The International Journal of **Medical Robotics** and **Computer Assisted Surgery**

THE INTERNATIONAL JOURNAL OF MEDICAL ROBOTICS AND COMPUTER ASSISTED SURGERY
Int J Med Robotics Comput Assist Surg 2012; **8**: 253–260.
DOI: 10.1002/rcs.1412

ORIGINAL ARTICLE

Modular multiple sensors information management for computer-integrated surgery

Alberto Vaccarella[1]*
Andinet Enquobahrie[2]
Giancarlo Ferrigno[1]
Elena De Momi[1,3]

[1]*NearLab, Dipartimento di Bioingegneria, Politecnico di Milano, Milano, Italy*

[2]*Kitware Inc., Clifton Park, NY, USA*

[3]*ITIA, National Research Council, Milano, Italy*

*Correspondence to: A. Vaccarella, Politecnico di Milano, Bioengineering Department, NearLab, Piazza Leonardo Da Vinci 32, 20133 Milano, Italy.
E-mail: alberto.vaccarella@mail.polimi.it

Accepted: 7 December 2011

Abstract

Background In the past 20 years, technological advancements have modified the concept of modern operating rooms (ORs) with the introduction of computer-integrated surgery (CIS) systems, which promise to enhance the outcomes, safety and standardization of surgical procedures. With CIS, different types of sensor (mainly position-sensing devices, force sensors and intra-operative imaging devices) are widely used. Recently, the need for a combined use of different sensors raised issues related to synchronization and spatial consistency of data from different sources of information.

Methods In this study, we propose a centralized, multi-sensor management software architecture for a distributed CIS system, which addresses sensor information consistency in both space and time. The software was developed as a data server module in a client–server architecture, using two open-source software libraries: Image-Guided Surgery Toolkit (IGSTK) and OpenCV. The ROBOCAST project (FP7 ICT 215190), which aims at integrating robotic and navigation devices and technologies in order to improve the outcome of the surgical intervention, was used as the benchmark. An experimental protocol was designed in order to prove the feasibility of a centralized module for data acquisition and to test the application latency when dealing with optical and electromagnetic tracking systems and ultrasound (US) imaging devices.

Results Our results show that a centralized approach is suitable for minimizing synchronization errors; latency in the client–server communication was estimated to be 2 ms (median value) for tracking systems and 40 ms (median value) for US images.

Conclusion The proposed centralized approach proved to be adequate for neurosurgery requirements. Latency introduced by the proposed architecture does not affect tracking system performance in terms of frame rate and limits US images frame rate at 25 fps, which is acceptable for providing visual feedback to the surgeon in the OR. Copyright © 2012 John Wiley & Sons, Ltd.

Keywords sensors management; IGSTK; surgical navigation; robotic neurosurgery

Introduction

In the past 20 years, technological advancements have modified the concept of modern operating rooms (ORs) with the introduction of computer-integrated surgery (CIS) systems, which promise to enhance the outcome, safety and standardization of surgical procedures (1,2).

CIS systems usually rely on technology that has been traditionally classified as passive or active (3); the first refers to surgical navigation systems, the latter includes robotic systems. Surgical navigation involves: (a) a registration process between physical space and pre-operative images space; and (b) intraoperative localization of the surgical tool in order to show the position and orientation of the instrument with respect to anatomical structures. Nowadays, navigation systems are used in orthopaedic surgery (4), abdominal surgery (5) and neurosurgery (6). Robotic technology can extend surgeons' capabilities by providing physiological tremor reduction and improving repeatability and accuracy (7) in different procedures, including orthopaedic surgery (8), abdominal surgery (9), neurosurgery (10) and cardiac surgery (11).

In CIS, different types of sensors, which can be classified as follows according to the information provided to the surgeon, are widely used:

1. Position information-sensing devices.
2. Force-sensing devices.
3. Intra-operative imaging devices.

Depending on their working principle, each of the previous classes can be further subdivided. Position information-sensing devices can be mechanical, ultrasound (US)-based, inertial, optical or electromagnetic. Mechanical localization systems have been used in ENT surgery (12). Ultrasound-based localization systems were proposed for the registration process in robotic neurosurgery (13). Inertial measurement units (IMUs) allow for the determination of position and orientation of an object at a high sample rate, and are mainly used for research purposes (14).

Force sensors are involved in several studies for biomechanical characterizations of tool–tissue interactions (15). The main application of force sensors is to provide haptic feedback to tele-operated surgical robots (16) and they rely on different technologies, such as strain gauges (17,18), optical fibre (19,20), capacitive-based, piezoelectric-based and vibration-based force-sensing (21).

Intra-operative imaging devices include magnetic resonance imaging (MRI), computed tomography (CT), fluoroscopy and ultrasonography (US), which are used to update the preoperative images and provide more information to the surgeons; this is of the utmost importance when dealing with soft tissues (e.g. brain) that undergo modification with respect to preoperative images during the intervention (22,23). Additionally, recent developments in high-definition, digital endoscopy offers an opportunity for vision-based navigation in image-guided interventions, such as laparoscopy, trans-nasal skull-base neurosurgery and natural orifice transluminal endoscopic surgery (NOTES) (24).

The need for libraries for the simultaneous management and integration of sensors information has recently arisen. In the surgical field, this has led to the design of a software framework for tracking system management and surgical navigation application prototyping, known as the Image-guided Surgery Toolkit (IGSTK) (25–27), an open-source project continuously expanding to also include other types of sensor (e.g. video and US images).

Furthermore, modern CIS systems often rely on multiple sensors from the aforementioned categories in order to ensure redundancy in case of failure and to provide different information to increase surgeons' perception. For instance, data fusion using an optical localization system and an IMU has been proposed (28) to overcome line-of-sight occlusion and achieve a higher update rate, in order to perform patient motion compensation with a robotic system. The combined use of a US imaging device and a localization system (typically an optical one) to achieve a three-dimensional (3D) volume reconstruction (freehand 3D ultrasound) (29), has been proposed for brain shift estimation in neurosurgery.

The management of multiple sensors information with different data generation rates and space scale entails a data fusion process in both time and space, in order to extract consistent information. Timing aspects are crucial in a multiple-sensor scenario, since each datum needs to be correctly synchronized to the other. Using a software-based application for time-stamping external information induces a temporal imprecision, which can be due to the offset of the PC clock with respect to a reference time and to the drift of the internal clock (frequency error) (30). Spatial consistency is also required when dealing with different sensors information; spatial transformations between reference systems are obtained with calibration procedures (31,32).

Most CIS systems were traditionally controlled by a single computer and program. Recently, a modular, distributed approach has been proposed (33) and in the EU-funded project ROBOCAST (FP7 ICT-2007-215190) (34); such architectures make it easy to add features, replace devices or re-use components of a CIS system. Synchronization and spatial consistency issues that occur when different clocks and several reference systems are involved (35) were addressed. Communication between components in distributed systems is in charge of a middleware layer, which enables transport of network-independent communication without time-consuming adaptations and redevelopment of domain applications (36). Following the classification in (37), surgical distributed architectures usually rely on object-oriented middleware, such as the Common Object Request Broker Architecture (CORBA) (33,38,39). As an alternative, transaction-oriented and message-oriented middleware may be used, although they are mainly used for distributed databases. The publish/subscribe pattern (also known as pub/sub) is part of a message-oriented middleware. In a pub/sub architecture, senders (publishers) and receivers (subscribers) are loosely coupled and have almost no knowledge about each other: for certain applications this can be an advantage, allowing a dynamic network topology. However, this raises safety issues, since possible crash of a publisher or a subscriber application will not be notified to the other (40).

In this study, we propose an IGSTK-based, multi-sensor management system for a distributed CIS architecture, based on object-oriented middleware that addresses information consistency in both space and time. The aim of this work was to integrate the IGSTK library in a modular architecture for CIS, to verify the satisfaction of clinical requirements and to test the applicability of such a framework in CIS.

The choice of an open-source framework such as IGSTK ensures a very well-maintained source code, the support of a worldwide community of users and developers and the possibility of customization and extension of the toolkit features. Compared to utilizing commercial products, the use of open-source platforms allows costs to be reduced.

The system was validated within a CIS system for robotic neurosurgery, in order to verify the suitability of the proposed central sensor management system for neurosurgery applications and to measure the latency in a client–server architecture.

Materials and methods

CIS system

The proposed sensor management architecture was developed and tested within the distributed CIS system of the ROBOCAST project. ROBOCAST combines navigated and robotic approaches (Figure 1) to address minimally invasive neurosurgery procedures, e.g. biopsy, through a small aperture in the skull (keyhole neurosurgery). In ROBOCAST, three robots connected in a kinematic chain (13 degrees of freedom) are used to optimize the positioning accuracy (35).

System accuracy and safety are extended by means of different sensors: an active marker optical tracking system, Optotrak Certus (NDI Inc., Ontario, Canada), which surveys the overall robotic chain; an electromagnetic tracking system, Aurora (NDI); and an ultrasound imaging device, Prosound Alpha 7 (Aloka Co. Ltd, Japan).

The main components of the ROBOCAST distributed architecture, shown in Figure 2, are:

- *Naming server* (NS), similar to a domain name system (DNS) of the ROBOCAST network, which translates domain names into the respective numerical identifiers.
- *Safety check* (SC), a process that periodically checks for consistency of the calibration and registration transformations and for possible occlusion of the line of sight of the optical tracking system.
- *High-level controller* (HLC), in charge of robots control.
- *Haptic controller* (HC), which controls a linear actuator motion according to a haptic device input and manages force information.
- *Human–computer interface* (HCI), which provides a surgeon interface for intraoperative navigation and workflow execution.
- *Sensor manager* (SM), an IGSTK-based application that gathers tracking data and US images and provides them to the other components upon request.

The middleware of ROBOCAST is the ACE ORB (TAO) (41), a freely available, open-source and standards-compliant, real-time C++ implementation of CORBA, based upon the Adaptive Communication Environment (ACE) (42).

Sensor manager

The sensor manager (SM) is a software component within the distributed architecture of the ROBOCAST system (43). It is implemented as a C++ service application without a user interface running on a dedicated machine. The main purpose of the SM is to gather data from localization systems (optical and electromagnetic) and

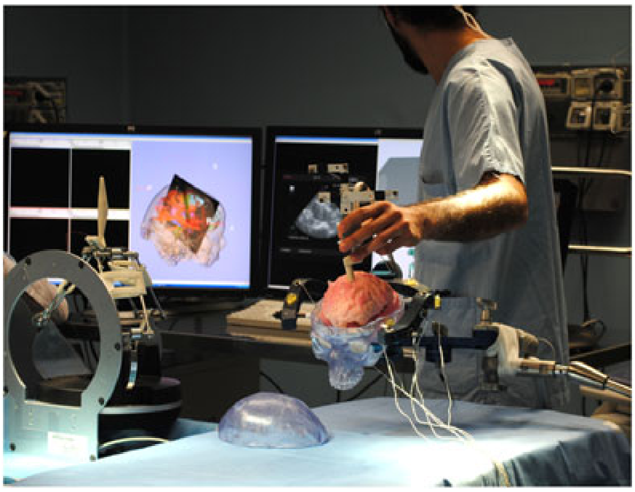

Figure 1. The ROBOCAST demonstrator used in *in vitro* trials at Ospedale Maggiore (Verona, Italy). The operator is manoeuvring an ultrasound (US) probe while looking at the intra-operative visualization monitors. The US image is superimposed on the pre-operative brain models

DOI: 10.1002/rcs

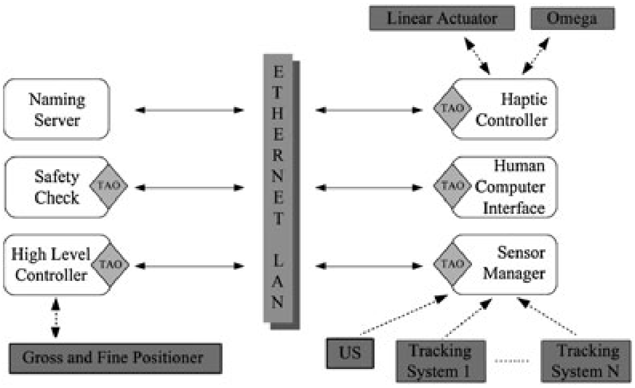

Figure 2. The ROBOCAST distributed architecture. Hardware devices (grey boxes) are connected to their respective controllers. A middleware layer (TAO, the ACE ORB), over a gigabit ethernet LAN, manages the communication of data and services between each software module of the system. The sensor manager (SM) is in charge of data acquisition from the US imaging device and localization systems

US imaging devices, and forward such information upon request to the other components of the system (clients). As is each component of the ROBOCAST framework, the SM is part of a client–server architecture based on the middleware TAO CORBA.

The SM encompasses two main parts; (a) interfacing the hardware based on the open source libraries IGSTK and OpenCV (44); and (b) acting as server to provide data and services to other modules in a CORBA-based network architecture. Also, an elaboration layer is provided to manage reference frame transformations. The registration/calibration, e.g. between the optical and electromagnetic reference systems, was performed using algorithms developed in (45), whereas patient registration to preoperative image space was accomplished using the Horn method (46). IGSTK is mainly used for spatial relationship hierarchy definition among reference frames and for tracking data acquisition. IGSTK provides software components to implement a spatial object hierarchy, which facilitates spatial transformation computation between couples of reference frames. This allows clients to ask for the transformation matrix of a reference system associated with an object in the ROBOCAST scenario, with respect to other reference systems.

The SM can transparently manage all the IGSTK-supported tracking devices through an XML configuration file, allowing flexibility in hardware arrangement.

According to the IGSTK architecture, as shown in Figure 3, tracking data are continuously acquired in a separate thread (a tracker thread, one for each tracking system) and stored in a buffer. The main thread updates spatial objects transformation by reading the stored tracking data from the buffer at a user-defined frequency (20 Hz), and marks each datum with a time-stamp provided by the IGSTK real-time clock. Pose data is checked for temporal validity prior to being provided to clients; the IGSTK 'IsValidNow()' function compares the current time (read from the real-time clock) and the time-stamp associated with the pose data, and forwards the information to the client only if less than K ms has elapsed since data acquisition (where K is a user-defined threshold).

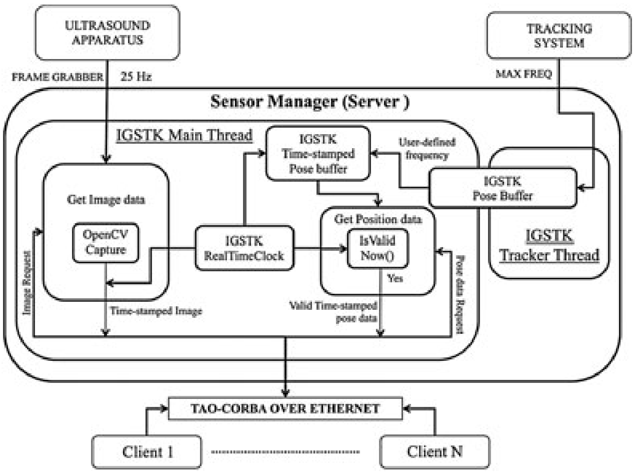

Figure 3. The SM data acquisition architecture. A separate tracker thread is spawned for each tracking system in order to continuously store new data in a buffer. The buffer is shared with the application main thread, where data are time-stamped by the IGSTK real-time clock and copied in another buffer. Through the IsValidNow() block, pose data are delivered to the client only if $<K$ ms have elapsed since data were acquired (where K = user-defined threshold). US image requests are managed in the main tracking thread with the OpenCV capture module and time-stamped by the IGSTK RealTime Clock

Open-CV allows the acquisition of images from US imaging devices, which are connected to the SM laptop via an USB frame grabber (EZ-Grabber, PAL: 720 × 576 @ 25 fps). US images are time-stamped using the same clock used for tracking data (IGSTK real-time clock).

Experimental protocol

The IGSTK-based application was validated in a modular and distributed CIS system. Experiment 1 was aimed at comparing time intervals measured with different clocks in the ROBOCAST distributed architecture. Experiment 2 computed the SM latency.

Experiment 1
The optical localization system (Certus, NDI Inc., Ontario, Canada) was connected to the SM (SERVER), and two client machines (CLIENT1 and CLIENT2) were set-up to request localization data. The three computers' specifications are listed in Table 1. Each machine was connected to a gigabit ethernet LAN in the ROBOCAST, CORBA-based client–server architecture.

The reference time for the experiment was provided by the SM, based on the time-stamp of localization data. CLIENT1 and CLIENT2 internal clocks (based on MS Windows API) were compared with the reference time in order to explore inter-machine variability. Since Windows NT operating systems' internal timer resolution is 10–15 ms (47), a higher resolution is achieved by accessing the high-precision event timer (HPET) incorporated in the PC through the Windows API; for each machine in our set-up, the frequency of the HPET proved to be nominally 3.57 MHz with a microsecond resolution.

SM and CLIENT clocks were started and stopped together in order to measure six different time intervals ΔT_i (15, 30, 45, 60, 75 and 90 min). The test protocol procedure is shown in Figure 4.

When the client requests tracking data, the SM, based on the IGSTK framework, provides data and time-stamp information; the client receives the data and the time-stamp ($T1_{server}$) and also stores its current time according to the internal HPET ($T1_{client}$). After the time interval ΔT_i, the same pattern is repeated and $T2_{server}$ and $T2_{client}$ are stored in order to compute ΔT_{client} ($T2_{client} - T1_{client}$) and ΔT_{server} ($T2_{server} - T1_{server}$). The difference ($\Delta T_{client} - \Delta T_{server}$) between each couple of time measurements is stored.

The experiment was repeated 10 times for each ΔT_i and for both CLIENT1 and CLIENT2.

Experiment 2
In the following experiment, the latency of the SM responses was evaluated in cases of tracking data and US image requests by clients. The experimental set-up encompassed:

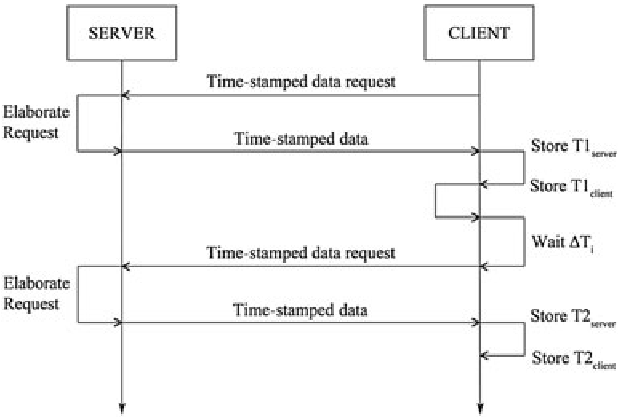

Figure 4. Experimental protocol to compare ΔT_i measured with different PCs: the CLIENT receives time-stamped data and compares SERVER time-stamp with its internal HPET time

- Two PCs, SM and CLIENT1 (see Table 1 for details) connected to the CORBA-based ROBOCAST network.
- An active markers optical tracking system, NDI Optotrak Certus.
- A passive markers optical tracking system, NDI Polaris Vicra (not used in ROBOCAST).
- An electromagnetic tracking system, NDI Aurora.
- An ultrasound imaging device, Aloka Prosound Alpha 7.

The protocol was as follows: the CLIENT1 clock (based on its internal HPET) measured the time ($\Delta T_{latency}$) elapsed from the instant a data request was issued to the SERVER to the instant data were received by the client. The experiment was repeated 6000 times for each tracking system and for the US imaging device.

Results

Experiment 1

Figure 5 shows the differences $\Delta T_{client} - \Delta T_{server}$ for CLIENT1 and CLIENT2. Mean and standard error of 10 repetitions is reported for each ΔT_i. The absolute value of the error linearly increases with the time interval ΔT_i. Linear regression was also calculated with reference to the IGSTK real-time clock running on the SM (SERVER); the CLIENT1 internal clock resulted in being slower (about 1 ms/min), while the internal clock on CLIENT2 proved to be faster (about 2 ms/min).

Table 1. Specifications of computers involved in the tests

	CPU	RAM	OS
SM	Intel T7600 @2.33 Ghz	2 GB	MS Windows XP Professional, SP3
CLIENT1	Intel Q9550 @2.83 GHz	3 GB	MS Windows XP Professional, SP3
CLIENT2	Intel E6600 @2.4 GHz	2 GB	MS Windows XP Professional, SP3

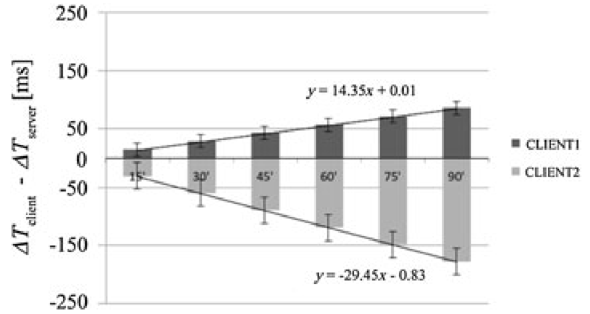

Figure 5. Time measurement comparison using the SM and two different clients: each column in the graph represents the mean with the associated standard error of 10 repetitions. Linear regression equations are reported

Experiment 2

The latency of the SM in providing data to clients is shown in Figure 6. Since the data distribution resulted in being non-Gaussian (Lilliefors test, $p < 10^{-3}$), median values (with interquartile ranges) were computed. Latency for tracking data resulted in being about 2 ms and did not depend on the tracking system used (Kruskal–Wallis test, $p > 0.5$). US images showed a latency of about 40 ms.

Discussion

All innovative systems for surgery assistance require sensor information integration in order to effectively carry out the procedure with an increased accuracy and safety, both for patients and operators.

In this study, we presented a centralized software application for the management of multiple sensor within an integrated robotic and navigated platform for neurosurgery based on IGSTK.

Within the ROBOCAST project, the SM was designed to acquire, time-stamp, process and broadcast data from localization systems (optical and electromagnetic) and from a US imaging device to other software modules distributed over a local area network (LAN). Moreover, a modular and general-purpose approach was adopted, rather than one tailored to the hardware, allowing interchangeability of sensors and seamless integration of more than two sensors. The main challenge of such an application is to provide consistent data from different sensors, in terms of both spatial relationship and synchronization, keeping the latency within a desired limit, depending on the clinical application, and without affecting the actual hardware frame rate.

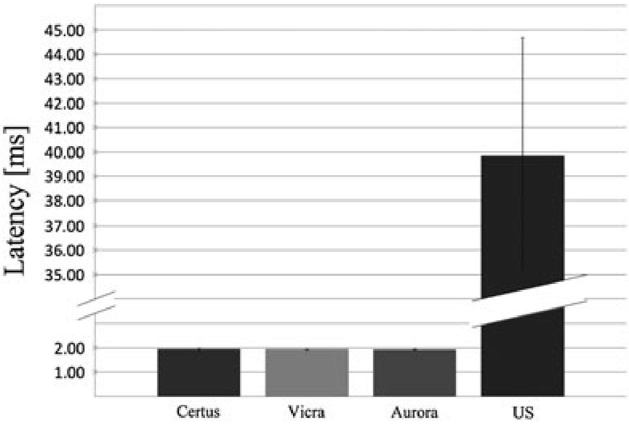

Figure 6. SM response latency for tracking data and US images requests

The IGSTK open-source library proved to be suitable for meeting the previous specifications. In particular, IGSTK encourages the development of hardware-independent applications and provides support for a wide range of localization systems (48); additionally, being an open-source project makes it possible to extend features, as was done for the NDI Optotrak Certus, which was not initially supported (49). IGSTK also provides software tools to manage spatial relationship hierarchies, which facilitate the development of computer integrated surgery applications. This is relevant in a multi-sensor robotic system for surgical applications, such as ROBOCAST, where multiple reference frames are involved (preoperative images, optical, electromagnetic, US images, actuators' reference frames). Once the calibrations between different spaces are provided, the SM automatically handles relative spatial transformations. The IGSTK real-time clock module is used as a common reference to time-stamp data acquired from all sensors (both tracking systems and US imaging device).

Synchronization in distributed architectures usually entails time-stamp exchanges and transformations between different computers. In (50), this is achieved by exploiting a synchronous bus network (the FireWire, IEEE 1394); the bus clock is used to estimate the drift of all computer clocks in order to exchange data time-stamps with high precision and to compute time equivalences between the different clocks for post-processing time-stamping synchronization.

When dealing with ethernet communication, as in the case of the ROBOCAST architecture, the communication bus does not carry synchronization information. In experiment 1 we proved that, in such conditions, the presence of different clocks is not acceptable, as it leads to time drift due to clock offset and an unpredictable combination of quartz inaccuracies that depend on the operating temperature and manufacturing process; the frequency tolerance for the single quartz is typically in the range ±10–100 ppm. The present study proposes the use of a unique time-stamping module for all sensors to avoid inaccuracy due to the offset between different clocks. It would also ensure a frequency drift within the tolerance of a single quartz, meaning an error of 9-90 ms every 15 min; such an error can be considered tolerable, depending on the specific application. For example, in the ROBOCAST scenario, where an optically-tracked linear probe is advanced in the brain at a maximum velocity of 2 mm/s, the maximum localization error of the surgical tool due to clock drift is 180 μm in the case of a 15 min acquisition; this is negligible compared to neurosurgery requirements and robot targeting accuracy (35).

The proposed SM, as part of a modular system, is also ready to be used in other possible architectures, thanks to the middleware abstraction layer. For example, a SM module can be instantiated for each sensor node (in order not to limit acquisition bandwidth) and a central SM would take care of processing data synchronized via precision time protocol (PTP) (51). We actually tested the worst-case scenario, where all the sensors were connected to the same acquisition workstation, which was also in charge of sensor data processing and integration. Furthermore, since we proved the feasibility of the proposed approach within the ROBOCAST scenario, which is a paradigmatic example of an advanced CIS system, it is expected that other applications, e.g. endoscopy or MRI-guided surgery, would also be satisfied by our architecture.

The client–server architecture based on TAO–CORBA was preferred to a pub/sub approach mainly for safety reasons. The pub/sub approach is reported to be unsafe, since application crash cannot be detected due to the decoupling of publisher and subscriber, which is not acceptable for life-critical applications such as computer-assisted surgery. This issue could be handled at application level (e.g. each application emits a periodic heartbeat) but this would increase complexity and bandwidth consumption. Whether client–server or pub/sub makes the most efficient use of bandwidth is a nuanced issue. On the one hand, publishers continuously stream data over the network, regardless of whether or not clients ask for that information. This can lead to unwanted bandwidth consumption, whereas in a client–server architecture the bandwidth is occupied only upon client request. On the other hand, a client–server approach requires twice as much network traffic to get a result (request and reply). However, in CIS applications, the need and the update rate of sensor data can vary depending on the step of the surgical workflow (e.g. registration/calibration, navigation, etc.).

Experiment 2 was designed in order to assess the overall latency for a client to receive requested data. The processing introduced by the SM must be transparent to the client, not limiting the actual hardware frequency.

The results of Experiment 2 showed a response latency of about 2 ms for each tested tracking system. With this latency, the client can issue up to 500 requests/second – much faster than the data generation rate for the majority of commercial tracking systems (NDI Optotrak Certus is the only tracking system that can reach frame rates > 500 Hz if less than seven markers are connected). It is relevant that this result does not depend on the tracking system; any possible, subsequent elaboration of tracking data can be designed regardless of the hardware in use. Other types of sensor could be integrated as well, e.g. force sensors, IMUs, etc.

Latency for US images proved to be around 40 ms, which is limited by the frame grabber acquisition rate (25 fps); whenever a higher frame rate is required for high-speed movement tracking or tissue elastography (52), a tailored high frame rate acquisition hardware must be adopted (53). The same latency is expected for any other kind of images with the same pixel number.

In ROBOCAST, a synchronous communication model (33) was adopted, which means that the application execution on the client is blocked until a response from the server is received. Thus, the latency observed in experiment 2 includes request transmission time, request elaboration on the SM and response transmission time.

The object-oriented middleware, TAO, provided an abstraction layer, which enables developers to skip low-level implementation of the communication protocol. It proved to be agile and did not introduce significant delay in the communication. Furthermore, with such a distributed architecture, many clients can simultaneously access the sensors, which is an important starting point towards the development of increasingly sophisticated CIS applications.

Conclusion

The feasibility of an IGSTK-based application for multiple sensors management in CIS was discussed. IGSTK proved to be a flexible, hardware-independent framework that helps in managing spatial relationships between different sources of information.

The proposed centralized approach limits the effect of clock inaccuracies on the overall system, proving to be adequate for neurosurgery requirements.

Latency introduced by the proposed architecture does not affect the tracking systems' performance in terms of frame rate; it limits US images to a frame rate of 25 fps, which is acceptable to provide visual feedback to the surgeon in the OR.

Acknowledgements

This project was supported by the EU (Grant No. ROBOCAST FP7-ICT-215190) and Scuola Interpolitecnica di Dottorato, Politecnico di Milano, Bari e Torino.

References

1. Joskowicz L, Taylor RH. Computers in imaging and guided surgery. *Comput Sci Eng* 2001; **3**(5): 65–72.
2. Taylor RH, Kazanzides P. Medical robotics and computer-integrated interventional medicine. *Adv Comput* 2008; **73**: 219–260.
3. DiGioia AM, Jaramaz B, Picard F. *Computer and Robotic-assisted Hip and Knee Surgery*. Oxford, University Press: Oxford, UK, 2004.
4. Hananouchi T, Saito M, Koyama T, *et al*. Tailor-made surgical guide based on rapid prototyping technique for cup insertion in

total hip arthroplasty. *Int J Med Robotics Comput Assist Surg* 2009; **5**(2): 164–169.

5. Zhang H, Banovac F, Lin R, *et al*. Electromagnetic tracking for abdominal interventions in computer aided surgery. *Comput Aid Surg* 2006; **11**(3): 127–136.
6. Willems PWA, van der Sprenkel JWB, Tulleken CAF, *et al*. Neuronavigation and surgery of intracerebral tumours. *J Neurol* 2006; **253**(9): 1123–1136.
7. Coste-Manière È, Olender D, Kilby W, *et al*. Robotic whole body stereotactic radiosurgery: Clinical advantages of the CyberKnife® integrated system. *Int J Med Robotics Comput Assist Surg* 2005; **1**(2): 28–39.
8. Chung JH, Ko SY, Kwon DS, *et al*. Robot-assisted femoral stem implantation using an intramedulla gauge. *IEEE Trans Robotics Autom* 2003; **19**(5): 885–892.
9. Hanly EJ, Talamini MA. Robotic abdominal surgery. *Am J Surg* 2004; **188**: 19–26.
10. McBeth PB, Louw DF, Rizun PR, *et al*. Robotics in neurosurgery. *Am J Surg* 2004; **188**(4): 68–75.
11. Woo YJ. Robotic cardiac surgery. *Int J Med Robotics Comput Assist Surg* 2006; **2**(3): 225–232.
12. Freysinger W, Gunkel AR, Bale R, *et al*. Three-dimensional navigation in otorhinolaryngological surgery with the viewing wand. *Ann Otol Rhinol Laryngol* 1998; **107**(11, pt 1): 953–958.
13. Varma TRK, Eldridge P. Use of the NeuroMate stereotactic robot in a frameless mode for functional neurosurgery. *Int J Med Rob Comput Assist Surg* 2006; **2**(2): 107–113.
14. Ren H, Rank D, Merdes M, *et al*. Development of a wireless hybrid navigation system for laparoscopic surgery. *Stud Health Technol Inform* 2011; **163**: 479–485.
15. Abolhassani N, Patel R, Moallem M. Needle insertion into soft tissue: a survey. *Med Eng Phys* 2007; **29**(4): 413–431.
16. Rossi A, Trevisani A, Zanotto V. A telerobotic haptic system for minimally invasive stereotactic neurosurgery. *Int J Med Robotics Comput Assist Surg* 2005; **1**(2): 64–75.
17. De Lorenzo D, Manganelli R, Dyagilev I, *et al*. Miniaturized rigid probe driver with haptic loop control for neurosurgical interventions. In 3 rd IEEE RAS and EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), 2010; 522–527.
18. Berkelman PJ, Whitcomb LL, Taylor RH, *et al*. A miniature microsurgical instrument tip force sensor for enhanced force feedback during robot-assisted manipulation. *IEEE Trans Rob Autom* 2003; **19**(5): 917–921.
19. Puangmali P, Althoefer K, Seneviratne LD. Mathematical modeling of intensity-modulated bent-tip optical fiber displacement sensors. *IEEE Trans Instrum Meas* 2010; **59**(2): 283–291.
20. Tan U, Yang B, Gullapalli R, *et al*. Triaxial MRI-compatible fiber-optic force sensor. *IEEE Trans Rob* 2011; **27**(1): 65–74.
21. Puangmali P, Althoefer K, Seneviratne LD, *et al*. State-of-the-art in force and tactile sensing for minimally invasive surgery. *IEEE Sens J* 2008; **8**(4): 371–381.
22. Clatz O, Delingette H, Talos IF, *et al*. Robust nonrigid registration to capture brain shift from intraoperative MRI. *IEEE Trans Med Imaging* 2005; **24**(11): 1417–1427.
23. Ohue S, Kumon Y, Nagato S, *et al*. Evaluation of intraoperative brain shift using an ultrasound-linked navigation system for brain tumor surgery. *Neurol Med* 2010; **50**(4): 291–300.
24. Mirota DJ, Hager GD, Ishii M. Vision-based navigation in image-guided interventions. *Annu Rev Biomed Eng* 2011; **13**: 297–319.
25. The Image-Guided Surgery Tooklkit [homepage on the Internet]. New York: Kitware Inc.: http://www.igstk.org 2012.
26. Enquobahrie A, Cheng P, Gary K, *et al*. The image-guided surgery toolkit IGSTK: an open source C software toolkit. *J Digital Imag* 2007; **20**: 21–33.
27. Yaniv Z, Cheng P, Wilson E, *et al*. Needle-based interventions with the image-guided surgery toolkit (IGSTK): from phantoms to clinical trials. *IEEE Trans Biomed Eng* 2010; **57**(4): 922–933.
28. Tobergte A, Pomarlan M, Hirzinger G. In Robust multi sensor pose estimation for medical applications. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2009; 492–497.
29. Unsgård G, Solheim O, Lindseth F, *et al*. Intra-operative imaging with 3D ultrasound in neurosurgery. *Intraop Imaging* 2011; **109**: 181–186.
30. Kais M, Millescamps D, Bétaille D, *et al*. A multi-sensor acquisition architecture and real-time reference for sensor and fusion methods benchmarking. In *IEEE Intelligent Vehicles Symposium*. 2006; 418–423.
31. De Momi E, Cerveri P, Gambaretto E, *et al*. Robotic alignment of femoral cutting mask during total knee arthroplasty. *Int J Comput Assist Radiol Surg* 2008; **3**(5): 413–419.
32. Schwald B, Seibert H. Registration tasks for a hybrid tracking system for medical augmented reality. *Journal of WSCG* 2004; **12**: 411–418.
33. Peters H, Raczkowsky J, Woern H. Approach to an architecture for a generic computer integrated surgery system. In IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2005; 2455–2460.
34. De Momi E, Ferrigno G. Robotic and artificial intelligence for keyhole neurosurgery: the ROBOCAST project, a multi-modal autonomous path planner. *Proc Inst Mech Eng H J Eng Med* 2010; **224**(5): 715–727.
35. Comparetti MD, De Momi E, Vaccarella A, *et al*. Optically tracked multi-robot system for keyhole neurosurgery. In IEEE International Conference on Robotics and Automation (ICRA), 2011.
36. Reintsema D, Preusche C, Ortmaier T, *et al*. Toward high-fidelity telepresence in space and surgery robotics. *Presence: Teleoperat Virtual Environ* 2004; **13**(1): 77–98.
37. Emmerich W. In: Software engineering and middleware: a roadmap. Proceedings of ACM Conference on the Future of Software Engineering, 2000; 117–129.
38. Tanaka K, Hori K, Kuroda T, *et al*. Integrated control of tele-surgery robot system using CORBA middleware. *Japan J Med Inf* 2004; **24**(1): 45–53.
39. Mönnich H, Botturi D, Raczkowsky J, *et al*. System architecture for workflow controlled robotic surgery. *J Inform Technol Healthcare* 2009; **7**(6): 345–352.
40. Eugster PT, Felber PA, Guerraoui R, *et al*. The many faces of publish/subscribe. *ACM Comput Surv* 2003; **35**(2): 114–131.
41. The ACE Orb [homepage on the Internet]. St. Louis: Object Computing, Inc. 2009: http://www.theaceorb.org
42. The Adaptive Communication Environment [homepage on the internet]. Nashville: Vanderbilt University: www.dre.vanderbilt.edu/ACE/ 2012.
43. Vaccarella A, Khreis G, Comparetti MD, *et al*. Data acquisition architecture for 3D ultrasound: temporal calibration. Proceedings of the 25th International Congress and Exhibition. *Int J Comput Assist Radiol Surg* 2011; **6**(Suppl 1): 37–38.
44. Open Source Computer Vision library (OpenCV) [homepage on the Internet]. Menlo Park: Willow Garage Inc., c2008–10: http://opencv.willowgarage.com
45. De Momi E, Cerveri P, Gambaretto E, *et al*. Robotic alignment of femoral cutting mask during total knee arthroplasty. *Int J Comput Assist Radiol Surg* 2008; **3**(5): 413–419.
46. Horn BKP. Closed-form solution of absolute orientation using unit quaternions. *J Opt Soc Am A* 1987; **4**(4): 629–642.
47. Nilsson J. Implement A continuously updating, high-resolution time provider for windows. *MSDN Mag* 2004: 78–88.
48. IGSTK-supported tracking systems [webpage on the internet]. New York: Kitware Inc: http://public.kitware.com/IGSTKWIKI/index.php/Supported_Tracking_System 2012.
49. Vaccarella A, Cerveri P, De Momi E, *et al*. A new IGSTK-based architecture for the integration of multimodal sensors and robots in neurosurgical robotics applications. Proceedings of the 24th International Congress and Exhibition, June 23–26. *Int J Comput Assist Radiol Surg* 2010; **5**: 308–309.
50. Bezet O, Cherfaoui V. Timestamping uncertainties in distributed data acquisition systems. In Proceedings of the IEEE Instrumentation and Measurement Technology Conference (IMTC), 2005; 2142–2147.
51. Correll K, Barendt N, Branicky M. Design considerations for software only implementations of the IEEE 1588 precision time protocol. In Conference on IEEE 1588 Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, 2005.
52. Park S, Aglyamov SR, Emelianov SY. Elasticity imaging using conventional and high-frame rate ultrasound imaging: Experimental study. *IEEE Trans Ultrason Ferroelectr Freq Control* 2007; **54**(11): 2246–2256.
53. Dai Y, Tian J, Yan G, *et al*. Real-time visualized freehand 3D ultrasound reconstruction based on GPU. *IEEE Trans Inform Technol Biomed* 2010; **14**(6): 1338–1345.