Using Processing to develop iCove: a tool for interactive coastal oceanographic visualization

R. L. S. F. George\textsuperscript{1} and J. C. Roberts\textsuperscript{1} and Alan. G. Davies\textsuperscript{2}

\textsuperscript{1}School of Computer Science, Bangor University, UK
\textsuperscript{2}Centre for Applied Marine Sciences, School of Ocean Science, Bangor University, UK

Abstract
We have been developing iCove: the interactive coastal oceanographic visualization environment. The challenge for the ocean scientists is that their models are complex and the datasets that are generated are huge; furthermore, the oceanographers wish to interactively investigate and quantitatively compare different runs of these models. We propose a novel visual analytics tool to permit detailed exploration through interactive data querying to enable their analysis. This paper presents our experience of building iCove in Processing especially in comparison with our previous oceanographic tool building in VTK.

Categories and Subject Descriptors (according to ACM CCS): I.3.8 [Computer Graphics]: Applications H.5.2 [Information Interfaces and Presentation]: User Interfaces

1. Introduction

In recent years, enhanced computing power and modelling techniques have helped improved scientific understanding in many fields of study – earth and ocean sciences included. Researchers have developed numerous mathematical models that simulate oceanographic systems. But these models typically produce extremely large, multi-variate datasets, furthermore researchers apply increasingly complex methods of integrating and analysing the data, which leads to “data flooding” \cite{EYD01} or “information overload” \cite{KMSZ09}, and conventional analytical methods have limited effectiveness \cite{BB02}. Thus, visualization systems need to evolve to support the analysis of these large oceanographic datasets. Furthermore, the integration of disparate datasets through visualization supports effective and flexible visual processing, facilitates comparison with other sources of data \cite{WSG00} and often provides the first opportunity to view datasets in their entirety \cite{GWK96,HPC97}, frequently revealing relationships and detecting features not previously apparent.

However there are few coastal visualization tools, and most of the current tools are used to display deep ocean models. But coastal shelf studies are increasingly important in climate change research, and are often more complex, unpredictable and detailed than deep ocean models \cite{WSG00,LG05}, presenting a new challenge to analysts to make effective predictions. Recently, advanced visual analytics techniques have evolved which are recognised as valuable for studies involving large volumes of unstructured spatio-temporal data \cite{KMSZ09}, and which could be applied to coastal shelf studies. So there is a compelling argument and need to develop innovative means of exploration to aid understanding.

In this paper we present our experience of developing iCove using Processing, building upon our earlier two prototypes \cite{GR09}, to develop a novel system that supports highly interactive controls and novel oceanographic visualization techniques, permitting detailed quantitative visual analysis and exploration of coastal and estuarine hydrodynamic and sediment transport data.

2. Background and Related Work

This work aims to develop a more advanced prototype, providing faster, high quality rendering, together with greater interactivity and explorative functionality. We outline the oceanographic and visualization challenges, and discuss the selection of software which may permit resolution of some of the difficulties encountered in our earlier VTK and OpenDX prototypes.
These two prototypes were developed to gain an early understanding of the challenges of visualizing coastal shelf tidal flow data. Prototype 1 – OpenDX, focused on building a visualization tool to study aspects of a specific oceanographic domain – the complex tidal flow of the Menai Strait, North Wales. However, it is recognised that visualization tools for a specific oceanographic domain are not easily extensible for generic use [BEKE06], especially in an area such as coastal shelf science, with its complex domains and specific geometry. The model and data for this prototype was provided by the School of Ocean Science, Bangor University [Mar06]

As a result, we developed Prototype 2 (VTK) as a generic visualization tool, which might be easily adapted to visualize different datasets, incorporate interrogation, and resolve some of the memory management and speed problems experienced with Prototype 1. For this tool, we used flooding scenarios of the Dyfi Estuary, North Wales, provided by the Centre for Applied Marine Science, Bangor University, [Rob08] and it is this work that we build on with the development of iCove.

2.1. Challenges – The oceanographic domain and its modelling systems

Oceanographic hydrodynamics is a large domain of study, with widely varying scales requiring differing visualization solutions. We focus on coastal shelf studies, as its complexity and numerous inter-connecting factors present many additional challenges to those associated with deep ocean studies [LG06]. There are a proliferation of models, modelling techniques and variables within the model [Jon02] but whatever model is chosen each of the datasets tend to be highly multi-varate containing both scalar and vector data, and are extremely large (the data we use has >20000 points and >700 timesteps), and have highly multi-variate, scalar and vector data. Many use unstructured adaptive grids, where objects are represented by unconnected points with three dimensional coordinates and also may change position over time (e.g. sandbanks and tidal channels); and datasets have abnormal distribution because of the nature of the sampling. Furthermore, although areas of small scale topography and processes are now studied in greater detail [DO06] conclusions from microscale analysis may not apply on a larger scale [BM00].

Issues of validity and uncertainty also present a significant challenge: Bethel, Johnson et al and Schroeder, Bertel et al citeBethel2007,Schroeder2005, highlight the tension between underlying scientific content and the visualization, through extrapolating extra detail to produce the visualization, yet there is also a need to simplify the complex environment to improve understanding. Equally, Bates and Anderson, Hardy and Bates and French and Clifford [BA97,HB99, FC00], amongst others point out the impact of changes to spatial resolution on the validity of simulations and visualizations.

Yet, despite these challenges, our early research [GR09] suggests that current practice amongst many ocean scientists is to rely on the limited functionality of the numerical model’s graphics post processor, rather than on more sophisticated data exploration techniques. So, we conclude there is a need for advanced analytic tools to aid exploration, understanding and decision making.

2.2. The visualization challenges

Coastal shelf and estuarine visualization exemplifies one of scientific visualization’s current major challenges – to move from systems simply representing and confirming a model’s output, to those permitting detailed analysis and exploration of the data, such that they become integral parts of the scientific knowledge discovery process [KMSZ09, BJJ07]. Yet, despite the increased use of visualization to support data analysis, a number of factors influence and limit researchers’ ability to effectively explore data. Our earlier prototyping work [GR09] highlighted two of these factors: system and information overload. The huge quantities of data and information now generated by numerical models and associated visualizations result in the requirement to balance speed of operation and quality of rendering. But there is also a need to clarify and illuminate the data – to “hide complexity” [BBC08], whilst highlighting spatial and temporal relationships, and enabling researchers to effectively interrogate their data in a timely manner.

Figure 3: Paradigm for visualizing coastal shelf and estuarine numerical model datasets: Data from the model feeds into the visualization. Visual analytics is performed on the visualization, which then produces a further set of data, which may be feedback into the system, visualized and then analysed.

The ultimate goal with iCove is to achieve an acceptable speed/quality balance (real-time computing but not at the expense of quality of rendering); to integrate multiple source, multiple type, temporal, structured and unstructured data; to

Figure 1: OpenDX Prototype visualizing a Menai Strait data set (TELEMAC2D model and data, Marten, 2006 [Mar06])

Figure 2: VTK prototype, visualizing the Dyfi dataset used to develop iCove (TELEMAC-2D flooding scenario, Robins 2008 [Rob08])

undertake interactive data querying and identify, extract and support study of features of interest.

Our research seeks to identify means of improving performance and analytical capability – to integrate visualization with novel techniques of visual analytics in coastal shelf studies, as such a combination of analytical reasoning with interactive visual interfaces [KMSZ09], has already proved applicable in a geo-spatial and temporal context. Figure 3 shows the paradigm we have adopted which effectively uses visual analytics to create a cycle to feed back new data into the research cycle, thus enabling our system to become part of the knowledge discovery process. Unlike the conventional visualization pipeline, the process does not have an end. For example, with tidal flux, the original data would be used to produce a visualization, upon which we would perform the tidal flux calculation using an analytic tool. This would then produce a further data set, which we could feed back into the visualization, and use for more detailed analysis of other aspects of the model, such as sediment movement.

An oceanographic example of a similar technique is the work of Turdukulov et al.’s [TB09] study of iceberg movement, using computational feature tracking to reduce information overload and produce quantitative information about each tracked feature.

3. Ideology and Motivation

Our dataset is an extremely large marine science simulation, which was developed to investigate flooding scenarios of the Dyfi Estuary, Mid Wales. The model was developed by our colleagues in the Centre for Applied Marine Sciences, Bangor University, and discretizes space as an unstructured grid of triangular elements [BA97]. The model is usually analysed and manipulated further by Rubens, a software system designed for use with the modelling tool. However this software does not support higher levels of exploration or interactive visualization.

3. Ideology and Motivation

Our dataset is an extremely large marine science simulation, which was developed to investigate flooding scenarios of the Dyfi Estuary, Mid Wales. The model was developed by our colleagues in the Centre for Applied Marine Sciences, Bangor University, and discretizes space as an unstructured grid of triangular elements [BA97]. The model is usually analysed and manipulated further by Rubens, a software system designed for use with the modelling tool. However this software does not support higher levels of exploration or interactive visualization.

3.1. Aims

The ultimate goal of iCove is to provide interactive data querying and visual analytics at optimum speed/quality. Thus, it will incorporate:

- visualization and analysis of multimodal, structured and unstructured, temporal, vector and scalar data;
- coordinated multiple views allowing visual and quantitative comparison of models;
- a fluid interface, allowing analysis of the data;
- oceanographic analytical tools e.g. transect profiler, flux calculator;
- the ability to produce high quality graphs and outputs;

We describe our initial development below.

3.2. Solution

iCove is written in Java, using Processing as the renderer, but also incorporating elements from VTK. Processing’s universal portability makes it attractive for use as a visualization tool, and in regard to VTK, our earlier work identified under-sampling at the mouth of the Dyfi Estuary as a major area of concern requiring rigorously guaranteed triangulation, provided by VTK’s robust Delaunay triangulation. We decided to retain this for the current version of our tool.

Processing is an open source programming language and development environment for images, animation and interactions, and originated from the MIT Media Lab [MIT]. We recognize our selection of Processing is unconventional, as it is not normally associated with oceanographic visualization, although Neil Banas of the University of Washington has used it in a coastal shelf application exploring the dynamics of coastal marine ecosystems [Ban08].

Nevertheless, Processing is increasingly used for visualization and visual analytics (http://processing.org), although it seems to be largely untested, as our research did not reveal a significant body of academic literature relating to its application for scientific visualization. However, in reviewing the features of readily available examples of systems build using
R. L. S. F. George & J. C. Roberts & Alan. G. Davies / Using Processing to develop iCove

the software, such as “In the Air” [Cal10] and “Just Landed” [Tho10], these seem to possess many of the attributes we require. In view of the need to resolve the speed/quality of rendering issues we had experienced with our VTK prototype, where we concluded it needed a larger computer system for optimum performance [GR09], we selected Processing for the following attributes: fast, runs on multiple platforms, has good graphics capabilities and the ability to integrate other libraries [MIT].

4. Development of iCove

There are three important parts to iCove: Window management, Data Import and Visualization, which we describe below.

4.1. Window management

Ideally, what we wanted was one Processing window (a super window) that could include several subwindows. But, Processing does not natively support multiple windows and the management thereof. Consequently, there are two different solutions. Either an external windowing system such as Swing could be used; or we could develop libraries to extend Processing to provide the required functionality.

If Swing (say) is used as the window management system, individual Processing applets could be embedded into individual Swing panels. This is advantageous because Swing would be used to manage the windows. However, this has the disadvantage that it would no longer be a Processing applet that would lose some of the Processing advantages (of ease of distribution and cross-platform development), there would be several applets running and importantly we would have to write the code to synchronize the drawing and updating of the individual Processing components. A better way may be to use a client/server model where the individual windows would be updated through the server. But, we also wished to allow graphical elements to be placed on top of the windows, such as to annotate the visualizations (and animations that could move between windows) which would be difficult to achieve within a Swing environment. Consequently, we decided that this model was not suitable for our implementation.

Our approach was to develop basic windowing commands for Processing. We decided to add two libraries: a basic window management system that we developed, and a GUI element library that we imported, as described below.

4.1.1. Basic window management system

Our basic window management system provides a set of classes to extend Processing to allow for multiple windows / panels. This framework allows for the multiple panels to be drawn and interacted with and includes basic windowing operations such as, focussing, moving, scaling, opening and closing of the panels. Advanced interaction can be built into each of the specific windows / panels. See figure ?? for an overview of the structure of the window management framework.

4.1.2. GUI Elements

Basic GUI elements (sliders and buttons) and their interaction were incorporated into iCove using Andreas’ Schlegel’s ControlP5 library for Processing. This has the advantage that the all basic user interface elements can be easily plugged into specific panels and can be drawn directly to the main super window without the need for a separate control panel that you would get with swing.

4.2. Data import

In order to get iCove developed quickly, we decided to re-use the data import routine written in Java from the previous VTK tool. This translated the TELEMAC data and meant that it could be readily imported into iCove and results quickly viewed but with the drawback that the data needs to be translated from its native format (Binary and ASCII) into the VTK data format, which is inefficient. We are currently rewriting this data importer to be more efficient and to read the native TELEMAC format directly.

4.3. Visualization

The principle visualization is the planar view of the estuary. This is the master view and all the other views are connected to it. Because the data is generated at a very high resolution and is on an irregular grid, there are several challenges to overcome. For instance, multiple pixels are plotted on the same space on the image; this both means wasted calculations and also the image does not appear clear. Also, the amount of data means that it is slow to process and also render updates are slow. Consequently, we decided to spatially index the data using a quadtree, to support quicker selection of the underlying data points. See figure 5

![Image: Spatially indexed data using a quadtree](image)

Figure 5: Spatially indexed data using a quadtree

A graph panel was built to display the multiple temporal variables of each point, initially only for the water elevation
Figure 4: Simplified class dependency diagram for the window management framework of iCove.

4.4. Using iCove

At its current stage of development, iCove allows users to import data from TELEMAC, providing an interactive 2D graphical result of the dataset. Users are able to draw graphs of the underlying data values (currently only the water elevation) by mouse clicking inside the output window, they can also view the dataset changing over time by using the animation controls (play, step forward / backward). All the panels can be arranged (interactively moved and scaled) to suit the user, see figure 6.

5. Future Directions

Short term, we are developing and implementing user testing of a key part of the user interface, not to test the content, but the mechanism for displaying and referencing data – how the system will relate a graph to its position in the grid, without adding to information overload. Conventionally, graphs are related to the 2D output window through explode lines, but the sheer number of graphs that oceanographers may need to review and compare to analyse these data may result in a confusing proliferation of lines drawn across the output window. So, we are developing an alternative method of referencing the graphs to the 2D output window to reduce visual clutter, which will be tested with users, who will be presented with both options, to establish their preferences.

Based on the outcome of user testing, future work will include development of feature identification and extraction, and tools such as the flux calculator. This is relevant to many areas of coastal shelf studies and has practical implications. For example the ability to identify and extract detailed hydrodynamical information is relevant to civil engineering works, such as jetties and piers, where an understanding of the sediment transport is essential if these are to be usable.

Longer term, traditional processing methods on a CPU are time consuming, so we will test whether transferring the system to a GPU will improve speed (our aim is to speed up the process to near real time). A further development will be to extend the system to support analysis of multiple datasets, enabling intercomparison of different models, or versions of the same model. This will support the predictive capability of the tool.

6. Discussion & Conclusions

Processing is easy to learn and can be applied to visualize oceanographic data, providing rapid prototyping, with its
Figure 6: Current stage of iCove, showing graphs of the tidal harmonic at selected point.

simple outlines, and producing a robust, reliable system. Unlike VTK, Processing does not have a large library of tools, and because of the development work required, it was decided to use the original VTK prototype data files, updated and converted to ensure correct 2D output, to enable a prototype to be produced relatively swiftly. Whilst this was successful, it is neither a long term nor an elegant solution, as it proved difficult to access some of the data. Consequently, we are now designing a data importer which will eliminate the need to use the VTK data, thus enabling us to use test data from other areas of coastal shelf study for later developments of the system. The new data importer will reduce bottlenecks and improve speed. The development of the quadtree has also improved access times to the data, by reducing the number of points to be searched from tens of thousands to around 500. This has enabled the points to be selected at an interactive rate.

Whilst the initial development of tools in Processing for this area of study proved a little time consuming, it was advantageous in that it provided a good understanding of the software. Furthermore, it permitted greater control and customising of the visualization system to our requirements, than that experienced with our earlier VTK prototype, which used an existing library of tools, parts of which were not relevant to our needs. As a result, we were able to create a more responsive system.

Another attribute of Processing is that it was originally designed as an interactive graphical web package, which means that interactive graphics can be readily exported to the web, without the need to develop a separate web version, as is the case with other visualization packages. This is of considerable value for the increasingly collaborative research taking place at numerous centres, worldwide, and will prove of value in our future development of the system.

Our initial impression is that even at this early stage of evolution, the system is running well and smoothly, because Processing is well optimised. Many of the problems experienced with our earlier VTK prototype, as a result of its need for a large system for optimum performance, have not been experienced with Processing. We have concluded that Processing has proved to be flexible, responsive and easy to run on smaller systems.

With regard to the interface we have developed for performing visual analytics, we now propose to undertake user testing of how this will reference information.

This paper described the initial development of iCove with a goal of using the tool to advance the understanding of coastal oceanography. We also hope to contribute to pushing the boundaries of what might be achieved with Processing by creating new libraries of tools for oceanographic studies.

References


[BB02] BELATON B., BRODLIE K.: Model centred ap-
proach to scientific visualization. *Journal of WSCG* 10, 1 (2002), 63–70. 1


[Cal10] CALVILLO N.: In the air. Website, January 2010. 4


[MIT] MIT M. L.: Processing. Website. 3, 4

[Rob08] ROBINS P.: Present and future flooding scenarios in the Dyfi Estuary, Wales, UK. Report for the countryside council for wales (number 3), Centre for Applied Marine Sciences, Bangor University, 2008. 2, 3


[Tho10] THORP J.: Just landed. Website, May 2010. 4
