Efficient databases for hyperspectral observations by array processing

Paulo Penteado[1]

[1]Instituto de Astronomia, Geofísica e Ciências Atmosféricas, Universidade de São Paulo (IAG/USP)
pp.penteado@gmail.com

Hyperspectral imaging, where datacubes are made of series of images at different wavelengths with contiguous spectral coverage, is becoming increasingly important in astronomical datasets. Spectral imagers have become standard complement in Solar System exploration missions, such as Cassini, Mars Reconnaissance Orbiter, Venus Express, New Horizons and Dawn, where they are often the prime instruments to determine the composition of planetary surfaces and atmospheres. Past, current and future ground-based sky surveys, including Sloan, J-PAS and LSST, all cover large areas of the sky, with spectral resolution ranging from a few to dozens of filters.

The large data volume generated by hyperspectral imagers is causing increasing difficulties to manage with traditional cube databases. While most researchers have focused on the difficulties arising from the large data read times (with data ranging from GB to PB in scale), in this work we show the importance of using array processing, instead of the traditional use of cube-only databases through sets of tuples. Here, through one example we developed, we show how such hyperspectral data need, in order to be useful: 1) a lot of table columns, since only the typical cube metadata is insufficient to make useful selection criteria, which require all the observed data, plus a lot of ancillary information; 2) databases of individual spatial pixels (instead of whole cubes), since the meaningful unit of selection is a spectrum, not a whole cube; 3) vertical partitioning, since typical selections make use of entire columns (of spatial pixels), but only a few of the columns.

While these needs can be somewhat handled by the recently developed SciDB, we show that more is needed: to identify subtle features of interest in the data, researchers need fast evaluation of user defined functions and these functions must be dynamically defined (instead of previously compiled); query results of such data also need integrated, fast visualization, so that the process can be interactive. SciDB is limited in these respects because user-defined functions need to be written in C++ and previously compiled into a library, query results must be exported to some visualization tool, and there is no large standard library for array processing of common scientific operations (integrals, differentials, transforms, cartography, interpolation, etc.). These 3 shortcomings prevent the fast interactive experimentation that is needed to explore complex datasets.

We show how these requirements for a hyperspectral imaging database were fulfilled with titanbrowse, which we developed for Cassini VIMS observations of Titan, and discuss how these can apply to other datasets. Titanbrowse was implemented

entirely in IDL (Interactive Data Language), as it provides efficient processing and expressive semantics of multidimensional arrays, interpreted code, a large standard library including scientific functions with array processing and visualization, and is platform-independent. Python, with several non-standard libraries (most notably NumPy, SciPy and matplotlib) would be a somewhat similar, though in some respects more limited, alternative to IDL in the implementation.

One recent example of results made possible by titanbrowse is the detection of the first tropical lake on Titan. We found this lake by exploring the observations with titanbrowse, to identify among the ~20 million spectra in the database those that indicated a liquid surface. Some of these data had been publicly available for several years, and despite the intense community interest in locating lakes on Titan, that lake had never been identified before we applied titanbrowse to this problem (Griffith et al., 2012; doi:10.1038/nature11165).