A dual compute resource strategy for computational model-assisted therapeutic interventions

Douglas Hackworth*, Prashanth Dumpuri, Michael I. Miga[a]
Department of Biomedical Engineering, Vanderbilt University, Nashville TN 37235

ABSTRACT

Acquiring and incorporating intraoperative data into image-guided surgical systems has been shown to increase the accuracy of these systems and the accuracy of image-guided surgical procedures. Even with the advent of powerful computers and parallel clusters, the ability to integrate highly resolved computer model information in the planning and execution of image-guided surgery is challenging. More often than not, the computational times required to process preoperative models and incorporate intraoperative data for feedback are too cumbersome and do not meet the real time constraints of surgery, for both planning and intraoperative guidance. To decrease the computational time for the surgeon and minimize the resources in the operating room, we have developed a dual compute node framework for image-guided surgical procedures: (i) a high-capability compute resource which acts as a server to facilitate preoperative planning, and (ii) a low-capability compute resource which acts as a server node/compute node to process the intraoperative data and rapidly integrate the model-based analysis for therapeutic/surgical feedback. In this framework, the preoperative planning utilities and intraoperative guidance system act as client-nodes/graphics-nodes that are assisted by the model-assistant. Processed data is transferred back to the graphics node for planning display or intraoperative feedback depending on which resource is engaged. In order to efficiently manage the data and the computational resources we also developed a novel software manager. This dual-capability resource compute node concept and the software manager are reported in this work, and the low-capability resource compute node is investigated within the context of image-guided liver surgery using data acquired during hepatic tumor resection therapies. Preliminary results indicate that the dual node concept can significantly decrease the computational resources and time required for image-guided surgical procedures.

Keywords: Model-assisted surgical planning, guidance correction, surgical feedback, preoperative planning, intraoperative computing, image-guided liver surgery, surface-based image registration

1. INTRODUCTION

As guided therapeutic interventions begin to acquire more information, the need for continual computational power throughout the planning and intraoperative phase of a procedure has increased. One such example is the numerical optimization of needle placement during radiofrequency ablation treatment (RFA) [1], which requires the iterated solution of a finite-element model of coupled partial differential equations (PDE's) for heat transport and the electro-potential distribution within tissue. Also, Cash et al [2] and Clements et al [3] have reported techniques that combine surface-based registration techniques with finite element computations to correct for the non-rigid deformations observed during hepatic tumor removal procedures. In addition, there is now a substantial body of literature concerned with brain deformation compensation during image-guided surgery [4-6]. These problems involve finite element mesh sizes containing upwards of one hundred thousand elements routinely, and often need to account for geometric and material nonlinearities. On a typical workstation (e.g. with a 3.0 GHz Intel processor) these problems can be challenging and can take considerable time to solve. Even with the advent of powerful computers and parallel clusters, more often than not the computational times required to solve these problems do not meet the real time constraints of surgical procedures. At present, image-guided surgical systems are already using a computer in the operating room (OR) to perform a rigid-body registration between the patient's physical-space and the preoperative image-space and a combination of two dimensional (2D) or three dimensional (3D) graphics to display the registered images and for

* doug.hackworth@vanderbilt.edu ; phone 1 615 343-4216; fax 1 615 343-7919; bmlweb.vuse.vanderbilt.edu
[a] Michael I. Miga is a co-founder of Pathfinder Therapeutics Inc

feedback. Adding the computationally intensive correction routines [4] to this computer will not only increase the computational time required for image-guidance but it will also decrease the computational efficiency of the guidance computer. Therefore it is desirable to offload the task of computing from the computer used in the operating room (OR). Similarly, preoperative planning could benefit from large-scale computations whereby surgical parameters could be explored for a particular patient to plan optimal operative approaches. With this realization, the concept of a dedicated high-capability compute resource for preoperative planning and a low-capability compute resource for performing more complicated computations intraoperatively would seem a promising strategy. With respect to this paper, the computer/system used for guidance in the operating room will be referred to as the *guidance client* (typically a compute platform capable of displaying 2D/3D images, handling tracking instrumentation, and performing very basic rigid body registrations). The *high-capability compute node* will refer to the compute platform capable of complex, intensive computations (typically some sort of compute cluster represented by a mini-blade configuration). The *low-capability compute node* will refer to the compute platform resident within the operating room and interacting with the guidance client. This platform is a multi-processor blade computer (e.g. 8 dual-core processors) that can handle mid-level complexity of computation and will allow for more sophisticated correction routines which will be necessary for integrating model data.

In this scenario, all of the software necessary for detailed and intensive computer model analysis resides on the high-capability compute resource (HCCR) while software concerned with correction resides on the low-capability compute resource (LCCR). It should be noted that these resources are not mutually exclusive. Often the data derived from the HCCR is essential for the correction approaches to take place intraoperatively. The pipeline that runs parallel to the preoperative planning and guidance clients is continuous and is facilitated by a novel submission system framework. The aforementioned correction routines also require that some form of intraoperative data be acquired which provides multiple inputs from the guidance client. Similary, inputs from the *preoperative planning client* are also required for the HCCR. This intraoperative data needed is stored on the guidance client and transferred to the LCCR when a correction is needed. The LCCR performs the computations, and corrected data is then transferred back to the guidance client for feedback. It is therefore important to carefully manage the flow of data, order of execution, and allocation of services and resources so that the computational routines run smoothly without any system errors. It is also important to relay feedback to the user while the computational routines are being performed on the LCCR and attribute appropriate failure messages to the tasks if the computational routines fail (similar requirements hold for the HCCR). We have developed a software system (the Compute Resource Manager, or CRM) for managing the workflow between the guidance and compute resources. CRM is based on a modular framework and allows for the addition of computational routines (to be performed on the HCCR or LCCR) via a plugin architecture. It should be noted that CRM acts simply as a software manager for (a) transferring data from the planning or guidance clients to the HCCR or LCCR and vice-versa, (b) for running the computational routines on the HCCR or LCCR, and (c) relaying feedback messages to clients. The CRM does not perform the routine tasks associated with planning and image-guidance (segmentation, and tool tracking with displays, respectively). While similar resources may have been used for database management and other applications with a single server compute resource, and open-source and commercial software exists for operating and maintaining a single server computer resource, to the best of our knowledge this is the first time such a system has been developed and reported for use in image-guided surgical procedures and involves a parallel multi-performance compute resource structure. The CRM has been described in detail below and a demonstration of its use in performing an intraoperative image registration task using the LCCR for image-guided liver surgery has been presented in the following sections.

2. METHODS

2.1 Compute Resource Manager (CRM)

The underlying architecture of the CRM is a client-server relationship, in which the client communicates with the server to provide computational services to the user. The CRM manages the clients and the servers independently and transparently. It should be noted that although the CRM can be used to manage any peri- or intraoperative computing task, for the remainder of this document CRM has been discussed in the context of image-guided liver surgery (IGLS) and within the context of engagement with the LCCR. Specifically, the surface-based registration technique reported in [3] to establish a relationship between the preoperative image-space and the patient's physical space has been used as an example to illustrate the working details of CRM. In our example, the guidance system is the *client* and the LCCR is the *server*.

On the client/guidance side, the CRM is an application programming interface (API) that integrates with existing software to collect the necessary data from the user. Therefore in this case the client collects the data required to utilize the surface-based registration technique. In addition to acquiring data, the API on the client side allows the user to (a) submit data to the server (LCCR), (b) ask the server to execute computational routines (in this case the surface-based registration technique), (c) query the status of computational routines as they proceed, and (d) retrieve the results once the compute node completes the computational jobs. On the server/LCCR side, the CRM has been designed to (a) manage requests from the clients, (b) invoke the computational routines, (c) allocate computational resources efficiently and, (d) monitor the status of the computational routines. On the LCCR the CRM is run as a perpetual process that is always waiting for new input. Figure 1 shows a schematic of the complete dual resource concept applied to IGLS with respect to the LCCR in this case.

The API on the client side has been implemented in the Python language to take advantage of the many flexible data structures native to Python. All interactions with the compute node are performed through this interface. On the server side, the CRM has been implemented with a modular design that allows for additional computational routines to be incorporated into the workflow using a plug-in architecture. Data is transferred between the client and the server nodes as serialized, (optionally) encrypted objects and reconstituted into memory-resident objects on the receiving end. This ensures that data resident on one node can be reproduced on the other and also minimizes the chances of data corruption during the transfer. We have used the XML-RPC protocol for transporting input and output data, as well as command/control directives, between the client and server nodes.

It should be noted that any type of computer hardware can be used for the compute node. It can be a desktop GPU computing resource or a multi-processor workstation capable of handling the compute-intensive atlas-based registration technique reported in . Also, the medium of communication between the client node and the server can be a traditional wired connection, or a wireless connection such as wireless Ethernet or Bluetooth.

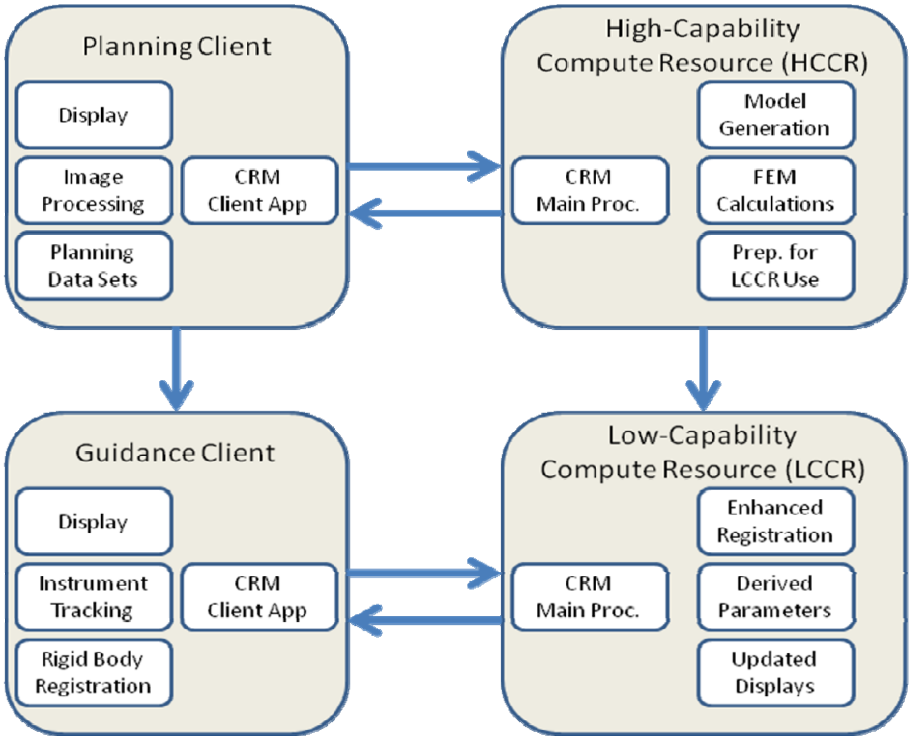

Figure 1. Schematic of the dual compute resource concept developed for image-guided liver surgery (IGLS). IGLS systems usually acquire intraoperative data to perform a registration between the patient's diagnostic preoperative image-space and the operating room space. The preoperatively segmented CT surface is extracted during planning and transmitted from the *planning client* to the HCCR for surface fitting and model generation. The HCCR communicates model data to the LCCR. In the OR, the intraoperative data is transferred to the LCCR and computationally intensive registration routines are executed, following which results are transferred to the *guidance client* for surgical display. We have developed a compute resource manager (CRM) to efficiently manage both the nodes.

2.2 Surface-based registration for IGLS procedures using CRM

A successful surface-based registration between physical-space to image-space is critical in IGLS to provide reliable guidance information to the surgeon. In brief, the intraoperative liver surface acquired in physical-space (or the patient's operating room space) must be registered to the patient's diagnostic preoperative images (image-space). A three-dimensional liver surface is created from the patient's preoperative images using the HCCR and this surface is registered to the intraoperative liver surface facilitated by the LCCR. For the purposes of demonstrating the dual compute resource concept and the CRM software, we used the surface-based registration reported in [3]. This method is a variant of iterative closest-point registration (ICP) [7] in which corresponding salient anatomical features marked on both preoperative and intraoperative surfaces are used to guide the registration by weighting those points in the ICP algorithm. Figure 2 shows a typical dataset used for the surface-based registration technique. Figure 2A shows the three-dimensional liver surface constructed from the patient's preoperative images. Figure 2B shows the intraoperative liver surface acquired using a laser range scanner (LRS). It should be noted that this intraoperative surface can be acquired using intraoperative ultrasound as well, but our group has been researching the use of LRS to acquire intraoperative surface data for brain and liver tumor resection therapies [3, 8-10]. For the purposes of this study, the falciform ligament and inferior ridge of the liver were used as salient anatomical features for the weighted ICP algorithm. These anatomical features were highlighted manually on the patient's preoperative three-dimensional surface as performed by the planning client and an optically tracked pen probe was used by the surgeon to delineate the points on the intraoperative liver surface as performed by the guidance client. These two features have been highlighted on the three-dimensional surface in Figure 2A. The features delineated by the surgeon have been shown on a textured laser range scanner surface in Figure 2C.

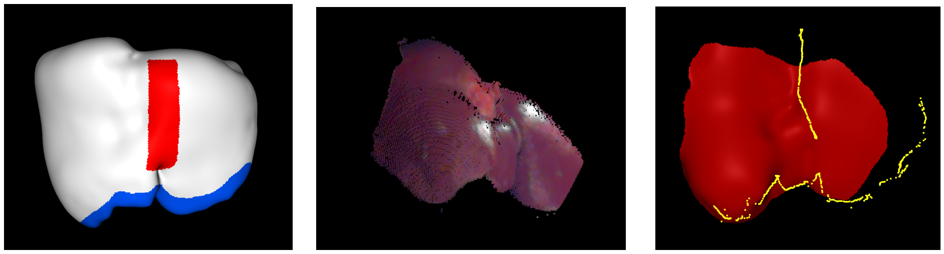

Figure 2. (Color online) Input data for salient point registration. **2A**: Preoperative liver surface segmented from the patient's preoperative CT images. Falciform ligament and inferior ridge highlighted in the figures were delineated manually on the preoperative surface. **2B**: Intraoperative liver surface obtained using a laser range scanner. **2C**: Textured liver surface obtained using the laser range scanner. Falciform ligament and inferior ridge were highlighted on the intraoperative liver surface by the surgeon using an optically tracked pen probe.

2.3 Clinical Cases

Two cases, 1 female, 1 male, were selected from the 75 patient clinical trial being conducted by Pathfinder Therapeutics Inc. at three major medical centers (University of Pittsburgh, University of Florida and Memorial Sloan Kettering Cancer Center, New York). Patient 1 was a 47-year-old female with metastatic cancer of the anus. This patient underwent a right lobectomy at University of Pittsburgh (UPMC). Patient 2 was was a 54-year-old male with metastatic colon cancer; this patient underwent a right lobectomy at University of Florida (UF). The weighted surface-based registration algorithm reported in [3] was used to align the intraoperative liver surfaces to the preoperative images. To illustrate the working details of the dual compute resource concept for IGLS, intraoperative and preoperative data for these 2 patient cases were stored on the guidance client and then transferred to the compute node using the API described earlier. The surface-based registration algorithm was invoked on the LCCR using calls from the CRM and results were transferred back to the guidance client. Data integrity checks were performed automatically throughout this process to ensure that data was not corrupted. Automatic measures also ensured that the resources were distributed efficiently on the compute node. Registration results have been presented in the following section.

3. RESULTS

Figure 3 shows qualitatively the results of the suface-based registration for Patients 1 and 2 reported in this study. Figures 3A and 3B (top row) show the registration results for Patient 1. The registered textured laser range scanner has been overlaid on the three-dimensional preoperative liver surface in Figure 3A. Figure 3B shows the signed closest point distances between the two surfaces. Similar results have been shown for Patient 2 in Figures 3C and 3D.

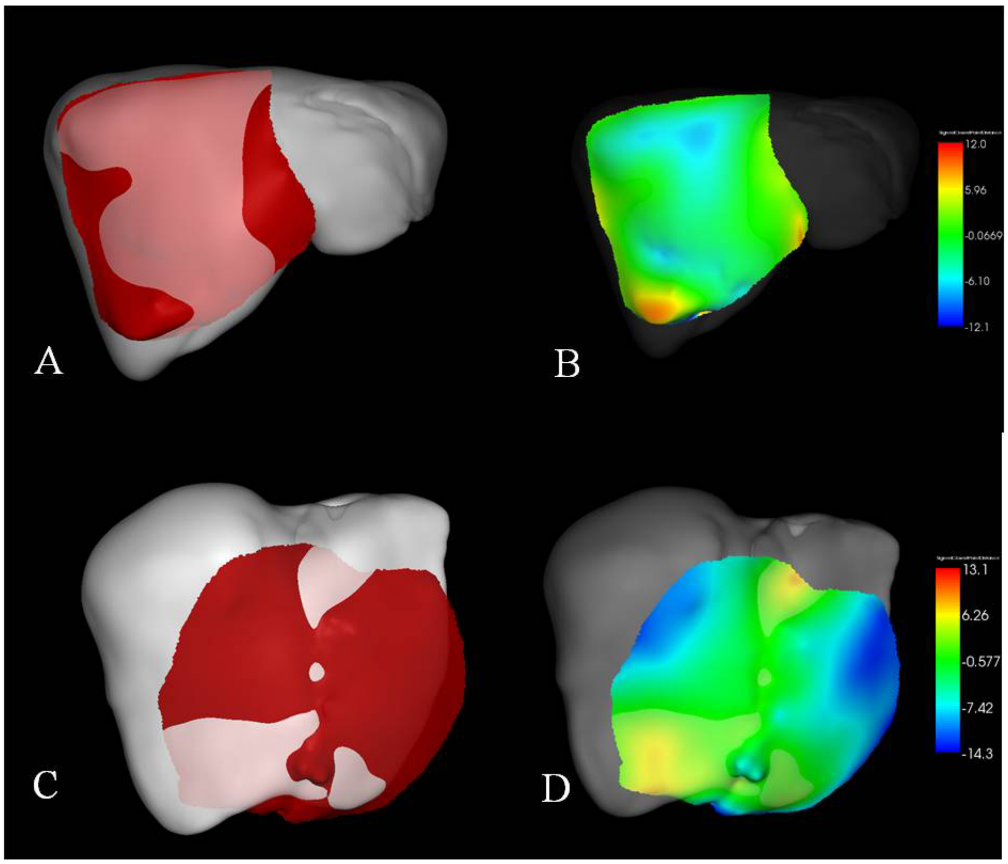

Figure 3. (Color online) Registration results for Patients 1 and 2. The registered texture LRS surface has been overlaid on the three-dimensional liver surface in figures 3A and 3C. Signed closest point distances between the two surfaces have been shown in Figures 3B and 3D. Top row (Figures 3A and 3B) shows the registration results for Patient 1 and the bottom row (Figures 3C and 3D) shows the registration results for Patient 2.

Table 1 summarizes the signed and unsigned closest point distances between the registered liver surfaces. It should be noted that the registration was performed on the LCCR using data obtained from the client.

Table 1. Quantitative results for the surface-based registration algorithm. Mean $\pm$ standard deviation (maximum) of the unsigned and signed closest point distances have been reported for both the patient cases. All results have been reported in millimeters.

Patient #	Unsigned Closest Point Distances (mm)	Signed Closest Point Distances (mm)
1	2.95 $\pm$ 2.08 (12.1)	2.95 $\pm$ 2.08 (12.0)
2	4.91 $\pm$ 3.56 (14.3)	4.91 $\pm$ 3.56 (13.1)

4. DISCUSSION

Systematic studies have shown that intraoperative data increases the accuracy of image-guided procedures. As image-guided procedures acquire more data and integrate more sophisticated data from model-based image anlaysis, there is a need for extensive computation to process the data, and a need to realize systems-design frameworks for integrating this data. In order to address this, we have proposed using two physically distinct compute resources for image-guided procedures that communicate with the standard planning and guidance clients associated with image-guided surgery. This dual compute resource framework involves (i) a high-capability compute resource (HCCR) which interacts with the planning client to facilitate model-based image analysis, and (ii) a low-capability compute resource (LCCR) which interacts with the guidance client to enable more sophisticated updates to guidance displays. In addition, connectivity between HCCR and LCCR, and the respective clients, is important and is provided by a novel software framework called the compute resource manager or CRM. The CRM has been used in this work to transfer the data acquired on the guidance client to the LCCR, to invoke computational routines on the LCCR using calls from the client, and to transfer results from the LCCR back to the guidance client for surgical feedback. The CRM was also used to manage the computational resources efficiently on the compute node itself. In the results, we used the dual compute resource concept and CRM to run a surface-based registration algorithm for IGLS procedures. Registrations were performed on the LCCR and preliminary results showed that offloading the computationally intensive routines to a physically distinct computer/node increases the efficiency of IGLS procedures. In data not reported here, we have executed model building tasks on the HCCR as an initial task in assembling the dual resource methodology.

Though we have introduced the concept of two physically distinct compute resources for IGLS, it is possible to perform the intraoperative computations on the same computer as the *guidance client*. Computations required to process the intraoperative data can be peformed on the computer's GPU while letting the CPU handle the guidance and surgical feedback. Regardless of the number of resources used for image-guided procedures, data transfer between the resources and client has to be handled with care so as to not compromise the patient's safety. We have designed CRM to handle sensitive information using data encryption and error checking procedures.

A simple surface-based registration algorithm was used to demonstrate the workflow using the CRM. It should be noted that the dual resource concept and CRM can be readily and easily extended to handle more intensive computational routines such as those required for model-updated image-guided procedures. We are currently working on applying this technology to generating finite element meshes and for the atlas-based routines reported in [4]. Also, the CRM needs to be tested in more clinical cases for robustness and repeatablity before extending its use to the operating room.

5. ACKNOWLEDGEMENTS

This work was supported by NIH grant R21EB007694-01 awarded by the National Institute of Biomedical Imaging and Bioengineering. The authors would like to thank the operating room staff at UPMC and UF for their assistance in data collection. The authors would also like to thank Logan W. Clements and the staff at Pathfinder Therapeutics Inc., for providing us with the data and their assistance in processing the data. All visualizations were performed using the Visualization Toolkit (www.vtk.org).

REFERENCES

1. Chen, C.C.R., M.I. Miga, and R.L. Galloway, *Characterization of tracked radlofrequency ablation in phantom.* Medical Physics, 2007. **34**(10): p. 4030-4040.
2. Cash, D.M., et al., *Concepts and preliminary data toward the realization of image-guided liver surgery.* Journal of Gastrointestinal Surgery, 2007. **11**(7): p. 844-859.

3. Clements, L.W., et al., *Robust surface registration using salient anatomical features for image-guided liver surgery: Algorithm and validation.* Medical Physics, 2008. **35**(6): p. 2528-2540.
4. Dumpuri, P., et al., *An atlas-based method to compensate for brain shift: Preliminary results.* Medical Image Analysis, 2007. **11**(2): p. 128-145.
5. Clatz, O., et al., *Robust nonrigid registration to capture brain shift from intraoperative MRI.* Ieee Transactions on Medical Imaging, 2005. **24**(11): p. 1417-1427.
6. Sun, H., et al., *Stereopsis-guided brain shift compensation.* Ieee Transactions on Medical Imaging, 2005. **24**(8): p. 1039-1052.
7. Besl, P.J. and N.D. McKay, *A Method for Registration of 3-D Shapes.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1992. **14**(2): p. 239-256.
8. Cash, D.M., et al., *Incorporation of a laser range scanner into image-guided liver surgery: Surface acquisition, registration, and tracking.* Medical Physics, 2003. **30**(7): p. 1671-1682.
9. Cao, A., et al., *Laser range scanning for image-guided neurosurgery: Investigation of image-to-physical space registrations.* Medical Physics, 2008. **35**(4): p. 1593-1605.
10. Miga, M.I., et al., *Cortical surface registration for image-guided neurosurgery using laser-range scanning.* Ieee Transactions on Medical Imaging, 2003. **22**(8): p. 973-985.