Volume 53, Number 3, 2012

ACTA TECHNICA NAPOCENSIS
Electronics and Telecommunications

MIPIOS – MEDICAL IMAGE PROCESSING APPLICATION ORIENTED FOR MOBILE DEVICES

Marius-Cristian URECHE Mircea-Florin VAIDA
Faculty of Electronics, Telecommunications and IT, Technical University of Cluj-Napoca, Romania
George Baritiu 26-28, 400027 Cluj-Napoca, Romania, Email: Cristian.Ureche@com.utcluj.ro

Abstract: Mobility is thought as a key concept that fundamentally changed our way of working and interacting during professional life. However, mobility is a concept of particular interest in the field of medicine and particularly in tele-health. In this paper we present a novel application whose main objective is to take advantage of the major advances of mobile devices for immediate application in the medical domain. We designed our application entitled MIPiOS (Medical Image Processing on iOS) for manipulating, analyzing and processing medical images in DICOM format by using the benefits of touchscreens. With the addition of the Retina Display for several mobile devices produced by Apple, our proposed application's benefits are even more easily to be exploited because of better finger gestures recognition, better resolution and better contrast.

Keywords: *medical image processing, iOS-based mobile devices, DICOM image visualization, mobile software application.*

I. INTRODUCTION

The way human society consumes information changed fundamentally in the last couple of years. The astonishing development of the internet, the appearance of mobile devices such as Personal Digital Assistants (PDAs), smartphones or more recently tablets, the standardization of wireless communication technologies enable very high throughput multimedia applications for mobile devices without remarkable loss of quality.

Medicine is a field that cannot be avoided by this general trend. With the advancements in digital electronics and integrated circuits, medical imaging devices passed from an analogical perspective to a digital one regarding the processing of medical images. Standardized formats such as DICOM (Digital Imaging and Communications in Medicine) were adopted [1].

Our paper is structured in several chapters. Chapter 2 gives an overview of what already exists related to medical image processing using iOS mobile devices and also presents our vision and the novelty of MIPiOS. In chapter 3 we present and explain the conceptual architecture of the application and the class diagram. Every tier is detailed. A brief hardware architecture is also proposed. Chapter 4 is dedicated to the presentation of the functionalities from the final user point of view which is expected to be a person interested in analyzing and manipulating medical images. We present our mid-term MIPiOS expectations in chapter 5. In this chapter the conclusion is also formulated.

II. DESCRIPTION OF PURPOSE

In the field of medical image processing using mobile devices there are not many applications available so far. OsiriX 2] is probably the best application for visualizing medical images in DICOM format and for transferring data using DICOM network format. However their iPad application should be integrated with their other solutions in order to function optimally. It just assures the visualization of medical images, but no registration or fusion is available. IoDicom [3] is another application available on iTunes that offers functionalities such as viewing and archiving medical images in DICOM format.

iMango is a medical image application implemented for viewing and analyzing images, providing tools to define, edit and save regions of interest. However, it lacks tools for registering images or enhancing them [4].

There is also an attempt to make ITK available for iOS mobile devices, but for the moment it is not at the right level for using it in developing commercial applications.

The aim of our project is to create a software prototype for iOS based mobile devices (that natively run on iPad, iPhone, IPod) to facilitate the visualization, manipulation and processing of multimodal medical images (e.g. Ultrasound (US) scans, Magnetic Resonance Imaging (MRI) slices, Computed Tomography (CT) results) in DICOM format and in other non-medical formats (TIFF, BMP, JPG).

The usefulness of the project appears naturally in a moment when mobility became the key element in many activity fields. The radiologist, the surgeon and even the medical advisor (or the personal physician) will be able to identify, diagnose and monitor the evolution of certain medical problems (including the evolution of tumors during therapies). Moreover, new mobile devices such as tablets pave the way to a better visualization and manipulation of multimodal medical images compared to traditional PCs that use standard keyboard and mouse. Rotating an image, performing a zoom, selecting the

Manuscript received July 15, 2012; revised September 5, 2012

region of interest (ROI) are faster, more efficiently and more naturally performed using touch screens and advanced finger gestures.

The novelty of our proposed platform comes from the possibility to use touchscreen-based mobile devices for manipulating, analyzing and registering multimodal medical images so that the advantages of each medical imaging technique will be exploited optimally.

The tremendous level of mobility that portable tablets offer could influence the way medical tracking and treatment planning is performed. New generations of physicians, surgeons and medical staff in general are used to have a personal digital assistant, so the adoption of such an application would be feasible.

All these integrated medical functionalities represent a true necessity at this particular moment when powerful processors enabled mobile devices emerge.

III. PROPOSED ARCHITECTURE

We designed MIPiOS as a three-tier application having:

- A mobile client
- A server tier
- A multimedia medical relational database

Figure 1 presents the conceptual architecture of our application with its three tiers.

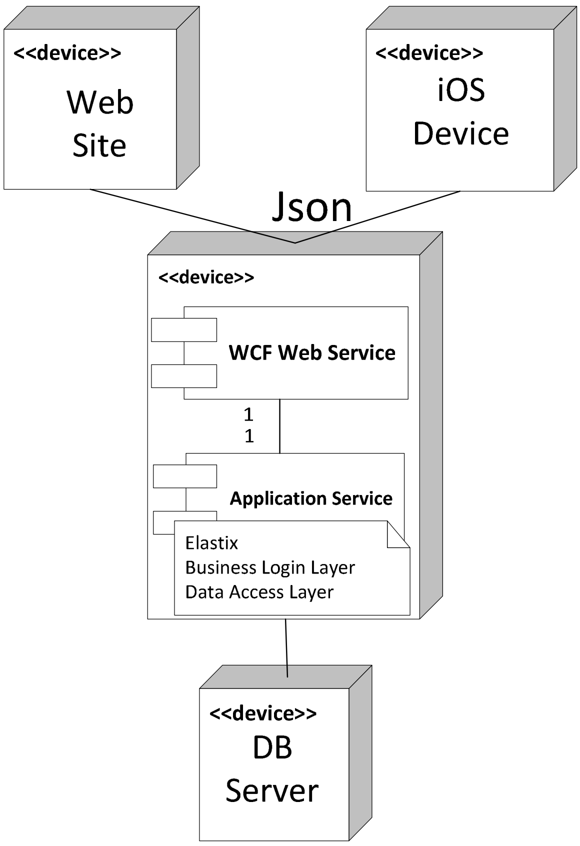

Figure 1. Conceptual architecture of MIPiOS

Auxiliary, there is a web browser backend application for managing users and user rights and performing CRUD (create, read, update, delete) operations on medical data.

III.i The client tier

The mobile client is an Objective C application developed for iOS based mobile devices. We focused our development on the iPad as it offers us more user controls than the iPhone approach. It should be stated that the development of an iOS based application must comply with the iOS Application Programming Guide [5] requirements. Every application needs the final acceptance of Apple, which is the creator of iOS and the producer of iPad. Therefore, the look and feel of the graphical user interface should respect the vision and the strategy of the producer of iOS.

For DICOM processing, we used DCMTK6] which is an API for basic operations compatible with iOS 4+.

III.ii The server tier

The server tier was developed using Microsoft .NET 4.0. It contains a backend application and a windows service in charge of communicating with Elastix [7] , an open source medical image registration extensible framework developed over [8] by Klein, Starling et al.

We use Microsoft's Windows Communication Foundation (WCF) platform to enable our core tier as a Service Oriented Architecture (SOA) module. The communication between the server and the client is made using Javascript Object Notation (JSON) and Representational State Transfer (REST) services. The reason for which we are using JSON and REST instead of XML and SOAP is the simplicity of the former. Hence, a non-negligible reduction of the traffic between the client and the server is assured. This is of real interest especially when GPRS, EDGE, HSPA or newer LTE connections are used and when the cost of the communication is traffic amount dependent.

We also developed a windows service that gives commands to Elastix. We have already used Elastix in our previous papers [9], [10] to register multimodal medical images. It is also an extensible framework that permits a kind of collaborative work between researchers engaged in medical image processing. Elastix takes commands using the standard Microsoft Windows terminal. The configuration of all Elastix components is made using a configuration file. This file is dynamically composed when a registration command is given using the mobile client. The configuration settings are controlled using the mobile application.

The Bussines Logic Layer performs operations on images, but also on parameters' set and users' actions.

The Data Acces Layer is responsible for assuring the communication with the database. We used Entity Framework [11], an Object Relational Mapping (ORM) technology developed by Microsoft that perfectly integrates with our database type.

III.iii The database tier

A medical image processing application needs data to be stored and retrieved in a secure and confidential manner. Our approach was to use a Microsoft SQL Server 2008 database for storing meta-information about medical data and to use a separate file-based repository for storing the effective sensitive medical data (e.g. DICOM files).

III.iv The hardware architecture

Figure 2 presents the proposed hardware architecture. As stated before, there is the principal client module used by the medical staff and the back-office application available via a web browser.

Both the mobile application and the back-end module communicate with the web server using JSON. We have also implemented a SOAP/XML communication alternative for the back-end part with the idea that this could facilitate the integration with a desktop based application for medical data management. Once recovered, these medical data should be stored in MIPiOS database by using the WCF services provided. The web server used is Internet Information Services (IIS) 7.0.

The application service is installed on a separate machine for security reasons and is included in the same private network as the webserver and the database server. Elastix is installed on the same physical machine and thus, it could be more convenient to call Elastix using local commands. It should be stated that our architecture permits theoretically the integration with other specific image registration frameworks like MIRT[12], FLIRT [13] with minimal modifications.

For the database part, we used Microsoft SQL Server 2008 and designed a schema that is comparable with the Data Transfer Objects (DTOs) diagram presented in figure 3. In fact, each DTO maps to a table in the database schema.

The development was done using Lion OSX and Xcode 4.0 for the iPad application and Microsoft Visual Studio 2010 under Windows 7 for the other parts of the application.

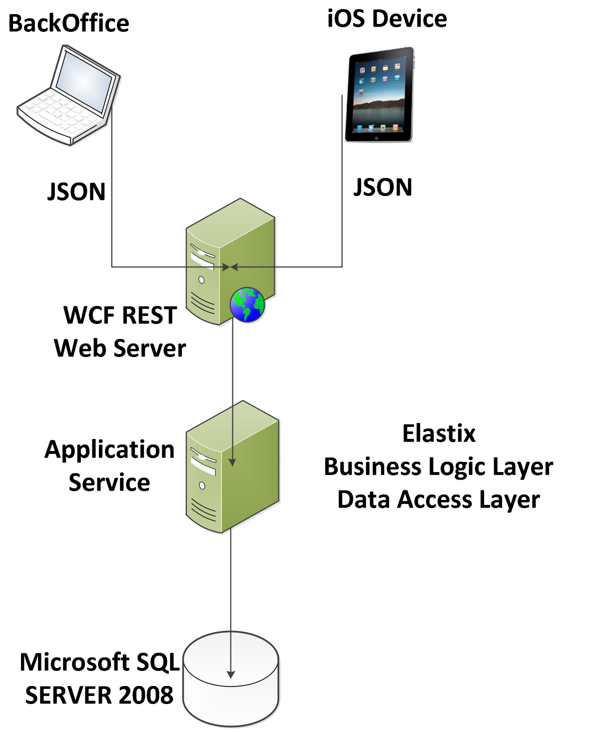

Figure 2. Hardware architecture for MIPiOS application

III.v The class diagrams

The communication between the iOS based mobile devices application and the image processing module is performed using REST web services. Every request made by the client to the web service is modeled using an object of type *ImageRegistrationRequestDto*. This object has the following components: the reference image (called FixedImage of type *ImageFileDto* in figure 3), the floating image (called MovingImage of type *ImageFileDto* in figure 3) and a set of parameters of type *PatameterDto* grouped in several categories of type *CategoryDto.*

All the parameters are defined by the *name* (attribute ParameterName), the *category* (attribute CategoryId), the *type* (attribute TypeId as reference to the table of parameters type – string, boolean, and list of options), the *default value* (attribute DefaultValue) and, finally, the *actual value* (attribute Value). For the parameters of type "list of options", the choices are modeled as an object of type *ListDto* containing elements of type *ElementDto* defined by an identifier and a value.

Designing an application that runs on a mobile device does not differ fundamentally from any other application when Object Oriented Programming (OOP) is in discussion. In figure 4 we present the conceptual diagram of the client application supposed to run on the iPad 3 Retina Display tablet.

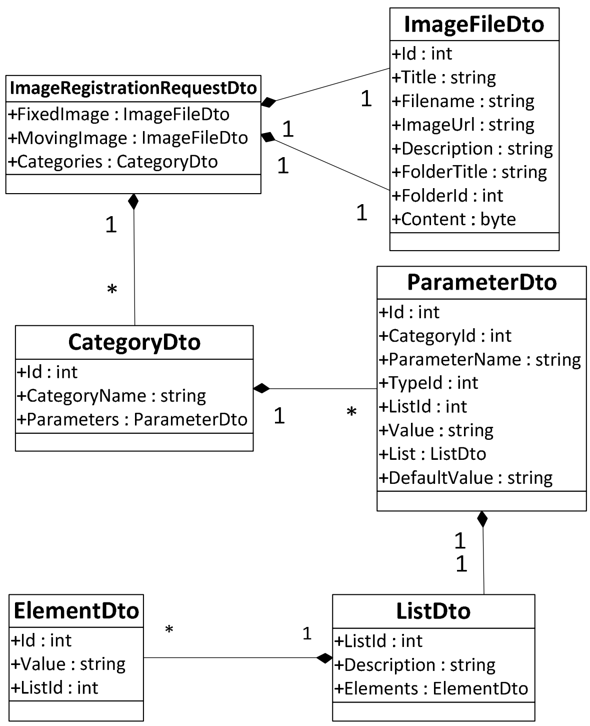

Figure 3. Data Transfer Objects (DTO) diagram for communication between MIPiOS and Elastix

The classes were grouped in three areas, depending on the functionalities in which they are involved: *analyzing and manipulating medical images, CRUD operations on medical images, medical image registration*.

For analyzing and manipulating images (region Image Manipulation Classes in figure 4), we considered all the classes related to rotation, translation and zoom operations (*ImageProcessingViewController*) as well as the classes for performing direct measurement on the medical images (*ImageMeasurer* and *MeasurerView* classes).

For managing medical data files, we designed this operation both locally on the mobile device (class *LocalFileManager*) and remotely on the server (class

ServerFileManager). The conceptual model takes into consideration the possibility to group files in folders (*DICOMFile* and *DICOMFolder* classes) and the possibility to navigate through them (class *FileBrowserViewController*).

For the image registration part, we used three types of parameters: *BooleanParameter, ListParameter* and *NumericParameter*. Client-side numeric type corresponds to the string of characters on server-side. This is due to the fact that Elastix parameters files are raw text files. Generally, it should be created a wrapping type between mobile client application parameter types and any other registration framework used at the Application Service level.

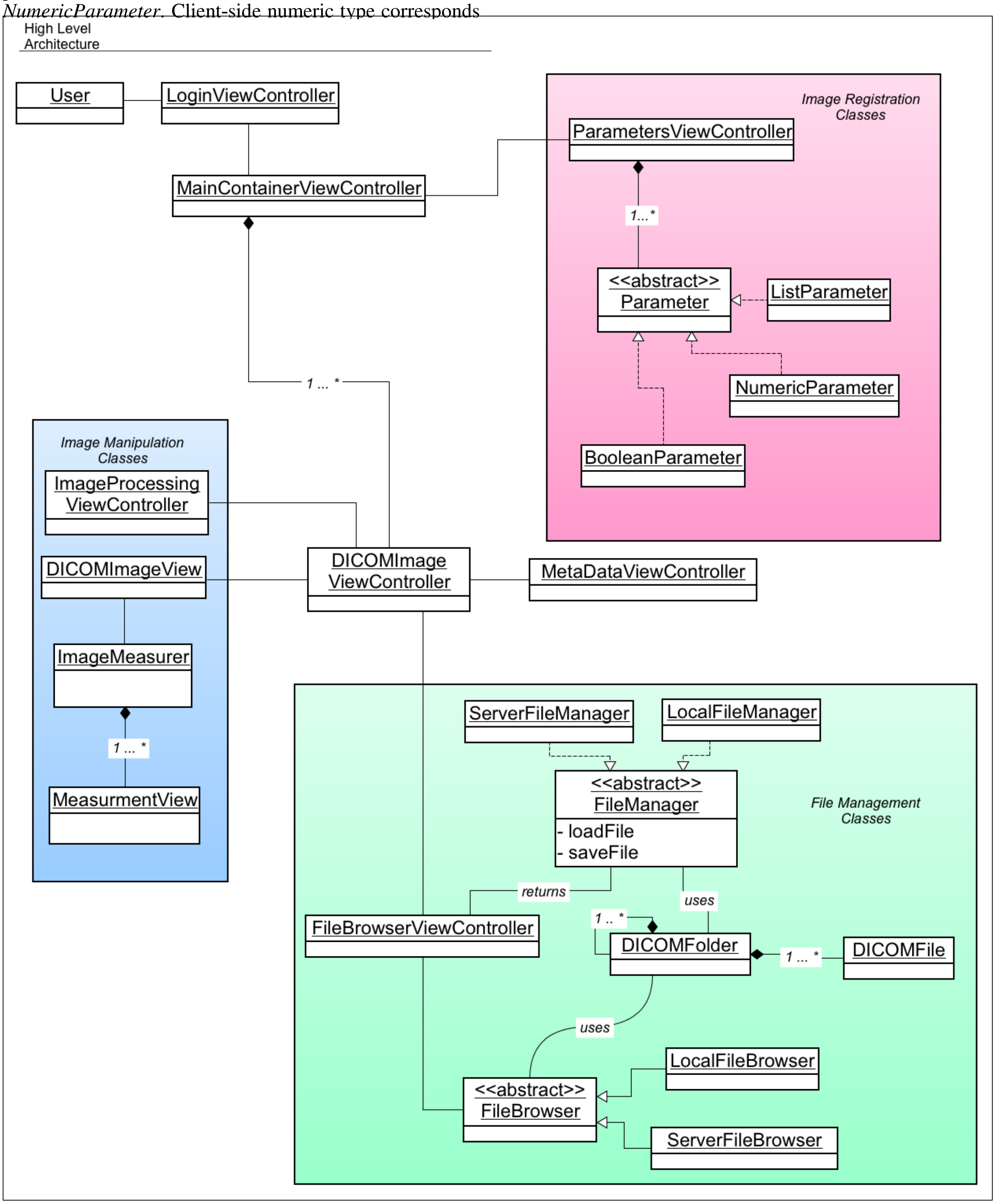

Figure 4. Class diagram for iOS based mobile device client

IV. PROPOSED FUNCTIONALITIES

We developed MIPiOS keeping in mind that it is an application whose purpose is to serve for medical use. The application has two major groups of functionalities: analyzing and manipulating multimodal medical images, registering multimodal medical images.

All devices that run iOS (iPad, iPhone, iPod) offer finger gestures interaction. With the addition of Retina Display which is the screen with the highest density of pixels (i.e. 78 micrometers wide pixels), MIPiOS could be capable of providing the best experience for medical image manipulations.

One could argue the fact that the small size of the iPad's display could be an impediment in properly using this mobile device for medical purposes. It should be stated that the majority of the human anatomical organs are of size smaller than that of the iPad. Moreover, when zooming in a region of interest, it is the quality of the acquired image and the density of pixels on the screen that really matter rather than the actual size of the display. Besides these, there were already reported experiments using iPad during surgical interventions for real-time patient monitoring (medical images and vital parameters).

For the moment, MIPiOS offers the following functionalities:

- Retrieving, opening and displaying medical images in DICOM format. All sensitive medical data are stored on a secure file server with meta-data saved in the relational database. MIPiOS offers the possibility to visualize a volume slice by slice (see figure 5).

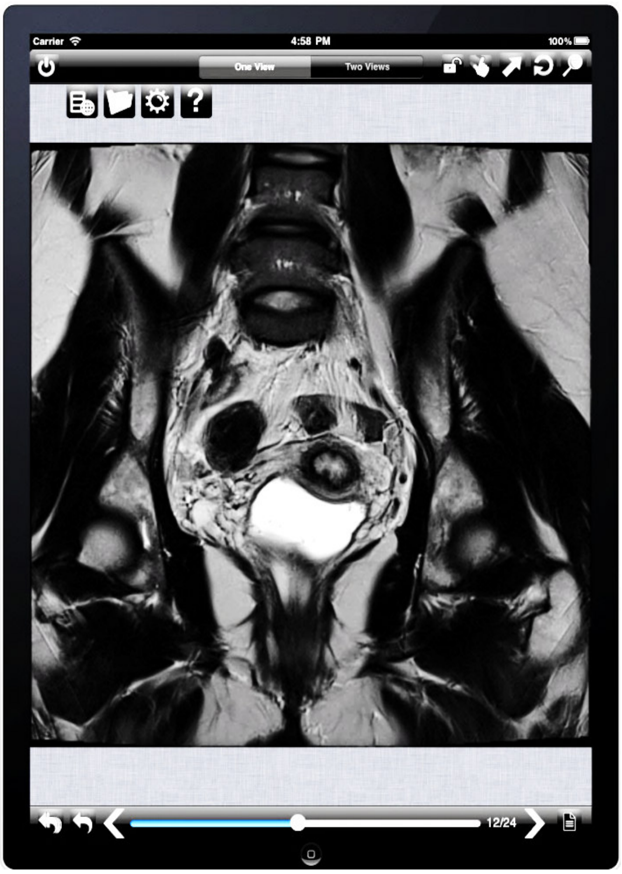

Figure 5. Displaying an MRI slice in the coronal plane using MIPiOS

- Analyzing ROIs using rotation, translation and zoom. Finger gestures are ideal for these operations. They better suit these operations than standard manipulation using classical PCs with keyboard and mouse. It is possible to cancel these manipulation operations one by one or all at a time (see figure 5).
- Performing direct measurements on the images for evaluating the ROIs (anatomical organs, tumors etc.). In figure 6 we present this operation that we consider to be a natural and helpful option. We have designed it to permit up to three simultaneous measurements. DICOM format contains metadata related to the stored image such as the spatial dimensions of the pixel/voxel. In this way, we can exactly measure spatial distances taking into account the pixel's size and the level of zoom applied.

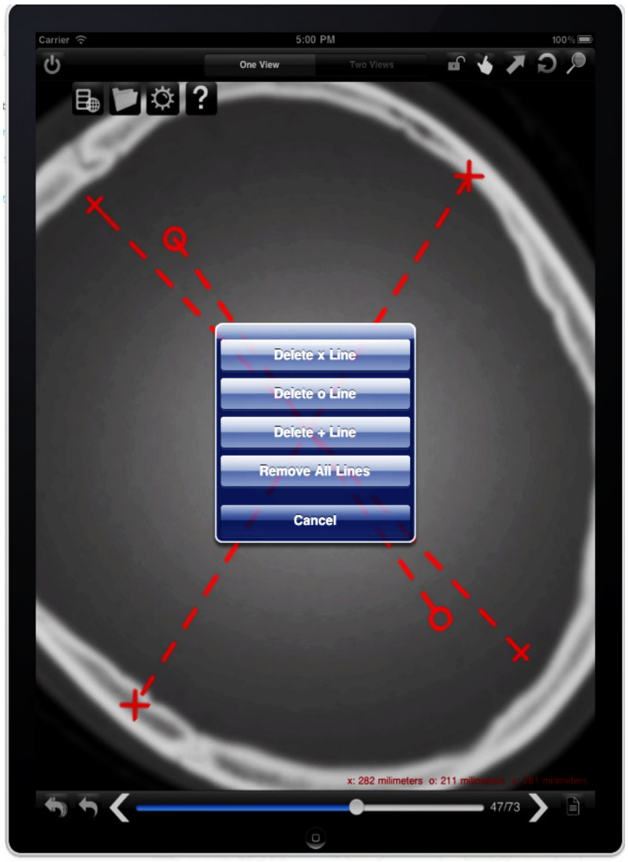

Figure 6. Multiple measurements over the same image. MIPiOS takes into account the pixel size and the level of zoom

- Medical image registration and data fusion is a functionality that is still in development. Our approach is to integrate and extend multiple registration frameworks for medical images. For the moment we integrated Elastix. Figure 7 briefly presents how the registration parameters can be configured using the mobile application. The reason for which we opted for a remote registration is the fact that during the registration operation, iterative optimization algorithms are used, and hence, very powerful processors are needed to perform these operations.

V. CONCLUSION AND FUTURE WORK

In this paper we presented MIPiOS, a medical image processing application using mobile devices based on iOS. Finger gestures and high resolution touchscreens are key elements that could change the way doctors, surgeons and physicians practice their profession.

Our application brings three major advantages. Firstly, medical images can be more conveniently manipulated using top-technology touchscreens and the best screen available at the moment (Retina Display from Apple). Secondly, a mobile device oriented medical application copes with the lack of specialists in rural areas. Diagnosing, driving and tracking the entire lifecycle of medical processes would be easier with this application that permits immediate access to all patients' data stored in a medical data repository. Thirdly, the fusion of medical data is an important trend nowadays and we follow it by performing medical image registration.

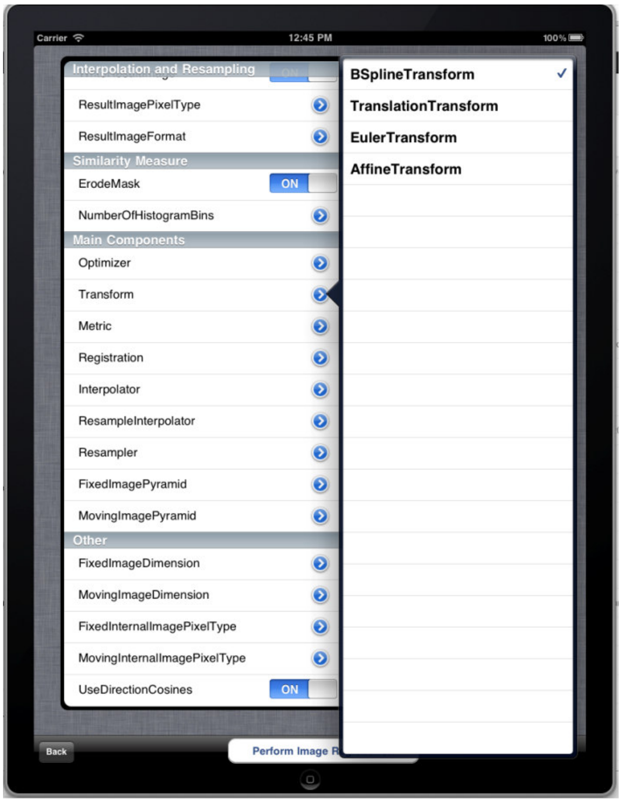

Figure 7. Setting parameters for medical image registration using MIPiOS

Future work on MIPiOS will try to integrate many other image registration frameworks with particular interest for those capable of running on a multi-processor and in distributed manner. Moreover, we will try to develop and integrate new elements for image registration and image fusion, we will try to use more powerful optimization algorithms to reduce the delay between the command for registration and the final delivery of the results.

ACKNOWLEDGEMENT

This paper was supported by the project "Doctoral studies in engineering sciences for developing the knowledge based society- SIDOC" contract no. POSDRU/88/1.5/S/60078, project co-funded from European Social Fund through Sectorial Operational Program Human Resources 2007-2013 and by a grant of the Romanian National Authority for Scientific Research, CNDI – UEFISCDI, project number PN-II-IN-CI-2012-1-0059.

REFERENCES

[1] DICOM Standard. http://dabsoft.ch/dicom/ [2012, July]

[2] OsiriX Imaging Software. http://www.osirix-viewer.com/AboutOsiriX.html [2012, July]

[3] Luigi Orso. iTunes.. http://itunes.apple.com/us /app/lodicom-viewer/id435701295?mt=8 [2012, July]

[4] University of Texas Health Science Center. Mango for the iPad.. http://ric.uthscsa.edu/mango /imango.html [2012, August]

[5] Apple Inc. iOS App Programming Guide. web site. http://developer.apple.com/library /ios/#DOCUMENTATION/iPhone/Conceptual/iPhoneOS ProgrammingGuide/Introduction/Introduction.html [2012, July]

[6] DCMTK - DICOM Toolkit. http://dicom.offis.de /dcmtk.php.en/ [2012, July]

[7] S. Klein, M. Starling, K. Murphy, M.A. Viergever, and J.P.W. Pluim, "Elastix: a toolbox for intensity-based medical image registration," IEEE Transactions on Medical Imaging, vol. 29, no. 1, pp. 196-205, 2010.

[8] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, The ITK Software Guide, 2nd ed.: Kitware Inc., 2005.

[9] M.C. Ureche, M.F. Vaida, and L.D. Chiorean, "Nonrigid multimodal medical image registration using features extracted from the monogenic signal," Proceedings of the 3rd International Conference on E-Health and Bioengineering - EHB 2011, pp. 395-398, November 2011.

[10] M.C. Ureche, A. Basarab, M.F. Vaida, and D. Kouamé, "Evaluation of monogenic phase apport in Ultrasound-MRI image registration," Proceedings of Telehealth 2012, pp. 793-799, Feb. 2012.

[11] Microsoft. Entity Framework. http://msdn. microsoft.com/en-US/library/aa697427(v=vs.80) [2012, July]

[12] Andriy Myronenko. Medical Image Registration Toolbox. https://sites.google.com/site/myronenko /research/mirt [2012, July]

[13] Bernd Fischer and Jan Modersitzki, "FLIRT: A flexible image registration toolbox," in Proceedings of the Second International Workshop in Biomedical Image Registration, WBIR 2003, 2003, pp. 261-270.