Titanbrowse: enabling access to hyperspectral remote sensing observations

Paulo Penteado[1*], David Trilling[1], Jason Barnes[2], Virginia Pasek[3]
[1]Northern Arizona University, [2]University of Idaho, [3]Free Radical Consulting
*pp.penteado@gmail.com, http://www.ppenteado.net

Introduction

There are many tools and standards that allow immediate retrieval of a variety of astronomical observations, mostly derived from the VO paradigm, but these are not applicable to remote sensing observations of Solar System bodies. While astronomical observations are confined to a 2D spatial domain (coordinates in the sky), with all objects always observed from essentially the same point of view, remote sensing (i.e., spacecraft) observations span different perspectives. More elaborate tools are needed, both to evaluate the geometrical conditions of the observations, and to archive them in an accessible way.

While NASA's Planetary Data System (PDS) provides access to all the public data from current and past missions, it focuses on preserving the data, in a separate archive for each instrument. This enormously rich dataset, which now contains orders of magnitude more images than could possibly be inspected by humans, can become much more useful to researchers with the addition of the higher level data products and tools proposed here. Currently, it is difficult to perform exploratory analysis on large data volumes, and to directly combine data from different instruments, given that each dataset is organized differently. Existing online tools for archive access and visualization lack some of the features needed to make more extensive use of the data, particularly of hyperspectral imagers.

Titanbrowse is a system to catalog archived Solar System remote sensing hyperspectral imaging, originally developed for Cassini VIMS observations of Titan. It provides a single interface and standard to access, visualize, explore and process observations for multiple targets, taken from multiple instruments. This level of data products can be seen as the hyperspectral imaging equivalent of the current Virtual Observatory (VO) services for astronomical images, while superseding VO capabilities, with tools to allow complex interactive queries, based on arbitrary, dynamically-defined functions to evaluate the selection criteria, assisted by interactive visualization, while also providing automated retrieval of query results. Other innovations of our system are the use of precise spherical geometry to determine the boundaries of each pixel, the inclusion of those pixels that do not intercept the surface, the possibility of and queries based on the data (not only metadata), at the level of individual spectra (as opposed to whole cubes).

Data and calibration

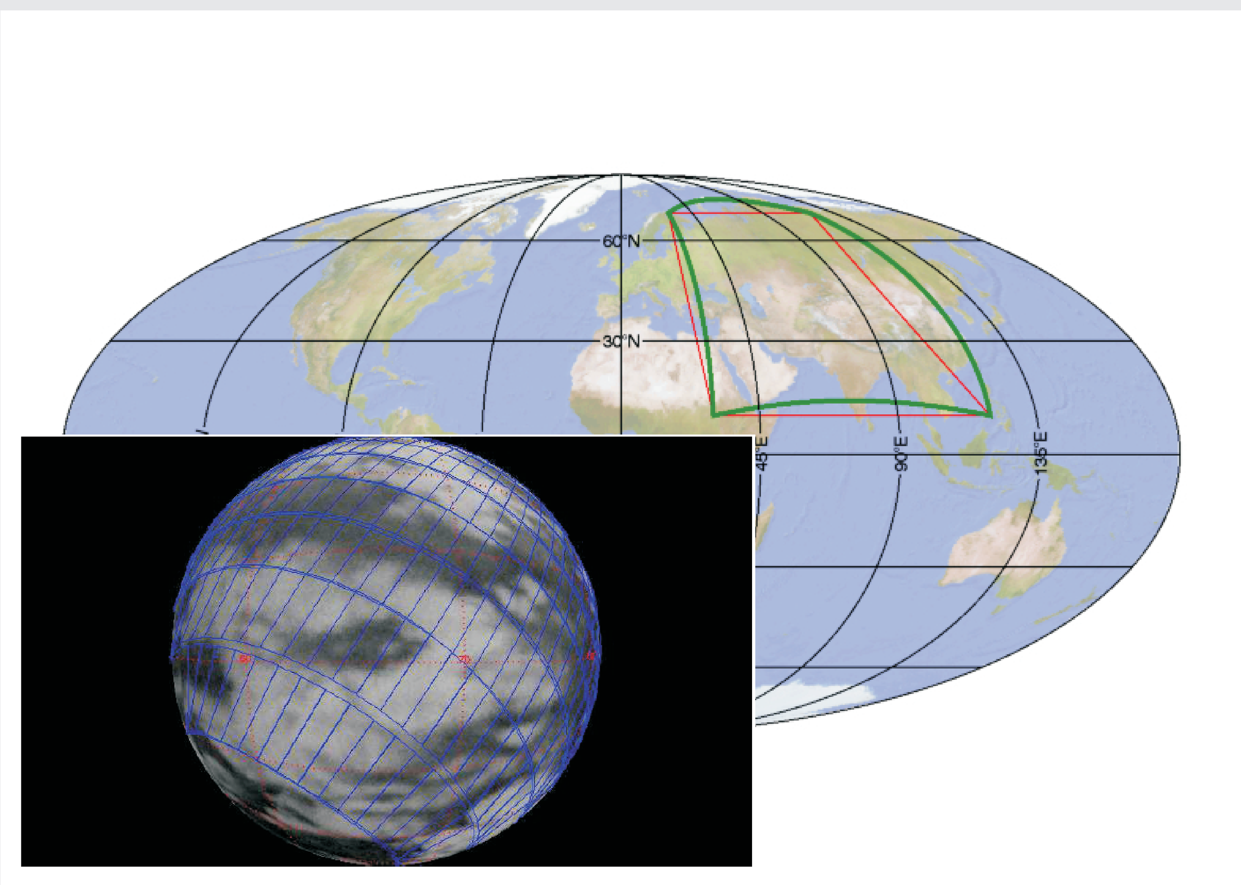
The raw data used is publicly available at NASA's Planetary Data System (PDS) Imaging Node (http://pds-imaging.jpl.nasa.gov/index.html).

The cubes retrieved from PDS need radiometric and geometric calibration to be ingested into our database. For this purpose, a custom pipeline was developed, making use of the default pipeline software (included in the PDS datasets), and the data made available by NASA's Navigation and Ancillary Information Facility (NAIF, http://naif.jpl.nasa.gov/naif/index.html), incorporated through the use of NAIF's SPICE library. The pipeline developed produces cubes with more geometric data than the standard pipeline, including:

- The geometric data is calculated for the center and each corner of every pixel, necessary to determine the projection of every pixel on Titan. This is of particular relevance for cubes taken at low spatial resolution, cubes with a small length in one dimension (even 1 pixel wide cubes), and cubes with large pointing changes between pixels.
- The geometric data is calculated even for pixels that do not intercept the surface, in which case they refer to the point in the line of sight nearest to the surface. This is essential for studies of Titan's atmosphere, with spectra taken over Titan's limb, which are unique as they provide direct vertical resolution of the atmosphere.
- More illumination data is calculated, including the location of the specular reflection point, and its distance from each pixel. This data is of particular relevance for detection of liquid surfaces, and studies of the surface's scattering function. These data are calculated for each pixel, to account for cubes where a large geometry change occurs between the time each pixel was recorded.

The geometric data are incorporated into the cubes into their header (for data which are constant over the cube), and into backplanes (for data which vary for each pixel). The current implementation produces 52 backplanes.

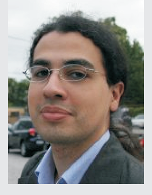
At this time, we have created pipelines to process and ingest data stored in ISIS cube format (used by Cassini VIMS) and FITS files (used by New Horizons LEISA).

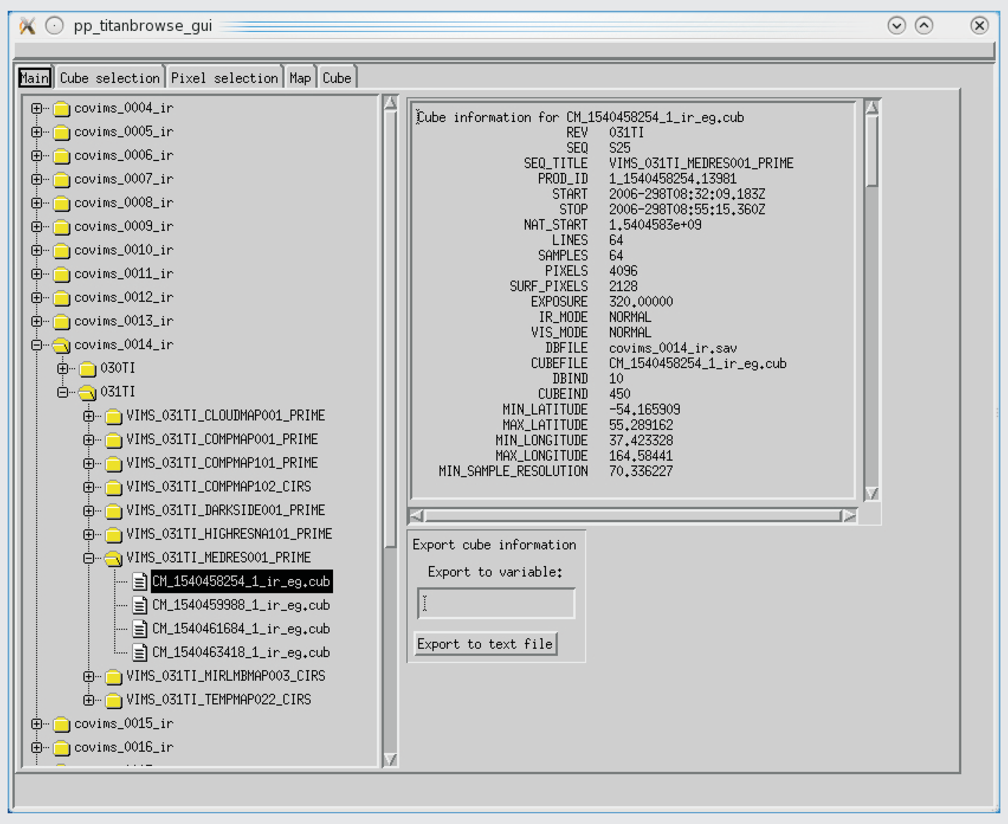


Background: The spherical polygon drawing algorithm we developed (green), compared to simply connecting a pixel's vertices by straight lines (red). Foreground: The outlines of several pixels mapped on Titan's surface, showing their extent and the gaps between them.

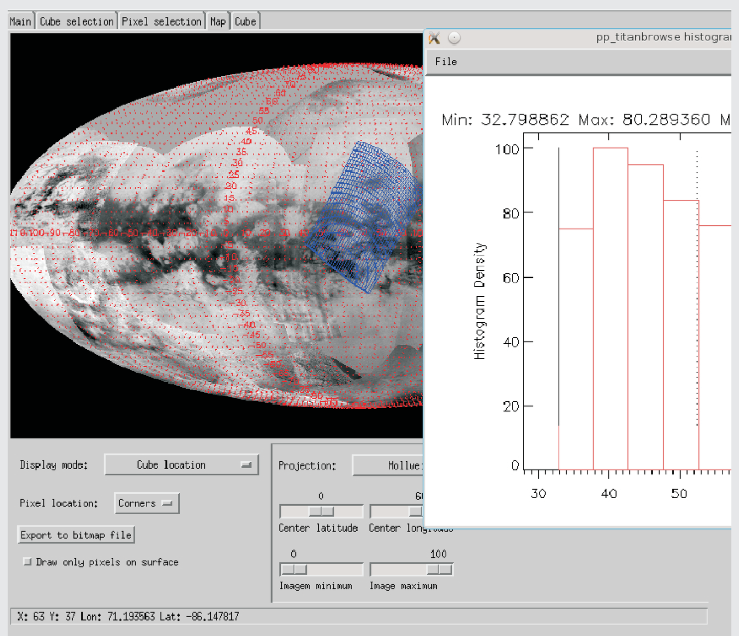
This work is supported by the Northern Arizona University Office of the Vice President for Research.

Poster available at
http://www.ppenteado.net/ast/pp_napsa2015_1.pdf

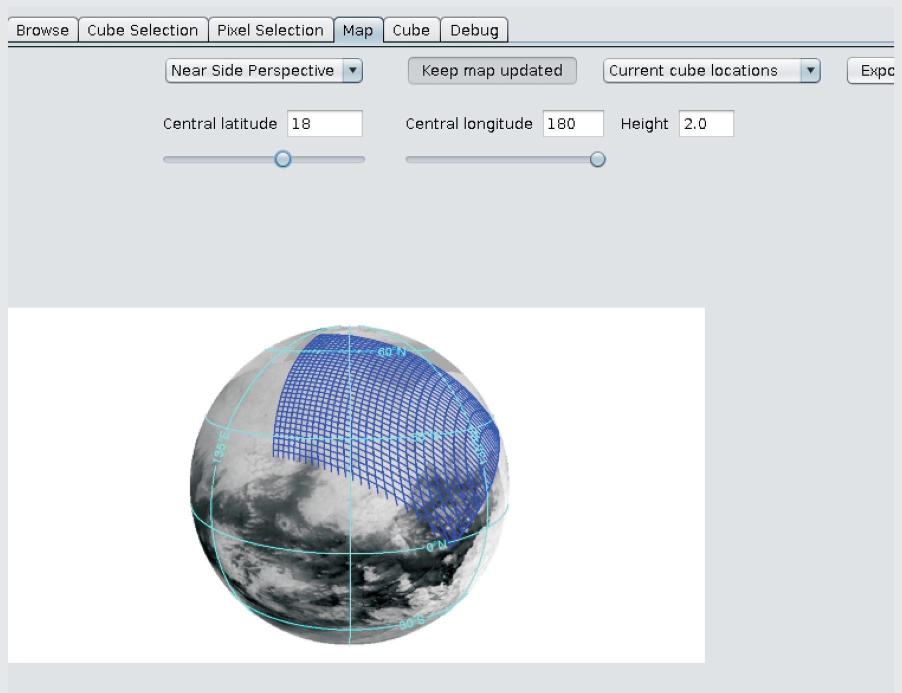




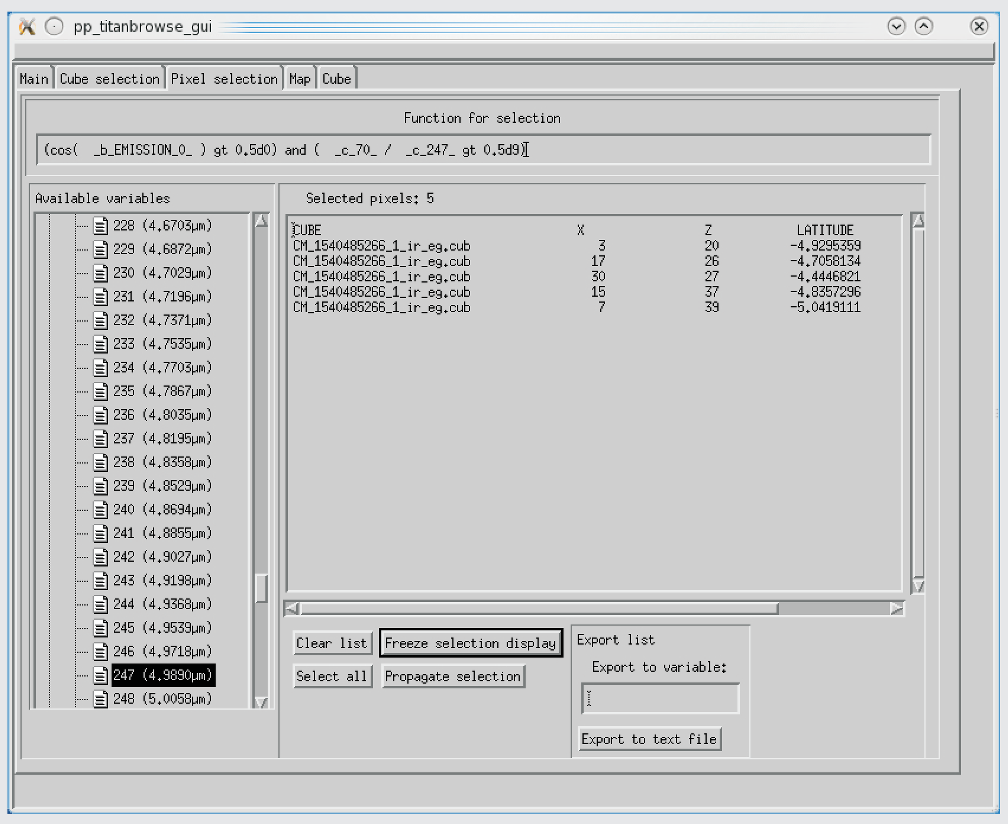
The cube browser panel of the graphical interface



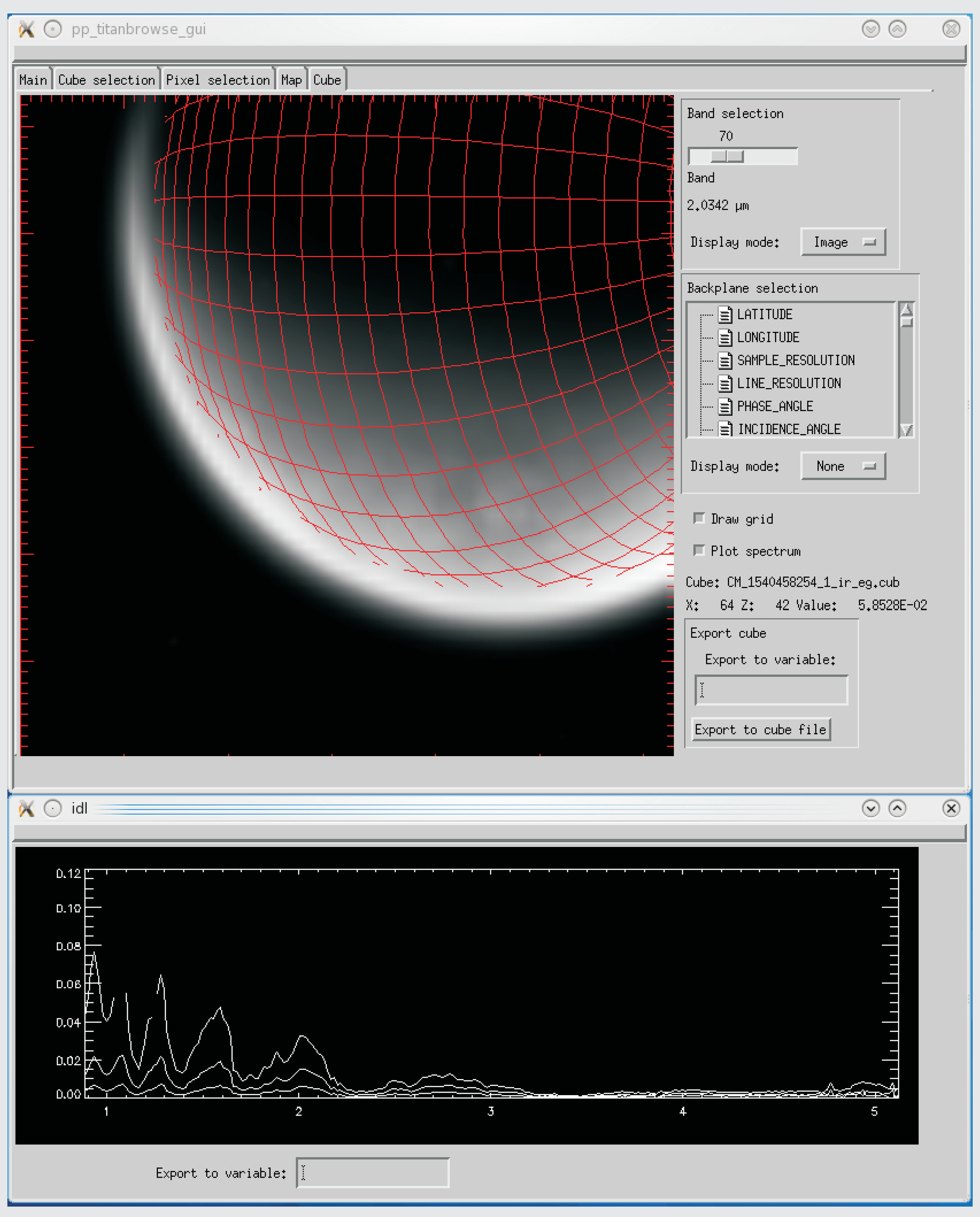
The map panel of the graphical interface



The map panel of the Java client



The pixel selection panel of the graphical interface



The cube visualization panel of the graphical interface

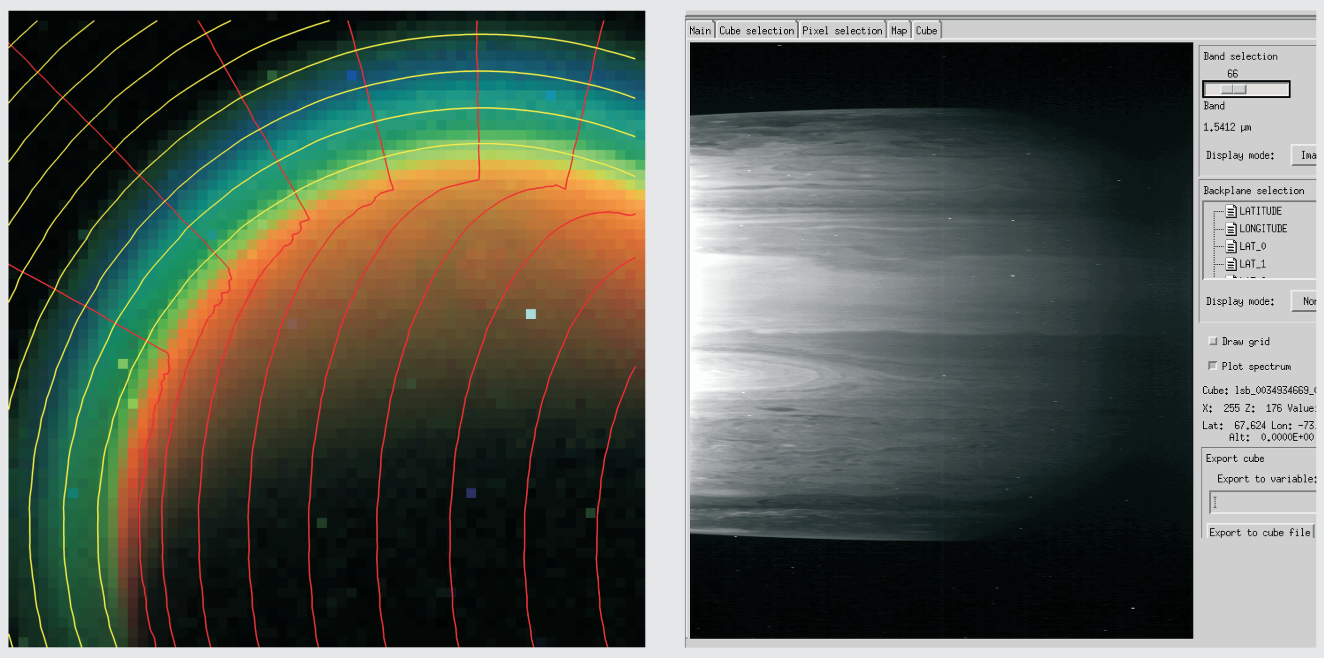
Database implementation and functionality

The database was implemented in IDL, instead of a dedicated database system, for several reasons:

- The data structures and the usual criteria for its access and selection are very array-oriented, and IDL provides ample support for efficient array processing.
- Dynamic interpretation of complex functions to be used for data selection and visualization, making use of IDL's standard library and user-defined functions that may be the same used in the data analysis.
- Integration of the database use with an environment commonly used for the subsequent data analysis. For instance, selected metadata, spectra and entire cubes retrieved from the database can be immediately used for analysis, with no need for separate data export/import through files.
- Integration between database access and visualization of the results, making use of the variety of visualization tools provided by IDL, including map projection functionality.
- Platform independency.
- Ease of development and maintenance.

The database works on the principle of making selections of entire cubes or individual spectra (spatial pixels). Starting from a selection that comprises the entire domain, arbitrary user functions of the cube metadata (for cube selection) or pixel data (for pixel selection) are used to filter the selections, reducing them to the data with the selected properties. The data that can be used in functions for pixel selection include all core bands, and all (currently 52) backplanes. The data that can be used for cube selection includes the ranges of the core and backplane values, and cube metadata.

Internally, the database objects contain all the processed cubes, stored as a collection of objects of a class called pp_editablecube. This class provides a convenient way to read, edit, write and store ISIS cubes or FITS cubes, allowing access (and modifications) to the entire contents of a cube (header, core bands, backplanes, sideplanes and bottomplanes) through an interface more complete and easier to use than any other routines that could be found. Cubes can be kept stored directly by these objects, and ISIS (.cub) or FITS files can be recreated with a simple call to one of their methods.
For each PDS dataset, the database keeps its data in two files: one contains cube metadata, and all cubes as pp_editablecubes, thus in a "cube-major" order. For more efficient access during pixel selections, the second file used by the database contains the cube core and backplanes in a "band-major" order, so that only the needed bands get read from disk. This allows the database to handle every VIMS titan cube recorded, and conveniently provide query and access to them, without the need to have the original cube files.



Left: An example of 3 bands of a cube mapped into RGB space, with contours of constant latitude (red) and altitude (yellow) for the pixel centers, to illustrate the geometric data included with the cube, which extends to the pixels that do not intercept the surface. Right: an example of a New Horizons LEISA cube of Jupiter in titanbrowse.

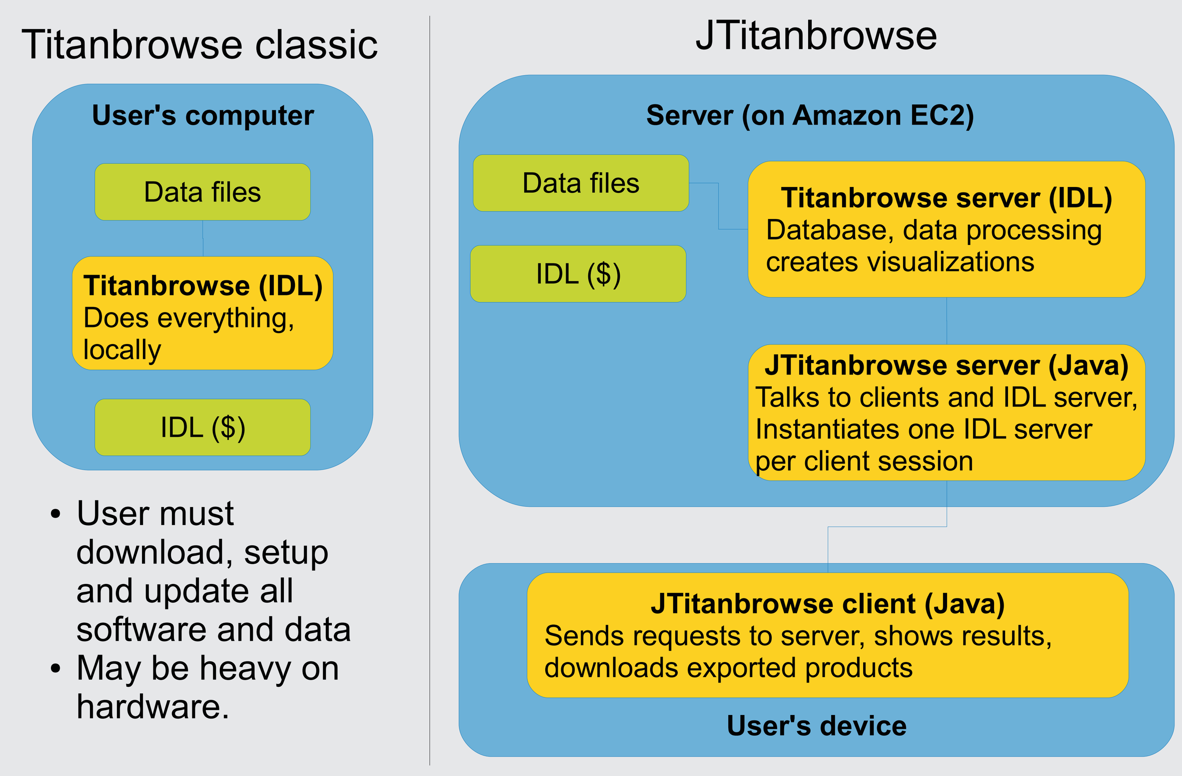
Graphical interface and visualization

Though the entire database was implement to be accessed programmatically (API), to aid in selection and visualization, titanbrowse includes integrated graphical interfaces (GUIs). In addition to the functionality provided by the API, the GUIs provide visualization of individual cubes and spectra, and geographical mapping of the selected pixels, to aid in interactive exploration of the data.

There are currently two implementations:

1) The IDL titanbrowse, where both the database and its access are run in the same IDL process. This provides easy integration with IDL for programmatic access and processing the results exported from the database, but requires the user to have an IDL license and download and setup the software and database files.

2) The Java client, jtitanbrowse, which is a lightweight client that runs on the user's computer, while the database and its processing reside on a server, which we maintain. This enables the user to access the system without the need to download and setup software or data.



The jtitanbrowse prototype under development can be accessed at http://www.ppenteado.net/titanbrowse.