

Oceanographic Analysis in the Bering Sea Using ArcGIS Engine/Java/Java3D

Tiffany Vance¹, Nazila Merati² and Christopher Moore²

1- NOAA/Alaska Fisheries Science Center
Seattle, Washington

2 - University of Washington/JISAO
NOAA/Pacific Marine Environmental Laboratory
Seattle, Washington

ABSTRACT

Ideally, scientists should be able to format, explore, analyze, and visualize data in a simple, powerful, and fast application that would seamlessly integrate georeferenced data from a variety of data sources into a powerful intuitive visualization. Virtual reality has made visualization easily accessible to many users using desktop tools but has limitations in scope and analysis. In this project we used a combination of Java/Java3D and the recently introduced ArcGIS Engine product to create a prototype of a tool to calculate mixed later depths for an oceanographic module for a scientific GIS. We combined the spatial tools exposed through ArcGIS Engine with the analytical capabilities of algorithms written in Java and the complex visualization capabilities of Java3D. Modules from each of these technologies could be combined to create innovative tools to allow users to import data, perform spatial and scientific analyses and output the results as visualizations for further examination.

Introduction

Geographic information systems (GIS) provide a high level of functionality for spatial analyses, but are not yet able to provide the extended functionality needed to create a truly “scientific GIS”. Examples of the functionality that is lacking include time series analyses, calculation of the volume of the overlap between two volumes - for example between a school of fish and a prey field, or calculation of the intersection of a vector path with a volume - for example the route of a marine mammal through a pool of cold water. Other functions might include the ability to specify a slice through a three-dimensional lattice of model output data and to make various analyses along that slice.

Java can be used to program scientific calculations and analyses, but it isn’t inherently spatial. Datasets can have a spatial component, but Java treats this as it would any type of coordinate system. Topology, or the spatial relationship between objects, is not stored with data. Functions such as map projections, slope calculations and spatial intersections are not native to the language. However, Java is easily extensible and functions written in other languages can be integrated.

VRML provides the ability to visualize scientific data and to allow the user to interact with the data by rotating, zooming and panning, but you cannot easily query VRML objects. Ideally one would be able to point at a three-dimensional location in a VRML view and easily return the values of the variable and associated information for that location. VRML scene navigation generally requires that a separate VRML plug-in application be installed on the client system. From the user’s perspective, plug-ins are poorly integrated into the browser environment since they are a separate application and often operate in a separate window from the browser. Additionally, useful plug-ins may only be available for the more popular client platforms. Recent developments in Java3D extend the functionality of VRML and answer a number of the limitations mentioned.

This applications is designed using a framework approach that enables the software (Java) to be integrated with other related packages (Java3D) and interfaced with major off-the-shelf software products (ArcEngine). We aimed to develop these tools for a broad range of potential users. This project focuses on tools that convert non-spatial data to GIS compatible data, expedite the transfer of spatial data to coastal professionals and emergency managers and enhance analyses used for disaster preparedness and response activities. As the tools we develop can be deployed without a full ArcGIS license, we hope to make them widely available to better integrate field activities during disaster responses.

Technology

In this project, we use the flexibility of Java to integrate GIS functionality with Java3D based visualization. Specifically we are using a combination of Java/Java3D and the recently introduced ArcEngine product to create a prototype of a scientific GIS. We combine the spatial tools exposed through the ArcEngine Java API with the analytical capabilities of algorithms written in Java with the complex visualization capabilities of Java3D. Modules from each of these technologies will be combined to create innovative tools to allow users to import georeferenced data, make spatial selections, perform spatial and scientific analyses and output the results

as visualizations for further examination. Use of the ArcIMS Java Connector will allow these modules to be implemented in ArcIMS sites for web-based analysis.

[ArcGIS Engine](#) is an ESRI developer product for creating and deploying ArcGIS solutions. It is a simple API-neutral cross-platform development environment for ArcObjects - the C++ component technology framework used to build ArcGIS. ArcObjects are the core of the ArcGIS functionality and include tools such as overlay - union, intersect; proximity - buffer, point distance; surface analysis - aspect, hillshade, slope; and data conversion - shapefile, coverage and DEM to geodatabase. ArcEngine's object library makes full GIS functionality available through fine and coarse-grained components that can be used in Java and other environments. Using ArcEngine, one can build solutions and deploy them to users without requiring the ArcGIS Desktop applications (ArcMap, ArcCatalog) to be present on the same machine. It supports all the standard development environments, including Java, and C++, and all the major operating systems. In addition, one can embed some of the functionality available in the ArcGIS extensions. This product is a developer kit as well as deployment packages of ArcObjects technology. Using ArcEngine we will integrate GIS functionality into an application with the data being available for calculations in non-GIS components. We will also be able to make these tools available to ArcIMS sites. ArcIMS has a limited set of spatial capabilities but it is capable of interfacing with the Java 3D API via the [Java Connector](#). This will allow the ArcIMS community to utilize tools built with this project.

Java allows us to make a variety of scientific calculations on the data and to provide the results back both to the GIS component and to a Java3D based visualization component. We are able to take advantage of a number of Java utilities such as the [UNIDATA Java tools including IDV, OceanShare, ncBrowse, SGT toolkit](#) and TimeSeries applet. The ArcIMS Java Connector could be used to produce map coordinates based that would allow data retrieval from a DODS server, and subsequent plotting with tools designed for interaction with gridded fields.

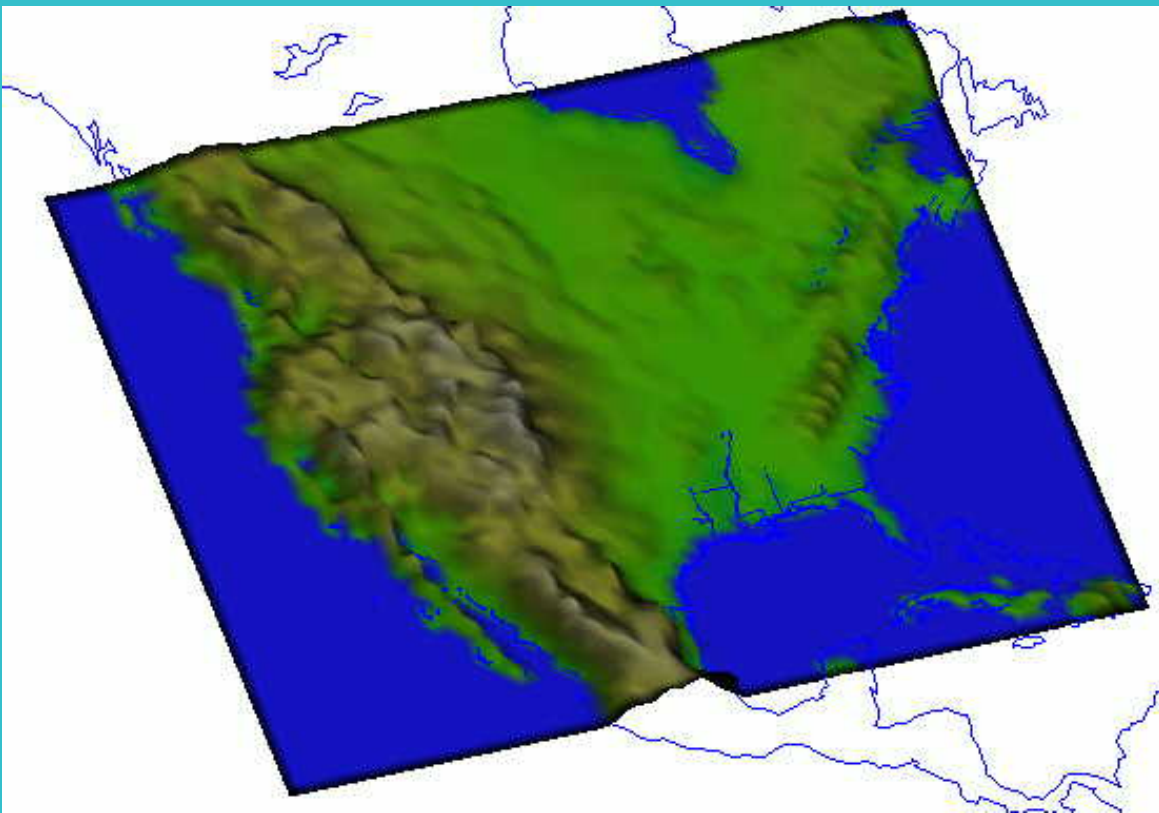


Figure 1. Visualization of topography made using IDV
[from <http://www.unidata.ucar.edu/content/software/IDV/gallery/index.html>]

The [Java 3D API](#) is an application programming interface used for writing stand-alone three-dimensional graphics applications or Web-based 3D applets. It gives developers high level constructs for creating and manipulating 3D geometry and tools for constructing the structures used in rendering that geometry. With Java 3D API constructs, application developers can describe very large virtual worlds, which, in turn, are efficiently rendered by the Java 3D API. The Java 3D API extension is designed as a high-level platform-independent 3D graphics programming API and is amenable to very high performance implementations across a range of platforms. To optimize rendering, Java 3D API implementations are layered to take advantage of the native, low-level API that is available on a given system. In particular, Java 3D API implementations that utilize OpenGL, Direct3D API, and QuickDraw3D are available. This means that Java 3D API rendering will be accelerated across the same wide range of systems that are supported by these lower-level

APIs. The Java 3D API is aimed at a wide range of 3D-capable hardware and software platforms, from low cost PC game cards and software renderers, through mid-range workstations, all the way up to very high-performance, specialized, 3D image generators. Support for run-time loaders was included to allow Java 3D API to handle a wide variety of file formats such as interchange formats, VRML 1.0, and VRML 2.0.

Combining these three technologies, we will be creating an application for the [NOAA nowCoast](#) project. The nowCoast project is an Office of Coast Survey effort to provide forecast model developers and the coastal community with centralized access to real-time physical, meteorological, oceanographic, river, and air/water quality information. The Web portal also provides NOAA forecasts for major estuaries, seaports, and adjacent coastal regions as well as the Great Lakes. The application this project develops will use ArcEngine and Java3D to create a GIS tool to serve 3D rendered objects of model data. It will enhance nowCoast by the addition of a number of 3D visualization and spatial analysis tools. These will include creating 2D and 3D plots from a polyline, creating 2D velocity plots from point input (columnar plot) and a capability for viewing model data in 3D using Java3D.

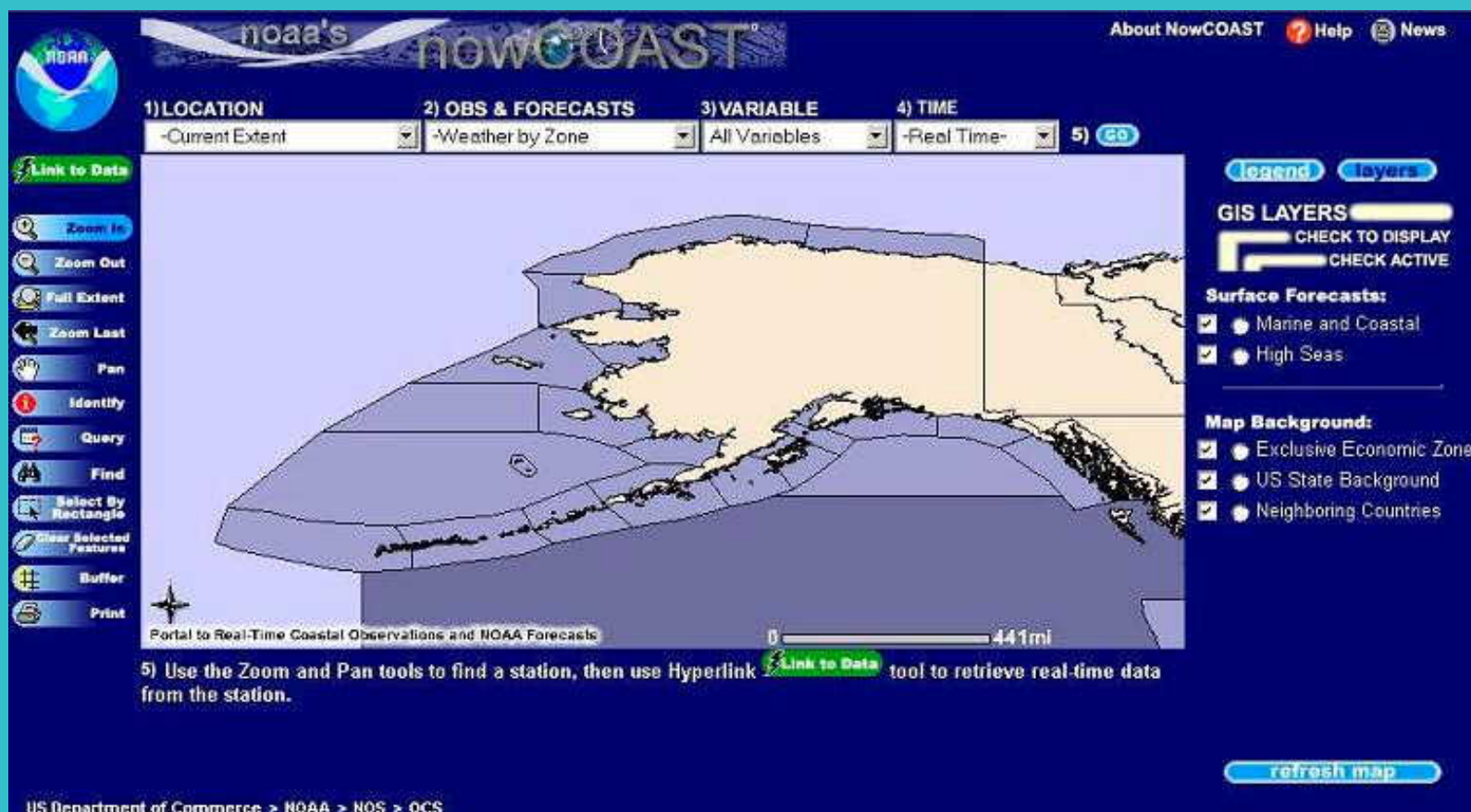


Figure 2. nowCoast ArcIMS site

A second application, for the [NOAA/NMFS/Alaska Fisheries Science Center](#), will allow scientists to calculate a variety of statistics and measures about the intersection of vector and volumetric objects with volumes in 3D. The objects could include marine mammal tracklines and schools of fish and the 3D volumes might include schools of prey and water masses such as cold pools. These types of volume on volume intersections could be generalized to a number of coastal and offshore applications. They will also serve as templates for many other tasks.

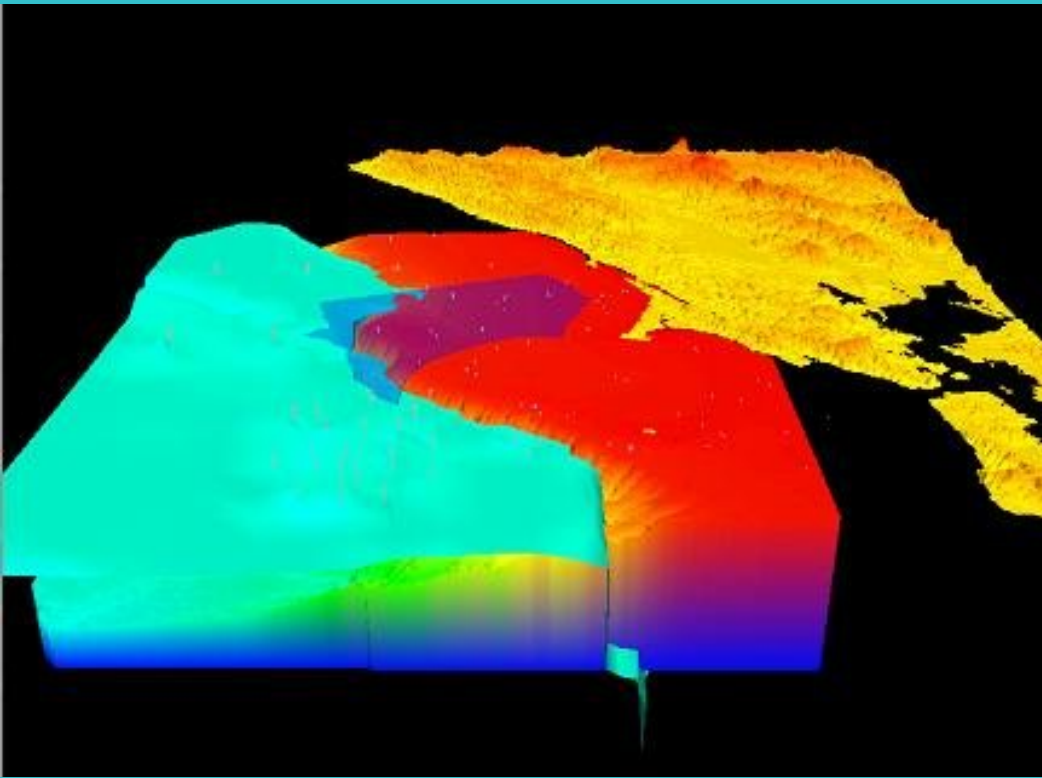


Figure 3. Visualization of the Cordell Bank National Marine Sanctuary showing water volumes

Future Directions

Delays in receiving funding have slowed our development efforts. Once a final evaluation of the costs and benefits of ArcEngine versus MapObjects - Java have been completed, we will begin development of the nowCoast and AFSC applications. The calculation of mixed layer depth is being used as a test algorithm. The system will be designed to allow the substitution of other modules for the calculation of a variety of oceanographic parameters. Once we have fully developed the three-dimensional aspects of the project we will develop methods for volume-volume intersection calculations and looking at the paths of individual animals or fish through oceanographic features.

Conclusions

While none of the three technologies we have used is complete in and of itself, the linking of GIS, Java and Java3D provides a powerful mechanism to create the beginnings of a "scientific GIS". Once the technologies have been linked, the form of the final tools deployed will be dependent upon user needs and the relative costs of the GIS portions of the technologies. Because we are creating scientific tools rather than a commercial product, license costs may have to play a large role in our choice of which GIS tools to use.

Acknowledgments

This project has been funded in part by the NOAA High Performance Computing and Communications (HPCC) program.

References

For more information about the Pacific Marine Environmental Laboratory's visualization efforts, please visit the PMEL home page at <http://www.pmel.noaa.gov/vrml/3DViz.html> and <http://www.pmel.noaa.gov/visualization/>

Author Information

Tiffany C. Vance, Computer Specialist
National Oceanic and Atmospheric Administration
Alaska Fisheries Science Center
7600 Sand Point Way NE
Seattle, Washington 98115
Telephone: 206-526-6767
Email: Tiffany.C.Vance@noaa.gov