NEW GMLJP2 STANDARD

SOFTWARE ARCHITECT MICHAEL GERLEK OFFERS DETAILS ABOUT HOW SPATIAL REFERENCE COORDINATES AND GEOGRAPHICAL FEATURES CAN BE EXPRESSED AS GML DATA EMBEDDED IN JPEG 2000'S XML BLOCKS.

The XML based Geography Markup Language (GML) has already been approved as an official standard by the Open Geospatial Consortium (OGC) and endorsed by the National Geospatial Intelligence Agency (NGA) and other organizations. But GML may achieve its highest calling when used inside another standard, JPEG 2000.

You've probably heard about JPEG 2000 by now: "JP2" is an ISO-standard image format which for a few years now has been steadily gaining traction in a number of technically-demanding markets. A number of applications in our own GIS arena are already supporting it as an import and export format.

But, until recently, there was no standardized way to make JP2 files "spatially aware". While the file format has always been perfectly usable for storing the raster data, there was no formal, portable means to store geospatial metadata, such as the geographic extent of the image. In this article, we will introduce GMLJP2, a new standard from the Open Geospatial Consortium (OGC), which defines a means for embedding powerful GML data within a JP2 file.

JP2 – An Almost Perfect Solution

JPEG 2000 is an advanced image format which uses wavelet-based algorithms to achieve high-quality lossy compression, much as proprietary formats like MrSID have provided for years – for typical imagery, a file size reduction of 20:1 or even 30:1 will show little or no visual data loss, allowing massive geospatial datasets to be more efficiently stored and managed.

Additionally, JPEG 2000 can store an image losslessly, preserving full numerical accuracy, with a typical file size reduction of 2:1. High-quality compression is the principle reason large geospatial datasets are often stored with wavelet-based compression formats. JPEG 2000 offers several other features of interest to the GIS community, including: support for multi- or hyperspectral data, support for signed and unsigned data with more than 16 bits of precision, support for storing multiple images per physical file, access to multiple resolutions, and much more.

However, the JPEG 2000 committee did not design JP2 with any specific application domain in mind; they explicitly created a format that would support a wide range of image types, and then left room for metadata extensions to be added by various communities of interest.

A crude but useful metaphor would be that JPEG 2000 is a newly developed cargo container that doesn't require tractor-trailers or railcars but can move through hyperspace to its destination. The problem is, you and I are in the meat shipping industry and there is no standard means of refrigeration described in the spec for this new container. We want to use this new conveyance, but because it is indifferent to the nature of its cargo we're responsible for including our own cooling mechanism each time we use the container. The container's spec allocates space for such a mechanism, but we need the meat shipping industry to agree on what that mechanism should be.

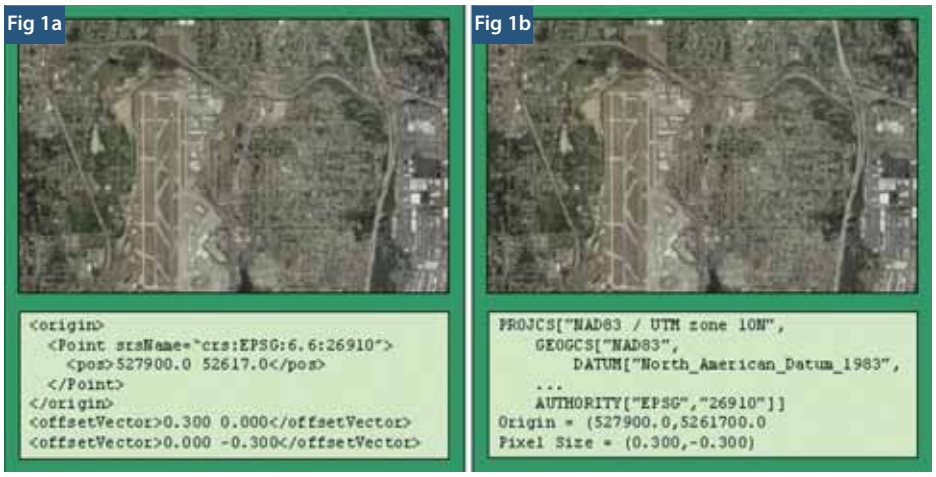

Fig. 1a - a JP2 file containing image data and GML georeferencing data
Fig. 1b - a GeoTIFF file, with the same information.

GML – The Rest of the Picture

Bringing this metaphor back to geospatial imagery, what was needed was not refrigeration but a mechanism for annotating JP2 files with geospatial information– akin to the well-known GeoTIFF extensions to the TIFF format. Fortunately, our community already has a rich mechanism for expressing geospatial information. GML – the Geography Markup Language – is an XML-based language and an OGC standard for describing the syntax and semantics of geospatial-specific primitives such as coordinate reference systems, features, geometries, coverages, and much more. It provides a common language (schema) for GIS systems to express needed information: metadata, annotations, and so on.

Under the aegis of the OGC, then, a number of vendors – led by LizardTech and Galdos – led an effort to define a scheme for using GML to encode arbitrary geospatial information within the JPEG 2000 format. By putting these two standards together, we can make JPEG 2000 spatially enabled.

The GMLJP2 Standard

The JPEG 2000 standard allows for XML data to be stored in a number of user-specific "boxes" within the file, alongside the compressed image data. The new GMLJP2 standard says what these boxes are to be named and what GML data they are to contain. In the simplest case, the GML will just describe the geographic extents of the image and the coordinate reference system used. And with the powerful expressiveness of GML, more complex annotations are possible. Let us consider a few examples common to many typical workflows.

Example 1: Including Coordinate Reference Systems

Figure 1a shows the simplest case: a JP2 file which contains the normal "image data", plus a box of "XML data" containing the GML-encoded coordinate information. Figure 1b shows how this is similar to the encoding information available with the GeoTIFF extensions to the TIFF standard.

Example 2: Integrating Other Useful Data

Now let's go beyond what we can do with GeoTIFF. Figure 2a shows an aerial image of Seattle, Washington, and Figure 2b shows the road network for the region. In most workflows, this data would have to be represented in two different files, in two different formats: perhaps a GeoTIFF and a Shapefile. Using GMLJP2, however, we can store the image data (with all the advantages JPEG 2000 has over GeoTIFF), add the geographic extent information as we did above, and then store the road features (with all the advantages GML has over Shapefiles); see Figure 2c.

Additional information might be added to the image by a later user. A specific building within the coverage might be called out as an area of interest, for example; GML can be used to locate the building and detail some observational data about it.

The advantage of having feature data and metadata within a single file is more than just a notational convenience: we now have a complete representation of the area and all its feature data that can be treated as a kind of self-contained database, suitable for distribution to third parties. Figure 2d shows a model of the JP2 file contents.

FIG.2A - a GeoTIFF image of Seattle
FIG 2B - a Shapefile of the road network
FIG 2C - a JP2 file containing both imagery and features
FIG 2D - a model of the JP2 file contents

Example 3: Providing a Junction for Distribution

GML is the common language used within the OGC web services ecosystem, such as for WFS data. GMLJP2 provides a ready means for image and feature data to be combined as a single output type for some downstream user. In Figure 3, we show how the above Seattle example might be played out using these web services for a utility company field engineer: a WFS and a WCS serve up the feature and raster data, respectively, for a given region, and this data is then presented as a GMLJP2 file for later offline consumption by a custom application on a hand-held device.

More to Come

Those are just three uses of GMLJP2. When you consider the full power of JPEG 2000 – such as its ability to encode multiple images per file, or its support for progressive transmission across a network – more use cases are possible: consider georeferencing of mosaicked tiles, correlation of stereo-pairs, or a customizable description of the sensor model used to collect the imagery.

The broad range of potential uses is why LizardTech products now support GMLJP2 starting with the release of GeoExpress 6.1. JPEG 2000 by itself is a powerful image format, but it lacks the geographic self-describing characteristics required to make it truly a useful tool for GIS. By taking advantage of the expressive and standardized power of GML, we can add those characteristics and much more.

Michael P. Gerlek *manages LizardTech's Engineering department and helps lead its efforts to promote open standards development in the geospatial arena and is a charter member of the Open Source Geospatial Foundation. He can be reached at mpg@lizardtech.com.*

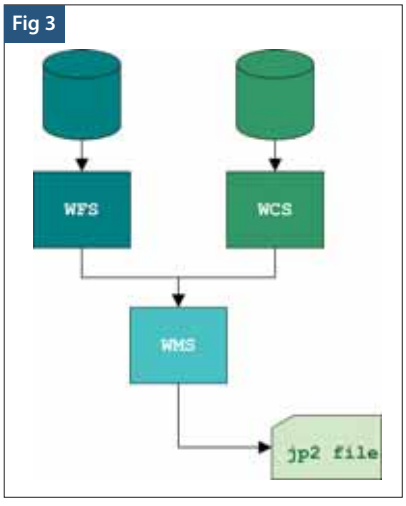

Fig. 3 a WCS and a WFS feeding a WMS, which returns GMLJP2 data to a client within a JP2 file.