Planetary and Space Science 59 (2011) 1265–1272

Contents lists available at ScienceDirect

Planetary and Space Science

journal homepage: www.elsevier.com/locate/pss

A working environment for digital planetary data processing and mapping using ISIS and GRASS GIS

Alessandro Frigeri [a,d,*], Trent Hare [b], Markus Neteler [c], Angioletta Coradini [a], Costanzo Federico [d], Roberto Orosei [a]

[a] Istituto di Fisica dello Spazio Interplanetario—INAF, Roma, Italy
[b] United States Geological Survey, Flagstaff, AZ, USA
[c] Fondazione Edmund Mach, Research and Innovation Centre, S. Michele all'Adige, Trento, Italy
[d] Geologia Strutturale e Geofisica, Dipartimento di Scienze della Terra, Universitá degli Studi di Perugia, Perugia, Italy

A R T I C L E I N F O

Article history:
Received 23 April 2010
Received in revised form
15 October 2010
Accepted 15 December 2010
Available online 17 January 2011

Keywords:
Planetary mapping
Planetary geology
Geographic Information Systems
Free open source software
Human–computer interaction
Digital maps
Image processing

A B S T R A C T

Since the beginning of planetary exploration, mapping has been fundamental to summarize observations returned by scientific missions. Sensor-based mapping has been used to highlight specific features from the planetary surfaces by means of processing. Interpretative mapping makes use of instrumental observations to produce thematic maps that summarize observations of actual data into a specific theme. Geologic maps, for example, are thematic interpretative maps that focus on the representation of materials and processes and their relative timing. The advancements in technology of the last 30 years have allowed us to develop specialized systems where the mapping process can be made entirely in the digital domain. The spread of networked computers on a global scale allowed the rapid propagation of software and digital data such that every researcher can now access digital mapping facilities on his desktop.

The efforts to maintain planetary missions data accessible to the scientific community have led to the creation of standardized digital archives that facilitate the access to different datasets by software capable of processing these data from the raw level to the map projected one.

Geographic Information Systems (GIS) have been developed to optimize the storage, the analysis, and the retrieval of spatially referenced Earth based environmental geodata; since the last decade these computer programs have become popular among the planetary science community, and recent mission data start to be distributed in formats compatible with these systems.

Among all the systems developed for the analysis of planetary and spatially referenced data, we have created a working environment combining two software suites that have similar characteristics in their modular design, their development history, their policy of distribution and their support system. The first, the Integrated Software for Imagers and Spectrometers (ISIS) developed by the United States Geological Survey, represents the state of the art for processing planetary remote sensing data, from the raw unprocessed state to the map projected product. The second, the Geographic Resources Analysis Support System (GRASS) is a Geographic Information System developed by an international team of developers, and one of the core projects promoted by the Open Source Geospatial Foundation (OSGeo). We have worked on enabling the combined use of these software systems throughout the set-up of a common user interface, the unification of the cartographic reference system nomenclature and the minimization of data conversion. Both software packages are distributed with free open source licenses, as well as the source code, scripts and configuration files hereafter presented. In this paper we describe our work done to merge these working environments into a common one, where the user benefits from functionalities of both systems without the need to switch or transfer data from one software suite to the other one. Thereafter we provide an example of its usage in the handling of planetary data and the crafting of a digital geologic map.

1. Introduction

In the 1960s the missions to the moon and solar system returned data in a non-digital domain, and processing required dedicated

---

* Corresponding author at: Istituto di Fisica dello Spazio Interplanetario (IFSI), Istituto Nazionale di Astrofisica (INAF), via del Fosso del Cavaliere 100, 00133 Roma, Italy. Tel.: +39 075 586 71 76.
E-mail address: alessandro.frigeri@ifsi-roma.inaf.it (A. Frigeri).
URL: http://www.unipg.it/~afrigeri (A. Frigeri).

0032-0633/$ - see front matter © 2011 Published by Elsevier Ltd.
doi:10.1016/j.pss.2010.12.008

personnel and facilities. As soon as digital conversion methods were developed, planetary exploration has always demanded digital methods and computerized processing techniques, offering digital data that can be copied without limits and without generational loss. During the late 1970s, processing techniques were very specialized and required expensive mainframe computer systems. It was only until the late 1980s when it became possible to cartographically process these images within a digital domain on affordable Personal Computers.

Specialized image-processing software such as the Integrated Software for Imagers and Spectrometers (ISIS) have been developed to process a wide range of planetary instrument data and to project this data in a geospatially referenced form, allowing the production of high level data products suitable for mapping purposes.

Beside planetary missions, Earth Observation (EO) satellites requires similar processing software, but the need to merge remote-sensing data with a large amount of direct measurements taken directly on the Earth's surface led to the spread of Geographical Information Systems (GIS) designed to manage administrate, display and modify multiple data layers and data types. In particular, beside the matrix based or *raster* model, GISes offer a *vector* data model to efficiently manage geometric features as points, lines, and areas associated with non-geographic data stored as *attributes* within a database system. Due to the capabilities of capture, store and handle large amount of spatially referred informations, in the last 10 years Geographic Information Systems (GIS) have become popular in the planetary science community (Hare et al., 2009). Among the various GISes available, the Geographic Resources Analysis Support System (GRASS) possess interesting similarities with ISIS, in its development history, software design, distribution policy and support system.

Both ISIS and GRASS are distributed as *free software* along with user licenses that declare legal statements about the use of the programs. Even if they are available from their respective websites at no cost, the term *free* in *free software* refers to the rights to (1) use without limitation, (2) copy, (3) study and (4) improve the software (Stallman, 1990). The rights to use without limitations and to copy the software, defined in the license agreements, guarantee the spreading of softwares. The rights to study and improve the software imply the access to the source code and the permission to modify it. In order to distinguish *free software* from *free*, *freely available* or *freely distributable* it has recently been proposed to use the term *Free Open Source Software* (FOSS) to indicate software provided with license agreements that guarantee the four rights above mentioned. Softwares distributed with FOSS licenses represent an ideal choice for a reproducible research, where the implementation of a method is published together with the results. As they are FOSS software, every procedure in ISIS and GRASS can be verified, and research institutions, companies and individuals can actively customize the programs as needed and participate in the development of the projects. Economical resources are thus concentrated in development and training instead of the acquisition of software, allowing a rapid implementation of new features, this is especially critical in ongoing scientific space missions, where international teamwork on the same code-base is typically more efficient over the replication of functionalities in different, autonomously developed or closed source projects.

This paper describes procedures developed to allow the combined use of ISIS and GRASS in order to provide an integrated environment for the analysis of planetary data in a reproducible research environment, from the low level form to the production of derived thematic data products as e.g. for planetary geologic maps. In order to develop a usable system, we have worked on the integration of these two systems focusing on the unification of the user interface, the unification of the projection definition system and the minimization of data conversion between the systems.

2. Historical background and software capabilities

Since the Apollo and Viking era, the handling and processing of planetary remote sensing data has represented a necessary component for studying geologic processes of planetary bodies (Greeley and Batson, 1990). In the same period, computer systems started to be used to handle ground-based observations and remotely sensed environmental data of the Earth. Early on, this software was only applicable on expensive mainframe machines. The development of more accessible user interfaces and the efforts to standardize programming languages to encourage code portability allowed non-computer scientists to process and analyze planetary data directly at their home institutes. The evolution of the ISIS and GRASS systems share many interesting common aspects. Time wise, they each represent the result of about 30 years of development efforts. On the software programming side, they share the same modular design and the same type of user interface. Moreover, they represent two projects originally developed in a single country and now used and continued by the effort of the international scientific community.

2.1. ISIS

The history of ISIS began in 1971 at the United States Geological Survey (USGS) in Flagstaff, Arizona. The first package developed was named the Flagstaff Image Processing System (FIPS), and was written for the Digital Equipment Corporation (DEC) Programmed Data Processor (PDP) computer system. As soon as DEC released their first VAX/VMS (Virtual Address eXtension/Virtual Memory System) systems in the early 1980s, FIPS was ported to take advantage of this more current architecture. This new system was named the Planetary Image Cartography System (PICS), and featured a strategically important component: the Transportable Application Executive (TAE) developed by NASA (National Aeronautics and Space Administration) Goddard Space Flight Center (Van Wie et al., 1980). While TAE would now be considered as a primitive user interface, it was the first step in making the PICS remote sensing application more accessible to non-computer scientists.

With the technologic evolution of optical instruments used onboard planetary missions in the late 1980s, the need to handle multi- and hyper-spectral image data increased the need for a more flexible system, thus the capabilities of PICS were ported into a new system named the Integrated System for Imaging Spectrometers (Torson and Becker, 1997; Gaddis et al., 1997). In the 1990s, the development of ISIS has continued and has been extended to support a large number of instrument sensors used for planetary missions. The availability of ISIS via the Internet allowed its use at different planetary facilities thanks to its portability to UNIX-derivatives. In the late 1990s, ISIS was ported to GNU/Linux, an operating system written from scratch but replicating UNIX functionalities. GNU/Linux was created by starting from the GNU (Gnu's Not Unix) project and the Linux kernel in 1991, and distributed with a FOSS licensing system, the General Public License (GPL), that encourages the collaborative development of the software by the industry, research institutions and individuals.

1990s were also the times when Graphical User Interfaces (GUI) were developed to provide a more intuitive access to the software, and ISIS started to introduce GUI-based programs beside the TAE environment and the command line tools in its suite. Starting from 2001, the availability of a Free Open Source Application Programming Interface "Qt" and the spread of new Object Oriented programming languages led to the development of a re-engineered version of ISIS (Anderson et al., 2004).

The evolution of ISIS, called ISIS3, written completely in C++, represents the latest generation of the software featuring a more efficient data format and input/output (IO) libraries, a GUI support based on "Qt", and also an improved licensing system. The new C++ Application Programming Interface of ISIS is well documented and specific high-level classes are provided for collaborating on the implementation of new camera models (Anderson, 2008).

ISIS adopts his own file format, the ISIS Cube format, which supports 8, 16 and 32 bit Data Numbers (DN) with and internal ordering optimized for imagers and spectrometers (Torson and Becker, 1997). The latest ISIS3 tiled cube format together with the updates of the IO routines of ISIS3, allow a flexible spatialwise and spectralwise access to sub-regions of the dataset.

ISIS is made by a set of libraries and software modules devoted to solve specific problems. This design is extremely useful to build processing pipelines within scripts that can recursively process large amounts of data. Every ISIS module can be executed from a Command Line Interface (CLI) or, for a more intuitive interaction with the user, through a GUI. Finally the license agreement provided with ISIS3 encourages the use and the collaboration for supporting its development. This combination of software design and licensing system, allow to expand the software capabilities for example with new modules, without affecting the performance of the overall system.

ISIS features also a web based support system where users at various level of experience and developers meet to discuss about ISIS. Within the ISIS support system (isis.astrogeology.usgs.gov/IsisSupport/) it is possible to track ISIS developments, discuss on ISIS functionalities, notify possible bugs and propose improvements.

2.2. GRASS GIS

The first release of GRASS GIS took place in 1982 as a result of development efforts of the US Army Construction Engineering Research Laboratory (USA-CERL) in Champaign, Illinois. A core team of programmers both from USA-CERL and some universities joined their forces into the development of a GIS suitable for environmental analysis, thus filling the lack of raster support of proprietary, mostly vector based GIS at that time (Westervelt, 2004). Since 1997, the software is now developed by the international GRASS Development Team. In 2006, the GRASS project's team became a founding member of the Open Source Geospatial Foundation (OSGeo, www.osgeo.org), and, through this, it holds today a formal membership in a not-for-profit legal entity. The GRASS project shifted its infrastructure to OSGeo in order to join a centralized home for the web site (grass.osgeo.org), Wiki-based help system, source code repository, community add-on module repository and an integrated bug tracking system. Since the beginning of the project, annual meetings, bulletins and publications, allowed the user community to keep track of updates and share of specifically developed programs.

Nowadays, GRASS is one of the core components of the Open Source Geospatial software stack. It includes a topological 2D/3D vector engine with Structured Query Language (SQL) attribute management, it handles 2D and 3D (voxel) raster data, and offers vector network analysis functionality. The user can choose from several database back-ends, file based as DBF, sqlite or client-server as Postgres, Oracle or MySQL. GRASS offers many spatial modeling algorithms, 3D visualization as well as image processing routines pertaining to Laser Imaging Detection and Ranging (LIDAR) and multiband imagery. It integrates well with other Open Source and proprietary software packages for geostatistical analysis, cartographic output and Web GIS applications (Neteler et al., 2008). It can be used on common operating systems and both a graphical user interface and a command-line user interface are available.

GRASS GIS stores raster and vector data, parameters for mapping and database connections into a directory tree called *Location*. Data in different Locations are exchangeable and reprojection is done internally in GRASS. Interoperability is granted by supporting the majority of common GIS data formats through the Geographic Data Abstraction Library/OpenGIS Simple Features Reference Implementation (GDAL-OGR) library (Warmerdam, 2008). This FOSS library supports an extensive range of raster and vector formats, including Open Geospatial Consortium (OGC) Simple Features, and has been adopted by a wide range of GIS software packages and virtual globes. Additional modules of GRASS allow to import maps from OGC WFS (Web Feature Service) and WMS (Web Mapping Service) external servers. Statistical analysis can be made in GRASS by using dedicated R and GStat interfaces. R Statistical Computing Environment can be accessed within GRASS thanks to the R/GRASS interface available from the Comprehensive R Archive Network (CRAN), while GStat comes with an option to support GRASS data. Data and graphic output is provided by modules that export maps in HTML, Cairo/PDF and MATLAB. Three dimensional data (e.g. topographic surfaces, volumes) can be exported from GRASS to POVray ray tracer and Paraview/Visualization Tool Kit (VTK) multi-dimensional visualization software (Neteler and Mitasova, 2008).

GRASS is written in a fully modular way with minimal overhead which allows the user to run the system or parts of it in even modern smart devices with limited RAM. Similarly to ISIS, the user that wishes to expand GRASS capabilities can develop new modules in C or in scripting languages, without affecting system performance. Recently a new Python library has been implemented which gives access to many common GIS library tasks, greatly enhancing and simplifying access to powerful and streamlined geospatial and workflow functions. For advanced programming a Python Simplified Wrapper and Interface Generator (SWIG) interface is offered to gain access to the full suite of C library functions. The availability of bash, Python and C as programming language for extending GRASS capabilities allow the advanced user to chose which language is the best suitable for the implementation of a new specific feature.

General GRASS GIS support is based on email list manager and several mailing lists allow users and developers of GRASS GIS to discuss about software usage and to coordinate the development of code, documentation and translations. Regional groups of users have specific mailing lists in their own language, and specific GRASS GIS or FOSS for geospatial meetings and conferences allow the users' and developers' community to meet face to face.

3. The GRASS/ISIS working environment

While GRASS is undergoing a partial re-writing for the upcoming version 7, we decided to work on the currently available version 6.4 which is suitable for a stable production environment. This way, the procedures presented in this work can be used now but will also be ready for future versions of both software systems. GRASS GIS supports many available computer architectures, while ISIS3 supports MacOSX and GNU/Linux systems. Due to this constraint, the combined ISIS/GRASS environment can currently be run on these last-mentioned architectures.

To set up a common environment to allow the combined use of ISIS and GRASS, we have set up a series of programs and configuration files that will fulfill three major needs: (1) to control the two software from the same shell, (2) to share the same cartographic reference system and (3) to avoid as much as possible the replication of data generated by export and import processes, without compromising efficiency.

3.1. Unifying the command line interface

In order to avoid cluttering the operating system with non common programs and libraries, during their runtime both ISIS and GRASS need to set up special file paths to enable the access to their vast range of executables and libraries (ISIS has about 240 modules, while GRASS has 340 executables plus several add-ons). To allow the combined use of ISIS and GRASS we prepared a shell script, named *isis-grass* that sets up the ISIS environment within a customized GRASS session, named ISIS-GRASS. This way, the same Command Line Interface (CLI) holds commands both from ISIS and GRASS as well as the Graphical User Interface (GUI) of both suites, and the user can execute commands from both suites without changing the shell working environment. As the basic operating system functionalities are still available within ISIS-GRASS shell, it is possible to use scripting languages to automate the execution of a sequence of ISIS and GRASS modules in the user defined order. As soon as the user quits the working environment, the operating system returns to his previous state, until the next call of the ISIS-GRASS.

Within the common ISIS/GRASS environment we have access both to GIS data organized in the GRASS *Location*, and to the ISIS data *cubes* stored in the user's file system tree.

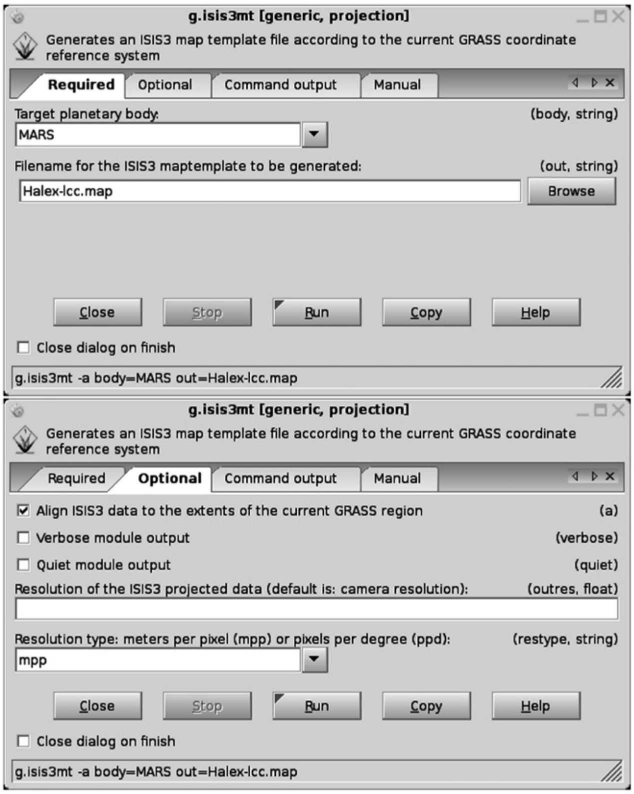
The different software foci of GRASS and ISIS combined, allow to manage all the data necessary for researchers interested in planetary mapping. Global maps, distributed as GIS-ready files or by Web Service can be imported and stored in the GRASS in their native reference system and resolution. As ISIS focusses on processing instrument data from the raw archive format (e.g. NASA/Planetary Data System or ESA/Planetary Science Archive) to the map projected data, and GRASS and ISIS define mapping parameters in different ways, in order to allow a seamless projection of ISIS *cubes* into the GIS layers' stack, it is necessary to introduce new capabilities within the *isis-grass* working environment.

3.2. A common cartographic reference system

As GRASS GIS does not include planetary bodies' ellipsoidal figures, we enabled this feature by translating the IAU2000 ellipsoidal parameters into the ellipse table format of GRASS GIS (Seidelmann et al., 2002). This ASCII file, added to the Earth ellipsoid definitions distributed with GRASS, allows to recognize, for example, the Mars spheroidal figure and correctly performs map re-projections and geometric computations on non-projected maps referenced by a Geographical Coordinate System (GCS).

A critical issue in the ISIS/GRASS environment is resulting from the different way and nomenclature in which GRASS and ISIS store cartographic projections' parameters. Every GRASS *Location* has an associated ellipsoid and spatial reference system (srs) defined during its creation, and all the data within the *Location* is referred to that srs. When comes to project data on a planetary surface, ISIS3 uses a group of mapping parameters included in an ASCII file called *map template* to define the srs where to project processed data.

In order to facilitate the projection of ISIS *cubes* on the GIS layers' stack, we developed a GRASS add-on module that automatically generates an ISIS compatible *map template* file according to the cartographic projection currently used by the GRASS *Location*. This module has been written in Python, using the GRASS/Python scripting interface and C library SWIG interface. The module has options to specify the resolution of the projected data according to original instrument resolution, or a user defined one in meters per pixel or pixel per degrees. Another option allows to limit the latitude and longitude range of the projected data so that ISIS will reproject data only in the current GRASS computational region. This option is extremely useful for high resolution and long swaths, where the projection of only the data within the region of interest can save cpu-time and disk-space. As well as the other ISIS and GRASS modules, it can be used through the Command Line Interface (CLI) or through the GUI, as shown in Fig. 1.



g.isis3mt [generic, projection]

Generates an ISIS3 map template file according to the current GRASS coordinate reference system

Target planetary body: (body, string)
MARS

Filename for the ISIS3 maptemplate to be generated: (out, string)
Halex-lcc.map

☐ Close dialog on finish

```
g.isis3mt -a body=MARS out=Halex-lcc.map
```

g.isis3mt [generic, projection]

Generates an ISIS3 map template file according to the current GRASS coordinate reference system

☑ Align ISIS3 data to the extents of the current GRASS region (a)
☐ Verbose module output (verbose)
☐ Quiet module output (quiet)

Resolution of the ISIS3 projected data (default is: camera resolution): (outres, float)

Resolution type: meters per pixel (mpp) or pixels per degree (ppd): (restype, string)
mpp

☐ Close dialog on finish

```
g.isis3mt -a body=MARS out=Halex-lcc.map
```

Fig. 1. The graphical user interface of the GRASS *g.isis3mt* module. Top: the required parameters are the planet's name and the name of the ISIS3 compatible map template file. Bottom: the optional parameters to specify the extension and the resolution of the map to be projected.

With the map template file generated by *g.isis3mt* module, ISIS modules devoted to cartographic projection, as *cam2map* or *map2map*, can use this file to project data into the same cartographic reference system used in GRASS (see Fig. 2).

3.3. Minimizing data replication

Another goal of the work is to detect how to avoid as much as possible data replication between ISIS and GRASS.

The GDAL library is the back-end used by GRASS and other GISes to import and export data in a large number of remote sensing data formats. Written in C/C++, it provides a single abstract data model to read and write geospatial raster data through an Application Programming Interface (API) in the most common modern programming languages (Perl, Python, Ruby, Java, C#, .Net) which allows many other applications and simple image viewers to internally use the library. GDAL is community maintained and released under the MIT Free Open Source License originated at the Massachusetts Institute of Technology (MIT, used by the MIT X Consortium.X/MIT) and has basic support for several planetary data formats including Planetary Data System (PDS), Flexible Image Transport System (FITS) formats and ISIS.

A specialized module available in the GRASS GIS suite, named *r.external*, uses GDAL functionality to register data in foreign formats into the GRASS *Location* without importing.

This way, the ISIS map-projected cubes, produced by ISIS projection modules using the *map template* from *g.isis3mt*, and registered by *r.external*, can be opened directly in GRASS GIS without the need of data conversion (see right-hand side of Fig. 2).

In order to evaluate the performance of computations made with *r.external* registered maps versus the same ones made on

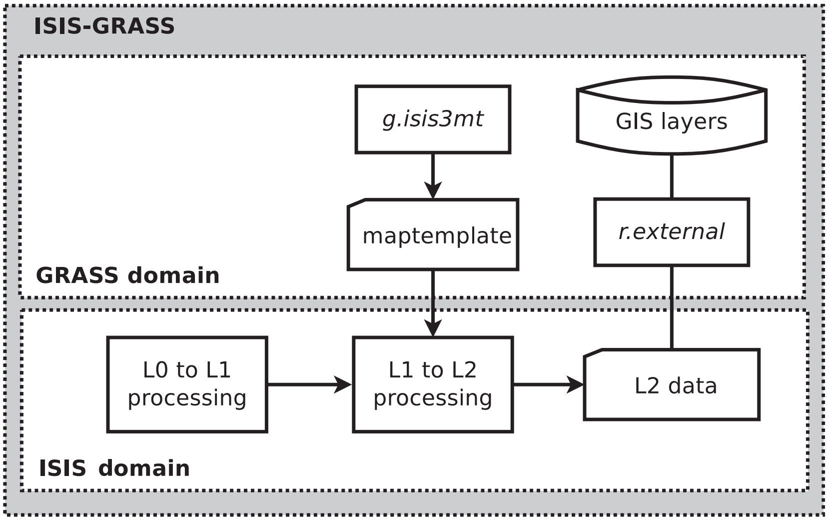


Fig. 2. Block diagram of the interaction between ISIS and GRASS within the common working environment. The *g.isis3mt* GRASS add-on developed in this work generates an ISIS *maptemplate* file used by ISIS modules to project data in the same reference system of the GIS layers of GRASS. The *r.external* module is used to register ISIS level 2 data into the GRASS layers' stack without importing the file.

GRASS native format maps, we computed the cpu-time needed to perform first and second derivative analysis on an uncompressed integer digital elevation raster map, and we found that the performance are about the same between the two cases, with a variance of less than two percent over different runs. As there is no noticeable performance loss in this specific case, it is reasonable to register projected ISIS *cubes* directly into GRASS without the need of data import.

4. Digital geologic mapping with ISIS/GRASS: a working example

As geologic mapping is the result of an interpretation process on actual data, having handy the original data in the same common cartographic reference allows a quick comparative observation of several dataset. GIS' internal organization of data in layers represent an ideal solution for mapping geologic features, as the mapper can interactively select which map to use as the base of his interpretative work. The comparison of co-registered topography and optical imagery at various wavelengths greatly helps the mapper to identify different units and geologic features.

Moreover, a geologic map is particularly well described by the digital *vector* data model of a GIS: every geometric entity is associated with a series of non-geometric attributes that further describe the element. The non-geometric information is stored in an attribute table within a quryable database. For example, a geologic unit is geometrically described by coordinate tuples which define an area, and has associated some non-geometric related information, as for example the unit's letter-sign, a description, the relative age and its color code. GISes allow to efficiently handle these information so the user can select and combine wanted information both geographically and by querying the tables of attributes within the database.

Using *isis-grass* here we set up the working environment for the crafting of a simple example of geologic map starting from raw instrument data and high-level data products, to the publication of the results in a format compliant with most existing system.

The base layers we used for mapping are global maps at moderate resolution and specific instrument frames at very high spatial resolution. Within *isis-grass*, we used GRASS GIS features to handle and re-project global maps, while ISIS functionalities have been used to process and project high resolution data.

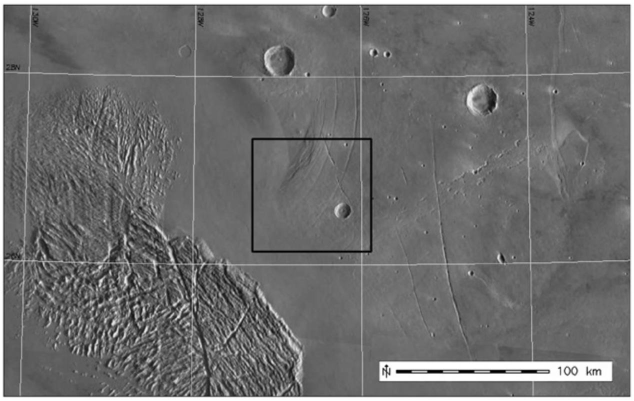


Fig. 3. MDIM21 of the terrains north-east of Olympus Mons, Halex fossae area is located within the black frame.

4.1. Set-up of the base layers

As shown in Fig. 3, the Mars Digital Image Model (MDIM21, Archinal et al., 2004) has been projected into a GRASS *Location* set to the Mars ellipsoid using Lambert Conic Conformal projection centered at 126.5°W, 26.5°N and with 26°N and 27°N as standard parallels. This area is located north-east of Olympus Mons and is characterized by various volcanic materials and tectonic features, as described in detail by Morris and Tanaka (1994). The black frame in Fig. 3 represents the area of Halex Fossae (Greeley and Batson, 1990, p. 211, Figure 7.11), where the evidence of different geologic features as contacts from different terrain, faults and craters offer an ideal test area to be re-investigated using ISIS/GRASS and recent data.

In order to evaluate the topographic contrasts within the area, the results from the Mars Orbiter Laser Altimeter (MOLA), gridded into an equirectangular digital map, offer a global coverage of the topography of Mars at a resolution of 128 pixels per degree. The Mission Experiment Gridded Data Record topography (MEGDR, Smith et al., 2002) has been projected into a the GRASS *Location* set to the Lambert Conformal projection used for the working session. The topographic and the derived slope map allow to evaluate quantitatively the morphologic features of the area, as shown in Fig. 4.

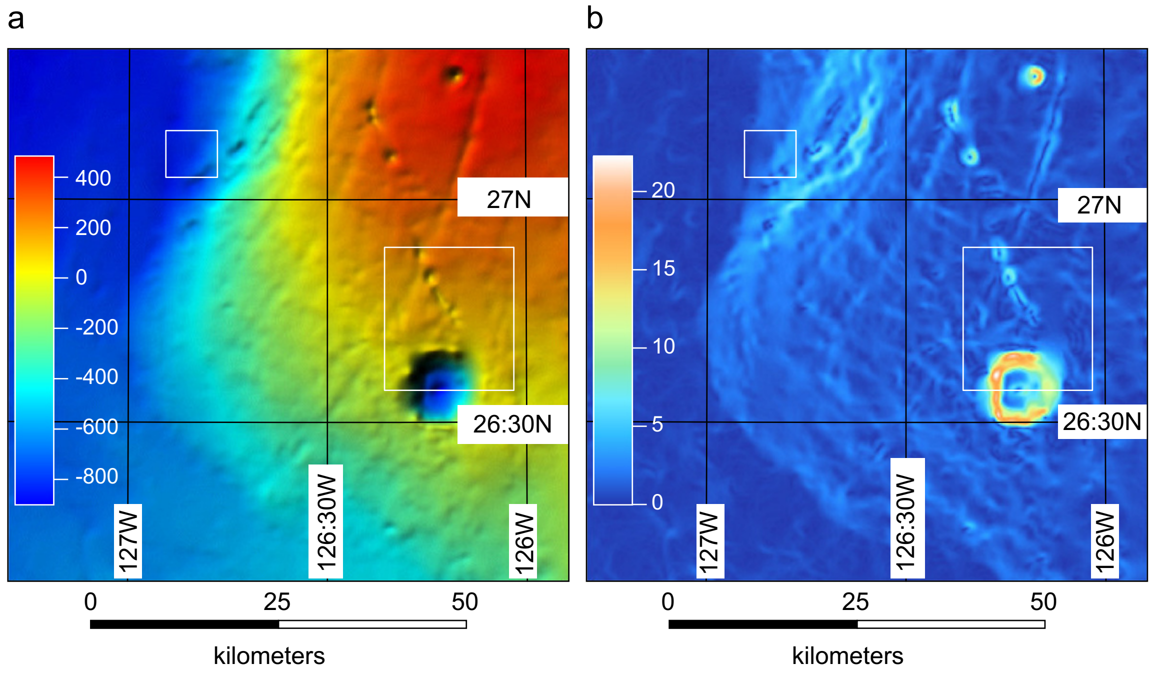


Fig. 4. Morphology of the study area located by the black frame in Fig. 3: (a) topographic map, color scale is in meters and (b) slope map computed in GRASS, color scale is in degress. These maps enhance morphologic breaks that often divide different terrains. White frames corresponds, respectively, from left to right, to imagery data in Figs. 5 and 6 (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

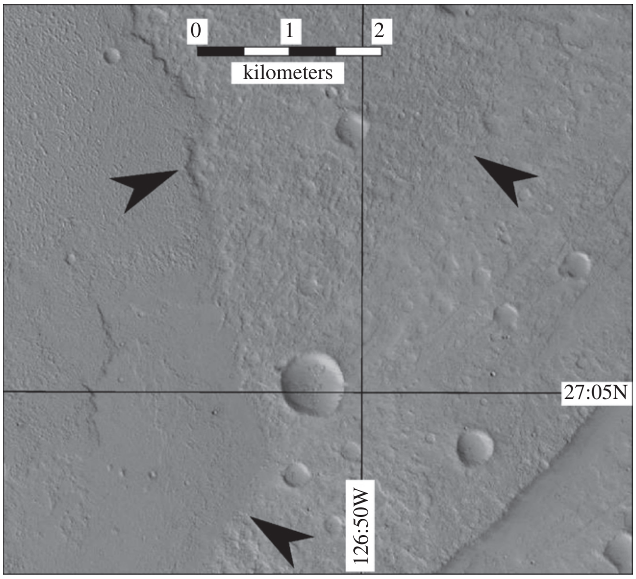


Fig. 5. Terrains with different textures and crater densities are show in CTX *P06_003212_2061_XN_26N126W*. The black arrows indicate the places where the boundaries are more evident.

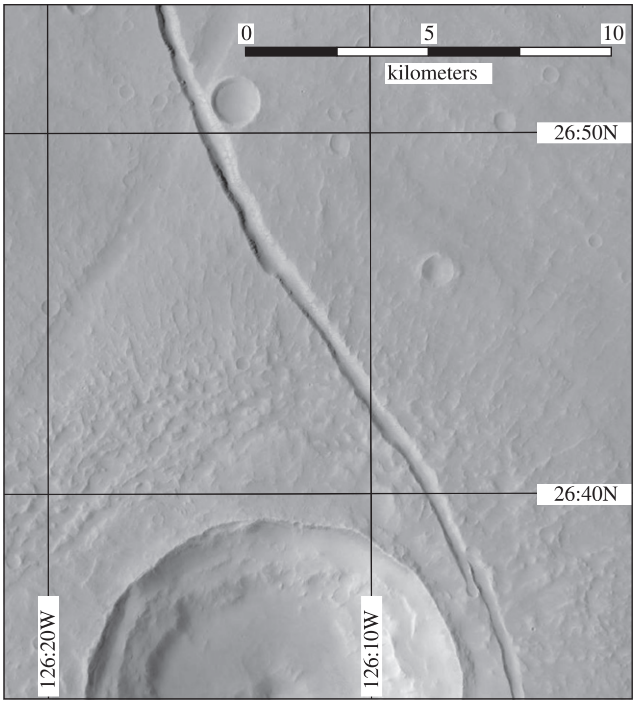
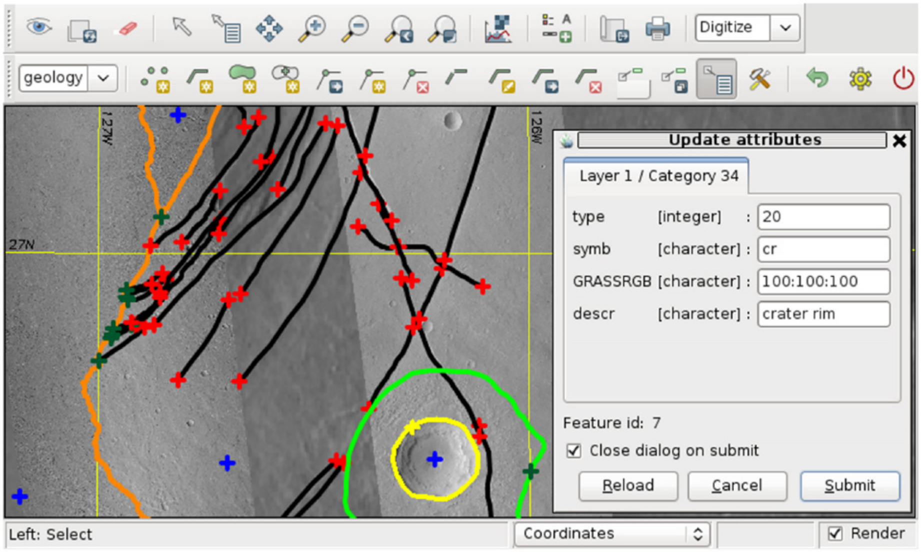


Fig. 6. From CTX *P21_009409_2075_XN_27N126W* it is possible to evaluate the sequence of tectonic events, and their relationship with the main cratering event in the studied area.

Different terrain textures and their geometric relationships are best observed by a high-resolution optical instrument. Two Mars Reconnaissance Orbiter Context Camera (CTX) images have been located in the study area and acquired in their original Planetary Data System format (Malin et al., 2007). The software modules of ISIS process instruments' data from the so-called level 0 to the radiometrically corrected level 1 and finally to level 2. Using the *g.isis3mt* herein developed, the CTX data we have selected and processed from levels 0 to 1 can be map projected (level 2) in the same cartographic reference system used in the current GRASS *Location*. Throughout the registration with *r.external*, level 2 CTX data can then be used in their native *cube* format as a data layer in the GRASS session, as shown in Figs. 5 and 6.

4.2. Crafting of the digital geologic map

While most GISes require to edit vector maps as separate layers of points, lines and polygons, GRASS offers an editing interface in which the user insert all these different geometries at once.



Layer 1 / Category 34

type [integer] : 20
symb [character] : cr
GRASSRGB [character] : 100:100:100
descr [character] : crater rim

Feature id: 7

☑ Close dialog on submit

Fig. 7. The digital geologic map crafting from the GRASS graphical user interface. The mapper chooses interactively from the GIS which layers are most useful for the detection of geologic features. Geologic units are defined by centroids within an area. Faults and crater rims are digitized as segments. The attribute table on the right shows the non-geometric information associated with the crater rim.

Vector maps in GRASS are edited as Eulerian graphs (Chartrand, 1985), where the connection of arcs and nodes can describe all the geometric entities from points to lines and polygons. This way of editing is very intuitive for geologic mapping as the mapper can trace all the geometric elements as faults, unit boundaries, crater rims all as a single layer.

Fig. 7 shows the editing process of a geologic map within ISIS/GRASS, with geometries of geologic features digitized over CTX data used as the mapping base.

Once the digital geologic map has been completed, it is possible to select its features and geometries throughout queries. For example, to produce a printout version of the map we can query and select all the elements as colors, unit-signs, and descriptions of the geologic units to be represented. Fig. 8 shows the geologic map of the studied area, produced by the *ps.map* module of GRASS. Every graphic element as units' letter-signs, colors and line-styles have been extracted by querying the attributes' table of the digital geologic map.

The geologic map produced in ISIS/GRASS can be exported to be read by other systems. The vector counterpart of GDAL, OGR exports vector data and associated attributes from GRASS to more than 30 formats in a consistent way. As OGR implements relevant Open Gis Consortium standards, data can then be easily transferred to other proprietary or Free Open Source desktop GISes, or digitally published on the web through an OCG Web Service capable server.

5. Discussion and conclusions

The computer codes and procedures presented in this study allow to set up the working environment for digital planetary data processing and mapping. The complementary functionalities of ISIS and GRASS within a common working environment allow the planetary scientist to handle a wide range of planetary data and extract necessary information for the crafting of a digital geologic map. Moreover, the licensing system that comes with them opens a wide range of perspectives as it facilitates and encourages local and international cooperation of research and governmental institutions, space agencies, companies and individuals.

The principal goals necessary to integrate the systems were as follows: the unification of the user interface, the unification of the projection definition system and the minimization of data conversion between the systems. With our developments, ISIS and GRASS commands are available within the same command line user interface by calling the isis-grass script, the additional g.isis3mt module developed herein allows to work within a common cartographic reference system, and data import and export is minimized thanks to the use of a GRASS module that makes use of the specialized data abstraction library GDAL without noticeable performance loss in displaying and processing.

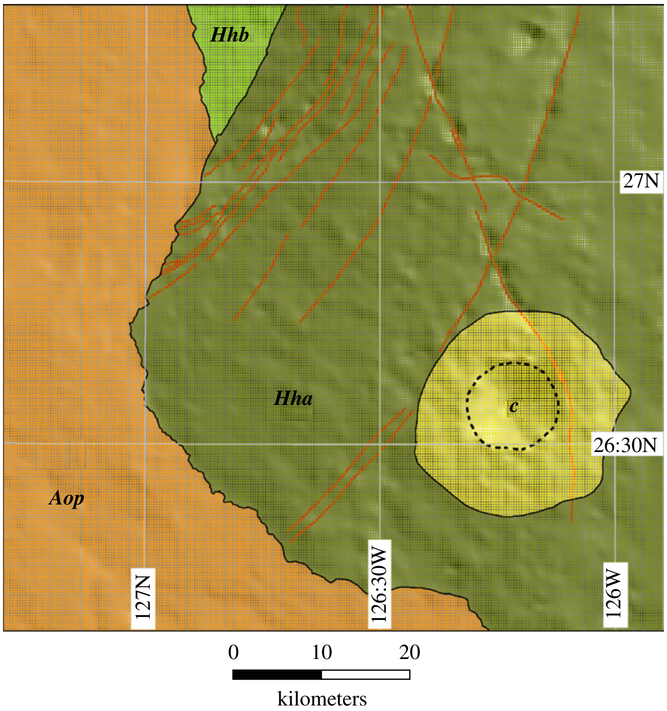


Fig. 8. The printout representation of the digital geologic map. The color of the graphic elements and the letter-signs are stored within the attribute table of the map. Units represented are the plain member of the Olympus Mons Formation *Aop*, the Halex terrain formation *Hh*(*a*,*b*) and crater-related materials *c*. Units' boundaries, craters' rims and faults are, respectively, represented as continuous black lines, black dotted lines and continuous red lines (for interpretation of the references to color in this figure legend, the reader is referred to the web version of this article).

Within the combined ISIS/GRASS system, unprocessed or semi-processed data from spectrometers' and imagers' sensors are treated by the specialized modules of ISIS and projected in a common cartographic reference system and registered as a GIS layer into the GRASS layers' stack without the need of data import/export, and without noticeable loss of performance.

For the mapper, GRASS GIS offers a vector editing that is particularly well suited for crafting digital geologic maps. In this work we have used the vector engine of GRASS to compile a simple digital geologic map on top of imaging sensor data and global maps.

The geologic map can then be exported in formats compliant to the Open Gis Consortium specifications, and thus it can be published electronically and imported in proprietary and FOSS GISes that implement OGC import capabilities. The open specifications of OGC guarantee future access to the maps as soon as upcoming data will require updates. Geologic maps would be more valuable if they are released in a digital geospatial context instead of just the usual printed maps due to their analytical utility, transportability and versioning. Starting in 2011, all NASA geologic maps funded by Planetary Geology and Geophysics Program (PG&G) must be released in an OGC compatible format, so they can be imported in GISes that support this open standard. We believe that, as soon as new planetary mapping programs will be defined by space agencies other than NASA, it will be critical to consider the release of digital maps in a GIS compatible open-format.

Acknowledgment

The first author would like to express his gratitude to the personnel of the USGS Astrogeology Science Center in Flagstaff, AZ for their kindness and valuable discussions during his visit in 2008. In particular, the author would like to thank Ken Tanaka and Trent Hare for the invitation and Paul Geissler for encouraging the author's first steps in using GRASS GIS with planetary data in 2002.

Appendix A. Software developed

The scripts, the configuration files, the *g.isis3mt* module and their relative documentation developed to set up the ISIS/GRASS environment are available from the author's web-page http://www.unipg.it/∼afrigeri/software/isisgrass and are free software as well as ISIS and GRASS, to facilitate future developments and the inclusion of the *g.isis3mt* module into the GRASS mainstream code repository.

References

Anderson, J.A., 2008. ISIS camera model design. In: 39th Lunar and Planetary Science Conference, March 10–14, League City, Texas (abstract no. 1391).

Anderson, J.A., Sides, S.C., Soltesz, D.L., Sucharski, T.L., Becker, K.J., 2004. Modernization of the integrated software for imagers and spectrometers. In: 35th Lunar and Planetary Science Conference, March 15–19, League City, Texas (abstract no. 2039).

Archinal, B.A., Lee, E.M., Kirk, R.L., Duxbury, T.C., Sucharski, R.M., Cook, D.A., Barrett, J.M., 2004. A new Mars digital image model (MDIM 2.1) control network. In: XXth Congress of the International Society of Photogrammetry and Remote Sensing, Commission IV, WG IV/9.

Chartrand, G., 1985. The Königsberg bridge problem: an introduction to Eulerian graphs. In: Introductory Graph Theory. Dover, New York, pp. 51–66 (Chapter 3.1).

Gaddis, L., Anderson, J., Becker, K., Becker, T., Cook, D., Edwards, K., Eliason, E., Hare, T., Kieffer, H., Lee, E.M., Mathews, J., Soderblom, L., Sucharski, T., Torson, J., McEwen, A., Robinson, M., 1997. An overview of the Integrated Software for Imaging Spectrometers (ISIS). In: 28th Annual Lunar and Planetary Science Conference, March 17–21, Houston, Texas (abstract no. 1226).

Greeley, R., Batson, R.M., 1990. Planetary mapping. In: Greeley, R., Batson, R.M. (Eds.), Cambridge Planetary Science Series, vol. 6. Cambridge University Press, Cambridge, UK.

Hare, T.M., Kirk, R.L., Skinner, J.A., Tanaka, K.L., 2009. Extraterrestrial GIS. In: Marguerite, Madden (Ed.), Manual of GIS. American Society for Photogrammetry and Remote Sensing, pp. 1199–1221 (Chapter 60).

Malin, M.C., Bell, J.F., Cantor, B.A., Caplinger, M.A., Calvin, W.M., Clancy, R.T., Edgett, K.S., Edwards, L., Haberle, R.M., James, P.B., Lee, S.W., Ravine, M.A., Thomas, P.C., Wolff, M.J., 2007. Context camera investigation on board the Mars Reconnaissance Orbiter. Journal of Geophysical Research (Planets) 112, E05S04, doi:10.1029/2006JE002808.

Morris, E.C., Tanaka, K.L., 1994. Geologic maps of the Olympus Mons region of Mars, 1:2,000,000. US Geological Survey Miscellaneous Investigations Series, Map I-2327.

Neteler, M., Beaudette, D., Cavallini, P., Lami, L., Cepicky, J., 2008. GRASS GIS. In: Hall, G.B., Leahy, M.G. (Eds.), Open Source Approaches in Spatial Data Handling, Advances in Geographic Information Science, vol. 2. Springer, Berlin, Heidelberg, pp. 171–199 (Chapter 9).

Neteler, M., Mitasova, H., 2008. Open Source GIS: A GRASS GIS Approach., 3rd Ed. Springer, New York.

Seidelmann, P., Abalakin, V., Bursa, M., Davies, M., De Bergh, C., Lieske, J., Oberst, J., Simon, J., Standish, E., Stooke, P., Thomas, P., 2002. Report of the IAU/IAG working group on cartographic coordinates and rotational elements of the planets and satellites: 2000. Celestial Mechanics and Dynamical Astronomy 82, 83–111.

Smith, D.G., Neumann, R.E., Arvidson, E.A., Guinness, E., Slavney, S., 2002. Mars Global Surveyor Laser Altimeter Mission Experiment Gridded Data Record. NASA Planetary Data System, MGS-M-MOLA-5-MEGDR-L3-V2.0.

Stallman, R.M., 1990. The GNU manifesto. In: Ermann, M.D., William, M.B., Gutierrez, C. (Eds.), Computers, Ethics, and Society. Oxford University Press, Inc., New York, NY, USA, pp. 308–317.

Torson, J.M., Becker, K.J., 1997. ISIS—a software architecture for processing planetary images. In: 28th Annual Lunar and Planetary Science Conference, March 17–21, Houston, Texas (abstract no. 1219).

Van Wie, P., Fischel, D., Howell, D., 1980. A transportable executive system for use with remote sensing applications software. In: International Symposium on Machine Processing of Remotely Sensed Data, Purdue University, West Lafayette, Indiana.

Warmerdam, F., 2008. The geospatial data abstraction library. In: Hall, G.B., Leahy, M.G. (Eds.), Open Source Approaches in Spatial Data Handling. Springer, New York, pp. 87–104 (Chapter 5).

Westervelt, J., 2004. GRASS roots. In: Proceedings of the FOSS/GRASS Users Conference, Bangkok, Thailand, pp. 12–14.