

Open-Source Web GIS for Arctic Seafloor Mapping: Improving the Interactivity of Public Data Dissemination

by

Madeline Vainionpää

Bachelor of Science, University of Waterloo

A Thesis Submitted in Partial Fulfillment
of the Requirements for the Degree of

Master of Science in Engineering

in the Graduate Academic Unit of Geodesy and Geomatic Engineering

Supervisor: Ian Church, Ph.D., Geodesy and Geomatic Engineering

Examining Board: Karl Butler, Ph.D., Earth Sciences

Marcelo Santos, Ph.D., Geodesy and Geomatic Engineering

This thesis is accepted by the Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

March, 2024

© Madeline Vainionpää, 2024

Abstract

The ease-of-use of online GIS software has encouraged many organizations to publish their geospatial data to a web mapping interface. With respect to the field of ocean mapping, this has been a breakthrough towards public accessibility since this data is easy to absorb in a visual format. There is a specific need for a Canadian Arctic-focused web portal which can host a unique dataset collected by the CCGS Amundsen over several decades. This project employs open-source JavaScript to not only display the Amundsen's dataset on a web GIS interface, but to display it alongside third-party seabed data. The end user can manipulate the datasets using two interactive toolkits. The first is a statistical analysis toolkit which compares bathymetry raster imagepyramids that are hosted using web mapping services (WMS). The second is a three-dimensional visualization tool which virtually draws bathymetric WMS information in the same scene as cross-sectional seabed subsurface data.

Dedication

Dedicated to my mother Elizabeth and my father Timo with love. Thank you for supporting me and for believing in me.

Acknowledgements

This project would not have been possible without the work of countless individuals in the open-source JavaScript community who graciously chose to share their hard work with the public under no expectation of return. In a world where knowledge is often hidden behind paywalls and commodified for profit, open-source developers demonstrate tremendous integrity by sharing the fruits of their intellectual labour for the common benefit.

Table of Contents

Abstract.....	ii
Dedication.....	iii
Acknowledgements.....	iv
Table of Contents.....	v
List of Tables.....	vii
List of Figures.....	viii
List of Symbols, Nomenclature, or Abbreviations.....	x
1. INTRODUCTION.....	1
1.1 Research Premise.....	2
1.2 Research Question.....	2
1.3 Objectives.....	2
2. FOUNDATIONAL WORK.....	3
2.1 Development of UNB OMG ArcticNet Web Suite.....	3
2.2 Ocean Modeler’s Web Mapping Application.....	4
2.3 Locally Hosted Arctic Seabed Data.....	6
3. LITERATURE REVIEW.....	10
3.1 Third-Party Arctic Bathymetry Data and Viewers.....	10
3.2 Web Mapping Architecture.....	12
3.2.1 GeoServer Back-End.....	13
3.2.2 OGC Standard Protocols.....	14
3.2.3 Querying a Geospatial Server – XMLHttpRequests.....	15
3.2.4 Querying Limitations: CORS Policy.....	17
3.2.5 Web Mapping Interfaces.....	18
3.3 Three-Dimensional Modelling in JavaScript.....	18
3.3.1 Converting GeoServer Output to JS-Parseable Format.....	19
3.3.2 Converting Sub-Bottom Profiles Into JS-Parseable Format.....	20
4. METHODOLOGY.....	23
4.1 TASK 1: Spatial Analysis Tool.....	23
4.1.1 Step 1 : Single Point Query.....	24
4.1.2 Step 2: Transform Areal Selection Into geoJSON Object.....	25
4.1.3 Step 3: Query Areal Selection.....	26

4.1.4	Step 4: Query Multiple Layers.....	28
4.2	TASK 2: Three-Dimensional Seabed Viewer.....	29
4.2.1	Step 1: Topographic Mesh Generation Using Three.JS and WCS Input.....	29
4.2.2	Step 2: Overlaying Image Data onto Bathymetric Wireframe.....	32
4.2.3	Step 3: Navigating 3D Viewer Using 2D Map	33
4.2.4	Step 4: Creating 3D Point Cloud from Sub-Bottom CSV	35
4.3.	Setting Up a Local Geospatial Server.....	37
4.4.	Javascript Plug-Ins Used.....	37
5.	RESULTS	39
5.1	TASK 1: Spatial Analysis Tool	39
5.2	TASK 2: Three-Dimensional Seabed Viewer.....	40
6.	CONCLUSIONS.....	42
6.1	Limitations	43
6.2	Recommendations for Implementation on UNB Back-End Servers	44
6.3	Recommendations for Future Work.....	45
	Bibliography	48
	Curriculum Vitae	

List of Tables

Table 1 - Arctic Bathymetry Portals 10

Table 2 – Third-Party JS Plug-ins Employed 38

List of Figures

<i>Figure 1 (a)</i> – Google Maps ArcticNet Interface. <i>(b)</i> ArcticNet Basemaps Series. <i>(c)</i> Transit data in stripmap format (Muggah, 2010).....	3
<i>Figure 2 (a)</i> – Main interface for Ocean Modelers’ Web Mapping Application. <i>(b)</i> Bathymetry point querying feature (Ruiz, 2018).....	5
<i>Figure 3</i> – Survey responses to the question ”What are the primary roles of your work unit?” (Ruiz, 2018).....	6
<i>Figure 4</i> – Bathymetry, backscatter, and sub-bottom profile data (UNB, 2014).	8
<i>Figure 5</i> – Sub-bottom inline popup on the UNB ArcticNet Interface (UNB, 2014)	9
<i>Figure 6</i> – Basic Web Mapping Architecture (modified from Ruiz, 2018).	13
<i>Figure 7</i> – OGC standard protocols (modified from Geosolutions, 2020).....	15
<i>Figure 8</i> – Sub-bottom profile png image projected onto vertical curtain in ThreeJS	20
<i>Figure 9</i> – Overlapping sub-bottom data due to stacking of multiple profiles. (UNB, 2014)	21
<i>Figure 10</i> – The results of a point and click query shown in the upper right. The input bathymetry layer is from EMOD’s geospatial servers.....	25
<i>Figure 11</i> – Histogram generated from querying the region bounded by the orange rectangle. Bathymetry layer provided by the IHO’s public WMS.	27
<i>Figure 12</i> – Single point query of 2 layers. The input bathymetry layers are WMS layers from the IHO’s and EMOD’s geospatial servers.....	28

<i>Figure 13</i> - Plane subdivided into triangular mesh. The coloured lines represent the axes of the 3D plot.	30
<i>Figure 14</i> - Mesh generated in three JS by modifying a plane. Data taken from the ArcticNet Basemaps Series (UNB, 2014).....	31
<i>Figure 15</i> – Backscatter imagery overlaid onto bathymetric wireframe.	33
<i>Figure 16</i> – Three dimensional topographic mesh generated from a user-selected bbox (orange) on the 2D map.	35
<i>Figure 17</i> - Sub-bottom input file represented in ThreeJS as a point cloud. Red pixels represent points of higher intensity return.	36
<i>Figure 18</i> – Bathymetry QA/QC Diagnostic Window. The WMS layers used for sampling are supplied by the IHO and the EMOD.	39
<i>Figure 19</i> – 3D seabed scene. Seafloor topography is represented as a cyan wireframe, and sub-bottom intensities by a point cloud.	41
<i>Figure 20</i> - Category zone of confidence (CATZOC) map presented in the CPPT (Chénier et al., 2018).	46

List of Symbols, Nomenclature, or Abbreviations

ASWMG	Arctic Seabed Web Mapping Guidebook
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
CATZOC	Category Zone of Confidence
CCGS	Canadian Coast Guard Ship
CORS	Cross-Origin Resource Sharing
CPPT	CHS Priority Planning Tool
CSS	Cascading Style Sheets
CSW	Catalog Services for Web
DOM	Document Object Model
GEBCO	General Bathymetric Chart of the Oceans
GDAL	Geospatial Data Abstraction Library
GIS	Geographic Information System
GUI	Graphical User Interface
HTML	HyperText Markup Language
HSL	Hue, Saturation, Lightness
IBCAO	International Bathymetric Chart of the Arctic Ocean
JS	JavaScript
JSON	Javascript Object Notation
OGC	Open Geospatial Consortium
OMG	Ocean Mapping Group
PNG	Portable Network Graphic

QA/QC	Quality Assurance / Quality Control
SOP	Standard Operating Procedure
SQL	Structured Query Language
UCD	User-Centered Design
UNB	University of New Brunswick
URL	Uniform Resource Locator
WCS	Web Coverage Service
WFS	Web Feature Service
WMS	Web Map Service
WMTS	Web Map Tile Service
WPS	Web Processing Service
XHR	XMLHttpRequest
XML	eXtensible Markup Language

1. INTRODUCTION

Several web portals are available online for displaying and downloading bathymetry, as well as other types of hydrospatial data (explored later in Section 3.1). These web portals often host single-source data collected by one organization rather than condensing many sources into a comprehensive repository. Hosting a single dataset is limiting when it comes to Arctic seabed mapping data since coverage at high latitudes is sparse due to the harsh climate and heavy sea ice coverage (Abulaitijiang et al., 2019). As of 2015, only 11% of the Arctic Ocean was mapped to modern resolution standards (Jakobsson et al., 2015). This scant coverage lessens the utility of continuous global bathymetric maps (such as the General Bathymetric Chart of the Ocean, or GEBCO) since much of the data in the Arctic regions of these continuous grids are infilled using mathematical interpolation. Current online Arctic data portals are also limited when it comes to the ways in which the end user can engage with the data, typically offering two-dimensional image-based map displays. These issues highlight the need for a comprehensive Canadian Arctic web portal for seabed data – one which combines all available datasets into a single repository, and which allows the end user to interact with the data in more meaningful ways. This can be done by offering both 3D visualization and spatial analysis tools. These tools would allow the end user to interact with the data, rather than simply acting as a 2D display. A 3D approach is specifically useful for hosting cross-sectional seabed data, such as sub-bottom profiles. These are currently often displayed as Portable Network Graphic (PNG) images (referred to as stripmaps by UNB’s Ocean Mapping Group in Muggah et al., 2010). This method of display distorts the data by projecting it to a plane and makes it difficult to georeference accurately.

1.1 Research Premise

Adding interactive elements to a multibeam and sub-bottom sonar-based web GIS environment increases the capabilities of the end user engaging with custom-source data.

1.2 Research Question

How can the current method of data dissemination used to host UNB's Arctic seabed mapping data be improved upon and modernized using interactive elements?

1.3 Objectives

The research objectives, intended to answer the above question, are as follows:

1. Implement and document a browser-based open-source data sampling tool that retrieves statistical diagnostics from several Web Mapping Services (WMSs) concurrently.
2. Create and document a 3D seabed visualization tool for remote scientists to access and discover data from UNB's sub-bottom profile repository using open-source JavaScript.

2. FOUNDATIONAL WORK

The task of creating a webmap portal for publicly hosting UNB's multibeam and sub-bottom data has been previously explored by Beaudoin et al. in 2008, by Muggah in 2010, and again by Ruiz in 2018.

2.1 Development of UNB OMG ArcticNet Web Suite

The foundational works of Beaudoin et al. (2008) and Muggah (2010) resulted in the release of three websites – the Google Maps Arctic Bathymetry viewer, the Arctic Basemaps Series, and the Arctic stripmap viewer. These are pictured in *Figure 1* below.

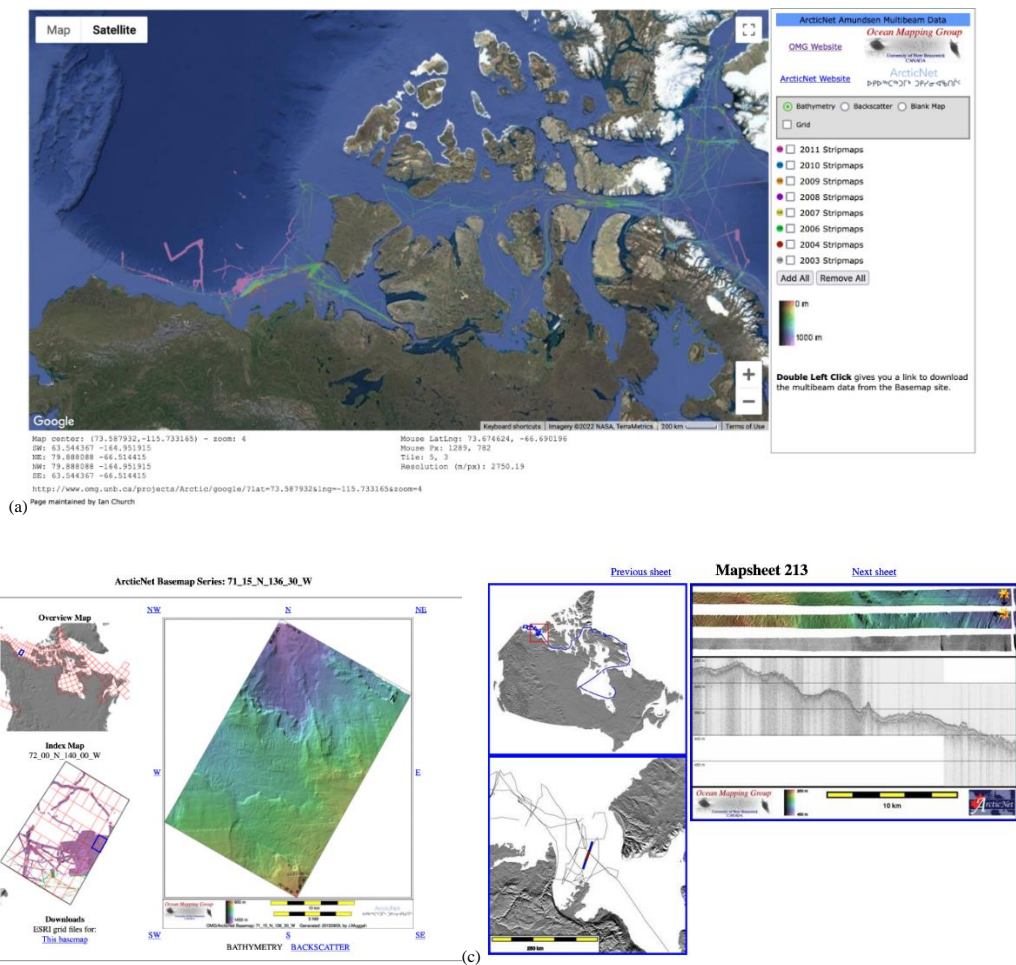


Figure 1 (a) – Google Maps ArcticNet Interface. (b) ArcticNet Basemaps Series. (c) Transit data in stripmap format (Muggah, 2010).

This global map displayed in the Google Maps ArcticNet Interface (Figure 1(a)) is intended to act as a navigational tool for the other two websites. It prompts the user to select their area of interest and shows the full spatial extent of the datasets available. The Basemaps Series webpage shown in 1(b) hosts UNB's bathymetric grid files for download, while the Stripmaps Series webpage shown in 1(c) hosts UNB's cross-sectional sub-bottom data in the form of png images. The sub bottom images are projected onto a plane which follows down the middle of the linepath of the bathymetry and backscatter data shown above it.

The stripmap portion of this website is currently used by scientists aboard the Canadian Coast Guard Ship (CCGS) Amundsen. They use this sub-bottom profile information to target locations when core sampling aboard the ship, so there is a significant built-in userbase for this website. In addition to hosting sub-bottom profiles, Muggah (2010) compiled the multi-year multibeam dataset to which UNB has access into a single image pyramid, which is displayed on the larger global map shown in *Figure 1(a)* as a way to aid the user when navigating to their area of interest on the larger national map.

2.2 Ocean Modeler's Web Mapping Application

In 2018, Ruiz began building on Beaudoin et al. (2008) and Muggah's (2010) work while incorporating different types of oceanographic datasets alongside the original multibeam data. This portal was focused on the lower Saint John River estuary, but the work could be expanded to cover other areas. While the portal is not currently active, the code remains on the UNB servers. Sample screenshots from Ruiz's web portal are shown below in *Figure 2*.

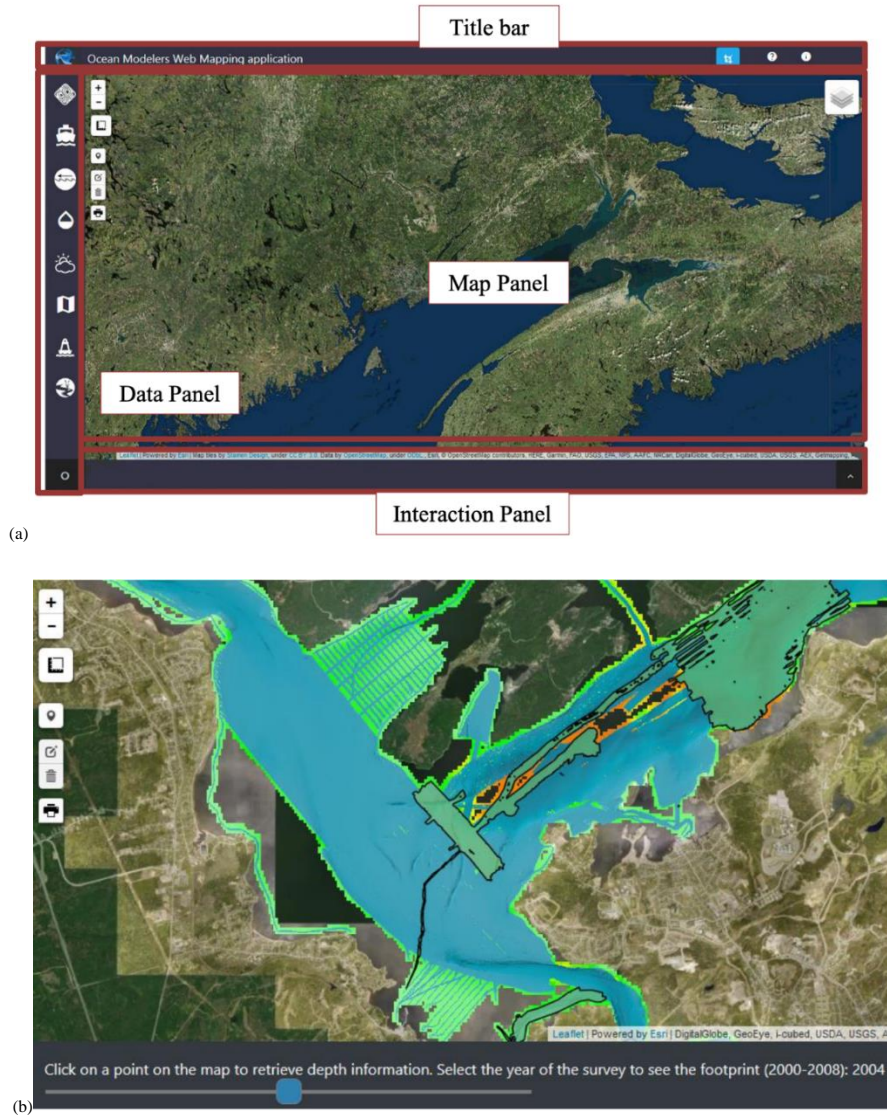


Figure 2 (a) – Main interface for Ocean Modelers’ Web Mapping Application. (b) Bathymetry point querying feature (Ruiz, 2018).

Ruiz’s paper was also focused around UCD (user-centered design), making the suggestion to prioritize the goals of the end-user by continually collecting feedback during each stage of production. Ruiz conducted a poll which asked potential users of their webmap what their field of work was. The results are displayed below in *Figure 3*.

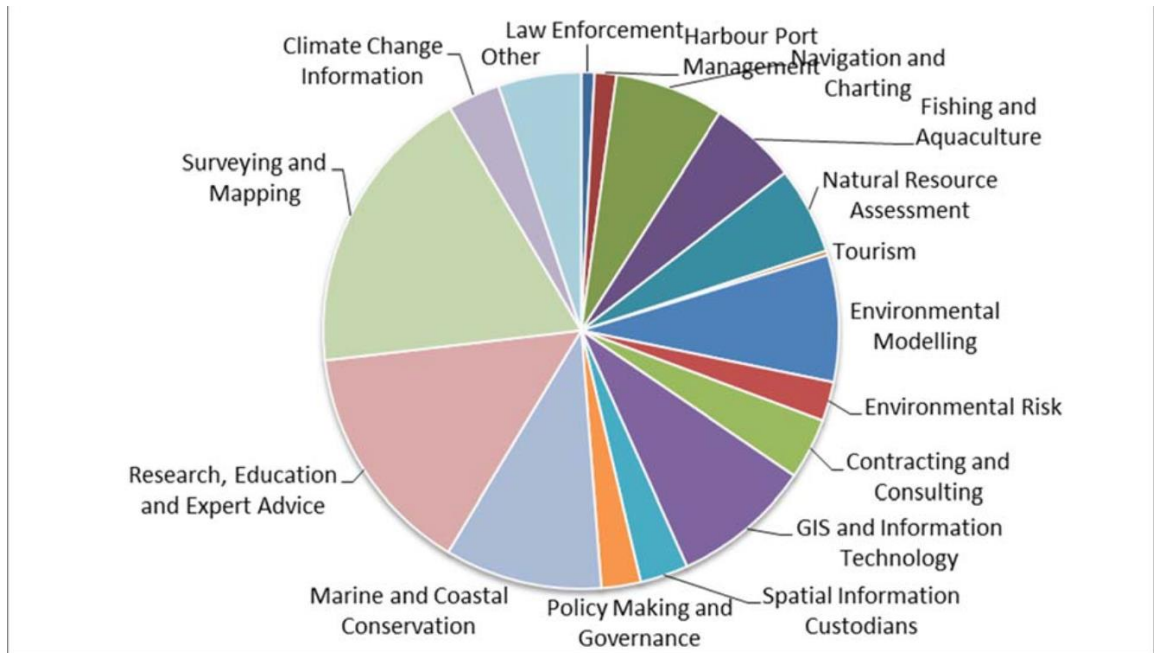


Figure 3 – Survey responses to the question "What are the primary roles of your work unit?" (Ruiz, 2018).

The results of this survey show how diverse the potential users of this type of seabed mapping website are.

The goal of this project is not only to modernize the previous foundational work of Beaudouin et al. (2008), Muggah (2010) and Ruiz (2018), but also to incorporate new browser-based spatial analysis and 3D modelling tools into the UNB OMG suite of web maps.

2.3 Locally Hosted Arctic Seabed Data

The web tools discussed in Section 2.1 above were created for the purpose of hosting only the Arctic seabed data that UNB's Ocean Mapping Group has collected in collaboration with ArcticNet and Amundsen Science. This data (hereafter referred to as "the Arctic Basemaps Series") is comprised of the following:

1. Bathymetry. This represents the depth of the surface of the seafloor, reduced to a common datum. Available in individual ESRI grid files as well as a single globally tiled raster WMS.
2. Backscatter. This represents the intensity of diffuse reflection scattered from the surface of the seafloor, which can be used to make inferences about the surficial geology. Available in individual ESRI grid files as well as a single globally tiled raster WMS.
3. Sub-bottom profiles. These represent a cross-sectional view of the subsurface in which interfaces between contrasting sediment types are plotted as regions of high intensity acoustic reflections. Hosted files are currently available in image format only.

An example of each data type is shown below in *Figure 4*.

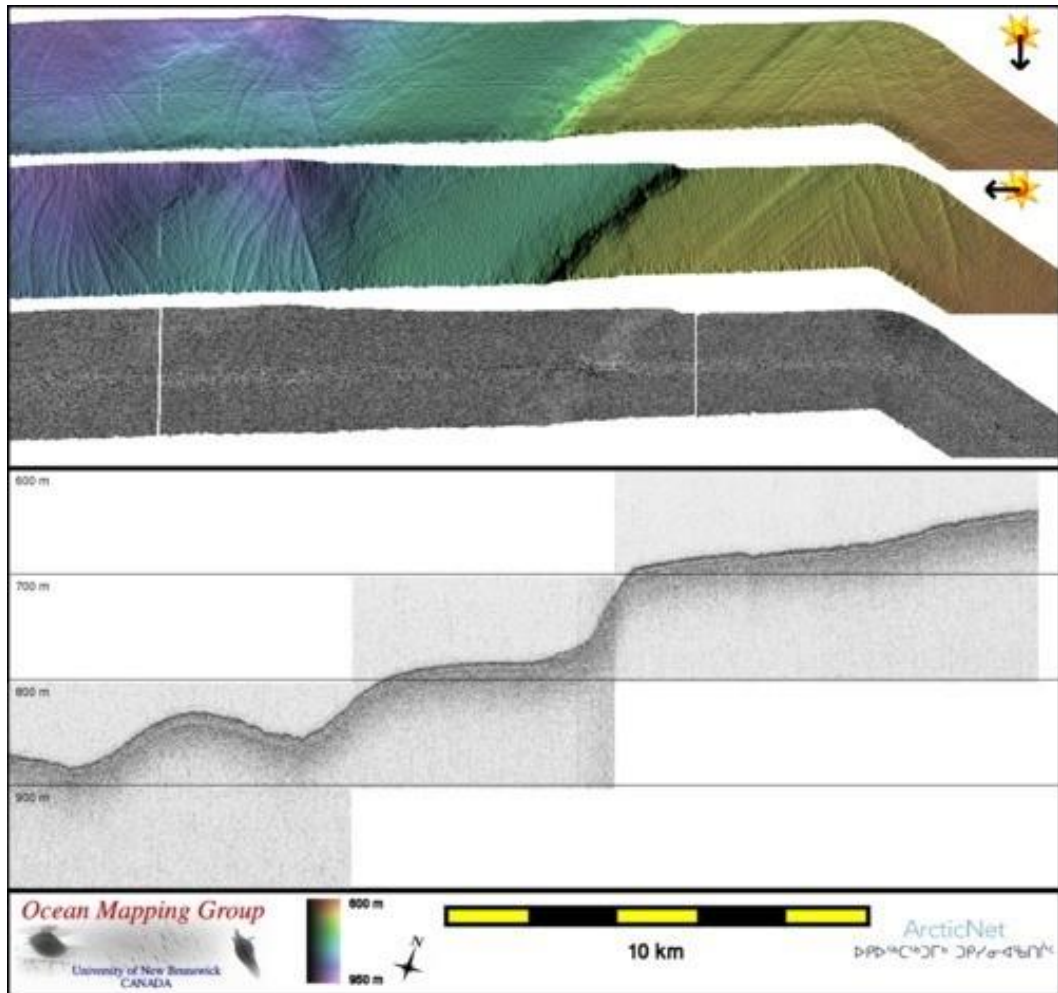


Figure 4 – Bathymetry (topmost two images, coloured), backscatter (middle, greyscale), and sub-bottom profile data (bottom) (UNB, 2014).

One of the major recurring uses for UNB’s current Arctic data portal is as a repository for data type (3) sub-bottom profiles. These aid field scientists in targeting core sample locations as they traverse through the Canadian Arctic. This data is not possible to display as an overhead raster tileset like the bathymetry since it is a cross-sectional vertical profile. The sub-bottom profile locations are instead currently displayed as linepaths. The end-user clicks on the linepath of their choice in order to open the

corresponding sub-bottom profile for the desired area (which then appears in an inline popup, as shown in *Figure 5*).

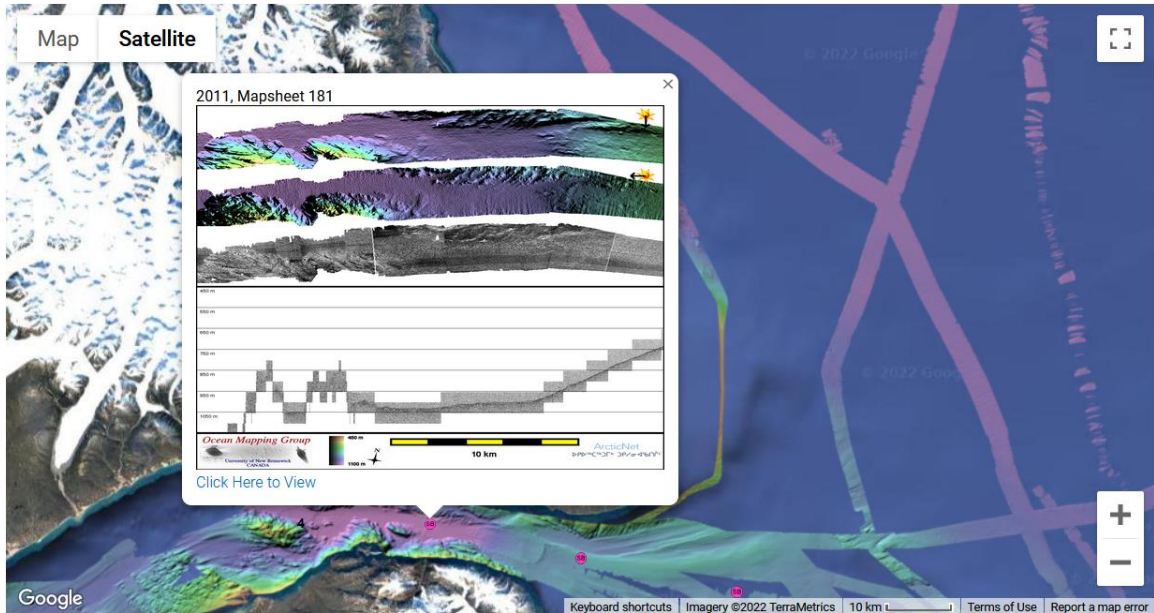


Figure 5 – Sub-bottom inline popup on the UNB ArcticNet Interface (UNB, 2014)

Although this 2D display method is serviceable, it has several drawbacks. For one, the sub-bottom profiles themselves are represented as simple png images. This makes it difficult to pinpoint the coordinates of areas along the profile accurately. The lack of tooltips or explanatory text assumes that the user is already familiar with the three different data formats that they are looking at. The 2D approach also neglects to consider any kinks in the shiptrack, representing the stripmap png images as though they are segmented into perfectly straight lines. Lastly, the 2D display lacks the ability for the client to visualize the profile within its surroundings. For these reasons, a 3D display tool would improve the interface.

3. LITERATURE REVIEW

3.1 Third-Party Arctic Bathymetry Data and Viewers

Arctic ocean mapping data is typically made available online through globally focused web mapping portals rather than Canadian or Arctic-specific ones. Many of these portals are either purely visual tools or are limited in the data formats that they can serve to an end user. A sample of some of these major publicly available resources for Arctic bathymetry is shown below in *Table 1*.

Hosting a WMS (Web Map Service) allows a mapping agency to serve their geospatial data openly to clients online (Open Geospatial Consortium, 2023). The mapping organization themselves might then provide a webmap interface where users can navigate through this data. Alternatively, third-party users from outside of the organization can overlay the WMS layers onto their own webmap. In the next column of *Table 1*, a WCS (Web Coverage Service) allows the user to retrieve the raw data points that were used to generate the raster WMS (Open Geospatial Consortium, 2023), among other things. This makes WCS an ideal format for modelling or analysis. In the next column to the right: an online mapping service that has Cross-Origin Resource Sharing (CORS) enabled will allow a third-party to access the full capabilities of both WMS and WCS services (Esri, 2023). Further details about WMS, WCS and CORS are discussed in Section 3.2.

Table 1 – Arctic Bathymetry Portals

ID	Organization / Affiliation	Coverage	WMS?	WCS?	CORS on?	Subjective Notes	URL
NOAA Bathymetry Viewer	NOAA, IHO	Global, but US-centric	Yes	Yes	Yes	Comprehensive & federated map of global bathymetry. Downside – data only available in raw form, no processed grids. No spatial analysis. Data is uncleaned in parts (particularly where raw data has been gridded directly)	https://www.ncei.noaa.gov/maps/iho_dcdb/
NONNA Viewer	Canadian Hydrographic Service	Canada	Yes	Yes	No	Comprehensive map of Canadian bathymetry. Includes cleaned data where IHO does not. Good WMS, but the portal is clunky & simple. WMS layers that are available are NONNA-10 and NONNA-100 (resolutions of 10x10m and 100x100m, respectively).	https://data.chs-shc.ca/map
IEDA Data Browser / GMRT MAP TOOL	MGDS (Marine Geoscience Data System). Part of the Interdisciplinary Earth Data Alliance (IEDA).	Global	Yes	No	No	Map of global bathymetry. WMS is interpolated with GEBCO to form a continuous basemap. Allows for the selection & download of grids.	http://app.iedadata.org/databrowser/
EMOD Bathy	European Marine Observation and Data Network	Global, but Euro-centric	Yes	Yes	Yes	Map of European Arctic bathymetry, interpolated with GEBCO to form continuous basemap Many WMS layers to choose from across many oceanographic disciplines.	https://portal.emodnet-bathymetry.eu/
Arctic SDI	Collaboration between many national mapping agencies	Arctic	Yes	No	No	Not ocean-mapping specific. Many datasets from related Arctic science disciplines are included. Only in polar stereographic view. Provides many spatial analysis tools, but GUI is slightly overwhelming & unfocused.	https://geoportal.arctic-sdi.org/
UNH Bathy Globe	UNH Center for Coastal and Ocean Mapping	Global, but US-centric	No	No	No	Visual tool only, no data available for download. Very visually appealing, concise & layman-friendly though.	http://com.unh.edu/project/bathymetry-globe
Mareano	Institute of Marine Research, Geological Survey and Norwegian Mapping Authority	Norwegian Sea	Yes	No	Yes	Norwegian bathymetry & oceanography tool. No Canadian data. Visually overwhelming, but very comprehensive. Imitates a desktop GIS app interface.	http://mareano.no/en/maps/mareano/en.html?language=en

In addition to the WMS layers listed above, many of the currently available Arctic bathymetry viewers also include the standardized basemap of GEBCO 2023, or its Arctic subsection IBCAO (the International Bathymetric Chart of the Arctic Ocean). GEBCO is a continuous global bathymetric grid. It was compiled through international cooperation between dozens of national hydrographic organizations. As such, its Arctic subset IBCAO represents the most complete bathymetric dataset currently available of the Arctic oceans (Jakobsson et al., 2020).

Although *Table 1* demonstrates that several options are currently available for viewing Arctic bathymetry data, few such tools exist for viewing or manipulating less common forms of seabed data. In particular, the author has found no existing interactive web map which displays sub-bottom profiles. This uncommon form of seabed data makes UNB OMG's Arctic Basemaps Series a unique set of web maps.

3.2 Web Mapping Architecture

Geographical databases provide geospatial data to geographical servers (geospatial servers, or GIS servers), which then communicate with the user through a series of standardized protocols to deliver the desired data. These geospatial servers can be provided by a proprietary commercial vendor, such as ESRI, or can be distributed through an open-source equivalent such as MapServer, or the eponymously named GeoServer (Open Source Geospatial Foundation, 2023). Open-source backend management of geospatial databases can be provided through PostgreSQL with a PostGIS extension (Ruiz, 2018).

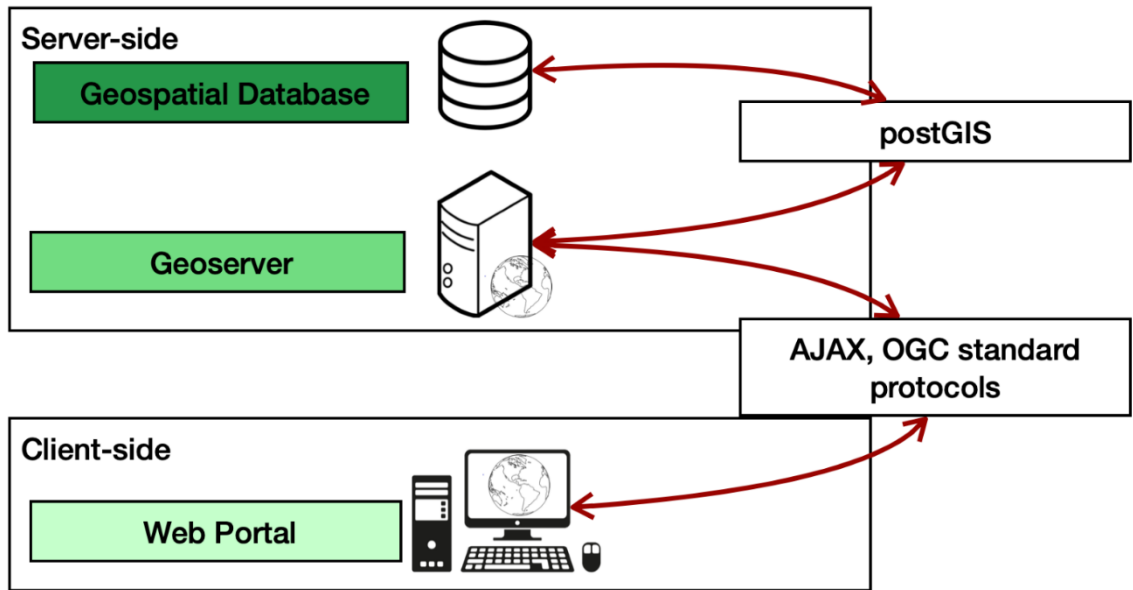


Figure 6 – Basic Web Mapping Architecture (modified from Ruiz, 2018).

3.2.1 GeoServer Back-End

This project makes use of datasets from both third-party geospatial servers (of varied brands), as well as hosting UNB OMG’s own geospatial data on a local geospatial server. The hosting software that was chosen for this task is the open-source GeoServer (Open Source Geospatial Foundation, 2023). GeoServer was chosen because Ruiz (2018) had previously laid out a web mapping framework that employs this software. GeoServer is ideal for a legacy project with multiple iterations because it has a large inbuilt userbase that provides online support. This online userbase also develops additional extensions to increase the software’s base functionality. An example of one such extension is discussed further in Section 3.3.1.

When it comes to the technical specifics of serving data to a client, the brand of the geospatial server bears minimal significance. All geospatial servers follow the

formatting regulations outlined by the Open Geospatial Consortium (OGC), which standardize web mapping protocols and services (GeoSolutions, 2023). As such, GeoServer provides all the services and requests listed in Section 3.2.2 below.

3.2.2 OGC Standard Protocols

OGC protocols consist of several different services and requests, outlined in *Figure 7*. Firstly, webmap (tile) services (WMS, or WMTS) display tiled raster/image data (GeoSolutions, 2023). Geospatial servers can also implement web feature services and web coverage services (WFS or WCS), which allow for the display and request of geographical feature information (GeoSolutions, 2023). Browsing capabilities are given under catalog services for web (CSW), and basic server-side transformation functions are given by web processing services (WPS) (GeoSolutions, 2023). Because of the OGC's standardized services and protocols, commercial and open-source geospatial servers perform most of the same functions and can be manipulated similarly.

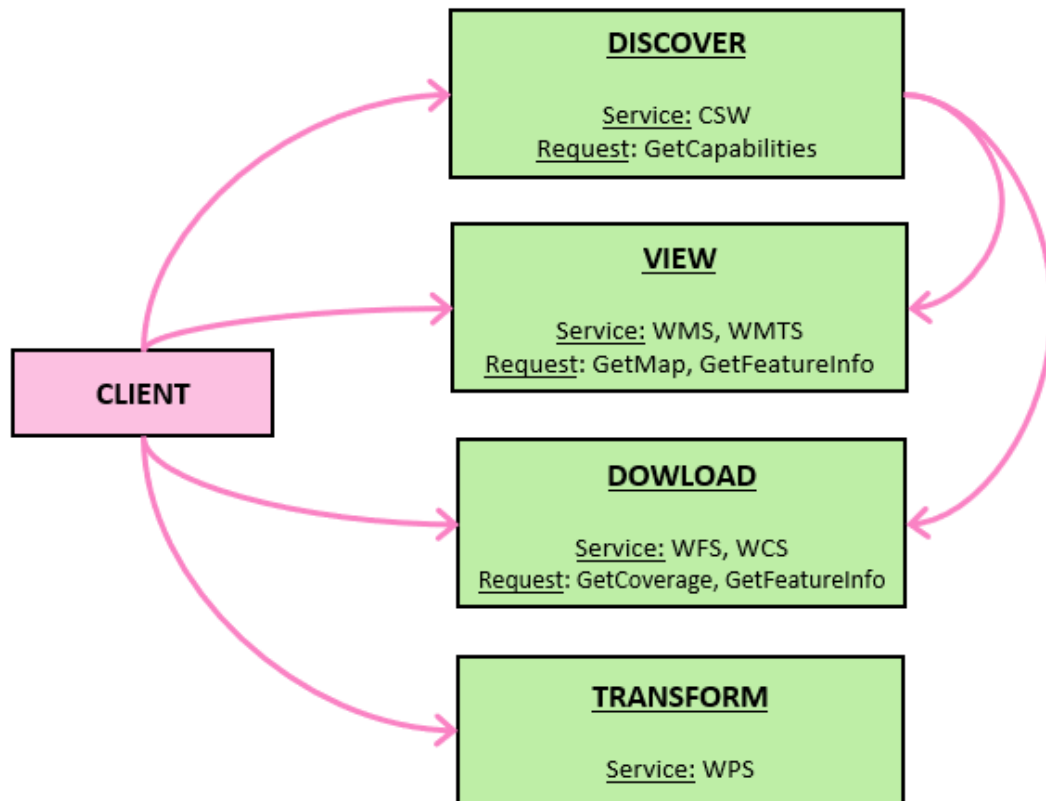


Figure 7 – OGC standard protocols (modified from Geosolutions, 2020).

3.2.3 Querying a Geospatial Server – XMLHttpRequests

Feature information can be queried from any public geospatial server that allows for CORS. The information is retrieved from the target geospatial server through the use of AJAX – Asynchronous Javascript and XML. AJAX is not actually a programming language itself, but rather it is a way of combining the languages of HTML, CSS, JS and XML together, along with DOM (Document Object Models) and XMLHttpRequests (XHRs) (Mozilla MDN, 2023). Together, this combination allows a webpage to be updated dynamically, without needing to refresh the page. An XHR is the object that is used to interact with the server in order to retrieve the requested information in situ

(Mozilla MDN, n.d). When this XHR follows along with the formatting guidelines outlined by the OGC, the information that is retrieved takes the form of GIS data. Several different formats can be requested for this output GIS data, depending on which formats the hosting party enables on the server-side. For example, if one were to perform a GetFeatureInfo request (a specific variant on a general XHR) directed at a geospatial server that is hosting bathymetric data, the output response returned could take the form of a geoJSON object, which contains the depth property as feature information.

Querying external geospatial servers through this method allows for the possibility of data federation within a web-mapping context. A federated data store is one that acts as a data repository but does not directly store data on its own servers. Rather, it scrapes the information as needed from third-party servers, such as those listed in *Table 1*. Hosting a WMS (Web Map Service) allows a mapping agency to serve their geospatial data openly to clients online (Open Geospatial Consortium, 2023). The mapping organization themselves might then provide a webmap interface where users can navigate through this data. Alternatively, third-party users from outside of the organization can overlay the WMS layers onto their own webmap. The WCS (Web Coverage Service) column of *Table 1* indicates that the user is able to retrieve the raw data points that were used to generate the raster WMS (Open Geospatial Consortium, 2023). This makes WCS an ideal format for modelling or analysis. The next column to the right indicates whether Cross-Origin Resource Sharing (CORS) is enabled, allowing a third-party to access the full capabilities of both WMS and WCS services (ESRI, 2023). Further details about WMS, WCS and CORS are discussed in Section 3.2.4.

In terms of raster imageservers (WMS/WMTS) specifically, this could look like retrieving the depth value at a pixel of a third-party bathymetry mosaic when a coordinate is given as an input. Doing this opens the possibility for performing fast, client-side raster math using the public bathymetry data collected and published by large hydrographic organizations. This in turn allows users to obtain meaningful analytic information about these different public datasets, and to compare them in quantitative ways without downloading data files. This concept of browser-based manipulation of third-party geospatial servers will act as the foundation for the spatial analysis tool discussed in Section 4.1.

3.2.4 Querying Limitations: CORS Policy

Cross-origin resource sharing (CORS) is a security precaution implemented by JavaScript to avoid allowing malicious scripts unfettered access to client-side resources (Launchcode Education, 2023). CORS is a JS-wide policy, rather than one that is specific to web mapping. Although the implementation of CORS helps keep user data secure on private websites; when developing an open-source and open-access website like the one hosting the Arctic Basemaps Series, CORS policy becomes limiting. To retrieve results from the querying method outlined in Section 3.2.3 above, the target geospatial server must have CORS enabled on the server side. This is already the default configuration for ESRI-based geospatial servers (ArcGIS Server, 2023), such as the one used by the IHO.

To enable CORS on a GeoServer-brand geospatial server, one must navigate into the WEB-INF subdirectory, and edit the web.xml file, uncommenting the lines which pertain to CORS (Launchcode Education, 2023). This task has already been performed on the GeoServer that was used to display the results of this project; however, if the

computer hosting the data were to change in the future, this workaround method would need to be implemented again.

Out of the third-party bathymetry data sources that are listed in *Table 1*, the only geospatial servers with CORS enabled are the ones provided by the IHO, EMOD and Mareno. This means that these are the only third-party bathymetry resources that are truly queryable at this time. As such, these geospatial servers have been used to provide example layers for many of the standalone code samples developed in Section 4.

3.2.5 Web Mapping Interfaces

All web mapping portals require a graphical user interface (GUI) in which users can interact with GIS datasets. These interfaces are written in JavaScript. Hosting parties who seek a subscription-based service have many options to choose from, such as ArcGIS Online, Mango Map, Carto and Mapbox. There are also open-source web mapping interfaces available, such as Leaflet, OpenLayers and OpenJUMP. The pros and cons of these popular services have been previously compared (Geoapify, 2019). This project has been developed using the Leaflet library, since it provides the largest support platform out of the open-source options, and was used in the previous foundational work discussed in Section 2.2. Leaflet includes basic functionality for display and analysis of spatial data (Ruiz, 2018), although it is focused on providing stylistic options rather than spatial analysis. User-developed plugins can expand Leaflet's capabilities in this regard.

3.3 Three-Dimensional Modelling in JavaScript

In addition to the two-dimensional web mapping interfaces listed above, there are also open-source options available for three-dimensional map display. Notably, Cesium.js is a JS plug-in designed for visualizing geographic data in 3D (Cesium GS, 2023). Its

capabilities do include the display of bathymetric rasters; however due to the unique nature of UNB's other data products, a 3D interface that allows for more manual customization was deemed better suited to this project's workflow. The scene modeling plug-in Three.js was thus chosen as the 3D interface for this project.

Three.js draws 3D shapes and surfaces within the browser using the web graphics library or WebGL (ThreeJS, 2023). Its base functions allow the end user to illuminate the scene using different lighting options, to control the camera panning behaviour using various orbit controls, and to render both point clouds and geometric mesh models. By manipulating a basic plane mesh, more complex mesh structures (such as seabed topography) can be generated. After a mesh is generated, an image texture can be overlaid onto it.

3.3.1 Converting GeoServer Output to JS-Parseable Format

The input for three.js must be in the form of a plaintext file since parsing raster input is not a task natively supported by JavaScript. The default file output types provided by GeoServer are all raster-based (for example, a tiff output). To retrieve data from GeoServer in a more appropriate format, a community-developed module needs to be added onto the base software. Before this can be done, the Geospatial Data Abstraction Library (GDAL) must be installed onto the computer that is meant to host the geospatial server (GDAL/OGR contributors, 2020). It must also be available within the PATH environment variable. The GDAL library provides a wide variety of geospatial functions, but the specific function that is required for this project is the `gdal_translate` function (which converts raster data into different formats).

The community-built extension that is needed to convert the GeoServer output data into text format is the “GDAL based WCS output” extension (GeoSolutions, 2020). This extension adds several new output formats, one of which is XYZ format. When enabled, a user can perform a query of the GeoServer’s WCS using the method outlined in 3.2.3., and request that the results of that query be returned in the form of a plaintext XYZ file. This file can then be passed along to three.js as input for mesh generation.

3.3.2 Converting Sub-Bottom Profiles Into JS-Parseable Format

With regards to using ThreeJS to display sub-bottom profile data, there are two ways in which this could be feasibly accomplished.

The first potential method is that the png images that UNB’s OMG has already pre-generated for the ArcticNet web suite could be projected onto a vertical surface. An example of this is shown below in *Figure 8*. A horizontal black surface is used to show the vertical orientation of the profile.

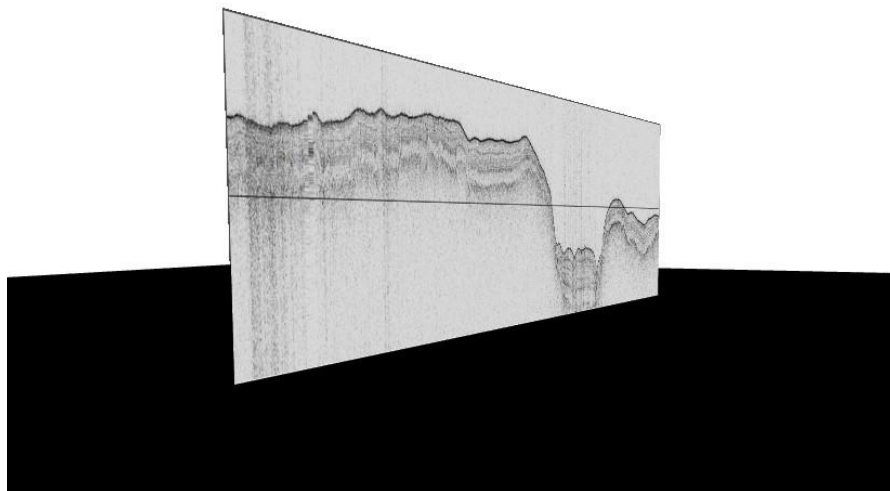


Figure 8 – Sub-bottom profile png image projected onto vertical curtain in ThreeJS

This is not an ideal solution because the images are not very high resolution. The exact intensity values are not made available to the user. The images also contain extra borders, titles and scalebars which need to be clipped out. There is another issue with the fact that several of these png images contain “stacked” data. That is to say that in some areas, data that was not collected in the same location was forced together horizontally to generate a single png image. The result of this is a sub-bottom profile with few discernable interfaces in the subsurface. An example of such a “stacked” profile is shown in *Figure 9* below. This effect is generated from the sub-bottom data files being projected to the back of a 25 km by 5 km box, to allow for a line with turns to be projected onto a plane.

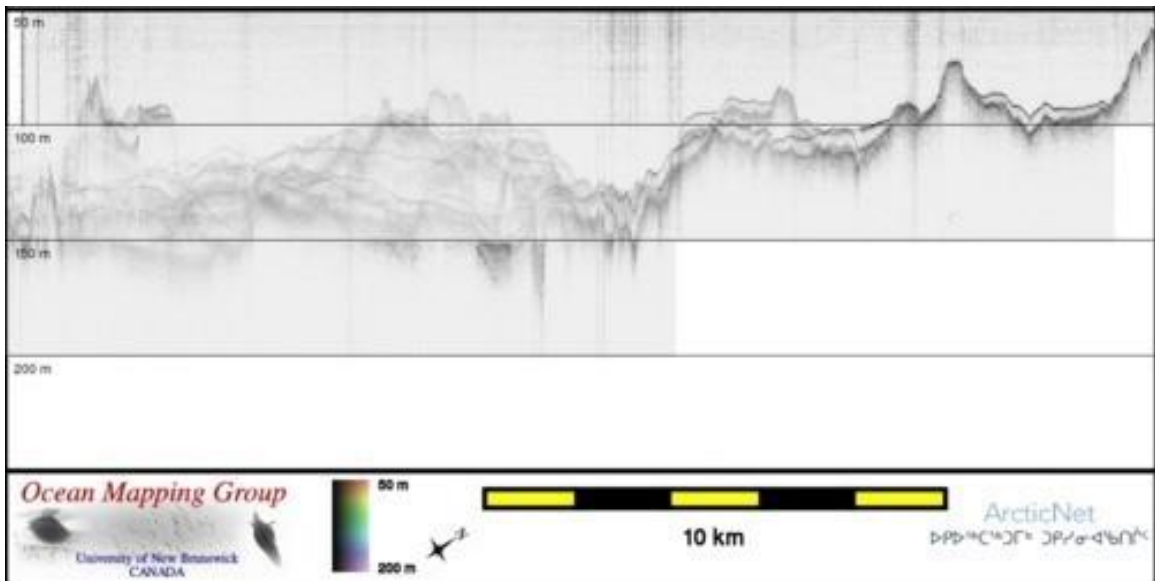


Figure 9 – Overlapping sub-bottom data due to stacking of multiple profiles. (UNB, 2014)

The second potential method involves reverting to the original sub-bottom data that was used as input when generating the profiles. The files should be reprocessed and converted into a more appropriate format. The target format is plaintext. The original file format is .keb, which is a proprietary file format released by the profiling system manufacturer, Knudsen. A single example profile was converted into a comma separated value (CSV) file for this project to demonstrate proof-of-concept.

4. METHODOLOGY

The major tasks of the methodology were twofold:

- TASK 1: Spatial Analysis Tool
- TASK 2: Three-Dimensional Seabed Viewer

Objectives 1 and 2 (outlined in Section 1.1) have been accomplished through the corresponding methodology tasks 1 and 2, respectively.

The sub-tasks associated with these tasks have been catalogued in a browser-based format in the Arctic Seabed Web Mapping Guidebook (ASWMG). The ASWMG is a website developed alongside this project which hosts the collection of standalone examples outlined in the methodology below. All of these standalone webpages contain heavy inline commentary on their source code, in an attempt to make them as instructional as possible. The purpose of this website is to provide templates and examples, so that future students can expand on the work presented in this thesis, adapting the methods employed to suit their needs.

4.1 TASK 1: Spatial Analysis Tool

By combining open-source JS plug-ins together, a webmap can emulate within the browser some of the tools and functions that are found in a desktop GIS environment. The sample tool that was developed for this thesis demonstrates this concept by providing a statistical diagnostic window for QA/QC of a bathymetric WMS. This tool was developed using the specific JS plug-ins listed in Section 4.4.

4.1.1 Step 1 : Single Point Query

The preliminary step in developing the statistical analysis tool was to create a base function to fire a single query at a geospatial server given a point on the map. The user-selected point was captured using a click listener bound to a Leaflet map. The results from this click event (the lat-long coordinate of the clicked point) were then passed on to an XMLHttpRequest. This request followed the OGC standard formatting that was laid out in Section 3.2.3. Since this is a query of a WMS layer, a GetFeatureInfo request was used. Since the displayed WMS layer is a bathymetry tileset, the feature property of that layer corresponds to the depth to the seafloor at that geographic coordinate. The most relevant portion of this code (parsing the XMLHttpRequest) is shown below:

```
map.addEventListener('click', function {
    getFeatureInfo
});

function getFeatureInfo {
    var url = this.getFeatureInfoUrl(e.latlng),
    //the function called above creates the request URL
    showResults = L.Util.bind(this);
    $.ajax({
        url: url,
        success: function (data, status, xhr) {
            var depth =
JSON.parse(data.features[0].properties.GRAY_INDEX);
            //creates a popup with the depth info
            alert('Depth: ' + depth + 'm');
        },
    });
}
```

The bathymetric WMS used in this example (hosted on EMOD's geospatial servers) has its depth feature property stored in a single raster band called "GRAY_INDEX". GRAY_INDEX is a common feature name because it is the default property name given to hillshaded rasters when they are exported via a desktop GIS application.

The accompanying example for this step in the ASWMG is number 3. Examples 1 and 2 are focused on introducing the user to the core capabilities of webmap display using Leaflet. The point-querying method outlined above (and shown in example 3) is valid for any geospatial server; however a JS plug-in exists to simplify requests fired towards ESRI-brand geospatial servers in particular. This plug-in is called `esri.js` and an example of how to use it is shown in ASWMG example 4.

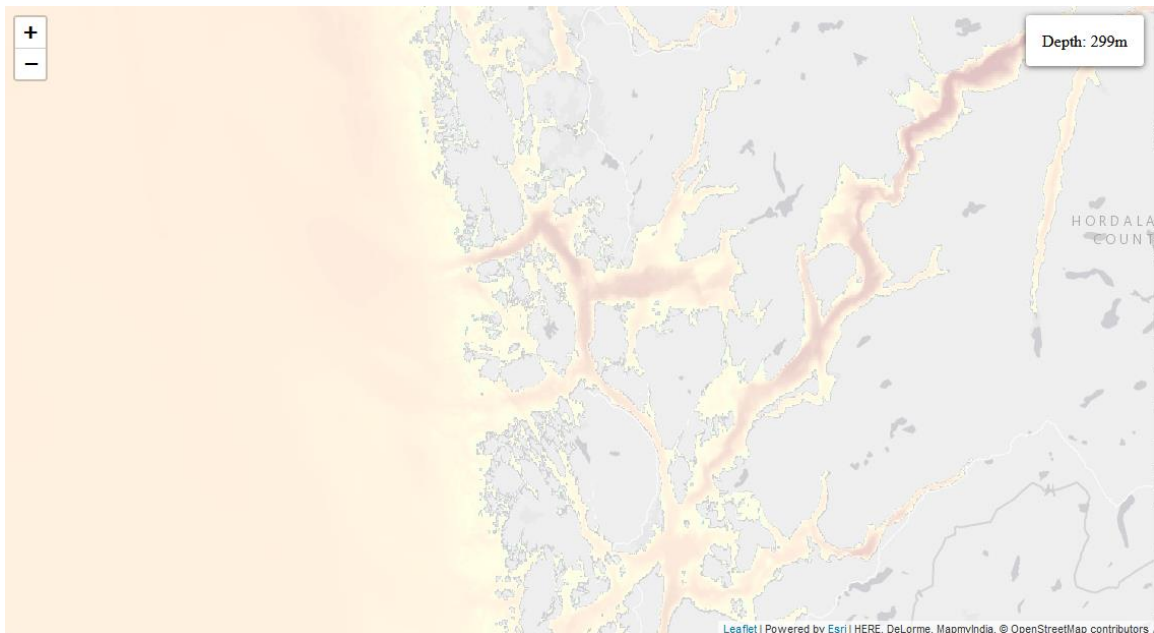


Figure 10 – The results of a point and click query shown in the upper right. The input bathymetry layer is from EMOD’s geospatial servers.

4.1.2 Step 2: Transform Areal Selection Into geoJSON Object

The next step in developing a QA/QC tool was to expand the point query into multiple queries – requesting results from an areal selection of points rather than a single lat-long point. This was done by incorporating the functions of additional open-source JS plugins. The plugin `turf.js` was used to generate a custom polygon on the webmap, in this

case a rectangle for simplicity. That shape was expressed as a geoJSON object. GeoJSON is a way of expressing a geographical shape (such as a shapefile, a linepath, or a polygon) in a form that Javascript can parse easily (plaintext). The geoJSON object that represents the rectangle is then used as a bounding area (bbox) for subsequent functions.

4.1.3 Step 3: Query Areal Selection

Using the shape generated in step 2 above, a for loop was then used to cycle through every pixel within an evenly spaced grid constrained within the bounds of that shape. The points were in this case arbitrarily selected to be spaced 0.02 degrees apart in each of the latitude and longitude directions.

Each point within the grid was used as the lat-long input for a separate query (following the querying procedure in Step 1). The results of these many queries (the depth values within the circle) were then stored as a text array. This array was assigned a variable (“points”) so that it could be recalled later.

The framework of the aforementioned for loop is shown in the code snippet below:

```
var points = [];  
  
for (let x = circle.getBounds().getWest(); x <=  
circle.getBounds().getEast(); x += 0.02) {  
  for (let y = circle.getBounds().getSouth(); y <=  
circle.getBounds().getNorth(); y += 0.02) {  
    wmslayer.identify().at([y,x]).run(function (error,  
IdentifiedPixel) {  
      if (error) { return };  
  
      var bathyvalue =  
Number(IdentifiedPixel.pixel.properties.value);  
      if (isNaN(bathyvalue)) {
```

```

        return;
    } else {
        points.push(bathyvalue);
    }
}
}

```

After the depth values were stored, that array was then called on to use as the input dataset for various statistical functions. In particular, the plugin chart.js was used to generate a histogram of the given depth values. The histogram is placed in a div box, which was overlaid onto the map. This can be seen in *Figure 11* below. Note the gap in the histogram, indicating that there is potentially an unnatural artifact in the data at the selected location. The accompanying example for this step in the ASWMG is example number 5.

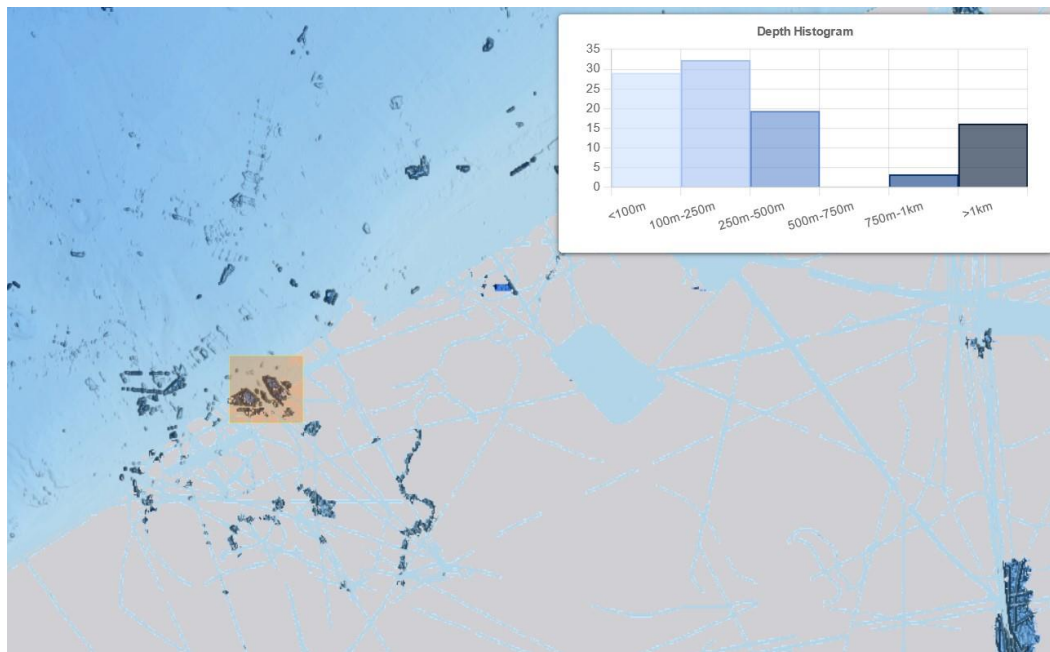


Figure 11 – Histogram generated from querying the region bounded by the orange rectangle. Bathymetry layer provided by the IHO’s public WMS.

4.1.4 Step 4: Query Multiple Layers

The XMLHttpRequest method that was used to query point information from a geospatial server (detailed in Section 4.1.1) can be employed for not just one geospatial server, but multiple geospatial servers at the same time. Since these queries are formatted as AJAX call functions, they can be performed dynamically in situ, without the need to refresh the webpage.

An example query of two different geospatial servers using the same input coordinate is shown below in *Figure 12*. The desired lat-long coordinate was logged using a click listener at a user-selected point on the map. The two XMLHttpRequests were then sent to the two separate geospatial servers, both requests following the OGC standard formatting for a GetFeatureInfo request. The result of both of these requests is then shown in the div box to the right of the webmap.

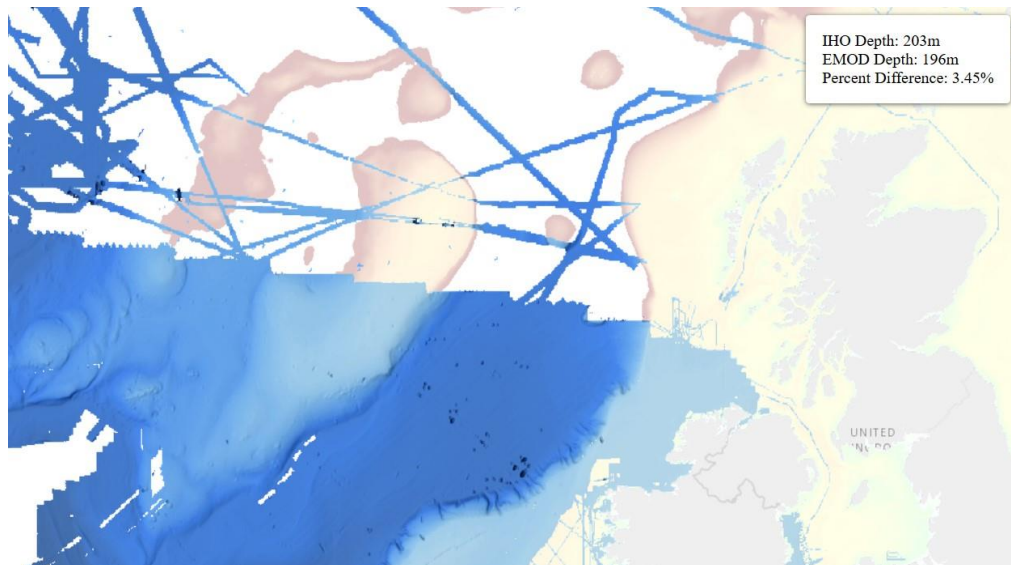


Figure 12 – Single point query of 2 layers. The input bathymetry layers are WMS layers from the IHO's and EMOD's geospatial servers.

The accompanying example for this step in the ASWVG is example number 6.

4.2 TASK 2: Three-Dimensional Seabed Viewer

The spatial nature of seabed data makes it easy to communicate and absorb in a visual format. Using the three.js plugin, a new spatial dimension can be added onto the Arctic basemaps Series.

4.2.1 Step 1: Topographic Mesh Generation Using Three.JS and WCS Input

Using the basic plane creation function provided by three.js, a plane can be subdivided into a triangular polygon mesh with a specified number, width, and height of segments. The plane displayed in *Figure 13* was generated using the command `THREE.PlaneGeometry(60, 60, 9, 9)`. The arguments represent, in order, the width of the plane segments, the height of the plane segments, and the number of x and y segments. The values assigned to each of these arguments (60, 60, 9 and 9) are arbitrary, and were chosen in an attempt to create a plane sizeable enough for its segments to be seen easily.

The accompanying example in the ASWVG is number 9.

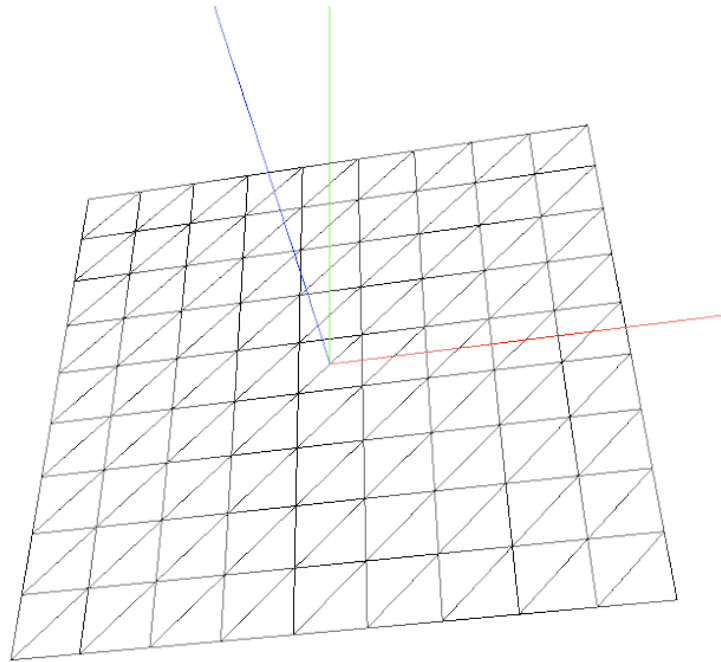


Figure 13 - Plane subdivided into triangular mesh. The coloured lines represent the axes of the 3D plot.

Three.js stores the xyz coordinates of the vertices of this plane as an attribute of the plane's BufferGeometry object. This vertex attribute takes the form of an array called `geometry.vertices`. This array can be called upon using a for loop which loops through each vertex coordinate of the flat plane, then changes its z elevation value into the corresponding value taken from an input xyz bathymetry file. An example of such a function is shown below.

```
for (var i = 0, l = geometry.vertices.length; i < l; i++) {  
    geometry.vertices[i].z = data[i];  
}
```

The input bathymetry file was generated by sending a WCS request to a geospatial server. In order to generate the appropriate WCS output format (xyz) when

using a GeoServer brand geospatial server, a user-developed add-on had to be employed, as outlined in Section 3.3.1.

An example of the input and output of the function above is shown in *Figure 14* below. The wireframe mesh model of the seafloor topography shown on the right was generated from the Arctic Basemaps Series' bathymetry dataset on the left, for an area in the Beaufort Sea. The exact coordinates of the South Eastern corner are 71°15'N, 136°00'W.

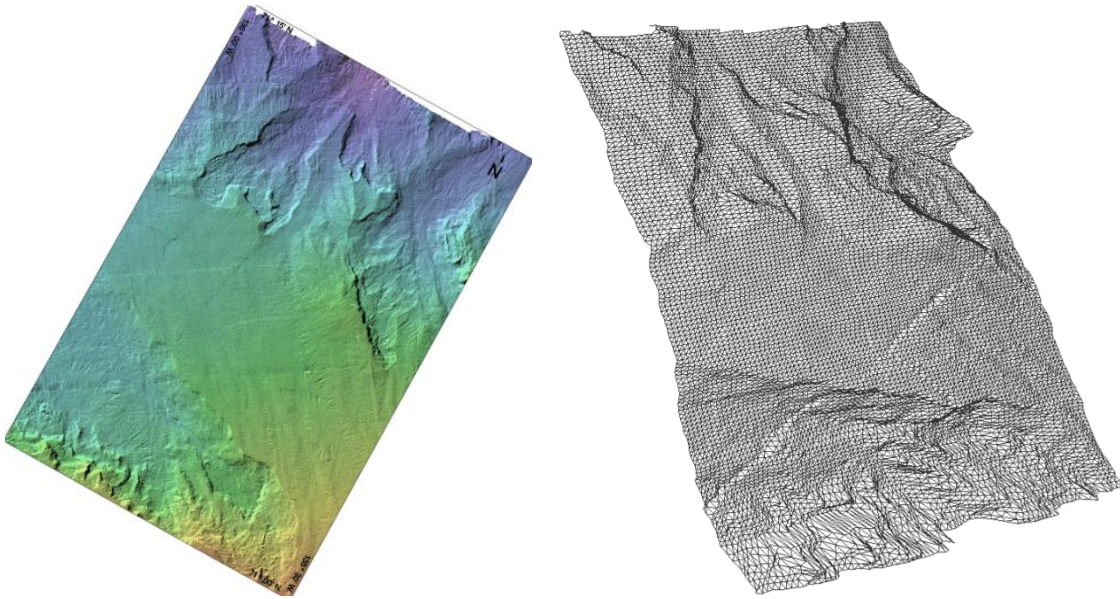


Figure 14 - Mesh generated in three JS by modifying a plane. Data taken from the ArcticNet Basemaps Series (UNB, 2014).

The accompanying example on the ASWVG is number 10. Since UNB's bathymetric geospatial server does not support WCS requests, the example shown in Figure 14 was generated using a local geospatial server to host the same data. Since this local server cannot be accessed online by the ASWVG, the online example in the ASWVG instead uses a WCS supplied by EMOD.

It is important to note that a certain amount of spatial distortion is present in the mesh due to discrepancies in geographic projection. ThreeJS will accept input coordinates from any geographic datum, since the points are represented on an equidistant mathematical grid. Any reprojection of coordinates must therefore be handled before the data is inputted to ThreeJS by modifying the parameters of the WCS request URL.

4.2.2 Step 2: Overlaying Image Data onto Bathymetric Wireframe

Rather than leaving this mesh as a wireframe, its surface can be overlaid with imagery (also a function of ThreeJS). This imagery can take the form of a single image file, or a clipped image output from a WMS raster tileset. Relevant overlays in a web GIS context could be satellite imagery, surface salinity etc.

After the topography of the seafloor was turned into a mesh model in Step 1, the wireframe was overlaid using raster imagery data, which was also hosted on GeoServer as a WMS. The raster dataset that was used was the backscatter data from the Arctic Basemaps Series. By overlaying this dataset, the end user is not only able to conceptualize the topography of the seafloor in 3D, but they are also able to get a feel for the probable sedimentology of the surficial sediments at the same time, visualizing the relationship between them.

An example code that overlays a geotiff clipped from a WMS is shown in the code snippet below.

```
var material = new THREE.MeshPhongMaterial({  
  
    map: THREE.ImageUtils.loadTexture(wmsURL),
```

```
});
```

```
var seafloor = new THREE.Mesh(geometry, material);
```

The results of this overlay are shown in *Figure 15* below. This example is from the same region of the Beaufort Sea as the wireframe example shown in *Figure 14*. The results provide a similar effect as the local scene layers function of the proprietary desktop-based ArcGIS.

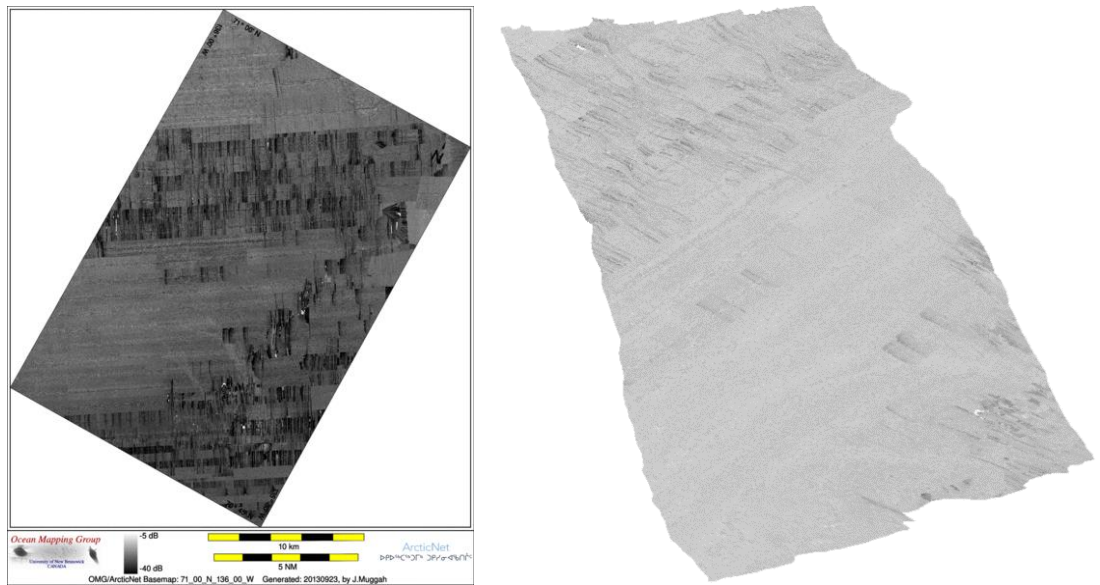


Figure 15 – Backscatter imagery overlaid onto bathymetric wireframe.

The accompanying example in the ASWMG is number 11.

4.2.3 Step 3: Navigating 3D Viewer Using 2D Map

Step 1 and 2 of Task 2 demonstrated how to create a topographic mesh and an image overlay for a single, predetermined area. In order to give the user the ability to determine the spatial extent of the data displayed in the 3D viewer, the user must be able to select their desired area on a 2D map.

This was done by creating two div boxes – one for the Leaflet map, and the other for the ThreeJS WebGL scene. A click listener was registered on the map portion. Both the up and down clicks were logged and were used to draw the upper and lower corners of a rectangle onto the 2D map (ie. The user drags a box onto the map). The box's corners were stored as variables – coordinates with a projection that matches that of the map. These coordinates were then translated using the JS plug-in proj4js. They were translated into the same projection as that of the target geospatial server hosting the data. Care was also taken to ensure that the coordinates were in the proper order. Leaflet by default uses a coordinate order of (lat, long), while most geospatial servers accept an input of (long, lat).

Once the appropriate conversion was made, a WCS request was then sent to the target geospatial server, and the results were plotted as a topographic mesh following the method outlined in Step 1. If the user drags a new region onto the map, the bounding box (bbox) variable changes, updating the 3D scene with a mesh for the new region. A sample of the selection window is shown below in *Figure 16*.

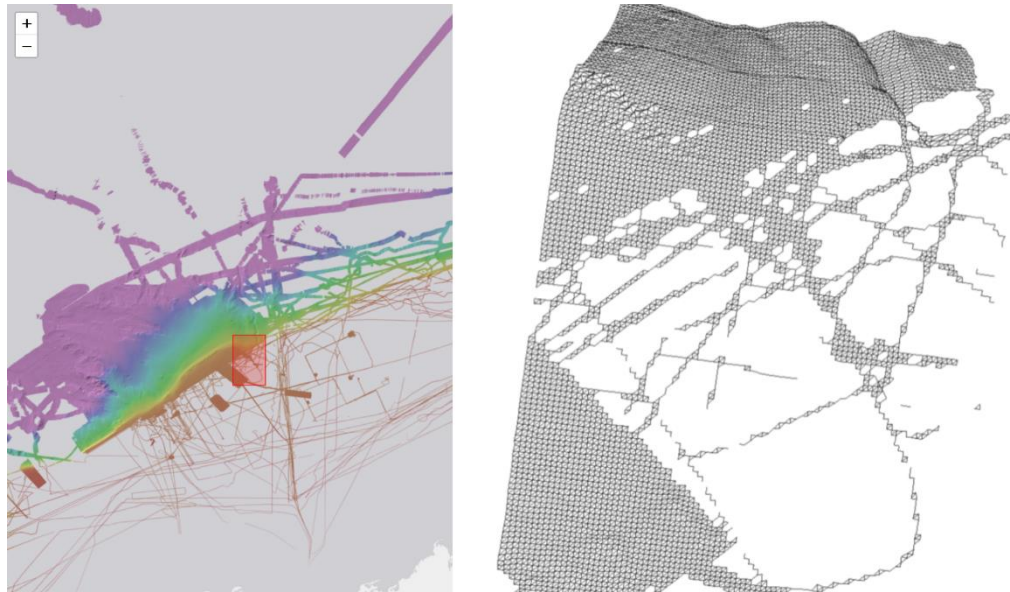


Figure 16 – Three-dimensional topographic mesh generated from a user-selected bbox (red) on the 2D map.

The accompanying example in the ASWVG is number 12.

4.2.4 Step 4: Creating 3D Point Cloud from Sub-Bottom CSV

In addition to the seafloor topography, the sub-bottom profiles were also represented in 3D. Since this data is cross-sectional, they were represented as a point cloud. The sub-bottom profile intensities were used to provide the colour values for each point.

Recall how Step 1 (the topographic mesh generation) used a for loop to cycle through each vertex, modifying its z attribute. The same method can be used to cycle through the colour attribute of each point, changing its value based on the intensity of the sub-bottom data at that point. An example of this type of for loop is shown below.

```
for (let i = 0; i < (numofpoints); i ++) {
    normalizedintensity = subbdata.intensity / 255;
```

```
color.setHSL( 0.3*(1-normalizedintensity), 1, 0.5 );  
}
```

The HSL in the code snippet above stands for Hue, Saturation, Lightness. The three arguments that follow represent those 3 colour properties, scaled between 0 and 1. The first argument, the hue, is what is being used to distinguish high intensity areas from low ones. It is assigned by normalizing the intensity values of the sub-bottom data based on the minimum and maximum hue range, 0 to 1, then multiplying by a slight reddening factor of 0.3. The second argument of 1 assigns full colour saturation. The final argument is the lightness, which is assigned a 0.5 for half brightness.

The result of this recolouring is shown in *Figure 17* below.

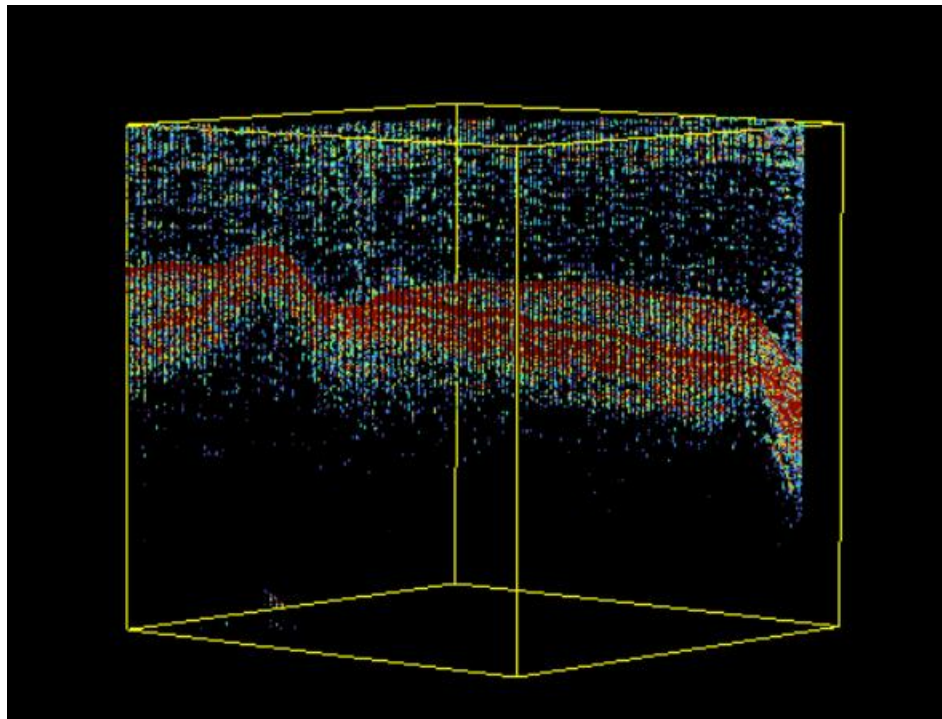


Figure 17 - Sub-bottom input file represented in ThreeJS as a point cloud. Red pixels represent points of higher intensity return.

The accompanying example in the ASWMG for this step is number 13; however because the sub-bottom input file is too large to host on github, only the raw code is provided. The example above was generated by hosting the data file locally, as outlined below in Section 4.3.

4.3. Setting Up a Local Geospatial Server

The website developed for the purpose of this research project uses a local server to store and access data. This step is not necessary if one is accessing data hosted by third parties, like the IHO. Hosting on a local server was necessary for this project because the full extent of UNB's Arctic Basemaps dataset is not currently hosted on a suitable geospatial server (one which is CORS enabled with XYZ text output available). Instead, single bathymetry and backscatter grid files were downloaded, converted to geotiffs, and hosted on a local Geoserver. Sub-bottom files were converted from keb files to csv files and were hosted on a local server using the python command below.

```
python -m http.server
```

The server is by default set to port 8000, ie. The Uniform Resource Locator (URL) used to access the website is <http://localhost:8000>.

4.4. Javascript Plug-Ins Used

The creation of the two toolkits outlined above would not have been possible without a handful of open-source JS plug-ins. Below is a complete list of the third-party

plug-ins that were used over the course of this project, along with a brief description of each of their functions.

Table 2 – Third-Party JS Plug-ins Employed

Leaflet.js	A plug-in for basic web map display
Sidebar.js	A Leaflet add-on that provides a sidebar GUI
ESRI.js	A Leaflet add-on for simplifying the handling of ESRI-brand geospatial servers
Turf.js	A spatial analysis plug-in
Chart.js	A plug-in for drawing basic 2D graphs
JQuery	A plug-in allowing easier handling of XMLHttpRequests
Proj4JS	A plug-in for converting between geographic projections
ThreeJS	A plug-in for 3D modelling, which draws scenes using WebGL

5. RESULTS

The results of the three major tasks that were outlined in the methodology are presented below, under the same headings.

5.1 TASK 1: Spatial Analysis Tool

The spatial selection tool developed in Task 1 of the methodology is shown below in *Figure 18*.

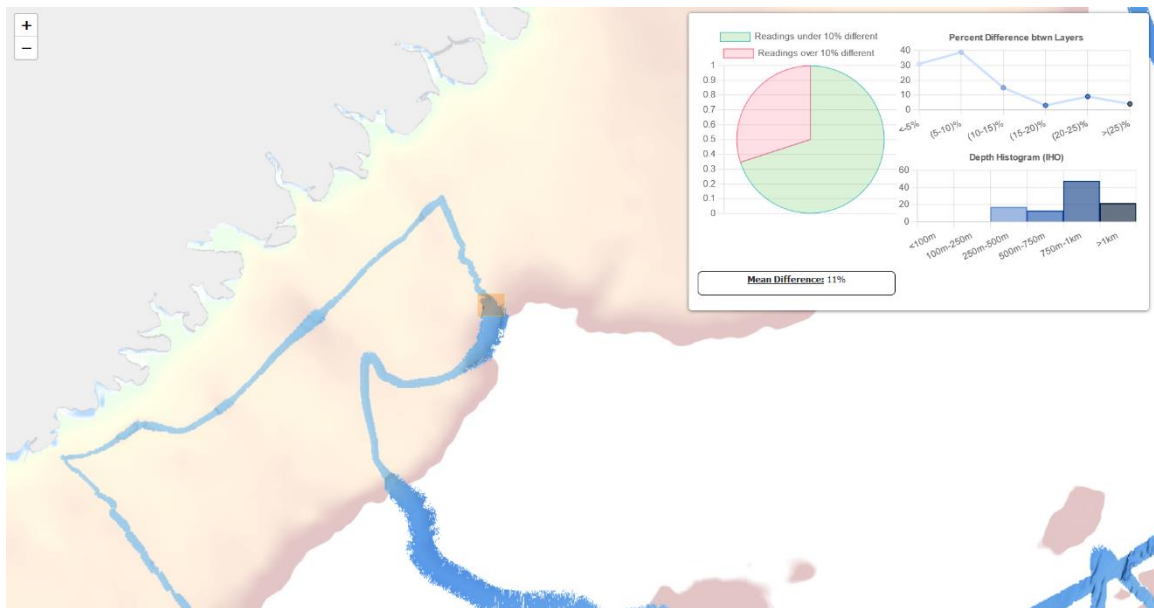


Figure 18 – Bathymetry QA/QC Diagnostic Window. The WMS layers used for sampling are supplied by the IHO and the EMOD.

The final diagnostic window builds on the previously queried information (depth values from a bathymetric WMS) developed in the methodology section and adds in a histogram of these depths. The diagnostic window also incorporates a second bathymetric dataset in order to compare the percentage difference between the two data sources. The percent difference is displayed in two forms - as a histogram distribution and as a pie chart which distinguishes how many of the measurements are within an arbitrarily pre-

defined acceptable limit (which in this case was set at 10%). The result is a tool that allows the user to compare the integrity of two datasets, to identify areas where there are major discrepancies, or areas with unreliable data.

5.2 TASK 2: Three-Dimensional Seabed Viewer

The 3D viewer developed in Step 2 of the methodology first instructs the end-user to make a selection on the 2D map. This selection area then acts as the bounding box for a WCS query towards the geospatial server that is hosting the target bathymetry data. The results of this query are used to generate a mesh surface of the seafloor topography. This mesh surface of the seafloor is represented as a wireframe, which can be coloured and hill-shaded, or overlaid with backscatter data in the form of a raster tileset. Sub-bottom data is displayed in the same scene as a point cloud. The colours of this point cloud reflect the intensity of the sub-bottom return at that point.

An example of a 3D scene generated from this viewer is shown below in *Figure 19*.

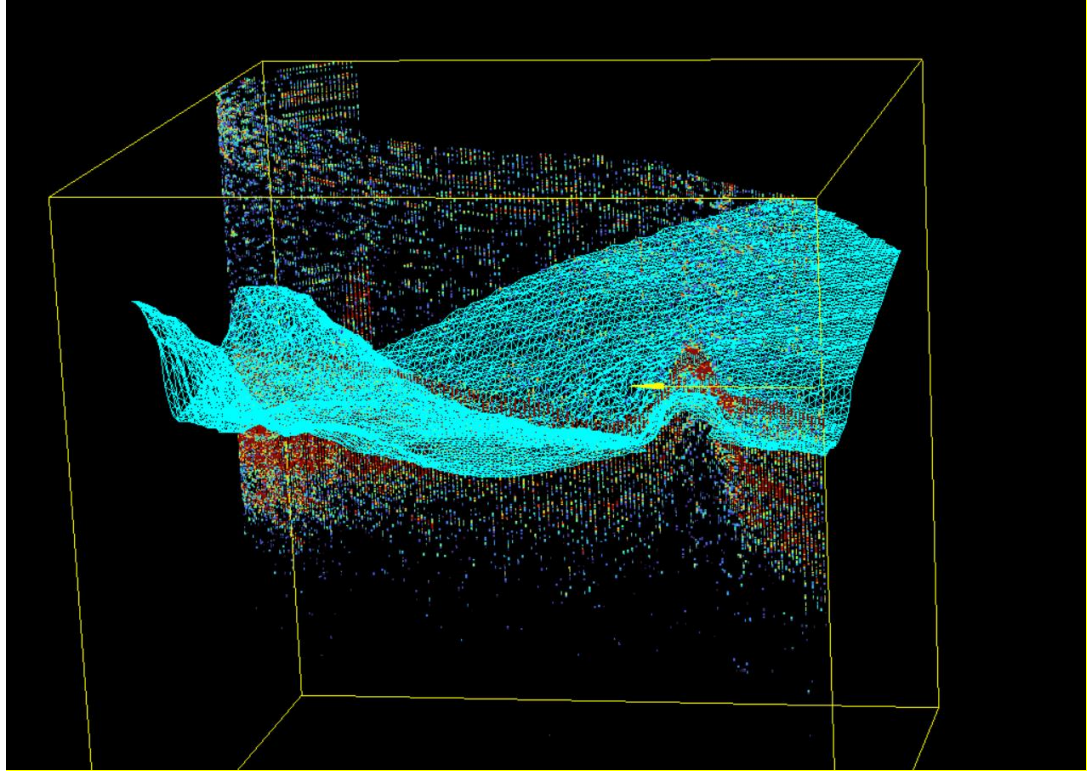


Figure 19 – 3D seabed scene. Seafloor topography is represented as a cyan wireframe, and sub-bottom intensities by a point cloud.

6. CONCLUSIONS

To revisit the overall goal of increasing the interactivity of Arctic bathymetry web resources, a portion of that goal is accomplished by adding the ability of the end user to select their own area of interest rather than forcing them to choose from pre-determined bounds. Adding onto that spatial querying and three-dimensional visualization increases the options that an end user has to manipulate the dataset.

Revisiting the objectives introduced in Section 1.1 of this research project, the first objective of creating a spatial analysis tool was achieved through Task 1 of the methodology, while the second objective of creating a 3D visualization tool was achieved through Task 2. These toolkits can be served either to the public via an online webpage, or remotely through a local webpage (in use cases such as aboard the CCGS Amundsen). In both cases, the displayed data must be hosted on a CORS-enabled geospatial server. An additional caveat to the 3D visualization tool is that the hosting geospatial server should also be able to provide WCS output in XYZ format. None of the third-party geospatial servers discussed in Section 3.1 fulfill these two criteria, although some fulfill one. As such, a local website was used to demonstrate proof-of-concept for most of this project. It is unlikely that these third-party sources will become CORS compliant in the future. It is therefore recommended that any third-party datasets used be downloaded and hosted on UNB's servers.

To address more of the specific problems with Arctic bathymetry web resources, the web resources listed in Section 3.1 are mostly visual display tools. Using the spatial analysis tool developed in Task 1, the end user can extract quantitative data values at the exact geographic coordinate or area of their choosing. Users are also able to retrieve

meaningful statistical information about the bathymetric data in their selected area, allowing them to QA/QC for things like noisy data, and to compare WMS layers.

Further issues with the previous method of data display were also highlighted with regards to the sub-bottom profiles. The previous iteration of the Arctic Basemaps Series represented sub-bottom profiles as non-georeferenced png images. This made it difficult to pinpoint the coordinates of target areas. By representing the profiles in ThreeJS, this issue is avoided by drawing the same data on a georeferenced point-by-point basis in WebGL. Not only does this method make it easier for the client to pick out the coordinates and depths of target areas, but it provides more versatility in how they choose to interact with the data. Users can clip out only the sections of the dataset that they are interested in rather than being confined to a pre-determined set of bounds. Users are also able to visualize the sub-bottom cross sections within the same scene as the 3D topography of the bathymetric WCS layer of their choice. With the raw intensity values now easily accessible and JS-parseable, there are more potential options available for visualization. For example, a threshold intensity value could be added to the pixel lightness to black out low intensity areas. This would effectively act as an intensity highpass filter. Representing the sub-bottom dataset in ThreeJS also rectifies the previous issue with the stripmaps “stacking” together by representing them in 3D space.

6.1 Limitations

As addressed earlier in this section, one limitation of browser-based mapping is that JavaScript is optimized for parsing plaintext input. The WCS bathymetry data that was used to generate the 3D seafloor topography must therefore be expressed in xyz format. This is done through a community-developed extension to the base GeoServer

software. Although the set up of this extension is straightforward, most third-party Arctic bathymetry geospatial servers are set up using the default file output formats only.

Another limitation of mapping within the browser is the implementation of CORS policy, as discussed in Section 3.2.4. Disabling CORS is the default configuration for many geospatial servers, and as a result many third-party bathymetric WMS providers are also CORS-disabled. For this project's website to be accessed correctly, CORS must be enabled on the geospatial server which hosts the target dataset. The method for enabling CORS is outlined in the same section – 3.2.4.

Due to the limitations of CORS policy, both the spatial analysis and the 3D viewing tool are restricted in the third-party datasets that they are able to use as input. For future iterations of the Arctic Basemaps portal, it is recommended that the organizations responsible for maintaining these third-party bathymetry geospatial servers be contacted and asked to consider enabling CORS on their own servers. In particular, the dataset hosted by the CHS (NONNA) would be a very useful addition to both tools developed in this project.

6.2 Recommendations for Implementation on UNB Back-End Servers

Since this research project was centered around front-end development, the back-end input data (ie. UNB's WMS tileset, ESRI grid files and png images) were left unaltered. Where altered file formats were required, single files were converted to generate standalone examples. Further work is needed on the back-end of UNB's geospatial servers in order to update the complete dataset that is being served.

For the bathymetry and backscatter datasets, the WMS tileset should firstly be updated to include data collected past 2013. This layer will also need to be uploaded onto a local geospatial server aboard the CCGS Amundsen.

For the sub-bottom dataset, the raw keb files hosted by UNB will need to be translated into plaintext format. The seabed cross section that was presented in Section 4.2.4 was generated using data from a single stripmap that was translated into a CSV file as an example to provide proof-of-concept. It was also sub-sampled to 100 000 points in order to maintain a manageable file size. When it comes to displaying the entire sub-bottom dataset across the whole Canadian Arctic, it is recommended that the files be compiled into a geodatabase.

6.3 Recommendations for Future Work

The QA/QC diagnostic window developed in this project is only one example of the type of tool that can be created by combining various smaller open-source plug-ins together to accomplish a larger goal. Any tool that is currently provided in a desktop GIS environment could, in theory, be imitated within the browser.

Specifically, a welcome addition in the future would be a tool to filter a WMS raster layer spatially, and to polygonize “problem areas”. That is to say, the end user would specify some value that they would like to use to constrict the data. For example, one might want to highlight only the areas where the depth values are shallower than a certain threshold. Alternatively, one might want to highlight areas where two datasets are above a certain percentage difference. This could be done by generating a percent difference map, representing these targeted “problem areas” as geoJSON polygons and

displaying them on the map using an intuitive colour scheme (for example, red is associated with a negative quality, like a high percentage difference).

A previously created example of this type of “problem area” polygon map is the CHS’ Priority Planning Tool, or CPPTO (Chénier et al., 2018). This tool was created to categorize surveys in the CHS’ bathymetry database based on where they stand in the category zone of confidence (CATZOC) scale. *Figure 20* is taken from an article published by the CHS in 2018 which details the creation of this map; however, the WMS remains unreleased to the general public.

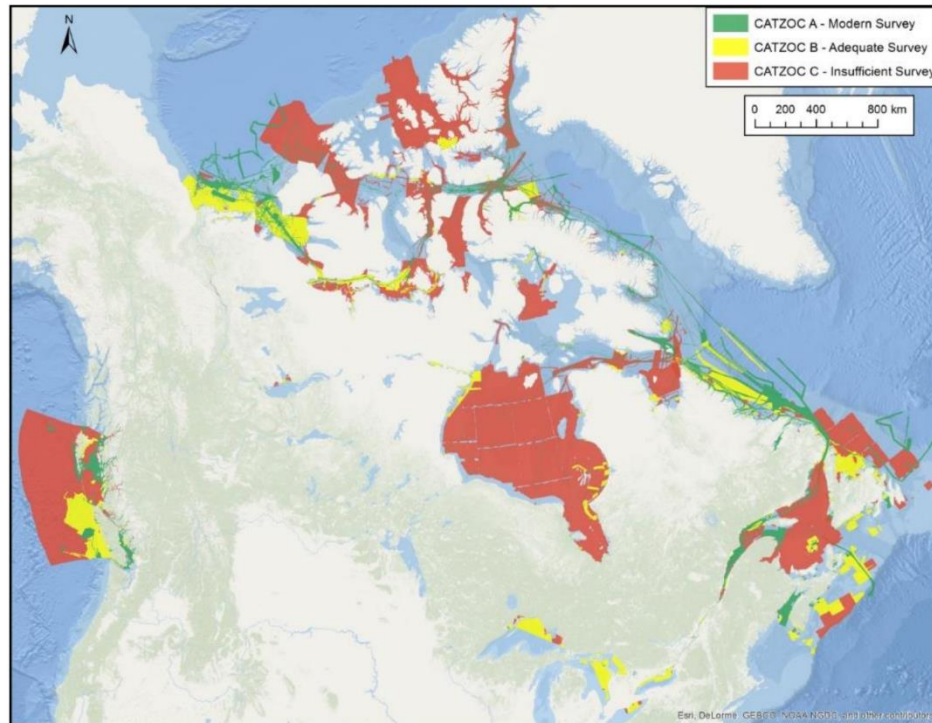


Figure 20 - Category zone of confidence (CATZOC) map presented in the CPPT (Chénier et al., 2018).

There are also some small tweaks that could help to optimize the 3D viewer portion of the web portal in the future. A request was made by a member of the Ocean

Mapping Group who had recently surveyed aboard the CCGS Amundsen. This request was to incorporate the capability for the end-user to click on areas of the 3D scene to place a marker. The marker could be expressed as a geoJSON object, which could then be exported into a desktop GIS program. Since the ship's navigation is done mostly using desktop GIS, this would improve shiplside communication between the surveying scientists and the navigational crew, helping them to expedite their requests for passage to target areas.

Bibliography

- Abulaitijiang, A., Andersen, O. B., & Sandwell, D. (2019). Improved Arctic Ocean Bathymetry Derived From DTU17 Gravity Model. *Earth and Space Science*, 6, 1336–1347. <https://doi.org/10.1029/2018EA000502>
- Agrawal, S., Gupta, R.D. (2017). Web GIS and its Architecture: a review. *Arabian Journal of Geosciences*, vol 10, 518-531. DOI:10.1007/s12517-017-3296-2
- ArcGIS Server. (2023). *Restrict cross-domain requests to ArcGIS Server*. Retrieved Jan 18, 2023 from <https://enterprise.arcgis.com/en/server/latest/administer/linux/restricting-cross-domain-requests-to-arcgis-server.htm>
- Beaudoin, J. H. C., Bartlett, J., Blasco, S., and Bennett, R. (2008). Mapping Canada's Arctic Seabed: Collaborative Survey Processing and Distribution Strategies. *Proceedings of the Canadian Hydrographic Conference and National Surveyors Conference 2008*.
- Cesium GS. (2023). *About Cesium.js*. Retrieved Feb 2, 2024 from <https://cesium.com/about/>
- Chénier, R., Abado, L. and Martin H. (2018). CHS Priority Planning Tool (CPPT)—A GIS Model for Defining Hydrographic Survey and Charting Priorities. *ISPRS International Journal of Geo-Information*. Vol 7, 204. DOI:10.3390/ijgi7070240
- CHS. (2023). *Canadian Hydrographic Service Non-Navigational (NONNA) Bathymetric Data*. Retrieved Feb 2, 2024 from <https://open.canada.ca/data/en/dataset/d3881c4c-650d-4070-bf9b-1e00aabf0a1d>
- Esri. (2023). *CORS with the API*. Retrieved December 11, 2023 from <https://developers.arcgis.com/javascript/latest/cors/>
- GDAL/OGR contributors. (2020). *GDAL/OGR Geospatial Data Abstraction software Library*. Retrieved from <https://gdal.org>
- Geoapify. (2019). *Leaflet vs OpenLayers. What to choose?.* Retrieved Dec 10, 2023 from <https://www.geoapify.com/leaflet-vs-openlayers>
- GeoSolutions. (2020). *GDAL based WCS Output Format*. Retrieved Oct 19, 2022 from <https://docs.geoserver.org/latest/en/user/community/gdal/index.html>
- GeoSolutions. (2020). *GeoServer Training*. Retrieved June 14, 2021, from https://docs.geoserver.geo-solutions.it/edu/en/ogc_protocol_intro/index.html

- GeoSolutions. (2020). *Introduction to OGC – GeoServer Training*. Retrieved Jan 18, 2023, from https://docs.geoserver.geosolutions.it/edu/en/ogc_protocol_intro/index.html
- IHO DCDB. (2024). *IHO Data Centre for Digital Bathymetry (DCDB)*. Retrieved Feb 2, 2024 from <https://www.ngdc.noaa.gov/iho/>
- Jakobsson, M., Mayer, L.A., Bringensparr, C. et al. (2020). The International Bathymetric Chart of the Arctic Ocean Version 4.0. *Sci Data* 7, 176. <https://doi.org/10.1038/s41597-020-0520-9>
- Jakobsson, Martin & Mayer, Larry & Monahan, David. (2015). Arctic Ocean Bathymetry: A Necessary Geospatial Framework. *ARCTIC*. 68. 7. 10.14430/arctic4451.
- LaunchCode Education. (2023). *Enabling CORS in GeoServer*. Retrieved Jan 16, 2023 from <https://launchcodeeducation.gitbook.io/using-geoserver-with-openlayers/rendering-data-in-openlayers/enabling-cors-in-geoserver>
- Mozilla MDN. (2023). *Ajax*. Retrieved Jan 18, 2023, from <https://developer.mozilla.org/en-US/docs/Web/Guide/AJAX>
- Mozilla MDN. (2023). *XMLHttpRequest*. Retrieved Jan 18, 2023, from <https://developer.mozilla.org/en-US/docs/Web/API/XMLHttpRequest>
- Muggah, J. (2010). *Developing a New Online Distribution Method for Multibeam Data*. [Unpublished M.Eng. Thesis]. University of New Brunswick.
- Muggah, J., Church, I., Beaudoin, J. and Hughes Clarke, J.E. 2010: Seamless online distribution of Amundsen multibeam data; Proceedings of the Canadian Hydrographic Conference, Paper S7.2.
- Open Geospatial Consortium. (2007). *OpenGIS Styled Layer Descriptor profile of the Webmap Service Implementation Specification*. Retrieved Nov 6, 2021, from <https://www.ogc.org/standards/sld>
- Open Geospatial Consortium. (2023). *Web Coverage Service*. Retrieved December 9 2023 from <https://www.ogc.org/standard/wcs/>
- Open Geospatial Consortium. (2023). *Web Map Service*. Retrieved December 9 2023 from <https://www.ogc.org/standard/wms/>
- Open Source Geospatial Foundation. (2023). *About - MapServer*. Retrieved December 11, 2023 from <https://mapserver.org/about.html>

- Open Source Geospatial Foundation. (2023). *About – GeoServer*. Retrieved December 11, 2023 from <https://geoserver.org/about/>
- Ruiz, M. (2018). *User-Centered Design for Web-Mapping Applications: A Case Study with Ocean Mapping Data for Ocean Modellers*. [Unpublished M.Sc.Eng. Thesis]. University of New Brunswick.
- Sandvik, B. (2014). Using webmap services with three.js. *MasterMaps*.
<https://blog.mastermaps.com/2014/03/using-web-map-services-with-threejs.html>
- Stanghellini, G. and Carrara, G. (2017). Segy-change: The swiss army knife for the SEG-Y files. *SoftwareX*, vol 6, 42-47.
- ThreeJS. (2023). Three JS Instruction Manual. Retrieved October 20, 2022 from <https://threejs.org/manual>
- University of New Brunswick (2014). [Beaufort Sea bathymetry grid file]. Retrieved Dec 20, 2021, from http://www.omg.unb.ca/projects/Arctic/basemaps/71_15_N_136_00_W_BATHY.html
- University of New Brunswick (2014). [Sub-bottom profile from Mapsheet 208]. Retrieved Sept 27, 2022, from <http://www.omg.unb.ca/projects/Arctic/stripmaps/nwp2010/208.html>
- Vainionpaa, M. (2023). *Arctic Seabed Web Mapping Guidebook*. Github.
<https://mcvainio.github.com>

Curriculum Vitae

Madeline Vainionpää

University of Waterloo, B.Sc.

Awards:

Larry Mayer Prize for Ocean Mapping (Oct 2021)

Publications:

N/A

Conference Presentations:

Canadian Hydrographic Conference (Jun 2022) - *Examining open-source web GIS methods for public dissemination of Arctic seafloor mapping data*

ArcticNet Annual Scientific Meeting (Dec 2022) - *Web GIS for Arctic Seabed Mapping: Improving data dissemination using open-source tools.*