Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

MITO: An advanced toolkit for medical imaging processing and visualization

Mario Ciampi, Luigi Gallo, Giuseppe De Pietro

RT-ICAR-NA-2018-02 **luglio 2018**

Consiglio Nazionale delle Ricerche, Istituto di Calcolo e Reti ad Alte Prestazioni (ICAR) – Sede di Napoli, Via P. Castellino 111, I-80131 Napoli, Tel: +39-0816139508, Fax: +39-0816139531, e-mail: napoli@icar.cnr.it, URL: *www.icar.cnr.it*

Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

MITO: An advanced toolkit for medical imaging processing and visualization

Mario Ciampi, Luigi Gallo, Giuseppe De Pietro

Rapporto Tecnico N: **RT-ICAR-NA-2018-02** ***Data:*** **luglio 2018**

MITO: An advanced toolkit for medical imaging processing and visualization

Mario Ciampi[1], Luigi Gallo[1], Giuseppe De Pietro[1]

[1]Institute for High Performance Computing and Networking of the National Research Council of Italy
Via Pietro Castellino, 111, 80131, Naples, Italy
{mario.ciampi, luigi.gallo, giuseppe.depietro}@icar.cnr.it

Abstract

This technical report presents MITO, an open software environment for medical image acquisition, processing, visualization, and navigation. The system is able to interact with PACS servers conforming to the DICOM 3.0 standard in order to retrieve and send radiological image data. Acquired 2D images can be manipulated with MITO for basic operations and advanced processing, like image segmentation, region extraction, image fusion, 3D reconstructions. Advanced 2D and 3D user interfaces allow users to interact with medical images and volumes through various input devices or in a completely touchless way. The high number of downloads of the software system, along with its widespread use in numerous experimental scenarios, show the high extensiveness and performance of the features developed.

Keywords: Medical Imaging Tool; Image Processing; Image Visualization; User Interaction; DICOM; PACS.

1. Introduction

Medical Imaging Toolkit (MITO) is an advanced medical image processing and visualization open software environment designed and developed since 2009 from the iHealth Laboratory of the Institute for High Performance Computing and Networking of the National Research Council of Italy (ICAR-CNR). The source code of the system has been released on SourceForge[1] under the terms of the GNU General Public License.

MITO was developed with the aim of providing interested users (medical researchers, software developers, physicians, etc.) with an open extensible desktop application able to process, visualize, and interact with medical images, which can be available on the user's host or fetched from a health information system conformed to the ACR/NEMA DICOM 3.0 standard, also known as Picture Archiving and Communicating System (PACS). More in detail, MITO is able to analyze different kind of imaging modalities, including CT (Computed Tomography), MRI (Magnetic Resonance Imaging), PET (Positron Emission Tomography), US (UltraSound), and CR (Computed Radiography).

The system provides the final user with advanced functionalities, like 2D-3D visualization, image segmentation and fusion, ROI (Region Of Interest) extraction, direct and indirect volume rendering reconstructions. Moreover, sophisticated interaction techniques allow users interacting with the rendered images both through touch-based and touchless ways.

This technical report presents the system architecture components, depicted using UML artifacts, and the main features and user interaction methods, whose characteristics are shown in representative pictures. MITO has been extensively experimented worldwide by physicians, researchers, and software developers. Comparative analyses conducted in literature attest thewide spread of the system due mainly to the extensiveness of the features developed.

[1] https://sourceforge.net/projects/mito/

The rest of the technical report is organized as follows. Section 2 provides a general overview of the software architecture of MITO, describing every building blocks and the interactions among them. Section 3 presents the functionalities of MITO implemented in the building blocks of the system. The graphical user interfaces are illustrated in Section 4. Section 5 outlines some implementation details. Finally, Section 6 concludes the technical report, emphasizing the impact of MITO through its experimental use and comparative analyses.

2. System Architecture

This section describes the software architecture of MITO and its main components.

2.1. Overview

The software architecture of MITO is depicted in the UML Package Diagram in Figure 1.

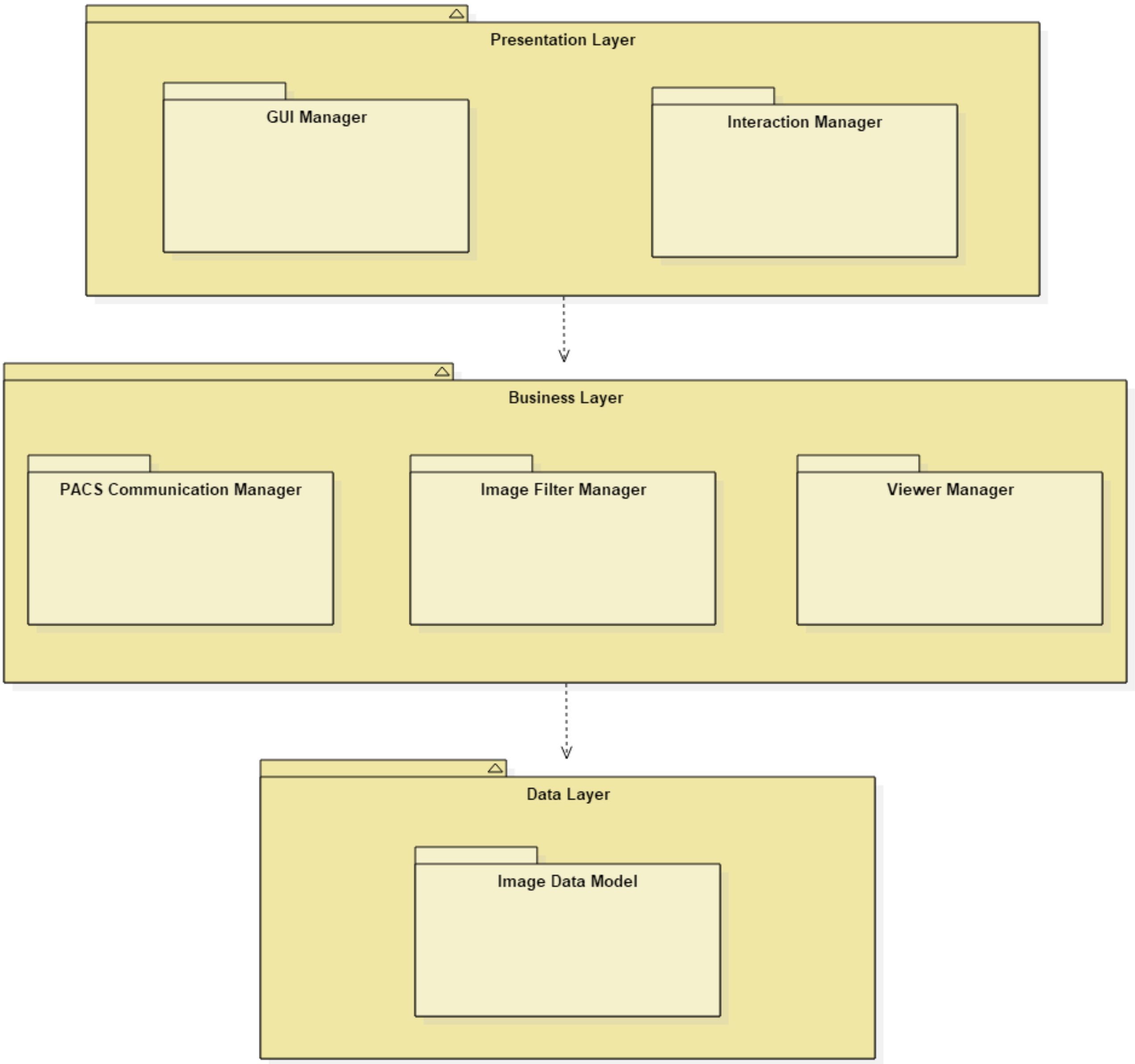

Figure 1. MITO software architecture

The architecture of MITO is organized in three layers:

- *Data Layer*: permits to represent and manage image data.
- *Business Layer*: implements logic and business rules for manipulating image data.
- *Presentation Layer*: catches the user requests and presents the results.

The layers of the system include six main building blocks:

- *Image Data Model*: implements the reference image data model used by the other blocks of the architecture.
- *PACS Communication Manager*: permits a user to interact with a PACS server for obtaining and sending medical images related to a patient.
- *Image Filter Manager*: provides a set of advanced techniques for processing image data both in 2-dimensional and 3-dimensional space.
- *Viewer Manager*: handles a set of software components (named viewers), which have the aim of presenting to the user the processed medical images.
- *GUI Manager*: manages the Graphical User Interfaces (GUIs) of the system on the basis of the type of rendering (2D or 3D) chosen by the user for a specific image dataset.
- *Interaction Manager*: catches and manages the interaction events generated by a user.

2.2. Building blocks

A description of each building block of the MITO architecture is provided below.

2.2.1. Image Data Model

The *Image Data Model* building block is organized in several objects, as shown in the UML Class Diagram in Figure 2.

The most important object is *PacsData*, which is based on a hierarchical structure compliant to the DICOM 3.0 standard used by a typical PACS server:

- *PatientData* has the aim of storing information about a patient, such as name, date of birth, identifier, sex, etc.
- *StudyData* is the object associated to a *PatientData* for handling information about a clinical study, like study date, accession number, referring physician, etc.
- *SeriesData* is the object related to a *StudyData* aiming at managing information about a series of images, like modality, series date, etc.
- *ImageData* represents the object associated to a *SeriesData* containing information about a single image of the series.

The list of *PacsData* objects exchanged with a PACS (represented by the *PacsNode* object) is memorized in the *PacsList* object. Every images can be saved on the local file system in files/directories represented by the *FileData* objects, which, in turn, are linked with the *itkVtkData* object that derives from the more general *appData* object. *itkVtkData*, representing the data structure handling the image study, is used by the other building blocks of the architecture for image processing, visualization, and interaction.

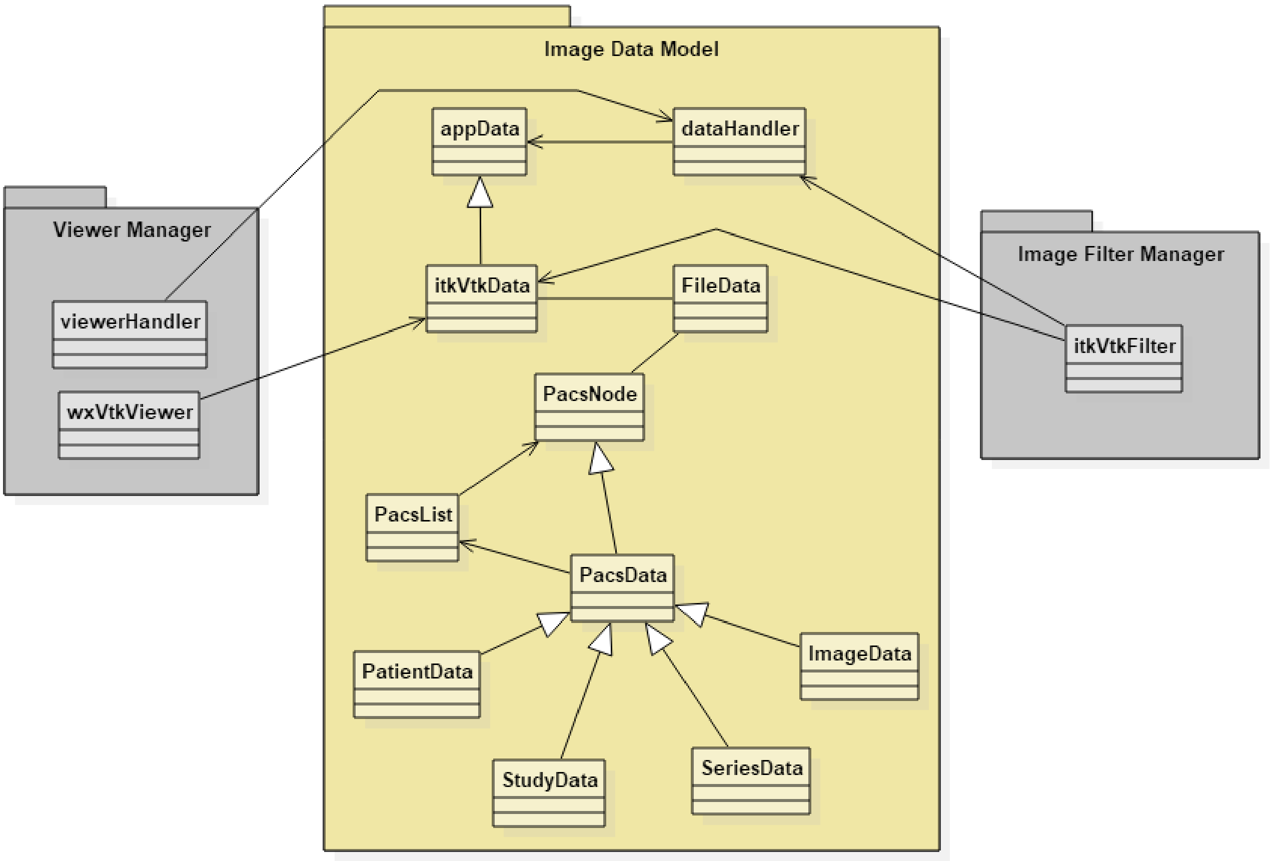

Figure 2. Image Data Model building block

2.2.2. PACS Communication Manager

The organization of the *PACS Communication Manager* building block is shown in the UML Class Diagram in Figure 3.

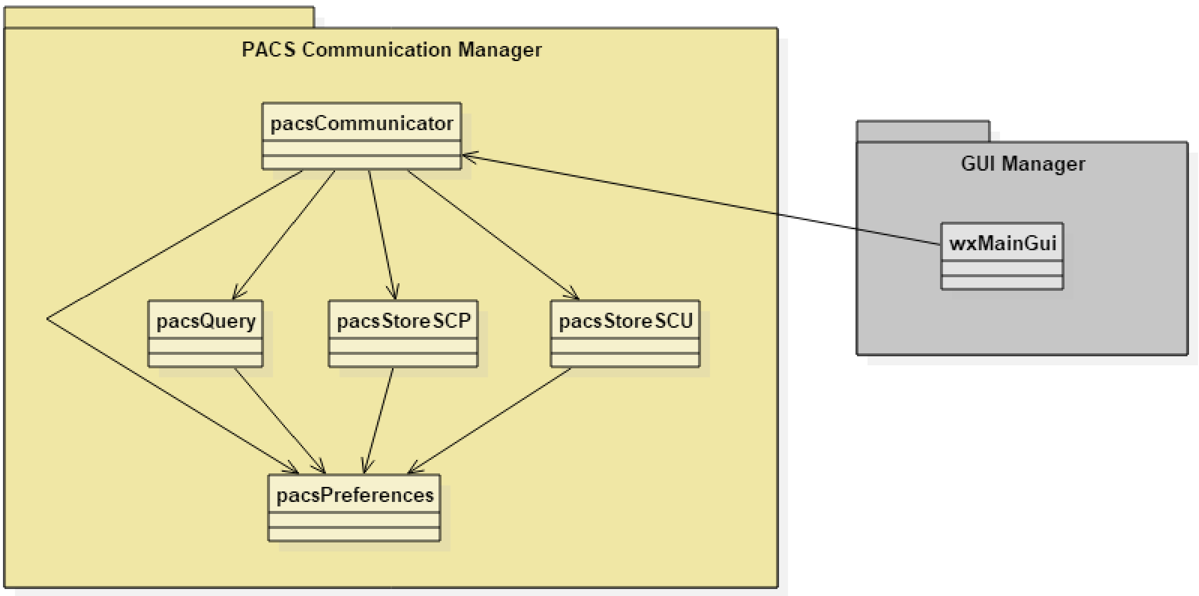

Figure 3. PACS Communication Manager building block

The building block provides a set of classes that permit a user to interact with a PACS server for querying, retrieving, and storing medical information and images related to a patient according to the DICOM 3.0 standard. On the basis on the type of operation, MITO and the PACS can act as a client, named Service Class User (SCU) according to the DICOM 3.0 terminology, or as a server, known as Service Class Provider (SCP):

- *Query*: MITO, acting as a DICOM Query/Retrieve SCU, queries a PACS Node, acting as a DICOM Query/Retrieve SCP, for some contents, like study date, patient name, patient identifier or patient birthdate. The results that satisfy the request are returned from the PACS to MITO and presented to the user.
- *Retrieve*: MITO, acting as a DICOM Query/Retrieve SCU, sends a retrieval request to a PACS Node, acting as a DICOM Query/Retrieve SCP, for a specific study or series of medical images. The results that satisfy the request are returned from the PACS, acting as a DICOM Storage Service SCU, to a DICOM Storage Server SCP previously specified, which memorizes the images requested. MITO can act as a DICOM Storage Server SCP: thus, it is able to receive images sent from the PACS and to store them in the local file system.
- *Store:* MITO, acting as a DICOM Storage Service SCU, sends a medical image to a PACS Node, acting as a DICOM Storage Service SCP, which memorizes it in the own archive.

2.2.3. Image Filter Manager

The organization of the *Image Filter Manager* building block is shown in the UML Class Diagram in Figure 5.

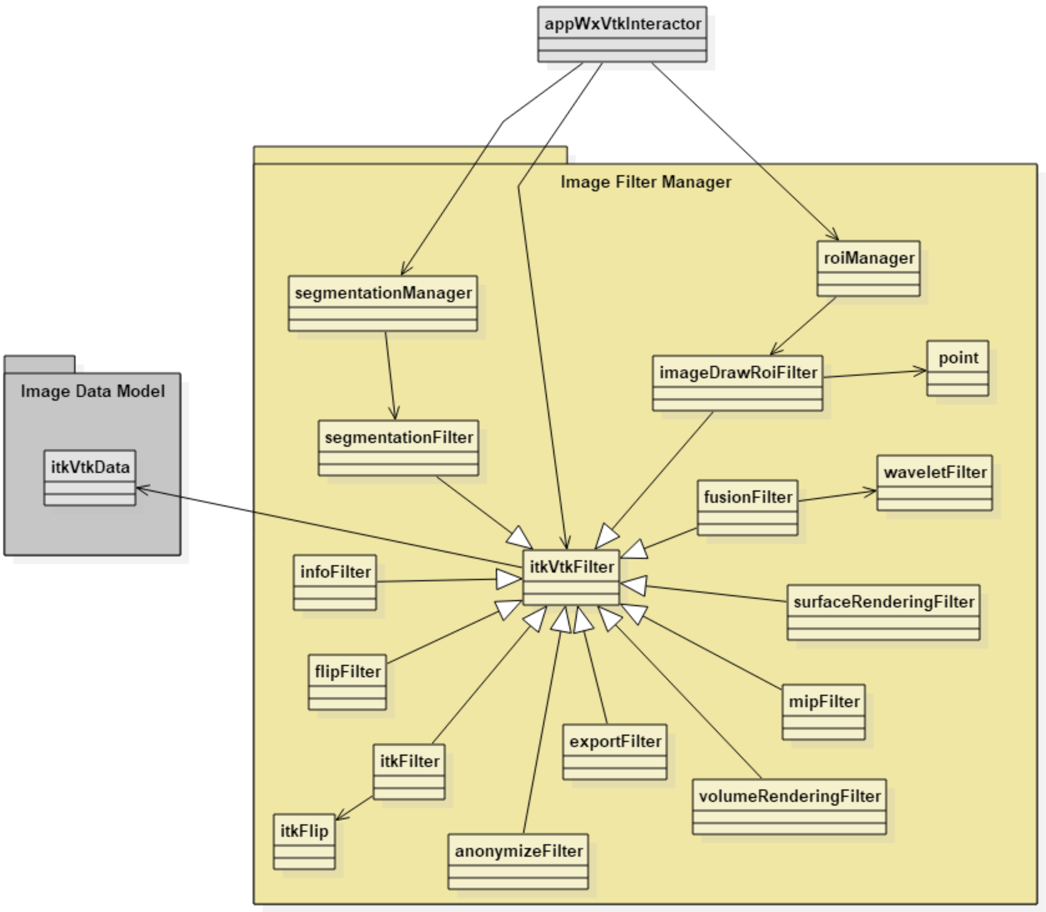

Figure 4. Image Filter Manager building block

The building block provides a set of basic and advanced techniques for processing image data both in 2-dimensional and 3-dimensional space, implemented as filters applied to the images. The core object is named *itkVtkFilter*, from which all the image filters derive. These filters include image flip, image segmentation, region extraction, image fusion, as well as 3D rendering techniques, like volume and surface rendering. They process the images represented in *itkVtkData* object. The modular organization of the building block permits to easily add further image processing techniques.

2.2.4. Viewer Manager

The organization of the *Viewer Manager* building block is shown in the UML Class Diagram in Figure 5.

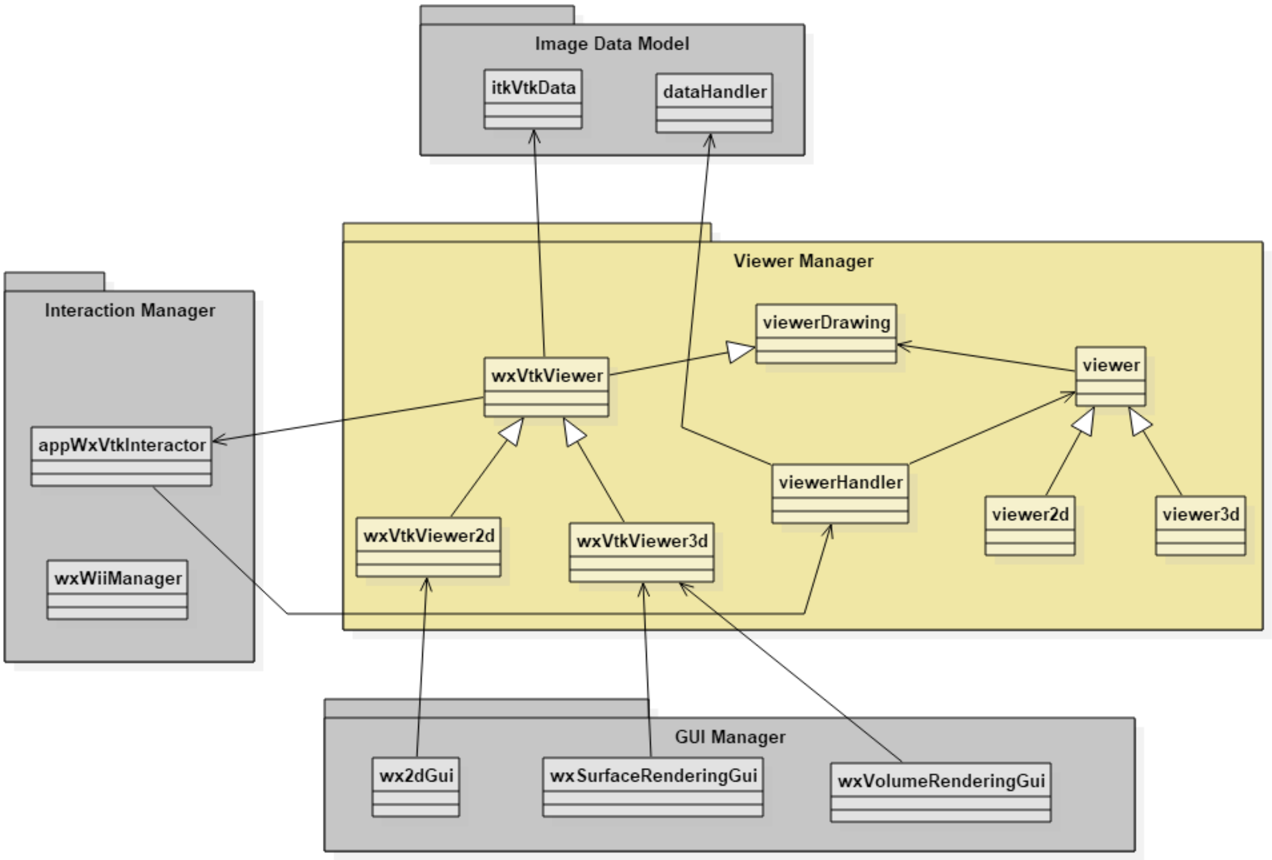

Figure 5. Viewer Manager building block

The building block manages a set of viewers, which have the aim of handling image data for their visualization. Two kind of viewers are implemented: one for handling 2D image data and another one for managing 3D image data. Each viewer is associated to one GUI, through which the user can visualize and interact with the images.

2.2.5. GUI Manager

The architecture of the *GUI Manager* building block is shown in the UML Class Diagram in Figure 6.

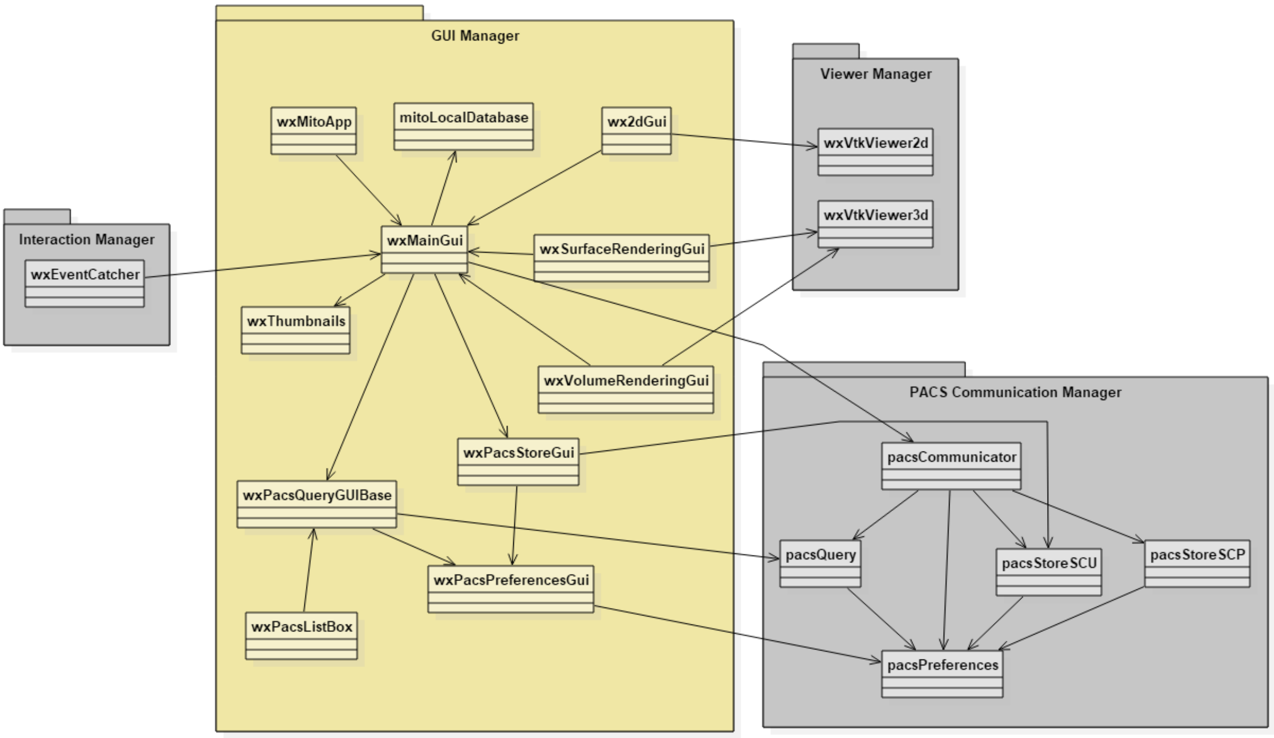

Figure 6. GUI Manager building block

The building block manages the graphical user interfaces of the system on the basis of the type of rendering (2D or 3D) requested from the user for a specific image dataset. Two kind of GUIs are implemented in MITO: general and specific GUIs. *wxMainGui* is a general GUI that has the aim of offering basic functionalities for processing visualized images, whereas *wx2dGui*, *wxVolumeRenderingGui*, and *wxSurfaceRenderingGui* are specific GUIs that provide a set of features depending on the type of visualization technique. Multiple GUIs can be simultaneously opened for navigating images rendered in different ways.

2.2.6. Interaction Manager

The organization of the *Interaction Manager* building block is shown in the UML Class Diagram in Figure 7.

The *Interaction Manager* catches the events generated by the interaction of the user with the user interfaces, interprets them, and dispatches derived information to the appropriate components; the user can interact with the system through both touch-based and touchless technology, in particular using mouse, keyboard, Wii or Kinect. Specifically, the graphical interfaces, through the main GUI, accept the user requests sent using the interaction tools preferred by the user. These requests are handled by the *wxEventCatcher* object, which recognizes the type of events and send them to the *appWxVtkInteractor* object. This one converts them in commands interpretable to the Viewer Manager and Image Filter Manager building blocks. Finally, the results are shown to the user by means of the appropriate GUI.

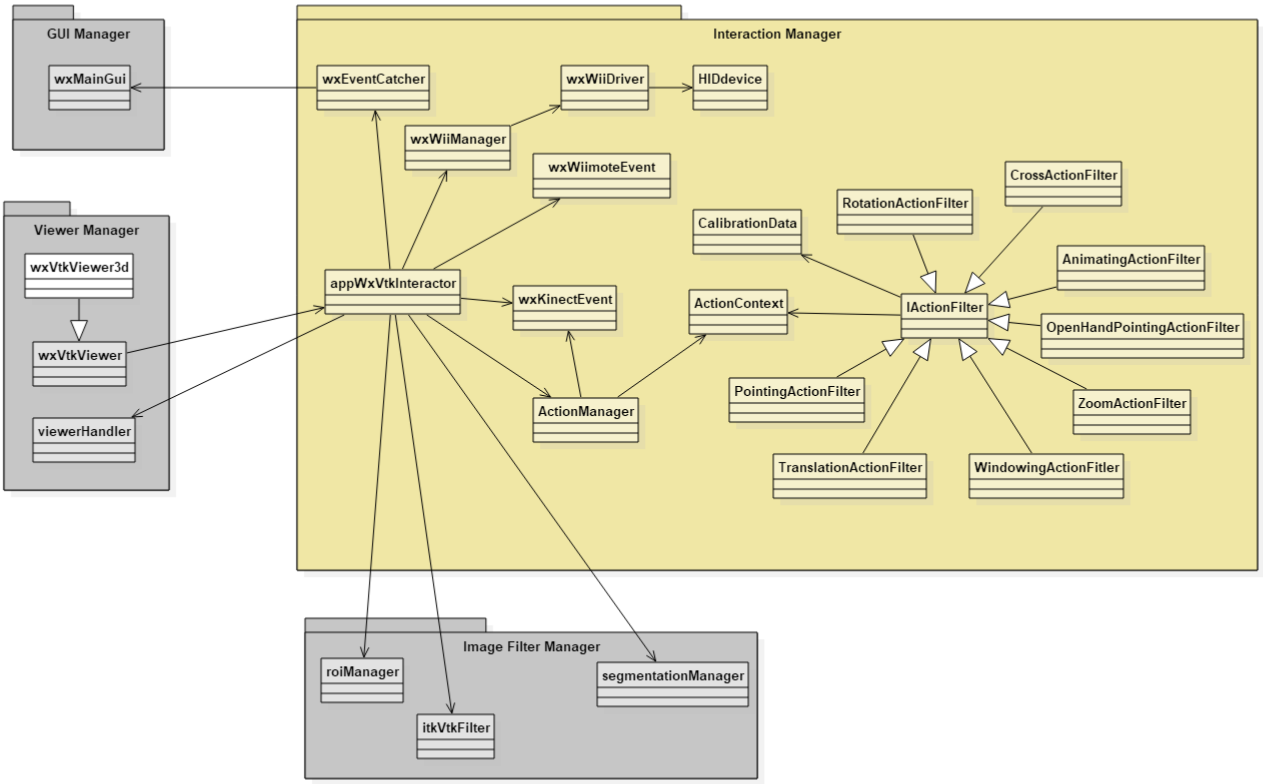

Figure 7. Interaction Manager building block

3. System Functionalities

This section describes the functionalities implemented in MITO for image acquisition, processing, and visualization. Such features, grouped in basic, image processing, and visualization functionalities, can be activated by the user interacting with the GUIs.

3.1. Basic functionalities

This sub-section describes the basic functionalities implemented in MITO.

3.1.1. Image Import and Export

Single DICOM images or complete DICOM image studies can be loaded in MITO. An album appositely designed is able to register the imported data by appending in a list of previously opened images an entry containing specific information, like patient demographic data, image modality, referring physician, etc. This list is loaded at each start of the MITO application.

In case a single image is imported, a single entry is added, containing only information about it. Instead, in case a DICOM study is imported, the entry is hierarchically organized in sub-entries, one for each series of images related to the imported study. All the entries added in the album can be deleted. Anyway, a single image is treated by the system as a study with one single image. For this reason, the rest of the technical report refers to study without loss of generality.

Then, as shown in Figure 8, MITO enables to show the DICOM header of the first series in order to visualize metadata present in the DICOM files.

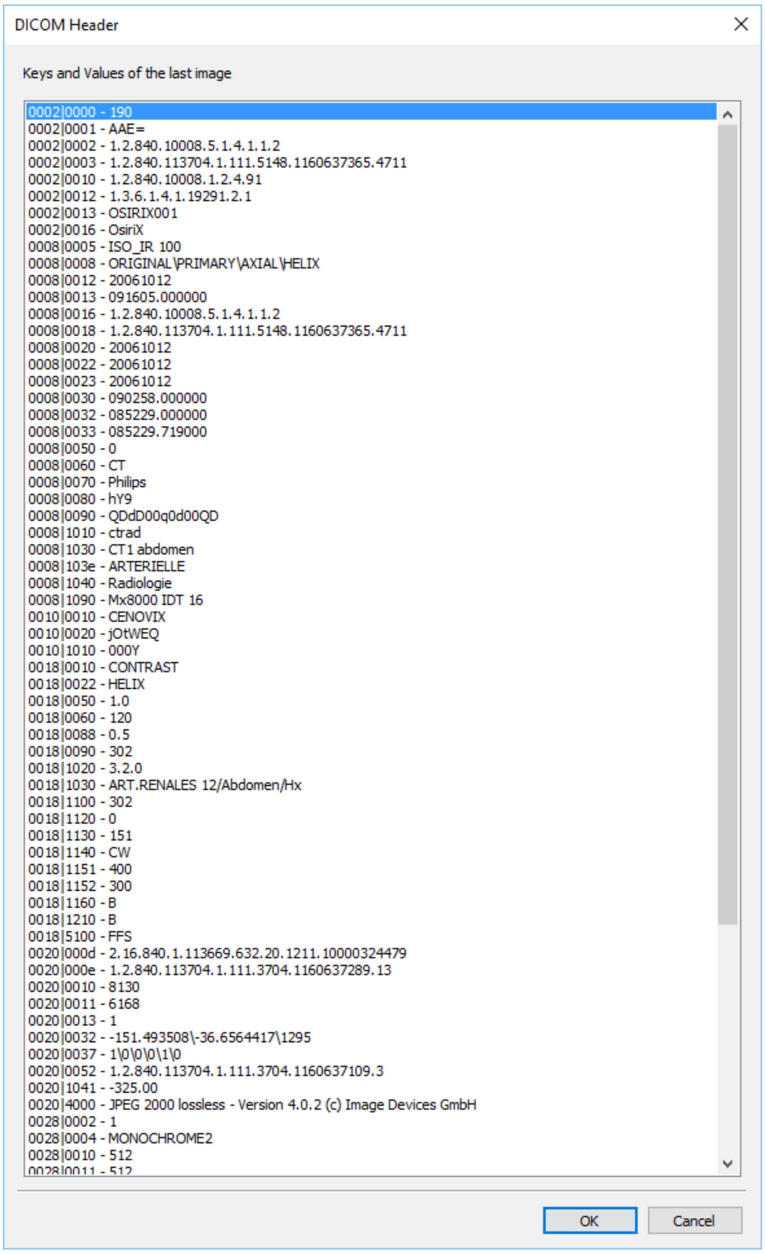

Figure 8. DICOM header

The image studies loaded in MITO can be exported in three different formats: Bitmap, DICOM, and DICOMDIR. It is also possible to export DICOM images in an anonymized form by specifying the metadata to remove, as shown in Figure 9.

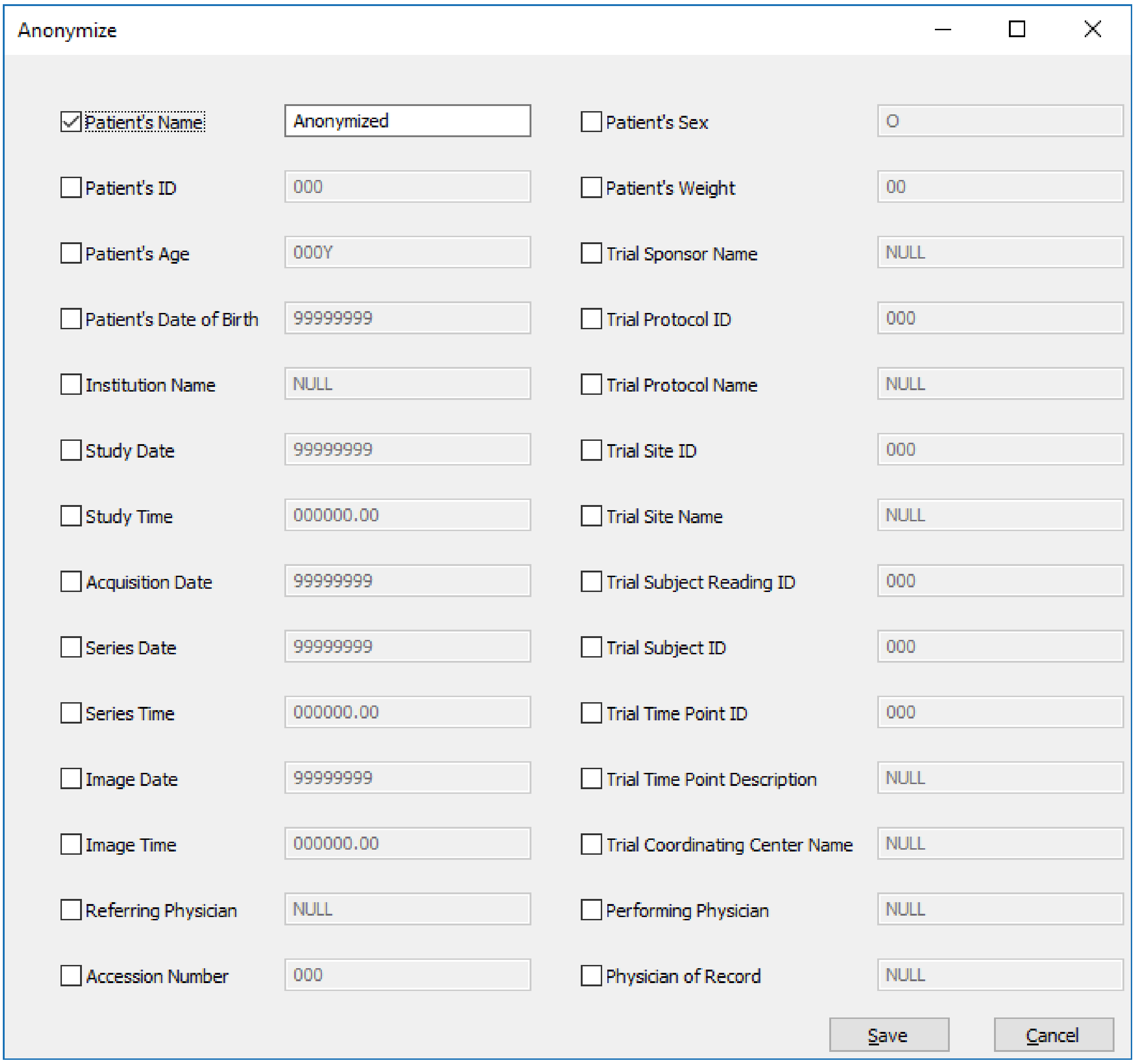

Figure 9. Anonymization

3.1.2. PACS Interaction

MITO can interact with PACS information systems for querying, retrieving, and saving DICOM images. In a preliminary configuration phase, the following parameters have to be specified:

- *Server Node:* indicates the host address of the PACS server.
- *Server Port:* indicates the port of the PACS server.
- *Called AP Title:* indicates the name of the Application Entity associated to the PACS server.
- *Calling AP Title:* indicates the name of the Application Entity registered in the PACS associated to the client where the MITO application is hosted.
- *Storage AP Title:* indicates the name of the Application Entity where the PACS server has to send images after that MITO invokes retrieval operations. It can correspond to the name of the Calling AP Title.
- *Storage Server Port:* indicates the port of the host selected for receiving DICOM images after a retrieval request.
- *Receive/Local Files Directory:* indicates the pathname where the local images have to be saved.

- *MAX PDU:* indicates the maximum Packet Data Unit to respect in the communication protocol with a PACS server.

It is possible querying a PACS server for obtaining the list of available studies satisfying some parameters, like patient name, patient identifier, patient birthdate, imaging modality, and study date, as shown in Figure 10.

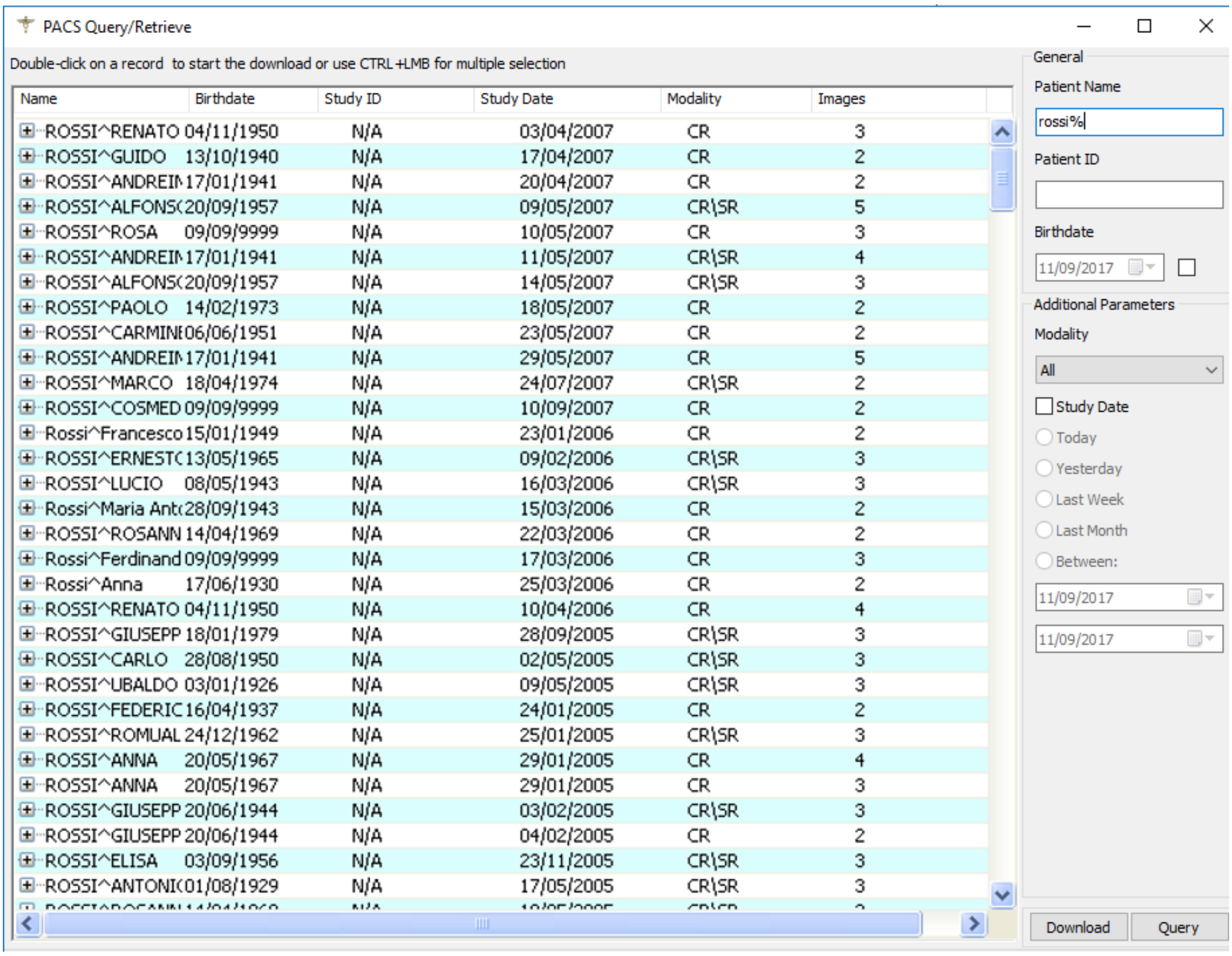

Figure 10. Querying data from a PACS

In order to import a specific DICOM study in MITO, the user has to select an item in the list and request the retrieval.

Finally, a DICOM series available on the host of the user can be stored in a PACS by selecting it through a file chooser, as shown in Figure 11.

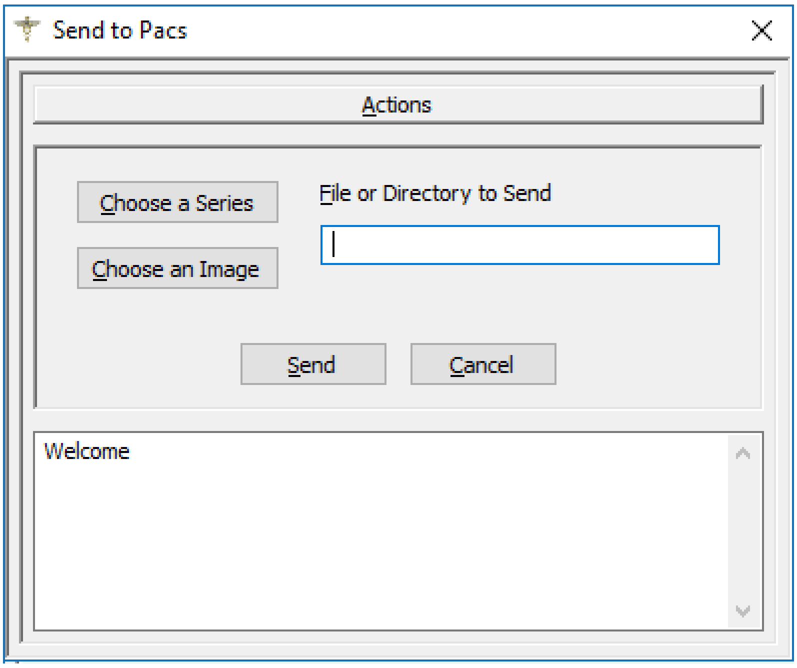

Figure 11. Sending data to a PACS

3.2. Image Processing Functionalities

The functionalities implemented in MITO for processing medical images are described in this sub-section, whereas the techniques developed for the reconstruction of 3D volume objects are detailed in the next sub-section.

3.2.1. ROI Selection and Extraction

This feature, as shown in Figure 12, allows selecting portions of one or more images in order to: i) measure some quantities; ii) extract regions of interest.

The following types of measurements on an image slice can be performed in MITO:

- length of a segment;
- area of a selected portion;
- volume of a portion applied to all the image slices of the series;
- angle of an arc.

Figure 12. Region of interest

3.2.2. Image Segmentation

Image segmentation is the process of separating an image into a sets of regions in order to simplify their analysis.

MITO includes three segmentation techniques based on the region growing algorithm, in order to isolate an object (e.g. an organ) into an image:

- *Connected Threshold:* starting from an initial seed, all the pixels whose intensity is between a lower and an upper limits, both specified by the user, are considered.
- *Neighborhood Connected:* all the pixels whose neighbors have an intensity that falls within a specified range are considered.
- *Confidence Connected:* starting from a specified multiplier f, the average m and the standard deviation σ of the pixels are computed, and all the pixels whose intensity belongs to the interval [m-$f\sigma$, m+$f\sigma$] are considered.

A sample of segmentation of a bone CT image is shown in Figure 13.

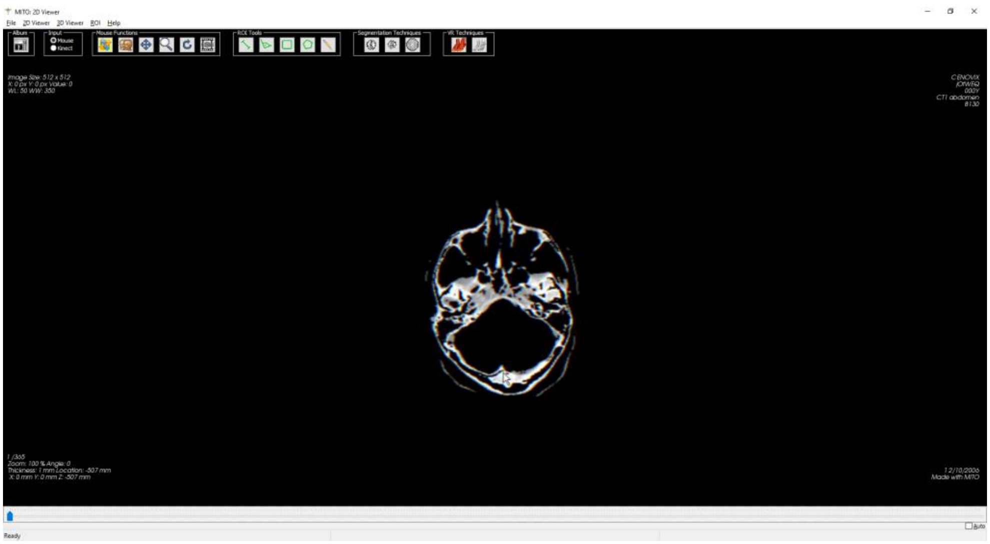

Figure 13. Image segmentation

3.2.3. Image Fusion

Medical images obtained from different sources provide often different information; for example, CT and MRI images focus on anatomical structures, whereas with PETs is possible to determine functional information. Thus, combining information coming from two or more co-registered multimodal medical images into a single image is an important facility to support medical diagnosis, as shown in Figure 14, where an example of fusion between a CT image and a PET image is schematized.

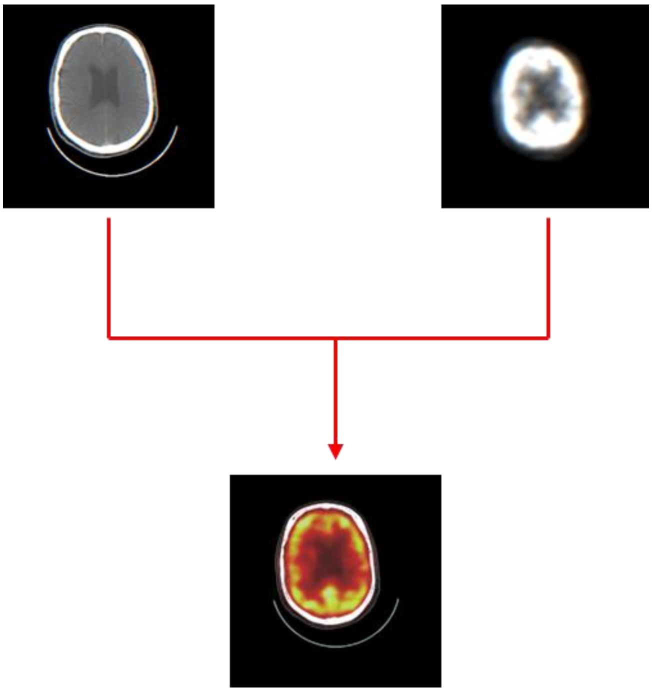

Figure 14. Image fusion

In MITO, two image series acquired with different modalities (CT-PET, MRI-PET, etc.) can be merged into a single image series according to two different approaches:

- *Arithmetic combination:* the fusion rule consists of a weighted average of the pixel intensities.
- *Wavelet-based combination:* a wavelet transform decomposes an image into low frequency components (approximations) and high frequency components (details), which are analyzed and combined independently. Specifically, the approximation components are combined by computing a weighted average of the intensities pixel per pixel, where the detail components are computed by calculating the maximum module of the intensities of the pixels that fall in a sliding window.

As shown in Figure 15, a user can choose the algorithm to apply and indicate if one of the two series have to be colored by applying a CLUT (Color Look Up Table) before that the fusion algorithm starts.

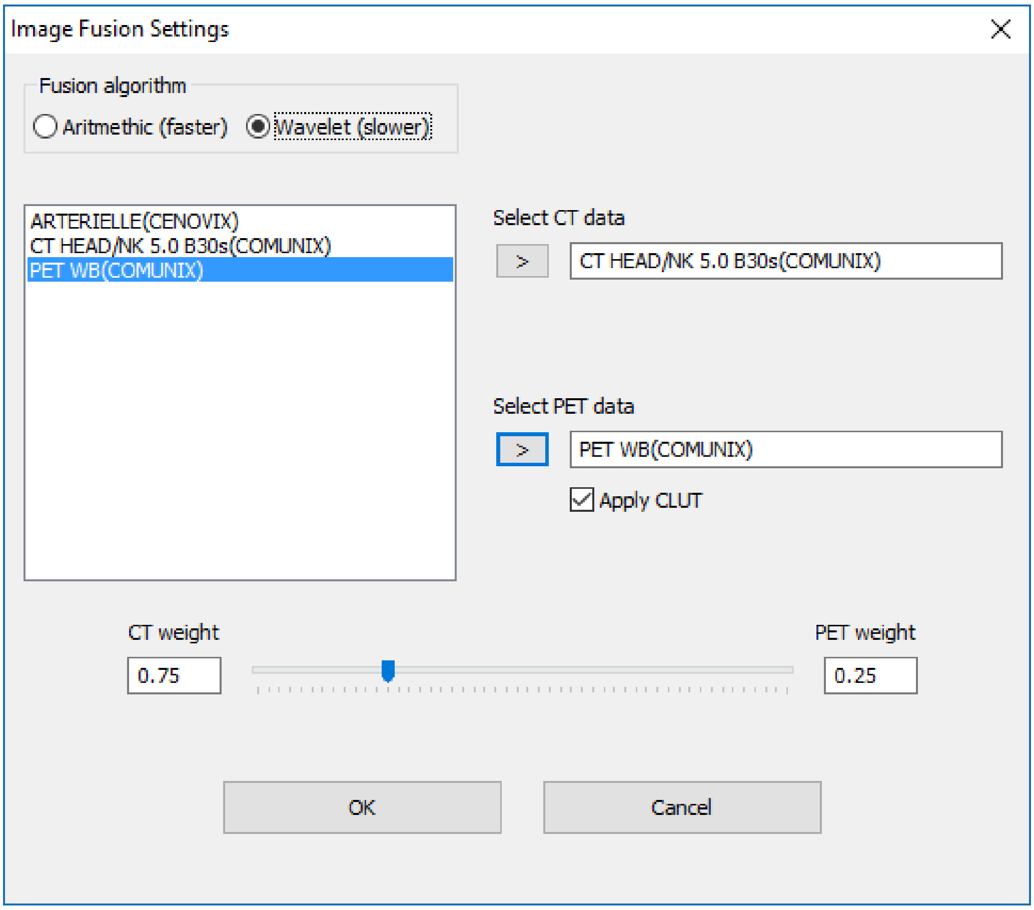

Figure 15. Image fusion settings

3.3. Visualization Functionalities

This section describes the functionalities provided with MITO for visualizing medical images in 2D and 3D.

3.3.1. 2D Visualization

Medical images like CT, MRI, PET are typically acquired in series of 2D image slices. As shown in Figure 16, MITO allows visualizing image slices in the way they were acquired (axial, coronal, sagittal) and to interact with them by applying a set of operations, like: zoom in/out, clockwise/anticlockwise rotation, translation, move, horizontal/vertical flip, brightness/contrast change.

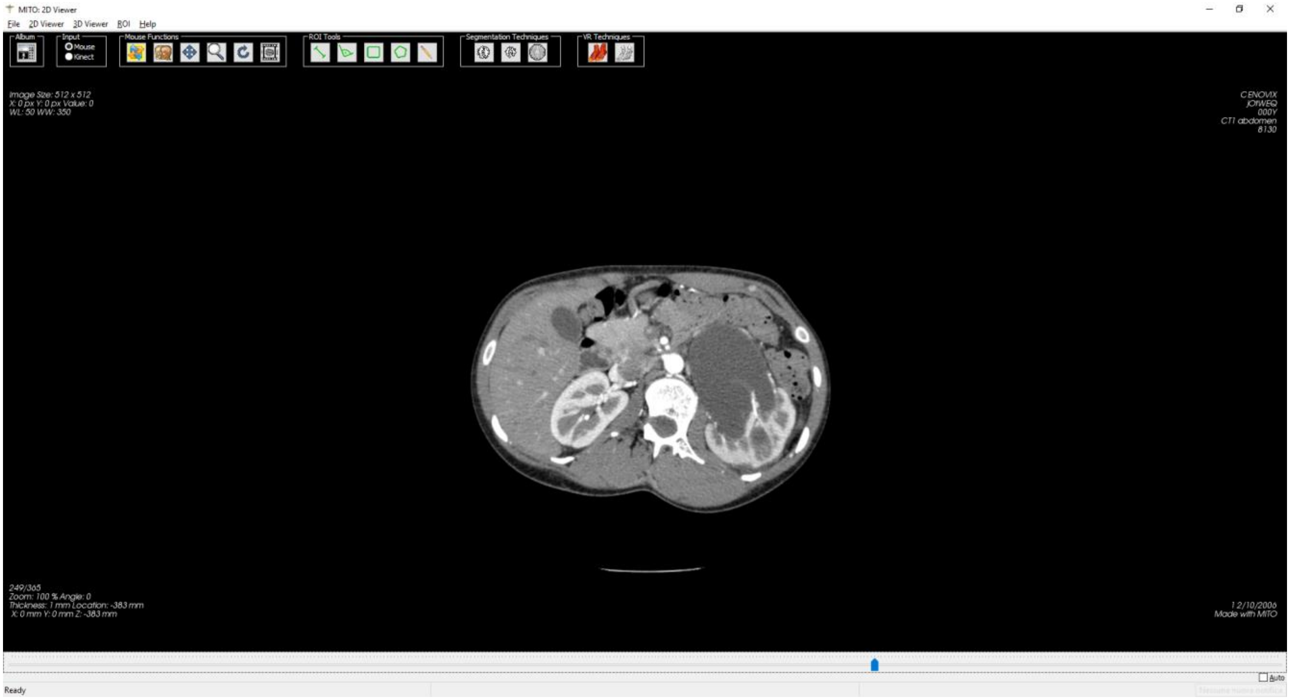

Figure 16. 2D viewer

3.3.2. 3D Visualization

3D direct reconstructions of two-dimensional studies can be obtained in MITO through two different direct volume rendering techniques:

- *Ray Casting:* starting from a viewpoint chosen by the user, a series of parallel rays enter the volume, the voxels crossed are sampled, interpolated, composed in RGBA, and finally projected onto a 2D view plane.
- *Maximum Intensity Projection (MIP):* this technique is a variant of the previous one. Data crossed are not composed in RGBA, but all the voxels whose maximum intensity falls along the paths of the parallel rays are projected on the view plane.

A user can interact with the reconstructed volume for adjusting the brightness/contrast or for navigating images by applying shift, zoom, rotation, translation, move, and several types of CLUTs.

An example of 3D direct reconstruction with application of a CLUT is shown in Figure 17.

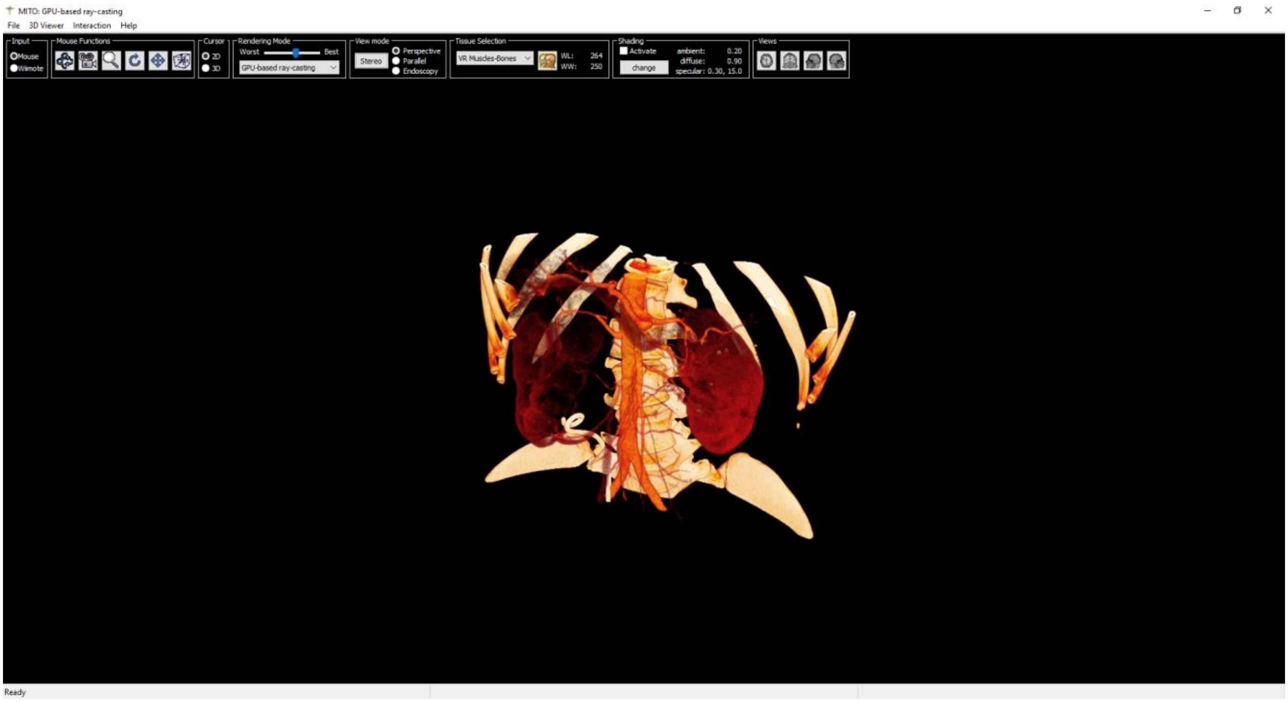

Figure 17. 3D direct volume rendering

A volume obtained by applying a volume rendering technique can appropriately be cropped. In this way, a user can interact with a volumetric portion of interest, as shown in Figure 18.

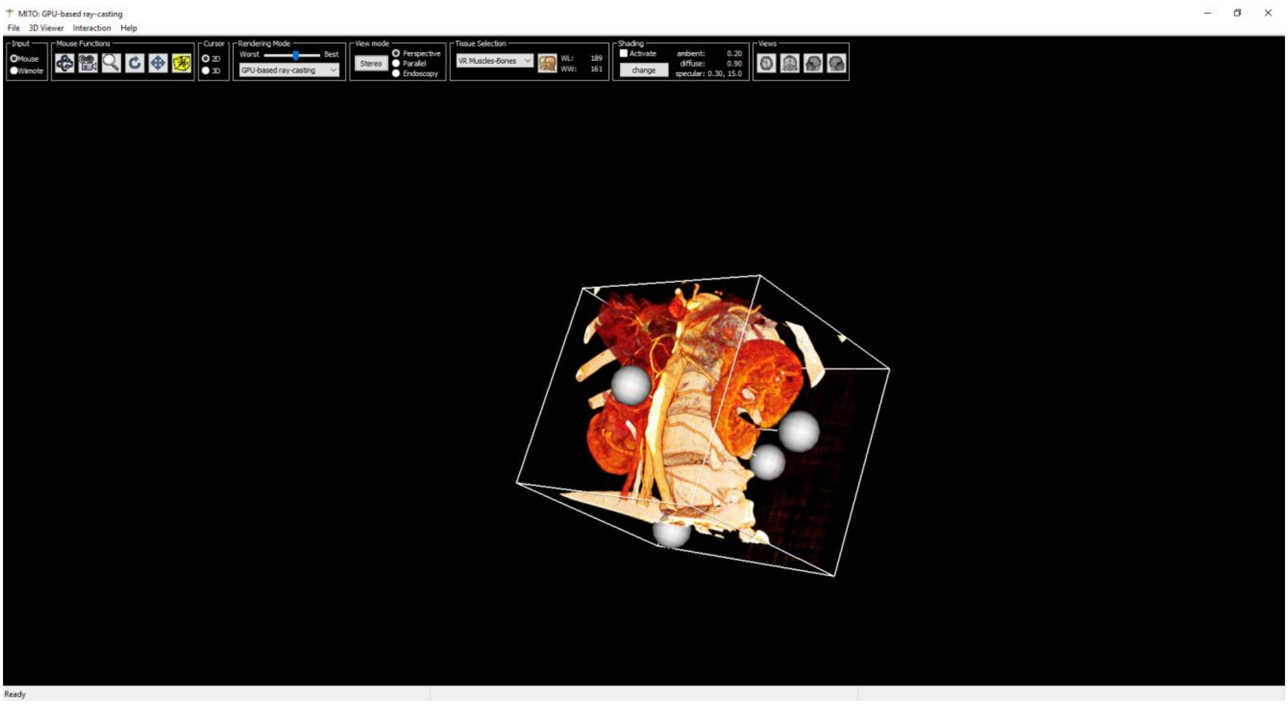

Figure 18. 3D cropping

MITO enables to reconstruct 2D studies also by means of 3D indirect volume rendering technique (also named surface rendering). This technique obtains an image from a volume into 2 phases:

- Construction of iso-surfaces from a user-defined value, said iso-value. This way, only the voxels with a value lower than the iso-value chosen are considered. On the basis of the iso-value set, it is possible extracting iso-surfaces belonging to specific body parts (like bones, skin, and so on).
- Construction of the final surfaces by interpolating and computing the gradient of the samples belonging to the iso-surfaces calculated.

A user can interact with the reconstructed volume by means of zoom, rotation, move, and translation operations.

An example of surface rendering reconstruction is shown in Figure 19.

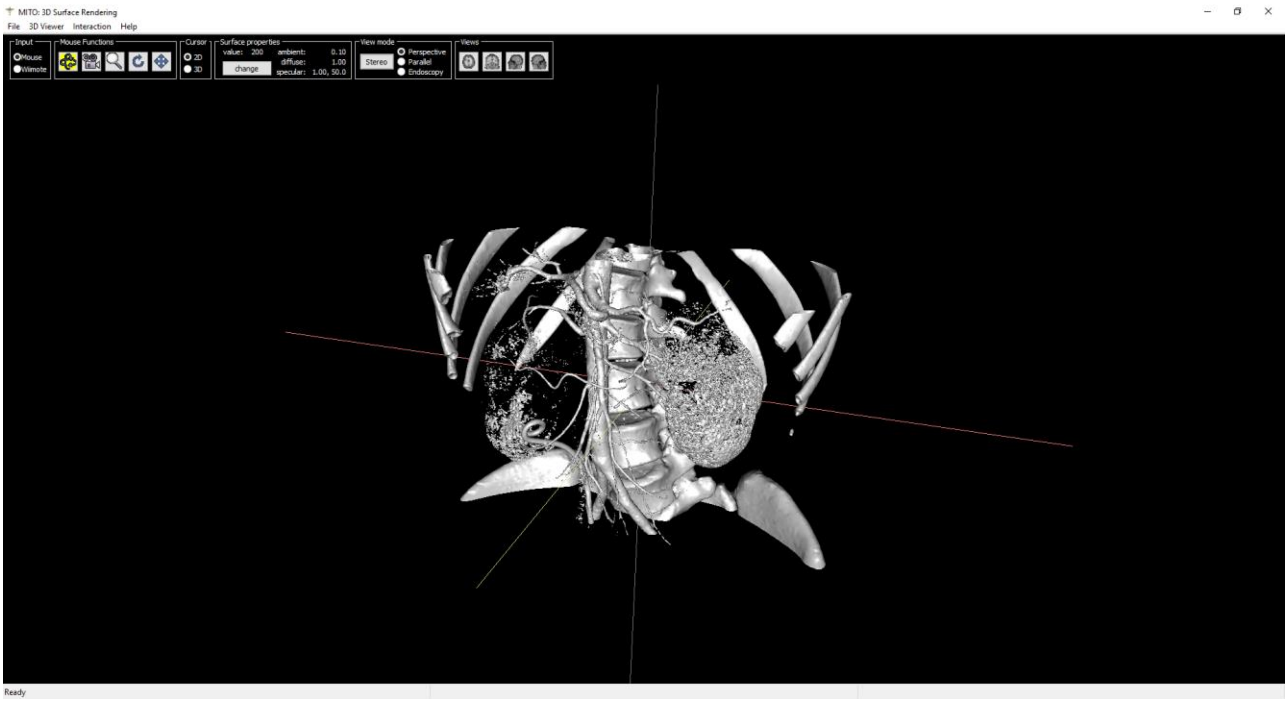

Figure 19. 3D indirect volume rendering

MITO offers also features for a semi-immersive visualization, in order to make a user able to interact with the environment in a much more intuitive manner. Specifically, a volume can be displayed in a stereoscopic mode (i.e., the volume is simultaneously rendered by two different points of view) according 3 different techniques:

- *Anaglyph:* two images are treated with two complementary color filters (red and cyan) and stereoscopy is achieved by wearing special glasses.
- *Passive polarization:* two overlapping images are projected on the same screen by 2 projectors with two separate polarized light beams. Glasses with lenses for two passive polarizing filters permit to correctly visualize the results.
- *Active polarization:* two images are alternately projected on the same monitor at a high frequency. Special glasses are required (shutter glasses) with lenses that alternately darken.

3.4. Interaction functionalities

MITO provides several user interfaces for browsing and manipulating both 2D and 3D medical images, using various input devices, as depicted in Figure 20. In more details, users can control medical images and volumes by using a Nintendo Wii Remote controller, a data glove equipped with additional sensors, or in a completely touchless way by using the depth sensors Microsoft Kinect v1 and v2 and the Asus Xtition Pro.

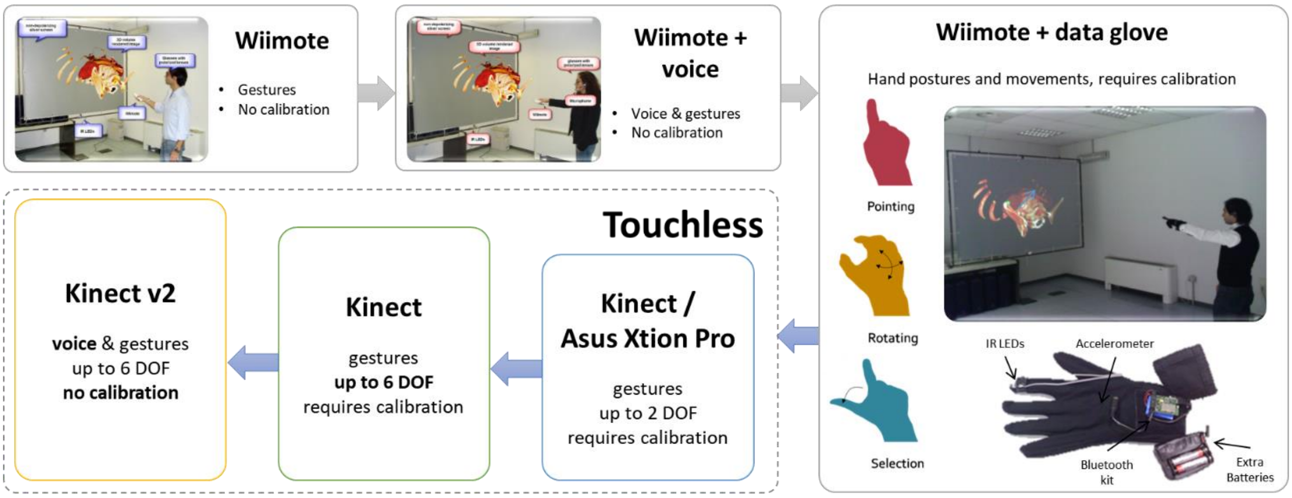

Figure 20. Available 2D/3D user interfaces in MITO

Specifically for the operating theatres, MITO features both a 2D and a 3D touchless interface, able to provide the surgeon with a direct, hands-on control on the images. In fact, in today's operating theaters images can be difficult to retrieve and manipulate. Surgeons need to visualize medical images without having to physically touch any control, since they cannot leave the sterile field around the patient. Scrubbing in and out to access the computer safely can result in a severe increase in the duration of the operation: cleaning to prevent bacterial contamination after checking a computer can take up to 20 minutes, sometimes adding a full hour to surgery. Another option could be to rely on a proxy, but she/he may not share the same level of professional vision of the surgeon.

MITO recognizes voice and hand gestures and static hand postures as commands to browse and display 2D medical images and 3D reconstructions of anatomical parts of the patient during a surgery. The user interfaces also comprise algorithms for filtering natural hand tremors, for recognizing static hand gestures from low resolution depth images at different distances and differently oriented, and for smoothing noisy movement data.

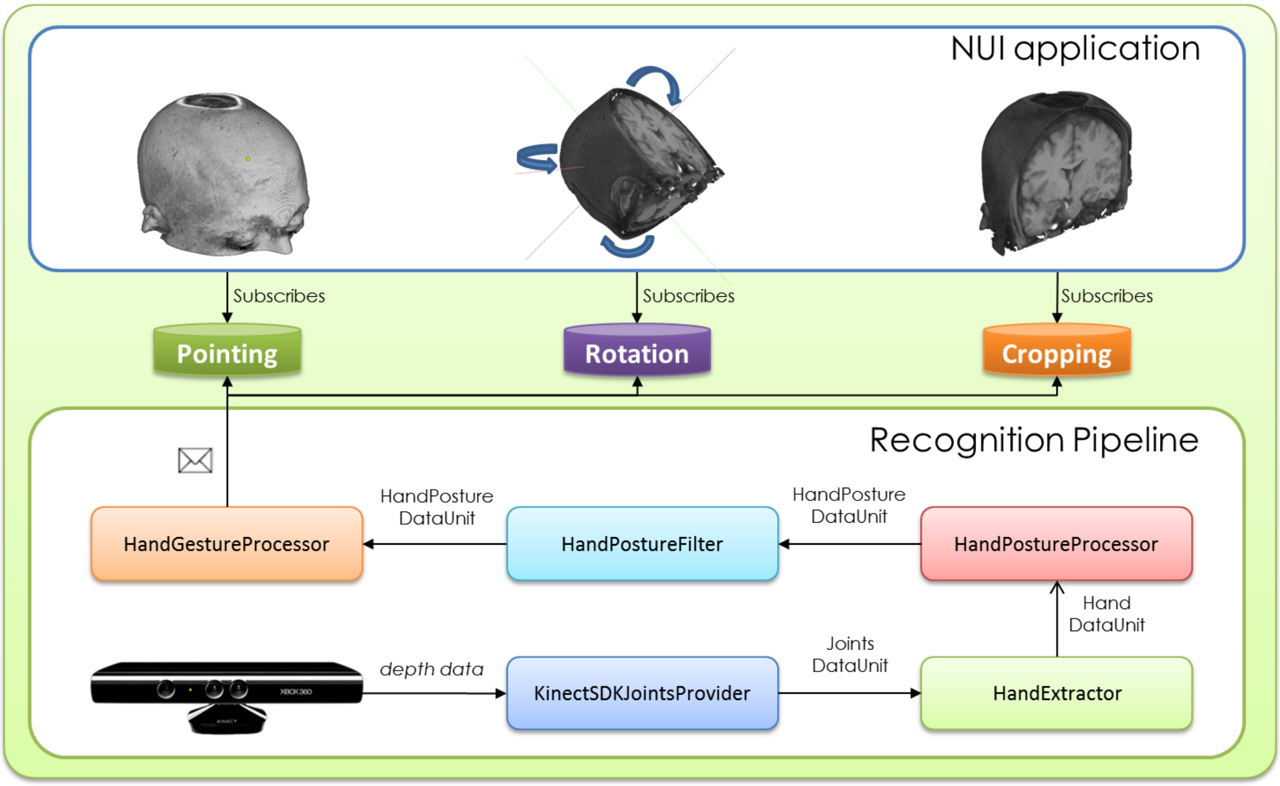

Figure 21. MITO 3D touchless interaction pipeline

4. Graphical User Interfaces

This section describes the GUIs implemented in MITO. At the start of the application, the user can interact with the system through the general GUI, named *wxMainGui*, which represents the user point of reference of the whole system. The design of this GUI is organized in three main panels, as shown in Figure 22.

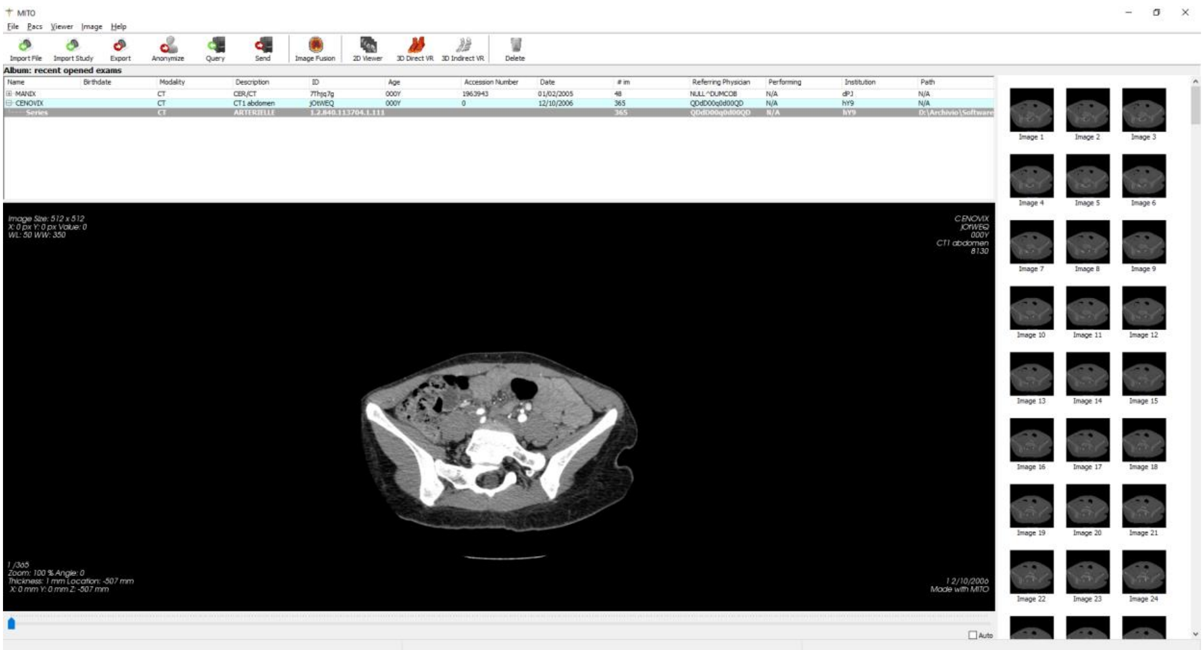

Figure 22. Main GUI

The first panel encloses the album able to register the most relevant metadata related to the imported image study in the time, along with its series.

The second panel contains a 2D viewer showing the image slices of an imported image data. After that a DICOM study is opened, the first image slice of the first series is visualized in the 2D viewer. The other images of the series can be visualized interacting with a slicer, whereas the other series of the study can be visualized selecting a series in the album.

Finally, the third panel presents a set of thumbnails showing all the image slices of the series currently opened.

The functionalities offered by this GUI are Image Import and Export, PACS Interaction, and 2D Visualization.

Starting from *wxMainGui*, the user can request to process and visualize medical images, by opening the appropriate specific GUIs, that are *wx2dGui*, *wxVolumeRenderingGui*, and *wxSurfaceRenderingGui*.

wx2dGui permits the user to navigate a 2D series of DICOM images in widescreen, providing all the image processing functionalities described in this technical report.

wxVolumeRenderingGui is able to show 3D images obtained by applying direct volume rendering techniques to the 2D series loaded in *wx2dGui*.

Finally, *wxSurfaceRenderingGui* renders 3D images obtained by applying indirect volume rendering techniques to the 2D series loaded in *wx2dGui*.

It is interesting noting that the direct and indirect volume rendering techniques can be applied also to the images processed by using image processing functionalities.

5. Implementation Details

MITO is entirely written in C++ language and built on open-source and cross-platform libraries, which were combined and extended in order to support medical imaging processing, semi-immersive 3D model visualization, and interaction functionalities.

In particular, the libraries used by MITO are the following:

- OpenGL (Open Graphics Library) for fast 2D and 3D display;
- VTK (Visualization ToolKit) for 2D-3D rendering;
- ITK (Insight segmentation and registration Toolkit) for image segmentation and registration;
- DCMTK (DICOM ToolKit) for DICOM handling support;
- IJG JPEG (Independent JPEG Group JPEG) for converting DICOM images to JPEG thumbnails;
- CxImage, for exporting DICOM images to other formats (e.g. BMP);
- WxWidgets (Windows and X widgets) for graphical user interface design and development.

The software development environment used is Microsoft Visual Studio, whereas CMake was used for generating and managing the project. MITO was extensively tested on Microsoft Windows operating systems.

6. Impact and Conclusions

The “MITO – Medical Imaging Toolkit” project coagulates a number of activities aimed at defining and implementing an open-source software architecture for advanced Medical Imaging.

MITO allows fetching radiological information and images stored in a PACS health information system according to the DICOM 3.0 format, and provides the final user with basic functionalities such as 2D-3D visualization (ray casting, maximum intensity projection, surface rendering), image segmentation and fusion, ROI extraction. Moreover, MITO implements interaction techniques for touchless image manipulation and volume navigation.

The impact of MITO in the developers' community has tangible, as at the time of this writing it has been downloaded more than 40,000 times. The system has been experimented worldwide in several healthcare facilities, universities, research centers, and private companies, including King's College London, University of Geneva, Lion City Radiology Pte Ltd in Singapore, Oregon Medical Group, INRIA Sophia Antipolis, Technical University of Delft, Department of Radiology of the Clínica las Condes, UK Aachen, University of Cambridge, Oregon Health & Science University, Integrated Health Information Systems Pte Ltd, University of the Saarland, Pfizer, Imagens Médicas de Brasília, Institute of Biostructure and Bioimaging of the Italian National Research Council, Hospital of the Incurables in Naples.

In a comparative evaluation study conducted in 2010, Barra et al. have found about 70 freeware medical image viewers and among them 11 were able to run as PACS clients. Six were selected for analysis: ClearCanvas Workstation, KPACS, Onis, Synedra View Personal, MITO, and Tudor DicomViewer. Sixteen functions selected according to the authors' needs were evaluated. At the end of the study, MITO was particularly appreciated for its image processing functionalities, with particular regard to PET-CT fusion capability. In November 2012, the MITO 3D user interface was awarded with the Second Place Award at the ChaLearn Gesture Challenge at the International Conference on Pattern Recognition (ICPR). On the Communication of the ACM edition of January 2014, Kenton O'Hara et al. included MITO within the state-of-the-art systems considering touchless control of medical images for surgical settings.

References

[1] MITO SourceForge project, https://sourceforge.net/projects/mito/

[2] ICAR-CNR, http://www.icar.cnr.it/

[3] iHealth Laboratory, http://ihealthlab.icar.cnr.it/

[4] F. Salaffi, S. Farah, M. Carotti, and S. Fioretti, *"Recent advances in visualization, volumetric analysis and 3D reconstruction in musculoskeletal imaging using the OsiriX open source DICOM software"*, International Journal of Scientific and Engineering Research, vol. 6, no. 7, pp. 1733-1747, 2015.

[5] M. C. B. Seixas, J. C. S. Cardoso, and M. T. G. Dias, *"The Leap Motion movement for 2D pointing tasks: Characterisation and comparison to other devices"*, in PECCS 2015: Proceedings of the International Conference on Pervasive and Embedded Computing and Communication Systems, pp. 15-24, 2015

[6] A. Manolova, *"System for touchless interaction with medical images in surgery using Leap Motion"*, in CEMA 2014: Proceedings of the 9th International Conference on Communications, Electromagnetics and Medical Applications, pp. 2-6, 2014

[7] É. S. Silva and M. A. F. Rodrigues, *"Design and evaluation of a gesture-controlled system for interactive manipulation of medical images and 3D models"*, SBC Journal on Interactive Systems, vol. 5, no. 3, 2014

[8] L. Gallo, *"Hand shape classification using depth data for unconstrained 3D interaction"*, Journal of Ambient Intelligence and Smart Environments, vol. 6, no. 1, pp. 93-105, 2014. doi:10.3233/AIS-130239

[9] F. Cosentino, N. W. John, and J. Vaarkamp, *"An overview of augmented and virtual reality applications in radiotherapy and future developments enabled by modern tablet devices"*, Journal of Radiotherapy in Practice, vol. 13, no. 3, 2014 , pp. 350-364. doi:10.1017/S1460396913000277

[10] L. Gallo, *"A study on the degrees of freedom in touchless interaction"*, in SIGGRAPH Asia 2013 Technical Briefs, ACM, New York, NY, USA, Article 28. doi:10.1145/2542355.2542390

[11] M. Magliulo, L. Cella, and R. Pacelli, *"Novel technique radio frequency identification (RFID) based to manage patient flow in a radiotherapy department"*, in BHI 2012: Proceedings of the IEEE-EMBS International Conference on Biomedical and Health Informatics, 2012. Doi:10.1109/BHI.2012.6211751

[12] L. Gallo, A. P. Placitelli, and G. De Pietro, *“A Kinect NUI for 3D medical visualization”*, in CHALEARN Gesture Demonstration Competition, in conjunction with the 21st International Conference on Pattern Recognition (ICPR 2012) (Tsukuba, Japan), Second Place Award, 2012

[13] L. Gallo, and A.P. Placitelli, *“View-independent hand posture recognition from single depth images using PCA and flusser moments”*, in SITIS 2012: Proceedings of the 8th International Conference on Signal Image Technology & Internet Based Systems, (Los Alamitos, CA, USA), pp. 898-904, IEEE Computer Society, 2012. doi:10.1109/SITIS.2012.133

[14] M. Ciampi and T. Shon, *“A collaborative architecture for healthcare applications”*, Information-An International Interdisciplinary Journal, International Information Institute, vol. 15, no. 2, pp. 607-618, 2012

[15] L. Gallo, A. P. Placitelli, and M. Ciampi, *“Controller-free exploration of medical image data: experiencing the Kinect”*, in CBMS 2011: Proceedings of the 24th IEEE International Symposium on Computer-Based Medical Systems, (Los Alamitos, CA, USA), pp. 1–6, IEEE Press, 2011. doi:10.1109/CBMS.2011.5999138

[16] F. R. Barra, R. R. Barra, and B. S. Alaor, *“Freeware medical image viewers: can we rely only on them?”*, Radiologia Brasileira, vol. 43, no. 5, pp. 313-318, 2010. doi:10.1590/S0100-39842010000500010

[17] M. Ciampi, *“Design of a layered component-based system for sharing visualization objects in collaborative environments”*, International Journal of Advanced Computer Engineering, Serials Publications, vol. 5, no. 1, pp. 15-20, 2012

[18] L. Gallo, A. Minutolo, and G. De Pietro, *“A user interface for VR-ready 3D medical imaging by off-the-shelf input devices”*, Computers in Biology and Medicine, vol. 40, no. 3, pp. 350–358, 2010. doi:10.1016/j.compbiomed.2010.01.006

[19] M. Ciampi, *“Medical image fusion for color visualization via 3D RDWT”*, in ITAB 2010: Proceedings of the 10th IEEE International Conference on Information Technology and Applications in Biomedicine, pp. 1-6, IEEE Press, 2010. doi:10.1109/ITAB.2010.5687754

[20] L. Gallo, G. De Pietro, and I. Marra, *“3D interaction with volumetric medical data: experiencing the Wiimote”*, in Ambi-Sys 2008: Proceedings of the 1st International Conference on Ambient Media and Systems, (Brussels, Belgium), pp. 1-6, ICST, 2008. doi:10.1145/1363163.1363177

[21] L. Gallo, G. De Pietro, A. Coronato, and I. Marra, *“Toward a natural interface to virtual medical imaging environments”*, in AVI 2008: Proceedings of the Working Conference on Advanced Visual Interfaces, (New York, NY, USA), pp. 429-432, ACM Press, 2008. doi:10.1145/1385569.1385651

[22] L. Gallo, G. De Pietro, and I. Marra, *“User-friendly inspection of medical image data volumes in virtual environments”*, in CISIS 2008: Proceedings of the International Conference on Complex, Intelligent and Software Intensive Systems, (Los Alamitos, CA, USA), pp. 749-754, IEEE Computer Society, 2008. doi:10.1109/CISIS.2008.33

[23] L. Gallo and G. De Pietro, *“Input devices and interaction techniques for VR-enhanced medicine”*, in Multimedia Techniques for Device and Ambient Intelligence (E. Damiani and J. Jeong, eds.), pp. 115-134, Springer US, 2009. doi:10.1007/978-0-387-88777-7_5

[24] G. De Pietro, L. Gallo, I. Marra, and C. Vanzanella, *“A new approach for handling 3D medical data in an immersive environment”*, in VECIMS 2007: Proceedings of the IEEE Symposium on Virtual Environments, Human-Computer Interfaces and Measurement Systems, (Los Alamitos, CA, USA), pp. 63-66, IEEE Computer Society, 2007. doi:10.1109/VECIMS.2007.4373929

[25] B. Alfano, M. Ciampi, and G. De Pietro, *“A wavelet-based algorithm for multimodal medical image fusion”*, in SAMT 2007: Proceedings of the 2nd International Conference on Semantic and Digital Media Technologies, Lecture Notes in Computer Science, vol. 4816, pp. 117-120, 2007. doi:10.1007/978-3-540-77051-0_13

[26] O'Hara, K., Gonzalez, G., Sellen, A., Penney, G., Varnavas, A., Mentis, H., Criminisi, A., Corish, R., Rouncefield, M., Dastur, N, & Carrell, T. (2014). Touchless interaction in surgery. Communications of the ACM, 57(1), 70-77