

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

R. L. S. F. George<sup>1</sup> and J. C. Roberts<sup>1</sup> and Alan. G. Davies<sup>2</sup>

<sup>1</sup>School of Computer Science, Bangor University, UK

<sup>2</sup>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 researchers improve their understanding in many fields of study. In particular for the earth and ocean sciences 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” [EYD01] or “information overload” [KMSZ09] and conventional analytical methods have limited effectiveness [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 [WSG\*00] and often provides the first opportunity to view datasets in their entirety [GWK96, HPC\*97], 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 com-

plex, unpredictable and detailed than deep ocean models [WSG\*00, LG05]. This presents a new challenge to analysts to make effective predictions. Currently, advanced visual analytics techniques are evolving that are recognised as valuable for studies involving large volumes of unstructured spatio-temporal data [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. The work builds and develops from our earlier two oceanographic prototypes [GR09] using OpenDX and VTK. The aim of iCove is to provide a novel system that supports highly interactive interaction techniques, novel oceanographic visualization techniques and provide a tool that delivers detailed quantitative visual analysis and exploration capability of coastal and estuarine hydrodynamic and sediment transport data.

## 2. Background and Related Work

The aim of iCove is 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 dis-

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. The tool was engineered such that it may be easily adapted to visualize different datasets, incorporate interrogation and resolve some of the memory management and speed problems experienced with Prototype 1. In particular we used this tool to visualize flooding scenarios of the Dyfi Estuary, North Wales. This data was provided by the Centre for Applied Marine Science, Bangor University [Rob08] and it on this latter work that we build for the development of iCOVE .

### 2.1. Challenges – The oceanographic domain and 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 to use [Jon02] but whatever model is chosen each of the data-sets tend to be highly multi-variate, contain both scalar and vector data, and are extremely large. For instance, one typical dataset we use has >20000 points and >700 timesteps with several multi-variate, scalar and vector data values. Most of these datasets are provided on 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). Furthermore, although areas of small scale topography and processes are now studied in greater detail [DO06] conclusions from microscale analysis are often not followed into a larger scale [BM00].

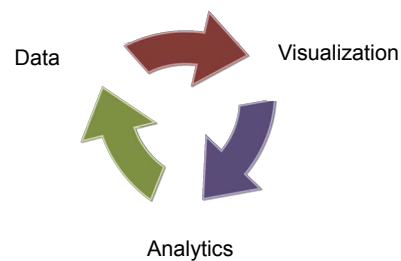
Issues of validity and uncertainty also present a significant challenge. Bethel et al. [BJJ\*07] and Schroeder et al. [SBM\*05] highlight the tension between underlying scientific content and the visualization through extrapolating extra detail to produce the visualization and a need to simplify the complex environment to improve understanding. Equally,

Bates and Anderson [BA97] Hardy and Bates [HB99] and French and Clifford [FC00] amongst others mention of the increase of validity of the simulation and visualizations to an increase of spatial resolution.

Yet, despite these challenges our previous research [GR09] suggests that current practice amongst many ocean scientists relies on static visualizations with limited interaction capability and much post-processing of the numerical data, rather than interactive data exploration techniques. Thus, there is a need for advanced analytic tools to aid exploration, understanding and decision making.

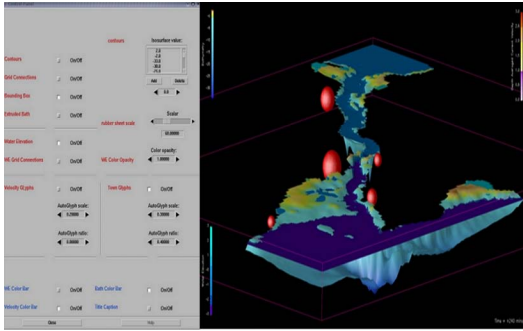
### 2.2. The visualization challenges

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, BJJ\*07]. Yet, despite the increased use of new visualization techniques to support data analysis a number of factors influence and limit a researchers ability to effectively explore data. In particular, our earlier prototypes [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” [BBC\*08], 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 feedback into the system, visualized and then analysed.

The ultimate goal of 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



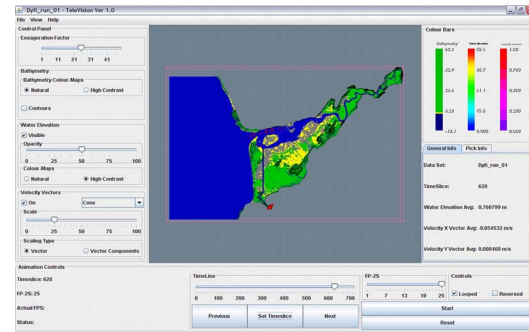
**Figure 1:** OpenDX Prototype visualizing a Menai Strait data set (TELEMAC-2D model and data, Marten, 2006 [Mar06])

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

Our research seeks to identify means to improve 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.



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

### 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:

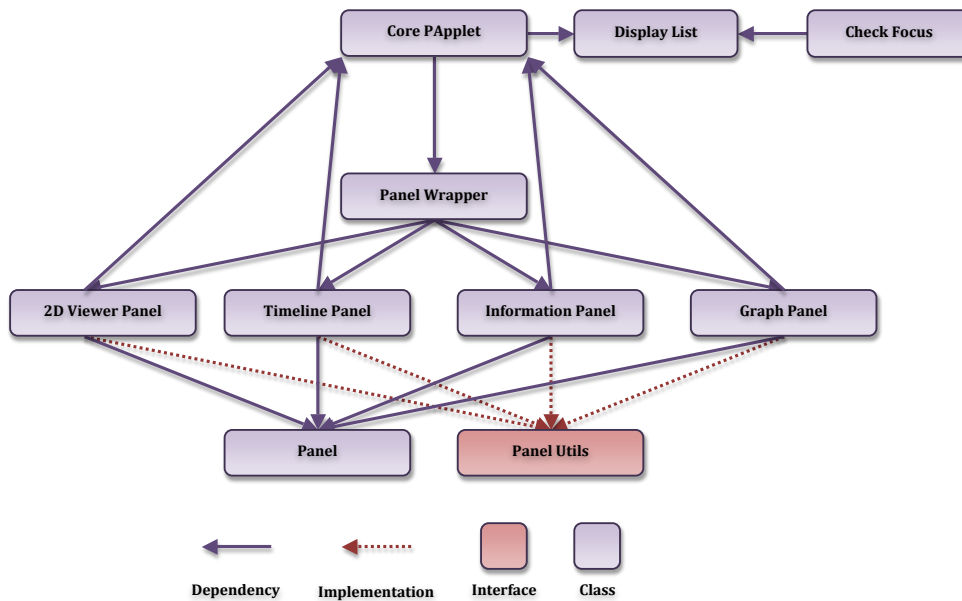
- visualization and analysis of multi-modal, structured and unstructured, temporal, vector and scalar data;
- coordinated multiple views allowing visual and quantitative comparison of models;
- a fluid interface, allowing analysis of the data;
- oceanographic analytical tools e.g., transect profiler and 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 we inherit some of the functionality written in VTK from previous tools to hasten development. 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 as unusual as it is not normally associated with oceanographic visualization. However, one other example of using Processing in this domain is by Neil Banas of the University of Washington who used it in a coastal shelf application exploring the dynamics of coastal marine ecosystems [Ban08].

Processing is increasingly used for visualization and visual analytics (<http://processing.org>) although it seems to be largely untested in the environmental domain. However, in reviewing the features of systems that have been built using it, such as "In the Air" [Cal10] and "Just Landed" [Tho10] it appears that Processing possess many of the attributes we require. We hope to overcome some of the speed and quality of rendering issues that we had experienced with our



**Figure 4:** Simplified class dependency diagram for the window management framework of iCove.

VTK prototype [GR09]. One of the other advantages of Processing is that it is web-enabled and would allow us to develop remotely and collaborate remotely with the domain scientists throughout the build and integrate it with other libraries [MIT].

#### 4. Development of iCove

There are three 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 sub-windows. 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 to manage the windows or we could develop our own windowing libraries in 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 and thus be easily operable with other Swing objects. However, this has the disadvantage that the main program would no longer be a Processing Applet and we would lose some of the advantages of using Processing. Such as the ease of distri-

bution over the web and cross-platform development. Furthermore, several Processing applets would need run in parallel and importantly a developer 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, one of the added functionalities we also required was to allow graphical elements to be placed on top of the windows. For instance, visualizations could be annotated and objects could animated and move between windows. Such functionality would be difficult to achieve within a Swing environment. Consequently, we decided that the separate window method 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. We name these *sub-windows panels*. This framework allows for the multiple panels to be drawn and interacted with and includes basic windowing operations such as focusing, moving, scaling, opening and closing of the panels. Advanced interaction can be built into each of the specific panels.

Figure 4 shows an overview of the structure of the window

management framework. Two implementations have been made. The first implementation was based on a single identifier kept a list. Associated with the identifier is stored the position, size, visualization type and whether the panel was in focus. The manager then determines where the window decorative borders and the contents of the panels are drawn. It also controls how they are moved, resized and the order by which the panels are drawn. The panels were updated in a round-robin method starting from the focus panel. However, this limits the possible arrangements that can be made with the panels and because more complex interactions and overlapping window placements were needed we moved to a second implementation that stored the id's of the windows in a hierarchy. This allows for a richer set of possible overlaps.

#### 4.1.2. GUI Elements

Basic GUI elements (sliders and buttons) and their interaction were incorporated into iCove using Schlegel's ControlP5 library for Processing ([www.sojamo.de/libraries/controlp5/](http://www.sojamo.de/libraries/controlp5/)). 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 reuse the data import routine written in Java from the previous VTK tool. This translated the TELEMAC data and meant that 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 re-writing 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. One challenge is that of resolution. Numerous pixels are plotted on the same pixel in the image. This means that calculations are wasted, information is being occluded and that the image does not appear clear. Additionally, 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 and provide a better way to visualize the data. See figure 5.

A graph panel was built to display the multiple temporal variables of each point. Initially this view only displays the water elevation (and shows the tidal harmonic), but will be

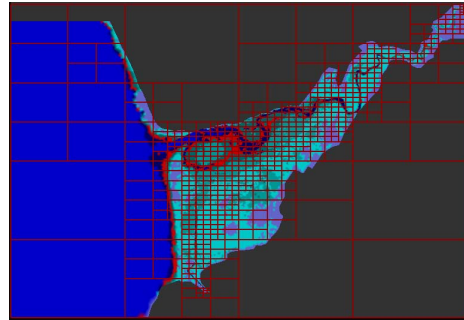


Figure 5: Spatially indexed data using a quadtree

extended to other data points. The point selection from the quadtree was used to determine which graphs to draw, (see figure 6). Currently, simple explode lines are used to reference the separate graph panel to mark the point that they represent. However, if the user requests several panels then they will lead to visual clutter. We are currently investigating different techniques to denote the sample point or transept points for the panels.

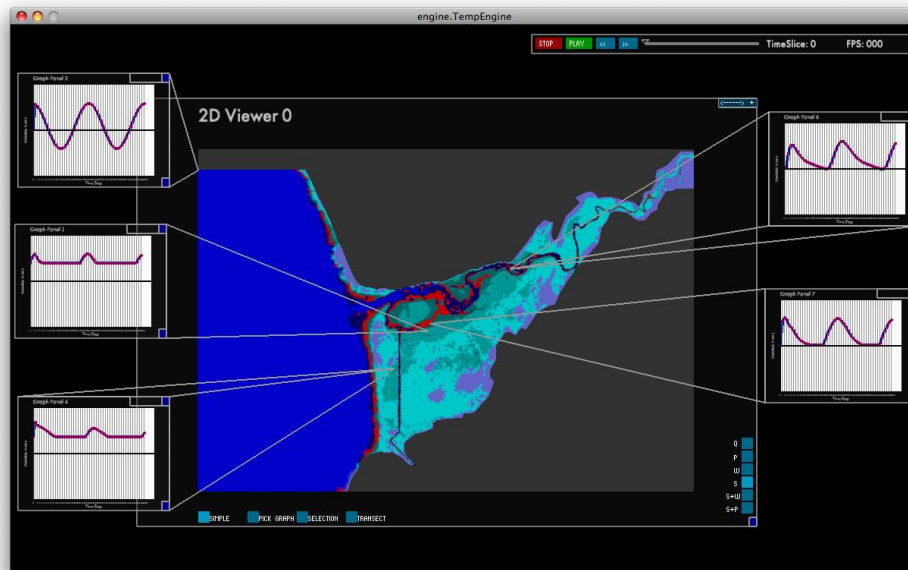
#### 4.4. Using iCove

At its current stage of development iCove allows users to import data from TELEMAC, and provides interactive 2D graphical visualizations of the dataset. Users are able to display graphs of the underlying data values (currently only the water elevation). By mouse clicking inside the output window the user can also view the dataset changing over time and controlling the animation through play, step forward/backward buttons. All the panels can be moved and scaled by the user, 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



**Figure 6:** Current stage of iCove, showing graphs of the tidal harmonic at selected point.

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 inter-comparison 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. It provides a system to rapid prototype interactive tools, it enables developers to create robust and reliable system that can be readily published and interacted with on the web. Unlike VTK, Processing does not have a large library of tools and as such it needs an investment of development time to create appropriate libraries for a particular domain.

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 centers worldwide and will prove valuable as the collaborations and future developments are multi and intra-disciplinary.

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 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.

Many of the problems experienced with our earlier prototypes in OpenDX and VTK have not been experienced with Processing. However, we have had to develop additional functionality within Processing, but now complete we conclude that Processing has proved to be flexible, responsive and easy to run on smaller systems.

With regard to the interface we have developed for per-

forming 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

- [BA97] BATES P. D., ANDERSON M. G.: Investigating the behaviour of two-dimensional finite element models of compound channel flow. *Earth Surface Processes and Landforms* 22 (1997), 3–17.
- [Ban08] BANAS N.: Npzvisualizer. Web, 2008.
- [BB02] BELATON B., BRODLIE K.: Model centred approach to scientific visualization. *Journal of WSCG* 10, 1 (2002), 63–70.
- [BBC\*08] BLOWER J. D., BLANC F., CORNILLON P., HANKIN S. C., LOUBRIEU T.: Underpinning technologies for oceanographic data sharing, visualization and analysis: Review and future outlook. In *GODAE Final Symposium* (2008).
- [BEKE06] BERNHOLDT D., ELWASIF W., KOHL J., EPPERLY T.: A component architecture for high-performance computing. *International Journal of High Performance Computing Applications* 20, 2 (2006), 163–202.
- [BJJ\*07] BETHEL E. W., JOHNSON C., JOY K., AHERN S., PASCUCCI V., CHILDS H., COHEN J., DUCHAINEAU M., HAMANN B., HANSON C., LANEY D., LINDSTROM P., MEREDITH J., OSTROUCHOV G., PARKER S., SILVA C., SANDERSON A., TRICOCHÉ X.: Scidac visualization and analytics center for enabling technology. In *J. Phys* (2007).
- [BM00] BROOKS S., MCDONNELL R.: Research advances in geocomputation for hydrological and geomorphological modeling towards the twenty-first century. *Hydrological processes* 14, 11-12 (2000), 1899–2108.
- [Cal10] CALVILLO N.: In the air. Website, January 2010.
- [DO06] DAVIES A. M., OEY L. Y.: Recent developments in physical oceanographic modelling: Part iii. *Continental Shelf Research* 26 (2006), 1359.
- [EYD01] ERLEBACHER G., YUEN D. A., DUBUFFET F.: Current trends and demands in visualisation in the geosciences. *Electronic Geosciences* 6, 3 (2001), 1–59.
- [FC00] FRENCH J., CLIFFORD N.: Hydrodynamic modelling as a basis for explaining estuarine environmental dynamics: some computational and methodological issues. *Hydrological Processes* 13 (2000), 2089–2108.
- [GR09] GEORGE R. L., ROBERTS J. C.: Coastal shelf visualization: A comparative study of the use of opendx and vtk in visualizing hydro-informatic numerical models. In *Eurographics UK - Theory and Practise of Computer Graphics* (2009).
- [GWK96] GALLOWAY D., WOLANSKI E., KING B.: Coastal oceanography data visualization using data explorer. In *Proceedings of the 1996 IBM Visualization Data Explorer Symposium* (1996).
- [HB99] HARDY R. J., BATES P. D.: The importance of spatial resolution in hydraulic models for floodplain environments. *Journal of Hydrology* 216 (1999), 124–136.
- [HPC\*97] HEAD M., PHU L., COSTOLO O., COUNTRYMAN K., SZCZECZOWSKI C.: Applications of 3D visualizations of oceanographic databases. In *Oceans, MTS/IEEE Conference Proceedings* (1997), vol. 2, pp. 1210–1215.
- [Jon02] JONES J. E.: Coastal and shelf-sea modelling in the european context. *Oceanography and Marine Biology: an Annual Review* 40 (2002), 37–41.
- [KMSZ09] KEIM D. A., MANSMANN F., STOFFEL A., ZIEGLER H.: Visual analytics. In *Encyclopedia of Database Systems* (December 2009). Springer, December, 2009.
- [LG05] LOWE J., GREGORY J.: The effects of climate change on storm surges around the united kingdom. *Phil. Trans. R. Soc A* 363 (2005), 1313–1328.
- [LG06] LEDOUX H., GOLD C.: Modelling oceanographic data with the three-dimensional voronoi diagram. *Review Internationale de Geomatique* 16 (2006), 51–70.
- [Mar06] MARTEN K.: *Validated numerical simulation of the flow in the Menai Strait - a TELEMAC model*. Master's thesis, School of Ocean Science, University of Wales, Bangor, 2006.
- [MIT] MIT M. L.: Processing. Website.
- [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.
- [SBM\*05] SCHROEDER W., BERTEL F., MALATERRE M., THOMPSON D., PEBAY P., O'BARA R. W., TENDULKAR S.: Framework for visualizing higher-order basis functions. In *VIS'05: Proceedings of the conference on Visualization* (2005), vol. Viz05, pp. 43–50.
- [TB09] TURDUKULOV U., BLOK C.: Visual analytics to explore iceberg movement. In *ITC - International Institute for Geo-Information Science and Earth Observation, PO Box 6, 7500 AA, Enschede, the Netherlands* (2009).
- [Tho10] THORP J.: Just landed. Website, May 2010.
- [WSG\*00] WOLANSKI E., SPAGNOL S., GENTIAN P., SPAULDING M., PRANDLE D.: Visualization in marine science. *Estuarine, Coastal and Shelf Science* 50, 1 (2000), 7–9.