March 30st 2011

3D Viewer: Errors and Data Quality

Martin Kümmel, Harald Kuntschner, Mark Westmoquette

Scope

The document contains information on the use of the errors and the data quality information of 3D cubes within the 3D Viewer, which is currently being developed as part of the CASA viewer. The extension of the CASA data model and the viewer shall be based on the usage scenarios described here.

The Data

The prime usage case of the 3D Viewer are MUSE data cubes. Such a reduced MUSE data cube is a multi-extension FITS image with the data in the first FITS extension, the data quality in the second extension and the error (variance or sigma) in the third extension.

During the data processing, a scheme used e.g. in the EURO3D Data Format (http://www.aip.de/Euro3D/E3D/) will be used. In this format the data quality information of a pixel is encoded in a 32bit integer value where every bit denotes a certain type of defect.

The fully reduced and interpolated 3D data cubes will no longer contain many pixels which can be linked to one defective type and thus it is foreseen to use a economic and less space consuming format to store the data quality information. This could either be a boolean value or directly addressing defective pixels with `"NaN"` in the data value, hence dropping the data quality extension for 3D cubes entirely (TBC).

Using the Data Quality in the Viewer

The 3D viewer is primarily a tool to display data, and currently no viewer operation is foreseen that would change the original cube.

For the viewer, the data quality information will be used to:

1. Display the quality of cubes (which pixels have what quality information).
2. Transfer this information to the display of data and errors.
3. Use the data quality information when further processing the cubes to generate e.g. reconstructed images or extract spectra for image regions.

While 1. is already guaranteed by the basic viewing capabilities of the 3D viewer, for 2. and 3. the full data quality information is not needed, but rather the information whether a pixel is "good" or "bad". For 2., good pixels would then be displayed "normal" and bad pixels in a special way, e.g. black or white. For 3., bad pixels would be neglected in the processing.

Hence it is sufficient that the viewer transforms the data quality information, either given as `"NaN"` values or in a data quality extension, during the reading of the data into a mask value "good" or "bad" for every pixel. In the case of reading a data quality extension with various bits indicating different types of defects, the user should be able to decide at every read-in, which bits are considered to be "bad" and "good".

When saving the results from processing data such as in 3. (reconstructed images or extracted spectra), the data quality information on whether an image pixel or spectral element contains valid data or not, should be part of the FITS data saved to disk.

Using the Error in the Viewer

Within the viewer, the error information will be used to:

1. Display the error information for the cube as well as for the products generated in the viewer.
2. As an input for the processing done in the viewer.
3. 3. As an output product from the processing of 3D cube data.

Concerning 1.

Besides the viewing of the pixel errors also a viewing of the errors for reconstructed images and for extracted spectra should be possible. While the viewing of 3D data and 2D data is already covered by the basic capabilities of the viewer, a plotting or overplotting of 1D errors is currently not possible and needs still to be implemented.

Concerning 2.

Processing that uses the error information include:

- some methods to compute a reconstructed image (e.g. error weighted mean);
- some methods to extract a spectrum from the 3D cube (e.g. error weighted mean);
- 2D Gaussian fits on reconstructed images or individual wavelength layers if a 3D cube;
- 1D Gaussian fit on extracted spectra.

Concerning 3.

The resulting error should be part of the output written to disk for the reconstructed images and for the extracted spectra.

Combine methods for reconstructed images and spectrum extraction

- sum (internal error or error propagation)
- average (internal error or error propagation)
- median (only internal error)
- error weighted mean (internal error or propagation)
- robust estimate of the mean (internal error or error propagation)

If possible, the processing methods will in addition to the values also deliver the associated errors. Some of these processing methods, e.g. the average, will have several methods to derive an error, e.g. via error propagation or an internal error derived from the ensemble of values, and the viewer should offer to compute both. However the result of computing a reconstructed image (or extracting a spectrum) will always be one image plus one error image (or a spectrum plus one error value for each spectral bin), thus the various methods of deriving an error will be offered independently.