

# Monteverdi 2.0 - Remote sensing software for Pleiades images analysis

Manuel Grizonnet<sup>1</sup>, Julien Michel<sup>2</sup>

<sup>1</sup>*CNES, Toulouse, France; manuel.grizonnet@cnes.fr*

<sup>2</sup>*CNES, Toulouse, France; julien.michel@cnes.fr*

**Abstract.** The development of the Orfeo ToolBox has been strongly related to the ORFEO preparatory program to Pleiades, and to the RTU program, its in-flight sequel. The interactions with the members of these programs guided the roadmap of functionalities to be added, and helped to validate the implemented algorithms. Since these users have a strong need for ready to use tools, the Orfeo ToolBox development teams have worked on several solutions to provide such tools. From 2006 until today, this paper will present the evolution of these tools from the first application package to Monteverdi and the new application package. It will introduce Monteverdi 2.0, the future solution currently in development. The main functionalities currently available in the beta version as well as the foreseen ones will be presented, as well as the user driven development process that has been adopted on this project.

**Keywords.** Remote sensing software, open-source, OTB, applications, image processing, Pléiades, CNES, ORFEO program, image processing.

## 1. Introduction

The Orfeo Toolbox (OTB) is an open-source remote sensing Image Processing software developed by CNES since 2006. It gathers a large set of state-of-the-art algorithms into a common framework allowing to easily build complex processing chains. It has been developed in the frame of the ORFEO program [1], which aimed at preparing the use of Pleiades imagery by the French and European institutional users for their applications. As such, the evolution of the Orfeo ToolBox and its derived tools is strongly tied to the ORFEO program before Pleiades 1A launch on December 2011 and to the Pleiades HR thematic commissioning program after the launch. This paper intends to tell the story of how those tools have evolved to match the expectations of these programs. It is organized as follows: section 1 will introduce the early ages of the Orfeo ToolBox applications, section 2 will describe how Monteverdi has been designed and used, section 3 will present the evolution of the OTB applications, and section 4 will explain how the idea of Monteverdi 2.0 has emerged, and what the current stage of its development.

## 2. The old OTB-Applications package (2007 – 2009)

The Orfeo ToolBox [2] has always been first and foremost a C++ library. It offers a great deal of state-of-the-art algorithms with an open-source implementation [3]. In order to prepare for Pleiades images, whose standard product is a pan-sharpened image of 40 000 by 40 000 pixels large, with 4 spectral bands (but larger mosaic can be ordered), whenever it is possible, these implementations are compatible with piecewise processing (called streaming) and parallelization (also called multi-threading). The library covers a wide range of functionalities, such as geometric and radiometric pre-processing, feature extraction, supervised classification, change detection, segmentation... And most of these functionalities have been pitched by workgroups and members of the ORFEO program. However, even if all new functionalities came with a documentation of its C++

API and documented code examples, one needed to write and compile their own C++ code to use them, which was a problem for some (most ?) users. Remember that at this time there was no binary distribution of the Orfeo ToolBox, so the users needed to achieve the compilation of both the Orfeo ToolBox and their own code.

This is why a new software package named OTB-Applications has been created in 2006. This pack-age contained a set of command-line tools, as well as some basic graphical tools. The graphical tools based on the FLTK toolkit aimed at providing more complex methodology with user supervision for instance, or an easier way to play with the parameters. All these applications were developed for the purpose of validation by the users, not for operational use. As such, their design and interfaces were inconsistent from one application to the other, and they could barely be used for actual remote sensing tasks. At this time, the development team focused on building a strong basis for the core of the Orfeo ToolBox, and these applications were only there to demonstrate what could be achieved with a few lines of OTB code.

### 3. Monteverdi (2009 - 2013)

Everything changed in 2009. The Capacity Building program led by the CNES Department of Strategy and Programs funded the development of end-users-oriented software on top of the Orfeo ToolBox, named after the composer of the Orfeo opera, Monteverdi [4]. This new software provided a simple graphical interface to visualize and dynamically chain most of the processing offered by the OTB in a user-friendly environment.

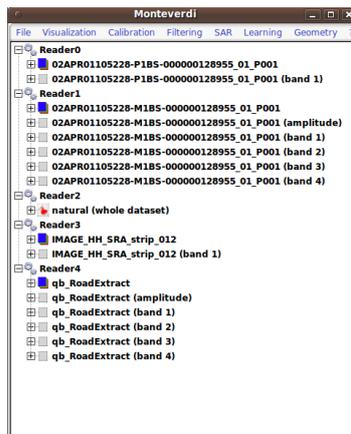


Figure 1. Monteverdi main window

Figure 1 shows Monteverdi main window. One can see the different available menus as well as different modules which have been set up for the processing. The reader modules allow loading input data. When opening a new module, the input data can be selected among the output data of all already created module instances, thus building a processing pipeline sequentially. Output(s) of modules are automatically added to the main window tree. The module and its parameters are registered and at any moment the user can decide to trigger the execution of the process or not.

While initially designed for training courses and capacity buildings activities, Monteverdi usage grew far beyond this initial scope and met a great success for OTB users in the ORFEO community, encouraging new non-developing users to gain interest in OTB. During the 2010 to 2012 period, Monteverdi became one of the main platforms for integration of OTB functionalities and a lot of modules have been developed, including most of the old OTB-Applications which have been integrated to Monteverdi.



Figure 2. Texture computation in the feature extraction module

The user feedbacks were very positive and led to an increasing number of available functionalities in Monteverdi. In 2011, Monteverdi became “Pleiades ready”, with an evolution of the viewer module to take advantage of JPEG2000 multi-resolution capabilities, as well as a dedicated module to uncompress a selection of JPEG2000 tiles, shown in Figure 2. This compatibility with Pleiades images from the very start of the mission even increased to use of Monteverdi in the ORFEO community.

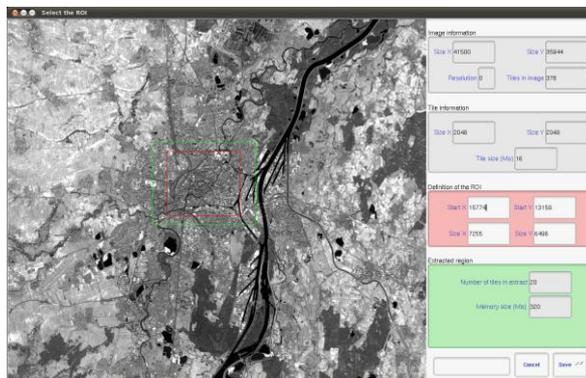


Figure 3. JPEG2000 decoding module

The release of Monteverdi at the end of year 2009 changed a lot of things as most of non-developer users, who were looking for quite a long time for an application providing Orfeo TooBox functionalities under a unified graphical interface. As a consequence, Monteverdi continued to grow in the past three years and it has now more than 40 modules available. However, if Monteverdi was indeed a great tool to perform numerous remote sensing and image processing tasks provided by OTB in a minute, it was not well suited for heavier and longer processing, scripting and batch processing. The implicit processing pipeline that the users can build by chaining modules can become very heavy to update, resulting in freezes or lags, a big issue with Pleiades images which can be very large. Last, the ergonomics could be better, with a lot of mouse operations needed to perform some actions and no history or undo functions.

#### 4. The OTB applications revamped (2012 - )

After the first release of Monteverdi, the old OTB-Applications package has been left aside for a while: end-users functions were developed directly as Monteverdi modules, and all the old applications proven to be stable and useful enough have been ported to Monteverdi as well. However, at some point the experience feedback from Monteverdi users highlighted the need for high level remote sensing functions that can handle long processing time for large data, batch processing and remote execution. The applications package has therefore been revamped to a set of command-line only tools that can be used for such high level tasks, like orthorectification, radiometric calibration, pan-sharpening, classification, large-scale segmentation, 3D reconstruction ... At this point, processing real large data with Orfeo ToolBox became accessible to all OTB users.

Later in 2011, the applications package has been further reworked. The reason for this is that these applications were merely a collection of independent simple tools with only the command-line argument parser in common. As such, they do not offer a common interface and are thus can barely inter-operate with other systems. Moreover, there were a lot of functions duplicated between Monteverdi and the applications.

To overcome these issues, a new model has been developed for OTB applications. It consists in an API to create a new application from a virtual application class. From an application author perspective, this API allows to declare all the parameters as well as the documentation, and to write the embedded processing chain in a dedicated method. From a user perspective, any application can be dynamically instantiated through a shared library plugin system. The instantiated application can then be inquired about documentation and parameters, those parameters can be set and the user can trigger the application execution.

The Orfeo ToolBox also comes with standard application “players”, which allows to run any application from a given user interface. Current players include a command-line player, a graphical player with an auto-generated interface to set parameters and see the documentation, and a Python API through SWIG.

The general philosophy behind this new application framework is to have to write the processing only once, and then use it from whatever environment, programming language or software you want through one of these access points. This capacity has been used in several different contexts:

- A simple python script allows to generate documentation for all the applications [5],
- A simple python script allows to generate applications file descriptors for integration in external tools like the KEO system from ESA,
- A simple python script allows to perform batched processing chaining several applications,
- Qt GUIs are really nice to use, with a lot of useful features like automatic parameters estimation (for instance, the orthorectification application provides you with the best-fit parameters for your image),
- The python interface allows nice integration within QGIS.

#### 5. Monteverdi 2.0 (2013 - )

##### 5.1. Why Monteverdi 2.0?

Monteverdi success went far beyond its initial scope since it opened the OTB world to a wide range of users who needed a ready to use graphical tool more than a library of components to write their own processing chains. With these four years of lifetime, we have gathered a lot of feedbacks regarding how useful the tool was, but also regarding what should be improved to move toward greater usability especially in operational contexts.

Despite its great success, Monteverdi has not been designed for the desktop all-purpose re-mote sensing image processing tasks. For instance it lacks of helpers functions like project saving or task cancellation, and is not really suitable for heavy processing of large remote sensing images. On the other hand, OTB functionalities wrapped up as OTB applications offer these real processing capabilities with a unified access but are not compatible with the architecture of Monteverdi.

We therefore decided to rework the Monteverdi concept into a new software, and in late 2012, the development of the 2.0 version of Monteverdi has been initiated. The aim of this new version is to provide a tool tailored for desktop remote sensing image processing, with a state-of-the-art image navigation system, very useful for large Pleiades images, a better user experience, and a seamless access to the OTB applications which will be the main and sole processing engine of this new Monteverdi. We also plan to offer in the longer term nice functionalities like a remote sensing persistent data manager, collaborative plugins, or tools for rich data annotations and tagging ...

### *5.2. User driven development*

Monteverdi 2.0 is still at an early development stage but already offers a large part of Monteverdi functionalities in terms of image manipulation and already overcomes the previous version in terms of processing capabilities. In order to avoid developing useless functionalities and converge faster toward Pléiades users real needs, the OTB team decided to conduct the development of the application as an iterative process with short iterations leading to the release of a new experimental version. Monteverdi 2.0 users are invited to give feedbacks at the end of each iteration about the software through the OTB mailing list and also through dedicated surveys where they can evaluate existing functionalities and ask for future improvement.

### *5.3. User interface*

The first point we wanted to improve in Monteverdi was the general interface of the software. In Monteverdi, the viewer module was integrated as one processing module among others. While making the interface quite generic, it was requiring a lot of clicks to open and navigate in images and was also leading to the dissemination of incomplete and incoherent visualization functionalities in a lot of other modules. Monteverdi 2.0 is designed as a user-centered interface built around the image navigation and manipulation, surrounded by tools for dataset manipulation, rendering and processing.

We think that this layout offers a more modern way to interact and get information from images, and decreases drastically the number of operations needed to open and manipulate images. As presented in figure 4, the main window is divided in 5 parts:

- Upper left: the dataset browser which allows to select and navigate about images loaded (1),
- Upper right: the zoom window linked to the visualization which to scroll inside the image (2),
- Center: the visualization zone (3),
- Lower right: different tabs are available to set color dynamics and rendering methods (4),
- Lower left: access to OTB processing through OTB-Applications (5).



Figure 4. Monteverdi 2 main window

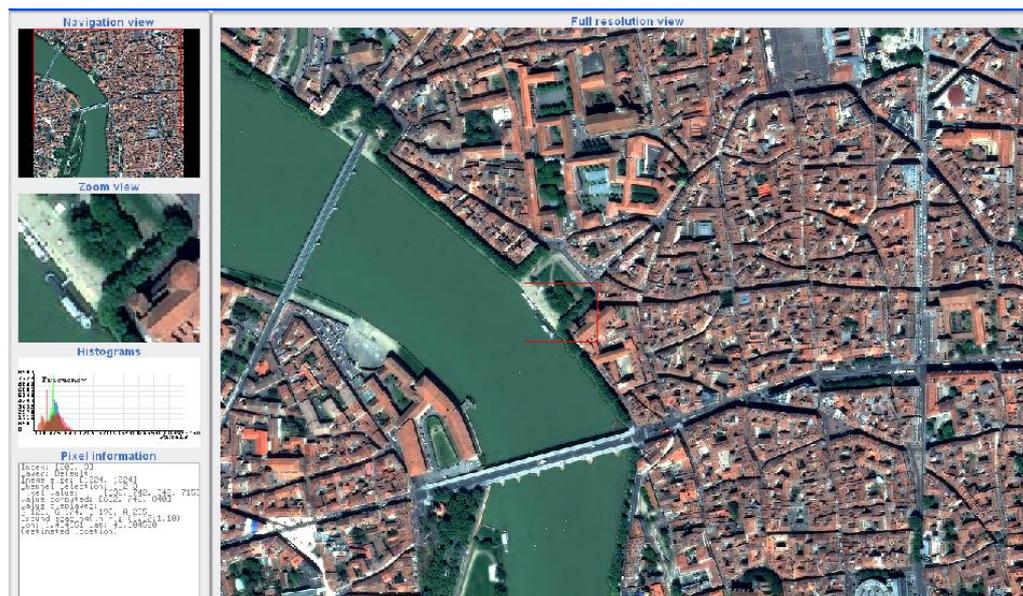
#### 5.4. Image rendering and navigation

The way people access and navigate into earth observation data evolved a lot in the last decade with the development of services like Google Earth or Géoportail for instance. Those services offer an easy and intuitive navigation and search through remote sensing images and are now used on a daily basis by people. On the other hand, most software dedicated to remote sensing image processing like Monteverdi remain highly technical with specific visualization tools which may appear outdated compared to state-of-the-art geospatial web interfaces.

In the case of Monteverdi, the visualization module was composed of 3 windows, as shown in figure 5:

- The scroll window allowed to navigate in the quick-look (like in Monteverdi 2.0),
- The full resolution window allowed to navigate in the finest image resolution only,
- The zoom window allowed to navigate through finer resolution on a subset of the full resolution region rendered.
- The 3 windows were linked together, allowing for navigation in large dataset, but with fixed scale and without drag and zoom support for instance.

Monteverdi 2.0 image manipulation is much more intuitive and allows to navigate through space and resolution using mouse dragging and mouse wheel zooming. Enabling this type of navigation in very large images within a desktop application is challenging, and requires to use JPEG 2000 multi-resolution capabilities for Pléiades imagery or offline generation of GDAL overviews (containing a multi-resolution pyramid) for generic large image files.



**Figure 5.** Monteverdi viewer module

### 5.5. Data loading and persistence

In most cases, working on a remote sensing dataset requires to display it several times during the analysis. Therefore, in Monteverdi 2.0 there is a clear need for saving as much information as possible to speed up image opening next time, which is also known as persistence. To do so, the application saves all settings over all datasets which have been loaded in the application: multi-resolution pyramid, quick-look, rendering settings, current displayed area and scale ... This allows to reuse, cancel or reopen the last settings for all datasets once opened in Monteverdi 2.0. All this information as well as the list of datasets are stored in a dedicated directory handled by Monteverdi 2.0 behind the scene. Next time Monteverdi 2.0 will open, all datasets will show up and be reachable in one mouse click.

### 5.6. Access to processing with OTB applications

One of the main objective of Monteverdi 2.0 is to stop writing dedicated processing modules like in Monteverdi and giving access to the remote sensing processing available in the ORFEO ToolBox instead. OTB applications are integrated in the software in a side panel where the user is able to search through the list of available applications. After the application has been selected, parameters can be set in a dedicated tab at the center of the main application, as shown in figure 6.

Input datasets can be dragged and dropped from the dataset browser, and once the parameters are set, the processing of the full dataset runs as a background task. Results are then automatically re-opened and loaded as new images in the dataset browser. Figure 7 shows the result of computing NDVI radiometric index using the band math application on a Pléiades multispectral image.

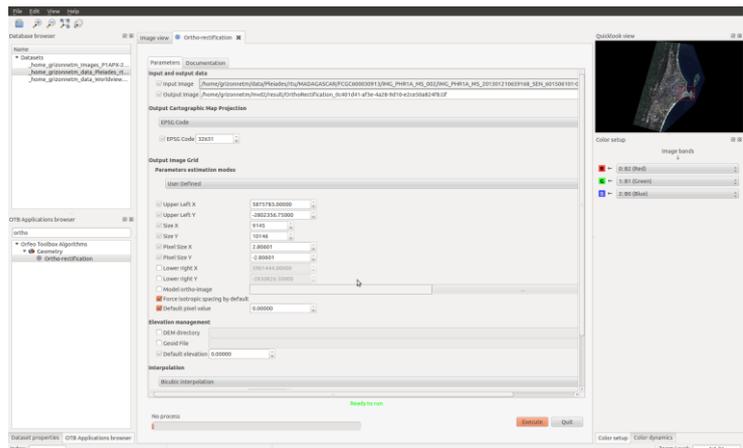


Figure 6 . OTB applications panels in Monteverdi 2.0

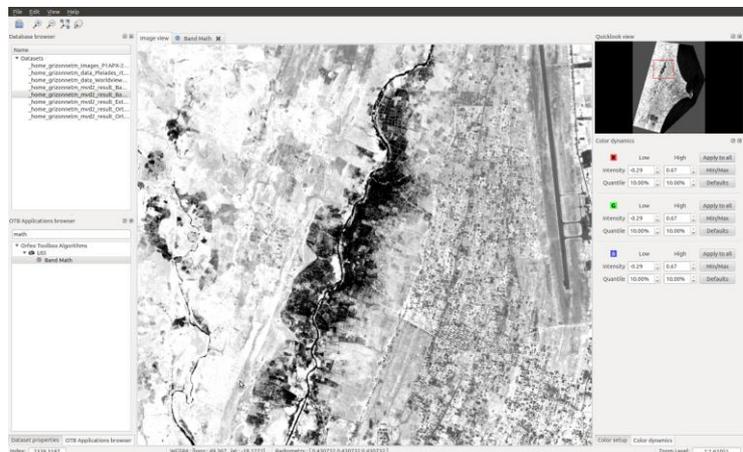


Figure 7. NDVI resulting from the application is automatically added to the datasets manager of Monteverdi 2.0

## 6. Perspectives: What’s next?

### 6.1. Adding more visualization modes and tools

Visualization and rendering methods will be improve in next versions to offer new functionalities for image rendering and histogram adaptation for contrast enhancement. Local contrast enhancement at the window or at the pixel level will be implemented, as well as more simple strategies like absolute rendering between arbitrary bounds. Of course, rendering of vector layers is also planned.

### 6.2. Customization and collaborative documentation

Being plugin based the OTB Applications framework allows Monteverdi users to load and access to their own applications in Monteverdi 2.0.

Another important point for such application is the availability of a user documentation which explains how to operate it. One key to the success of Monteverdi was the availability of a specific user documentation called the Cookbook [5] which presented a general introduction of the software but also provided recipes, i.e. detailed guided steps to accomplishing classical remote sensing tasks. We would like to build such documentation for Monteverdi 2.0 by taking advantage of new collabo-

rative platforms for documentation which are a good way to get non-technical newcomers participating to the software.

### 6.3. Image collection manager (like a photo manager),

Finally, we foresee that the way people use remote sensing images will change in the future with the increasing acquisition capacity of sensors, the arrival of new satellites constellation and the emergence of Open Data initiatives in which more and more contents will be free to use and easy to access.

The use of remote sensing images by end-users might therefore evolve from the manipulation of one or two images from time to time to the use of a more complete collection of data updated more frequently. We can expect that this evolution will also create different needs in terms of services and change some of the remote sensing applications requirements. For instance, such applications could manage large collections of images as in a photo manager, provide dedicated functionality to search and navigate inside large geographic databases, support annotations, bookmarks or tags... Since the philosophy of the Orfeo ToolBox (and Monteverdi) is also to experiment new things and ideas, we plan to implement some of these in future versions.

## 7. Conclusions

This article presents the history of the end-users tools based on the Orfeo ToolBox and developed for the need of the ORFEO program users from 2006 until today. It shows the evolution of the maturity of these tools and its relation to the evolution of user needs. Monteverdi has been a big step forward as it allowed to access to OTB functionalities for non developers. The arrival of Pléiades images stressed the need for dedicated tools to perform full imagery processing using OTB, which led to the creation of the new OTB applications.

Monteverdi 2.0 aims at providing both advantages of Monteverdi and of OTB applications, by offering an up-to-date software to visualize remote sensing images more easily and giving access to a large set of processing tools in a unified graphical interface. Note that Monteverdi 2.0 is still in an early development stage: a first experimental version has been released in March 2013, offering only the new navigation tools, and its release has been followed by user feedbacks gathering. The development team is currently working on the second experimental version which will provide the database manager and the access to the OTB applications as well. This second version should be released in June 2013. With the help of our users, we will continue to develop this tool and hope it will help digging into the richness of earth observation images information.

## References

- [1] Kubik, P., De Boissezon, H., Tinel, C., Michel, J., Grizonnet, M., 2012. *The Pleiades system: High resolution capability suited to users needs*. Revue Française de Photogrammétrie et de Télédétection, 200:28-35.
- [2] "The ORFEO toolbox software guide", <http://www.orfeo-toolbox.org>, 2013.
- [3] Christophe, E. and Inglada, J., "Open Source Remote Sensing: Increasing the Usability of Cutting-Edge Algorithms," in IEEE Geoscience and Remote Sensing Newsletter, pp 9-15, mar. 2009 .
- [4] Monteverdi - Remote sensing software from educational to operational context ; Manuel GRIZONNET and Jordi Inglada, Earsel Symposium 2010
- [5] "The ORFEO toolbox Cookbook : A guide for non-developers", <http://www.orfeo-toolbox.org>, 2013.

