HAL
open science

Visualization Software for Real-time, Image-guided Therapeutics in Cardiovascular Interventions

Stefan Pintilie, Labonny Biswas, Kevan Anderson, Sandy Dick, Graham Wright, Perry Radau

To cite this version:

Stefan Pintilie, Labonny Biswas, Kevan Anderson, Sandy Dick, Graham Wright, et al.. Visualization Software for Real-time, Image-guided Therapeutics in Cardiovascular Interventions. CI2BM09 - MICCAI Workshop on Cardiovascular Interventional Imaging and Biophysical Modelling, Sep 2009, London, United Kingdom. pp.141-148. inria-00417831

HAL Id: inria-00417831

https://inria.hal.science/inria-00417831v1

Submitted on 17 Sep 2009

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Author manuscript, published in "CI2BM09 - MICCAI Workshop on Cardiovascular Interventional Imaging and Biophysical Modelling (2009) 141-148"

inria-00417831, version 1 - 17 Sep 2009

Visualization Software for Real-time, Image-guided Therapeutics in Cardiovascular Interventions

Stefan Pintilie[1], Labonny Biswas[1], Kevan Anderson[1], Sandy Dick[1,2], Graham Wright[1,3], and Perry Radau[1]

[1] Imaging Research, Sunnybrook Health Sciences Centre
[2] Cardiology, Sunnybrook Health Sciences Centre
[3] Medical Biophysics, University of Toronto

Abstract. This paper introduces RtViewer, a four-dimensional (3D + time) real-time visualization software for guiding cardiovascular interventions that is open source and freely available. RtViewer was designed to be part of a pipeline that can connect it to a magnetic resonance imaging (MRI) scanner, actively tracked catheters, and navigational devices. The architecture and features of RtViewer will be described with examples of guiding percutaneous cardiovascular interventions. The paper concludes with a brief description of the work in progress on the next generation of this platform, named Vurtigo.

1 Introduction

Magnetic resonance imaging (MRI) has been used primarily as a diagnostic tool in clinical practice and has recently been applied to the guidance of interventional procedures with the development of rapid imaging acquisition protocols. In practice, real-time MR imaging refers to frame rates greater than a few frames per second (FPS), considerably faster than the several minutes required to acquire a gated 3D cine volume. Guidance by real-time MRI is attractive compared with x-ray fluoroscopy, because MRI has better soft tissue contrast and is capable of displaying ischemic, infarcted or even arrhythmogenic tissue that impacts interventional decisions. Furthermore, MRI is not a source of harmful radiation which is a concern for long procedures under x-ray fluoroscopy. [23]

The RtViewer software presented in this paper is designed to enhance the use of MRI for cardiovascular therapeutic interventions.[16, 12] Frameworks for various image-guided applications such as surgery [19–21], and specifically for MRI-guided, percutaneous cardiovascular interventions [18] have been previously described. The RtViewer represents an attractive alternative, and is portable to a variety of systems outside of the communication system for which it was developed.

RtViewer provides the user with a roadmap, i.e. 3-D visual context, for the 2D real-time images by means of a cine volume that is acquired just before real-time scanning begins. The orientations of the volume and real-time slice are known so that they can be rendered together in proper spatial alignment. The

inria-00417831, version 1 - 17 Sep 2009

software also provides a feature for overlaying the position of an actively tracked interventional device that is agnostic of the tracking system hardware. Examples of navigating a catheter to a porcine model of peripheral chronic total occlusions (CTOs) will be described.

2 Architecture

This section of the paper will discuss both RtViewer and the communication system connecting it to the MRI scanner. RtViewer is intended to be used within this environment although it can be executed separately if real-time updates are not required.

2.1 Communication System Design

The communication system[17] is composed of several pieces of software that communicate over TCP/IP sockets between themselves and the MRI scanner. The central piece is the Geometry Server, that serves as a storage location for the most recent information. Multiple clients can connect to the server simultaneously, sending and receiving information, and all server data will remain synchronized. The Geometry Server contains several types of data including images, orientations of the image plane(s), respiratory phase and trigger delays, and catheter parameters.

Between the MRI scanner and the Geometry Server is RTHawk [1, 2]. RTHawk is both a 2D viewer and a real-time MRI control system. It provides an interface to MRI data acquisition. This communication system was tested on GE 1.5 T Signa Excite 12.0 and 14.0 systems, although it is theoretically compatible with all MRI systems supported by RTHawk.

All other applications, including RtViewer, are connected directly to the Geometry Server and as such are independent of MRI scanner architecture. Clients such as the RtViewer are able to passively read the scan plane orientation from the server, or drive the location of the scan plane. Similarly, another client is a robotic arm that can either display the scan plane (passive mode), or be used to actively navigate to the correct scan orientation (drive mode) [13, 14]. The robotic arm can be used as a complement to the RtViewer, with one or the other in drive mode.

2.2 RtViewer Design

The RtViewer base was written in Python[7], a high level language that simplifies prototyping and portability across operating systems. It makes use of several powerful development libraries. Qt 4.4 [3] was used to develop the Graphical User Interface (GUI). Kitware's Visualization Toolkit (VTK) 5.0 [5] was used for rendering. Both libraries are fast, stable, open source and have a very large user base that improves the quality of the code. Both libraries are written in C++ and wrapped for use in Python. The ATAMAI 3D medical visualization library [6], written completely in Python, provides additional functionality.

inria-00417831, version 1 - 17 Sep 2009

RtViewer has a threaded architecture to allow both real-time updates and responsiveness to user interaction. The core of the viewer uses three processing threads, each dedicated to a separate task: user input, Geometry Server updates, and rendering.

The input thread is under the control of the Qt (C++) event passing system for all user interface elements, and is responsible for processing the user's direct interaction with the system.

Updates from the Geometry Server, including image updates, are received by a Python thread. This thread will store new information locally until the render thread is ready to display it.

The rendering thread, the most computationally intensive part of RtViewer, is a Python thread that calls VTK to render different windows when their contents change. Although this is a single thread in Python, VTK can split off multiple C++ threads to take advantage of additional processors and improve rendering speed.

3 Features

The many features in RtViewer are geared toward MRI guided visualization for cardiovascular interventions. In this section we discuss some of these features.

3.1 2D Visualization

The MRI scanner can obtain image updates alternately from two 2D slices. The two scan planes are completely independent and each can have a different orientation, position, or field of view (FOV). RtViewer is capable of displaying and updating these two real-time scan planes simultaneously, and within a prior volume visualization.

3.2 3D/4D Visualization

Non-real-time MRI pulse sequences can produce high quality 3D or gated cine (4D) datasets. After export of these files in Digital Imaging and Communications in Medicine (DICOM) format, RtViewer is able to read them with accompanying metadata describing the image and its orientation. RtViewer has three display techniques for this content.

The mode that renders the fastest is a set of three orthogonal planes that cut the volume, and may slide along each axis or be tilted. The 3-Plane view renders quickly even when manipulating the plane locations because each plane is texture mapped. A stack of DICOM slices can also be rendered as a volume where ray tracing is used for rendering and opacity is determined by image intensity. RtViewer can also display a Maximum Intensity Projection (MIP), a type of ray tracing that emphasizes the high intensity regions of the volume. The gated cine volumes (4D) can be played in a 3D movie loop, or, if real-time cardiac phase information is available with the real-time 2D images then the movie will be synchronized by cardiac phase. The rendering of the 4D dataset can also be achieved by any of the three ways that a 3D dataset can be visualized.

inria-00417831, version 1 - 17 Sep 2009

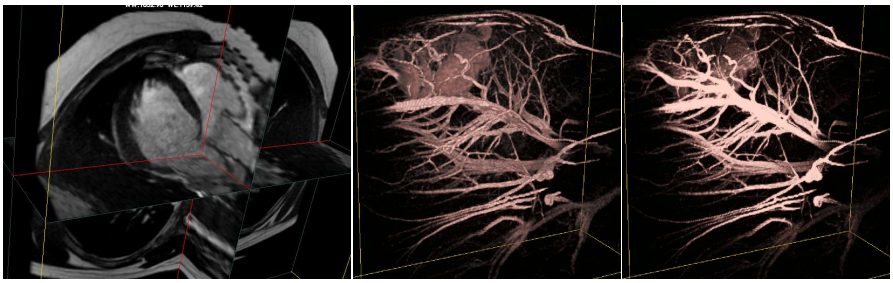


Fig. 1. From left to right: 3-Plane, Volume and MIP rendering.

3.3 Roadmap

The real-time images represent the current situation at a lower resolution while the volume provides a high resolution spatial guide. Since the prior volume data and the real-time slices are obtained on the same scanner during the same exam, these datasets will be spatially aligned. Rendering the volume and an interventional device together improves the cardiologist's ability to navigate with respect to the patient's anatomy. During the procedure the device location is updated in real-time from the Geometry Server. The viewer can also receive real-time updates of the 2D images. The images and a graphical representation of a device can be displayed in the same space as a previously loaded 3D volume.

Frame Rates For interventional guidance, rapid rendering of the display is very important. RtViewer was tested under different rendering conditions. The results presented are averages over 500 iterations. The 3D volume rendering and MIP ray trace had similar frame rates. The effect of instrument rendering (polygonal mesh) was not included in the table as it made little difference to the frame rates. All of the tests were performed on an 8 core, 32-bit MacPro (Apple Inc.) running Ubuntu 8.04 OS, 2 GB RAM and a GeForce 7300 GT graphics card. As expected, the MIP and volume rendering are the slowest. In the most computationally intensive case, RtViewer can still perform at 3 FPS.

Table 1. Rendering Frame Rates

3D Rendering(256×256×172)	Real-Time Planes(256×256)	Frames Per Second(FPS)
None	1	99.4 ± 6.4
None	2	55.8 ± 2.5
3-Plane	1	92.4 ± 6.2
3-Plane	2	52.3 ± 2.5
MIP/Volume	1	6.2 ± 1.2
MIP/Volume	2	3.1 ± 0.4

inria-00417831, version 1 - 17 Sep 2009

3.4 Plane Prescription

RtViewer has the ability to remotely control the scan plane prescription. If RTHawk is placed in read mode then it will allow client applications to control the location and orientation of the scan plane. RtViewer has both a "Drive" mode and a "Passive" mode. In "Passive" mode it will listen for new information from the scanner. In "Drive" mode the RtViewer interface allows the user to click and drag the plane to a new position or orientation in the 3D view. Fine control over plane geometry can be achieved through keyboard control. This will actively change the position of the scan plane. The scanner will acquire an image at the requested position and then pass the new image back to the Geometry Server to be read by RtViewer. This way of moving scan planes is more intuitive than sliders or line prescriptions and has the advantage that once a prior 3D volume is loaded it can be used as a guide to determine where the plane should be positioned.

3.5 Motion Compensation

If patient body motion is significant, then correction is required for use of the prior 3D volume as a roadmap with the real-time images. Furthermore, the respiratory and cardiac motion require correction as they may diverge from the pattern evident in the prior volume. RtViewer is equipped with an iterative rigid registration algorithm that will search through a series of in-plane rotations and translations for the image with the greatest similarity to align the real-time and prior volume images, and then apply the linear transformation to adjust the volume's location relative to the real-time image. [10]

In cases of gross motion of the subject, a rigid translation can be estimated using MR projection [11]. RtViewer can read the volume displacement from the Geometry Server and adjust the 3D rendering accordingly.

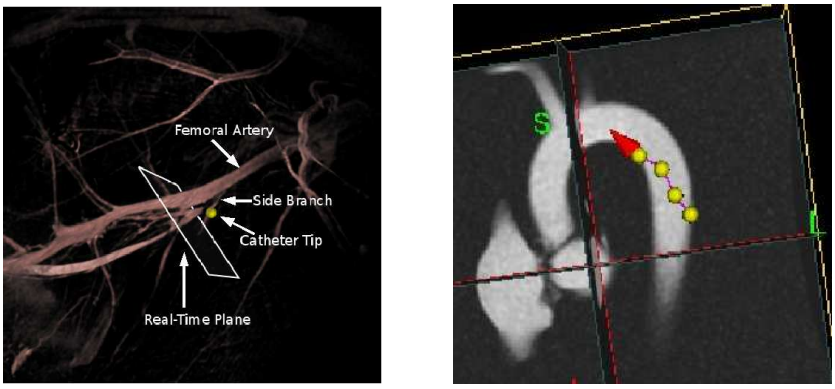
4 Interventional

Passive interventional devices can be difficult to identify on MR images. To track them, they can be modified to include one or more coils along their length. These coils can be actively tracked using an MR projection technique [15]. Depending on the number and location of tracked positions, RtViewer is able to display the tip of the device, the tip and the direction the device is facing, or a spline indicating the shape of a flexible device. With RtViewer in "Drive" mode, real-time planes can be attached to, or at an offset from, a tracked location on the device and automatically updated to follow the device's movement.

4.1 Example Use Case

To demonstrate the capabilities of the system, we advanced a 7F active catheter in a porcine model to the site of an chronic total occlusion (CTO) in the deep femoral artery. The active catheter consisted of two concentric and orthogonal

inria-00417831, version 1 - 17 Sep 2009



(a) Catheter tip tracking on a 3D angiography of the femoral artery in a porcine model. The square cutting through the vessel is a real-time scan plane.

(b) St. Jude Medical five-coil catheter tracked using GE's MR Tracking System in an aortic arch phantom

Fig. 2. Tracking two different catheter systems.

microcoils located at the tip that are used for tracking and forward-looking intravascular imaging [22]. A 5-inch-diameter surface coil was placed on the surface outside the body and used for larger field-of-view imaging. A contrast-enhanced angiogram of the peripheral vasculature was acquired (3D FSPGR, FOV=24cm, 0.75x0.75x1.0 mm), loaded into RtViewer and volume rendered to be used as a static "roadmap". During catheter tracking, the calculated position of the two microcoils was averaged and rendered as a ball within the volume. Intravascular real-time imaging was also performed in two orthogonal planes. One plane was just ahead of the catheter such that the vessel could be imaged in cross-section while the other depicted the vessel in-plane. The position of the two planes were fixed relative to the position of the catheter tip and their position was updated as the catheter was moved. The visualization could be seen by the operator inside the MR room via an MR compatible display. One could envision using real-time intravascular images and a static rendering of the vasculature visualized together in this manner to guide the revascularization of chronic total occlusions.

5 Work in Progress

Based on the knowledge and experience gained from RtViewer the program is being rewritten entirely with an improved design. Vurtigo, currently in its alpha development stage, is being written in C++ with a plug-in based architecture. Like its predecessor, Vurtigo is free and open source. Unlike RtViewer which runs mainly on Linux, the Vurtigo project uses CMake [4] as the build system and runs on Win32, Linux and MacOSX. Another major advantage to Vurtigo

inria-00417831, version 1 - 17 Sep 2009

is its plug-in design which provides a modular and easily extensible framework for developers. The RtViewer framework required a developer to have a good understanding of the software before any new feature could be implemented. Vurtigo gives developers a plug-in interface that is easy to interact with. It is our hope that Vurtigo users may implement their own features thus expanding the code base and appeal of Vurtigo. The switch to C++ was motivated by three considerations: C++ is faster than Python for operations such as loops, C++ threads can take advantage of multiple processors, and C++ can use a wide range of external libraries without requiring wrappers.

5.1 Design

The Vurtigo system can be conceptually separated into the core and the plugins. The core comprises the Graphical User Interface (GUI) as well as storage and rendering of a dynamic set of 'objects'. Examples of objects are: a transformation matrix, a 3D volume, a polygonal mesh, and a set of catheter points. Plug-ins add functionality by interacting, adding, removing, or editing these objects. Plug-ins can also provide a configurable user interface. Most of the features of Vurtigo will be implemented as plug-ins. Plug-in developers need not worry about how rendering will occur, what other plug-ins exist, what other objects exist, or managing objects in memory. As objects are modified, the viewer will re-render as needed.

5.2 Features

The majority of the features in RtViewer are currently being written into Vurtigo including the ability to display and update 2D scan planes in real-time as well as render tracked devices. But Vurtigo is not limited to the features that exist in RtViewer; it is already able to simultaneously render multiple volumes, multiple scan planes and multiple tracked devices. A plug-in allows it to communicate with the Geometry Server and a plug-in to communicate via OpenIGT Link is being planned. While RtViewer is primarily used for MRI, Vurtigo will be a multi-modality real-time system. For example, it can be adapted for fluoroscopy or ultrasound guided procedures. Since Vurtigo uses generic objects, developers can easily import custom instruments as polygonal meshes.

6 Conclusions

In this paper we have presented the open-source, freely available RtViewer, that is a software tool for MR-guided cardiovascular intervention. In addition we have described the work in progress for the next generation of this software, called Vurtigo.

7 Acknowledgements

The authors would like to thank the Canadian Foundation for Innovation and Canadian Institutes of Health Research for their support.

inria-00417831, version 1 - 17 Sep 2009

References

1. Santos JM, Wright GA, Pauly JM; Flexible Real-Time Magnetic Resonance Imaging Framework; 26th Annual Int. Conference IEEE EMBS, 1048, 2004.
2. Santos JM, Cunningham CH, Lustig M, Hargreaves BA, Hu BS, Nishimura DG, Pauly JM. Single Breath-Hold Whole-Heart MRA Using Variable-Density Spirals at 3T. Magn Reson Med, 55:371-379, 2006.
3. Blanchette J, Mark Summerfield M. C++ GUI Programming with Qt 4, 2nd Ed., 2008.
4. Martin K, Hoffmann B. Mastering CMake, 4th ed., 2008.
5. Schroeder W, Martin K, Lorensen B. Visualization Toolkit: An Object-Oriented Approach to 3D Graphics, 4th Ed., 2006.
6. Atamai Interactive Visualization. www.atamai.com
7. Rossum G, Drake FL, Jr. (Editor). An Introduction to Python. 2003.
8. Thread State and the Global Interpreter Lock. http://docs.python.org/c-api/init.html#threads
9. Boost C++ Libraries. http://www.boost.org/
10. Chung D, Wright G, Radau P. Real-time 2D MR scan-plane alignment during guidance of cardiac interventions. RSNA (InfoRAD), 2005.
11. Anderson KJT, Biswas L, Chung D, Huang J, Wright GA. Correcting Static Interventional Roadmaps for Subject Displacement Using One-Dimensional Projections. E-Poster 3393, Proc Intl Soc Mag Reson Med, 2007.
12. Radau P, Hu N, Stainsby JA, Wright GA. Visualization software for scan plane navigation in real-time cardiac MRI. RSNA (InfoRAD), 2004.
13. Yi D, Wright GA, Hu B. Device and process for manipulating real and virtual objects in three-dimensional space. US Patent Application 10/776,421. 10 Feb. 2004.
14. Yi D, Radau P, Wright GA. Visualization and surgical planning using a cost-effective, six degree-of-freedom (6-DOF) navigator. RSNA (InfoRAD), 2004.
15. Dumoulin CL, Souza SP, Darrow RD. Real-time position monitoring of invasive devices using magnetic resonance. Magn Reson Med. 29, 411-415, 1993.
16. Radau P, Hu N, Stainsby JA, Wright GA. Four dimensional visualization system for real-time scan plan and catheter navigation. Proc Intl Soc Mag Reson Med, 2004; 11:375.
17. Stainsby JA, Hu N, Yi D, Radau P, Santos JM, Wright GA. Improved visualization and control for scan plane navigation in real-time cardiac MRI. Proc Intl Soc Mag Reson Med, 2004; 11:537.
18. Guttman M, et al. Real-time accelerated interactive MRI with adaptive sense and unfold. Magn Reson Med. 50,315-321, 2003.
19. Carter TJ, Tanner C, Crum WR, Beechey-Newman N, Hawkes DJ. A framework for image-guided breast surgery. In: Yang, GZ, et al. (eds.) MIAR 2006. LNCS, vol. 4091, pp.203-210. Springer, Heidelberg, 2006.
20. Balazs V, Simon D, Anton D, Peter K, Rajesh K, Christopher H, Russell T. The Surgical Assistant Workstation. Midas Journal, paper ID 624.
21. Gary K, Ibanez L, Aylward S, Gobbi D, Blake M, Cleary K. IGSTK: An Open Source Software Toolkit for Image-Guided Surgery. IEEE Computer, 2006; pp 46-53.
22. Anderson KJ, Leung G, Dick AJ, Wright GA. Forward-looking intravascular orthogonal-solenoid coil for imaging and guidance in occlusive arterial disease. Magn Reson Med. 2008 Aug;60(2):489-95.
23. Ladd ME, Quick HH, Debatin JF. Interventional MRA and intravascular imaging. J Magn Reson Imaging 12(4), pp. 534-46, 2000.