Astronomical Data Analysis Software and Systems XIII
ASP Conference Series, Vol. 314, 2004
F. Ochsenbein, M. Allen, and D. Egret, eds.

E3D, The Euro3D Visualization Tool

S.F. Sánchez

Astrophysikalisches Institut Potsdam, And der Sternwarte 16, 14482 Potsdam, Germany

Abstract. We present the first version of E3D, the Euro3D visualization tool for data from integral field spectroscopy. We describe its major characteristics, based on the proposed requirements, the current state of the project, and some planned future upgrades. We show examples of its use and capabilities.

1. Introduction

The Euro3D Research Training Network (RTN) (Walsh & Roth 2002) was put forward with the intention to promote integral field spectroscopy (IFS), or "3D" spectroscopy, and to help making it a common user technique. In order to accomplish this, one of the major tasks was identified as the need of providing standard software tools for the visualization and analysis of datacubes. These tools should be general enough to be entirely independent of the origin of data, i.e. 3D instrument. Previously, a heterogenous collection of instrument-specific data formats and software tools (e.g. XOASIS), proprietary software packages and a lack of any standard have hampered a break-through of this powerful observing method, leaving it merely as an expert technique with comparatively limited scientific impact. The complexity of this problem is distribed by Pecontal (2004).

Recognizing the importance of this problem, a work plan was devised to start creating a package of tools for the analysis and visualization of IFS data. Entitled *3D Visualization*, Task 2.2 of this work plan foresees the development of a programme, which should be capable of reading, writing, and visualizing reduced data from 3D spectrographs of any kind. We have named this tool "**E3D**". In Sánchez (2004), we presented the detailed description of the program. We present here the current status of the project, give a brief description of the programme as it is now, point out some requirements which have not yet been met, and explain some problems that were encountered during the development. We also present some examples with real data, trying to explore the potential of the tool already at its first stage of development.

2. Background

One of the major problems for the development of a standard visualization tool is the lack of a standard data format. Every group has developed its own *3D data format*, both for the spectral and the position information (cubes, FITS images,

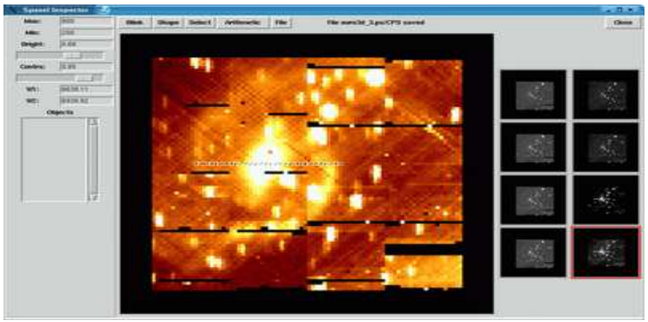

Figure 1. Spaxels Inspector. This is the GUI for plotting monochromatic or polychromatic maps. It is possible to select different spaxels, to be displayed subsequently on the Spectral Inspector.

FITS tables, MIDAS images, etc...). In order to overcome this problem, the RTN has proposed a unified data format, the "Euro3D Data Format" (Kissler-Patig et al. 2004). Taking into account previous experience from more than a decade of operating 3D instrumentation in the visible and the near-infrared, this data format is supposed to cover most foreseeable requirements of existing and future instruments. The Euro3D visualization tool was written specifically to make use of this data format.

It was the scope of the network from the very begining to provide a freely distributed software, that could be installed/used on the largest possible number of computers. This prevents us from developing the software in any commercial (e.g., IDL) or non commercial environment (e.g., MIDAS) that could create a long-term dependence or limit its use. The possibility of using/adapting a previous existing tool (like DS9, XIMAGE or GIPSY) was considered. However, the specific requirements of IFS prevented us from choosing this solution. A major caveat was the requisite of that tools to handle with regular gridded data, like datacubes, which force us to interpolate (i.e., alter) the data to visualize them. Due to all these reasons it was decided to write a stand-alone software in C.

A C-coded library ("LCL") was developed to handle the input/output of data on the proposed format (Pecontal-Rousset et al. 2004). This library allows to read and write not only Euro3D format files, but also reads/writes single spectra, monochromatic datacube slices, FITS images, and FITS tables. We have tested different graphical libraries (NCARG, PLPLOT, X11 low-level routines,etc.) and created different prototypes based on these various libraries. As a result, it was decided to use PGPLOT, mainly due to its flexibility, portability, and in particular its capability to interact with Tcl/Tk. The latter property allowed us to implement a scripting capability.

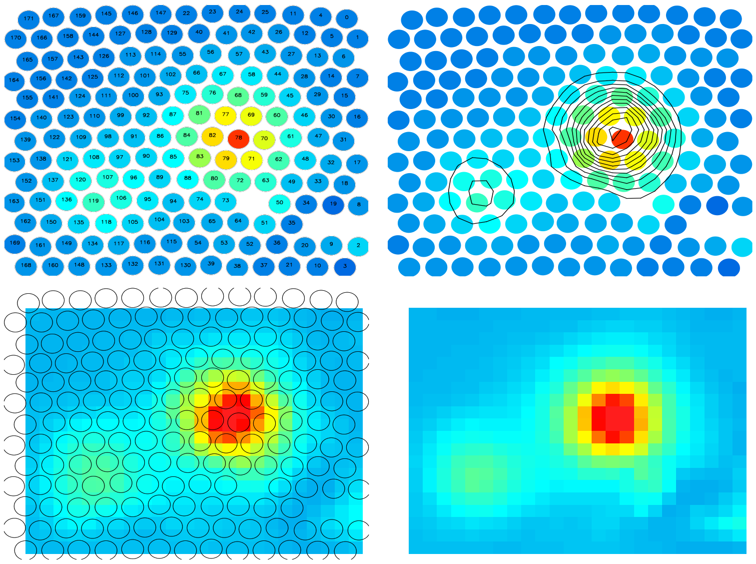

Figure 2. **Top-Left:** Polychromatic map INTEGRAL data of HES 1104-185, using the spaxel representation. **Top-Right:** Same map including a countour plot of the data. **Bottom-Left:** Interpolated representation of the same map, using a Spline interpolation routine. The original spaxels pattern is overplotted. **Bottom-Right:** Similar interpolation, without the spaxels pattern, and using a Natural Neighbour interpolation routine.

3. Requirements & Characteristics

Decisions upon the specification of E3D were made after extensive discussion in various RTN meetings. The main requeriments are: (1) Display all the spectra stored on the file as a single 2D image (*stacked spectra*, one spectrum per row); (2) Display different spatial representations of the data (*maps*) from the stacked spectra representation; (3) Select spectra from the map representation and (4) use of alternative representations, like pseudo-slits. It was demanded that E3D were built with a modular philosophy that allows to integrate different packages on the future. E3D should be able to interact with the major astronomical data analysis packages, like IRAF/PyRAF or IDL. A Shared Memory Server (SHM) was integrated into E3D for this propose, although its capabilities have not been already fully tested. Additional communication methods have been developed and tested, based on the scriptable capabilities of E3D.

E3D comprises a C-coded core, with three main elements: (1) a library, `Euro3D.o` containing low-level I/O and plotting functions; (2) `tk_e3d` , a Tcl/Tk interpreter, which adds the Euro3D routines to the standard Tcl ones and (3) a number of stand-alone C-coded tools that help to handle the Euro3D format. Together with the C-core, we have coded a Tcl/Tk Graphical User Interface (GUI), that uses the Euro3D-Tcl routines (`tk_e3d.tcl`). The GUI comprises three main windows: (1) the Main window or Stacked Spectra Inspector. It comprises the main Menu with different options to handle Input/Output and different representations of the data; (2) the Spaxels Inspector. It comprises a main canvas for plotting monochromatic/polychromatic datacube slices and (3) the Spectral Inspector. It comprises a main canvas for plotting the spectra corresponding to spaxels which were selected from any of the two previous windows. Figure 1 shows a snap-shot of the Spaxels Inspector. As explained above the GUI has been coded in Tcl/Tk using the Euro3D-Tcl routines. This is probably the most powerful outcome of the adopted programming philosophy, since a scripting capability for handling Euro3D data is included "for free". This means that any user can create his/her own Tcl-coded scripts by making use of the Euro3D-Tcl routines, and run them by invoking `tk_e3d` .

E3D actually performs relatively simple routines, like spaxel selection or image reconstruction. Figure 2 shows an example of four different representation of the maps. The top-left panel shows a spaxel representation of a slice cut of INTEGRAL data taken on the gravitational lens HES 1104-185 (Gómez et al. 2004). Each spaxel has a radius of $0.27''$. The top-right pannel shows the same representation including a contour plot. For displaying the contour plot it is needed to interpolate the data, creating a regular gridded map. Five different interpolation algorithms are already available on E3D. The two bottom pannels of Fig.2 show two examples of these interpolation algorithms, both using a $0.3''$/pixel grid. We have also included different representations of the spectra, both in a pseudo slit-spectra form (a spatial cut in the datacube, mimicking a slit-spectrum observation). A number of simple analysis tools has been added to E3D. Among these tools are the `specarith` and `spaxarith` routines. Both routines allow one to perform arithmetic operations between selected spectra and selected polychromatic maps, respectively.

4. Future work

We have designed E3D to be a data visualization and data analysis tool. It is our goal to integrate as many different tasks of the Euro3D software package as possible into E3D. In the end, this strategy will provide a powerful analysis/visualization tool. There are a number of bugs still to be fixed, some of which have been identified. A few basic requirements are still on the queue. For example, it has still to be decided how to handle different wavelength units (for now: Angstroms), and different spatial units (for now: arcsec). There is a need for improved zooming capabilities. We have to think how to treat the data quality flags, and which are the best and most flexible defaults. Different methods of selecting spaxels (area selection) have been proposed, but they have not been coded. It is still under discussion how to handle science tables, and how to plot their contents.

So far, we have tested E3D with data from a variety of instruments : INTEGRAL (Arribas et al. 1998), OASIS, PMAS (Roth et al. 2000), SAURON, SparsePak (Bershady et al. 2003), SPIFFI, TIGER, VIMOS, and with different mosaic patterns (e.g. Sánchez et al. 2004). Some memory bugs and overloading problems have been detected, rendering the program not very efficient for massive reloads of big frames. We need further investigations of how to interact with external packages (IDL, PyTHON, ...), and further tests with the SHM are needed.

However, given this early stage of development, E3D seems to be a promising tool, which has already proven to be useful for visualize and help the analysis of real 3D data.

Acknowledgments. This project is part of the Euro3D RTN on IFS, funded by the European Commission under contract No. HPRN-CT-2002-00305. I like to acknowledge S. Foucaud for his help with python. A.Pécontal-Rousset, P. Ferruit and the entire Lyon group for their advice, help, and their marvelous work on the LCL library.

References

Arribas S., Carter, D., Cavaller, L., et al., 1998, Proc. SPIE, 3355, 821

Bershady, M.A., Andersen D.R., Harker, J., Ramsey, L.W., Verheijen, M.A.W., 2003, PASP, submitted

Pecontal-Rousset, A., Ferruit, P., et al., 2004, Euro3D Science Workshop, 21-23 May 2003, IoA, Cambridge, AN, in press.

Pecontal-Rousset, A., 2004, this volume, 491

Gómez, P., Mediavilla, E., Sánchez, S.F., et al., 2004, Euro3D Science Workshop, 21-23 May 2003, IoA, Cambridge, AN, in press.

Kissler-Patig, M., Copin, Y., Ferruit, P., Pécontal-Rousset, A., Roth M.M., 2004, Euro3D Science Workshop, 21-23 May 2003, IoA, Cambridge, AN, in press.

Roth, M.M., Bauer, S., Dionies, F., et al., 2000, in Proc. SPIE, Vol. 4008, 277-288

Sánchez, S.F., 2004, Euro3D Science Workshop, 21-23 May 2003, IoA, Cambridge, AN, in press. (astro-ph/0310677)

Sánchez, S.F., Christensen, L., Becker, T., Kelz, A., Jahnke, K., Benn, C.R., García-Lorenzo, B., Roth, M.M., 2004, Euro3D Science Workshop, 21-23 May 2003, IoA, Cambridge, AN, in press. (astro-ph/0310293)

Walsh, J. R. & Roth, M. M., 2002, The Messenger, 109, 54