

Theia: Multispectral Image Analysis and Archaeological Survey

Vito Roberto^{1,2} and Massimiliano Hofer¹

¹ Dipartimento di Matematica e Informatica

² Norbert Wiener Center

University of Udine, Italy

{vito.roberto,massimiliano.hofer}@dimi.uniud.it

Abstract. Theia is a software framework for multispectral image analysis. The design is grounded on the object-oriented approach and a model combining parallel computation with selective data processing. Multispectral images from the archaeological site of Aquileia, Italy, have been used as the experimental testbed in order to assess the effectiveness and performance of the system; satisfactory results are reported, and are quite promising towards the use of the framework as a dynamic, interactive interface to real-time data exploration and processing.

Keywords: Multispectral, Hyperspectral, Image Processing, Interactive, Visualization, Object Oriented Design, Cultural Heritage, Archaeological survey, Remote sensing.

1 Introduction

The analysis of multispectral images is a central issue in a number of research and management tasks, such as environmental planning, medical diagnosis, archaeological survey and surveillance for both military and civilian applications. Problems arise from the data acquisition by heterogeneous sensory systems; the massive data sets to be handled; the need for efficient algorithms for data visualization, filtering and mining.

As a consequence, a number of challenging topics are to be faced by the designers. Proprietary frameworks are available like ENVI® [2], or freeware like ISIS [3], which are the outcome of long-term research projects, and so are mature enough to address a large number of applications. More recent developments include Opticks [4] and Next ESA SAR Toolbox (NEST) [5].

However, most computers now process gigabytes of data in a second and load large data sets in the RAM; uniprocessor systems are being rapidly replaced by CMP, SMT and SMP architectures even in low-end hardware. Novel software systems should be put into operation to take advantage from this state of affairs [10]: new solutions can be explored towards real-time, dynamic processing of huge amounts of information, to be driven by fully interactive user interfaces.

On the other hand, the fast growth and differentiation of applications suggest to adopt well-posed criteria of software design – inspired by modularity

and openness – enabling the developer to re-use or embed new modules, and customize the core system to domain-specific applications.

Along these lines we designed and realized Theia, a software framework for image exploration and analysis, and the results are reported in the present paper. The next section introduces the system architecture, with details on the organization of modules and data flow; Sections 3 and 4 report two steps of the analysis of multispectral images taken for archaeological survey on the Aquileia area (Italy); Section 5 contains our conclusions.

2 Theia: Software Architecture and Implementation

The goals of the project are: Object-Oriented (O-O) design; portability among OS platforms; efficient use of parallel architectures; real-time processing of massive data sets; interactivity of the user interface; extendability with customized components.

Two architectural schemes are reported in Figure 1. Three functional blocks have been defined: data management; data processing; graphic user interfacing (GUI). The first includes data encoding and storing; the second, numerical processing and graphic rendering; the GUI manages the flow between user, events, components and processes. The accurate O-O design and interface definitions ensure that each block can be developed independently: for example, the GUI management block can be replaced by a scripting engine to drive the whole application.

In order to achieve efficiency, we adopted a computational model that combines selective (attentional) with parallel processing techniques; a scheme is reported in Figure 2. The management unit in the GUI performs the following steps: ▶ Notifies to update an image region; ▶ The visualization filter pre-processes data, if needed; ▶ The region is partitioned into tiles; ▶ Multiple threads are started, depending on the number of available CPUs; the tiles are inserted

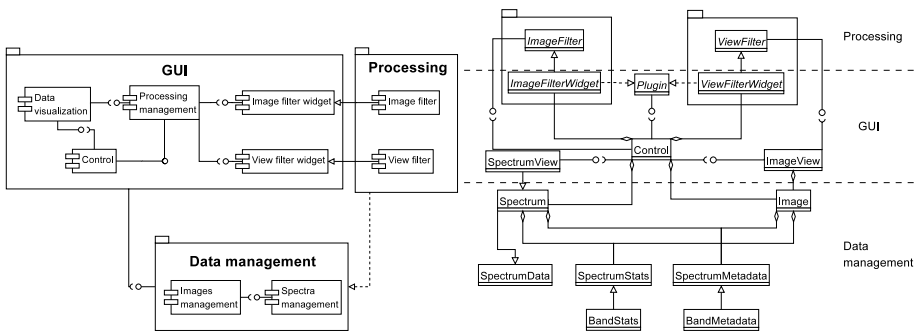


Fig. 1. General architecture of Theia, in UML 2 notation [12]. *Left:* Components organized into three functional blocks, and communications channelled through a few, well-defined interfaces. *Right:* Class diagram, with the same organization outlined.

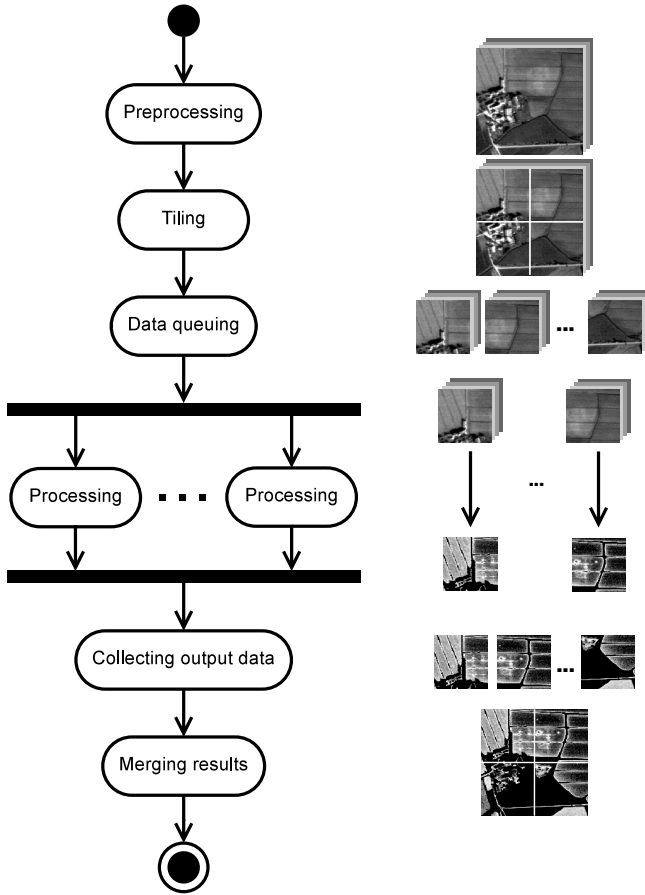


Fig. 2. UML 2 activity diagram of the parallel computations triggered by a screen update

into the input queue; ▶ Each thread pulls the tiles, processes them and pushes the results on an output queue, until the input is empty; ▶ The output is collected and dispatched to the GUI.

Theia addresses two basic tasks, image visualization and processing. The *Filter* is a central concept, mapped onto two modules: the *ViewFilter* and the *ImageFilter*, charged of visualization and numerical processing, respectively. They consist of three abstract classes: *Plugin*, exchanging parameters with the GUI; *Processor*, triggering the computations and providing results; *Widget*, customizing the GUI to a specified filter.

Theia has been implemented on the LinuxTM OS (openSUSE 11.0, kernel 2.6.28) for x86-64 CPUs; the C++ language has been preferred to JavaTM because of its higher efficiency in processing massive data sets; in addition, it allows a more effective use of the O-O and generic paradigms, by offering solutions such as multiple inheritance, virtual base classes and templates.

The infrastructure – not shown in Figure 1 – provides basic operations and support for common data formats (ENVI-BSQ [13], multichannel TIFF [7], PNG [14]). Standard transmission protocols like Web Map Service (WMS) [15] ensure the interconnection with network data sources and systems.

The system employs two cross-platform libraries: QT4 [1], v. 4.4.3, a framework for GUI development, distributed under GPL v. 2.0 or 3.0; and LibTIFF [6], v. 3.8.2 under the X11/MIT license.

Theia acts as an expandable core of software operators. The GUI is capable of handling plugins: event management, mouse ownership, drag-and-drop of basic data, switching between components and driving their placement and visualization. A new filter can be developed without modifications to the main program, and linked at run-time from a separate library; a minimal implementation requires just one class and one method.

All basic components to process images and show data in the GUI are fully re-usable by a plugin element: spectrum visualization components, for example, are used in different panels, as shown in Figures 4, 5.

Portability is another issue addressed in the project. Theia is coded in C++, which implies that portability between different OS and CPU architectures is to be addressed explicitly. For this sake, the external APIs are limited to cross-platform libraries only, and all architecture-dependent tasks – such as data encoding – are addressed within encapsulated common libraries.

3 Model-Based Orthoprojection

The multispectral images were acquired by the MIVIS airborne sensor system (Multispectral Infrared and Visible Imaging Spectrometer [16]). The data is subject to a systematic distortion due to the mechanical structure of the sensor system; a sketch is reported in Figure 3. A rotating mirror conveys radiation from the terrain into a detector. By taking repeated measurements at fixed time intervals, a strip of data of the terrain is acquired. The whole multispectral image is collected by an aircraft. In order to correct the distortions of this kind of sensor, we developed a mathematical model and designed a specific algorithm.

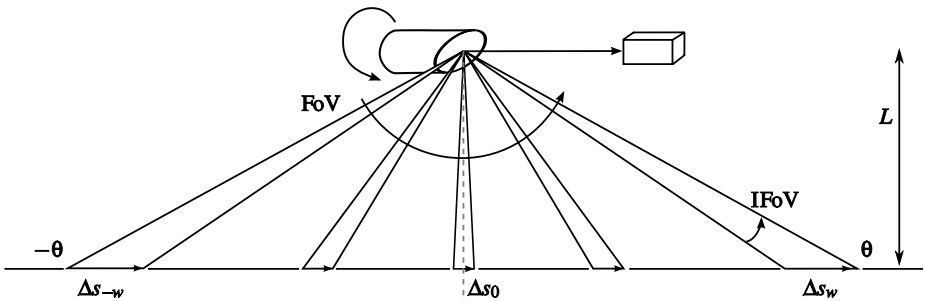


Fig. 3. Model scheme of the MIVIS airborne sensor system

We define *Field of View* (FoV) the angular width of the acquired data and *Instantaneous Field of View* (IFoV) the angular width of a single measure.

Let $\Delta s_i, i \in [-w, w] \cap \mathbb{N}$ be the $2w + 1$ samples collected at regular angular distances in the range $[-\theta, \theta]$ at an altitude L . The model in Figure 3 shows that the maximum resolution – expressed in $\frac{m}{\text{pixel}}$ – is reached in Δs_0 .

The minimum and maximum angles corresponding to Δs_i are:

$$\left(\frac{2\theta(i+w)}{2w+1} - \theta, \frac{2\theta(i+1+w)}{2w+1} - \theta \right) \quad (1)$$

The minimum and maximum coordinates corresponding to Δs_i , then, are:

$$\left(L \tan \left(\frac{2\theta(i+w)}{2w+1} - \theta \right), L \tan \left(\frac{2\theta(i+1+w)}{2w+1} - \theta \right) \right) \quad (2)$$

We call *orthoprojection* the mapping between the vector of input values $\Delta \bar{s}$ and a new vector with components $\Delta o_j, j \in \left[-\frac{\tan \theta}{\theta} w, \frac{\tan \theta}{\theta} w \right] \cap \mathbb{N}$ of elements having the same resolution as Δs_0 . The mapping is defined as follows:

$$\Delta o_j = \Delta s_{k_j} \text{ with } k_j = \left\lfloor \frac{w}{\theta} \tan^{-1} \left(\frac{\theta}{w} j \right) \right\rfloor \quad (3)$$

In order to check the validity of the model, we used four multispectral images acquired by MIVIS at the altitude $L = 1500\text{m}$. We compared them with georeferenced orthophotos with known precision and identified a set of common reference points (16 for each MIVIS image). We then computed the optimal least-square coefficients of a linear transform between the two coordinate systems.

The result provides us with georeferenced frames with a resolution between 2.45 and $3.70 \frac{m}{\text{pixel}}$, depending on the distance from the central axis, with standard deviation 8.61 m.

4 An Application to Archaeological Survey

Our framework has been tested on a multispectral image of Aquileia, an ancient roman city in the North-eastern Italy area.

A number of tasks have been implemented: ► Visualization and rendering: multispectral-to-RGB image mappings; real-time rendering of animated sequences. ► Processing: multispectral-to-multispectral mappings and filterings; band selection; pixel classifications and segmentation techniques [9]; model-based geometric transforms (othoprojection).

The tasks were carried out on a notebook equipped with 3 GB RAM and a dual core CPU with a 2 GHz clock. Each multispectral image consists of frames acquired in 102 wavelength bands between 433 and 12700 nanometers, with 755×4000 pixel resolution, 16-bit sample precision, and a load of 587 MB per frame.

The design solutions allowed us to explore and analyze data interactively. Frames were analysed by estimating distances with respect to reference spectra using appropriate measures [11]. Two such measures have been reported in

Table 1. Two spectral distance measures

Spectral angle	$\frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$
Spectral information divergence	$\sum_{i=1}^n x_i \ln \frac{x_i}{y_i} - \sum_{i=1}^n y_i \ln \frac{y_i}{x_i}$

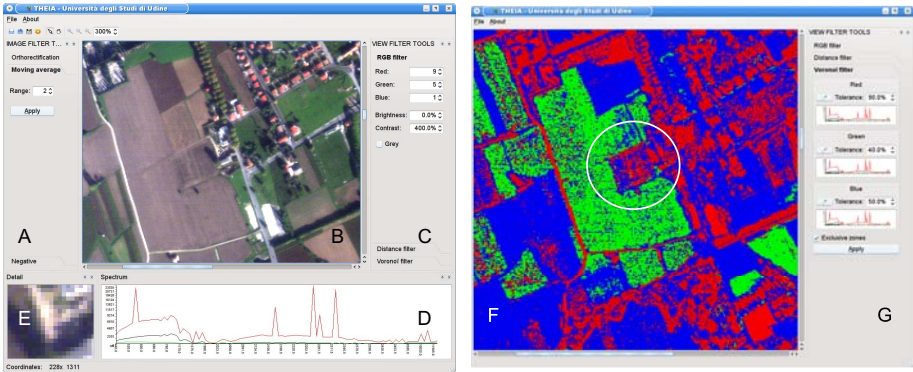


Fig. 4. Two screenshots of Theia, after processing frames of the same area of the roman city of Aquileia. *Left:* Moving average filtering in the visible energy band. Five panels at work: A) Processing filter parameters ; B) Output color frame; C) View filter parameters; D) Spectral data of a selected pixel: besides the true spectrum, two additional curves report the minimum and maximum values over all bands; horizontal scales are in nanometers; E) Magnified frame region surrounding the pixel (228, 1311). *Right.* The GUI has been reconfigured. Segmentation of the same frame into Voronoi regions; color rendering of stone (red) and two kinds of cultivated terrains (green, blue). F) Output frame, showing buried structures of a building (encircled), not distinguishable on the left frame; G) View filter configuration panel, reporting the reference spectral data.

Table 1. An advantage is that no calibration data are needed when extracting reference spectra from known areas within the image. Dozens of spectra could be tried in a few minutes to search for relevant patterns within the flood of data.

Interesting results have been obtained using the spectral angle distance [11] to highlight areas with strong matching with known archaeological findings. Novel evidences of buried structures were found in this way, and confirmed by expert archaeologists: two results have been reported in Figures 4, 5.

In all tests Theia exhibited a remarkable time efficiency: with a visible portion of 755×768 pixels, most visualization filters require a few tenths of a second for a



Fig. 5. *Left.* Orthorectified frame in the visible band from a different area of Aquileia. *Right.* The same area. Spectral angle distance filtering of frames taken in the near infrared. The reference spectral data (panel on the right) are from a known building. A buried structure is apparent (circled), not distinguishable in the visible band.

complete refresh; much less for the partial renderings needed while panning. The most CPU-intensive filters – e.g., false color segmentation – need at most 1.5 seconds for the first rendering, without activating multithreaded computations.

A measure in the case of a view filter with spectral angle distance, averaged over hundreds of proofs, yields the following results:

	1 thread	2 threads
CPU usage rate	0.98–1	0.72–0.75
Frames per second	2.6	4.3

This means that we reach 83% of the theoretical maximum efficiency.

5 Conclusions and Perspectives

We have designed, implemented and tested Theia, a software system devoted to the analysis of multispectral images. The results of our research work demonstrate that the goals listed in Section 2 are within reach.

The O-O software design provides effective solutions to address the project concerns: encapsulation allows for a clear-cut definition of modules at different granularity: classes, components, blocks; this yields a readable and reusable code. Portability is ensured by a careful implementation; porting on Windows 2000 and XP for i386 has been accomplished with little effort. We also verified that adding filters to the system or linking modules from separate libraries, are both readily affordable tasks.

Computational efficiency has been achieved by adopting a model which exploits parallelism and selective processing techniques. Data processing modules have been kept physically separate from the other ones throughout the system. In addition, features of the C++ language have been exploited to optimize performance, such as template specialization and inline method expansion for basic data types. As a consequence, satisfactory performance has been obtained towards interactive visualization and real-time data processing.

Theia is an open system under various respects: it can be extended by adding new components (e.g., image filters), in order to address specialized applications; can be integrated into a distributed network environment, so to exchange data through standard protocols with external sources, services and systems.

Realizing a system comparable with well-established software products was not the aim of our project. A considerable amount of work is needed for Theia to reach an adequate maturity with respect to the wide range of applications of multispectral analysis. Rather, after further systematic tests, Theia is expected to remain a light, open framework for exploring images and data, as well as developing specialized applications.

References

1. Qt, <http://trolltech.com/products/qt>
2. ENVI ITT Visual Information Solutions, <http://www.ittvis.com/envi/index.asp>
3. ISIS US Geological Survey, <http://isis.astrogeology.usgs.gov/>
4. Opticks Ball Aerospace & Technologies Corp., <https://opticks.ballforge.net/>
5. NEST Array Systems Computing Inc. under ESA Contract, <http://www.array.ca/nest/>
6. LibTIFF, <http://www.libtiff.org/>
7. Adobe Developers Association, TIFF Revision 6.0. (1992), <http://partners.adobe.com/public/developer/en/tiff/TIFF6.pdf>
8. Perrizo, W., Ding, Q., Ding, Q., Roy, A.: On mining satellite and other remotely sensed images. In: Proceedings of ACM SIGMOD Worksh. on Research Issues in Data Mining and Knowledge Discovery, pp. 33–40 (2001)
9. Landgrebe, D.: Some fundamentals and methods for hyperspectral image data analysis. In: Systems and Technologies for Clinical Diagnostics and Drug Discovery II. School of Electrical & Computer Engineering Purdue University, vol. 3603. SPIE (1999)
10. El-Ghazawi, T., Kaewpijit, S., Le Moigne, J.: Parallel and Adaptive Reduction of Hyperspectral Data to Intrinsic Dimensionality. In: Proceedings of the IEEE International Conference on Cluster Computing. The IEEE Computer Society, Los Alamitos (2001)
11. Sweet, J.: The spectral similarity scale and its application to the classification of hyperspectral remote sensing data. In: IEEE Worksh. on Advances in Techniques for Analysis of Remotely Sensed Data, pp. 92–99 (2003)
12. Eriksson, H.E., Penker, M., Lyons, B., Fado, D.: UML2TM Toolkit. Wiley Publishing, Inc., Chichester (2004)
13. Frew, J.: Difficulties of dealing with large image cubes. In: Proceedings of AIAA Aerospace Sciences Meeting (1991)

14. Boutell, T., et al.: PNG (Portable Network Graphics) Specification Version 1.0. RFC 2083 (1997)
15. OpenGIS® Web Map Service (WMS) Implementation Specification. v. 1.3.0 Open Geospatial Consortium Inc. (2006)
16. Vagni, F.: Survey of hyperspectral and multispectral imaging technologies. Technical Report TR-SET-065-P3 NATO Research and Technology Organization (2007)
17. Boccardo, P., Mondino, E.B., Gomarasca, M.A., Perotti, L.: Orthoprojection tests of hyperspectral data in steep slope zones. In: ISPRS 2004, p. 872 (2004)