Integrated Medical Workflow for Augmented Reality Applications

[1]Denis Kalkofen* [1]Bernhard Reitinger [2]Petter Risholm [1]Alexander Bornik [1]Reinhard Beichel
[1]Dieter Schmalstieg [2]Eigil Samset

[1]Graz University of Technology
Institute for Computer Graphics and Vision
Inffeldgasse 16a

[2]The Interventional Centre
University of Oslo

Abstract

Augmented Reality (AR) and Virtual Reality (VR) have the potential to improve surgical interventions and image-guided therapy by providing clinicians with interactive three-dimensional visualizations in all phases of treatment. However, experimental AR research applications typically focus on a single medical procedure and do not attempt to cover the whole clinical workflow from pre-operative to intra-operative and post-operative support. In the scope of the EU-funded Marie Curie research project ARIS*ER (Augmented Reality in Surgery), we are investigating a more complete workflow support with AR technology. This paper discusses a component-oriented software architecture which specifically aims at addressing the needs of medical AR, in particular strategies for ensuring uniform application quality throughout the medical workflow. We discuss the management of off-line and on-line image acquisition methods and methods for consistent management of modeling, processing and visualization of medical datasets throughout the medical workflow. Finally, we present three different medical application examples which illustrate the use of the framework.

CR Categories: D.2.6 [Programming Environments]: Graphical environments—Integrated environmentsInteractive environmentsProgrammer workbench H.5.1 [Multimedia Information Systems]: Artificial, augmented, and virtual realities— [I.3.7]: Three-Dimensional Graphics and Realism—Virtual reality

Keywords: Medical augmented reality; software solutions for augmented reality; medical applications of augmented reality

1 Introduction

The surgical workflow is commonly divided into three separate phases – pre-, intra- and post-operative. In the pre-operative phase clinical data is studied and a plan for the intervention is devised. Once a suitable approach for treatment has been identified, the plan is used to guide the intra-operative phase. This includes continuous revision of the plan when unforeseen evidence is encountered while surgically treating the patient. In the post-operative phase, quantitative and qualitative results of the procedure such as blood loss, resection margins, or deviations from the plan are reviewed and evaluated.

Throughout the workflow, image data is a major source of information for decision making. Virtual Reality (VR) and Augmented Reality (AR) aim at lifting this support to a new level by presenting more informative and realistic three-dimensional visualizations. AR is a relevant tool whenever dealing with the patient directly, in particular as a navigation aid during a surgical intervention. VR is useful when the patient is not available, for example during planning or when assessing ”what if” scenarios through simulation.

AR is a rather complex aggregate technology, and research on AR-guided treatment is typically targeted at a single phase or aspect in a surgical procedure. Embedding AR into a clinical workflow requires careful design of a software architecture that provides consistent services for image processing, visualization, tracking etc. throughout a variety of situations and requirements. The challenge here is to provide useful, high-performance AR components that are also sufficiently flexible to allow re-use across different applications. For example, re-using visualization algorithms across pre-operative planning and intra-operative navigation ensures that the physicians have a consistent view of the image data, and are not distracted by potential image artefacts introduced by a conversion from one visualization methods to another. Moreover, componentization also allows more careful testing of components in isolation (unit tests), and makes approval of software components for clinical evaluation more straight forward.

In the Marie Curie research project Augmented Reality in Surgery (ARIS*ER[1]), a consortium of research groups across Europe is developing user-friendly, fast and reliable AR technology for a number of clinical use cases. The intended applications have a wide scope of hardware, software and infrastructure requirements and rely on rapid prototyping based on input from the partner physicians.

In this paper, we describe a software framework for integration of AR in the clinical workflow, aimed at supporting ARIS*ER. It facilitates development of applications which are able to support physicians during all phases of medical treatment including pre-operative planning, intra-operative navigation support and simulation, and post-operative analysis. In the description of the system, we focus on those components that are of specific interest for medical AR, and describe some test applications built to evaluate the approach.

2 Related Work

AR systems cover a wide range of medical scenarios such as radio frequency ablation [21], laparoscopic liver resection [13, 9] and needle biopsy [12, 25]. Research in these areas focuses primarily on tracking and registration quality as well as specific visualizations in support of the medical procedure. The literature does not particular focus in the way of software architecture or re-use for different medical procedures.

Other groups have proposed frameworks for use in image guided procedures. For example, 3D Slicer [16] is an integrated visualization system which incorporates image registration, segmentation

*e-mail: kalkofen@icg.tu-graz.ac.at

[1] http://www.ariser.info

and visualization. It is mainly used for image analysis and planning, but has also been applied for intra-operative guidance [15]. MeVis[2] provides a well-known platform for diagnosis and treatment of cancer and diseases of the cardiovascular system, lung, and brain which is mainly aimed at surgery planning.

The Image-Guided Software Toolkit (IGSTK) [14] aims at system-level support for image-guided procedures. It contains components for tracking, reading medical images, registration and calibration and builds on the InsightToolkit (ITK) [18] and the Visualization Toolkit (VTK) [19]. IGSTK is noteworthy in its emphasis on software quality for quality assurance and clinical approval.

General purpose AR frameworks such as DWARF [1] or DART [20] are well suited for rapid prototyping of AR user interfaces, but lack some aspects necessary for medical applications. In particular, general-purpose communication between components such as the CORBA based event passing used by DWARF is not always suitable for the stringent real-time requirements of medical applications. Moreover, medical applications have specific demands concerning the visualization techniques, which must have higher quality than for many general AR applications. For example, many mobile AR applications rely on simple text labels, while medical AR requires volume rendering of medical datasets. Consequently, a framework specifically supporting medical AR is required.

3 Workflow Analysis

As based on the workflow analysis done by human interface designer of the ARISER consortium [17], a workflow was developed covering most aspects of surgical interventions: pre-operative planning, intra-operative navigation, and post-operative analysis. This workflow constitutes a template, from which more detailed descriptions for specific procedures can be derived. Figure 1 shows a flow diagram of the template workflow.

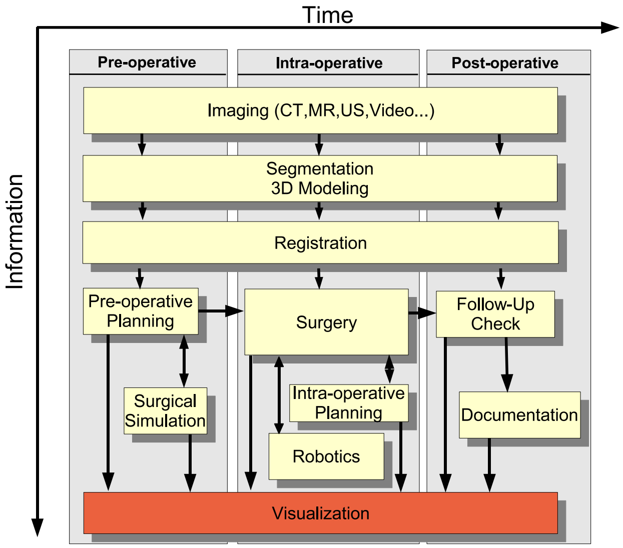


Figure 1: Medical Workflow

Each column in the diagram corresponds to a major stage in computer-aided surgery characterized by an independent workflow, which could be addressed using distinct applications. However, computer-aided surgery in an all-embracing manner can only be achieved by a system combining the stages to a pipeline.

There are several tasks in an image-guided surgery framework, which are shared across the stages of the medical workflow and are thus best addressed with a unified tool set. For example, input data from different modalities such as computed-tomography (CT), magnetic resonance (MR), or ultrasound (US) are used throughout the pipeline. The same is true for video acquisition from conventional or endoscopic camera sources. Segmentation algorithms for data preparation and the generation of 3D surface and volume models are difficult to implement in a robust and versatile way, and once available can be shared across stages. Registration and visualization should always be done consistently independent of stage and scenario.

It can be seen that the majority of tasks can be approached independent of the stage. What remains are the application specific task, which must of course reflect the requirements of the stage. At first, before the actual surgical intervention can take place, a detailed plan must be elaborated to generate all necessary information which is required during the actual surgery. Pre-operative planning is usually decoupled from the surgical intervention, since multiple scenarios can be evaluated a priori and optionally simulated using surgical simulators. Important aspects in pre-operative planning are for instance to determine relevant anatomical areas in pre-operative images or determine trajectories for surgical instruments such as biopsy needles.

When a suitable plan has been devised for a certain procedure, it can be put into effect by intra-operative guidance software. The plan can either guide a surgeon, or it can be used for automated treatment such as robot-assisted liver biopsy, instructing a robot where to insert a needle and which regions to avoid. The pre-operative plan may often be rendered invalid due to intra-operative discoveries – for example, additional tumors may be detected during tumor resection. Consequently, intra-operative updates to the plan become necessary to keep it consistent with the progress of the procedure in order to be able to use it for documentation later on.

The outcome of the surgery can be documented in various ways. All acquired data such as video, tracking of instruments, image streams from US as well as the updated surgical plan and custom annotations (e. g., by audio) from the surgeon constitute the documentation. This elaborate documentation can help in post-operative follow-up checks, and also has relevance for liability issues.

Our work concentrates on the recurring tasks for computer-aided surgical treatment. In the following, we describe a framework of components for image acquisition, tracking, video processing, synchronization, medical data modeling and rendering techniques which guarantee a consistent data representation throughout the clinical workflow.

4 Software Framework and Components

The work described in this paper builds on the *Studierstube* framework [26], an AR framework which has now been under development for 10 years. The most recent version of the framework, Studierstube Medical, features enhanced modularity and has been extended with a number of components specifically targeted at developing medical software applications such as LiverPlanner [5, 4, 23]. We identified the following main requirements:

- *Real-time data acquisition* In contrast to modalities such as CT or MR, which are are normally not acquired in real time, AR applications require the management of streaming input data such as tracking, US or video data. On the one hand, the handling of such data requires real-time algorithms and also careful synchronization, in particular between simultaneously acquired data from heterogeneous sources, which must be correlated. For example, consider simultaneous acquisition

[2] http://www.mevis.de

of endoscopic video and US images with tracked probes. On the other hand, data streaming must be flexible to permit reuse of the solutions across different applications. We present a software architecture for streaming data based on a pipes-and-filters pattern [8], which can fulfill these requirements and provides specific support for tracking and video data.

- *Consistent data models* The requirements of surgical planning, navigation and simulation go beyond the simple display of volumetric data acquired in previous image scans. For sophisticated computer-aided medicine, anatomical and pathological structures must be explicitly modeled and manipulated. Diverse application requirements make multiple redundant structures necessary – for example, the same sampled volumetric dataset may be used to compute a surface model of an organ, a graph model of the contained vessel structure and a polyhedral model of internal tissue. Physicians demand predictable and reproducible results, so all these representations must be kept in a consistent state throughout the medical workflow while permitting arbitrary operations on the data. Section 4.3 describes a medical modeling component which addresses this requirement.

- *Visualization* Compared to conventional screen-based navigation, AR has elevated requirements for visualization. It should support a wide variety of display devices with very specific properties, such as tracked head-mounted display (HMD), stereoscopic or video-see through configurations. The actual visualization techniques must be mostly independent of the viewing mode and display type, and must be able to simultaneously display all kinds of data models in real time, for example scanned volumetric images with direct volume rendering together with polygonal rendering of segmented surface models.

The requirements lead us to a design of a component-based system. This section describes the system's main components. The kernel itself and how it manages integration and synchronisation of the components is discussed in the following section.

4.1 Tracking

Real-time acquisition and streaming of tracking data and other multi-modal device data is done with a tool called OpenTracker described in detail elsewhere [24]. OpenTracker follows a dataflow graph architecture which can be distributed across multiple networked computers if desired. Source node for a large variety of tracking devices exist to acquire the device data. Filter nodes can further process the data, and sink nodes deliver the data to an application. Multiple components can register to receive data from a single source in OpenTracker, which allows to achieve synchronized behavior of multiple objects in the application.

4.2 Video

Video management is handled by a tool called OpenVideo, which implements a general dataflow system for video data based on a pipes-and-filters architecture [8] in a similar way OpenTracker is handling tracking data. However, in contrast to trackers and conventional sensors, video sources consume very high bandwidth and consequently require specific techniques to avoid undesirable behavior such as frame drops or stalling. Existing video libraries such as Microsoft DirectShow achieve such efficiency by platform-specific optimizations. Consequently, OpenVideo was designed as a wrapper library encapsulating efficient single-purpose video libraries and add a consistent scriptable interface and missing features such as network streaming, simultaneous delivery of video to multiple application objects, and a notification service for components interested in video updates.

OpenVideo encapsulates raw video data in video objects, which reside in shared memory and are passed around by reference. Video objects have a number of advantages over a fixed frame structure. They can be annotated with arbitrary meta data, allowing simple and experimental multi-modal data structures. For example, adding resolution and format description allow a receiver to deal with a variety of sources. A tracked camera or image probe can deliver a video stream where each frame is annotated with the corresponding pose information from the attached tracker. The video object also implements multi-buffer support with a buffer locking mechanism, which enables us to smoothly integrate and serve the video data from asynchronous threads.

OpenVideo supports three different strategies to trigger an update of its data structure. A timer with a pre-defined update rate can be used to guarantee updates in a controllable and – depending on the real time capabilities of the operating system – in an almost equidistant time frame. For a dedicated machine running OpenVideo, we have implemented support for using all available system resources to trigger updates of the graphical data structure. Finally, OpenVideo can be configured to wait for external signals to trigger dataflow graph traversal. The default traversal strategy is in topological order of the dataflow graph, i. e., a node is accessed after all its predecessor nodes have been accessed. Should the application require it, traversal order or dataflow graph can be modified at runtime by pausing the video stream.

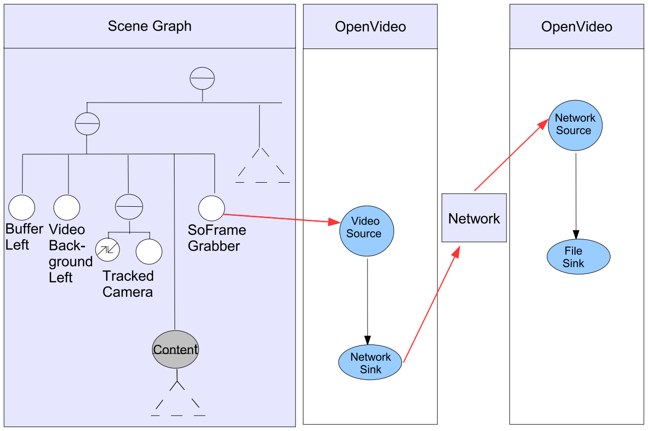


Figure 2: A sample setup for documentation of surgical procedures by coupling a frame grabber and OpenVideo.

Documentation via OpenVideo and SoFrameGrabber: One of the advantages of flexible handling of videos is the documentation of medical procedures using AR. A video augmented with computer graphics can be documented by adding a frame grabber node to which streams the grabbed frames to OpenVideo – potentially over the network to a dedicated video documentation server. OpenVideo acquires the video stream through the use of a VideoSrc and saves it to disk with a FileSink (Figure 2) .

4.3 Modeling

Digital models of data acquired from the patient are the central source of information in image-guided surgery. Raw volumetric data from scanners such as CT or MR provides reference imagery, but this data cannot always be used directly for the medical procedures. For example, automated volume estimation of anatomical structures requires segmentation of the relevant structures. Another example is the simulation of tissue deformation in real time, which requires the extraction of boundary and/or internal geometric structures for performance reasons.

The generation and storage of derived models must adhere to consistency requirements. The correspondence between original and derived data must be retained at all times, and conversions from one representation to another must only happen explicitly, to prevent accumulation of conversion errors. Using the same models and operations on the models throughout the entire workflow not only guarantees predictable quality of the data, but also accelerates implementation work through improved code reuse.

The following data models are used:

- **Raw image data** (voxel grid). Obtained from scanners such as CT, MR or US. Used as ground truth and for direct volume rendering.

- **Deformable surface model**. For segmentation or registration tasks, deformable surface models are the best choice, since they reduce algorithm complexity to the relevant boundary region of the dataset, and enable interactive processing speed. We use a simplex mesh [10] data structure, which is particularly suitable for fast deformation of large meshes, and also lends itself to real-time rendering.

- **Polyhedral model**: Surface models are unable to represent the properties of object interiors. This deficiency makes it necessary to use a polyhedral mesh (more precisely, a tetrahedral mesh). The implementation is based on simplicial complexes [11], which ensures consistency of the mesh and its contained substructures for all kinds of volumetric manipulation (for example, simulated removal of a tumor). The polyhedral model is designed to be annotated with additional information, for example, when deriving surgical plans. Rather than discarding information originally present in the mesh, for example as a result of surgical simulation, the polyhedral model retains all original information even after manipulation and is additionally able to store all operations applied to each individual volumetric element as annotations, for planning or documentation.

In contrast to ad-hoc conversions from one representation to another, our system of data structures builds on a common software infrastructure, and is able to convert between representations at any time. Conversions to and from voxelized data are by nature approximations, but the error is bounded by user-specified margins, and the conversion procedure is only used when the workflow demands it, typically in the beginning of the treatment when new image data of the patient has been acquired. The surface model and polyhedral model are always kept consistent, and switching between them does not degrade model quality at all. In fact the surface model is extracted from the polyhedral model on demand, and both models share the same vertices.

By using these data models in all stages of the workflow, algorithm improvements are implicitly shared by all applications, and misrepresentation of the data are avoided.

4.4 Visualization

The visualization component of our system is based on Coin[3], a scene graph library which implements the Open Inventor [27] standard. Through extension of Coin's class hierarchy of scene graph nodes, the special requirements of AR visualization can be addressed.

Visualization for AR needs to answer two questions, What to display and How to display it. The viewer component of our framework was designed to address the How. It can create one or multiple views of a scene, accommodating a wide variety of display options. These options include frame buffer configuration, camera parameters including camera calibration and real-time image warping, stereoscopic display modes, viewer and display tracking, and general purpose image composition, for example using video textures. Through configuration options it allows to accommodate all viewing conditions including VR, AR and conventional displays. For example, the same viewer is able to display a classic 4-panel view on a monitoring screen or a stereoscopic head-tracked video-see through display on an HMD(see Figure 4 for an example of a tracked video see through hmd). By completely decoupling viewing conditions from viewed content and application, the same features and rendering quality are simultaneously available to all applications.

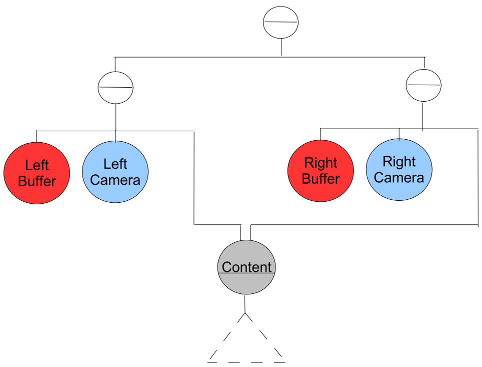


Figure 3: Scene graph setup for stereo viewing

This flexibility is achieved by encapsulating all relevant viewing commands affecting the viewing condition as nodes in the scene graph. The scene graph can then be scripted to deliver any viewing for any content, by mixing and matching viewing commands with content. For example, a stereoscopic display can be scripted as a two-pass rendering method (Figure 3):

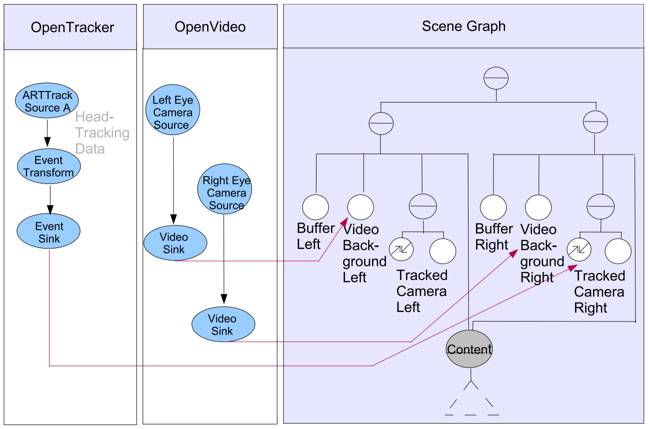


Figure 4: Setup of a tracked video see through

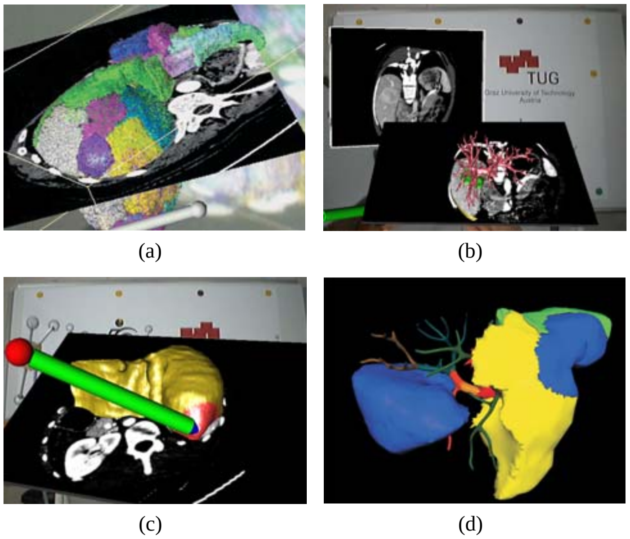
Concerning the What of visualization, special rendering techniques were developed to fit the medical data models. Raw image

[3] http://www.coin3d.org

data can either be visualized using texture-based direct volume rendering (see Figure 5(a)), or using textured cutting planes, which can be arbitrarily placed in the dataset volume (see Figure 5(b)).

The deformable surface model is visualized as a polygonal geometric representation, which can easily be rendered at real-time frame rates on current graphics hardware. Recent techniques such as vertex buffer objects are employed to speed up the rendering, while the model deforms. In addition, deformations are decoupled from the rendering process through multi-threading. Figure 5(c)) shows a user interactively correcting a segmentation error using deformation with a 3D input device.

The polyhedral data model is not rendered directly, but has fast methods for extraction of arbitrary boundary surfaces including internal boundaries (for example between tumor and surrounding tissue). The result of this extraction is then rendered using the same surface rendering techniques as mentioned above. Figure 5(d) shows a visualization of liver segments extracted from a polyhedral model.



(a)

(b)

(c)

(d)

Figure 5: These images show examples of possible rendering techniques using three different data sources. (a) A ray-casting volume renderer is used for visualizing color-coded regions of a liver dataset. In addition, CT data is registered in place for validation purposes. (b) Cutting planes are 2D images which are extracted from the 3D raw dataset and can be positioned in space using tracked input devices. (c) Deformable models are visualized using hardware-accelerated rendering techniques. (d) A volumetric data model allows for storing and rendering heterogeneous information stored consistently into one model.

4.5 Kernel and Scheduling

Our architecture builds on a microkernel pattern [8]. The kernel is responsible for loading and unloading system components at runtime. The kernel is also in charge of the system's overall scheduling.

The basic scheduling consists of two stages. The first stage calls all subscribed components for update. The second stage invokes Coin's scene graph traversal. This strategy ensures that all data is refreshed before they are integrated in the rendering of the current frame (Figure 6).

Such a scheduling runs in a single thread, which makes debugging easy, but does not support concurrent processes which should occur simultaneously well. In particular, multi-threading requires more complex synchronization behaviors. OpenTracker and OpenVideo nodes are frequently implemented with an independent thread of control to service an independent hardware device. The acquired or processed data is buffered and finally collected during rendering.

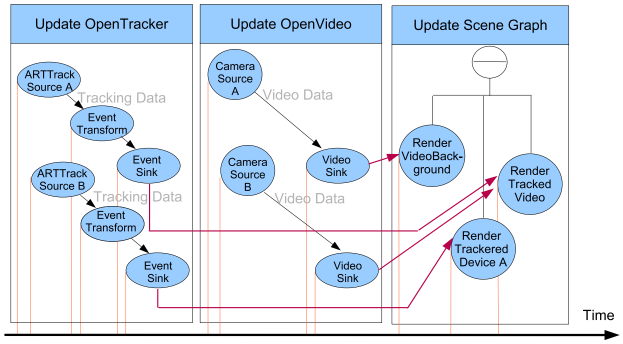


Figure 6: Example of a serial data aquisition

This strategy of asynchronous data acquisition can be used for fast rendering. It is most appropriate when latency should be minimized, for example when rendering from the viewpoint of a tracked camera. However, the asynchronous delivery of data from the individual system components needs to be handled with care, since two separate sources can deliver data at different rates.

Synchronised Triggered, Asynchronous Data Acquisition: To ensure synchronised data, we implemented a scheduling where each component registers for updates by the kernel, while they are still set up in their own threads. This strategy triggers the data acquisition just before the scene graph traversal starts. The acquisition is triggered, but the execution commences in parallel to the rendering. If the data processing takes longer than the scene graph traversal, the system can either be set to just use what is still in the buffer from the last request or it waits until the the processing is finished. Both approaches produce useful synchronisation behavior (Figure 7).

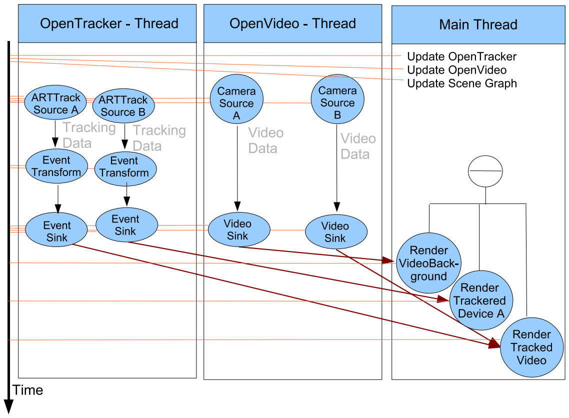


Figure 7: Example of a synchronised parallel data acquisition

Synchronisation Error Estimation:

Since latency and the corresponding temporal error cannot be completely avoided in AR applications, time critical applications such as interventional support applications must at least provide some measure of this error and the resulting uncertainty. By attaching timestamps to the data at the time of acquisition, the variability of data within a single frame can be assessed. This error estimate can for example be used in a visualization of uncertainty or a prediction of the current safeness of the system.

With timestamped data we are also able to detect time lags between for example two tracking events of a single device. Those time frames can again consult us by predicting the current system's safeness. Timestamped data can be used to compute the current age of the integrated data which again, can be used to give some hints about the safeness of the current frame.

5 Applications

Studierstube Medical allows rapid prototyping of powerful medical applications. The following sections mention some prototype applications developed using the Studierstube Medical framework.

5.1 Liver Surgery Planning

In liver surgery, tumor resection is often the only curative treatment for patients suffering from liver cancer. A detailed planning must precede the actual intervention, because information about liver shape, tumor location, and the arrangement of the vascular structure is required.

In current clinical routine, an intervention plan is usually elaborated using the information retrieved from a tomographic imaging modality such as X-ray computed tomography. By inspecting a stack of gray-valued two-dimensional images, the physician must establish a mental three-dimensional model. While radiologists are trained for this type of analysis, surgeons often have problems, because they are naturally oriented towards 3D work.

The virtual liver surgery planning system, or *LiverPlanner* [4] for short, is an on-going research project at Graz University of Technology [6]. *LiverPlanner* combines image analysis and computer graphics in order to simplify the clinical process of planning liver tumor resections.

Image analysis algorithms provide largely automatic segmentation of liver, tumor and vessels [2, 3, 22]. Segmentation refinement permits user-guided verification and correction of any defects in the automatic segmentation [7]. Finally, the treatment planning stage enables the surgeon to elaborate a detailed strategy for the surgical intervention including an analysis of important quantitative indices such as the volume of remaining healthy liver tissue [23]. The outcome of pre-operative planning can then be used directly for the surgical intervention.

The segmentation verification/editing as well as treatment planning are highly interactive tasks and utilize the Studierstube Medical framework for real 3D interaction and stereoscopic visualization. An advantage of using VR was clearly found in a better spatial perception, a simple 3D user interface, and in the duration of the planning process. Evaluations have shown that planning can be carried out much faster (on average less than 25 minutes) than using conventional tools. In addition, more information can be generated due to various surgical planning tools (e.g. generation of liver segments, liver partitioning tools for simulating the surgical intervention, and measurement tools for quantitative analysis). Figure 8 shows some screenshots of the surgical planning tool.

5.2 Surgical Simulation with Haptic Feedback

A preliminary surgical simulator was developed for ARIS*ER. Its main feature is integration of haptic devices with the scene graph based visualization such that robotic telemanipulation, for instance during laparoscopic liver surgery, can be simulated.

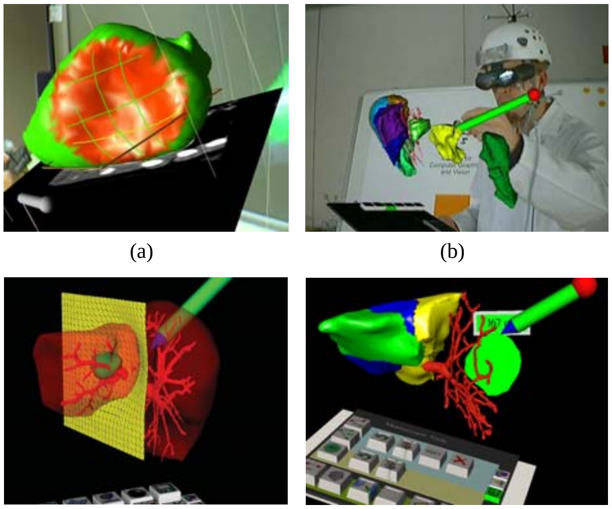

(a) (b)
(c) (d)

Figure 8: The virtual liver surgery planning system integrates various modules for pre-operative planning of liver tumor resection. (a) 3D refinement tools allow online modification of an incorrectly segmented liver dataset using deformable models. (b) Tracked input and output devices allow the simulation of the actual interventions (i.e. removing individual liver segments). (c) Various partitioning tools can be used for simulating non-standard tumor resection methods. (d) A set of measurement tools can be used for quantitative analysis.

Two Phantom Omni haptic devices connected to two separate computers constitute the hardware setup (figure 9). One haptic device is used as a 3D mouse for positioning a virtual laparoscopic camera in a virtual anatomy. The rendered frames are captured and streamed over a network connection to a second computer. The video stream is used as a video background and augmented with a segmented liver, kidneys and other organs. Another phantom device simulates a laparoscopic manipulator which can be moved around in the virtual anatomy. Whenever the manipulator is touching a certain organ, movement of the manipulator is restricted by the haptic device such that penetration of critical organs is avoided. Haptic forces are generated in accordance with the specific tissue properties of an organ. Tissue properties can be set for a certain organ using nodes which encodes tissue stiffness, damping and friction.

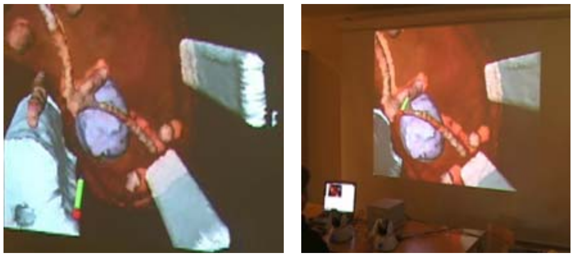
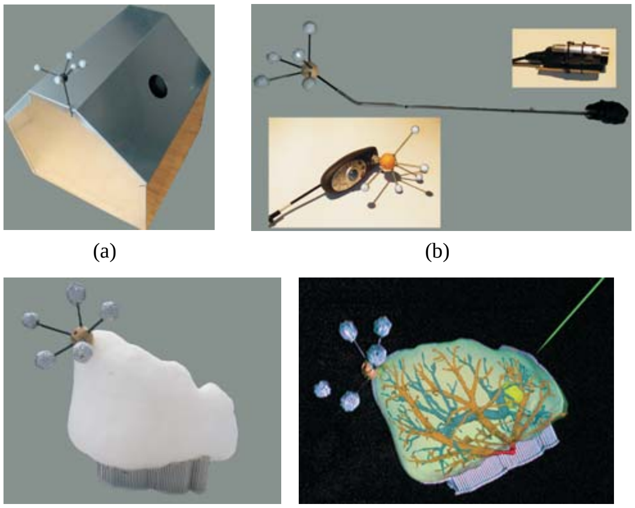

(a) (b)

Figure 9: Screenshots of the haptic application. a) Shows the video augmented with segmented structures and the manipulator which is moved by the haptic device. b) Shows the operator using the two haptic devices and the corresponding visualization.

[4] http://liverplanner.icg.tu-graz.ac.at

5.3 Simulation of Endoscopic Radio Frequency Ablation

Hyperthermia ablation is used for treatment of benign and malignant tumors. The tissue response to heating is related to the absolute temperature induced in the tissue and exposure time. The procedure has been widely used for treating malignant liver tumors not amenable to conventional surgical resection. For safe and efficient placement of ablative probes, we have set up a system that uses Augmented Reality to assist the surgeon while targeting a tumor during laparoscopic liver surgery (Figure 10). Targeting is affected by tumor location and surrounding critical structures that may limit the safe window of probe insertion. Our current implementation shows the targeted tumor and its surrounding vessel trees on an endoscopic video image. It indicates the needle's current position and it supports a virtual extension of the needle's tip to predict its future path.



(a) (b)

(c) (d)

Figure 10: Mock up for simulating an endoscopic radio frequency ablation procedure. a) Tracked box used to simulate a tracked abdomen during the laparoscopic procedure. b) Tracked needle and the tracked camera which acts as the laparoscope. c) The liver phantom is made out of the used CT scan. d) Screenshoot of the augmented video, showing the liver phantom augmented with the targeted tumor, the hepatic vessel tree, the portal vessel tree and the parenchyma. The ablation needle is indicated by a green rendered apex (For tracking the liver phantom we use the target attached to the box and not the one shown on the image)

6 Conclusions

We have developed a new software framework for Virtual and Augmented Reality in image guided medicine. Studierstube Medical has shown its usefulness as a rapid prototyping development tool in applications such as liver surgery planning, radio-frequency ablation and haptic simulation. It constitutes a robust and scalable software framework for medical AR that can be used to develop a wide range of image guided applications. It provides a modular design and a consistent data model that allows manageable testing for quality assurance and an integration a various parts of the medical workflow.

7 Acknowledgments

This research was sponsored by the European Union under contract MRTN-CT-2004-512400 and the Austrian Science Fund *FWF* under contract Y193.

References

[1] Martin Bauer, Bernd Bruegge, Gudrun Klinker, Asa MacWilliams, Thomas Reicher, Stefan Riss, Christian Sandor, and Martin Wagner. Design of a component-based augmented reality framework. In *Proceedings of the International Symposium on Augmented Reality (ISAR)*, October 2001.

[2] R. Beichel. *Virtual Liver Surgery Planning: Segmentation of CT Data*. PhD thesis, Graz University of Technology, 2005.

[3] R. Beichel, T. Pock, Ch. Janko, and et al. Liver segment approximation in CT data for surgical resection planning. In J. Michael Fitzpatrick and Milan Sonka, editors, *In SPIE Medical Imaging '04*, volume 5370, pages 1435–1446, San Diego, 2004. SPIE Press.

[4] A. Bornik, R. Beichel, E. Kruijff, B. Reitinger, and D. Schmalstieg. A hybrid user interface for manipulation of volumetric medical data. In *Proceedings of IEEE Symposium on 3D User Interfaces 2006*. IEEE, March 2006. in print.

[5] A. Bornik, R. Beichel, B. Reitinger, G. Gotschuli, E. Sorantin, F. Leberl, and M. Sonka. Computer aided liver surgery planning: An augmented reality approach. In Robert L. Galloway, editor, *Medical Imaging 2003: Visualization and Display, Proceedings of SPIE*, volume 5029. SPIE, SPIE Press, May 2003.

[6] A. Bornik, R. Beichel, B. Reitinger, G. Gotschuli, E. Sorantin, F. Leberl, and M. Sonka. Computer aided liver surgery planning: An augmented reality approach. In *SPIE Medical Imaging '03*, pages 395–405, February 2003.

[7] A. Bornik, B. Reitinger, R. Beichel, E. Sorantin, and G. Werkgartner. Augmented reality-based segmentation refinement. In *Medical Imaging 2004, Proceedings of SPIE*, pages 77–87, San Diego, 2004.

[8] Frank Buschmann, Regine Meunier, Hans Rohnert, Peter Sommerlad, and Michael Stal. *Pattern-oriented software architecture: a system of patterns*. John Wiley & Sons, Inc., New York, NY, USA, 1996.

[9] Paloc C., Carrasco E., Macia I., Gomez R., Barandiaran I., Jimenez J.M., Rueda O., Ortiz de Urbina J., Valdivieso A., and Sakas G. Computer-aided surgery based on auto-stereoscopic augmented reality. In *Information Visualisation, 2004. IV 2004. Proceedings. Eighth International Conference on*, volume 13, pages 189–193, July 2004.

[10] H. Delingette. General object reconstruction based on simplex meshes. *International Journal of Computer Vision*, 32(2):111–146, September 1999.

[11] H. Edelsbrunner. *Geometry and topology for mesh generation*. Cambridge University Press, New York, NY, USA, 2001.

[12] Wacker FK, Vogt S, Khamene A, Jesberger JA, Nour SG, Elgort DR, Sauer F, Duerk JL, and Lewin JS. An augmented reality system for mr image-guided needle biopsy: initial results in a swine model. *Radiology*, 2(238):497–504, February 2006.

[13] Henry Fuchs, Mark A. Livingston, Ramesh Raskar, D'nardo Colucci, Kurtis Keller, Andrei State, Jessica R. Crawford, Paul Rademacher, Samuel H. Drake, and Anthony A. Meyer. Augmented reality visualization for laparoscopic surgery. In *MICCAI '98: Proceedings of the First International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 934–943, London, UK, 1998. Springer-Verlag.

[14] Kevin Gary, Luis Ibanez, Stephen Aylward, David Gobbi, M. Brian Blake, and Keven Cleary. Igstk: An open source software toolkit for image-guided surgery. *Computer*, 39(4):46–53, 2006.

[15] D. Gering, A. Nabavi, R. Kikinis, N. Hata, L. Odonnell, W. Eric L. Grimson, F. Jolesz, P. Black, and W. Wells III. An integrated visualization system for surgical planning and guidance using image fusion and an open mr. In *Journal of Magnetic Resonance Imaging*, volume 13, pages 967–975, June 2001.

[16] David T. Gering, Arya Nabavi, Ron Kikinis, W. Eric L. Grimson, Nobuhiko Hata, Peter Everett, Ferenc A. Jolesz, and III William

M. Wells. An integrated visualization system for surgical planning and guidance using image fusion and interventional imaging. In *MICCAI '99: Proceedings of the Second International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 809–819, London, UK, 1999. Springer-Verlag.

[17] A. Jalote-Parmar, P.M.T. Pattynama, R.H.M. Goossens, A. Freudenthal, E. Samset, and H. DeRidder. Exploring a user centric methodology to investigate and integrate information gathered during medical intervention. In *In Proceedings of the 16th World Congress on Ergonomics*, 2006.

[18] Kitware, Inc. *The Insight Toolkit Manual Pages*.

[19] Kitware, Inc. *The Visualization Toolkit User's Guide*, January 2003.

[20] Blair MacIntyre, Maribeth Gandy, Jay David Bolter, Steven Dow, and Brendan Hannigan. Dart: The designer's augmented reality toolkit. In *ISMAR*, pages 329–330, 2003.

[21] S. Nicolau, A. Garcia, X. Pennec, L. Soler, and N. Ayache. An augmented reality system to guide radio-frequency tumour ablation. *Computer Animation and Virtual World (previously the Journal of Visualization & Computer Animation)*, 16(1):1–10, 2005.

[22] T. Pock, R. Beichel, and H. Bischof. A novel robust tube detection filter for 3d centerline extraction. In *14th Scandinavian Conference on Image Analysis*, pages 481–490, 2005.

[23] B. Reitinger. *Virtual Liver Surgery Planning: Simulation of Resections using Virtual Reality Techniques*. PhD thesis, Graz University of Technology, 2005.

[24] G. Reitmayr and D. Schmalstieg. An open software architecture for virtual reality interaction. In *VRST '01: Proc. of the ACM symposium on Virtual Reality software and technology*, pages 47–54. ACM Press, 2001.

[25] Michael Rosenthal, Andrei State, Joohi Lee, Gentaro Hirota, Jeremy Ackerman, Kurtis Keller, Etta D. Pisano, Michael Jiroutek, Keith Muller, and Henry Fuchs. Augmented reality guidance for needle biopsies: An initial randomized, controlled trial in phantoms. In *Medical Image Analysis*, volume 6, pages 313–320, September 2002.

[26] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. Szalavàri, L. M. En carna çao, M. Gervautz, and W. Purgathofer. The studierstube augmented reality project. *Presence: Teleoper. Virtual Environ.*, 11(1):33–54, 2002.

[27] Paul S. Strauss and Rikk Carey. An object-oriented 3d graphics toolkit. In *SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, pages 341–349, New York, NY, USA, 1992. ACM Press.