IOP Conf. Series: Earth and Environmental Science **18** (2014) 012062 doi:10.1088/1755-1315/18/1/012062

Managing and serving large collections of imagery

V Viswambharan[1]
Esri, Redlands, California, USA

Email: vinayv@esri.com

Abstract. A common challenge to organizations working with statewide imagery collections is the sheer volume and processing that has to be dealt with. This paper presentation will illustrate how this challenge has been addressed by way of image services. Image services are becoming increasingly popular as a means to serving out large holdings of imagery to both internal to an organization and external. We will introduce a typical real world scenario and elucidate how we've gone from an imagery holding on disk to a service disseminating dynamically mosaicked imagery – processed on the fly – to a variety of geospatial applications.

1. Introduction

Data management requirements have changed over the years. Various influential factors requiring this change; huge quantities of imagery already exist and these volumes are growing exponentially. Based on the increasing number of sensors and data providers, Estimates for this growth could exceed 30% per year.

There are multiple sources of new imagery, including but not limited to satellites, digital aerial cameras, scanned film, and maps. Not only are the volumes increasing, but also the depth of the imagery in terms of:

- The resolution of the sensors is now much higher than before, resulting in finer details being visible and much larger files to cover the same area.
- The bit depth or spectral resolution is increasing from 8 to 12 to 14 bits per channel. This higher dynamic range in the sensors is providing better spectral detail providing, for example, the ability to see details within shadows.
- Many of the sensors new have more spectral bands. Many aerial cameras capture the near infrared band in parallel with red, green, and blue. Satellites with higher number of bands are being launched as well as hyperspectral sensors that have hundreds of bands.

The overlap of imagery is also increasing. The same areas are being taken repeatedly by satellites thereby providing highly temporal data. In aerial photography it is standard to take imagery with a high overlap for stereo coverage.

Imagery is becoming more affordable and there are many datasets available for free. One of the key problems with geospatial imagery has been accessibility. Imagery is available, but often not accessible. There is plenty of imagery available, but only a small fraction of the imagery is actually accessed and used.

Finally, the needs of specific users might necessitate custom workflows. With traditional methods, this can lead to significant data redundancy and large stores of intermediate data products.

Here is where image services have come in and caused a paradigm shift in both, managing and serving imagery. Services are replacing the conventional idea of 'a file at a time' workflow, not completely displacing them though as there are times when users would require to perform analytics on single files. This paper discusses serving out large collections of imagery from multiple sources as different products (True color, false color) for visualization purposes and/or analytics. There is no apparent need to work with separate files in this case and image services would be apt.

2. Information model used

The Mosaic Dataset is the information model in ArcGIS that is used in this study to manage and serve imagery. Mosaic Datasets and Raster Datasets are served out as image services for various clients to consume through REST end points or as WMS/WCS services. The mosaic dataset serves as an image library, warehousing imagery, image metadata and processing information. The mosaic dataset is served through a web service (image service). Image services are more than just pictures or base maps; Image services provide access to imagery as an image for visual analysis, pixel analysis (DNs) and with access to the image catalog (collection with metadata). Image services serve out multiple products from the same source images and are non-destructive in nature.

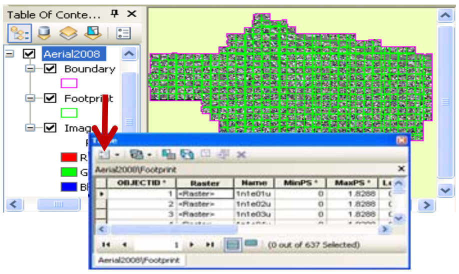

Figure 1. Mosaic datasets and raster datasets.

2.1 Empowering technology

The empowering technology driving image services are primarily dynamic mosaicking and On-The-fly processing.

- Dynamic mosaicking enables fusing imagery from multiple sources based on a user defined ordering. By default users see the best available imagery. (Figure 2.1.a)

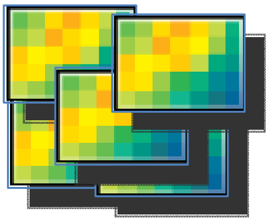

Figure 2.1.a. Dynamic mosaicking**.**

- On-the-fly processing capabilities enable processing imagery as its being accessed. It enables creation of multiple products from a single set of source files. This is made possible by way of Raster Functions. (Figure 2.1.b)

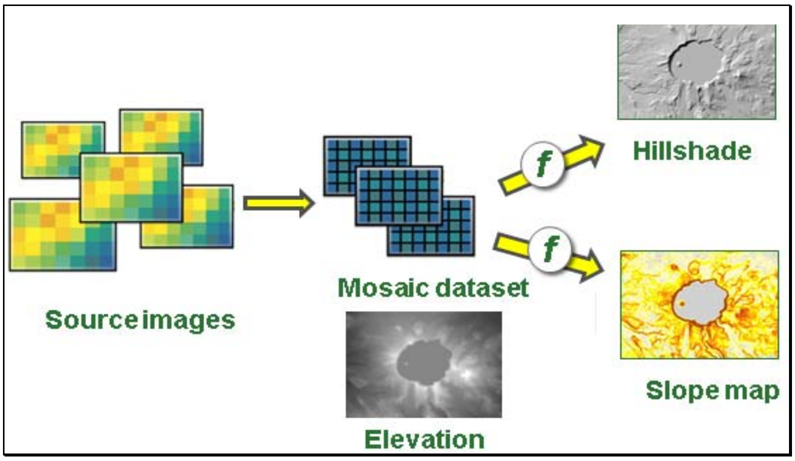

Figure 2.1.b. Single image service to multiple products.

Figure 2.1.b above indicates how a single source of elevation data can be served as a single image service, but can be consumed as multiple products:

1. Elevation - A product used for analytical purposes, which provides real pixel values (heights on ground)
2. Hillshade - A product purely for visualization purposes, depicting a Hillshade representation of the elevation data
3. Slope map – A product for visualization purposes depicting the slope of the input elevation source

3. Best practices when building image services

With the data management tools that are provided with ArcGIS, building image services is rather trivial. However, depending on the type of imagery and the targeted workflow, there are several best practices users can follow when creating their services. When building the examples for Virginia and New York State, several best practices evolved from the work done. This resulted in several recommendations as well for users working within this community or working with this volume/type of imagery.

3.1. Optimized image formats

An optimized image format to start with is critical to achieving a performant and scalable system. Some imagery can be slower to read than others due to their storage format or compression and it is recommended that these formats are converted into more optimal formats. ASCII DEM image format, Raw un-tiled TIFFs, wavelet based compression such as jp2k (generally more CPU intensive to decompress while providing only marginally better compression), jpeg formats are typically formats that are slow to access. It's recommended to store your imagery holdings as Tiled TIFF for optimized performance and faster disk access (Tiled). Using LZW, jpg compression vs. wavelet based jp2k compression (CPU intensive) is recommended for compression. When converting imagery isn't an option, building pyramids and/or overviews on the mosaic dataset is a viable option.

Compressed formats may be all that is available for older collections. For newer collections, keeping the original imagery data in lossless form enables richer options for on-the-fly processing and rich image products. Since the on-the-fly processing does not store intermediate results, the efficiency gained compensates for storing the original data as an online resource.

3.2. Prepping your data

Managing and publishing imagery using a mosaic dataset can save you time over traditional methods of mosaicking image collections together or producing multiple outputs; however, there are times when you want to consider some preprocessing. The recommended preprocessing applies to creating the fastest and best mosaicked imagery display. Building pyramids improves the display and processing performance of raster datasets. This needs to be done on the source Raster Dataset once. By doing this it improves the performance of the Mosaic Dataset. Statistics are required for a Raster Dataset or Mosaic Dataset to apply a contrast stretch, in classifying data, color balancing and other radiometric enhancements. Pyramids require additional disk space and are typically written to separate files with an .ovr extension. Their total size is smaller than the original image and typical usage shows that they get very high utility because of the way users navigate in modern online sessions. These files are written in the same location as the raster data.

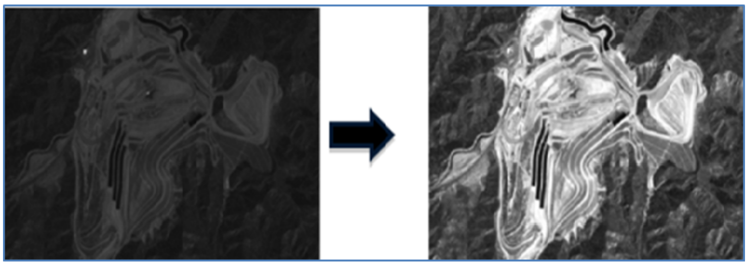

Figure 3.1. Applying a contrast stretch.

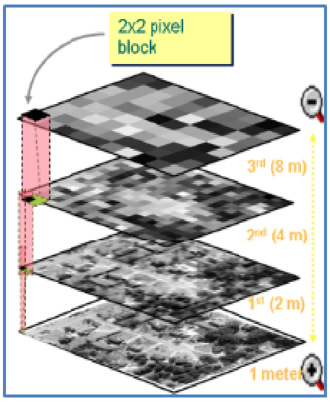

Figure 3.2. Raster pyramids.

3.3. File layout on disk

It's generally recommended to store your data on disk based on an intuitive attribute. With reference the imagery used in this case study, storing the imagery in subfolders based on sensors, and thereafter collection time range would be an appropriate approach. Structuring the storage of files on disk do not in any way impact performance of an image service 'consuming' those images. Structuring is merely done for file management purposes. When the data is structured well on disk, 'building' the mosaics are also a lot easier.

Similar to file layout on disk, good practices can be followed when building mosaics as well. Build separate mosaics for each sensor type. It's recommended to use raster function templates for varying processing options (instead of creating multiple mosaics for multiple products. This reduces the load on the server, and makes it easier for the end user as he would require connecting to a single service end point for multiple products as opposed to having to deal with multiple services for multiple products.

3.4. Raster functions (processing chains)

An Image Service is more than just a service that streams out image data. The processing time to serve the data can be affected by adding image processes (Raster functions). Raster functions vary in terms of how they affect performance of a service; therefore, depending on the processes that are added to an image service, the processing time may increase.

Generally, when adding radiometric processes to enhance the appearance of the imagery, such as a linear stretch, the processing time may be increased by 5 percent. If you apply a convolution filter process, the processing time may increase by 25 percent when using a 3 x 3 kernel or by 50 percent for a 5 x 5 kernel. When using an image fusion process, such as Grayscale, or an Band math function, such as NDVI, the processing time may increase by 10 percent. Additionally, if you apply a process to the individual raster's versus applying the process to the whole image service definition, the processing time could increase by 5 percent.

When using geometric processes (including reprojection), the processing time can be affected by the complexity of the transformation. Image Services do not transform each pixel; instead, it performs an analysis of the transformations that are required and breaks the requests into small tiles, where the deviation within a tile of a pixel is not to be more than half a pixel in the output. Most simple transformations require few tiles to achieve this accuracy and the amount of tiles is determined automatically. The processing times are also dependent on the complexity of the sensor model being used.

The sampling method used in the geometric transformation also affects the total processing time. A nearest neighbor sampling method tends to be 10 percent faster than the bilinear interpolation, whereas cubic convolution tends to be 20 percent slower. This is because for these transformations, each has to consider a different number of pixels surrounding each pixel, with cubic convolution factoring in the most and nearest neighbor the fewest.

Performance can be significantly improved by building processing/display caches (enables dynamic mosaicking, and pixel analysis – but no dynamic processing), caches (fast, used purely for visualization purposes), or overviews.

Although Caches, Item Caches and Overviews have the penalty of storage, they significantly reduce the on-demand processing budget.

3.5. Image service settings

Image services are optimized to work well with most use cases and types of images. However the system has exposed several flags and options to further optimize the performance of Image services. Flags that were modified in case of our example of the statewide imagery collection, include:

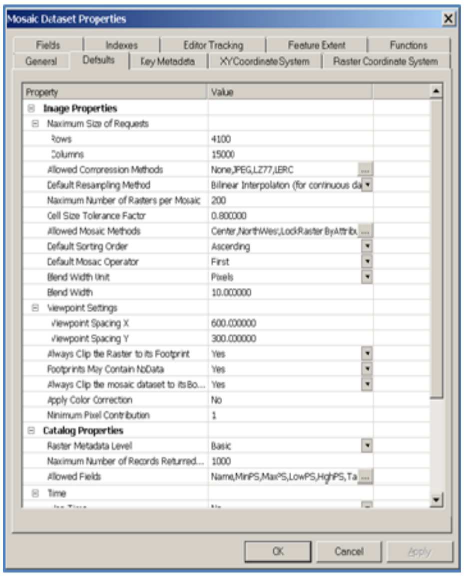

Figure 3.3. Image service settings.

'Default Compression' (when image is being transmitted) is set to jpeg. This significantly improves performance of the services on low bandwidth networks. The 'Mosaic Method' is altered to 'Closest to center' ensuring the number of raster's mosaicked were a minimum for each screen scrape. The clip to footprint flag is an optimized method to handle no data areas, as opposed to working with no data masks. Enabling the 'Footprints May Contain No Data' flag ensures the system assumes the images contain no transparent pixels and hence disables transparency checks on individual pixels in the image.

3.6. Serving optimizations

On advanced clients, a cached image service can be consumed as a dynamic image service and it can be consumed as a cached service which is purely for visualization purposes. This dual capability makes image services much more valuable as it empowers users to use the service for both – analytical and visualization purposes.

Server raster functions eliminate the users need to create multiple mosaic datasets for multiple products. For instance, in the Virginia and New York study, we've had to serve out a true color, false color and an NDVI product. Serving a single service with Server raster functions attached enables easier management and a lesser load on the server as well. A client can point to a single service end point and choose the product they would like to render/work with.

4. Conclusion

From what's been discussed and showcased with the Virginia state implementation, Image services will result in more efficient access to imagery. Services have created a paradigm shift for Imagery users eliminating the complexities involved with managing large volumes of imagery and collections of files.

For the image manager, dynamic image services can be updated rapidly, they are scalable, and provide rich functionality to the user community. For the user, this is imagery being processed on demand which is available as *original image data* or as a cached map for visualization purposes.