. The core area is raycasted using alpha accumulation.

4.3. School Selection

As previously mentioned, the school selection is heavily aided by the visualization. The visualization provides the user with a way of searching for and identifying schools of fish. After segmenting a feature the visualization highlights the selected area and the user can now in 2D and 3D check if there are parts of the school that have not been included in the selection. In addition the 3D view gives a context for the 2D slices. The user can in 3D see where in a volume the school is located and can then navigate the slices to this area. Another way of exploring is by adjusting the transfer function. Adjusting the transfer function will enable the user to a certain degree to highlight interesting values and suppress values that do not contain any useful information. Choosing a good transfer function results in better segmentation since the algorithm adapts to the given transfer function. The selection process is first performed in one time-step and then propagated to neighboring time-steps.

Feature extraction in one time-step is a two step process. First the volume is filtered and then any interesting features are manually segmented using region growing. The filtering is typically performed with a median filter. Due to the anisotropic nature of the volume we also employ an anisotropic 3D kernel. The kernel's dimensions mimic the anisotropy of the volume. We have used a kernel with dimensions 3x3x23. One aspect we do not consider is that the vertical and horizontal distances between voxels changes linearly as the kernel moves towards the perimeter.

After filtering the user will manually select a school through visual identification in the volume. The segmentation is based on a flood-fill algorithm that grows from a user-given seed point. The alpha value from the transfer function

for a voxel is used as a basis for the flood-fill algorithm. The absolute difference between the seed point's value and the value of the currently processed voxel is used to find out if a voxel is part of the fish school. A user provided tolerance value decides how large this difference can be before a voxel is discarded. A tolerance slider is visible in the control panel in the blue part of Figure 5. The difference and the tolerance is also used to calculate a mask value. This mask value is one when the difference is zero and zero when the difference equals the tolerance value. We use a fall-off function to create the values in between. The present function implemented is:

$$f = 1 - \left(\frac{d}{t}\right)^2$$

Where f is the falloff, t is the tolerance, and d is the absolute difference between the seed point value and the currently processed voxel. We always clamp the fall-off value to the range: [0, 1].

After a school has been identified and segmented in one time-step, the user can use the segmented school as a template for propagation to neighboring time-steps. First we determine the current tolerance of the segmented area. Then we calculate the center of gravity of the school and the bounding box. These properties are used to effectively propagate the school both forward and backwards in time.

To be able to have a seed point that will be valid in a neighboring time-step, we calculate the center of gravity with the assumption that the school is convex. The center of gravity that we calculate is weighted with the segmentation mask values. During this calculation we also determine the bounding volume of the mask. The bounding volume is used in the propagation to limit the flood-fill from growing into the sea bottom or up to the surface. We increase the size of the bounding volume to consider that the size of the selection may change from time-step to time-step. Currently this technique will stop the propagation before the school leaves the volume. Segmenting schools that are partly in a volume will be subject of future work.

Since the selected feature usually is the result of multiple flood-fills the selection's actual range of values will exceed the range of the tolerance. So any automated region growing using this tolerance will be sub-optimal. Our solution to this is to find the range of the selection and then calculate the tolerance that will cover this range. We also determine the density that corresponds to the center of the tolerance range. This center value is used in the region growing as a comparison value instead of the value located at the seed point determined by the center of gravity. This makes the tolerance value more effective.

By using the center of gravity from one time-step as a seed point in the next time-step we have an estimate of a location that should still be inside the school. After the region growing is completed, we recalculate the center of gravity and the

bounding volume. We increase the size of the bounding volume and if the bounds are outside of the sonar volume we stop the propagation. This means that schools that are only partly in the volume will not be segmented.

5. Results

At the end of our pipeline we have two results. First we provide a way for the user to save the selection masks. The selection masks give the statistical analysis tool the information it needs to perform calculations that are interesting to the user. The other result from our application is the survey map. The survey map illustrates for the user the survey path but also the location of schools. In Figure 6 the survey map has been manually composited with a colored depth map to provide context for the data. The survey map shows the results from three different observations. Each of the small green spheres indicates an observation starting point. The missing observation between the research vessel and the lower right observation was dropped because it did not contain any schools.

Performance: The application operates at interactive speeds but there are some bottlenecks that reduce the performance of the application. On the visualization side there are two tasks that are the major bottlenecks. Volume rendering and observation view rendering. Both of these visualizations use a raycaster to create the results. The volume renderer is slow because of the need to have a high sampling rate to overcome the anisotropy of the volume. The observation view does not have this issue since it renders in a direction that has a low resolution. The performance loss in this view happens if the number of segmented time-steps increases. Due to these issues the application lets the user turn off the volume renderer and the observation view if they are not needed. There is room for optimization and in the future we plan to solve some of the visualization problems by caching results and rendering at a lower resolution during interaction. Another performance bottleneck is filtering before selection. Median filtering is a demanding algorithm but is in fact currently the filtering method that has provided the best results. We would like to use the GPU in the future to increase the filter performance.

6. Sonar Explorer Use Scenario

The long-term goal for fishery-survey scientists is absolute abundance estimates of stock size for selected, important components of the ecosystem. Herring is now the largest fish stock on the Northern Atlantic, converting maybe as much as 100 million tons of zooplankton to human food. In ecological terms it is therefore a key component both as plankton feeder and prey for predators like cod and saithe. The Sonar Explorer will be an important tool for improving the accuracy of the biomass estimates of herring, how herring utilize its prey field and for quantifying its distribu-

tion within national boundaries. This is important for international negotiations when partitioning the catch quota. The uncertainty in the estimate of abundance is now closely connected with how the fish schools react to the presence of the vessel. The new system will certainly reduce this bias on herring stock estimate and hopefully also on the mackerel and capelin stock estimates. Future integration of data from other available instrumentation in a common display has a great scientific potential.

Sea bottom detection and removal, noise estimation, proper school detection with extraction of relevant morphological and energetic parameters of the schools are natural tasks here. Similarly, averaging and volume echo integration over sailed distance in range bins from the vessel to the maximum observation range can also be solved. These displays and outputs will give the operator and cruise leader invaluable information for changing or adapting the survey strategy during transecting. It will also provide data for proper statistical analysis of transect data in time and space.

The Sonar Explorer may also integrate other instruments operating simultaneously onboard the vessel, and which are for the data sampled from the MS70. These are the data from the low frequency fishery sonar SP72, giving one rough slice backscattering from about 2000 meters around the vessel, the EM 300 bottom mapping system, creating a detailed bottom map from a fan of beams under the vessel, the current velocity as measured from the Acoustic Doppler Current Profiler (ADCP), and the vertically observing echo sounders. Tracking the movement of the schools relative to the water masses may then be extracted from combining the information from several sensors.

Sonar Explorer now covers the important missing link between raw data collection and the postprocessing stage of the data from the sonar. The true geographical representation gives the cruise leader a good overview of the density distribution and structure, necessary for immediate decisions for adaptive sampling strategies. The possibilities for data reduction and detailed school analysis are important tools in surveys for pelagic fish.

7. Conclusions

In this paper we have presented Sonar Explorer, an application framework for semi-automatic generation of fish school 3D survey maps. The fish school selection is carried out by a visual analysis and an easy and intuitive selection via picking. This selection is automatically propagated in time by tracking fish schools in the temporal domain. We have presented various aspects of the underlying data that imply modifications to the standard GPU based volume rendering.

The 3D survey of fish schools results into a survey map where all identified schools are located. Identification of volumes with and without school data can be used as an efficient data compressor. The survey and the observation map serve

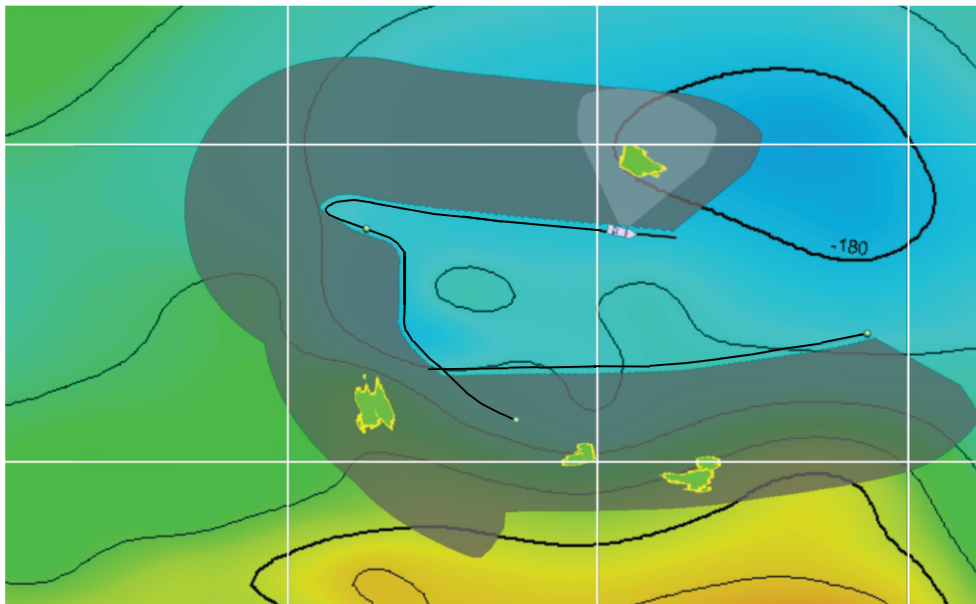


Figure 6: Survey map showing detected herring schools. Rough depth contours are indicated in the background, with the sonar search volume shown in grey.

as contextual views that are linked to the focus view (2D and 3D view) where individual pings can be analyzed. When used in real time, the system offers the cruise personnel an immediate overview of the survey situation for immediate actions, and provides tools for a detailed analysis of single schools. Further connectors to survey charts and statistical analysis systems are foreseen.

8. Acknowledgments

We thank Rolf Korneliusen for the data, Stefan Bruckner for helping with the GPU raycaster, and Torsten Möller for many interesting discussions.

References

- [ABGL06] ANDERSEN L., BERG S., GAMMELSÆTER O., LUNDE E.: New scientific multibeam systems (me70 and ms70) for fishery research applications. *Journal of the Acoustical Society of America* 120, 5 (2006), 3017.
- [DGH03] DOLEISCH H., GASSER M., HAUSER H.: Interactive feature specification for focus+context visualization of complex simulation data. In *Proceedings of VisSym '03* (2003), pp. 239–248.
- [DSH05] DOLEISCH H., STONAWSKI G., HAUSER H.: Simvis: Interactive visual analysis of simulation results. In *Proceedings of the NAFEMS Seminar on Simulation of Complex Flows (CFD)* (2005).
- [Gar90] GARRITY M.: Raytracing irregular volume data. In *Proceedings of SIGGRAPH '90* (1990), pp. 35–40.
- [Gun93] GUNDERSON D.: *Surveys of Fish Resources*. John Wiley & Sons, 1993.
- [Hau05] HAUSER H.: *Scientific Visualization: The Visual Extraction of Knowledge from Data*. Springer, 2005, ch. Generalizing Focus+Context Visualization, pp. 305–327.
- [KW03] KRÜGER J., WESTERMANN R.: Acceleration techniques for GPU-based volume rendering. In *Proceedings of IEEE Visualization '03* (2003), pp. 287–292.
- [MS93] MACLENNAN D., SIMMONDS E.: *Fisheries Acoustics*. Chapman & Hall, 1993.
- [ODK*06] ONA E., DALEN J., KNUDSEN H., PATEL R., ANDERSEN L., BERG S.: First data from sea trials with the new ms70 multibeam sonar. *Journal of the Acoustical Society of America* 120, 5 (2006), 3017.
- [RGW*03] ROETTGER S., GUTHE S., WEISKOPF D., ERTL T., STRASSER W.: Smart hardware-accelerated volume rendering. In *Proceedings of VisSym '03* (2003), pp. 231–238.
- [SIM96] SIMRAD: *Simrad EK500. Theory of operation*. Tech. rep., Simrad, Norway, 1996.
- [sim06] Simrad web site <http://www.simrad.com/>, 2006.
- [WCA*90] WIHELMS J., CHALLINGER J., ALPER N., RAMAMOORTHY S., VAZIRI A.: Direct volume rendering of curvilinear volumes. In *Proceedings of SIGGRAPH '90* (1990), pp. 41–47.