Navigation Display

Technical Specification

# Summary

This specification defines the desired technical capabilities and feature set for navigational display software to be included on all VideoRay professional systems.

The design goals are:

1. Offer useful capabilities to end-users, ideally ones that competitors do not possess.
2. Provide a user experience exceeding similar offerings in our market (Seanet, Hypack, most AUV mission planners, etc.)
3. Require minimal VideoRay developer resources and have a short development cycle.

It is envisioned that VideoRay will offer multiple tiers of navigation products. The lowest tier (baseline offering) would have minimal navigation sensors, possibly only a surface GPS. The highest tier would include multiple positioning systems. The navigation display should support all tiers, higher complexity tiers may be supported by for pay add-ons, but the base software platform should be identical across all tiers.

It is envisioned that this will be a VideoRay branded product, possibly with original developer sub-branding (for example “VideoRay Navigation by X”). Ideally VideoRay would have source code access or there would be some facility (SDK, open plugin API, etc.) to enable VideoRay to add custom features in the future. It is desired that the baseline level will be provided without license keying and would be available for download from VideoRay. Additional add-on features can be hard or soft-key locked.

The product must work offline without an active internet connection. Ideally any caching of background maps will be as seamless as possible.

# Product Specification:

Potentially not all features need to be included in the baseline product.

## Display Engine

### Vehicle Position Display

#### Allow for relative (N/E) or Geo-positioning (Lat/Lon)

UTM northing/easting is acceptable provided there is an easy method to set the base position.

SeeByte: Lat/Lon WGS84 datum will be the supported Geographic Location format; if required a relative coordinate frame to some arbitrary (0, 0) can also be supported. UTM zones / grid-references will not be supported (at least initially – is there really a demand?).

#### Provide a continuous track of a single surface platform position

This could be from a surface GPS, for example on a surface ship. This could also be a user entered static position.

SeeByte: Supported

#### Provide a continuous track of multiple surface platform positions (via multiple GPS feeds)

SeeByte: Supported

#### Provide the ability for EASY estimation of ROV or other subsea target based on range and bearing from surface.

For example: Display a range bearing circle via a mouse drag. This would allow the end user to estimate the ROV position based on an operators location and a range/bearing shot through a rangefinder.

SeeByte: We propose to support dialog entry of range/bearing from an existing contact (e.g. surface vessel; marker; point on a chart); mouse-based can also be supported (similar to CoPilot-style).

VideoRay: As discussed w/ JE this will ideally be a drag able range circle/line UI element with textual range/bearing information.

#### Provide a display of a sub-surface vehicle based on discrete points, with an aging display based on time of last fix.

Display each point from an ROV mounted GPS for example, provide some feedback as to age of last fix (fading, turning red, increasing uncertainty circle, etc.)

SeeByte: Supported.

Understand that VideoRay may wish to charge for this capability.

#### Provide a display of a sub-surface vehicle based on a more continuous track

For example using one based on the feed from a USBL/DVL/or smart tether.

SeeByte: Supported. It is assumed that the data is in a NMEA/GPS style format. No additional filtering of position data.

Understand that VideoRay may wish to charge for this capability.

#### Provide a continuous track multiple sub-surface vehicles

SeeByte: Supported, to the same level feature level as a single sub-surface vehicle.

Colors / labels will be used to distinguish between targets.

Understand that VideoRay may wish to charge for this capability.

#### Provide a display of vehicles heading/depth.

Heading should be displayed as an arrow or similar graphic at the current Position

Depth can be displayed numerically on a 2-D display.

### Provide a display of sensor look area/volume, based on selected sensor and vehicle heading for each position fix.

For example see KCF SmartTether software.

### Provide a display of ROV tether when used with a tether tracking system

### Chart Background Display

#### Allow for the use of raster charts

NOAA bsb

Geotiff

JPEG (Exif or side-by-side world file)

User registered, maybe via external tool like blue marble geographic

SeeByte: We can only support initially whatever World-Wind will support natively / easily. We have previously used GeoTIFF successfully. There may be a 3rd party (license-dependent) integrated converter for NOAA bsb files; otherwise this would have to be via 3rd party tool like Blue Marble/Global-Mapper. (Not sure was JTAG files are).

#### Allow for the use of tiled raster data sets

Automate download of legal Landsat/Bing/etc. imagery. See <http://code.google.com/p/gmapcatcher/>)   
Needs to support caching for offline usage

SeeByte: We will provide some form of support for caching of support; this actually quite tricky. Our initial ideas are to manipulate the existing World-Wind cache management to “save” areas already visited. Regarding utilities or integrated code to take download large sets of imagery; we have no experience of this, and will investigate, but it may have to wait until later developments.

#### Allow for the use of multiple stitched raster charts

Ideally would support multiple resolutions/multiple zoom levels

#### Allow for the use of Vectored Charts

S-57

DXF

#### Allow for the use of Vectored overlays

Ship shape

Subsea structure shape

DXF, etc.

#### Allow for the use of raster iconography

Allow for multiple types of objects to be designated on the display.

Such as mine shape, rock, anchor point, etc.

Standard .mil iconography would be a plus.

SeeByte: We can support standard 2D-iconography which we have artwork for. We have used military iconography in COIN/SeeTrack Military so this should be possible.

### Data Background Display

#### Allow for the import and display of side scan sonar imagery in native data formats.

#### Allow for the import of bathymetric data

#### Allow for the import of gridded sensor data

This would be things like radiation sensors or ph meters, where the data is gridded via an external program.

### User Spatial data entry

#### Allow for the entry of navigational offsets

Allow to correct for things like GPS being offset from the ROV launch point and similar.

SeeByte: Supported.

#### Provide the ability to Mark positions (lat/lon/depth/time)

This should be mouse click and a way to mark current position of some tracked vehicle.

SeeByte: Supported.

#### Provide the ability to generate sets of positions (way points) for mission planning

Different levels of sophistication could be supported by for pay add-ons

#### Provide the ability for the user to draw arbitrary shapes for markup purposes

These shapes should be spatially positioned data sets. Polygons and Circles will probably cover most use cases. It is envisioned that these “shapes” contain significantly more points than the data sets above.

#### Allow for the calculation of various spatial measurements

In addition to range/bearing to target, measurements such as lengths, areas, volumes, x-track errors, etc. would be nice.

SeeByte: We can initial support a standard “ruler” tool to estimate range/bearing between two points; other more complex spatial measurement will require more development effort and will have to later releases.

### General Display Features

#### Pan/Zoom/Rotate

#### Hide /Show tracks, points, etc.

#### Save screen shot as JPEG

#### Allow for user defined systems of measure and Datum

This would include parameters such as Imperial/metric, wgs84, etc.

SeeByte: All supported, though note previous comment that WGS84 Lat/Lon will be the only datum supported.

## Navigation Data Import/Export

### Support data import via serial ports and network sockets

Standard com ports and udp sockets should be supported

SeeByte: Supported.

### Support standard positioning data streams

Standard Positioning NMEA-0183 (GGA, GLL, RMC)

Some Range/Bearing sentence (trackpoint, micronnav, etc.)

Standard Heading/depth sentences (HDG, DPT)

SeeByte: Supported.

### Support a range-only sentence

This is to support a device such as a payout sheave or tether tender

### Required Hardware Support

These can be supported via standard communication interfaces

ROV Depth sensor

ROV Attitude sensor (heading, pitch, roll)

ROV GPS mast

USBL

Tether Payout sheave

Surface GPS

Surface Attitude (heading, pitch roll)

Atmospheric pressure sensor

Smart tether

### Support a to be defined multiple position data sentence

This is to support a smart tether

### Support generic ASCII comma delimited data strings

Provide the user some method of mapping the fields to some actual data context

### Provide diagnostic display of raw data

For example sort of terminal display

### Provide simplified display of individual data streams

Simple text dialogs of parsed data, for example.

### Support export of waypoints to external packages

This is seamless export to mission planning software, hypack, etc.

### Provide the ability to export data sets as ASCII and KML

## Data Replay

### Provide the ability to replay stored data files

SeeByte: Assuming this is replay of the navigation fixes (GPS; manual entry; etc) with the associated timeline, ideally with some speed-up/down functions then this will be supported. Synchronized replay of (for example) video or sonar files against the navigation track will **not** be supported initially (though is on the roadmap).

### Provide the ability to display properly coded non-spatial data products

This would be the ability to click on a track point and call up a video still that was saved with appropriate temporal or spatial information. This some sort of “database” or other metadata protocols to be defined.

### Provide the ability to filter the display based on time or location

This is essentially just displaying snippets of data.

SeeByte: Some playback support to jump to particular times/periods. Tools to permanent edit log files based on time criteria will **not** be included initially.

## Data Manipulation/Storage

### Provide the ability to store raw data files

SeeByte: Support to log navigation inputs (GPS, Depth, Heading etc) in raw form will be provided. No support to log (say) sonar or video data will be in the initial release.

### Provide the ability to filter raw data to improve navigation quality in real time

#### Provide the ability to average multiple surface gps fixes to improve static point estimations

### Provide the ability to fuse raw data streams into a full navigation solution

This would allow for the calculation of a sub-sea position based on streams from ship attitude sensors and multiple sub-sea navigation sensors.

### Provide the ability to calculate a subsea position based on DVL and attitude data

### Provide the ability to correct sensor data based on ancillary data

This includes features such as correcting depth for atmospheric pressure, correcting headings for compass variation, etc.

### Provide the ability to re-run navigation routines on raw data files

### Allow for the creation of gridded sensor data by combining sensor feeds with navigation information

## Support multiple external display clients

### Provide an embedded WMS server to publish mapping /position data to multiple clients

### Provide a KML server to act as a network link for compliant viewers such as google earth.

## SDK/API/Extension Mechanisms

Ideally VideoRay would have access to the source code for the navigation display. If this is not possible some facility for extending functionality is desired.

SeeByte: Significant amount of background IP will be used in the system, so access to main source would be problematic.

SeeByte would be happy to open any required interfaces; plus provide example source code to use them.

### Provide a method to display spatial data points, with associated textual meta data.

This could be a standard communication interface

SeeByte: Support for display points (with textual meta-data) on the main display to provide functionality similar to pointing a GPS point can be provided. It is not **envisaged** that this would be used for repeated, high-speed, real-time access (for example animations).

### Provide a method to get real-time spatial button click events

This could be a standard communication interface

### Provide a Programmatic Interface to all functions and features of the navigation display

SeeByte: We would be happy to develop such open interfaces for access to as much as possible, but this would certainly slow down the initial developments and hence release date. SeeByte would suggest providing this for later releases.