Access to 3-D/N-D Image Data in VO (DRAFT)

D. Tody, February 13, 2006

Discussions of how to handle regularly sampled 3D data have been ongoing. At this stage most of this has been offline in smaller discussions, mainly with projects having relevant 3D data which they wish to publish to the VO. This has developed into a very interesting and productive investigation with important implications for VO data access in general, not just cube data, due mainly to the sophisticated data analysis use cases involved. The intent of this posting is to summarize these discussions and bring everyone up to date, while inviting broader comment.

Our focus here is on uniformly sampled 3D (or N-D) data: that is, data where the samples can be represented as an N-dimensional array with an associated world coordinate system (WCS). This is the general N-dimensional "image" model used within astronomy for many years, and represented by the Simple Image Access interface within VO. We do not address here other forms of pseudo-3D data such as MOS or IFU data consisting of an set of extracted 1D spectra sampling a 2D image plane, although a similar approach is probably appropriate for this data as well. Also, while our focus here is on direct access to the data, this is not the only way to approach the problem of analysis of remote data. Another approach is to write a fully distributed application, with data intensive components of the application running remotely on the data server. That is also an important problem but is best addressed separately.

S. Gibson (Arecibo), M. Kissler-Patig (SINFONI), R. Taylor (CGPS), M. Marquarding (SGPS), F. Bonnarel, M. Dolensky, A. Richards, A. Rots, R. Williams, and others have thus far contributed to these discussions.

This is a fairly lengthy analysis. The topic is complex enough that it should probably be published as an IVOA Note, however it would be good to have some further discussion here beforehand.

Use Cases

At this point we have looked at the following as use-case examples of projects supplying pixelated 3D data which we want to be able to access and analyze with VO:

```
GALFA   Galactic HI/continuum survey, Arecibo L-band feed array
CGPS    Canadian galactic plane survey (DRAO)
SGPS    Southern galactic plane survey (Parkes/ATCA)
SINFONI ESO near-IR integral field spectrograph
```

The first three are all galactic HI radio surveys, producing 3D velocity cubes and related data products such as continuum images. SINFONI provides an example of an IFU spectroscopic imager which can generate true 3D image cubes based on multiband image slicing techniques. GALFA is a new survey just getting underway, while the others already have data products online or otherwise available. As noted in earlier emails, some work has already been done on publishing data from CGPS and SGPS to VO.

Some more details on the individual projects follow (this information was generally difficult and time consuming to obtain online - a good illustration of why we need to get all this in VO).

GALFA

GALFA consists of a Galactic 21cm HI emission line survey (GALFA-HI) plus a 1420 MHz full-Stokes continuum survey (GALFA-CON or GALFACTS) covering the entire sky visible from Arecibo (declination -1.5 to +38) with a spatial resolution of 1.8 arcmin and a spectral (velocity) resolution of 0.184 km/s.

The details of the data products to be produced are not yet clear, but a GALFA-HI cube might be 512 x 512 pixels square, with 7676 velocity channels, for about 3.8 GB per cube. The GALFACTS cubes have only around 1000 frequency channels, but they are full Stokes observations (polarization measures for each of IQUV) hence are comparable in size. Since Arecibo is a transit telescope the data is most naturally expressed in equatorial coordinates, although analysis will often be most naturally performed given a projection in galactic coordinates. Various standard 2D projections (HI column density, Tmax, etc.) will be associated with the 3D data.

CGPS

CGPS is a high-resolution (1 arcmin, 1.3 km/s) survey of atomic hydrogen and radio continuum emission covering 657 square degrees of the glactic plane. The CGPS HI cubes are 1024 x 1024 with 272 velocity channels, for a total of 544 MB per cube. Stokes I, Q, U and V continuum polarization images at 1420 MHz, and a Stokes I continuum image at 408 MHz are also available as is a CO cube. Complementary images in the four IRAS infrared bands are also available. A viewer based on the current SIA with extensions and Aladin is available; see http://www1.cadc-ccda.hia-iha.nrc-cnrc.gc.ca/cgps/navigate.html

SGPS

SGPS is a high resolution HI plus continuum survey of the southern regions of the galactic plane, covering about 210 square degrees, although partial coverage is provided for a larger region. The SGPS cubes are (apparently) about 900 x 260 with 400 velocity channels, for about 375 MB per cube. The spatial resolution is about 2 arcmin and the spectral resolution is 0.8 km/s. Data is available both with continuum and with continuum subtracted. Data access is based on dynamic extraction of sub-cube cutouts which are returned as FITS files. A Web-based visualization tool (RVS) is also available, as well as provision for viewing channel maps in a browser. The related wide-area survey HIPASS also includes lower resolution cube data.

SINFONI

SINFONI is a near-infrared (1 - 2.5 um) integral field spectrograph fed by an adaptive optics module. The spectrograph operates with 4 gratings (J, H, K, H+K) providing a spectral resolution around 2000, 3000, 4000 in J, H, K, respectively, and 1500 in H+K. The spatial resolution is selectable from 0.25”,

0.1” to 0.025” per image slice, which corresponds to a field-of-view of 8x8 arcsec, 3x3 arcsec, or 0.8x0.8 arcsec respectively. Since this is an image slicing IFU true 3D cubes are produced, although due to the multi-band nature of the instrument the spectral axis contains gaps where there is no coverage, and the spectral resolution varies for the different bands (but is uniform within a band). Current cubes are about 65 MB each, and the IFU-oriented Euro-3D format (E3D), which is based on FITS binary table, is used as the native format.

Other similar data collections exist; however this set is representative of current (uniformly sampled) 3D data and is probably sufficient for initial use-case analysis purposes. We did not try to do an exhaustive survey. The most important thing missing is data which has a time axis such as a time cube.

Access modes

Reviewing our use-cases we can make a number of general observations. Data cubes can be very large, requiring dynamic subsetting capabilities, or other forms of server-side computation, for remote data access to be practical. While 2D visualization techniques are useful, 3D data often requires true 3D data analysis. With complex data such as 3D cubes which often have associated preview, continuum, etc., images it becomes increasingly important to be able to view the data in various ways. Visualization software is complex and it is important to be able to interface to existing tools, most of which are FITS-based. Many of the techniques used to access 3D data are useful for 2D data as well. While we talk mainly about 3D data here, these discussions apply equally well to general N-dimensional data, although in practice data with more than 3 dimensions is rarely seen (ignoring the case of degenerate axes).

The following access modes are required:

- **Whole image**. This is the simplest access mode but it can be impractical for very large datasets. It is important for access to smaller datasets, and for applications which need to track actual physical datasets in archives, e.g., to index data or to track replicas. Whole image access normally returns native format data. Even for whole-image access to native format data, query metadata is still mediated.
- **Spectrum extraction**. In the case of a spectral data cube, a 1D spectrum is extracted along the spectral axis, through a synthetic aperture the location and size of which is defined by the client. The details of how the extracted spectrum is computed can be data dependent and are determined by the service. For time series data the comparable operation would extract a time series from a time cube.
- **Cutout 2D planes**. A cutout does not resample the data; the original data samples (e.g., pixels) are returned. The image axes are not changed. For many types of analysis resampling degrades the data more than is acceptable and a simple cutout is what is desired. A cutout is also relatively simple and efficient to generate. In the case of 3D data, a 2D cutout can be made along any 2 of the 3 axes, although extraction of a full XY plane (Z constant) is probably the most common case. Of course, the entire 2D plane does not have to be retrieved; a subset of a 2D plane can be cutout if desired, as for a 2D image. This is the primary access mode for the current CGPS/Aladin interface.

- **Cutout 3D sub-cube**. This is similar to a 2D cutout except that a 3D sub-cube is extracted. This is a common case for true 3D analysis of cube data where the individual cubes are too large to be practical to retrieve without subsetting. This the primary access interface for the current SGPS survey.
- **2D projection**. In this case a 3D sub-cube is extracted, but the cube is collapsed along one axis to produce a 2D image. This is a form of projection since the pixels are resampled along one axis: collapsing an axis in effect changes the sampling on that axis to cause the sampling bin width to encompass the full region which is collapsed. The effect is similar to looking through the cube face-on along the selected axis.
- **3D projection**. This is the generalization of the 3D cutout. This case produces a 3D sub-cube like the 3D cutout, but the pixels are resampled in 3D. The axes of the extracted sub-cube do not have to be aligned with the original cube, axes can be transposed or flipped, the sampling can be changed along any axis, any axis projections can be changed (such as a 2D sky projection or a velocity axis scaling), and any coordinate systems (e.g., equatorial vs galactic) can be modified as desired. While this sounds complicated, all it really means is that the client is allowed to specify the WCS of the 3D sub-cube to be generated. It is directly analogous to reprojection of a 2D image.
- **General 2D slice through a 3D cube**. This is the generalization of the 2D cutout, and is equivalent to the 3D projection except that a 2D image is produced. The reference point of the 2D image can be anywhere in the 3D space of the cube, and the orientation of the 2D slice can be arbitrary, hence this case represents a general 2D slice through the cube. For example, one might slice the cube along the major axis of a galaxy as determined from a 2D continuum image, with the Y axis of the extracted 2D slice representing velocity. The analogous case for a 2D image is an arbitrary 2D slice which produces a plot of flux versus 2D position. This access mode occurs commonly in 3D cube visualization.

Clearly these access modes are mostly variations on the same thing. Ignoring spectral extraction and the trivial case of whole-file access there are only two fundamental types of access: cutout or projection. Within these two areas the access modes differ only in the dimensionality of the image subset produced. In the most general case precision data access can be obtained by specifying the WCS and image geometry of the data to be generated.

A secondary issue with how the output image is generated is whether any mosaicing of individual datasets takes place to produce the generated output data. In principle mosaicing techniques can be used both for cutouts and for projections, although mosaicing cutouts from several image tiles to produce a larger image is possible only if the the input images have been intentionally generated to have the same projection, alignment and scaling (for the observable, e.g., flux or velocity, as well as for the sampled axes).

Strategy

The problem of understanding and accessing complex remote 3D data breaks

down into two parts:

- Data and service discovery and description, including sufficient metadata to associate related data products, detailed metadata describing each data product, and the services available to access the data, so that data access can be planned.
- Data access, where virtual data is generated by the remote service and retrieved by the client for local analysis. In the case of large complex datasets such as survey data cubes, repeated access requests may be required to access different data regions or different "views" of the data.

In simple cases, data discovery and access for image data can be streamlined into a single SIA query, often posed to multiple data services, followed by retrieval of selected datasets. This is the way most VO data access works now. We pose a simple POS,SIZE query for the region of interest, possibly with additional query constraints, find all relevant data, and select the most appropriate images for immediate retrieval and analysis. For this type of analysis we usually don't care too much about the details of where the data comes from and how a subset is generated, so long as the data is well described and characterized, and well suited to the type of analysis being performed.

While this works fine for direct region-based multiwavelength analysis, in the case of our 3D use-cases there are two problems with this approach:

- Analysis of large 3D cubes may require precision data access where the client makes a series of requests to access regions of a single dataset, for example extracting successive 2D planes or slices from a cube. This requires a 2-step approach where detailed metadata is first retrieved and this knowledge is then used to directly access regions of a particular dataset.
- There is no easy way with the current SIA to describe complex data consisting of a number of related data products, e.g., a cube and several 2D projections of the same field, possibly with an associated detection catalog for the same field. Using a logical name to associate related data elements helps, but a higher level approach is needed to describe how the different data elements are related. Worse yet, if different service types are available to access the same data, e.g., SIA for image access and SSA for spectral extraction, it becomes difficult to discover all the available ways to access a dataset - one would have to make separate SIA and SSA queries and try to determine by inference if they provide different views of the same data. A more direct approach is needed.

A promising approach to address this problem is to move forward with the ***generic dataset*** DAL interface. The concept for this has been around for some time, but until now we never had a use-case which needed it so it has not yet been implemented. What we want to do is make more of a distinction between general data discovery and actual data access which can subset, filter, or otherwise transform data to generate virtual data.

The existing DAL interfaces such as SIA and SSA are oriented towards generation of virtual data on the fly, hence are already well suited for precise data access. SIA will need to be generalized to 3D to support cube access and to provide additional query parameters and updated dataset metadata, but the "image generation parameters" already

present in SIA 1.0 already provide a basic capability for precision data access since the WCS and image geometry of the generated image can be specified by the client.

Generic Dataset Discovery

Most of the DAL interfaces target a specific type of data object: catalog, image, spectrum, time series, and so forth. Each has a standard data model and query interface specific to the type of data being accessed. This approach makes it possible to implement advanced capabilities such as metadata mediation and virtual data generation.

If we look at the class hierarchy for the DAL interfaces there is a class called Dataset at the root:

Dataset

Catalog
Image
Spectrum
TimeSeries
. . .

The Dataset class is the generic (typeless) dataset. All data in DAL is derived from this class. A Dataset object is described by the generic dataset metadata we have been developing over the past year or more, a version of which is present in the current SSA interface. This generic metadata includes things like dataset identification, characterization, provenance, and so forth, which are general enough to apply to any dataset regardless of its type or dimensionality.

What Dataset lacks is a specific object data model and access methods which can generate virtual data - these can be defined only if the type of data is known. Hence, Dataset deals with physical datasets rather than dynamically generated virtual data. Access is limited to simple whole-file access, but since the interface can find any type of data it is good for general data discovery.

A query interface for Dataset has not yet been defined, but would be a somewhat simplified version of other DAL interfaces. Since Dataset is not specific to images, positional queries would be by search radius as for cone search and SSA. Additional query parameters based on the generic dataset metadata would be supported.

The output of the Dataset query would be a list of matching datasets, much as for other DAL interfaces. However, any type of dataset could be described and uniformly characterized by the query response, e.g., images, 1D spectra or SEDs, or object catalogs (the catalog “datasets” not the individual catalog rows) for example corresponding to an image found by the query. The access reference, if direct access is provided, would support whole file data retrieval in the native file format in which the data is stored, however access would normally be a separate step, using SIA/SSA, cone search, or whatever.

An open question is whether, if an observation consists of multiple data elements (e.g., a cube plus associated 2D projections), should the query response describe the aggregation or the individual data elements? Since the individual data elements will be directly

accessible, probably they should be described individually, using some means to associate the individual elements in the query response. This solves part of our problem, by providing a way to associate related data elements (datasets) while still providing fairly detailed metadata for each.

Finally the Dataset query needs to tell the client what forms of access are available for the listed datasets. Exactly how this would be done is not yet clear, but probably it would be a second table within the same query response. Hence the query response would have a table of (possibly associated) datasets, plus a table of access methods (DAL services) for accessing those datasets. A data cube for example, might have a 2D or 3D "cutout" access method, a 2D or 3D projection access method, and a spectrum extraction access method, all for the same dataset. These would all be described in the access methods table, with each dataset record pointing to zero or more accessors.

Given information on these access methods, i.e., pointers to the services available to access the data, the client would then use an existing SIA, SSA, cone search, SkyNode, etc., service for precision data access, to dynamically slice and dice the data as desired. Each such access would function the same whether or not the Dataset query is used, as a query followed by data retrieval via the access reference. Repeated accesses to a given dataset would be straightforward. Advanced data access capabilities such as authenticated access, data staging to VOStore, etc., would be supported intrinsically since the standard access services are used.

One concern is that if the Dataset query provides a capability to point to services, this may overlap with the function of the Registry. The key distinction appears to be that this is being done at the level of individual datasets. The individual services would still be registered in the usual fashion, discoverable via a Registry query, and directly callable, however there is no direct way to determine via the Registry alone the access methods available for an individual dataset or aggregate observation.

The Dataset query service would be registered like any other service, and would be used to discover data much like any other DAL service, but would be able to find data of any type, would have an additional capability to describe logical associations of different types of data, and provide pointers to the service instances available to access the data. Data access would normally be a two step process, using the Dataset query as the primary data discovery method, followed by direct calls to SIA, SSA, etc. services for precision data access and retrieval. Use of the Dataset query would be required only for access to complex multi-element data where multiple views of the same data are required. To merely find and retrieve typed data, interfaces such as SIA and SSA would be used directly as at present, without any need for the Dataset query.

Examples

Example 1: 2D projection of a SINFONI data cube

In this case we want to collapse SINFONI spectral data cube data along the spectral axis to produce a 2D image. The cube is not sky-subtracted, and we wish to exclude wavelength regions containing night sky emission or absorption lines.

The process is as follows:

- Perform a Dataset query to locate the data and get a description of all the data elements. Probably all that is needed is a simple positional query assuming SINFONI has its own discovery service, otherwise COLLECTION=SINFONI can be specified, or a time range can be specified to select a single observation if multiple observations of the same field are available.

  The response is a list of all the individual data elements (datasets) comprising the observation. These include the main spectral data cube plus a 2D continuum image of the same field. The cube provides a 3D SIA cube accessor service plus a 1D SSA spectrum extraction accessor.

- The 2D continuum image is retrieved and displayed or otherwise analyzed by the client to determine one or more sky regions.

- One or more 1D spectra are extracted from the cube via the SSA service provided. These are analyzed to locate all the night sky lines.

- The 3D SIA cube accessor service is called to produce the 2D projection, collapsing the cube along the spectral axis. The BANDPASS parameter is passed a range list specifying the wavelength regions (those which do not contain night sky lines) to be used to compute the projection.

Example 2: CGPS Data Viewer

Here we want to reproduce the current CGPS/Aladin data viewer, which can display all the elements of a CGPS survey field and display selected elements in Aladin, including 2D cutouts along the velocity axis of a cube.

- Perform a Dataset query to locate the data for a given CGPS field. There are various ways this could be done, but since this is a galactic plane survey, the simplest approach is to specify the galactic longitude at the center of the field.

- For CGPS fields quite a lot of data is available for a single field:

  2D image at 408 MHZ

  - 2D image at 1420 MHZ (Stokes I continuum)
  - 3D cube at 1420 MHZ (HI line)
  - 2D Preview of 3D cube at 1420 MHZ (HI line)
  - 3D cube at 115 GHZ (CO line)
  - 2D Preview of 3D cube at 115 GHZ (CO line)
  - 2D IRAS image at 12 microns, to same scale
  - 2D IRAS image at 25 microns, to same scale
  - 2D IRAS image at 60 microns, to same scale
  - 2D IRAS image at 100 microns, to same scale

- Various accessors could be provided, but at present these are whole image for the 2D images, whole image for the cubes, or 2D slices at constant velocity for the cubes.
- A list of the available data elements for the field is displayed in Aladin. The user selects the images or slices to be downloaded and displayed.
- Since only 2D image access (cutout style) is required, access is straightforward. To specify a 2D slice of a cube BANDPASS is set to select a single Z value in velocity units. If the cutout service is 2D that is all that is required. If the cutout service is 3D, image geometry parameters can also be specified to ensure that a 2D image is produced.

Example 3: Sub-Cube Extraction

In this case we want to extract a sub-cube of a larger cube, using cutout techniques. The axes of the cube are RA, DEC, and velocity.

- The Dataset query may or may not be required depending upon the complexity of the data and the analysis to be performed. Either the Dataset query, or an SIA query, is performed to locate the cube and determine the characteristics and geometry of the cube.
- A query to the 3D SIA cutout service for the cube is then performed to get an access reference for the sub-cube. The query parameters identify the cube in some fashion, e.g., by its dataset ID, and specify the coverage of the sub-cube in equatorial coordinates for X, Y, and in velocity for Z.
- The query response verifies that the operation is correct and gives the client standard metadata for the image to be returned. The access reference is used to retrieve the cube, which comes back as a FITS file.
- Some pre-existing cube visualization tool is used to view and analyze the sub-cube returned.