Chapter 10
Visualization in Fisheries Oceanography: New Approaches for the Rapid Exploration of Coastal Ecosystems

Albert J. Hermann and Christopher W. Moore

10.1 Introduction

Powerful new measurement technologies and numerical models are rapidly expanding our three-dimensional knowledge of the ocean and its biota. The effective visualization of this expanding database is not trivial, and its use in fisheries oceanography presents special challenges. Oceanic circulation and the life experience of marine fish are irrefutably three dimensional; despite this fact, most ocean data and model output has traditionally been viewed using two-dimensional maps. Furthermore, a thorough analysis of any fish population ought to include both: (1) the local world experienced by representative fish (a pseudo-Lagrangian record of events, as in spatially explicit individual-based models); (2) the changing regional habitat of the population (an Eulerian description of the regional environment, as in spatially explicit lower trophic level models). In this chapter we survey visualization approaches, from the simple to the complex, which can facilitate such multifaceted analyses by researchers and managers. In particular we emphasize that modern approaches to scientific visualization include *immersive* techniques: by taking advantage of human binocular vision, these allow the user to experience and interact with measured and modeled features as "virtually real" objects in our three dimensional world. The experience is not unlike holding a part of the ocean basin in one's hand, turning and examining it from different angles. In fisheries oceanography, such modern approaches can rapidly and effectively reveal: (1) the structure of spatially patchy prey fields, such as phytoplankton; (2) three-dimensional flows near topography; (3) spatial tracks of individuals and the prey fields they experience along those paths. Recent developments allow immersive visualization through web servers, giving scientists the ability to collaboratively "fly through" three-dimensional data stored half a world away. We explore what additional insight is gained through immersive visualization, and describe how scientists of very modest means can easily

A.J. Hermann (✉)
Joint Institute for the Study of the Atmosphere and Ocean, University of Washington, Seattle, WA 98115, USA

avail themselves of low-cost immersive technology. Further examples and resources are available at http://www.pmel.noaa.gov/people/hermann/vrml/stereo.html.

10.1.1 3D Models vs. 2D Graphics

All fisheries oceanographers have a compelling interest in developing clear ways of visualizing their data. Modern observational networks generate huge volumes of oceanographic data; numerical models generate even larger volumes of output (whether this output should strictly be called "data" is a semantic argument, but for our purposes we will refer to it as such). How to visualize these enormous data files? Two-dimensional plots along horizontal or vertical planes have served oceanographers nobly in the past three centuries (from Ben Franklin's charts of the Gulf Stream to the present) but fail to capture many of the subtleties present in even two-dimensional data sets.

Consider the bathymetry of the ocean, which contains both large and small-scale features. Typically the large-scale features have a greater amplitude than the small-scale features. A simple contour plot of the ocean floor, with broad contour levels chosen based on the largest amplitude signals, will hence reveal the large scales but obscure the small (or alias them into the larger scales; see Fig. 10.1). By comparison, if we render the bathymetry as a surface in three-dimensional space, we reveal both large and small scales at once. In effect, a rendered surface is equivalent to having an infinite number of contour levels.

In modern times it is easy to render such a three-dimensional surface and examine it on a computer monitor, or to print the result on flat page. We have seen many such illustrations in popular scientific journals. However, consider how primitive even this technique really is. Billions of years of evolution yielded our ability to perceive the depth of an object (a talent shared with many other species); we can easily tell when an object is further away from us, rather than simply smaller. We possess this ability because we have two eyes and a brain to combine the information from each (stereo vision). Viewing a three-dimensional surface printed on flat page is like viewing the world with one eye closed – the depth perception which stereo vision affords us is lost. Viewing a three-dimensional object with two eyes from far away – think of a mountain range viewed from a high-altitude jet airplane – is really not much different from the single-eye view. A much more powerful way to examine a surface or other three-dimensional object is to view it "up close" with two eyes, from any angle, as if you were holding the object in your hands. With two eyes you clearly see which parts of the object lie in front of or behind other parts, and clearly distinguish up from down, left from right, and forward from
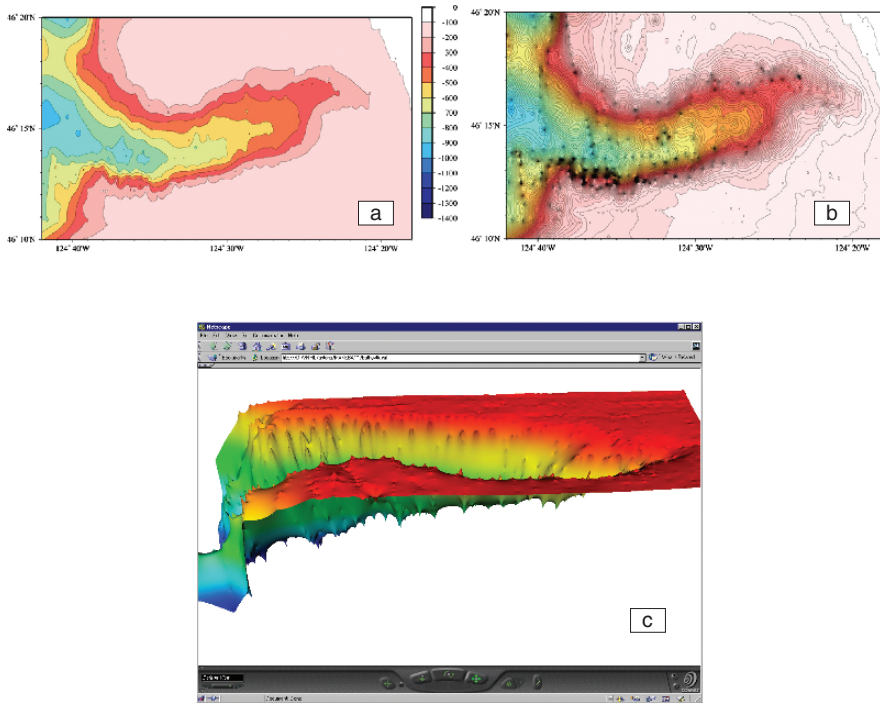
Fig. 10.1 Contour plots compared with surface rendering of gridded bathymetric data. These data contain significant sampling artifacts which include spurious fine-scale minima and maxima. A standard contour plot with only a few contour levels (**a**) misses these fine-scale artifacts. A contour plot with a very large number of contour levels (**b**) reveals many local features, but obscures their amplitude or sign. A rendered surface using all data points (**c**) simultaneously reveals both the large-scale structure and the small-scale irregularities. With proper software, this surface can be easily rotated and magnified; here, a VRML viewer was used for this purpose

backward. You cease to confuse small features as being further away simply because they are smaller. Regional planners have for many years utilized widely separated photographs of topography to achieve this "god-like" (albeit static) view of the land.

Obviously it is not practical to create a plastic model of every three-dimensional surface one might want to view, and hold it physically. Instead, modern computing hardware and software make it easy to generate images on your computer monitor which are virtually identical in appearance – that is, "virtual reality". It is obviously not "reality" in the sense of what you encounter in your daily life, but rather what you would encounter in your field of view if you were to create a solid model of the surfaces, vectors, and volumes representing various aspects of the data set you wish to analyze.

The quest for virtual reality is hardly new; 3-D photographs and movies have been with us for many years, beginning in the 19th century with stereo postcards. Their presence in popular culture has waxed and waned over time, including a few less-than-memorable films in the 1950s. The present popularity of IMAX 3D theaters suggests that 3D films have become a significant force in scientific education. What is new in the past decade is the tremendous power and low-cost of modern computing platforms, which allow interactive viewing of 3D worlds. Graphically speaking, what was inconceivable a decade ago has now become routine. Anyone with a low-cost personal computer can generate complex surfaces representing some aspect of a large (megabytes to gigabytes) data set – indeed, many do so on a daily basis as they play computer games. For a modest additional investment one can acquire the ability to view those surfaces in immersive stereo. In effect, one has the ability to fly through and rotate the virtual world with both eyes wide open. It is hard to describe the power of this illusion in words; picture yourself as a giant swimming through the ocean, able to view whatever feature you might choose, up close, far away, and from any angle.

Our goal in this realm might be termed "scientific Virtual Reality for the masses". We are seeking ways to allow immersive visualization of numerical model output (or data) through inexpensive methods, powerful enough to handle large worlds and networked in a way which allows access to large datasets and collaboration with remote colleagues. We and our colleagues have not yet achieved all of these goals, but have made significant progress. Here, we begin with examples of fisheries phenomena whose illustration can benefit from the use of 3D graphics. We follow this with a brief summary of basic graphic concepts, and then a description of low-cost hardware and software which can be used to generate, explore, and interact with such immersive worlds.

10.1.2 Examples from Fisheries Oceanography

Real fish live in a three-dimensional world, with complex fluid velocities and spatial paths. Here we present several examples of 3D features and phenomena relevant to fisheries oceanography, and emphasize how 3D visualization can communicate fundamentally new information, relative to simple 2D visualization techniques. Features of interest include the physical environment (e.g. seafloor bathymetry, currents, temperature, nutrients, and passive particle tracks), prey fields (e.g. zooplankton) and fish life histories (e.g. position, life stage, and size). Three dimensional visualization, and immersive visualization in particular, allow the fisheries scientist to quickly examine velocity fields and the animated spatial paths of measured or modeled fish in their native environment. Immersion facilitates the exploration of such data sets by quickly revealing spatial relationships (left from right, up from down, and closer vs. further away).

10.1.2.1 Bathymetry

Ocean bathymetry contains features on many spatial scales which affect circulation and fish life history. Simple contour plots tend to reveal only the large-scale features and hide the smaller ones. Stated more technically, a contour plot may show the global minimum and maximum of a particular area, but hide many significant local minima and maxima. In Fig. 10.1 we compare such simple contour maps with a rendered surface. Note how the rendered surface more clearly revealed many small scale errors (in this particular case, the errors were due to biases of a regridding scheme which had been used to interpolate from irregularly spaced bathymetry data to a regular latitude-longitude grid). If this bathymetry were used "as is" in a numerical model, the errors could have a substantial effect on the results. Here, rendering the data as a surface flagged these errors beforehand.

10.1.2.2 Fish Life History

Fish eggs and larvae are essentially planktonic organisms, advected passively by the ocean currents. This advection is supplemented by vertical motions due to buoyancy and directed swimming. In both the real ocean and in numerical models, this can yield complicated three-dimensional trajectories (a.k.a. "individual life histories") through time. We have generated many such life histories in our modeling work (Hermann et al. 1996, 2001, Hinckley et al. 1996, 2001). This 3D information is only partially revealed when only two dimensions are displayed. Colored polygons can be used to represent the third dimension of depth on a 2D contour plot of bathymetry, but fail to give a visceral sense of a complex looping path through three dimensions and simultaneously communicate biological attributes of the fish. It has been our experience that the animation of a full 3D world is a far more instructive way to communicate such complex life histories (Fig. 10.2).

10.1.2.3 Circulation and Hydrography

Both real and modeled oceans exhibit motions across a broad swath of spatial scales – indeed the real ocean exhibits scales of motion from thousands of kilometers down to a few millimeters. As a result both velocities and physical scalars such as temperature and salinity exhibit patchiness at many scales. Examples of 3D circulation phenomena familiar to fisheries scientists include coastal upwelling (which provides nutrients to the food chain), tidal motions (which provide nutrients through mixing, and interact with behavior to affect fish migration), and mixed layer turbulence (which affects feeding). We can represent some of these phenomena more effectively through the use of 3D isosurfaces and 3D vectors (see Sections 10.2.1.1, 10.2.1.2 and 10.2.1.3).
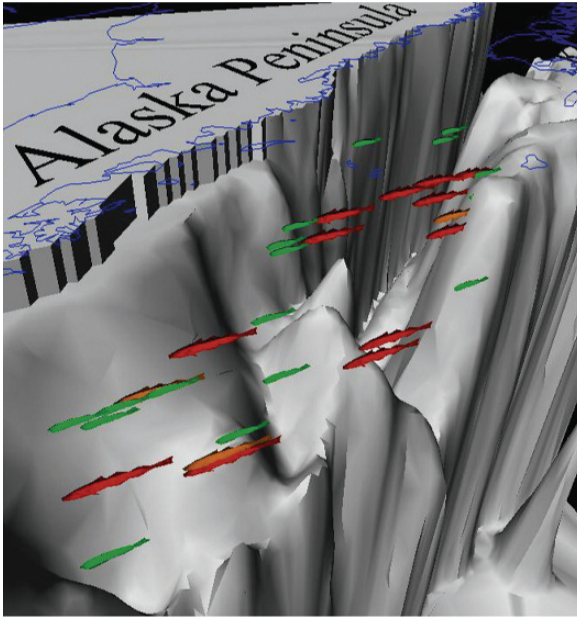
Fig. 10.2 Example of 3D rendering to illustrate modeled walleye pollock (*Theragra chalcogramma*) life histories near Kodiak Island (in *upper right* of figure). *Grey* surface is ocean bathymetry used by the hydrodynamic model; maximum depth shown here (*lower right*) is ∼500 m. *Blue line* indicates coastline. Shape and color of fish indicate size and life stage. This frame, derived from the animation of a 6-month biophysical model hindcast, represents the population in mid-June, 1978. Animation of this world illustrates looping trajectories which would be obscured by simple 2-D graphics. Immersive viewing of this scene provides additional depth cues

10.1.2.4 Prey Fields

Many fish consume planktonic species as a major prey item. Plankton fields are subject to the wide range of motions in the ocean; they are further affected by reproduction, predation and death. As a result, plankton fields can in fact be *even patchier* in space than physical scalars (e.g. Powell et al. 2006); like clouds in the atmosphere, they can be richly patchy in 3 dimensions. Contour plots fail to convey this structure as well as 3D graphics. Indeed, a 2D contour plot of atmospheric moisture reveals far less information (or at least very different information) about the spatial structure of clouds than a simple oblique view from the ground.

10.2 Computer Rendering of 3D Biophysical Worlds

Here we describe basic objects rendered in 3D graphics, and the hardware needed to render them. This discussion is not specific to immersive visualization, but the advantages of immersion are noted where relevant.

10.2.1 3D Graphics Concepts

10.2.1.1 Isosurfaces

An isosurface is the 3D equivalent of a 2D contour. A contour plot exhibits lines connecting points along a 2D surface which all have the same numerical value of some property. The most common example is a topographic map which shows lines of constant elevation of the land surface. Oceanographers frequently use contour maps of sea surface height or sea surface temperature. An isosurface is a surface connecting points which all have the same value of some property in *three* dimensions. For example, an equatorial oceanographer might wish to examine the surface along which the ocean has a temperature of 10°C, to diagnose El Nino dynamics. A fisheries oceanographer could use similar information to examine the habitat of some species of fish which prefers a particular range of temperatures or prey density; the isosurface illustrates the boundaries of that habitat.

10.2.1.2 Volume Rendering

In some circumstances the most effective way to represent a patchy 3D field is to render it opaque where values are highest. In nature, we see clouds where the density of water vapor is high; on a computer we may render virtual "fog" (and color it) where a scalar property exceeds a particular value. This is a very natural way to experience a patchy distribution of chlorophyll, for example. In Fig. 10.3 we demonstrate our use of these approaches to explore modeling results for the Gulf of Alaska (Hermann et al. 2008, Hinckley et al. 2008).

10.2.1.3 Vectors

Oceanographers frequently represent velocity as a vector – graphically, as an arrow whose length and orientation illustrate the speed and direction of the currents at a particular location. Such arrows are usually drawn to represent horizontal velocity: eastward velocity, u, and northward velocity, v. 3D vectors (u, v, and vertical velocity, w) can be rendered as well. Such 3D vectors can be confusing on a 2D page, as it is difficult to discern whether a vector is actually short, or merely pointing away from the viewer. In such cases, the ability to rotate the 3D image on a computer, and/or to view it immersively, provides valuable depth cues which 2D plots lack.

10.2.1.4 Lighting

Shadows cast by 3D features help reveal their spatial structure. Typically a scene is rendered using a single lighting source, which emanates from the coordinates of the viewer. Multiple lighting sources from the sides can add additional shadows and cues, just as in the real world.
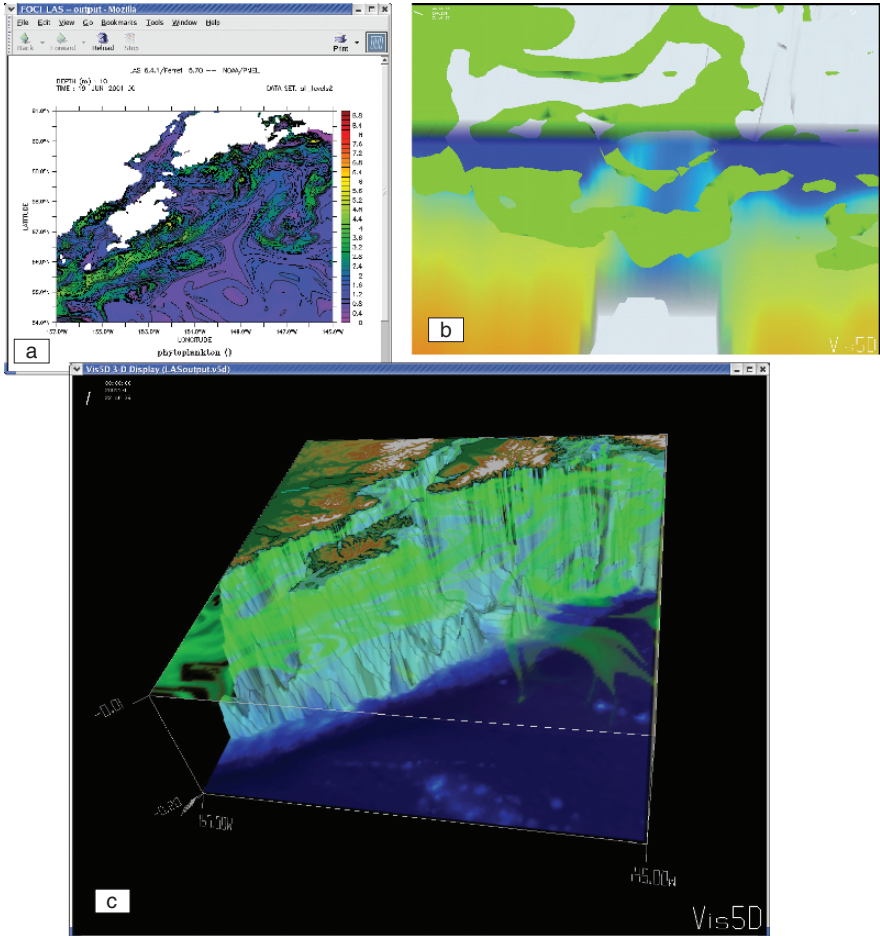
Fig. 10.3 Modeled chlorophyll in the Gulf of Alaska, rendered using 2D versus 3D graphics. (**a**) Shaded contour plot of surface chlorophyll. (**b**) Oblique 3-D view (looking downward and to the southeast) of biological properties near Portlock Bank, southeast of Kodiak Island. Chlorophyll isosurface (*green*) is shown superimposed on bathymetry (*grey*) and a vertical slice of nutrients (*blue/orange* palette denotes low/high nutrient concentrations). (**c**) Oblique 3D view of chlorophyll rendered as a virtual fog. Note how 3D views simultaneously convey horizontal and vertical extent of patches. Note also how the isosurface view (**b**) is confusing without immersion or animation to provide spatial cues

10.2.1.5 Navigation

Quick navigation – that is, easily changing one's point of view – is essential for rapid exploration of complex 3D worlds. Indeed, this "ease-of-use" issue frequently makes all the difference between popular and unpopular software. In the natural world, we (and many predators) alter our point of view to get a

better sense of the spatial layout of our environment. Consider how a hunting cat bobs and weaves its head from side to side to judge the distance to its intended prey. In a virtual world the viewer may "fly" or "walk" through a rendered world, or manipulate an object in that world. These actions are accomplished by keyboard, mouse, or other input device.

10.2.1.6 Animation

Animation of a 3D world has a double benefit. First, and most obvious, it reveals the path of an object (e.g. an individual modeled fish) through time. Second, it provides additional clues as to the location of objects, even if they are not appreciably mobile. In a sense, it is equivalent to changing one's point of view for examination of a stationary object.

10.2.1.7 Collaborative Viewing

Sophisticated yet inexpensive technologies now exist for collaborative viewing of 3D worlds. Such approaches are a natural extension of shared desktop software, and in fact share much in common with interactive online gaming. Collaborative viewing allows researchers at multiple institutions to examine and discuss the same 3D world. Viewers may be represented as "avatars", similar to virtual characters in a multiplayer game.

10.2.2 3D Computer Hardware Issues

10.2.2.1 Central Processing, Memory, and Graphics Processors

As of this writing, anyone with a modern commodity PC can render and explore complex 3D worlds. Obviously, the more powerful the computer, the faster any rendering or navigation through a 3D world can take place. Less obviously, the bottleneck in such rendering may exist in the raw CPU, the memory to which it has fast access, or the power of the graphics processor. The latter typically exists as a separate element of the computer, and is responsible for the persective rendering of 3D objects (or more precisely, the individual polygons which make up a 3D digital object), given their location, color and texture. Driven by the relentless market for better and faster computer games, inexpensive yet power-ful graphics cards have become widely available to consumers. Fisheries scientists obviously benefit from these developments.

10.2.2.2 Data Access

A large amount of memory is required to render large artificial worlds in fine spatial detail. Some graphics programs (and many computer games) deal with this problem by storing on disk both coarse and fine-resolution versions of the

world. The coarse version is loaded into memory and rendered when the observer is looking at the large-scale structure of the world (i.e. is "far away" from the world); fine-scale information is loaded into memory as necessary when the observer zooms in on a smaller region of the world. Data may be stored locally or on a remote server. Several popular geography and astronomy viewers utilize such interactive "data on demand" transfers. Extensive data sets are stored on a remote server; only those portions being viewed are downloaded to the local machine. In some cases the software anticipates and pre-fetches what data will be needed next, based on the trajectory of the viewer.

10.2.2.3 Parallel/Distributed Rendering

Careful exploration of a virtual world requires both a broad spatial view and the ability to zoom in on fine detail. Even with efficient access to remote data, the rendering of complex worlds can be very taxing on a single CPU or graphics processor. One approach to handling complex worlds is to farm out separate pieces of the rendering to separate processors. This can be effected on multiple processors of a single computer, or even on widely separated processors connected through a network. These are stitched together by a central processor, or a central graphics processing unit.

10.2.2.4 Navigation Hardware

In addition to keyboard and mouse control, navigation through virtual worlds can take place with simple game controllers, e.g. joysticks (Fig. 10.4). In some circumstances this provides the most natural way to "fly" through 3D worlds. In other cases, features are best explored through rotating the world with a standard mouse. Many specialized controllers have been built to facilitate navigation through three dimensions. Recently, simple handheld devices have become popular for electronic versions of games such as baseball and tennis. These measure the position, orientation, and acceleration of the user's arm to manipulate virtual bats and rackets; we can easily imagine potential uses for scientific software as well.

10.3 Stereo-Immersive Approaches to Visualization

Here we focus specifically on the hardware and software requirements for *immersive* visualization, which take advantage of the natural human capacity for binocular ("stereo") vision. The hardware and software needed to render and immersively display 3D worlds, formerly available only at sophisticated visualization centers, is now accessible to scientists of modest means. Ever more sophisticated (and high-cost) technologies will doubtless emerge; here we focus on lower cost gear readily available today through local commodity computer

Fig. 10.4 Example of immersive joystick navigation through a virtual world on a laptop computer

dealers. Broadly speaking, the minimal requirements for a single user are a reasonably powerful CPU and a graphics card with a stereo driver. Immersive viewing by a group can be achieved with the addition of two projectors and an appropriate screen.

The basic approach with all immersive techniques is to render two snapshots of the scene you wish to view, one for your left eye and one for your right. The differences between the two snapshots depend on how far away the viewed object is, and how widely spaced your "virtual" eyes are, relative to the size of the object. Once you have the two snapshots, you need to deliver each of them to the proper eye. The old technique for doing this, very popular around the beginning of the 20th century, is to produce an oversized postcard with the left eye's view on the left half and the right eye's view on the right half. A simple mechanical device then focuses each snapshot to the proper eye: the stereograph. With some training (but typically with some eye strain) you can focus the proper image from a stereo pair on the proper eye without any mechanical assistance. Sometimes pairs are rendered with the left eye's view on the right and vice versa; the viewer then crosses her eyes to focus the proper image on the proper eye. Again, this involves some eye strain.

10.3.1 Stereo Hardware

On present-day computers, the task of generating a 3D scene of graphical objects is handled by the CPU, central memory and graphics processor working together; the task of rendering two different views of that scene (and possibly sending them to different monitors or projectors) is typically handled by the graphics processor itself. Here we describe three different common approaches to stereo visualization on computers: anaglyphs, active shutterglasses, and passive polarization.

10.3.1.1 Red-Blue Anaglyphs

One simple technique for achieving immersive 3D is to print both the left and right eye's view of a greyscale scene on top of one another, but in different colors (see Fig. 10.5). Such pictures are called *anaglyphs*. Inexpensive glasses with colored filters then screen out the left eye's picture from the right eye and vice versa. This was the basic approach for 3D movies and comics of the 1950s.
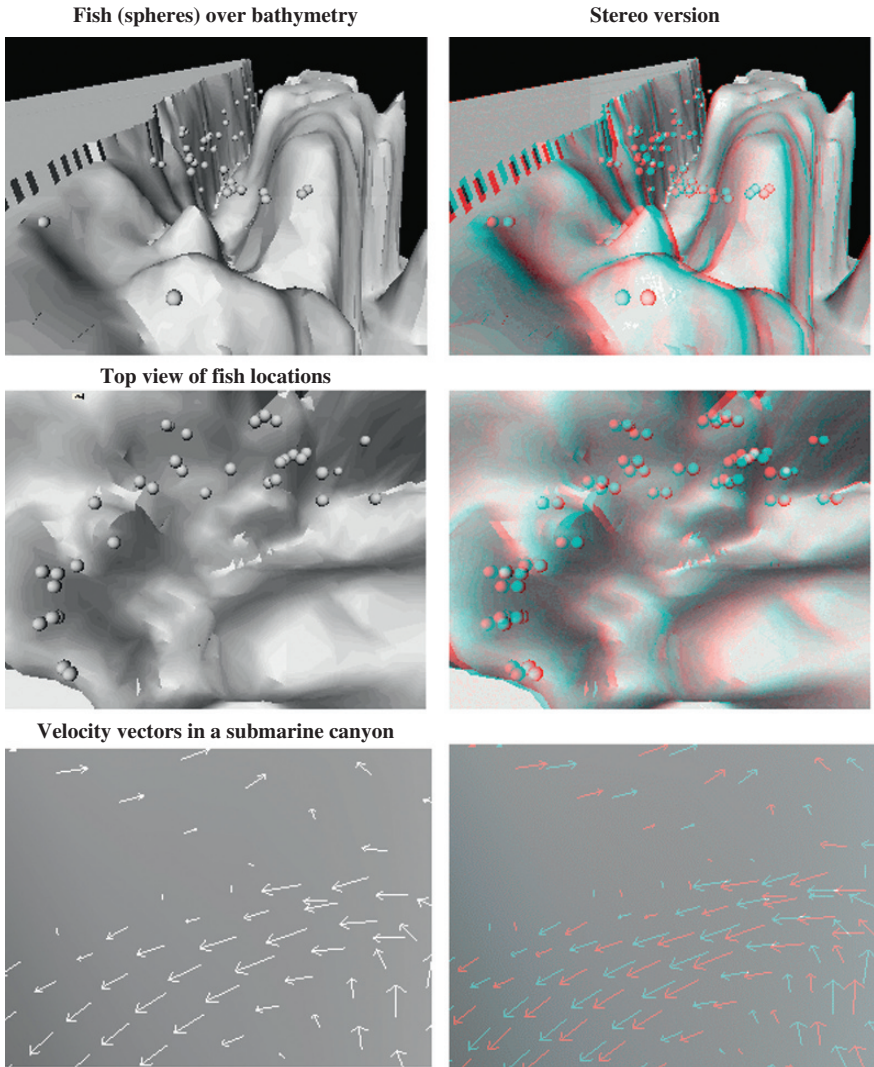

Fig. 10.5 Comparison of stereo and non-stereo images of particles and vectors in 3D worlds. For stereo viewing use *red/blue* anaglyph glasses with the *red* lens over your *right* eye and the *blue* lens over your *left eye*

Anaglyphs are in fact easy to display on computer monitors, and the results are suprisingly realistic for greyscale worlds. A limited degree of color is possible in such worlds as well. Shown in Fig. 10.5 are some examples of virtual worlds and greyscale stereo anaglyphs we have created from biophysical model output. The top figures illustrate positions of fish (represented as spheres) from a spatially explicit 3D individual-based model of walleye pollock (*Theragra chalcogramma*) in Shelikof Strait, Alaska (Hermann et al. 1996, Hinckley et al. 1996). Kodiak Island is the prominent feature in the right half of the top figure; our viewpoint looks to the northeast. The bottom figures are from a circulation model of Pribolof Canyon in the Bering Sea (Hermann et al. 2002). A cheap pair of red/blue glasses will cause the spheres and vectors to pop out of the page at you, revealing the true 3D location of the modeled fish, the steep slopes of the bathymetry, and the vertical motions near the submarine canyon.

10.3.1.2 Active Shutterglasses

In the active shutterglass approach, still images and animations appear in full color on a computer monitor or projection screen. The approach is based on the fact that the computer, like a television set, refreshes the image on the screen many times per second. The computer alternately displays the left eye's view, then the right eye's view, while electrically controlled lenses (the application of a voltage renders them opaque) alternately allow the left eye to view the display, then the right eye, in sync with the alternating images. The synchronization signal is sent to the glasses by wire or through an infrared emitter. If the images are alternated quickly enough (say, 120 times per second), the viewer sees a steady image in stereo.

There are three common techniques used with shutter glasses. (1) Frame Flipping: all available pixels on the monitor (or projector) are used to render each eye's image in sequence. This approach was used in many high-end visualization environments through the 1990s, e.g. the VR-CAVE and the IMMERSADESK. (2) Interleaving: the monitor alternately displays the left and right eye's view using the even-numbered columns (or rows) of pixels for the left-eye view, and the odd numbered columns (or rows) of pixels for the right-eye view. This requires less graphics memory than the frame flipping approach, but results in a lower resolution image for a given monitor. (3) Synch Doubling: the computer generates the left eye view on the top half of the screen, and the right eye view on the bottom half of the screen. A *synch doubler* then spreads out the top half image to fill the entire screen, and does the same with the bottom half image – they are alternately displayed at half the original refresh rate of the monitor. This requires no special graphics hardware within the computer itself, but as with interleaving results in a lower resolution image than frame flipping.

A variation of the interleaving approach continuously displays left/right views on the odd/even columns, and deploys special lenses on the monitor itself

to direct odd/even columns to the left/right eye of the viewer. This auto-stereo technique requires no special glasses, but typically requires the viewer to sit at a fixed location in front of the screen.

10.3.1.3 Passive Polarization: The Geowall

A third and very appealing technique entails projecting the different images through two projectors fitted with polarizing filters, effecting different polarization for the left versus right eye's view (Leigh et al. 2001). The two images are aligned and projected onto a polarization-preserving screen (see Fig. 10.6). Simple polarized lenses (much like sunglasses) are then used for stereo viewing, delivering the proper image to each eye. Unlike most anaglyphs, this allows full color images to be displayed. The low cost of both the projectors and the screen has made this approach very affordable; indeed, it has now become an attractive alternative to the shutterglass systems described in Section 10.3.1.2. DLP (Digital Light Processing) projectors are used because they naturally emit unpolarized light, which is subsequently polarized by the filters. This arrangement is usually termed a "Geowall", and many hundreds are now in use worldwide (for a partial list, see the Geowall consortium webpage: http://geowall.geo.lsa.umich.edu/). The system is compact and portable, can be installed in a single investigator's office (Fig. 10.7), and has been very efficient for display of oceanographic and astrophysical data at
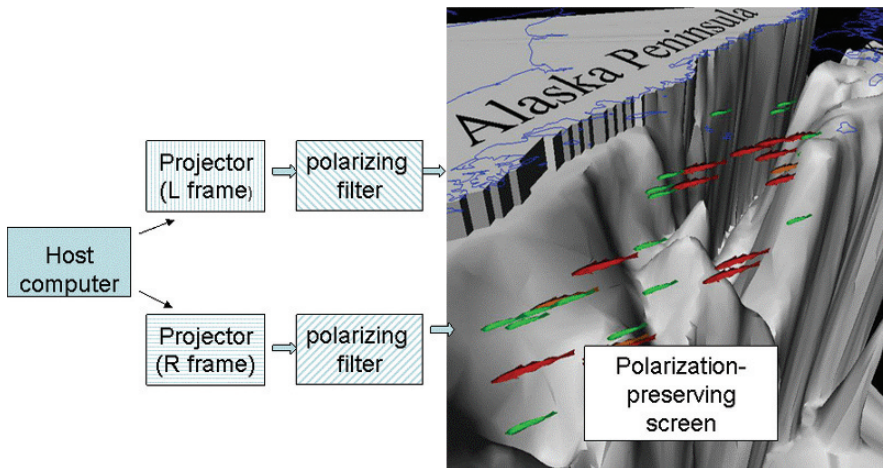

Fig. 10.6 Schematic of passive, dual-projector stereo display system (a "Geowall"; see http:// geowall.geo.lsa.umich.edu/). Components include host computer (commodity PC or workstation), DLP projectors, polarizing filters, and polarization-preserving projection screen. Passive polarized glasses are worn by the viewer, to deliver the proper image to each eye. *Thick arrows* denote light path

Fig. 10.7 Setup of a Geowall system in a modest-sized office. Projectors with filters are placed at the back of the office, and driven with a linux PC outfitted with a graphics card supporting two displays. A polarization-preserving pull-down screen was installed in front of the window at the opposite end of the office. A model of the North Pacific (Curchitser et al. 2005) is displayed on the screen
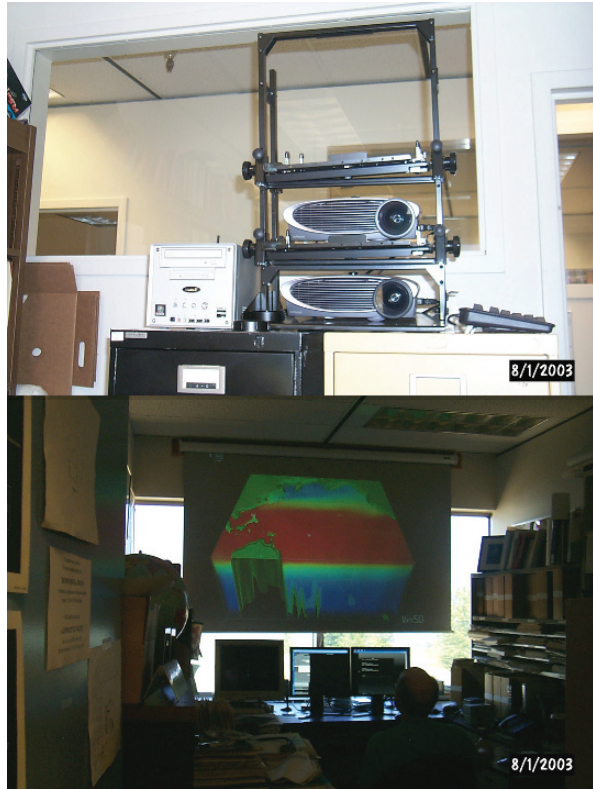
conferences, workshops, and science fairs (Hermann and Moore 2004). The passive polarization approach has grown rapidly in popularity due to its simplicity and low cost, and is now the basis of many large-scale IMAX 3D theaters.

10.3.2 Stereo Software

A large number of commercially available packages will render 3D worlds in stereo using the hardware described above. One of the earliest (and in our view still one of the best) open-source visualization packages for gridded data was originally developed by Bill Hibbard and colleagues at the University of Wisconsin: Vis5d (Hibbard and Santek 1990; http://vis5d.sourceforge.net/), which is featured in Figs. 10.3 and 10.7. When the C code for this package was originally developed, few computers could run the code speedily, and inter-active viewing was a frustrating endeavor for all but the smallest of grids.

However, times have changed (i.e. the hardware has caught up with the software), and modern PCs running Linux can execute this program handily with smooth navigation for grids with a million or more gridpoints. A related Java-based toolbox, VisAD (Hibbard 2002; http://www.ssec.wisc.edu/~billh/visad.html), has been used to develop software such as the Integrated Data Viewer (http://www.unidata.ucar.edu/software/idv/) with similar 3D capabilities. Such Java-based tools have the ability to communicate with large data-sets through the Internet, which can be of great advantage when data is not stored locally. Our own experience has been that Java-based tools are slower than the old C code-based visualization of locally stored data – provided it all fits in local memory!

10.3.2.1 3D Through the Web: Virtual Reality Modeling Language

The Virtual Reality Modeling Language (VRML) is a scene description language that describes three-dimensional environments in a simple ascii format, and which allows users to access, navigate, explore and interact with environmental data in three dimensions on the Web. An international open standard was accepted for VRML in 1997 (ISO/IEC 1997). This language has been very useful for the visualization of 3D oceanic data (Moore et al. 2000a, 2000b). Many of the virtual worlds shown in this chapter were constructed using this standard. VRML is scalable across platforms ranging from PCs to high-end workstations, and can be viewed either with a Web browser plug-in or with stand-alone software. A VRML world typically consists of polygonal surfaces that mimic the real environment. Objects rendered for oceanographic/atmospheric analyses include contoured and shaded slices, vector fields, isosurfaces, and spheres representing drifters or organisms. These objects can be touched, rotated, or animated using controls that the browser provides. VRML objects can be primitive (cubes, spheres, etc.), or user-defined (elevation grids, polygonal surfaces, lines), and can be given traits such as color, texture, sound, and video. Animations can be created by swapping surfaces of arbitrary shape using cached memory or morphing a surface by changing its defining coordinates over time. The user can define touch sensitive objects and assign actions (typically through simple Javascript routines) that allow the user to interact with the world. VRML can also interact with Java to create a myriad of 3D and 2D user interfaces. Free VRML plug-ins are currently available, and some PCs already have this capability preinstalled.

A VRML file represents three-dimensional objects as a series of coordinates, independent of perspective; hence there are not stereo VRML *files* per se, but rather stereo *viewers* of VRML files. As noted earlier, the graphics card (with an appropriate driver) typically renders the two views of a particular scene. VRML worlds may be generated from popular data analysis software, such as Matlab. We have developed a web interface to serve up
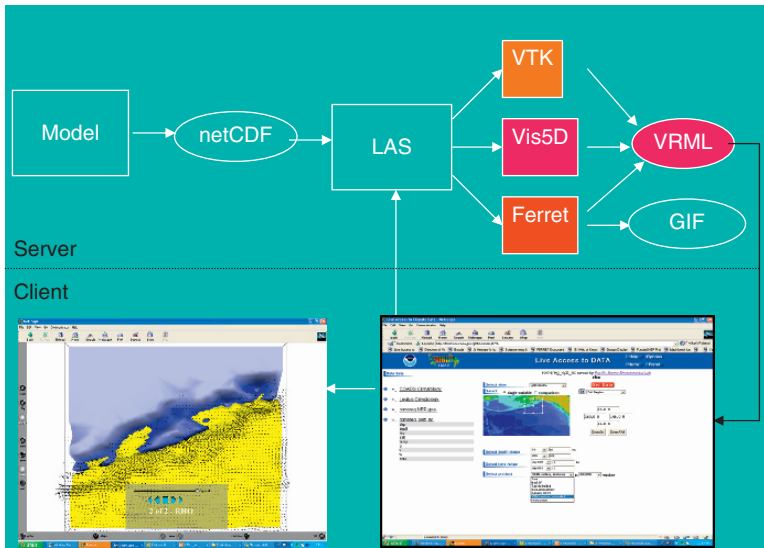
Fig. 10.8 Implementation of Live Access Server for remotely stored data, delivering immersive graphic on demand (*red/blue* anaglyph) to viewer. Model generates data in netCDF file format, stored on remote server. Live Access Server (LAS) extracts graphical objects (here, isosurface of salinity and velocity vectors), using graphical software (VTK, Vis5D, Ferret) running on the server. Graphical objects are transferred as a VRML file, for immersive display on the local client

VRML worlds on demand for some of our data sets, through a Live Access Server (Hermann et al. 2003; Fig. 10.8).

10.3.3 Stereo APIs

There are presently several software standards (Application Program Interfaces, APIs) for graphical manipulations at the level of computer hardware. Open Graphics Language (OpenGL) has emerged as a standard of choice for scientific applications, and many "professional" graphics cards now support it. Most modern graphic cards feeding the huge gamer's market support an API known as DirectX. As noted earlier, such cards perform mathematical manipulations (matrix multiplications, ray tracing) needed for rotating and rendering objects on your display; this is more efficient than having to calculate everything on your CPU. It is worth noting that there are presently two different "flavors" of stereo, one based on DirectX (full-screen-stereo) and one based on OpenGL (stereo-in-a-window). Most consumer graphics cards (and stereo-enabled games) support at least the former; many currently available graphics cards are capable of rendering in either of these modes.

10.4 The Future of Modeling and Visualization

Our ability to model the ocean and its fisheries at relevant scales is partly dependent on the raw speed of computer hardware. According to Moore's Law, computer processing speed increases ∼8-fold every 5 years. Our ability to resolve horizontal scales in the global ocean will double in that time frame, as it requires four times as many grid points and twice as many time steps for numerical stability. Our ability to store data inexpensively appears to be increasing even faster than Moore's Law; at the time of this writing, the cost of storage is approximately \$0.15/GB. Clearly, both model output and measured data will grow exponentially in the coming decades, and effective exploration of these results will pose a great challenge. Consider the real ocean: we cannot measure everything at once, and limit our attention to what is easily observed (e.g. surface properties) or what contains the largest information about the system as a whole (e.g. subsurface current measurements which are known to be well-correlated with larger circulation patterns). In the modeled ocean, we are not limited by technology to any particular depth, and the sheer immensity of what could be explored is daunting. Immersive visualization is a tool to facilitate more rapid exploration of real and modeled worlds, to help identify the new and unusual phenomena contained therein. It does not replace traditional graphics, but rather adds a powerful exploratory tool for research, collaboration, and communication of results to the public.

While the raw processing speed of a single CPU has multiplied, the speedier execution of numerical models in this decade is in fact largely due to new massively parallel architectures, and more efficient parallelization of code. New visualization hardware and software are taking advantage of this parallelism as well. The emergence of dual- and quad-processor PCs, and ever more powerful graphics processing units, will further accelerate the graphical capabilities of a single desktop machine in the laboratory or office. Increasingly this obviates the need for remote (and potentially expensive) dedicated graphic centers. Indeed, it may be that immersive visualization will become a routine and widely available tool for many scientific endeavors.

As computer hardware grows more powerful and less expensive, larger datasets will be rendered in finer detail. New (e.g. holographic) techniques will likely be developed, which will eliminate the need for special glasses of any type. If the past 10 years are any guide, the driving economic force behind these new technologies will be the computer gaming market; in particular, the market for massive multiplayer online games. As with the powerful graphics hardware already available, it behooves the scientific community to take advantage of these new technologies as they emerge.

References

Curchitser EN, Haidvogel DB, Hermann AJ, Dobbins EL, Powell TM (2005) Multi-scale modeling of the North Pacific Ocean I: Assessment and analysis of simulated basin-scale variability (1996–2003). J. Geophys. Res. 110 (C11021) doi:101029/2005JC002902.

Hermann AJ, Hinckley S, Megrey BA, Stabeno PJ (1996) Interannual variability of the early life history of walleye pollock near Shelikof Strait, as inferred from a spatially explicit, individual-based model. Fish. Oceanogr. 5 (Suppl. 1): 39–57.

Hermann AJ, Hinckley S, Megrey BA, Napp JM (2001) Applied and theoretical considerations for constructing spatially explicit Individual-Based Models of marine larval fish that include multiple trophic levels. ICES J. Mar. Sci. 58: 1030–1041.

Hermann AJ, Stabeno PJ, Haidvogel DB, Musgrave DL (2002) A regional tidal/subtidal circulation model of the southeastern Bering Sea: Development, sensitivity analyses and hindcasting. Deep-Sea Res. II (Topical Studies in Oceanography) 49: 5495–5967.

Hermann AJ, Moore CW, Dobbins EL (2003) Serving 3-D rendered graphics of ocean model output using LAS and VTK. In 19th International Conference on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, 2003 AMS Annual Meeting, Session 5.3, Long Beach, CA, 9–13 February 2003.

Hermann AJ, Moore CW (2004) Commodity passive stereo graphics for collaborative display of ocean model output. In Proceedings of the 20th International Conference on Interactive Information and Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology, 2004 AMS Annual Meeting, Seattle, WA, 12–15 January 2004, paper 8.13.

Hermann AJ, Hinckley S, Dobbins EL, Haidvogel DB, Mordy C (2008) Quantifying cross-shelf and vertical nutrient flux in the Gulf of Alaska with a spatially nested, coupled biophysical model. Deep-Sea Research Part II, in press.

Hibbard W, Santek D (1990) The Vis5D system for easy interactive visualization. In Proc. Visualization '90, IEEE CS Press, Los Alamitos, Calif., pp. 28–35.

Hibbard W (2002) Building 3-D user interface components using a visualization library. Comput. Graph. 36(1):4–7.

Hinckley S, Hermann AJ, Megrey BA (1996) Development of a spatially explicit, individual-based model of marine fish early life history. Mar. Ecol. Prog. Ser. 139: 47–68.

Hinckley S, Hermann AJ, Meir KL, Megrey BA (2001) The importance of spawning location and timing to successful transport to nursery areas: a simulation modeling study of Gulf of Alaska walleye pollock. ICES J. Mar. Sci. 58:1042–1052.

Hinckley S, Coyle KO, Gibson G, Hermann AJ, Dobbins EL (2008) A biophysical NPZ model with iron for the Gulf of Alaska: Reproducing the differences between an oceanic HNLC ecosystem and a classical northern temperate shelf ecosystem. Deep-Sea Research Part II, in press.

ISO/IEC 14772-1:1997 (1997) Information technology – Computer graphics and image processing – The Virtual Reality Modeling Language – Part 1: Functional specification and UTF-8 encoding

Leigh J, Dawe G, Talandis J, He E, Venkataraman S, Ge J, Sandin D, DeFanti TA (2001) AGAVE: Access Grid Augmented Virtual Environment, Proc. AccessGrid Retreat, Argonne, IL, January 16, 2001.

Moore CW, McClurg DC, Soreide NN, Hermann AJ, Lascara CM, Wheless GM (2000a) Exploring 3-dimensional oceanographic data sets on the web using Virtual Reality Modeling Language. In Proceedings of Oceans '99 MTS/IEEE Conference, Seattle, WA, 13–16 September.

Moore CW, Soreide NN, Hermann A, Lascara C, Wheless G (2000b) VRML techniques and tours: 3D experiences of oceans and atmospheres. In Proceedings of the 16th International Conference on IIPS for Meteorology, Oceanography, and Hydrology, AMS, Long Beach, CA, 9–14 January 2000, 436–438.

Powell TM, Lewis CVW, Curchitser E, Haidvogel D, Hermann A, Dobbins E (2006) Results from a three-dimensional, nested biological-physical model of the California Current System: Comparisons with Statistics from Satellite Imagery. J. Geophys. Res. 111 (C07018), doi:10.1029/2004JC002506.