

Suitability of spectral remote sensing for coral reef surveying, monitoring, and mapping

Richard Gomez*, Menas Kafatos, Ambrose Lewis
School of Computational Sciences, George Mason University

ABSTRACT

Many researchers consider coral reefs the "rainforests of the oceans" because they cover such a small area and yet provide homes for literally thousands of unique marine species. A multispectral or hyperspectral remote sensing satellite, with its spectral coverage, offers significant advantages over traditional methodologies for coral reef surveying, monitoring, and mapping. This paper presents research into the suitability of spectral remote sensing for coral reef surveying, monitoring and mapping using the SeaWiFS multispectral ocean color data for illustration. We describe the information technology developed to support this research and provide an overview of the database driven web application, which was developed to allow live interaction with the data. A database of in situ observations from the ReefBase web site (www.reefbase.org) was used as validation data as part of this investigation. This discussion includes details on the XML representation of the satellite and in situ data and metadata. It also introduces a dynamic Java Visualization applet developed to allow the users to visually interact with the data. The paper concludes with a discussion of the suitability and additional advantages of using hyperspectral remote sensing technology for this application that exploits the full spectral characteristics of submerged coral reefs.

Keywords: Coral Reef, Spectral, Remote Sensing, SeaWiFS, Multispectral, Hyperspectral

1. INTRODUCTION

Many researchers consider coral reefs the "rainforests of the oceans", since they are cover such a small area, yet provide homes for literally thousands of marine species. Human and climatic induced changes are stressing coral reefs globally. The first step in trying to understand these effects on a global scale would be to develop a global map of all coral reefs. Updating the map over time will provide the means to monitor the health of all reefs. Scientists could use such a map as the basis for future analysis, and may be able to draw conclusions into the root causes of coral reef problems by using statistical means. Spectral remote sensing of coral reefs would be very helpful in this effort. The current global maps rely on local in situ observations, which are expensive and not reliable.^A With a database of in situ observations used for validation purposes, spectral remote sensing data comparisons to models of coral reef spectral features will be a less expensive and more reliable method to survey, monitor, and map the coral reef. This paper describes an interactive web-based, data driven application, which can be used to analyze the coral reef remote sensing data.

2. SCIENTIFIC ASPECTS

This paper provides a detailed discussion on the scientific aspects of the remote sensing approach. First, we discuss coral reefs in general and the specific problems facing them. Next, we present the Lubin's model for the spectral signatures of coral reef features from space. Finally, we provide an in-depth discussion of the spectral remote sensing platform, instrument, and data that used in the analysis.

2.1 Coral reefs background

According to The World Atlas of Coral Reefs (2001), "Coral Reefs in their natural state have been considered a highly stable ecosystem", but that "minor changes to the ecosystem will have significant repercussions to the health

^A Inconsistencies between different observers can effect how the reef is reported to an in situ global map. For instance, it is somewhat difficult for the causal observer to differentiate between similar coral species. In the reefBase database, for instance, anyone may register and submit data. Quality control is a significant issue.

*rgomez@gmu.edu, phone 1 703 993-3629, fax 1 703 993 3628, <http://www.ceosr.gmu.edu>, George Mason University, School of Computational Sciences, MS 5C3, 4400 University Drive, Fairfax, VA, USA 22030-4444

and safety of the reefs.”¹ According to the ReefBase web site there are four broad categories of human-induced impacts on coral reefs:

- **Pollution:** The major pollution impact on coral reefs comes from nutrient enrichment. This is linked to human waste, agricultural run-off, and increased coastal development.
- **Sedimentation:** This has a significant effect on the health of coral reefs. This is the result of sediments hanging in the water above the coral reefs that block light from reaching them.
- **Over-fishing:** Humans can so over-fish a reef, the balance of species is significantly changed. This can result an unhealthy distribution of marine organisms from the reef’s perspective. The author notes that specific types of fishing may cause greater damage to the reefs. These techniques include blast fishing, trawling, and “bleach fishing”.^B
- **Climate Change:** Coral reefs appear to be the most temperature sensitive ecosystem in the world. Reefs are especially susceptible to Coral Bleaching when stressed by higher ocean temperatures. Bleaching is said to occur when the coral dies and the rich color of alive coral is replaced with the white color of the underlying skeleton.^C

2.2 Lubin’s model

Lubin, in “Spectral Signatures of Coral Reefs: Features from Space”, describes a coupled ocean-atmosphere radiative transfer model, which is used to investigate the potential of satellite remote sensing to map the global distribution of coral reefs throughout out world. The model that Lubin proposes examines how the “spectral and radiometric signatures of coral reef objects should appear above from low earth orbit after being modified by the water column and atmospheric attenuation.”³ As an example, he tells us that based on in situ observations, sand has a considerably higher reflectance than any type of living coral. The question to answer is does this significant reflectance difference that can be measured in situ manifest itself into a suitable radiance contrast at the top of the atmosphere.³

Holden and Ledrew⁴ build the basis for spectral characteristics of in situ coral observation in part on a 1999 paper. Their paper demonstrates that:

1. There are “robust” differences between in situ reflectance spectra of living corals and those of related non-coral objects.
2. These differences are independent of geographic sampling and of coral morphology.
3. A scene identification algorithm based on differential reflectance can be constructed with these spectral contrasts.

Lubin⁵ builds upon their work to pose the question, “Does this type of differential reflectance spectroscopy remain applicable to scenes observed from above the atmosphere?” Previous works by Lyzenga⁶ and Maritorena⁷ have shown that differential absorption by the water column will modify considerably the spectral reflectance of an object at depth. Lubin’s work, however, is the first attempt to combine the effects of the water column with an atmospheric model for the purpose of studying coral reefs.

The Lubin’s paper investigates the reflectance spectra of eight different objects: sand, corallien red algae, green macro-algae, algal turf, and four coral species. The non-coral objects are necessary to distinguish coral reefs from their surroundings. The four coral species were selected based on their spectral characteristics and the availability of a study by Mazel of a Bahamian reef patch named Rainbow Gardens. Lubin utilizes a discrete ordinates radiative transfer algorithm to model how these different objects might appear from space after light reflected from them has been further attenuated by the water column and scattered by the atmosphere.³ Stamnes first presented this algorithm in 1998.

Lubin modeled the atmosphere using 35 layers of uniform optical depth to compute τ_a or the total optical depth of the atmosphere. The τ_a is the sum of Rayleigh scattering, aerosol scattering/absorption, an ozone absorption optical

^B For more details on “bleach fishing” as a technique of acquiring specific species for use in aquariums, we refer the reader to an article in the May 2001 Scientific American.

^C NOAA has a web site dedicated to the study of coral bleaching effects of sea surface temperature using remote sensing data. Please refer to “Tropical Ocean Coral Bleaching Indices” available on the web at URL http://www.osdpd.noaa.gov/PSB/EPS/SST/dhw_news.html

depth, and is a function of wavelength.³ The water column is modeled as a single layer having an optical depth of τ_w , which is also a function of wavelength. Lubin's model assumes a flat and static air-water interface, and ignores the effects of ocean roughness. Further, he assumes the bottom surface will be modeled as a Lambertian surface albedo.³ Lubin's scaled radiance for satellite imagery in nadir or near-nadir view is actually smaller than the true albedo due to the angular redistribution of radiation that favors the direction of the solar beam.³

The useful signal for remote sensing of coral reefs lies within the range 400 nm and 600 nm based on Lubin's calculations. This range lines up with the instrument selected for our study. Unfortunately, the effects of the atmospheric Rayleigh scattering at the top-of-atmosphere reflectance spectra between 400 nm and 600 nm are very similar for most of the objects selected by Lubin.³ He presents many graphs of the upwelling spectral radiance as a function of depth in his paper.

2.3 Remote sensing technology utilized

2.31 SeaWiFS platform

The Sea-viewing Wide Field-of-view Sensor (SeaWiFS) instrument is on-board the Orbview-2 platform. Orbview-2, formerly known as "SeaStar" and built by the Hughes Electronics Santa Barbara Research Center, was launched on August 1 1997, and orbits in a sun-synchronous, descending node orbit at an altitude of 705 km. The local time of the orbit's descending node is 12:05 pm.

2.32 SeaWiFS instrument

SeaWiFS is an eight channel visible light radiometer dedicated to global and regional ocean color measurements. SeaWiFS is a follow-on mission to the Coastal Zone Color Scanner (CZCS) instrument. SeaWiFS improves on the CZCS mission by having better bands for atmospheric correction or the removal of the light scattering by the Earth's atmosphere.¹⁰ This improvement makes the SeaWiFS instrument much better at estimating chlorophyll and suspended matter along the coastal regions.

The orbital characteristics of the platform allow coverage of more than 90% of the ocean surface every two days.¹⁰ The data is collected in two spatial resolutions: global data at 4.5 km and regional data at 1 km.¹⁰ The instrument's primary optics consist of an "off-axis folded telescope and a rotating half-angle mirror."¹⁰ This telescope collects the radiation back scattered by the Earth's surface and atmosphere and reflects it onto the instrument's mirror. From here, the beam is directed through a series of dichroic beam splitters, which transmit some wavelengths and reflect the rest. After the beam splitters separate the light into four wavelength regions, spectral bandpass filters are used to narrow these regions into the desired spectral bands. Next, the light signal is converted from analog to digital, amplified, and time-tagged for transmission back to Earth.¹⁰ (For more information on the characteristics of the SeaWiFS data, please refer to reference 10.)

2.33 SeaWiFS data

SeaWiFS Level 1 data consists of at-instrument raw radiance counts. Calibration and navigation data is available in a separate data file.

SeaWiFS Level 2 data consists of normalized water-leaving radiances and geophysical parameters derived from the radiance data. The normalized water-leaving radiance data is corrected for atmospheric light scattering and sun angles differing from nadir. The following list defines the classes of SeaWiFS Level 2 data¹⁰:

- Normalized water-leaving radiance at 412 nm, 443 nm, 490 nm, 510 nm, 550 nm, and 670 nm
- Chlorophyll a concentration (derived)
- K(490) (derived)
- Angstrom coefficient, 510-865 nm (derived)
- Epsilon of aerosol correction at 765 and 865 nm (derived)
- Aerosol optical thickness at 865 nm (derived)

SeaWiFS Level 3 data consists of these geophysical parameters binned to a 9km square global spatial grid. Data are available over several temporal resolutions, including 8-day, monthly, and annual intervals. The SeaWiFS Level 3 data product consists of a main file of meta-data and data set wide information and a set of 13 subordinate files for

geophysical parameters. SeaWiFS Level 3 data can be quite large. For example, a monthly main file can average about 62 megabytes and each subordinate file can average 26 megabytes each!¹⁰

2.34 SeaWiFS ocean color measurement

A quantitative measurement of the global ocean color is crucial to understanding the ocean's roles in the planet's carbon cycle. According to Dr. Charles McClain, the SeaWiFS Project Scientist, "The concentration and predominant identity of substances and particles in the euphotic, or lighted, zone of the upper ocean influences the apparent color of the ocean, which can range from deep blue to varying shades of green and ruddy brown."¹¹ Living phytoplankton, which contain chlorophyll and associated pigments, inorganic sediments, detritus (particulate organic matter), and dissolved organic matter all contribute to the color of the ocean.¹¹

On a proof-of-concept mission, the first attempt to measure ocean color from space was the Coastal Zone Color Scanner (CZCS). CZCS not only demonstrated that ocean color could be remotely sensed from space, but provided scientists with eight years of operational data, from November 1978 till June 1986.¹¹ SeaWiFS is a critical element of NASA's biogeochemistry research program.

Remote sensing of ocean color is based on the principle that particulate and dissolved substances suspended in water will interact with incident light.¹¹ In open ocean conditions, concentrations of particulate matter and dissolved substances are low. In these conditions, the water molecules scatter light similar to the way the atmosphere scatters light, producing a characteristic deep blue color.¹¹ The scattering of light that is caused by particulate reflection and absorption of light by dissolved substances will alter this deep blue ocean color.¹¹ For example, the photosynthetic pigment found in phytoplankton absorbs strongly in the red and blue regions of the visible spectrum and reflects strongly in the green region. Therefore, as the concentration of phytoplankton in the ocean increases, the water will appear increasingly green in color. The absorption of light by phytoplankton's chlorophyll can be quantified to determine the concentration of chlorophyll in the water and allow the estimation of phytoplankton abundance in a given area.¹¹

In addition to the required atmospheric corrections, remote sensing of ocean color can be strongly effected by the presence of light-scattering inorganic particulate matter in the water.¹¹ Since particulate concentrations generally increase closer to the coastline, the water's color tends to change from blue-green to brown or reddish-brown.¹¹ It becomes harder to distinguish the light absorption due solely to chlorophyll in these conditions. By collecting light in several bands, SeaWiFS is able to minimize the effects of suspended inorganic materials on ocean color calculation.¹¹

3. COMPUTATIONAL ASPECTS

The purpose of the computational aspects section is to document the information technology that has been developed and utilized to implement the analysis defined in the project methodology. This section begins with an overview of the high-level system architecture. The second part goes into the data engineering utilized to present the information contained within the remote sensing files and the database of in situ observations. The third part provides the reader with the detailed design of the Coral Reef Analysis Application. This includes a brief discussion on each of the programs developed. Finally, the computational aspects section concludes with a quick look at the Coral Reef Analysis Application from a user's perspective.

3.11 High level system architecture overview

As with most web-based systems, the Coral Reef Analysis Application design is based on a multi-tier system architecture. This approach divides the application into several layers of discreet functions. Example tiers for the application include the user interface/presentation tier, the business logic tier, and the data access tier. Since the programming for a particular tier can be modified or relocated without affecting the other tiers, this 3-tier model makes it easier for an enterprise to continually evolve an application as new needs and opportunities arise. Also, existing applications or critical subsystems can be permanently or temporarily retained and encapsulated as a member of the appropriate architectural tier.¹²

The user interface tier of the Coral Reef Analysis Application will be implemented by a standard web-browser delivering a combination of static and dynamic web pages. Client-side browser based scripting will be utilized to provide more robust features, such as input verification.

The business logic tier provides the Coral Reef Analysis Application specific logic and business rules. Since the critical application logic is encapsulated in this business logic tier, the application does not require any installation on individual client systems, beyond a standard web browser. This architecture also better supports an evolutionary development process, since there is no installed base of client software to support or upgrade.

The data access tier provides interfaces to the required data repositories. It acts as the gateway for data between the users of the system and the back-end files and databases. This is very beneficial in a production system, since the specific logic required to access the data is encapsulated here, change is localized to the data tier when external systems evolve or are upgraded. The main engine behind the Coral Reefs Analysis application is Sun's Java Servlet technology. In general, Servlets represent the Java world's response to Common Gateway Interface (CGI) programming. They reside on a web server and act as a middle-ware layer between front-end web browser requests and back-end application logic. Typically the back end provides access to databases and/or server hosted applications.

3.2 Data engineering

There are three data repositories within the Coral Reef Analysis Application: HDF files, a small database, and XML documents. This sub-section will discuss each of these at length and the next sub-section on Detailed Design, will explain how they are to be used within the project.

3.21 HDF files

HDF stands for Hierarchical Data Format. It was developed at NCSA as a technique for data archiving large volumes of data. A good archiving format should meet the requirements of being easy and efficient to store and maintain over long periods of time and should permit easy access over long time frames. Two major characteristics of the HDF format make it suitable as an archiving format. First, it is self-describing. This means that the file itself contains meta-data about the structure and format of the data itself.¹⁴ Since this information is contained within the file itself, there is little danger that the semantics of the archive could be lost, rendering the data worthless. The second characteristic is the fact that HDF stores the information very compactly.¹⁴ This is critical given the enormous volume of data from applications, such as remote sensing. Further, since HDF uses features like standard time/date stamps and checksum algorithms, it is very successful at storing data in a binary format that is portable to different hardware. For the Earth Observing System, HDF was extended to become HDF-EOS. The extension adds three features to the standard HDF: grids, points, and swaths. The addition of these features allows the data to be accessed via spatial and/or temporal methods.¹⁵

3.22 XML documents

XML stands for "eXtensible Markup Language". Like HDF, it is a self-documenting data description format. XML is a set of rules for defining semantic tags that break a document into parts and identify the different parts of the document.¹⁵ It is a meta-markup language, that allows the user to define a syntax to create other domain-specific, semantic, structured markup languages.¹⁵ This format is being used more frequently for remote sensing applications. For the Coral Reef Analysis Application, several Java applications and Servlets were written to create and parse XML documents. We attempted to create a logical structure for these XML documents.

Each document contains the following information:

First, the root element is the XML feature that all others are members. In this case, the top-level feature is:

```
<SEAWIFS_GRIDDED_DATA>
```

Next, there is an element that defines information used by each cell in the row. This includes information such as the vertical and horizontal cell size, which is common to all cells in a row. Here is a row specific example:

```
<ROW>
  <ID>1000</ID>
  <NORTH_LAT>6.5833</NORTH_LAT>
  <SOUTH_LAT>6.5000</SOUTH_LAT>
  <VSIZE>0.0833</VSIZE>
  <HSIZE>0.0839</HSIZE>
  <CELL_COUNT>4291</CELL_COUNT>
</ROW>
```

Finally, there are elements for each cell in the file. The field “cell_count” in the row element tells us how many cells of data are contained in this XML document. The cells are number from zero to (cell_count – 1). Each element contains information specific to that cell, such as the bands of remote sensing data. Here is a cell example:

```
<CELL_0>
  <WEST_LON>-180</WEST_LON>
  <EAST_LON>-179.9161</EAST_LON>
  <BAND_0>58.6196</BAND_0>
  ...
  <BAND_5>0.0288</BAND_5>
</CELL_0>
```

3.23 SQL database

In the most general sense, a database is a collection of information related to a particular subject or purpose, such as tracking customer orders or maintaining a music collection. Typically the term denotes the software application that is used to hold the information. Information is stored within database tables. Tables are made up of columns, which represent a feature for each entry in the table, and rows, which represent individual records. For instance, in the reefBase database, the table contains columns like the reef’s name and location. Each reef should have data in each of these columns. The rows, however, are used to hold information about different coral reefs. Therefore, the information associated with row 1 is the record for Iki Island, while the information associated with row 3 is related to the Suvadiva Atoll. If databases just held information they’d be rather boring. Every database allows the user to pull information from them in a variety of ways. In addition to proprietary access techniques, most modern databases support some level of Structured Query Language (SQL). SQL queries form a request to the database that defines what information is pulled from which table, under some constraint. For example, the Coral Reef Analysis Application uses the following SQL query:

```
SELECT ID, REEFNAME, COUNTRY, LAT, LONG
FROM REEF
WHERE ((LONG >= inWest) AND (LONG <= inEast) AND (LAT >= inSouth) AND (LAT <= inNorth));
```

This query asks the database to pull out the columns id, reef name, country, lat, and long from the table reef. The constraint is specified in the WHERE clause. This limits the results to within a user specified bounding box.

3.3 Detailed design

This section describes in detail the purpose and design of each software component developed for the Coral Reef Analysis Application. **Figure 1** shows a block diagram that illustrates each of these software components. The shape of the objects in the figure is used to convey the type of the object. Rectangles with the lower right corner turned up are used to represent data files. The system’s database is shown as a 3-d cylinder and the commercial off the shelf (COTS) system components are shown as 3-d rectangles. Color is also used in the figure to classify the various pieces of the system.

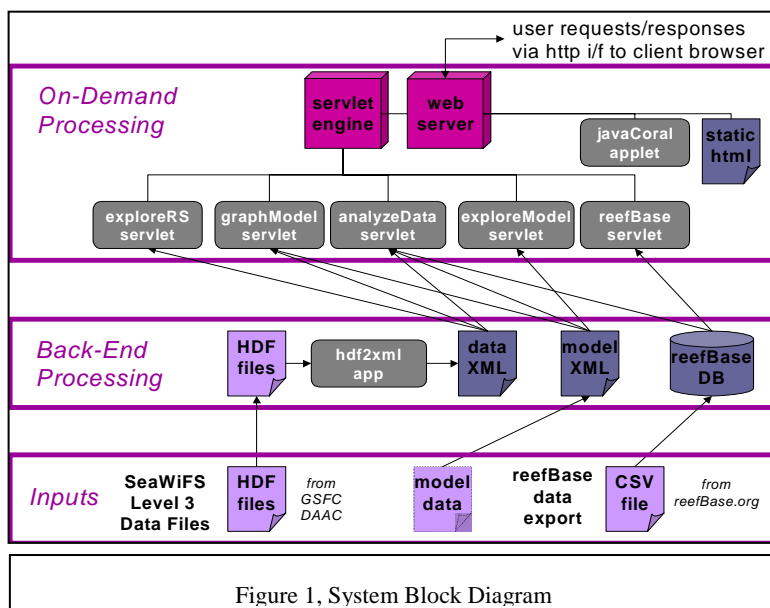


Figure 1, System Block Diagram

Unmodified input data files are given a pale purple color. Created data products are shown as slate blue and COTS software applications are shown as magenta. The applications that were developed for this project are shown as dark gray rectangles. There are three major stages in the block diagram.

The first stage is used to identify the three project inputs: the SeaWiFS HDF files, the export from reefBase, and the model data. Since the model data only exists in the paper and not as a standalone data product, we made it semi-transparent in the figure. The second stage, back-end processing, is used to prepare the data products for use in the Coral Reef Analysis Application.

During this stage, the HDF data is translated into an XML format via a developed program, the model data is implemented as a stand-alone data file, and the reefBase data is imported into a local database. In an operational application, these first two stages would be run infrequently.

The third stage is on-demand processing. As the name suggests, the purpose of this stage is to respond to user requests. This stage is comprised of the COTS networking pieces, all static HTML files for the user interface, and the Java Servlet program used to analyze the system's data.

3.31 Detailed Software Design

First, let's look at the back end processing stage. The program `hdf2xml.java` is to translate the SeaWiFS Level 3 HDF data files into appropriate XML documents. This program is a stand-alone program that receives two input parameters from the user: the starting and ending row. Based on the values of the start and end row parameters, `hdf2xml.java` pulls out the appropriate row information from the `binIndex Vgroup`. For each row, the `binIndex Vgroup` defines the row number, starting location, the vertical/horizontal distance, and the height.

As the program loops over all the rows from the `binIndex Vgroup`, it checks to see if the current row is between the starting and ending row. If so, the data from the `binIndex Vgroup` and from the supporting files are parsed and output to the correct XML file. If not, the program uses the `binIndex Vgroup` data to skip the proper number of non-required data in each of the HDF supporting data files. The output is an XML file for each row of data. As it is parsing the data, the program also computes the latitude and longitude for each row and cell. A text file of summary information is also produced. This text file is parsed by other programs to locate required XML files.

Next, let's look at the on-demand processing stage. The `exploreRS` Java Servlet runs when a user selects the appropriate hyperlink. This program reads the text file created by `hdf2xml.java` program and creates as output a dynamic HTML page containing a table of the row number, a count of data elements in that row, and the north and south latitude for each row. Also, a direct hyperlink to the row's XML data file is provided.

The `exploreModel` Java Servlet runs when a user selects the appropriate hyperlink. It reads the Model XML data files and uses HTML to produce a bar graph of the model data. Based on the data values read for each band of data, the program scales a small 1 pixel by 1 pixel image. First, the largest data value for the model is located, and this value is scaled as 100% of the allowable 200-pixel graph display area. The other band data values are calculated as the integer representation of their relative percent of the distance to the maximum band data. This technique allows for decimal heights to be represented in the integer screen height by approximating a fixed point numbering system.

The `reefBase` Java Servlet runs when a user selects the appropriate hyperlink. This program requires 4 parameters: North Latitude, South Latitude, East Longitude, and West Longitude. Together, these 4 parameters define a bounding box, which is used to pull information out of the `reefBase` database as a SQL query. As the program executes, it builds a SQL query out of the parameters and retrieves the name and Latitude/Longitude for each coral reef within the bounding box. A constraint is applied to pull only the data that has a longitude less than or equal to the North Latitude, greater than or equal to the Southern Latitude, greater than or equal to the Western Longitude, and less than or equal to the Eastern Longitude all at the same time. The results are returned as a HTML table within the dynamically generated HTML document.

The `analyzeData` Java Servlet is the main analysis piece of the Coral Reef Analysis Application. It is called from a static HTML page where the user defines the analysis parameters. These parameters include the North, South, East, and West values for the bounding box and the model data values in each of the bands. The `analyzeData.java` pulls information from the remote sensing data XML files and the `reefBase` database. The data from these two sources are combined into the same grid by the program. First, based on the user's input, the program queries the `reefBase` database to retrieve all coral reefs within the bounding box. Second, `analyzeData.java` reads the text file of row information that was created by the `hdf2xml.java` program. It loops over the rows of data from this file and ignores rows that are not within the North and South limits of the bounding box. To do this, the program determines if the bounding box's north value is less than or equal to the northern latitude of the row and at the same time the bounding box's south value is greater than or equal to the row's southern latitude. If this is the case, a boolean variable is set and all rows are processed until the last one is found. This is determined by testing to see if the bounding box's south value is less than or equal to the northern latitude of the row and at the same time the bounding box's south value is greater than or equal to the row's southern value. Next, the XML data file associated with a selected row is opened and parsed. Since each cell has an associated east and west value, it is a simple test.

A test is done to determine if the bounding box's west value is greater than or equal to the current cell's west value and at the same time the bounding box's west value is less than or equal to the current cell's east value. When this is the case, all cells are processed until the exit condition is reached. Cells are no longer processed after the bounding box's east value is greater than or equal to the current cell's west value and at the same time the bounding box's east value is less than or equal to the current cell's east value. For each cell selected for processing by this algorithm, analyzeData.java checks to see if the reefBase database has any coral reefs present there. If so, this information is retrieved and processed for output. Also for each cell, the remote sensing data in each band is retrieved and parsed from the XML file. The RMS statistical measure is computed as defined in

$$\sqrt{\sum_{i=0}^5 (data_i - model_i)^2}$$

The RMS statistical measure is equal to the square root of the sum of the squared differences between each data and model point. Since each point in the sequence is compared, a small value of RMS indicates that the two sequences are closer together. The computed RMS value is compared to a user-supplied value of epsilon. The relationship between RMS and epsilon is combined with the presence or absence of a coral reef from the reefBase to determine if the model hit or missed a reef. There are several possible outcomes. First, we have a reef present according to the in situ observations and the RMS value is less than or equal to the user supplied epsilon. In this case the model accurately predicted that a coral reef was present. Therefore, we define this as a hit. In the second case, there isn't a reef present according to reefBase, but the RMS value is less than or equal to the user supplied epsilon. This is defined as a false positive. It may be an indication of a failure of the model to predict the existence of a coral reef or it could represent a reef that exists but isn't in the reefBase database. In the third case, there is a reef present according to the reefBase database, but the calculated RMS is greater than the user supplied epsilon. Therefore, we define this as a miss. Finally, if there isn't a reef present and the RMS is greater than epsilon, we've got a situation defined as a so-called soft hit. In this case, the model is saying that nothing is present and the database is saying nothing is present. The output from analyzeData.java is sent back to the user's web browser as an HTML table within a dynamically generated HTML page. Each table row represents a unique combination of remote sensing data row and cell. The columns of the HTML table contain the cell's north, south, east, and west values. It also contains a list of reefs, in any, pulled from the reefBase database that are within this cell's boundaries, the calculated value of RMS, and two dynamically generated hyperlinks to the helper Java Servlets graphData and vizData. The graphData.java functions identically to modelData.java, except that for each band both the model and actual remote sensing data for the selected cell are displayed side by side. Please note that it is necessary to scale both the model and remote sensing data identically so that the graphs for both data sequences line up over all bands. The vizData Java Servlet dynamically builds the HTML page of parameters for the Coral Reef Visualization applet discussed in detail below. The vizData Java Servlet creates an HTML file on the server for the user. Once it is finished processing, it gives the user a direct link to this newly created file. This is done so that we can ensure that all the parameters have been written to the HTML page before the applet is downloaded and begin execution on the user's system. If this technique wasn't used, there would be a race condition and the applet may fail because all the data it requires wasn't written to the enclosing page before it began executing.

3.36 Coral reef visualization applet

The Coral Reef Visualization Applet allows the user to visually interact with the remote sensing and model data. It is launched from a link off of the dynamic page generated by the analyzeData Java Servlet. This link invokes vizData. Java Servlet, which dynamically builds the HTML page that wraps the Coral Reef Visualization Applet. This page contains the information necessary for the applet to display the data as HTML "PARAM" tags. For example, let's look at the following line from the page:

```
<PARAM name="cell195" value="-2.25 N to -2.3333 S,41.0702 E to 40.9868 W,6.6376, 3.148, 2.391, 1.521,0.703,0.0861,Takawa Milinga$511">
```

This line tells us that cell 1195 is bounded by -2.25 North to -2.33 South and 41.07 East to 40.986 West. Following this in the value field are the six bands of remote sensing data. Finally, this example has a coral reef present. The name is "Takawa Milinga" and the reefBase database id is 511. A Java string tokenizer is used to pull out the various pieces of data. String conversion methods are called to translate the data required from strings to floating point numbers. Below is a screen shot of the Coral Reef Visualization Applet in action. In the upper left-hand corner, you'll notice a large area of blue and red squares. Each square represents one cell of remote sensing data.

The color indicates if a coral reef is present in the reefBase database within that cell. There are 11 rows by 21 columns. This gives us five rows above and below and ten rows to the right and left of the center cell. The center cell is the one the user clicked on from the analyzeData.java generated page. The user may click on any cell to move around in main mapping area. The currently selected cell has a white x in it. On the lower left there is a zoom area, showing the eight neighbors of the currently cell.

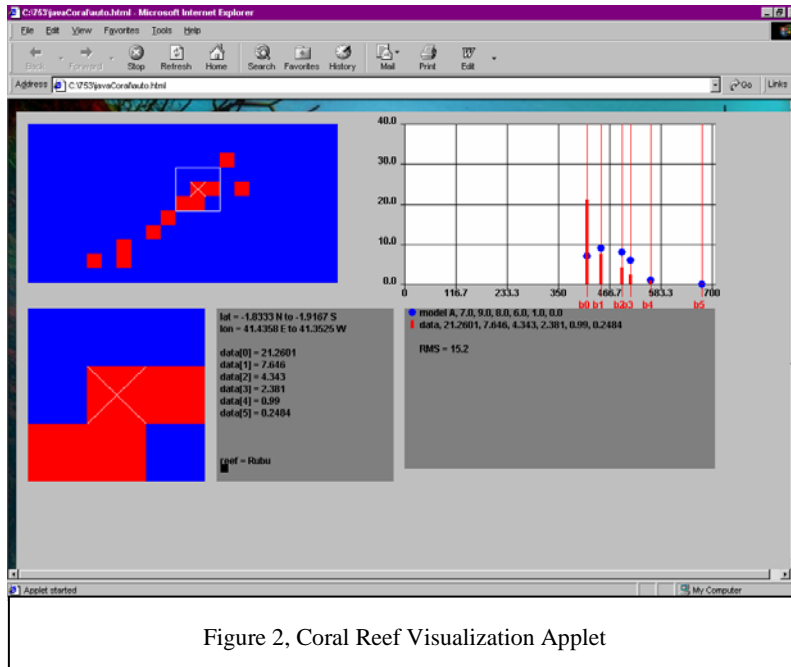


Figure 2, Coral Reef Visualization Applet

of the cell's array of remote sensing data, and a list of coral reefs from the reefBase database (if any).

The black box next to the reef name in the screen shot is actually a hyperlink to the detail.java Java Servlet, which displays the full information about the reef from the reefBase database in a separate browser window. On the right-hand side of the display is a graph and table designed to show the user how closely the remote sensing data from the currently selected cell compares with the model. The bar graph plots the model data as small circles on each of the red band lines. The remotely sensed data is drawn on top of the red band line as a bar. Below the bar graph is a data table that shows the model and data points in a tabular format. The RMS statistical metric is also displayed here. There are many ways in which the Coral Reef Visualization Applet could be enhanced. For example, the cells in the main and zoom area could be colored based not only on reef presence, but also on the RMS comparison between that cell's data and the current model. Also, the tool could be extended to display multiple models at the same time.

4. ANALYSIS AND RESULTS

4.1 Distribution analysis of SeaWiFS remote sensing data

One interesting area would be to investigate the SeaWiFS band data for cells that have at least one coral reef present in the reefBase database. The ultimate goal would be to establish a pattern that is unique to these cells only. In order to do this, we modified a version of the hdf2xml.java tool to output a comma separated values file for cells that have coral reefs present according to the reefBase database. This new program, coralData.java, generates a file containing the row/column number, the bounding box values, and the data values for bands zero to six.

This data was read into Matlab for analysis. We ran against rows 500 till 1500, giving coverage from approximately 33 degrees North to -33 degrees South. This resulted in 5392 unique cells of data in the coralData.java output file. Please note that this number is much lower than full contents of the reefBase database, because there are a few coral reefs outside this coverage and because many cells contain more than one coral reef. Figure 3 shows a histogram

It is very important to note that this isn't just a picture. Each of these cells in the main and zoom areas is a full-fledged Java object with both associated data and methods. For example, each object has its own array of remote sensing band data, as well as the appropriate methods to access this array. By using get and set methods for the object's data, the actual implementation can vary without affecting the calling code. The technique enables the application to be very extensible. It would be a simple matter to extend the data associated with the cells for a more robust version of this problem or for another remote sensing analysis job.

Just to the right of the zoom area is a text box that displays the associated data for the currently selected cell. The information displayed includes the cell's latitude/longitude, the contents

distribution for each of the SeaWiFS data bands. Table 1 provides some cumulative statistics for the same set of data.

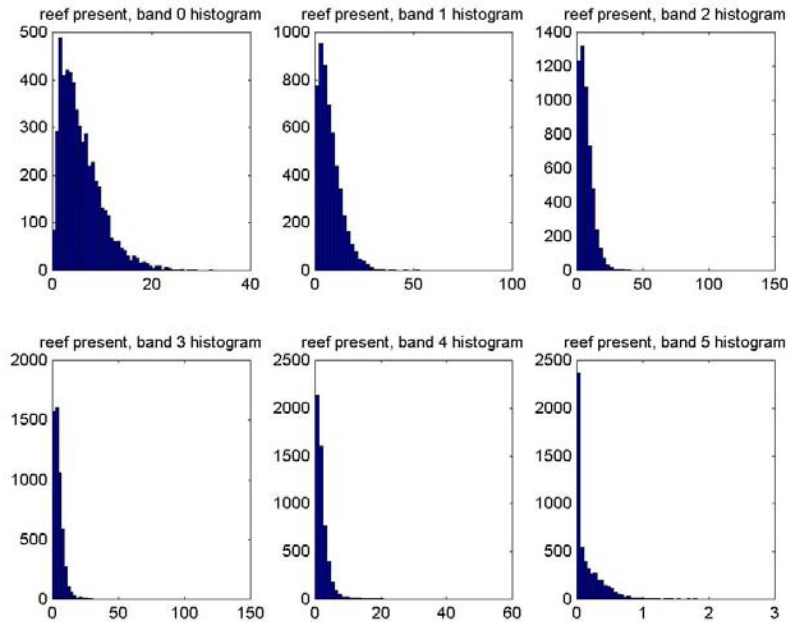


Figure 3, Histograms of SeaWiFS Data Bands with Reefs Present

Table 1, Cumulative Statistics for SeaWiFS Data Bands with Reefs Present

Band	Min	Max	Mean	Median	STD
0	0.00	32.37	5.91	4.870	4.27
1	0.03	97.85	7.71	6.191	6.62
2	0.09	137.14	7.35	5.922	7.15
3	0.15	109.73	4.89	3.833	5.32
4	0.04	57.62	2.28	1.544	3.03
5	0.00	2.52	0.20	0.085	0.28

4.11 Comparison of in situ Data to Lubin’s Model

This section contains the results from comparing Lubin’s model for the four objects against the SeaWiFS remote sensing data and the in situ observation. Please note that there are no in situ observations available for the non-coral objects. We used an epsilon value of 10.0 and 12.0 and compared all cells in the rows defined above in the distribution analysis. The models assume a depth of 5 meters in the following discussion. For sand, the model and data cells were within the epsilon of 10.0 approximately 6.333% of the time. This jumps to 8.25% when the epsilon is set of 12.0. Again, there is no in situ database to compare against. It is comforting that this number for sand is much larger than the two coral objects, since you would expect to have more sand visible than coral throughout the coverage area. For algae, the model and data cells were within the epsilon of 10.0 approximately 3.2123% of the time. This jumps up to 4.0120% of the time with an epsilon of 12.0. For the two types of coral, *Montastrea cavernosa* and *Dichocoenia stokesi*, the model data cells were within the epsilon of 10.0 approximately 6.4321% of the time in cells that contain coral reefs according to the in situ database. When epsilon is set to 12.0, this jumps to 6.7778% of the time, again constrained to cells containing coral reefs according to the in situ database. Things get much worse when we look at data where there aren’t coral reefs present according to the model. A false positive was defined earlier as a cell where there is no reef present in the in situ data, but the model and the data are within the specified epsilon. Here we get a false positive 7.3333% and 7.4523% of the time with an epsilon of 10.0 and 12.0 respectively. As you can see, we had a very difficult time getting the remote sensing data and the model data to line up. Why? There are at least three possible shortcomings in the existing system to hinder this complete

analysis. First, there is no depth associated the remote sensing grid cells. Without this information, it is difficult to know which model to test against a specific of remote sensing data. In the near term, a SeaWiFS data quality flag, which indicates the water is too shallow for accurate ocean color, could be used to filter the data for the suitable model. In the longer term, the XML produced could be combined with another product, such as the Digital Elevation Model (DEM) to get a more accurate estimation of the average cell depth. Second, the data we chose to analyze may be too large for the model to be effective. It would be interesting to see if better results could be found with a data set of higher spatial resolution, such as a SeaWiFS Level 2 Local Area Coverage data set or even Landsat Thematic Mapping. Third, this may be cause because the wrong coral reef model is being applied. We only check for the existence of a reef, not the type. The type data within reefBase doesn't typically have the specific coral species. Without this information, it is not possible to determine the exact model to apply. This shortcoming could be corrected by pulling the species data for particular coral reefs from 'analog' sources, such as the World Atlas of Coral Reefs.¹

4.12 Mean band values as an alternative model

One thought we had, was to use the mean value from each band of cells with coral reefs present as an alternative model. We tested this alternative using Matlab. When epsilon equals 10.0, the average hit is 73.33. It jumps up to 86.44 percent when epsilon equals 12.0. This is a fairly good match against the cells that contain coral reefs. In order to answer how well, we needed to run the mean band values through all points within a region to see the hit and miss counts. We started with the region from rows 990 to 1020, giving a bounding box of approximately 6.5 degrees North to 4.2 degrees South over all 360 degrees. In 289 cells with coral reefs present in this region according to the reefBase database, 185 of them had a model and remote sensing RMS value of less that an epsilon of 10.0. This is a hit rate of about 67%. However, in the 132834 cells where there is no reef present according to the reefBase database, 58589 cells had a model and remote sensing RMS value of less than the same epsilon. This is a false positive rate of approximately 44%.

4.13 Hyperspectral sensors

One area that may help yield significantly better results is the use of hyperspectral sensors and appropriate data analysis techniques. Imaging spectrometers collect a very large number of wavelength channels at the same time. Hyperspectral systems collect on the order of hundred of wavelengths per pixel. For example, the CASI 2 sensor collects 288 bands of data for each pixel imaged. This differs substantially from multispectral systems, which collect on the order of tens of bands per pixel. For example, the Landsat Thematic Mapper sensor collects three bands in the visible, one in the near infrared, two in the mid infrared, and one in the thermal infrared only. Another way to distinguish hyperspectral from multispectral data is the channel size. Hyperspectral sensors have very small wavebands when compared to their multispectral cousins. For example, the Landsat Thematic Mapper channels vary from about 60 nm in channel 3 to 270 nm in channel 7. This varies considerably from a hyperspectral sensor, such as the CASI 2, which has a spectral resolution of 2.2 nm FWHM at 650 nm. By having so many small wavebands over such a large portion of the electromagnetic spectrum, it is possible to use hyperspectral data to determine the unique spectral signature of remotely sensed objects. Since hyperspectral datasets offer a tremendous increase in spectral resolution, it is very likely that analysis based on them will match the models much closer. There has been some success in the utilization of hyperspectral systems for the identification of coral reef features. The reader is referred to Holden and Ledrew work in the International Journal of Remote Sensing.¹⁶ Another example is NOAA's use of hyperspectral imaging to remotely monitor coral health and mortality at Buck Island National Marine Reserve.¹⁷

5. CONCLUSIONS

This paper presents two major results. First, as the analysis section details, the comparison of SeaWiFS data against Lubin's spaceborne spectral model proved less successful than initially hoped. While less than ideal, our results do show that there is considerable promise in this approach. Research areas we are currently pursuing include utilization of datasets with higher spatial resolution and higher spectral resolution. We are considering using different hyperspectral datasets taken from airborne sensors such as Aurora and AISA and spaceborne sensors like EO-1 Hyperion. We intend to investigate fusing the remote sensing observational data with ocean depth information. We believe this multi-sensor data fusion approach will further improve our outcome. The second major result this paper provides is a valuable overview to the techniques and trade-offs associated with designing and implementing a fairly detailed architectural and functional description of a web-delivered database-driven remote sensing analysis application.

ACKNOWLEDGMENTS

The George Mason University VAccess project and the TRW Foundation provided funding support to this work.

REFERENCES

1. Marine Program, UNEP World Conservation Monitoring Centre *World Atlas of Coral Reefs* UN Publishers, 2001.
2. ReefBase homepage, available on the web at URL <http://www.reefbase.org>
3. Lubin, Dan, "Spectral Signatures of Coral Reefs: Features from Space", *Remote Sensing of the Environment*, 75:127-137 (2001), page 127.
4. Holden, H and Ledrew, E., "Hyperspectral Identification of Coral Reef Features" *International Journal of Remote Sensing*, 20(13):2545-2563, page 2546.
5. Lubin, Dan, "Spectral Signatures of Coral Reefs: Features from Space" *Remote Sensing of the Environment*, 75:127-137 (2001), page 128.
6. Lyzenga, D., "Remote Sensing of bottom reflectance and water attenuation parameters in shallow water using aircraft and Landsat data.", *International Journal of Remote Sensing*, 2:71-82 (1981).
7. Maritorena, S., "Diffuse reflectance of oceanic shallow waters: influence of water depth and bottom albedo", *Applied Optics*, 39:1689-1703 (1994).
8. Mazel, C., "Diver-operated instrument for in situ measurement for spectral fluorescence and reflectance of benthic marine organisms and substrates", *Optical Engineering*, 39:2612-2617 (1997), page 2614.
9. Stamnes, T., "Numerically stable algorithm for discrete-ordinate method radiative transfer in multiple scattering and emitting layered media", *Applied Optics*, 27:2502-2509.
10. SeaWiFS Level 1A and Level 2 HDF Dataset Guide Document, Version 2.0, June 2000. Available on the web at the following URL: http://daac.gsfc.nasa.gov/DATASET_DOCS/SeaWiFS_L1A2_Guide.html
11. NASA "SeaWiFS Level 3 Dataset Guild Document, Version 2.0". Available on the web at URL http://daac.gsfc.nasa.gov/DATASET_DOCS/SeaWiFS_L3_Guide.html
12. Sun Microsystems, "Java Certified Architect Course Notes", 2001, page 132.
14. Folk, M., "HDF as an Archive Format: Issues and Recommendations". Available on the web at URL <http://hdf.ncsa.uiuc.edu/archive/hdfarchivefmt.html>
15. NASA "HDF-EOS: The Universal Language". Available on the web at URL <http://earthobservatory.nasa.gov/Study/HDFEOS/index.html>
16. Holden, H. "Hyperspectral Identification of Coral Reef Features", *International Journal of Remote Sensing*, Vol. 20, No. 13, pages 2545-2563.
17. "Hyperspectral Imaging Monitors Coral Health". Available on the web at URL <http://carstad.gsfc.nasa.gov/topics/hyperspectralarticles.htm>