Dynamic Mosaicking of Heterogeneous Digital Images*

Andriy Selivonenko, Nagarajan Prabakar and Naphtali Rishe

High Performance Database Research Center (HPDRC), School of Computer Science

Florida International University, Miami, FL 33199

Tel: (305) 348-6885 E-mail: selivona@cs.fiu.edu Fax: (305) 348-1705

Abstract

There is a high demand for mosaicking of digital images. The digital images are of different formats, different resolutions, band combinations, sizes, etc. This increases the complexity of the mosaicking of images. We propose a dynamic, real-time approach for mosaicking digital images of different temporal and spatial characteristics into tiles. Also, the source images at the time of mosaicking could be accessed from LAN locations and remote Internet locations. Further any specific source image can be reconstructed from the mosaicked tiles. This dynamic approach reuses digital images upon demand and generates mosaicked tiles only for the required region according to user's requirements such as resolution, temporal range, target bands, etc.

Keywords: Mosaicking, Digital Images, Raster Data, Tiling

1. Introduction

Digital images are being acquired in large scale by satellites, aircrafts, and other sources. The recent advancement in image processing capabilities of personal computers has increased the demand of digital image applications. Most of the spatial images are of very large size and they often spatially overlap with each other. The large size of these images causes tremendous network traffic, and requires huge disk space and memory at the client site. Hence the images are mosaicked to remove overlaps and segmented into tiles for effective accessibility [3, 9].

* This research was supported in part by NASA (under grants NAGW-4080, NAG5-5095, NAS5-97222, and NAG5-6830) and NSF (CDA-9711582, IRI-9409661, HRD-9707076, and ANI-9876409).

The digital images acquired by several sources differ in size, format, resolution, number of bands, etc [10]. This increases the complexity of mosaicking images. For example, a user application may require mosaicking of a pair of images of different formats or of different time periods for a specific region. Static mosaicking of images for all possible mosaicking requirements for all regions is computationally and storagewise prohibitive. Similarly, R-tree based solutions [6, 7, 8] would be suitable only if all image overlaps are known apriori and the number of overlaps is not very large. As all the mosaicking requirements are not known in advance, the dynamic mosaicking approach provides an efficient alternative. This dynamic method of mosaicking scheme reuses the digital images upon need and integrates image segments that are relevant to the user requirement.

The next section outlines the properties of different images. Section 3 describes the dynamic mosaicking scheme and the related computational techniques. Subsequently, implementation issues are addressed in Section 4. The final section presents our conclusions.

2. Characteristics of Digital Images

Digital images are classified as different types based on the method of acquisition (satellite, aircraft, static sensors, etc.) as well as the image format (TIFF, JPEG, BMP, PPM, etc.). Images are further characterized by properties such as temporal, spatial resolution, image size, number of bands, and location as below:

- For a given region, we may have several images of the same type that were captured at different periods. Also, these images may be acquired by different sources. Since applications require mosaicking of images that are from different sources as well as images that are acquired at different times, we need a dynamic mosaicking scheme.
- Images differ in spatial resolution. For instance, each Landsat image pixel corresponds to 28.5m x 28.5m area whereas an aerial photography pixel corresponds to 1m x 1m area. Images of different resolutions for a specific area need to be transformed to a uniform resolution during mosaicking.

- The size of images varies. Mosaicking of images of different sizes will create image fragments of various sizes. The mosaicking process should be able to handle arbitrary image fragments.
- Each type of image may have a different number of bands; i.e. each band corresponds to a spectral reflectance for a wavelength bandwidth and contains one value for each pixel of the image. For display purpose, spectral values from several bands of the image are used to generate composite view. The Aerial photography images contain 3 bands; Hyperspectral imagery has 220 bands; LandSat satellite images have 7 bands; and IKONOS satellite images have 5 bands (4 bands with 4m/pixel resolution, 1 band with 1m/pixel resolution).
- The images could be located on the local system, accessible through LAN, or from Internet (http, ftp servers).

The dynamic mosaicking system needs to take into account all the above properties of images that have to be mosaicked at runtime effectively.

3. Dynamic Mosaicking Approach

In this section, first we introduce a grid structure to store image segments and to facilitate dynamic mosaicking. Then the mosaicking process is presented for homogeneous and heterogeneous raster data sets.

3.1. Structure of Dynamic Mosaicking System (DMS)

The dynamic mosaicking system employs a zoomable tile grid (ZTG) structure. A ZTG consists of several zoom planes (ZP) where each ZP corresponds to a specific zoom resolution. Each ZP contains a rectangular array of tiles where each tile contains raster data for a specific area. The ZP of the finest resolution of ZTG is referred to the base zoom plane (BZP). Other ZPs are derived from the BZP. We employ a composition strategy for any arbitrary resolution (Res_X, Res_Y) to generate coarser ZPs. For instance, each 2 x 2 tile area of the BZP may constitute one tile of the next coarse resolution ZP. The derived ZPs facilitate faster access of zoomed out images at the cost of additional disk storage.

DMS uses two types of ZTG, namely Static Zoomable Tile Grid (SZTG) and Dynamic Zoomable Tile Grid (DZTG) as shown in Figure 1. For each image a separate SZTG is constructed by first segmenting the image into fixed size tiles and generating the BZP of the grid. Subsequently, all upper ZPs are derived one by one for the grid. This process of building SZTG is performed offline. When a part (extension or missing part) of an image newly arrives, it can be tiled and combined with the corresponding SZTG. The SZTGs are used at runtime as they needed. For each user request of dynamic mosaicking, a DZTG is constructed at runtime by accessing the SZTG tiles that overlap with the area of interest for the specified resolution. . The DTZGs are transient objects and removed when they are no longer needed.

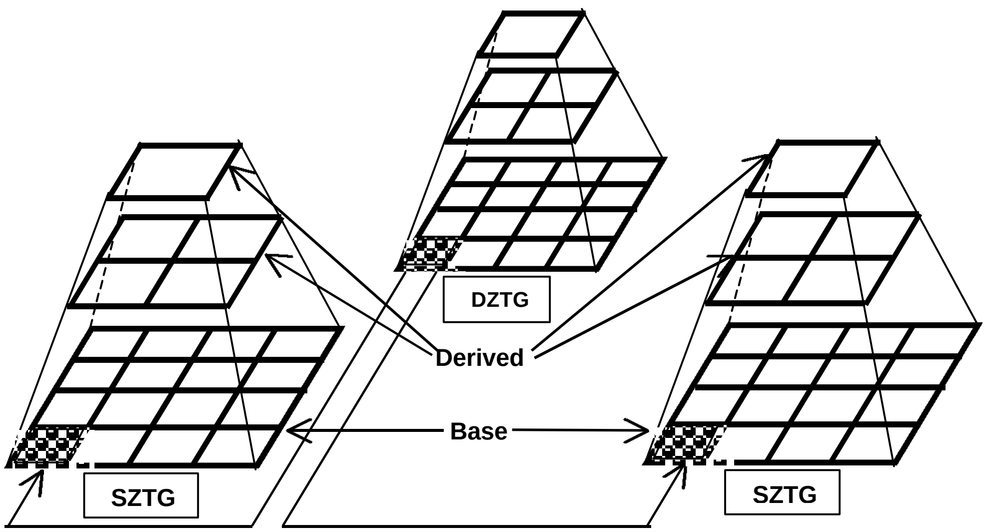

Figure 1. Static and Dynamic Zoomable Tile Grids

The raster data are geolocated with the most widely used Universal Transverse Mercator (UTM) projection. We use UTM as the base projection and build ZTG. In UTM grid, the world is divided into 60 north-south zones [2], each covering a strip of 6° wide in longitude as in Figure 2.

In each UTM zone, coordinates are measured North and East in meters. The northing values are measured continuously from zero at the Equator, in northerly direction. Southerly values are similarly measured from the Equator, south. A central meridian through the middle of each 6° zone is assigned an easting value of 500,000 meters as shown in Figure 3. Grid values to the west of this central meridian are less than 500,000; to the east, more than 500,000.

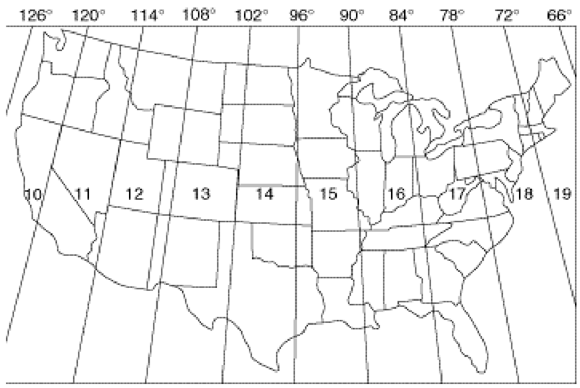

Figure 2. UTM zone coverage for the continental United States.
The Universal Transverse Mercator grid that covers the conterminous 48 states comprises 10 zones - from zone 10 on the west coast through zone 19 in New England.

A rectangular tile geometry is chosen and is fixed for all grid structures within the system. The length and width of tiles are measured in pixels (integer values). We have chosen the default value for both TileLength and TileWidth as 512 pixels. Nevertheless the tile length and width can be chosen to any arbitrary values but must be consistent within the entire grid structures.

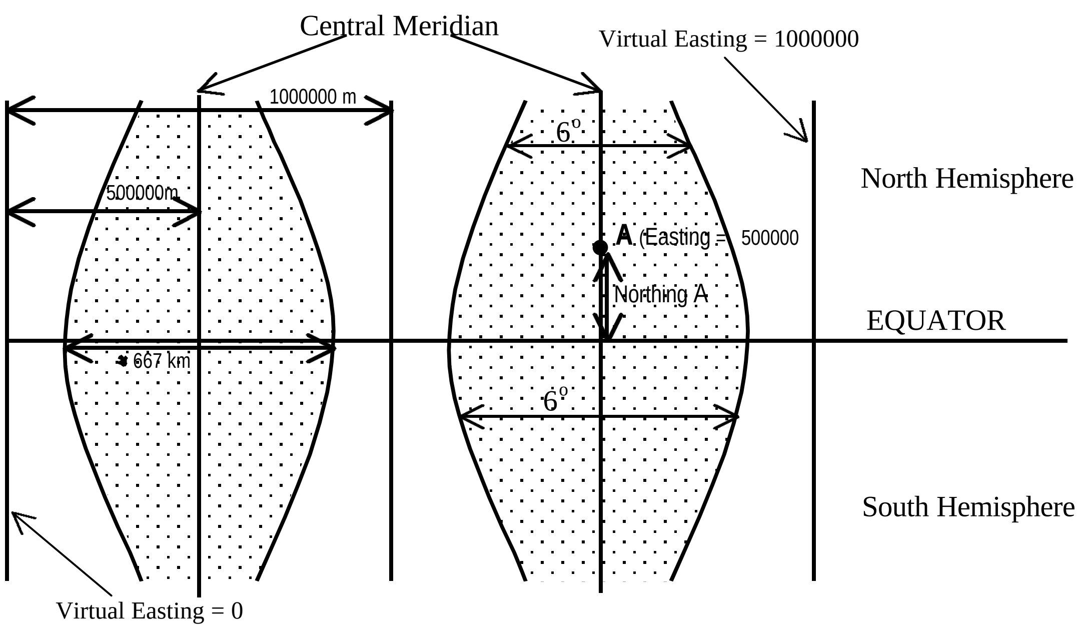

Figure 3. UTM zone layout

The tile boundaries are referred to LeftBorder(X_L), RightBorder(X_R), TopBorder(Y_T), and BottomBorder(Y_B). The values X_L and X_R are specified by Easting and values Y_T and Y_B are given by Northing. For a pair of adjacent tiles, they must have three common values as depicted in Figure 4.

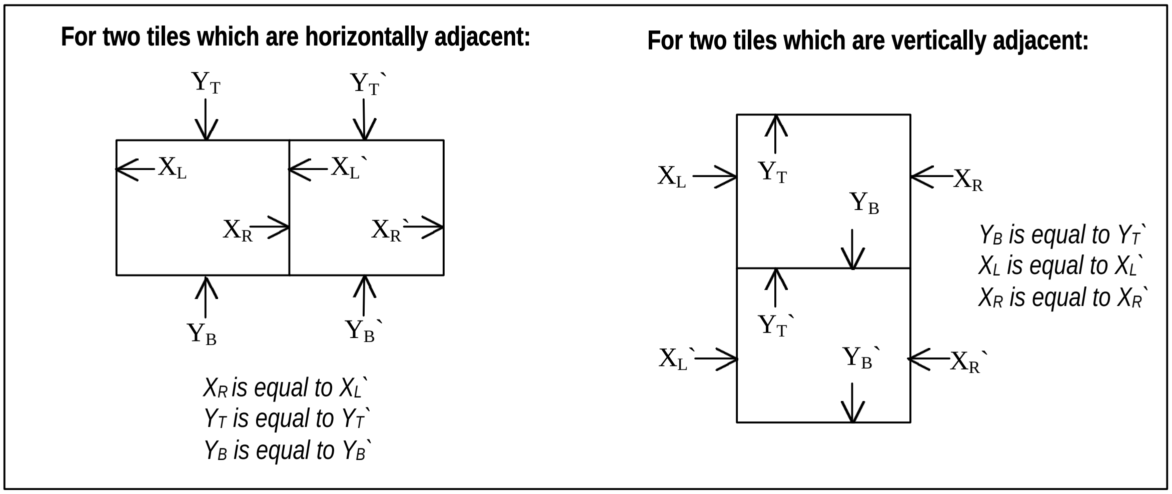

Figure 4. UTM coordinates for adjacent tiles

When an image is segmented into tiles for its SZTG, the tile boundaries are positioned such that tile alignment is mathematically associated with UTM zones as in Figure 5. The reference point of a tile is the upper left vertex of the tile, which is (X_L, Y_T). Let the index of a tile be (Xi, Yi). The UTM coordinates of the tile are determined as below:

X_L = Xi * TileLength * Res_X + 500000

Y_T = Yi * TileWidth * Res_Y

X_R = (Xi + 1) * TileLength * Res_X + 500000

Y_B = (Yi – 1) * TileWidth * Res_Y

Where: Res_X is resolution in m/pixel in horizontal direction

Res_Y is resolution in m/pixel in vertical direction

500000 is Easting offset value of central meridian of the given UTM zone

Since USGS aerial photography raster data has uniform resolution, for tiles of this type of data, both Res_X and Res_Y will have the same value. Nevertheless some raster data could be delivered by original data supplier with Res_X ≠ Res_Y (e.g. Russian satellite SPIN-2 imagery).

Res_X and Res_Y are real numbers. They have fixed value for each Zoom Plane(ZP) within a specific ZTG.

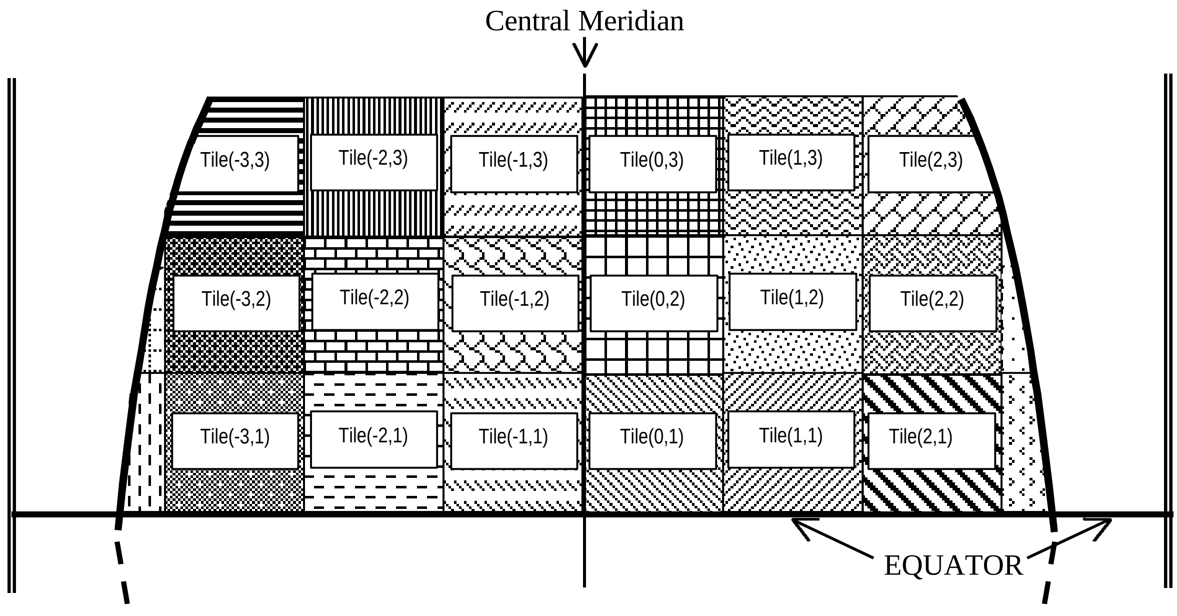

Figure 5. Tile enumeration (Xi, Yi) within a UTM zone

3.2. Mosaicking Process

The dynamic mosaicking process consists of two phases. In the first phase, the original images are cut into tiles. There are two possibilities for the image encoding of the tile:

– the tiles could retain the original image format

– the original image format can be converted into a different format.

Some of the format conversions are PPM -> PNG (retains original quality and reduces disk storage but increases processing overhead), TIFF -> JPEG (significantly reduces disk storage but increases processing overhead and decreases image quality), PNG -> PPM (retains original quality and minimizes processing overhead but increases disk storage). The decision for image format conversion will be based on the processing speed (response time) requirements, disk space availability and image quality (loss tolerance factor). After optional format conversion, the tiles are stored in SZTGs. The metadata (attributes) of the original image are retained as metadata of resulting SZTG.

In the second phase, the SZTGs relevant to the mosaicking request are accessed and a DZTG is constructed. During this phase only one ZP of each SZTG is retrieved according to the resolution of the mosaicking request. If the DZTG is planned for reuse, it could be made persistent and stored as a SZTG. When several SZTGs are mosaicked together, they can be mosaicked in multiple stages as in Figure 6.

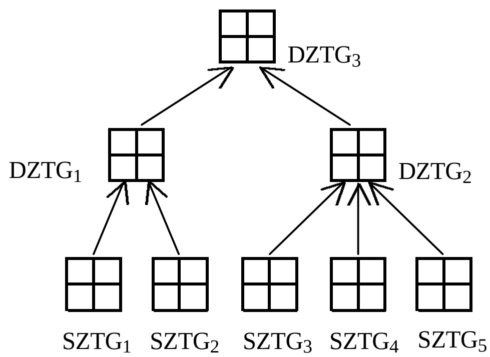

Figure 6. Multiple stage mosaicking

The intermediate mosaicked results could be transient (removed after use) or persistent DZTGs.

Homogeneous mosaicking

The SZTGs used in a mosaicking process are considered as *homogeneous* if the following two criteria are met:

1. spectral bands of the STZGs match all bands needed for the mosaicking request
2. each SZTG has a ZP that matches the resolution specified in the mosaicking request

For each set of tiles on SZTGs with the same (Xi, Yi) and satisfy the above mosaicking criteria, one target mosaicked tile is created. The set of mosaicking criteria could be flexible to allow mosaicking of raster data, which fits to specific time range and/or to specific remote sensing instrument type.

Heterogeneous mosaicking

DMS is able to handle monochrome as well as color data and hyperspectral data. If the first criterion of the homogeneity is not met, we deal with band heterogeneity and generate intermediate spectrally homogeneous DZTGs (as discussed in Section 4) for further homogeneous mosaicking.

If the second criterion of the homogeneity is not met, DMS resolves the resolution mismatch by accessing the ZPs with resolution closest to the resolution of the request and generate intermediate DZTGs with the required resolution.

We suggest as suitable set of resolutions for the derived ZPs be:

$$ {}_{\text{Original}}\text{Res} * 2^1,\ {}_{\text{Original}}\text{Res} * 2^2,\ {}_{\text{Original}}\text{Res} * 2^3,\ \ldots\ {}_{\text{Original}}\text{Res} * 2^n $$

For USGS aerial photography (original resolution = 1m/pixel), the most practical derived ZPs resolutions are 2, 4, 8, 16, 32, 64, 128 m/pixel. Such a set of ZPs allows to generate an effective dynamic mosaicking with any ad-hoc resolution. For instance, if the user requests mosaicking at 28.5 m/pixel resolution from the aerial photography, DMS will generate this data from nearest ZP (i.e. 32 m/pixel).

The selection of the nearest ZP is based on the desirable quality of image and processing speed. There are three approaches:

- optimal quality: when real time response is not needed, optimal quality is achieved by selecting the ZP with resolution that is closest and less than or equal to the requested resolution.
- optimal speed: when best response time is required, select the ZP with resolution that is closest and greater than or equal to the requested resolution.
- optimal compromise of speed and quality: to minimize the loss of quality and loss of speed, two ZPs with resolution closest to the requested resolution are selected and the requested resolution is compared with the geometric mean of the resolutions of the ZPs. If it is less, the ZP with finer ($ZP_{Res} < Req_{Res}$) resolution is selected. Otherwise, the ZP with the coarser resolution is selected.

For SZTGs with Res_X ≠ Res_Y, the extensions to these approaches are presented in [11].

4. Implementation and Results

The Dynamic Mosaicking System has been implemented on the server-side using Sun Java v.1.2.2. The client-server interaction is based on the Java servlet mechanism. Client module is a Java applet, which runs in Java v.1.1 – compliant browser on any remote location (Netscape v. 4.06 or higher; Microsoft Internet Explorer v. 4.0 or higher) [1]. The following types of images have been used: LandSat satellite, USGS Aerial Photography, Emerge Aerial Photography, IKONOS satellite, OrbImage Aerial Photography and SPOT satellite imagery.

All client-server interactions take place via http channels. Each client request initiates a dynamic mosaicking process. The set of parameters of a user request is delivered to the server via HTTP variables using GET and POST methods. After dynamic mosaicking, the tile is returned back to the client as result of HTTP request. Usually client issues many mosaicking requests simultaneously.

There are two types of client requests: single tile request and ad-hoc rectangle area request.

A single tile request includes a set of parameters, which specify the position of the requested tile. Such position is uniquely identified by Xi, Yi, Res_X, Res_Y, UTM Zone number and NorthOrSouthHemisphere values. This type of requests has least processing overhead.

Ad-hoc rectangle area requests can span more than one UTM zone. This type of request should also specify geographical constraints in one of the following two formats:

- X_L, Y_T, $\mathrm{UTMZone}_1$, $\mathrm{NorthOrSouthHemisphere}_1$, X_R, Y_B, $\mathrm{UTMZone}_2$, $\mathrm{NorthOrSouthHemisphere}_2$, ${}_{\mathrm{Target}}\mathrm{Res\_X}$, ${}_{\mathrm{Target}}\mathrm{Res\_Y}$
- Long_L, Lat_T, Long_R, $\mathrm{Lat}_{B,}$ ${}_{\mathrm{Target}}\mathrm{Res\_X}$, ${}_{\mathrm{Target}}\mathrm{Res\_Y}$

Requests of this type are particularly useful for creation of large printouts, further digital processing, data export, and original raster data reconstruction.

All client requests specify the set of dynamic mosaicking options such as the type of image source(s), temporal range and band combinations. The rules for band manipulations are specified within the client request for one of two cases:

- dynamic mosaicking of heterogeneous data sets with different number of bands;
- ad-hoc band manipulations during dynamic mosaicking.

For example, let us consider two datasets: Aerial Photography and LandSat. Neither Aerial Photography nor LandSat provide contiguous coverage of the area. Also, presume that the user wants to mosaick these two datasets together at 28.5 m/pixel resolution to provide better coverage of the area. LandSat has 7 spectral bands, so bands 2 (green), 3 (red) and 4 (near IR) will be used for color infrared tiles generation. Aerial Photography data will be zoomed to 28.5 m resolution using nearest ZP (i.e. 32 m/pixel ZP). There could be 4 possible cases for each tile during the mosaicking process:

1) neither Aerial Photography nor LandSat tile is available. Blank tile will be generated;
2) only LandSat tile is available. LandSat color infrared composite tile will be generated;
3) only Aerial Photography is available. Aerial Photography tile, zoomed to 28.5 m/pixel will be generated;
4) both Aerial Photography and LandSat tiles are available. A mosaicked tile will be generated.

User could specify other pixel rules for mosaicking (case 4) such as:

1) pixel which belongs to image that was acquired later will override the pixel which belongs to image which was acquired earlier. Usually this rule is used with the exception that empty pixel cannot override non-empty pixel;
2) value of resulting pixel is equal to arithmetical median of all non-empty corresponding pixels from all planes.

As we noted before, when client specifies mosaicking constraints that uniquely identify the original data set (some specific DOQQ [4, 5] or specific satellite image), the dynamic mosaicking process reconstructs the original raster data (original DOQQ or original satellite image).

DMS could be configured to store a color tile as a single color composite image (PPM, JPEG) or it could be stored as three separate monochrome tiles – red, green and blue component. JPEG gives better compression ratio for a color composite image than for three separate monochrome images. Each hyperspectral (*n* bands) tile is stored as set of *n* monochrome tiles to facilitate later processing.

5. Conclusion

We presented a zoomable tile grid structure to store image segments with multiple resolutions. This structure facilitates mosaicking of images with different resolutions as well as mosaicking of images from heterogeneous sources. Further, we introduced static and dynamic zoomable tile grid structures where SZTGs are created offline (one grid per image). At runtime, for each mosaicking request, a DZTG is created by reusing relevant tiles of SZTGs.

Also, we outlined criteria for homogeneous mosaicking and described how heterogeneous mosaicking is performed when any of these criteria fails. Mosaicking parameters for single tile requests and ad-hoc rectangular requests as well as mosaicking options for the type of image source(s), temporal range and band combinations were addressed.

The proposed dynamic mosaicking system has been successfully implemented and the system is scalable and platform independent. We believe that the dynamic mosaicking approach presented here, provides a framework for an efficient mosaicking for an arbitrary region from images of different spatial resolutions as well as from images acquired by different sources.

References

1. FIU-HPDRC, *TerraFly project documentation*, http://www.terrafly.com/

2. U.S. Geological Survey. *The Universal Transverse Mercator (UTM) Grid. Fact Sheet Number 157-99 (February 1999)*. http://mapping.usgs.gov/mac/isb/pubs/factsheets/fs15799.html

3. Afek, Y. and Brand, A., (1998) *Mosaicking of Orthorectified Aerial Images*, Photogrammetric Engineering and Remote Sensing, Volume 64, No: 2 pp. 115 –125

4. Fairgrieve, G. L., (1992) *Digital Orthoquads*, U.S. Geological Circular.

5. U.S. Geological Survey, (1999) *Digital Orthophoto Quadrangles* http://edcwww.cr.usgs.gov/Webglis/glisbin/guide.pl/glis/hyper/guide/usgs_doq

6. N. Beckman, H. Kriegel, R. Schneider, and B. Seeger, "The R*-tree: An Efficient and Robust Access Method for Points and Rectangles", Proc. ACM SIGMOID Int. Conf. On Management of Data, pp. 322-331, 1990.

7. O. Gunther, *Efficient Structures For Geometric Data Management*, Lecture Notes in Computer Science 337, Springer-Verlag, Berlin, 1988.

8. Timos Sellis, Nick Roussopoulos, and Christos Faloutsos, "*The R+ - Tree: A Dynamic Index for Multi-Dimensional Objects*", Proc. 13th Int Conf. On Very Large Databases, pp. 507-518, 1987.

9. N. Prabhakaran, S. Sridhar, and N. Rishe, *'A Two Phase Digital Ortho Photo Mosaicking System'*, Proceedings of the International Conference on Imaging Science, Systems, and Technology (CISST'99), June 28 - July 1, Las Vegas, pp. 151-154, 1999.

10. N. Rishe, D. Barton, N. Prabhakaran, M. Gutierrez, M. Martinez, R. Athauda, A. Gonzalez, and S. Graham, *Landsat Viewer: A Tool to Create Color Composite Images of Landsat Thematic Mapper Data*, Proceedings of the International Conference on Geospatial Information in Agriculture and Forestry, June 1-3, Orlando, pp. 529-536, 1998

11. FIU-HPDRC, *TerraFly technical documentation*, http://hpdrc.cs.fiu.edu/terrafly/doc