

THE “MARINE GIS” – DYNAMIC GIS IN ACTION

Christopher Gold (1), Michael Chau (2), Marcin Dzieszko (1), Rafel Goralski (1)

(1) Dept. Land Surveying and Geo-Informatics Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong SAR, PRC.

(2) Hong Kong Marine Department, Pak Sha Wan, Hong Kong SAR, PRC.

Christophergold@Voronoi.com

KEY WORDS: GIS, Oceanography, Navigation, Algorithms, Data Structures, DEM/DTM, Graphics

ABSTRACT:

The sea moves: the land usually stays still. It is not surprising that the underlying structure of a land-based GIS is rarely appropriate for marine applications. Add the third spatial dimension and it is clear that an attempt to simulate the sea requires a major overhaul of the appropriate algorithms and data structures.

It has seemed obvious to us for some time that spatial data structures need to adapt locally to change, and that the Voronoi diagram provides a conceptually simple framework for which dynamic and kinetic algorithms may be developed. The opportunity to work on real marine data for the Hong Kong area provided the incentive to put our ideas into practice. The challenge was to produce a dynamic three dimensional equivalent to the classical “Pilot Book”, which contains the rules for navigation in the proximity of individual harbours.

While we have done some work on true dynamic three dimensional data structures, as required for marine profiling, the Pilot Book application could be achieved with a kinetic two dimensional structure, but in several layers. The terrain (above and below the sea surface) was modelled with the dual Delaunay triangulation, and the coastline at any particular tidal time was captured by its intersection with the current local sea level. This, together with the locations of individual ships and other surface features, was used to form a two dimensional dynamic Voronoi diagram at the sea surface for proximity and collision detection. Other layers were used to indicate fairways, marine markers, submarine contours, etc.

However, in order to provide a realistic simulation, we needed to take concepts (and models) from 3D games development and provide marine markers such as lighthouses and buoys, and simulate fog and darkness. We also needed to provide a variety of camera views: overhead and on board a selected ship – a deck view, above and behind, below and behind. This required an appropriate scene graph structure to manage the scales, objects, lights and cameras in order to give us the flexibility required for realistic simulation. The result, while still requiring work (especially on ship navigation) may provide a feasible replacement for the Pilot Book, especially for practice simulations.

1. INTRODUCTION

The title “Marine GIS” gives us two contradictory thoughts. On the one hand, a “GIS” refers to a land-based static representation of a two-dimensional surface (maybe with hills). On the other hand, “Marine” implies three dimensional, dynamic representation and analysis. So far, we have used the first as an approximation of the second. This paper attempts to go a little further.

Paper maps are two dimensional and even with modern technology, so are computer screens (stereo systems, although available, still appear awkward). Thus visualization issues become critical for three-dimensional models: either we work with a volume representation (usually voxels) that are either sliced or transparent in places, or else we work with a surface representation (more familiar to most of us) – again with issues of visibility and transparency. This is the mind-set of 3D modelling and games, and many fundamental techniques have been developed in recent years.

Thus a 3D GIS should be dynamic, which is entirely consistent with the marine imperative. However, in the marine case the argument for full dynamism of features as well as observers becomes compelling. Ships move, as do fish, pollution plumes,

tides, and potential obstacles. Therefore we need to look at dynamic (or, more properly, kinetic) data structures to represent our features and their spatial relations.

While we are also working on fully three-dimensional volumetric modelling, for such things as changes in salinity with depth, this paper is primarily an extension of the ideas of Gold and Condal (1995), which worked with one or more two-dimensional surfaces simultaneously. They suggested the simple Voronoi diagram/Delaunay triangulation for the bathymetric model, and the kinetic moving-point Voronoi diagram for the sea-surface navigation and collision-detection layer. We would like to report progress on that approach.

In brief, we started with the ideas of Gold and Condal, but felt that the visualization issues mentioned previously had to take priority. Thus our first step was a well-designed graphical interface, and a well-designed object-oriented spatial data model. Of particular concern were the issues of flexible data structure design, and of a user navigation system within the map space that was intuitively easy to understand. We believe that we have achieved these objectives, which then made the implementation of the spatial models relatively simple. We found our work greatly simplified by the use of OpenGL (Woo et al., 1999) for the visual display (especially with the use of

transparency), Delphi for the programming language (because of its excellent development environment, and object-oriented programming facilities), and the Quad-Edge data structure (Guibas and Stolfi, 1985) for the representation of spatial relationships (because of its simultaneous preservation of both primal and dual relations).

2. RELATED WORK

In recent years, many papers have discussed the problems of using a land-based GIS for marine applications. Probably one of the first articles about Marine GIS is the one by Davis and Davis (1988) in which they state that variables in three dimensions, plus time and attributes are needed to correctly represent the marine phenomena. Li and Saxena (1993), Lockwood and Li (1995) and Wright and Goodchild (1997) give a complete review of what sets apart a Marine GIS from a land-based GIS; they state that objects at sea are different from objects on the ground because they are likely to change position over time, they can have three-dimensional coordinates, they are mostly represented by points (lines and polygons are scarce at sea) and the distribution of samples is usually 'abnormal' (data are sparse in one or two dimensions but abundant in the others). If the nature of the data is completely different, why should a land-based GIS be used for marine applications? The major problems of traditional GIS are that their spatial model is built for static two-dimensional land applications and their data structure is based on the 'overlays' as a definition of adjacencies between objects. The only thing that has been done, by commercial companies, to solve these problems is trying to extend – by adding new kinds of analysis and by modifying the database model – the spatial data model of current GIS; but the real problems are the spatial model used and the building of the topology process, not the database aspect (Gold, 1991). A Marine GIS needs a dynamic data structure to handle time and objects moving over time – local updates of the topology when an object is added, deleted or moved must be possible because now with traditional data structures it is a global operation – not just an extension of the current polygon-arc-node on the plane.

Most GIS research on time tries to extend the current architecture of GIS by adding temporal information in the database, but GIS are usually closed systems and the extensions cannot handle every spatio-temporal problem. To overcome these problems, van Oosterom (1997) proposed a spatial model that can manage changes both over time and topology in the same database by a data structure usually used in CAD systems; and Gold (1996) managed the topology of a map and the spatio-temporal operations made on it – objects can be added, removed or freely moved – with a spatial model based on the Voronoi diagram. This same spatial model was also used by Gold and Condal (1995) and Gold (1999) to build a prototype of a Marine GIS where objects (mostly unconnected points) are handled by the 'tiling' properties of the Voronoi diagram. This prototype, entirely based on the Voronoi diagram to handle topology and perform analysis and operations, has many advantages over the traditional GIS for many applications at sea. Wright and Goodchild (1997), affirmed that this method was the only published attempt that could solve important problems related to the nature of marine data ('abnormal' distribution of samples, dynamism of the sea, etc.).

3. THE VORONOI DIAGRAM AS A SPATIAL DATA MODEL

The static Voronoi diagram (VD) for a set of points in the Euclidian plane is the partitioning of that space into regions such that all locations within any one region are closer to the generating point than to any other. The VD has a geometric dual structure called the Delaunay triangulation (DT) that is defined by the partitioning of the space into triangles – where the vertices of the triangles are the points generating each Voronoi cell – that satisfy the empty circumcircle test (a circle is empty when no points is in its interior, but more than 3 points can be directly on the circle). The DT for a set of non-cocircular points is unique; and so is the VD. In the case of 4 or more cocircular points, an arbitrary choice must be done to form 2 triangles. The static VD for points on the plane has been used in many fields (see Aurenhammer (1991) for a summary). Its properties are well known and many algorithms have been developed to create it. The incremental algorithm is the most interesting in the GIS field because it permits the addition, deletion or movement of points in the VD without rebuilding the whole diagram.

The VD, which can be constructed from the DT and vice versa, also provides the adjacency relationships between points: each cell, which is convex, has a finite number of neighbours. It also has the advantages of both the field-type (raster) and vector representation. Each object (points, lines or polygons) can be represented individually (each one has its own cell) and the space-covering 'tiling' of the plane gives a definition of spatial adjacency between objects (even if they are unconnected). The same spatial model can therefore be used to manage both vector data and images.

The spatial model of commercial land-based GIS is usually based on a vector-based representation (the 'polygon-arc-node' structure being the most popular) and the topological relationships between objects are defined by the 'overlays'. In other words, polygons are formed firstly by identifying the intersections between the lines, and secondly by finding a 'closed loop' among these lines using 'graph-searching' algorithms. This global operation must be done not only to build the initial topology, but also each time there is a modification in the map (insertion, deletion or movement of an object), and for the whole map. An operation is local when, after a modification, only the immediate neighbours of an object must be updated. A GIS based on a static vector-based data model can obviously not manage a moving object or temporal data, two important factors in a Marine GIS

4. GRAPHIC SYSTEM DESIGN

One concept in the development of computer graphics (e.g. Foley et al., 1990) is the ability to concatenate simple transformations (rotations, translation (movement) and scaling) by the use of homogeneous coordinates and individual transformations expressed as 4x4 matrices for 3D worlds. (Blinn, 1977) showed that these techniques allowed the concatenation of transformation matrices to give a single matrix expressing the current affine transformation of any geometric object. More recent graphics hardware and software (e.g. OpenGL: Woo et al., 1999, Hill, 2001) adds the ability to stack previous active transformation matrices, allowing one to revert to a previous coordinate system during model building.

A further development was the "Scene Graph" (Strauss and Carey, 1992; also Rohlf and Helman, 1994) which took this

hierarchical description of the coordinate systems of each object and built a tree structure that could be traversed from the root, calculating the updated transformation matrix of each object in turn, and sending the transformed coordinates to the graphics output system. While other operations may be built into the scene graph, its greatest value is in the representation of a hierarchy of coordinate systems. These coordinate systems are applied to all graphic objects: to geometric objects, such as ships, or to cameras and lights, which may therefore be associated with any geometric object in the simulated world. Such a system allows the population of the simulated world with available graphic objects, including geometric objects, lights and cameras (or observers). An object (with its own particular coordinate system used to define it) is taken from storage and then placed within the world using the necessary translation and rotation. If it was created at a different scale, or in different units, then an initial matrix is given expressing this in terms of the target world coordinates. Geometric objects may be isolated objects built with a CAD type modelling system or they may be terrain meshes or grids – which may require some initial transformation to give the desired vertical exaggeration.

In most cases world viewing is achieved by traversing the complete tree, and drawing each object in turn, after determining the appropriate camera transformation for each window. Usually an initial default camera, and default lighting, is applied so that the model may be seen!

The system described so far creates and views the world from different perspectives. It is designed for full 3D surface modelling, where the surface is defined by a “triangle soup” or the equivalent, composed of unrelated triangles. While it is often desirable for other operations to preserve the topological connections between the components (vertices, edges, faces) of an object, this is not necessary for the basic visualization. For terrain modelling and viewing, for example, it is usually desirable to preserve the topology in order to permit simulation of runoff, etc. as described later.

In addition, objects may themselves move, and the “Animator” component is used to update the appropriate transformation matrix prior to redrawing the scene. Because of the scene graph structure, this mechanism may be used to animate the change of camera view (the observer) which may be associated with some position with respect to a moving object (a pilot on board a boat, for example).

In the real world, objects may not occupy the same location at the same time. There is no built-in prohibition of this within the usual graphics engine. For the Marine GIS we use the kinetic Voronoi diagram as a collision detection mechanism in two dimensions on the sea surface, so that ships may detect potential collisions with the shoreline and with each other. Shoreline points are calculated from the intersection of the triangulated terrain with the sea surface, which may change at any time. The Animator is used to change the transformation matrix of each moving object. It may also be used to change its properties, such as colour. In the case of lights or cameras the view direction may change without changing the location – e.g. for a lighthouse.

The heart of the visualization system “GeoScene” is the Graphic Object Tree, or scene graph. This manages the spatial (coordinate) relationships between graphic objects. These graphic objects may be drawable (such as houses, boats and triangulated surfaces) or non-drawable (cameras and lights).

Redrawing the whole simulated world involves starting at the root of the tree, incorporating the transformation matrix, and drawing the object at that node (if any). This is repeated down the whole tree. Prior to this the camera position and orientation must be calculated, again by running down the tree and calculating the cumulative transformation matrix until the selected camera was reached. This could be repeated for several cameras in several windows. This process must be repeated after every event that requires a modified view. These events could be generated by window resizing or redrawing by the system, by user actions with the keyboard or mouse, or by automated operations, such as ship movements.

5. DESIGN OF A MARINE GIS

Our idea of Marine GIS was a system working with several two-dimensional Voronoi/Delaunay meshes. One of these meshes should represent the bathymetric model – the land and the sea-floor, and a second would be a kinetic moving-point Voronoi diagram to represent the surface of the sea, with a static coastline and boats as moving points. The second mesh would be dependent on the first, to calculate and generate coastline points, when the water level is given, and to check the depth at any location.

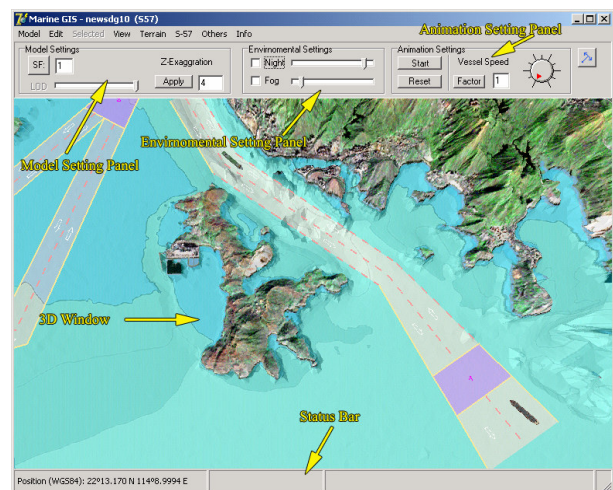


Figure 1: MGIS Model - Graphic User Interface

After implementing the graphic engine, we had to design the structure for our Marine GIS. It had to work with the previously described package and extend it by adding new classes suitable for specific Marine GIS purposes. We had to implement a number of new graphic objects to represent the bathymetric model, the sea surface, navigation markers and boats. Through tests and experiments involving human perception and natural habits of manipulating objects in space, we implemented a system which allows us to perform all needed operations, and which is easy to learn. The system allows us to manipulate the entire scene, like a kind of interactive map in our hands, or to see the scene from a boat-related perspective, using three different points of view: on board, from behind and above, and from under the boat. The observer can look around, and observe the continuously changing surroundings. The system also allows for interactive real-time boat manipulation and navigation, regardless of the chosen display mode.

Boats may have different directions or trajectories, and speeds. They can be added, selected and deleted, or configured by the user. Animation can start or stopped at any time. All boats may

move simultaneously, and then the model becomes something like a live map, showing real-time changes. The model may still be manipulated in the same way, as when animation is off. Each boat can be controlled and navigated by the user, who can change its speed and direction, stop it or set its destination point.

There are four different camera modes. First is the general view (Figure 1). Other modes are boat-dependent, and to use them one of the boats has to be selected first. The camera is then set in a position dependent on the selected boat's location, and moves with the boat through the scene. It can be placed behind-and-above the boat, behind the boat under the water, or on board. In these modes the user can look around and control the boat to turn left or right, to speed up or slow down, to stop or to move. Figure 2 shows the view from behind and above the boat.

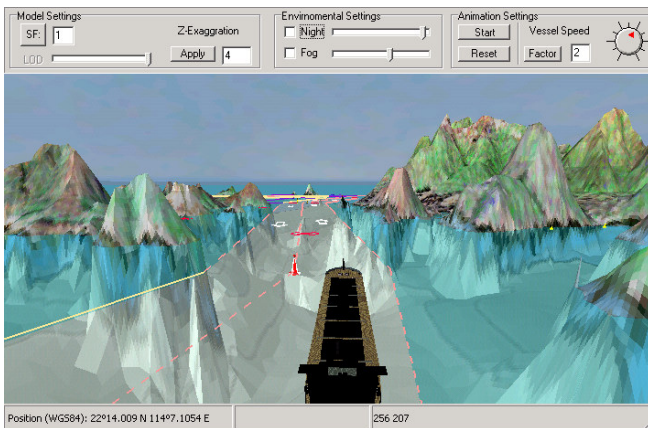


Figure 2: MGIS Model – Scene of Navigational Mode. When animation is activated, the viewpoint will follow the movement of the ship model. The movement of the vessel could be controlled by using mouse clicks.

The system can also display the Voronoi mesh on the sea surface. In this mode points inside the boats, points on the coastline, and Voronoi diagram edges can be seen, and the user can watch real-time Voronoi diagram changes, as points move through water mesh (Figure 3). This mesh is used for collision detection and obstacle avoidance.

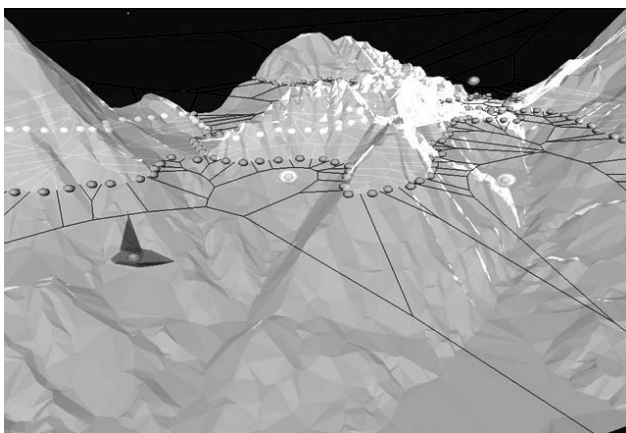


Figure 3. The Marine GIS in Voronoi diagram display mode

6. IMPLEMENTED FEATURES

- Depth testing

Each boat can monitor the depth of water under itself. The depth can also be checked at any point on the water surface. Given the desired location, the height of the underlying mesh at that point can be obtained by using natural neighbour interpolation (Sibson, 1981, Gold, 1989).

- Deepest channel navigation

Because of the depth monitoring feature, and the automatic boat control, the boat can be navigated to the deepest channel over the chosen path between the current location and a given target point.

- Collision detecting and preventing

While moving a boat through the Voronoi mesh, collision with another point could occur. The mesh can detect this and initiate steps to avoid it. It could be enough merely to change the point's/boat's direction, or speed. If a collision with another boat is likely then the directions and speeds of both boats can be changed, and restored after boats safely pass each other. If a coastline data point is an obstacle the boat should be redirected to prevent running aground.

- Changing water level (handling of 2 meshes)

Changing the water level, to simulate tides, can be done even during the animation. The water mesh has to be reconstructed after such a change, as the coastline points will change. The underlying mesh has to be scanned, and every triangle of it has to be checked to see if it is being intersected by the horizontal plane of the new water level. It is necessary for points to be distributed quite uniformly and frequently enough along all coastlines, to make sure that no boat can pass through the coastline without a collision being detected.

7. A PILOT APPLICATION – THE “PILOT BOOK”

Perhaps the ultimate example of a graphics-free description of a simulated real world is the Pilot Book, prepared according to international hydrographical standards to define navigation procedures for manoeuvring in major ports (UK Hydrographic Office, 2001). It is entirely text-based, and includes descriptions of shipping channels, anchorages, obstacles, buoys, lighthouses and other aids to navigation. While a local pilot would be familiar with much of it, a foreign navigator would have to study it carefully before arrival. In many places the approach would vary depending upon the state of the tides and currents. It was suggested that a 3D visualization would be an advantage in planning the entry into the harbour. While it might be possible to add some features to existing software, it appeared more appropriate to develop our own 3D framework. (Ford, 2002) demonstrated the idea of 3D navigational charts. This was a hybrid of different geo-data sources such as satellite pictures, paper chart capture and triangular irregular network data visualized in 3D. The project concluded that 3D visualization of chart data had the potential to be an information decision support tool for reducing vessel navigational risks. Our intention was to adapt IHO S-57 Standard Electronic Navigation Charts (International Hydrographic Bureau, 2000, 2001) for 3D visualization. The study area was Hong Kong's East Lamma Channel.

Our own work, based on the ongoing development of GeoScene, was to take the virtual world manipulation tools already developed, and add those features that are specific to marine navigation, and in particular those features identified in the IHO S57 standards.

So, on top of GeoScene, a general “S57Object” class was created, and sub-classes were created for each defined S57 object. These included navigational buoys, navigational lights, soundings, depth contours, anchorage areas, pilot boarding stations, radio call-in points, mooring buoys, traffic separation scheme lanes, traffic scheme boundaries, traffic separation lines, precautionary areas, fairways, restricted areas, and underwater rocks. Other objects include ship models, sea area labels, 3DS models, range rings and bearing lines, and oil spill trajectory simulation results. Various query functions were implemented, allowing for example the tabulation of all buoys in an area or the selection of a particular buoy to determine its details. Selecting “Focus” for any buoy in the table moves the window viewpoint to that buoy. Safety contours may be displayed along the fairways (Fig. 4), and a 3D curtain display emphasizes the safe channel. Fog and night settings may be specified, to indicate the visibility of various lights and buoys under those conditions. Safety contours and control markers may appear illuminated if desired, to aid in the navigation. The overall result is a functional 3D chart capable of giving a realistic view of the navigation hazards and regulations. For example, Fig. 5 uses 3D labels for sea area identification, Fig. 6 shows details of navigation buoys and lights, Fig. 7 shows the results of querying a particular buoy in the 3D display, and Fig. 8 shows the result of adding oil-spill simulation.



Figure 4: Visualization of Safety contour with vertical extension

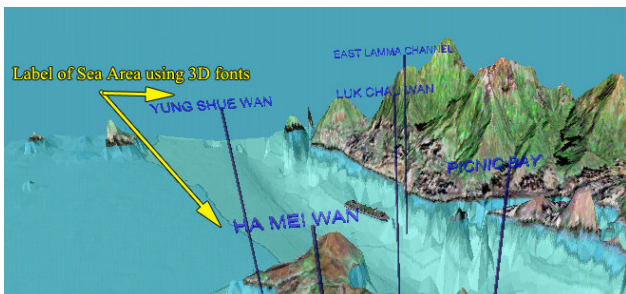


Figure 5: Visualization of Sea Area Label Using 3D fonts

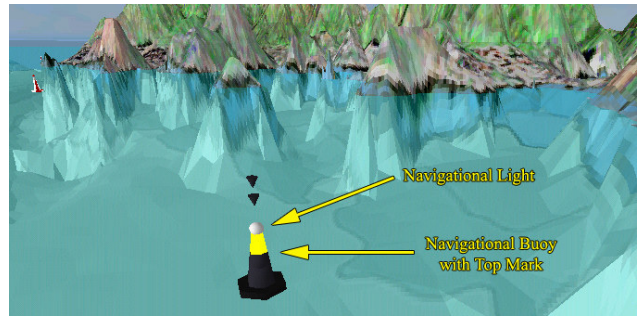


Figure 6: Visualization of Navigational Buoy and Lights

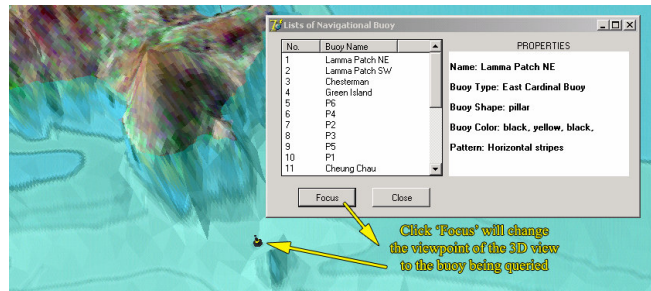


Figure 7: Lists of Navigational Buoys

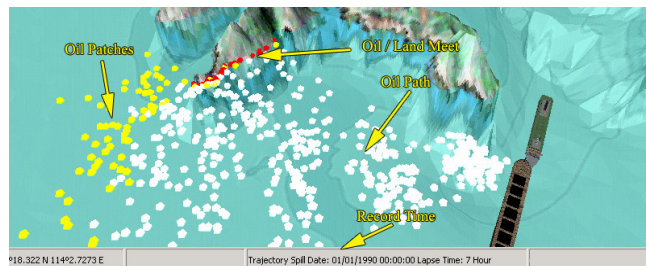


Figure 8: Visualization of Oil Trajectory Record

8. CONCLUSIONS

In this paper we have outlined the difficulties in developing a real “Marine GIS”, and our attempts to respond to the problems. We believe that a combination of modern interactive visualization tools and the appropriate kinetic spatial data structures is a significant step in the right direction.

9. ACKNOWLEDGEMENTS

The first author would like to acknowledge the Research Grants Council, Hong Kong, project PolyU 5068/00E for the support of this research.

10. REFERENCES

- Aurenhammer, F., 1991, Voronoï diagrams - a survey of a fundamental geometric data structure. *ACM Computing Surveys*, **23**, pp. 345-405.
- Blinn, J.F, 1977. A homogeneous formulation for lines in 3-space. *Computer Graphics* v.11 no. 2, pp.237-241.
- Davis, B.E. and Davis, P.E., 1988, Marine GIS: Concepts and considerations. *Proceedings, GIS/LIS'88*, San Antonio, TX, USA, pp. 159-168.

- Foley, J.D., van Dam, A., Feiner, S.K. and Hughes, J.F., 1990. *Computer Graphics, Principles and Practice, Second Edition*. Addison-Wesley, Reading, Massachusetts.
- Ford, S.F., 2002. The first three-dimensional nautical chart. In: Dawn Wright, ed., *Under Sea with GIS*, pp.117 -138, ESRI Press, 2002.
- Gold, C.M. 2000. An Algorithmic Approach to Marine GIS. *Marine and Coastal GIS*. Taylor & Francis (37-52).
- Gold, C.M. and Condal, A.R., 1995, A spatial data structure integrating GIS and simulation in a marine environment. *Marine Geodesy*, **18**, pp. 213-228.
- Gold, C.M., 1989, Chapter 3 - Surface interpolation, spatial adjacency and G.I.S. In *Three Dimensional Applications in Geographic Information Systems*, edited by Raper, J., (London: Taylor and Francis).
- Gold, C.M., 1991, Problems with handling spatial data - the Voronoï approach. *CISM Journal*, **45**, pp. 65-80.
- Gold, C.M., 1996, An event-driven approach to spatio-temporal mapping. *Geomatica*, **50**, pp. 415-424.
- Gold, C.M., 1999 An algorithmic approach to a marine GIS, In: Wright, D.J. and Bartlett, D.J., 1999, *Marine and Coastal Geographical Information Systems*, London, Taylor and Francis.
- Guibas, L. and Stolfi, J., 1985, Primitives for the manipulation of general subdivisions and the computation of Voronoï diagrams. *Transactions on Graphics*, **4**, pp. 74-124.
- Hill, F.S., Jr. 2001. *Computer Graphics using OpenGL*. New Jersey: Prentice Hall.
- International Hydrographic Bureau, 2000. *IHO Transfer Standard for Digital Hydrographic Data Edition 3.0, Special Publication No. 57*.
- International Hydrographic Bureau, 2001. *Regulations of the IHO for Internal (INT) Charts and Chart Specification of the IHO*.
- Li, R. and Saxena, N.K., 1993, Development of an integrated marine geographic information system. *Marine Geodesy*, **16**, pp. 293-307.
- Lockwood, M., and Li, R., 1995, Marine geographic information systems: What sets them apart? *Marine Geodesy* **18**:157-159.
- Oosterom, Peter J.M., van (1997) Maintaining Consistent Topology including Historical Data in a Large Spatial Database. *Proceedings Auto-Carto 13*, Seattle WA, 8-10 April 1997, pp. 327-336
- Rohlf, J. and Helman, J., 1994. Iris performer: A high performance multiprocessing toolkit for real-time 3d graphics. *Proceedings of SIGGRAPH 94*, pages 381-395.
- Sibson, R., 1981, A brief description of natural neighbour interpolation. In *Interpreting Multivariate Data*, edited by Barnett, V. (New York: John Wiley and Sons), pp. 21-36.
- Strauss, P.S. and Carey, R., 1992. An object-oriented 3D graphics toolkit. In E.E. Catmull, editor, *Computer Graphics (SIGGRAPH '92 Proceedings)*, volume 26, pages 341-349.
- The United Kingdom Hydrographic Office, 2001. *Admiralty Sailing Directions – China Sea Pilot Volume I*, Fifth Edition 2001, United Kingdom National Hydrographer, Taunton.
- Wright, D.J. and Goodchild, M.F., 1997, Data from the deep: Implications for the GIS community. *International Journal of Geographical Information Science*, **11**, pp. 523-528.
- Woo, M. et al., 1999. *OpenGL(R) Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*. Addison-Wesley, Pub.