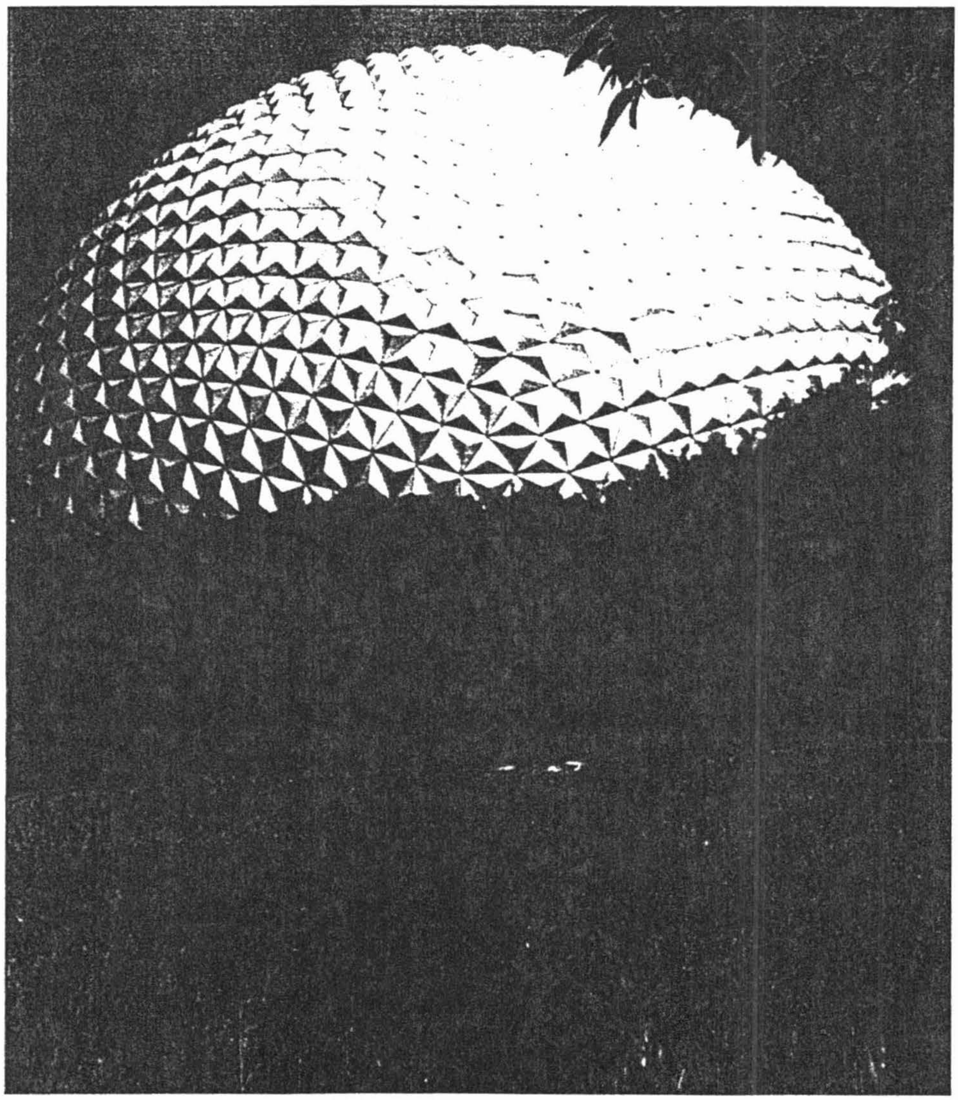
98-LV

*Proceedings of the First International Conference*

# Geospatial Information in Agriculture and Forestry

Decision Support, Technology, and Applications



Volume I

1-3 June 1998

Lake Buena Vista, Florida, USA

## SESSION D: GEOSPATIAL INFORMATION—ANYBODY, ANYWHERE, ANYTIME

# Landsat Viewer: A Tool to Create Color Composite Images of Landsat Thematic Mapper Data*

Naphtali D. Rishe, David Barton, Nagarajan Prabhakaran, Martha E. Gutierrez, Maria C. Martinez, Rukshan I. Athauda, Alejandro Gonzalez

High Performance Database Research Center
School of Computer Science
Florida International University
University Park, Miami, FL 33199, U.S.A.
(305) 348-1706, Fax (305) 348-1705
hpdrc@cs.fiu.edu, http://hpdrc.cs.fiu.edu

ABSTRACT

Data from Landsat Thematic Mapper (TM) sensors detect reflected radiation from the Earth surface in the visible and infrared wavelengths. The characteristics of the TM bands can be selected to maximize their capabilities for detecting and monitoring different types of the Earth resources. The ground area covered by one Landsat scene is over 34,000 square kilometers and represents about 260 MB of data.

This paper provides a description of a web page interface to a Landsat TM Semantic Database, being developed at the High Performance Database Research Center (HPDRC) at Florida International University. The web interface allows the user to graphically select areas of the Earth to be examined showing latitude and longitude coordinates. The user can further choose the size of the Landsat scene or quad and the color composite image to view based on the seven available sensors. The color composite images are generated in real time in 24-bit color and are subject to various user selected picture enhancement algorithms before being recomposed and exported to the client program in a standard image format. This Landsat Viewer facilitates image processing from the Internet.

## 1.0 INTRODUCTION

It has taken the database community some time to recognize the impact of the uniqueness of spatial data. In the 1970's and 1980's, the database community lumped every kind of data other than fixed-format records (including spatial data) into a heterogeneous group called 'non-standard data'. It was tempting to extend relational database technology, with this simple conceptual structure, to handle all kinds of data. However, relational data is not just a way to represent data, it also implies or suggests certain access algorithms that are particularly efficient on data naturally represented by rows and columns. It is true that much business data is of the tabular form and that this data lends itself to such regular access patterns. However, if we force spatial data into tabular form, for example by introducing relations faces, edges and vertices, this may have harmful consequences. Geometric proximity is not reflected by proximity in memory. For example, all vertices no matter how far apart in space are stored contiguously in the same relation, whereas a vertex and its incident edges and faces are scattered all over storage. This may have grave consequences when data is stored on disk, where instead of accessing one entire object as a unit we may have to gather bits and pieces of this object in many separate disk accesses (Nievergelt94).

---

*Presented at the First International Conference Geospatial Information Agriculture and Forestry , June 2, 1998, Buena Vista, Fl.

**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.

## 1.1 HPDRC

The High Performance Database Research Center (HPDRC) is a division of Florida International University (FIU), School of Computer Science. It conducts research on database management systems and various applications, leading to the development of new types of DBMS, new database techniques, and the refinement of existing ones. HPDRC's largest project is the development of algorithms and a prototype of a massively paralleled Semantic/ Object Oriented DBMS. Our system is useful for most typical database applications, as well as for specialized domains such as Earth Sciences.

## 1.2 RAC

The NASA Regional Applications Center (RAC) at FIU is a subdivision of HPDRC. NASA has established approximately 15 RAC's across the country at this time. The RAC Program was initiated by NASA Goddard Space Flight Center's (GSFC) Applied Information Sciences Branch, Code 935, to extend the benefits of its information technology research and cost-effective system development to a broader user community. The RAC objectives are based on the goal of fostering the use of environmental and Earth resource data by regional institutions. The ultimate goal of the RAC is to establish a fundamental set of remote sensing technologies that can be assembled by a specific user community, to meet the information needs of that community.

# 2.0 SEMANTIC DATABASE

The HPDRC's Semantic DBMS is based on the Semantic Binary Model. In the Semantic Binary Model, the 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(Rishe92b).

## 2.1 DESCRIPTION

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 (Rishe92b).

The efficient retrieval and updates are a requirement of the semantic database. Requests are maximized by decomposing queries into atomic retrieval operations and each atomic retrieval request normally requires only one disk access. 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 transaction can be generated by a program fragment containing numerous update commands contained among other computations. 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 (Rishe92b).

The Semantic Database is perceived by its users as a set of facts about objects. These facts can state that the objects belong to a category, they can state that there is a relationship between objects or they can be fact relating objects to data, such as numbers, texts, dates, images, etc (Rishe92b). 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.

- Compression will not negatively affect the efficiency of a search since the logical B-tree order between data and index blocks will not be altered by such compression.
- Compression will reduce the amount of I/O time required for long sequential data transfers which are commonplace in scientific database applications.
- In a distributed system, compression will significantly decrease the amount of communication overhead.
- Data compression will be transparent to the user.

Even without special compression algorithms, the semantic DBMS is very storage efficient. However, contemporary compression techniques can help achieve significant savings (Rishe92a).

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 need to be considered.

## 2.2 LANDSAT SEMANTIC DATABASE SCHEMA

We used a Semantic Binary Database for the storage of Landsat TM data. The first step involved in creating the database is the design of the schema. HPDRC has acquired some Landsat TM data of scenes and quads observed by Landsat 4 and 5. These spatial data along with its meta-data are integrated to the database by the schema design. Fig. 1 shows the current schema design for the Landsat TM database.

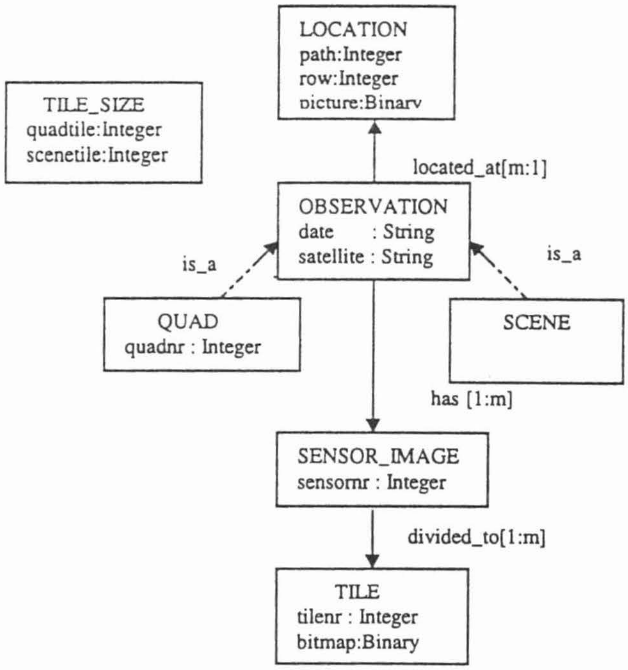



Fig. 1 Schema for Landsat Thematic Mapper database

Description of schema

LOCATION—category (A catalog of locations defined by path and row which is the coordinate system used for Landsat observations)

OBSERVATION—category (A catalog of observations which refers to observations made by a Landsat satellite for a particular date)

QUAD—category (A catalog of quads which is a particular area of observation specified by the quadnr)

SCENE—category (A catalog of scenes which is the area of observation)

SENSOR_IMAGE—category (A catalog of images observed by a sensor on-board Landsat satellite sensor specifies the sensor which made the observation)

TILE—category (A catalog of tiles which are segments of a sensor image)

TILE_SIZE—category (A category of TILE_SIZE which contains the sizes of quad and scene tiles)

located_at –relation from OBSERVATION to LOCATION (m:1,total) (An observation must have a location that it observes. There are many observations with the same location)

divided_to–relation from SENSOR_IMAGE to TILE (1:m) (A sensor image is divided in to smaller segments called tiles)

has–relation from OBSERVATION to SENSOR_IMAGE (1:m) (An OBSERVATION consists of many sensor images observed by different sensors in the satellite)

path – attribute of LOCATION of type Integer (The path number of the coordinate system)

row – attribute of LOCATION of type Integer (The row number of the coordinate system)

picture – attribute of LOCATION of type Binary (The image depicting the region covered by scene)

date – attribute of OBSERVATION of type String (The date when the observation was made)

satellite – attribute of OBSERVATION of type String (The name of satellite which made the observation)

quadnr – attribute of QUAD of type Integer (The quad number which specifies the area of observation)

sensornr – attribute of SENSOR_IMAGE of type Integer (The sensor number which specifies the sensor that observed the image)

tilenr – attribute of TILE of type Integer (The tile number which identifies the tile)

bitmap – attribute of TILE of type Binary (The binary data observed by the sensor on-board the satellite)

The LOCATION category specified by a path and row number gives a specified location for an OBSERVATION category. An OBSERVATION object has a relation located_at, of cardinality [m:1], with LOCATION specifying that there are many observations for a particular location. Since the relation is total, every OBSERVATION must have a particular LOCATION. Picture attribute contains an image of the area covered by the scene for the particular location . OBSERVATION is described by a date specifying the date of observation, and the name of satellite, which made the OBSERVATION. An OBSERVATION object could be either a SCENE or QUAD (specified by a quad number). Every OBSERVATION can have up to 7 sensor images observed by the seven instruments on board Thematic Mapper instrument. This is specified by a has[1:m] relation and SENSOR_IMAGE category. Sensor_Nr specifies the sensor that made the observation. Each sensor image is divided into smaller sub-images called tiles. A scene's sensor image is divided into 930 tiles and a quad's sensor image is divided into 378 tiles. The justification is that in most applications only a very small percentage of the sensor image is used for viewing by the user. Hence, when a request is made to view a particular part of the image, queries to the specified tiles can be made and the image created, instead of retrieving the whole image at once and selecting out regions to view. This methodology will enhance the performance in most cases when only a small number of tiles are retrieved. This is captured by a divided_to[1:m] relation from the SENSOR_IMAGE to TILES category. TileNr specifies the tile and Bitmap attribute is used to store the observed data.

## 3.0 DESIGN OF THE APPLICATION

The overall structure of the application consists of design and implementation of 3 main components :

- The design and implementation of a storage-retrieval medium for Landsat images. In our case, it is a Semantic database developed at HPDRC. This is described in the previous section.
- The design and implementation of a client program that acts as the front-end of the application.
- The design and implementation of a server program that queries the database to fulfill the requests of the client.

The client program will interact with the user to compose a query using easy-to-use Graphical User Interface (GUI). It will send the user's request to the server for processing, and will display the results that are received from the server for a particular request.

The server program acts as the back-end of the application interacting with the database to fulfill the requests of the clients. This program will query the database to obtain for the client's requests and communicate with the client to send and receive results

## 3.1 CLIENT PROGRAM

The main task of the client program is to obtain the user's query using an easy-to-use GUI. The client shows a map like Fig. 2(a) (map of US) from which the user select a state (for example Florida, see figure 2(b)).

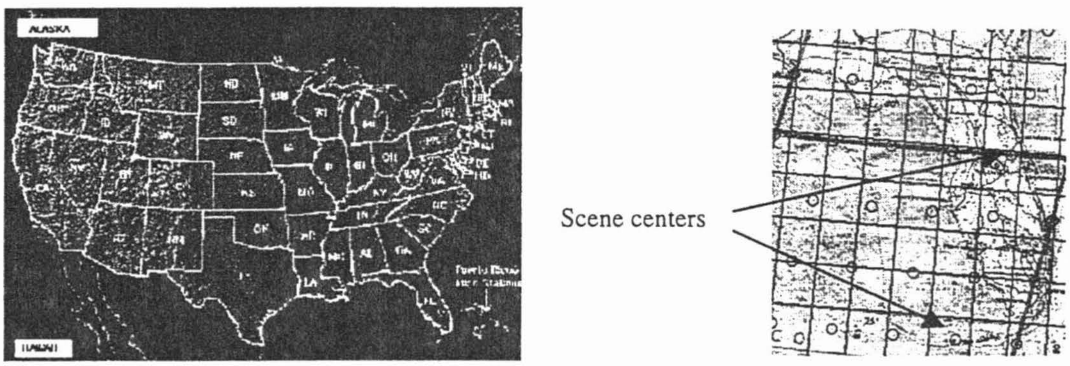
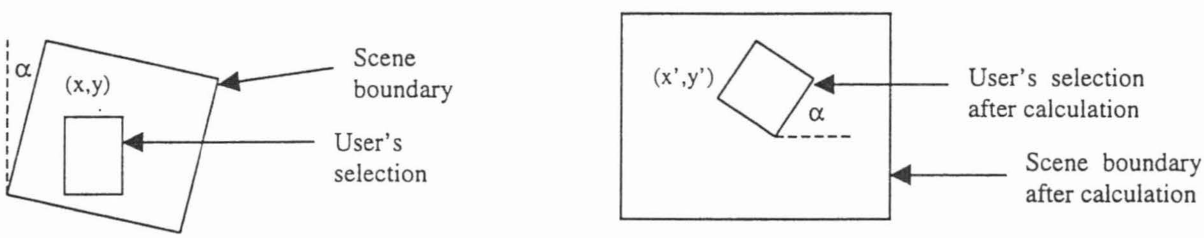



Fig. 2 (a) Map of the US from which the user selects a state of interest (b) Map of the state of Florida with the scene centers marked by circles

The user in most cases requires only seeing a small area of this region. The map of Florida is marked with scene centers. The user selects a scene center of interest, which gives the client program a particular path and row number that the user requires. This information is transmitted to the server program as Query1. The results of Query1, which are received by the client program, contains meta-data on all the Landsat Thematic Mapper images present in the database for the selected path and row. Also, a name of a picture file, which contains the region covered by the scene for the selected path and row, are received. The picture file depicting the selected region is displayed from which the user selects a smaller region of interest.




(a) (b)

Fig. 3 (a) The alignment of an angle α to the horizontal axis in the scene boundary (b) The scene boundary and user's selection.

The client program then calculates the tiles for quads and scene involved in the selected region. There is an alignment in the scene boundaries of the displayed picture (see Fig. 3 (a)) when compared to the user's selection rectangle. This alignment is due to the path Landsat satellite takes as it observes the Earth. Thus, we need to find a rectangle, engulfing the user's selected rectangle, and whose sides are parallel to the sides of the scene boundaries to calculate the tiles required to be queried for the user's selection. This calculation is performed using the following method.

Assume that there is an alignment angle of α as shown in the Fig. 3(b). We move each vertex (x,y) of the user's selected rectangle by an angle of α to calculate (x',y'). This is performed by the following equation:

$$x' = x \cos \alpha - y \sin \alpha$$
$$y' = y \sin \alpha + y \cos \alpha$$

The same calculation is performed for the scene boundary co-ordinates.

The engulfing rectangle's, shown by the dashed lines in Fig. 4(a), are calculated as points A, B, C, D

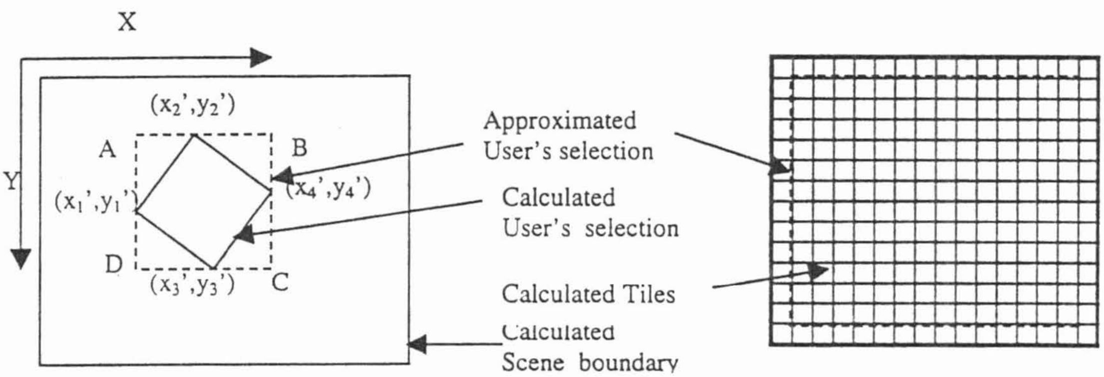



Fig 4(a) Approximated user's selection (b) The selection of tiles to the approximated user's selection

With these co-ordinates we are able to approximate the user's selected rectangle and calculate the tile numbers for the scene and quad that need to be queried (Fig 4(b)). Once the tiles required by the user's selection are calculated, it is checked whether the tiles are present in the database. Since the results of Query1 contain all meta-data of Landsat images present in the database for the particular path and row, this can be easily performed. Next, the dates available for the selected region are displayed from which the user selects a date of the observation he/she prefers to view the image. Next, the sensor images available are computed and displayed for the user to select to produce color-composite images. Note that we require sensor images to be computed because the user's selection may span across multiple quads and it is possible that the database may not have sensor images of a quad selected. This is resolved by finding the intersection of sensor images present in database for the user's selected quads. Also, image enhancement could be done to the resultant image by applying a filter. Next the selected tiles, date, sensor numbers and filters are composed into a query (Query2) and transmitted to the server program to process. Finally, the result of Query2 is an image created by applying the selected sensors to red, green and blue and the selected filter. The client program will display this image.

The client program is implemented as a JAVA applet running on a WWW browser. The states (e.g. florida.html) where the user clicks on a preferred state to view an image, is implemented in HyperText Mark-up Language (HTML) with path and row numbers embedded as parameters. The connection between the client and the server is handled by a TCP/IP (reliable byte stream) connection using socket implementation.

## 3.2 SERVER PROGRAM

The main method or controlling body of the server program provides two major functions. It opens the Landsat database and waits for a client to contact. When a client does make a connection, it creates a thread or process and lets the client communicate with the new process for its future transactions as it continues to wait for more clients.

The client requests for two different types of queries from the server.

1. Query1: For a given path and row number, provide meta-data on all the Landsat quads and scenes present in the database for the particular location along with a picture of the region.
2. Query2: For a given path, row and sensor numbers, query the bitmaps for the selected tiles.

The server waits for query from the client. On receiving a request, checks whether it is of type Query1 or Query2 and perform the necessary tasks accordingly. It then sends the results for the queries to the client and waits for another query. The server program is implemented in C++. It uses the C++ interface developed for the Semantic Binary Database at HPDRC to query the database. The server runs on a Sun Sparc station using Solaris as the underlying Operating System. Note that the server side is kept simple intentionally so as to make the application easily portable between different schemas of the database. The main components that require to be modified are the two queries.

# 4.0 IMAGE ENHANCEMENTS AND COLOR-COMPOSITE IMAGES

The application gives the capability of producing color-composite Landsat images by applying any of the different sensors images to the RGB color model. Applying filters can further enhance these resultant images. Landsat viewer has the following filters:

1. Linear Contrast Enhancement (Jensen96)
2. Low Frequency Filtering - *mean filter* ($LFF_{5,out}$) (Jensen96)
3. High Frequency Filtering ($HFF_{5,out}$) (Jensen96).
4. Linear Edge Enhancement : Vertical (Jensen96).
5. Linear Edge Enhancement : Horizontal (Jensen96).
6. Linear Edge Enhancement : NE Diagonal (Jensen96).
7. Linear Edge Enhancement : SE Diagonal (Jensen96).
8. Emboss East (Jensen96).
9. Emboss NW (Jensen96).
10. Compass Gradient Masks - North (Pratt91)(Jain89).
11. Compass Gradient Masks - NE (Pratt91)(Jain89).
12. Compass Gradient Masks - East (Pratt91)(Jain89).
13. Compass Gradient Masks - SE (Pratt91)(Jain89).
14. Compass Gradient Masks - South (Pratt91)(Jain89).
15. Compass Gradient Masks - SW (Pratt91)(Jain89).
16. Compass Gradient Masks - West (Pratt91)(Jain89).
17. Compass Gradient Masks - NW (Pratt91)(Jain89).
18. Compass Gradient Masks - vertical (Richards86).
19. Compass Gradient Masks - horizontal (Richards86).
20. Compass Gradient Masks - Mask R, diagonal (Richards86).
21. Compass Gradient Masks - Mask S, diagonal (Richards86).
22. Laplacian Convolution Masks - Mask T (Jahne91)( Pratt91).
23. Laplacian Convolution Masks - Mask U (Jahne91)( Pratt91).
24. Laplacian Convolution Masks - Mask V (Jahne91)( Pratt91).
25. Nonlinear Edge Enhancement - Sobel edge detector (Jensen96).
26. Nonlinear Edge Enhancement - Robert's edge detector (Jensen96).

Items 2 - 26 are categorized as *Spatial Convolution Filtering*. Items 4 - 24 are filters used in *Linear Edge Enhancement*. We used 3 X 3 convolution masks in applying these filters.

Each color in the RGB color table has intensities of up to 255 colors, providing for $2^{24}$ different colors. (24 Bit Color) Each of these Bands has a very limted range in the RGB Color Table. Putting the Three Band Combinations to the Screen in RGB is not enough to fully see the entire image in detail. In a Landsat Image combining three bands without any further image processing will produce an image in very low contrast, some features may be hard to see with the humen eye. As a solution to this, we use an image processing algorithm called "Linear Contrast Stretching", that stretches the colors in the image to use the entire color. This algorithm must be done with each band used in the RGB Combination. For each band, you find the maximum digital number, and the minimum digital number. With these two numbers, you can apply this formula for each pixel's digital number (DN) in the band:

$$DN' = (255*(DN - Min))/Max-Min$$

Using this formula, the image is utilizing the entire Color Table, making it easier to view the image in full color. To explore the full information content of Landsat images, higher order image processing algorithms such as Histogram Equalization, and Band Ratios can be used.

## 5.0 EXTENSIONS

An important extension is to store large amounts of Landsat Thematic Mapper data in the Semantic database so as to allow the user to easily retrieve and access a larger data set. Another extension to this application would be to use a new technology like Common Object Request Broker (CORBA) as the implementation medium for communication between the client and server. Currently, the communication is via a byte stream (TCP/IP) where the query and the results are put in a certain format and transmitted to the receiver. This introduces a new task of retrieving the byte stream at the receiver and processing the message. This phase can be eliminated with the use of technology as CORBA or object serialization whereby objects can be transmitted, hence does not require processing at the receiver's side to retrieve the message. This may result in a performance gain.

## 6.0 REFERENCES


B. Jahne , *Digital Image Processing,*. New York: Springer-Verlag, p383, 1991.

A. K. Jain, *Fundamentals of Digital Image Processing,* Englewood Cliffs, NJ, Prentice Hall, pp.342-357, 1989.

J. R. Jensen, *Introductory Digital Image Processing - A Remote Sensing Perspective,* 2nd ed. NJ: Prentice Hall, pp139-165, 1996.

J. Nievergelt, M. Freeston, "Special Issue Editorial Spatial Data: applications, concepts, techniques", *The Computer Journal*, Vol. 37, No. 1, 1994.

W. K. Pratt, *Digital Image Processing,* 2nd ed. New York:Wiley, p698, 1991.

J.A. Richards, *Remote Sensing Digital Image Analysis,* New York: Springer -Verlag, p281, 1986

N. Rishe, Accepted grant proposal to NASA. Grant NAGW-4080. 1992a.

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