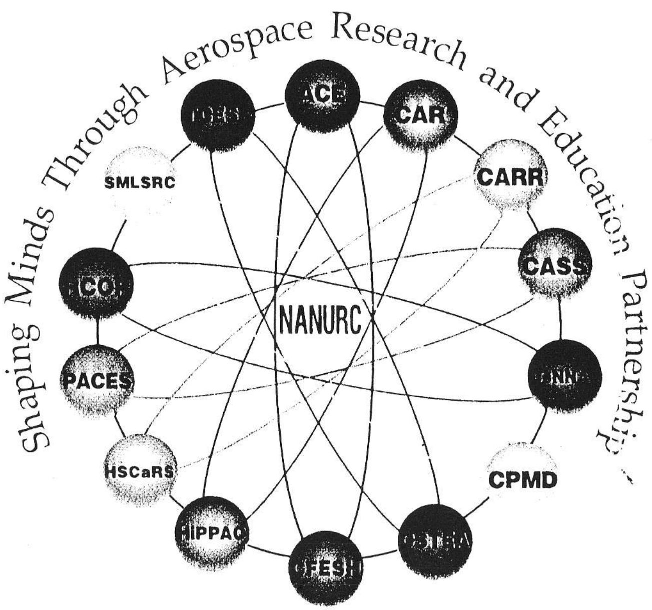
Published 1998
98-HP

Sponsored by
Office of Equal Opportunity Programs
NASA Headquarters
Washington, DC

NASA
UNIVERSITY RESEARCH CENTERS

TECHNICAL ADVANCES IN AERONAUTICS, SPACE SCIENCES AND TECHNOLOGY, EARTH SYSTEMS SCIENCES, GLOBAL HYDROLOGY, AND EDUCATION

VOLUME III



National Alliance of NASA University Research Centers

EDITORS
T. L. Coleman, B. White, and S. Goodman

ASSOCIATE EDITORS
P. Sakimoto, L. Randolph and D. Rickman

Alabama A&M University
Center for Hydrology, Soil Climatology
and Remote Sensing

HIGH PERFORMANCE DATABASE MANAGEMENT FOR EARTH SCIENCES*

Naphtali Rishe, David Barton, Frank Urban, Maxim Chekmasov, Maria Martinez, Elma Alvarez, Martha Gutierrez, Philippe Pardo

High Performance Database Research Center, School of Computer Science
Florida International University, ECS 243, University Park, Miami, FL 33199 U.S.A.
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

The High Performance Database Research Center at Florida International University is completing the development of a highly parallel database system based on the semantic/object-oriented approach. This system provides exceptional usability and flexibility. It allows shorter application design and programming cycles and gives the user control via an intuitive information structure. It empowers the end-user to pose complex *ad hoc* decision support queries. Superior efficiency is provided through a high level of optimization, which is transparent to the user. Manifold reduction in storage size is allowed for many applications. This system allows for operability via Internet browsers. The system will be used for the NASA Applications Center program to store remote sensing data, as well as for Earth Science applications.

INTRODUCTION

The NASA Regional Applications Center (RAC) at Florida International University (FIU) is being established as a storage facility and distribution center for all kinds of spatial and conventional data of local and regional interest. Access to this data is provided to a group of simultaneous local users via 150Mbps ATM communication lines and also to remote users via the Internet at much lower speeds. The volume of data is such that it can be represented in only one form in the database and for interactive access, must be compressed. Up to now access to such a large amount of satellite data has been by cumbersome FTP file transfer, sometimes taking a week or more to arrange for file availability. It is widely believed that this difficult access is hindering potentially valuable and widespread applications of remotely sensed satellite data. To make the compressed spatial data easily available to users we have implemented several database user views that enable user programs to interact with the data as if it were stored in many different formats. This technique greatly simplifies access to the spatial data and enables the rapid development of user programs.

The Everglades National Park is currently considered one of the most environmentally sensitive regions in the United States. The problem of monitoring this territory is increasingly important. Environmental surveys have been performed in the Park for more than half a century, and the results of these surveys need to be stored and managed in an efficient manner that is also easy to use. To achieve this goal a joint project between Everglades National Park (ENP) and the High Performance Database Research Center (HPDRC) at FIU has been established. The project is aimed at constructing a set of databases which contain the environmental data that has been collected in the park since 1911 to date. These data sets describe trends in vegetation, land and marine fauna as well as the physical parameters of the area. The design stage of the project is based on the semantic binary model database approach that is being developed at the HPDRC. The implementation stage consists of table creation; data loading, forms, user views and reports, as well as Internet access support using Oracle Relational Database Management System.

Along with the improved use and management of this environmental data, the ongoing research contributes towards better understanding and documentation of historical data. It will aid in future survey designs by allowing the opportunity to check the data sets and correct errors in them. Through the use of modern database technology, survey results can be made available to the public through the Internet. Thus, new environmental surveys will have a stronger research background and the data will be widely available for urban planing and educational purposes. Plans are currently underway for the construction of a semantic database, which will integrate the present textual data with satellite images of the territory. This will give a broader view of the development of South Florida.

---

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, and NRA-97-MTPE-05), NSF (CDA-9711582, IRI-9409661, and HRD-9707076), ARO (DAAH04-96-1-0049 and DAAH04-96-1-0278), DoI (CA-5280-4-9044), NATO (HTECH.LG 931449), and State of Florida .

SEMANTIC DATABASE MANAGEMENT SYSTEM

HPDRC is presently developing, under NASA sponsorship, a high performance semantic database system. Among its present applications is storage of spatial data collected from the OrbView-2 (formerly SeaStar) Satellite and other remote sensors. This system is used by the RAC. HPDRC's Semantic Database Management System (DBMS) is based on the Semantic Binary Model. In the Semantic Binary Model, information is represented by logical associations (relations) between pairs of objects and by the classification of objects into categories. The Semantic Binary Model is the most natural and convenient way of specifying the logical structure of information and for defining the concepts of an application's world. It is represented in the form of a semantic binary schema (Rishe 1992).

The Semantic Database models are potentially more efficient than the conventional models for two main reasons. The first is that all the physical aspects of the representation of information by data are invisible to the user and the second is that the system knows more about the meaning of the user's data and about the meaningful connections between such data. The first reason creates a potential for optimization by allowing more changes without affecting the user programs. The second allows this knowledge to be utilized to organize the data so that meaningful operations can be performed faster at the expense of less meaningful operations (Rishe 1992). Decomposing queries into atomic retrieval operations maximizes the efficiency of retrieval requests and each atomic retrieval request normally requires only one disk access.

Efficient updates are also a requirement of the semantic database. A transaction is composed of a set of facts to be deleted from the database, a set of facts to be inserted and additional information needed to verify that there is no interference between transactions of concurrent programs. A program fragment containing numerous update commands contained among other computations can generate a transaction. However, until the last command within a transaction is completed, the updates are not physically performed but instead are accumulated by the DBMS. Once the transaction is completed, the DBMS checks its integrity and performs the update. This insures the consistency of the database, with regard to applications and users. Until the transaction is completed, its effects are invisible (Rishe 1992). HPDRC's Semantic DBMS contains semantic facts and inverted semantic facts. This fact inversion scheme assures efficiency of queries including range queries and content access and also exhibits low entropy of data blocks, which facilitates compression. Even without special compression algorithms, the semantic DBMS is very storage efficient. However, contemporary compression techniques can help achieve significant savings (Rishe 1993).

The mathematical abstraction of the relational model has allowed the introduction of powerful and easy-to-use languages for retrieval and updates of databases. The semantic model however, offers a higher degree of abstraction, which results in more concise user programs, speedier processing (due to optimization), and a wealth of other features. Relational databases are good for general conventional database applications. However, in situations where the structure of information is complex, or where greater flexibility is required (objects with unknown identifiers, or objects moving from one category to another, etc.), or where non-conventional data is involved (spatial data, long text, images, etc.), semantic databases are successful.

We are currently designing a parallel database system based on the semantic/object-oriented approach that will accommodate many servers, databases, and users. A server (S1) will gain access to other databases hosted by another server (S2) if there is a link between the two servers. S1 will keep information about the accessed database through its cache and update the database on S2 upon commitment of transactions. The databases will be stored as binary files. A server will access blocks of a file through the file system interface. A file server may use many disks for storage and thus will be able to retrieve different blocks for different databases concurrently.

Our database system provides exceptional usability and flexibility. This strength is inherent to the object-oriented approach of our binary semantic model. A database developer will find designing in the semantic model intuitive since it mostly reflects the real world. The model captures the data along with its meanings (through relations). The model allows the production of the same database design for clients and developers. A client will have a greater understanding of a schema and will be in the position to offer more valuable input. This understanding comes at the price of more iteration in development. Once clients have a better understanding of their design, they usually find it easier to make more demands. A client reviewing the design can easily spot flaws in the understanding of the problem requirements.

A database for Earth Sciences data must address the issue of storage size reduction since the amount of data tends to be large. The relational model carries a lot of overhead in the space allocations for its tables and indices (keys). The semantic database reduces space allocation by eliminating this overhead. Although we also store inverse facts, the absence of additional tables and index files allows us to offer a better space management than relational model does. Usually database systems use various data representation structures where the semantic database system only uses a B-tree as its data structure that facilitates data packing. Since our data are in lexical

order, prefix packing is used in a block of data regarding an object; we only record the object id for the first entry. Access to the data is faster in semantic database since all facts about an object will be stored in contiguous disk blocks. One or a very few disk accesses are needed to satisfy some kinds of queries while in relational world, access to many tables may be necessary thus causing many disk accesses per query.

A multi-user semantic database engine has been developed and is now in the testing phase. A user interface to this engine has been developed using C++ and is also being tested. A Java interface is nearing completion. We are also completing the development of a suite of tools for the semantic database engine: (1) text processing tools to manipulate our semantic data description language and to create databases and documentation and relational images of existing databases, now in alpha releases; (2) graphic tools to allow the creation and modification of semantic databases, which are now in the testing phase; (3) graphic tools for providing reporting functions on the database are partially functioning; (4) graphic tools to define and access a relational image of a semantic database are in the final testing phase.

One of Java based applications is a Semantic Editor. This is a tool, which works as an interface between the user and the semantic database, allowing the user to modify exiting semantic database schemas or create new ones. This is possible by creating new categories, and relations, modifying the attributes, and all the related information, and by being able to delete existing relations and categories. This is a portable and flexible application that can be run locally or remotely through the Internet. Figure 1 shows a picture of the system interface, after a semantic database schema for a university database was opened for modifications.

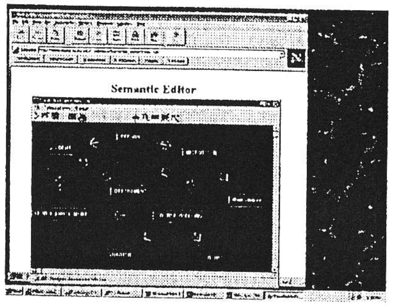


Figure1: Semantic Editor Interface running on the WWW

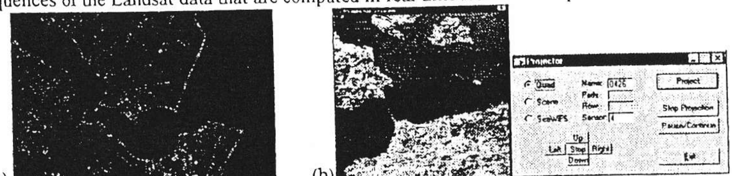
DATA ACCESSABILITY

At HPDRC, we have acquired from NASA Goddard Space Flight Center about 5G bytes of Landsat Thematic Mapper (TM) data observed by the Landsat 4 and 5 satellites. We have available 27 Quads of areas of the continental United States and 9 Scenes of areas of Brazil. For the storage of this TM data we used a Semantic Binary Database for the reasons mentioned in the above section. We have compressed the TM data prior to storage and so reduced the storage demands to about 2G bytes. We present here two Web based applications that demonstrate random rapid access to small parts of this compressed database via two distinct database views. The first view renders the compressed database as filtered rectangular false color maps of pixels. The second view renders the same database as a series of single sensor maps from which a movie can be generated

An HTML client program acts as the front-end application for the Landsat page. The user's database query is obtained by first showing a map of the region for which the database contains data. The user then selects an area (say Florida from a map of the U.S.). The client machine then displays an image from which the user selects a Landsat scene or quad. Finally from within the quad/scene the user selects the area to be displayed. Only a small portion of the total data set can be viewed in complete detail at one time owing to bandwidth and screen size limitations. Having selected the geographic area of interest, the user next selects the observation date from the dates presently available in the database for which this scene or quad was observed. The three sensors to apply as RGB colors for the false color composite image are next selected and, if required, an image-processing filter is chosen to modulate the desired image. The client program transmits the query data to a CGI script on the server, which makes a query to a view of the semantic database that yields a filtered false color image. This image is converted to JPG format to reduce the bandwidth demands on the network and returned to the client for display, this is shown in Figure 2.

In order to improve access to our Landsat database we have developed a program simulating a movie projector that allows the user to "fly" over the satellite images using the mouse or the keyboard to control the "flight path". The previous application provides access to satellite images, but because of the large size of the quad, the user is only able to see a small part of that image. The projector we have developed provides access to the entire satellite image, which is approximately 12M bytes, by gliding over the quad's surface in a continuous motion. The movie presently accesses only one sensor at a time for any given quad.

The movie system consists of a client/server combination and in this case the client side program must be downloaded from our site and installed as a browser helper. This helper program is written in the C++ programming language. It establishes a socket connection between the client and server through which the data is transmitted. The user then chooses any Landsat quad in the database and about 2% of the central area of the quad is displayed in complete detail in a fixed size window. Then, using the mouse, the user can cause the window to move (or fly) over the remaining 98% of the quad in any manner desired and at various speeds. Of course, it is possible to stop at any time and to "fly" backwards. Figure 2b below, shows the application along with the control panel. As the user navigates over the image the projector (client program) retrieves frames from the server. This server accesses the same compressed Landsat database that is used by the other applications described elsewhere in this paper. The server program makes database queries to a view of the Landsat database that generates movie sequences of the Landsat data that are computed in real time from the compressed database.



(a) (b)

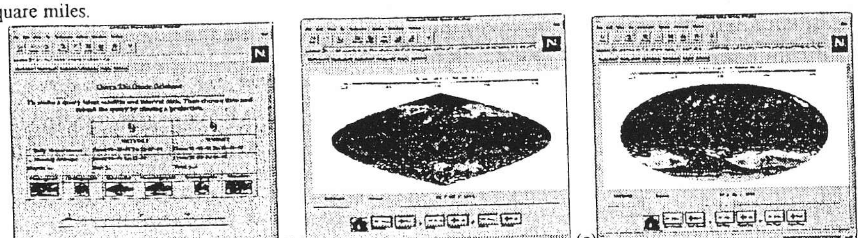
Figure 2(a)Landsat image (RGB=4,3,2) with Sobel edge detector (Jensen 1996).(b) Projector application with control panel.

A technique has been developed to reduce the bandwidth demands and requires less data transfer per frame. Transferring the border regions of the images that come into the new view instead of refreshing the entire image each time the frame changes does this. This technique reduced the bandwidth demands from a quadratic function of frame size to a linear function and will enable us to use a larger frame size or alternatively a faster refresh rate. In due course, Internet 2 should mitigate all the problems that arise from limited bandwidth.

The Ozone database at HPDRC contains all the data from the Total Ozone Mapping Spectrometer (TOMS) deployed on the Meteor-3 and Nimbus-7 satellites, and spanning a period of 15 years (1978-1994). The data is recorded for two different frequencies, monthly and daily. A total of about 0.5G bytes of Ozone Layer Thickness data are available from our on-line semantic database (Rishe 1992). We present here two Web based applications that demonstrate random rapid access to small parts of the database via a simple map view.

In order to manipulate, analyze, and display the information contained in this data set through different client platforms, an Ozone HTML/CGI based web page was developed. Through this page users can obtain detailed information about Ozone using queries to the spatial database and can retrieve textual information, as well as the Ozone maps from different dates, satellites, and frequencies. The query page is presented in Figure 4a. Once the user submits the projection parameters, the client passes the request to a CGI script that queries a database view of the Ozone database. The query returns a rectangular map of Ozone Layer thickness over the Earth's surface. This map is transformed on the fly to the requested projection and encoded as a GIF image for transfer to the client.

Through this page, users can view how the thickness of the Ozone layers change from season to season. Figure 4b shows the Ozone layers at the beginning of the year and as can be seen all values are greater than 230 Dobson Units (DU) (Asker 1993) (icair.iac.org.nz)(rtfm.mit.edu). Figure 4c depicts DU values less than 100 DU over the South Pole (Antarctic). These values were recorded by NASA's TOMS instrument flying on the Russia's Meteor-3 Satellite on October 6, 1993. On this day the ozone hole, the region where the total ozone column is less than 200 DU, covered 9 millions square miles, slightly smaller than the 1992 record ozone hole of 9.4 millions square miles.



(a) (b) (c)

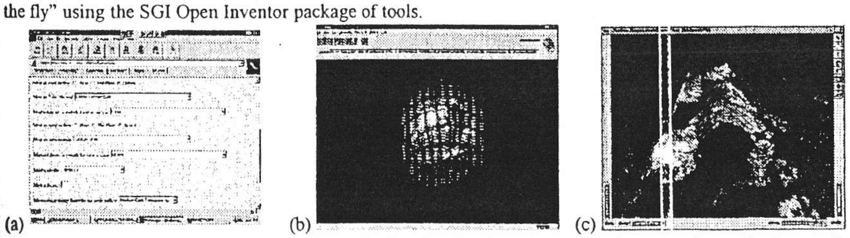
Figure 4.(a) The Query Page of the Ozone Web Page. The user enters a Satellite (Meteor-3 or Nimbus-7), Frequency (daily or monthly), Date, and then clicks on the projection desired to query the database (b)The result of a query from the Meteor-3 daily reading for January 1993 in the Sinusoidal projection.(c)The result of a query from the Meteor-3 daily reading for October 6, 1993 in the holographic projection. The light pink area over the South Pole represents the Ozone hole.

Virtual Reality Modeling Language (VRML) is a standard language describing interactive 3-D objects and worlds for delivery across the World Wide Web. The language is specified in a plain text file known as a VRML World. VRML is based on the SGI Open Inventor Format. VRML defines 3D graphics in terms of geometry, transformations, attributes, lighting, shading, and textures. Further VRML adds some language extensions to the Open Inventor Format to allow linking into the World Wide Web, and adds features and options to allow the user to choose the level of detail that the computer can handle efficiently (vag.vrml.org) (hiwaay.net).

At HPDRC, VRML is used to produce a 3D viewing environment for scientific research applications using the Total Ozone Mapping Spectrometer (TOMS) for ozone layer thickness and ocean temperature data stored in a semantic database. These worldviews are created dynamically, exported through the World Wide Web, and then displayed locally on the client computer with a VRML viewer. The application provides a Web based form to allow input from the user. The application creates a worldview consisting of a set of two surfaces, the inner and the outer surface. The user chooses what spatial data to bind into each surface, that is to say what spatial data should be used to determine the colors rendered on that surface. The user also has the option to perturb each surface from the simple plane or sphere to an irregular "mountainous" terrain. Thus, in principal, the options would allow the simultaneous display of four spatial data sets: two to determine the color of the two surfaces and two to determine the "mountainous" perturbation of those surfaces.

Figure 5b shows a static picture of the VRML World View of two data sets, Ozone Layer Thickness and Ocean Temperature. The World contains two concentric spheres. The inner sphere is unperturbed and colored to represent Ocean Temperature with continents visible. The outer sphere is "colored" only by the Latitude and Longitude lines and otherwise is transparent gray. This outer sphere in greatly perturbed so that its radius represents the thickness of the Ozone Layer at the associated point on the surface of the Earth. The overall effect is that of the Earth with colored oceans representing temperature enclosed in a transparent gray plastic bag of Ozone on which are drawn latitude and longitude great circles. The huge dent in the bag over the South Pole clearly exhibits the Ozone hole.

The VRML viewer (browser plug-in) on the client enables the World View, shown in Figure 5a, to be studied locally without further interaction with the server program that generated the view. The VRML viewer allows the user to rotate, zoom, spin in continuous motion, walk around, and indeed walk into the worldview, all without remote computation. However, the worldview itself represents a great deal of data. The worldview presented in Figure 5b, and available on our Web Site, is deliberately a low-resolution image (hence the blurred great circles) to ensure reasonable download time. However similar worldviews at higher resolution are possible on local clients and can be made more widely available over Internet 2. On local clients we have developed a similar application that uses the Inventor 3D Data format and presents the same 3D images that can then be viewed with 3D glasses and a suitable graphics workstation, this is shown on Figure 5c. The 3D-dataviewer system is implemented on the client using HTML and a VRML plug-in as shown on Figure 5a. On the server, a CGI script poses queries to a view of the database that presents spatial maps and then generates the VRML World View "on the fly" using the SGI Open Inventor package of tools.



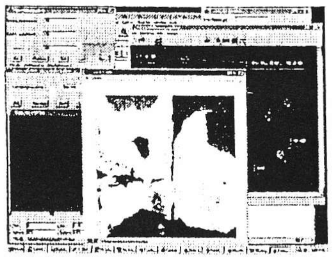
(a) (b) (c)

Figure 5. (a) 3D Dataviewer HTML Form (b) VRML World: Ozone Perturbing the Outer Surface with Ocean Temperature in the Inner Surface(c) Local Version of 3D Dataviewer.

Another data set at HPDRC is simulated and real-time Sea-viewing Wide Field-of-view Sensor (SeaWiFS) data. This data is derived from the broad area color imaging satellite, OrbView-2 satellite (formerly SeaStar) and SeaWiFS sensor. The SeaWiFS instrument is 1.1 Km multispectral sensor that acquires six visible and two near infrared channels for broad area applications such as fishing operations, harmful algae bloom monitoring, sediments monitoring, offshore oil and gas operations, agriculture and forestry management (Beck 1997). In order to efficiently store and manipulate this huge amount of data, we have created a SeaWiFS semantic database (Rishe, 1992) containing about 100 MB of simulated SeaWiFS data. We are in the process of building another spatial database to store the real-time OrbView-2 data. To view this data set, we presently have two Java based applications that portray random access to a small part of the data via a simple viewer interface.

In order to manipulate, analyze, and display the data contained in this data set through different platforms a SeaWiFS Java based web page was developed. Through this page, a user can retrieve textual and technical information about the SeaWiFS data including date/time, scan line, gain-tdi mirror mode, sensors, points-colors and many others, as well as the satellite coordinated SeaWiFS map. A user can select at random which area of the whole (1984x3560-pixel) image, they want to view and retrieve only this area from the database. Another feature about this system is that the user can request from the database a particular value of a point of the whole image, then the value (digital number) and the actual color are displayed on the window. Figure 6a shows a screen capture of the application running.

The other Java based application displays simulated SeaWiFS (SeaStar) data, Ocean Temperature data (supplied by the University of Miami Rosenstiel School of Marine and Atmospheric Science) and Ozone (TOMS) data. The main difference from this application and the previously describe is that it concentrates more in the image processing part and less in the technical information about the data. Using this cross-platform application a user can view the simulated SeaWiFS data geolocated and in different projections including Orthogonal, Orthographic, Sinusoidal, Holographic, Stenographic, and Macerator. Figure 6b portrays a screen capture of the system interface after retrieving some information.


(a)

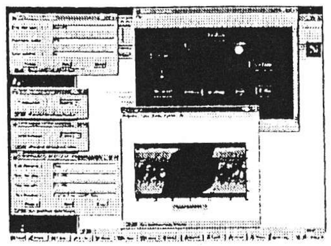

(b)

Figure 6: (a)Interactive System displaying some technical information and the map for the simulated SeaWiFS (SeaStar) data (b)Multi-dataset interactive system, after some information about the simulated SeaWiFS data that was retrieved from the spatial database.

SUMMARY

At the High Performance Database Research Center we are finalizing the development of a semantic database system to store remote sensed data as well as other Earth Science databases. Applications that use Internet technologies are used to overcome the access and display problems, which have limited the scientific community. These provide swift and efficient retrieval of huge amounts of data from the semantic database.

REFERENCES

Asker, J. R. "US/Russian satellite maps worst Antarctic ozone hole", Aviation Week & Space Technology, Vol. 139, No. 17, Oct 1993, p.72.

Rishe, N., Database Design: The Semantic Modeling Approach, McGraw-Hill, 1992.

Beck, J. M., Albasini, J., Hill, C., Techniques for Marketing SeaWiFS Ocean Color Satellite Imagery on the Internet, 1997.

Rishe, N., Accepted grant proposal to NASA, Grant NAGW-4080, 1993.

Jensen, J.R., Introductory Digital Image Processing-A Remote Sensing Perspective, 2nd ed., Prentice Hall, 1996.

http://vag.vrml.org/VRML_FAQ.html

http://hiwaay.net/~crispen/vrml/faq.html

http://icair.iac.org.nz/ozone/ozone.html

ftp://rtfm.mit.edu/pup/usenet/news.answers/ozone-depletion/intro