Vis Comput (2010) 26: 639–648
DOI 10.1007/s00371-010-0445-y

ORIGINAL ARTICLE

# Collaborative telemedicine for interactive multiuser segmentation of volumetric medical images

**Seunghyun Han · Niels A. Nijdam · Jérôme Schmid · Jinman Kim · Nadia Magnenat-Thalmann**

Published online: 28 April 2010

**Abstract** Telemedicine has evolved rapidly in recent years to enable unprecedented access to digital medical data, such as with networked image distribution/sharing and online (distant) collaborative diagnosis, largely due to the advances in telecommunication and multimedia technologies. However, interactive collaboration systems which control editing of an object among multiple users are often limited to a simple "locking" mechanism based on a conventional client/server architecture, where only one user edits the object which is located in a specific server, while all other users become viewers. Such systems fail to provide the needs of a modern day telemedicine applications that demand simultaneous editing of the medical data distributed in diverse local sites. In this study, we introduce a novel system for telemedicine applications, with its application to an interactive segmentation of volumetric medical images. We innovate by proposing a collaborative mechanism with a scalable data sharing architecture which makes users interactively edit on a single shared image scattered in local sites, thus enabling collaborative editing for, e.g., collaborative diagnosis, teaching, and training. We demonstrate our collaborative telemedicine mechanism with a prototype image editing system developed and evaluated with a user case study. Our result suggests that the ability for collaborative editing in a telemedicine context can be of great benefit and hold promising potential for further research.

**Keywords** Telemedicine · Teleradiology · Multi-user segmentation

## 1 Introduction

Moving ahead from simple text, image, and video communications, modern telemedicine applications are capable of supporting complex communications and networked multimedia technologies [26] and have led to many healthcare benefits, e.g., telesurgery applications that use image-guided robotics and real-time consultation (teleconferencing) for distance surgery [15] and telemonitoring systems for elderly prevention and care by using wireless sensory devices to communicate the status/condition of patients with physicians [19]. In particular, teleradiology, a telemedicine system that involves the electronic transmission of digital radiographic images (e.g., Magnetic resonance imaging (MRI), and X-ray) from one geographical location to another, has revolutionized the healthcare practices by enabling efficient distribution and sharing of medical images. Teleradiology has evolved from teleconferencing and shared 2D images running on imaging workstations with local area network (LAN) [27] to current state-of-the-art systems that are capable of streaming volumetric medical images in real-time via a wireless network [18]. Moreover, these modern systems now support thin client/cloud computing [1] that processes (e.g., volume rendering) and stores all the medical image data in a client–server (C/S) architecture.

S. Han (✉) · N.A. Nijdam · J. Schmid · J. Kim · N. Magnenat-Thalmann
MIRALab, University of Geneva, Geneva, Switzerland
e-mail: han@miralab.ch

N.A. Nijdam
e-mail: nijdam@miralab.ch

J. Schmid
e-mail: schmid@miralab.ch

J. Kim
e-mail: jinman.kim@miralab.ch

N. Magnenat-Thalmann
e-mail: thalmann@miralab.ch

However, despite the remarkable growth in telemedicine capabilities for distributed/networked applications, little progress has been made in the mechanism that enables real-time interactive collaboration among multiple users in, e.g., teleradiology. In collaborative telemedicine applications, typical tools include the usual text- and video-based interaction, image navigation, and in more sophisticated systems, image editing functions which we refer to hereon as "collaborative editing." Basic collaborative editing involves appending information to the image without changing the image's state, e.g., textual annotations and measurements using a ruler or simple brightness/contrast state changes of an image and lookup table of the image stack using a simple strict locking concurrency control mechanism [18, 22]. Unfortunately, these studies limit real-time interactive performance of users because only a single user, who has ownership of the shared object (lock of the object), can edit the object, while the others have to wait until the lock is released by the current owner. Such mechanism fails to provide sufficient scalability in terms of interactivity among a group of users. These studies also limit shareability of information because they assume that all the shared data is located on a specific server.

In this study, we present a collaborative telemedicine system for real-time and interactive segmentation of volumetric medical images. In our approach, we relax the temporal and spatial constraints for collaborative editing. Unlike with conventional strict locking mechanism, our approach allows multiple users to simultaneously update the shared medical images without waiting for the lock to be released by a user. To achieve this, we introduce a two-tier sharing architecture [8] with an iterative simulation process. In our approach, multiple users can request updates on the shared data within a time range (one simulation cycle), where these requests are used to the simulation parameters and let wrong inputs be amended by simulation algorithm iteratively and progressively. This is an extension to our previous study [21], in which the concept of using a collaborative mechanism for real-time interactive medical image segmentation among multiple users was demonstrated by prototyping scenarios for education and clinical usage. To utilize the diverse medical data scattered in the distributed sites for collaborative telemedicine, we relax a spatial constraint in the traditional C/S architecture which exploits the publish/subscribe paradigm [6] with a "Group" concept which provides functionality for efficiently formulating a collaborative session and supporting interactions among users in the session. In our approach, each site can be a server node if it has the medical data to be shared by a group of users. A user can load the medical data, and other users can join to the session at anytime.

Two case scenarios, representing the potential usage of our collaborative telemedicine system, are evaluated with a user survey to measure the usefulness and the practicability of our system in real usage situations. The two scenarios are (i) "Teacher–Students" scenario: An experienced physician is over-looking and guiding the iterative segmentation of an image by one or more trainees; and (ii) "Expert–Expert" scenario: Two or more experts are simultaneously segmenting the same volumetric image data. Our user survey results suggest that the proposed system is a promising collaborative system for the evaluated scenario applications.

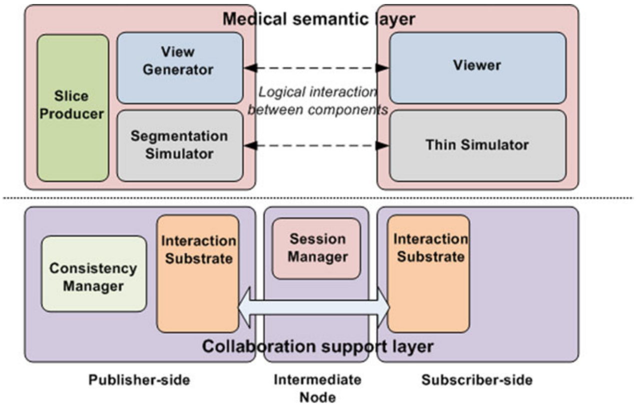


**Fig. 1** Overview of our proposed system architecture for collaborative editing. The medical semantic layer (*top*) handles the simulation and user interaction, while the collaborative support layer (*bottom*) provides the collaboration mechanism

The rest of the paper is as follows. In Sect. 2, different components of the proposed framework are presented. This is followed by the introduction of our proposed collaborative editing mechanism and interactive segmentation algorithm in Sect. 3 and 4, respectively. Then, implementation details are given in Sect. 5. The experimental results and user case study are analyzed and discussed in Sects. 6 and 7, respectively. Finally, concluding remarks are stated in Sect. 8.

## 2 The collaborative telemedicine system

Our collaborative telemedicine system consists of two architectural layers: the collaboration support layer and medical semantic layer as shown in Fig. 1. The first layer provides mechanisms to enable real-time interactive collaboration among the users by relaxing the temporal and the spatial constraints that limit the conventional collaborative medical systems. The second layer contains the simulation of the medical data and also the user interface components that controls the simulation.

### 2.1 Collaboration support layer

To make users exploit the medical data in their local site for sharing during collaboration, a user who has the data can become a host, according to the event-based publish/subscribe paradigm [6]. In this approach, medical data can be shared from any user and does not need to be located in a specific server. However, it does not provide sufficient methods to formulate a group for collaboration and to customize interactions among users in the group adapting to application semantics or policies. For this, the Group concept is introduced on top of the publish/subscribe paradigm. A Group is defined as a collection of distinct entities that share the same contexts satisfying constraints, according to

$$Group = \{e | f(e) : constraints\} \tag{1}$$

where $e$ is an entity, and $f$ is a context function. An entity $e$ is a user or an object. Each entity has its own property and can be included in multiple distinct groups. A context function $f$ is defined with properties of entities and external contexts, e.g., device capabilities and network conditions.

The Collaboration support layer consists of three main components: Session Manager, Interaction Substrate, and Consistency Manager which utilize the "Group" concept to relax the spatial constraint. We define a session, which is inherited from group component, as a group of users who share the same interest on a medical dataset. It provides users with the interfaces for entering or leaving its collaborative editing of medical data and membership management and contains specific constraints applied to the session. Session can easily control membership using the basic functions of the group (e.g., add and remove). Multiple sessions are managed in an intermediate node which runs on a separate machine. For dynamic management of sessions, the multisession controller, which holds a list of session references, provides users with the interfaces for initiation, termination, selection, join, leave, creation, and deletion of a medical collaboration session. Once a user initiates the collaboration session and loads a medical data, multiple users can connect to the session and start the collaboration. When one of the users selects a segmentation command, an iteration of the segmentation is executed, and the results are relayed to all the users in the session. The results are composed of the image data (slice) and segmentation data (segmentation overlay). At any point, any user can collaboratively segment the image by adjusting the segmentation parameters via interactive image annotation. These parameters are then inserted as new constraints in the segmentation and used in the next segmentation iteration process which will be described in detail in Sect. 4.

To enable users to interact with each other, a set of well-defined interaction protocol is required. Group provides basic functions, e.g., union, intersection, and difference. Application developers can define their own context function $f$ to create a group based on application semantics. For example, to make a subset of members interact with each other, filter function can be defined, and to broadcast to other groups, aggregation function can be defined as follows:

A filter is a method that makes a subgroup from a group, which holds the following conditions:

- If $x$ is a member of a group $A$, then $x$ is also a member of group $B$, where $B = f(A)$
- $B \subset A$, where $B = \mathit{filter}(A)$

An aggregation is a method that makes a supergroup from groups, which holds the following conditions:

- If $x$ is a member of a group $A$, then $x$ is also a member of group $B$, where $B = f(A)$
- $A \subset B$, where $B = \mathit{aggregation}(A)$

Apart from traditional Distributed Virtual Environment systems (DVEs) [13, 16, 23] which mostly exploited spatial relationship, e.g., region of interest [5, 13], to filter interaction events, in our system, context function is used to manage interactions among users. A context function is formulated with the current contexts of devices, such as network and computational resources, display size, etc. It gives freedom to developers to make filters with specific application requirements. In the current implementation, we developed an exemplary context function which allows users to participate in a collaborative session with diverse devices, e.g., Ultra-Mobile PC (UMPC), desktop computer, without degrading interactive performance of users.

One of the key issues in the Consistency Manager is to provide ways to manipulate the shared medical data concurrently and to share consistent views among the users. The proposed system provides a monolithic view as well as polymorphic views sharing among users in a session. The former implies that a group of user shares the exactly same view of the shared data, and the later implies that a group of users has the different views of the shared data but user interactions are shared among users in both approaches. The proposed system provides a policy-based optimistic concurrency control scheme which allows users to update objects without conflict checks with others and thus to interact with objects more naturally exploiting application contexts. More details will be described in Sect. 3.

### 2.2 Medical semantic layer

The medical semantic layer provides the segmentation simulation, rendering of the 3D medical data and the application policies to the collaboration support layer, which allows customizing the interactions among the users and the collaboration algorithms. The core components are the Segmentation Simulator, View Generator, Slice Producer, Thin Simulator, and the Viewers, as shown in Fig. 1.

The Segmentation Simulator is encapsulated into a service and continuously provides the deformed mesh after a simulation cycle to its subscribers. The Slice Producer and View Generator take the output of the simulation and

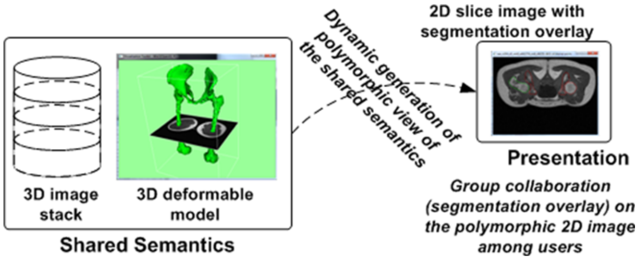

2D slice image with segmentation overlay
Dynamic generation of polymorphic view of the shared semantics
3D image stack
3D deformable model
Shared Semantics
Presentation
Group collaboration (segmentation overlay) on the polymorphic 2D image among users

**Fig. 2** Presentation-Semantics two-tier split model

process it further to generate views on the subscribers. The Thin Simulator on the subscriber, which is a logical correspondence of the Segmentation Simulator on the producer side, is capable of sending commands to the simulation in order to modify its parameters. The Thin Simulator also subscribes to the Slice Producer for the visualization of the current slice. Through its interface, an input taken from keyboard and mouse is translated into simulation commands and sent to the Segmentation simulation service. The Slice Producer processes the deformed mesh into a contour plot for the Thin Simulators. Whenever a subscriber changes a slice, the Slice Producer sends the corresponding MRI image to the subscriber. The Viewer on subscriber side, which provides 3D visualization of the full mesh (segmentation), is separate from the Thin Simulator. The Viewer presents a remote rendered 3D representation provided by the View Generator on publisher side. The View Generator takes the deformed mesh and renders the 3D mesh. The resulting 2D image is compressed and sent to the subscribers. Another functionality of the View Generator is to generate user interfaces adapting to the current device capabilities of the subscribers.

## 3 Collaborative editing mechanism

To enable real-time and interactive editing over conventional methods [3, 13], such as strict "locking" concurrency control, a two-tiered sharing approach [8] that consists of a semantics-tier and a presentation-tier is used as shown in Fig. 2.

This approach enables the semantics (3D Image + 3D deformable models) to be shared and thus can instantiate more than one presentation (2D slice image + segmentation overlay), which is in term, shared among all the users in a collaborative session. A polymorphic presentation of the shared semantics is dynamically generated when a group of users request edits (e.g., new slice position on the stack, B/C change, segmentation constraints) within a simulation cycle. The generated presentation is replicated to each user of the group, not only to support direct manipulation on the replicated presentation but also to continuously support the users to collaborate even in the event of a transient network failure. Users can temporally manipulate, update, and annotate on the presentation within a simulation cycle.

In our approach, rather than accepting inputs from only a single subscriber within a time frame, multiple parameters (changes) from multiple subscribers are used in the segmentation process. Parameters can be global (e.g., weight coefficients) or local (e.g., local constraints) as used in an example in Sect. 4. This mechanism allows the users to manipulate objects without waiting to get ownership of the lock to update the segmentation overlay. However, in collaborative editing, it is difficult to identify which input parameters are responsible for the segmentation evolution, especially when the segmentation result is unexpected, due to all inputs from the users having an influence on the segmentation evolution. A common method to track the parameter for segmentation is to implement a rollback mechanism [5]. Rollback is the process of recording all the input parameters which can be used to "roll" back to the previous input in the segmentation evolution. Nevertheless, it is hard to apply such rollback mechanism because this makes the system complex for the users in undoing and redoing of the previous actions.

To overcome this, three strategies are proposed. Firstly, segmentation algorithm must be iterative, thereby enabling different users to amend, stop, and resume the segmentation process. Most importantly, each iteration has to provide intermediate results that provide sufficient feedback to the user for correct interpretation. This provides efficient monitoring of the algorithm evolution. Secondly, changes made by the users have an immediate influence in its local view so that the user may receive instant feedback. Thirdly, weight factor policy is used to fuse inputs from the users. For example, inputs from an expert have higher weight than inputs from student in our "Teacher–Students" use case scenario.

## 4 Multiuser iterative image segmentation

Our segmentation algorithm is based on an iterative and progressive evolution of physically based discrete deformable models [7, 20, 24]. In our case, these deformable models are represented by 2-Simplex meshes [4] that deform under the influence of "forces." Each mesh vertex is considered as a particle with mass whose state (position and velocity) is derived from the Newtonian law of motion and the applied forces. At each iteration step, the particles state is updated by an implicit Euler numerical integration. External forces are based on image information (e.g., gradients, intensity distribution) to drive the model towards the desired anatomical boundaries. Conversely, internal forces enforce the mesh geometry to respect smoothness and shape constraints. Shape constraints derive from statistical shape models (SSM) [10] that ensure that meshes can only adopt valid configurations expressed by statistics inferred from a collection of training shapes. SSMs proved to be very efficient and robust in medical image segmentation [9] and have been

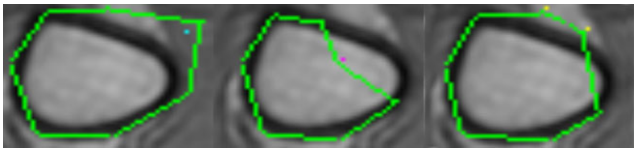


**Fig. 3** Example of constraint points on an MRI image. The *blue* (*left*), *red* (*middle*) and *yellow* (*right*) points represent the internal, external and frontier points, respectively. The deforming mesh is in *green*

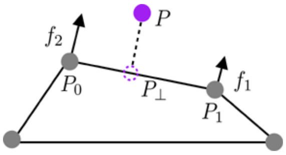


**Fig. 4** Illustration of internal constraint points (CPs): The closest face ($P_0$; $P_1$) to the CP $P$ is attracted by creating 2 forces $f_1$ and $f_2$ on $P_0$ and $P_1$, respectively, whose calculation depends on $P$ and its projection $P_\perp$ on the face

successfully applied to segment a wide variety of structures (e.g., bone [12, 20], liver [9], and bladder [2]).

Our segmentation algorithm allows the simultaneous segmentation of various structures of interest. To cope with models interpenetration, efficient collision detection and response [25] are implemented. Coupled with a multiresolution approach (from coarse to fine), a fast and interactive segmentation algorithm is derived. In order to monitor and possibly correct the algorithm evolution, interactive control must be provided to the users [17], especially in our context of collaborative editing. This is achieved by the means of "internal", "external", or "frontier" constraint points that deform the mesh so that the points are respectively in the interior, at the exterior, or on the surface of the mesh (see Fig. 3 and [21]).

In practice, constraint points attract or repel meshes by creating forces on some vertices. An example of an internal constraint point $P$ is depicted in Fig. 4, where the closest face of the mesh, here represented by two vertices $P_0$ and $P_1$, is attracted under the action of two forces $f_1$ and $f_2$. Each force $f_i$ is computed as

$$f_i = \alpha * w_i(P - P_\perp) \tag{2}$$

where $w_i$ is the barycentric weight computed from the projection $P_\perp$ of $P_p$ on the face, and $\alpha$ denotes a global weighting coefficient specific to the constraint point type (internal, external, or frontier). These external forces have a local influence (closest faces are only affected), while the modification of the force weight $\alpha$ can globally affect the segmentation since all constraint point forces of same type share the same weight. The use of the weights in our collaborative editing mechanism is discussed in the following sections and explains how such weights can be tuned to account for the various collaborative segmentation scenarios.

This segmentation algorithm is thus a good candidate for our collaborative application as it fulfills the requirements defined in Sect. 3 and allows the concurrent segmentation of multiple structures. In this case, the model's contour and the constraint points are overlaid in the slice and represent what was previously denoted in Sect. 3 as "segmentation overlay" (see also Fig. 2).

## 5 Collaborative telemedicine system implementation

For collaborative segmentation, we have selected a volumetric MRI data of the lower limbs. This data was selected to provide an interesting multiuser case scenario, where the segmentation is challenging due to the poor image quality and inhomogeneous intensities within its structure (imposed by hardware and protocol restrictions) [20] and thus stands to benefit largely from multiple users to concurrently aid in the segmentation process. Moreover, this data offer many different types of anatomical structures that need to be segmented, e.g., bones and various types of muscles, thus benefiting from the collaborative segmentation of multiple structures concurrently.

### 5.1 Scenarios

In order to illustrate the concepts of the proposed collaborative telemedicine system, two scenarios were designed and evaluated.

*Teacher–Students scenario.* A teacher shows to students the reading of medical images and the concept of semi-automatic segmentation for ROI annotation on these images. In this scenario, both the teacher and the students have access to their own computers which have our proposed collaborative editing medical viewer installed. A group session is started, and all the students observe the segmentation evolution as the teacher initiates the segmentation and changes the viewing slice across the dataset. Then the teacher shows how to manually and locally correct the segmentation evolution by inserting various constraint points. Once this simple principle is explained, students are invited to interact with the segmentation process. The teachers are assigned greater "weight" to the constraint points than the students, thus, enabling the teacher to override students' constraints.

*Expert–Expert scenario.* Two (or more) experts are segmenting the same dataset. Each of them can monitor and modify the segmentation in different parts (structures) of the volumetric dataset hence expediting the segmentation process. As with the Teacher–Students scenario, when experts work in the same slice, weights may be allocated based on the experience of the users, thus giving more priority on the segmentation constraints to the more experienced user.

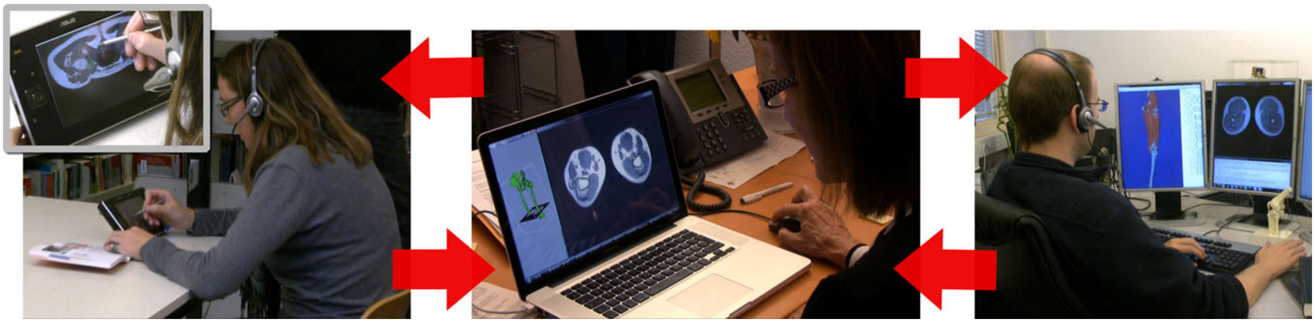


**Fig. 5** A collaborative segmentation scenario. A teacher (*center*) uses her laptop to monitor and guide the two students in a collaborative segmentation session, where one student is using a portable device (*left*), and the other is using a conventional workstation (*right*)

### 5.2 Prototype development

A publisher owns the semantics and constitutes the 3D processing modules in Medical Semantic Layer (Fig. 1), while subscribers share and edit the polymorphic presentations. A publisher has a 3D rendering view and a 2D view. In this current implementation, the 3D rendering is only for illustrative purposes and not used in the collaborative editing, and only the 2D view is distributed to the users. The views can be synchronized, which means that all the users observe the same slice. This particular mode is appropriate in the Teacher–Students scenario described in Sect. 5.1, in which the teacher first segments, while the students only observe as shown in Fig. 5.

A second mode consists in letting the users to view a slice independently to the other users in the same session. This option is essential for the second case scenario, in which experts can segment different parts of the same dataset. The 2D viewer consists of the standard image manipulations controls: LUT change, B/C, scaling, and annotation, in addition to be able to change image slices (moving up and down in the image stack). Each subscriber is assigned to a unique annotation color and can either add or remove constraint points on the image (via mouse selection) as a segmentation parameter. Per iteration cycle, the publisher compiles all the parameters and executes the control based on time-stamp (first-come, first-serve) and event management in Interaction substrate component. Multiple manipulations are possible, e.g., LUT change and segmentation change, in a single iteration cycle.

## 6 Performance analysis

We evaluated the interactive performance of our proposed collaborative telemedicine system with two example scenarios (in Sect. 5.1). In the first experiment, the Teacher–Students scenario was simulated, where 20 users concurrently joined in a collaborative session, where one user (acting as the teacher) operated on an image volume (260 × 511 × 242) consisting of 21 models of the muscles, while the other users (acting as the students) observed the segmentation. In the second experiment, the Expert–Expert scenario was considered, i.e., two users with their own computers were requested to concurrently segment four different bone models on an volumetric MRI (dimensions of 483 × 358 × 270 voxels). We configured subscribers and a publisher with Windows XP, P4 CPU at 3.40 GHz, 2 GB RAM, GeForce 7800GT connected to LAN. For each experiment, we measured the frame per second (fps) on both the publisher and the subscriber sides. We averaged the fps over 100 segmentation iterations (simulation time period). In the first experiment, we had 3.8 fps on the publisher and 3.2 fps on the subscribers on average. In the second scenario experiment, we had 4.6 fps on the publisher, and the averaging of the subscribers resulted in 4.2 fps. Simulation results show that our system does not expose significant degradation in system responsiveness, while the number of participants is increased.

Performance was mostly bounded by the segmentation iteration on the publisher. A subscriber, which only needs to display the transmitted images and send segmentation parameter requests, used less than 1 MB of memory and less than 1% of CPU consumption. The complexity of the segmentation, in terms of memory consumption and computation speed, was mostly related to the size of the segmented image and the number of structures (models) that were simultaneously segmented. These factors were expected to also have an impact on the system as the image data, and the models contours (overlaid on the slice at the subscriber side) need to be sent over the network.

## 7 User case study

In order to measure the usability of our collaborative system, we ran a controlled experiment with 10 participants (5 males and 5 females; ages between 20 to 36 years). Among these 10 participants, two were considered experts (users with experiences in medical image analysis), and eight were included as students.

### 7.1 Experiment design

We conducted repeated measures experiments, where the same subjects were used for the two scenarios in Sect. 5. In this experiment, the only user feedback variable manipulated is that of "type of collaborative editing algorithm." Two dependent variables were measured as follows:

- Task Completion Time: time taken to complete the segmentation on a single image.
- Segmentation Error: errors between segmented and the reference segmentation.

We also measured user satisfaction using questionnaire as follows:

- User Satisfaction: which type of algorithm fulfilled the best requirements for the specific scenario?

We devised a questionnaire based on a well-known IBM Post-Study questionnaire framework [14]. Ten questions were designed to measure the proposed system's usefulness based on 5-point graphic scales, anchored at the end points with the terms "Strongly agree" for 1 and "Strongly disagree" for 5. Spaces were available at the end of the questionnaire for user comments.

### 7.2 Tasks and procedures

Participants were asked to learn our system with a simple introductory manual which exemplifies the usage of the system. All participants had no prior exposure to the application. After participants were given only a basic introduction and a few minutes to play with the application, they were then further instructed on the use of the application by participating in the Teacher–Students scenario.

Participants were given two tasks: each participant had to collaborate with another user in a segmentation task on an image using firstly the strict locking algorithm and then our proposed collaborative mechanism. Pairs of participants were randomly chosen. The segmentation consisted of segmenting four bone structures on an MRI (single slice image) as accurately as possible within the shortest time possible. Bone models were initialized sufficiently close to the structures to be segmented so that the users could drive the meshes towards the structures through applying the three types of constraint points (see Sect. 4). The ground truth segmentation was a priori computed by experts for use as a benchmark in calculating the segmentation error. For a given mesh contour resulting from a user segmentation, an error $e_1$ was computed as the Euclidean distance between each point of the contour and its closer point on the reference contour. The same was done by inverting roles of user and reference contours to get an error $e_2$. The final segmentation error for a mesh was thus the sum $e_1 + e_2$. By averaging the errors for all the bones, average symmetric distance error in millimeter (mm) was calculated. After completing the tasks, participants filled in a questionnaire.

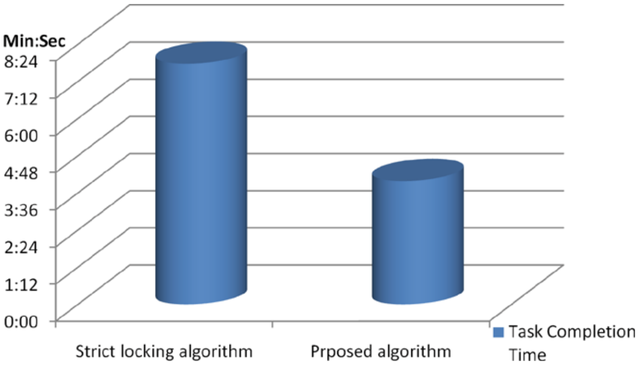


**Fig. 6** Comparison of task completion time for stick locking and our proposed collaborative mechanisms

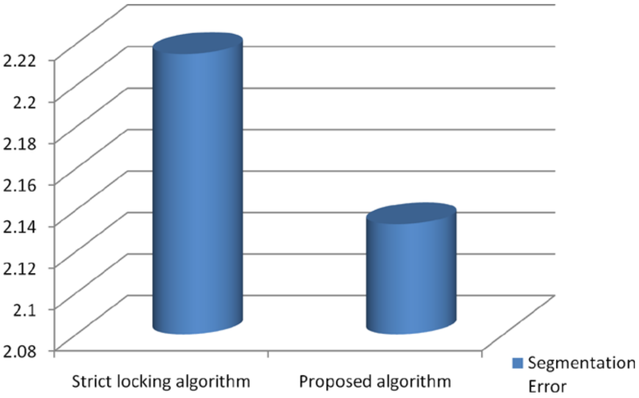


**Fig. 7** Segmentation error

### 7.3 Results

We calculated discrete statistics for every dependent variable and user satisfaction from the conducted survey.

*Task Completion Time.* Figure 6 shows the average time taken by nonexpert participants to complete their tasks. Using the proposed system, participants finished their task earlier compared to the strict locking-based system because participants could update their inputs without waiting the other user to release the lock. Another reason is that each participant could work on different model contours at the same time. We observed that in some cases with strict locking approach, a participant did not release the lock preventing the other participant to improve the segmentation even though she/he had a better idea to complete the task.

*Segmentation Error.* The obtained results revealed that participants' segmentation error using strict locking was 4% greater on average than using our proposed mechanism, as shown in Fig. 7. From our observation, this was attributed to

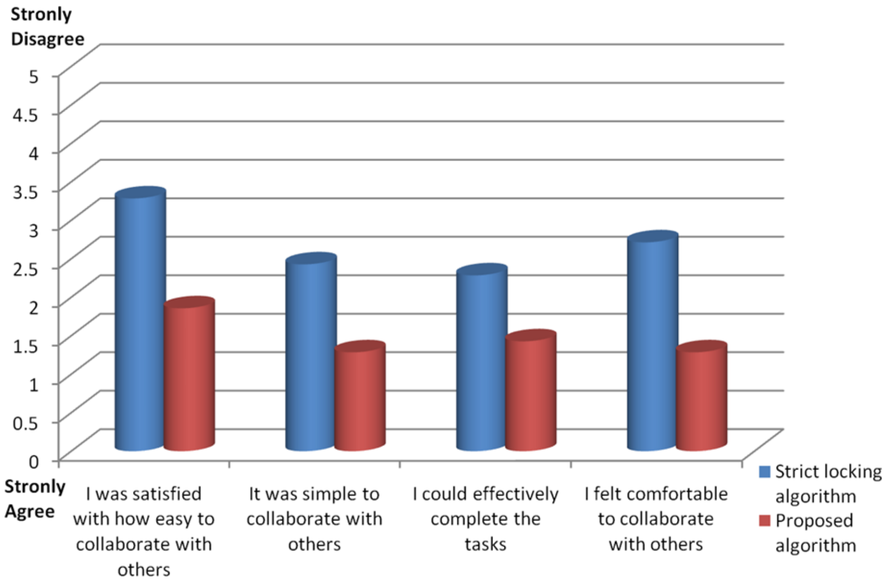


**Fig. 8** Average of the selected questions in questionnaire

the fact that the participants were able to correct each other's errors, where the agreements from both the participants were more accurate than from an individual user.

*User Satisfaction.* From the questionnaires a qualitative user satisfaction analysis was derived. Most participants were strongly or moderately satisfied with the collaborative algorithm as shown in Fig. 8. No participant strongly disapproved the use of both the proposed algorithm and the strict locking algorithm. Similarly, participants reported that they could effectively complete their tasks both with the proposed algorithm and strict locking algorithm. Importantly, participants felt more comfortable to collaborate with others with the proposed algorithm because more freedom was given and it was not necessary to wait for the lock to be released.

Participants were asked to comment in the questionnaire. Two participants reported that strict locking algorithm needed a floor control which manages participants' turn to access the shared object or at least other communication facilities, such as chat, to negotiate their turn because they felt uncomfortable to wait until lock is released. Two participants reported that they felt a little bit uncomfortable to interact with others because the simulation responsiveness was relatively low in both cases since their inputs (constraint points) progressively affected the segmentation instead of having an instantaneous effect. As explained in Sect. 3, the progressive changes of the segmentation evolution are essential to be able to detect and correct errors sufficiently early. Finally, in the context of our specific physically based segmentation, creating large "brutal" changes may create instabilities in the simulation and thus ultimately affect the quality of the segmentation. Further, as mentioned in Section 6, a more powerful workstation for the publisher would yield a faster simulation.

## 8 Concluding remarks

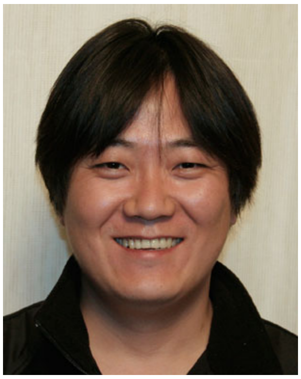
In this paper, we proposed a collaborative telemedicine system for real-time and interactive segmentation of volumetric medical images and demonstrated its usefulness with two case scenarios: Teacher–Students and Expert–Expert collaborative editing. User evaluation was conducted which measured the enhancements with our approach in comparison to the conventional strict locking collaborative system. We conducted some preliminary tests on a UMPC over a wireless network, and similar performance was observed. Future work will mostly focus on dynamic polymorphic presentation (volumetric view and user interface) adaptation and context-aware network adaptation according to the current device capability in order to support variety of users to collaborate with each other exploiting diverse devices. Moreover, alternative segmentation approaches, e.g., based on ITK [11], will be also considered to improve the extensibility of our framework.

**Acknowledgements** We would like to thank our partners, the University Hospital of Geneva and the University College London for providing us with the MRI datasets. This work is supported by the InterMedia (NoE-IST-2006-038419) and the 3D Anatomical Human (MRTN-CT-2006-035763) projects funded by the European Union.

## References

1. Constantinescu, L., Kim, J., Chan, C., Feng, D.: Automatic mobile device synchronization and remote control system for high-performance medical applications. In: IEEE Proc. Engineering in Medicine and Biology Society (EMBS), pp. 2799–2802 (2007)
2. Costa, M.J., Delingette, H., Novellas, S., Ayache, N.: Automatic segmentation of bladder and prostate using coupled 3D deformable models. In: Int. Conf. on medical image computing and computer assisted intervention (MICCAI), pp. 252–260 (2007)

3. Delaney, D., Ward, T., McLoone, S.: On consistency and network latency in distributed interactive applications – a survey. Presence, Part I **15**(4), 465–482 (2006)
4. Delingette, H.: General object reconstruction based on simplex meshes. Int. J. Comput. Vis. **32**, 111–146 (1999)
5. Dollimore, J., Kindberg, T., Coulouris, G.: Distributed Systems: Concepts and Design. Addison-Wesley, Reading (2005)
6. Eugster, P., Felber, P., Guerraoui, R., Kermarrec, A.: The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003)
7. Gilles, B., Moccozet, L., Magnenat-Thalmann, N.: Anatomical modelling of the musculoskeletal system from MRI. In: Larsen, R., Nielsen, M., Sporring, J. (eds) MICCAI 2006, LNCS, pp. 289–296 (2006)
8. Han, S., Lee, D., Ko, I.: A deputy object based presentation semantics split application model for synchronous collaboration in ubiquitous computing environments. In: Proceedings of the Third International Conference on Collaboration Technologies, July (2007)
9. Heimann, T., Munzing, S., Meinzer, H., Wolf, I.: A shape-guided deformable model with evolutionary algorithm initialization for 3D soft tissue segmentation. In: N. Karssemeijer, B. Lelieveldt (eds) Proc. Int. Conf. Information Processing in Medical Imaging (IPMI), pp. 1–12 (2007)
10. Heimann, T., Meinzer, H.: Statistical shape models for 3D medical image segmentation: A review. Med. Image Anal. **13**, 543–563 (2009)
11. Ibanez, L., Schroeder, W., Ng, L., Cates, J., et al.: The ITK software guide. Kitware (2003)
12. Kainmuller, D., Lamecker, H., Zachow, S., Hege, H.-C.: An articulated statistical shape model for accurate hip joint segmentation. In: Proc. IEEE Engineering in Medicine and Biology Conference (EMBC), pp. 6345–6351 (2009)
13. Lee, D., Lim, M., Han, S., Lee, K.: ATLAS: A scalable network framework for distributed virtual environments. Presence **16**(2), 125–156 (2007)
14. Lewis, J.: IBM computer usability satisfaction questionnaires: psychometric evaluation and instructions for use. Int. J. Hum.–Comput. Interact. **7**(1), 57–78 (1995)
15. Marescaux, J., Leroy, J., Gagner, M., Rubino, F., Mutter, D., Vix, M., Butner, S.E., Smith, M.K.: Transatlantic robot-assisted telesurgery. Nature **413**, 379–380 (2001)
16. Morillo, P., Orduna, J.M., Fernandez, M., Duato, J.: Improving the performance of distributed virtual environment systems. IEEE Trans. Parallel Distrib. Syst. **16**(7), 337–649 (2005)
17. Olabarriaga, S., Smeulders, A.: Interaction in the segmentation of medical images: A survey. Med. Image Anal. **5**, 127–142 (2001)
18. Park, S., Kim, W., Ihm, I.: Mobile collaborative medical display system. Comput. Methods Programs Biomed. **89**(3), 248–260 (2008)
19. Rialle, V., Lamy, J.B., Noury, N., Bajolle, L.: Telemonitoring of patients at home: a software agent approach. Comput. Methods Programs Biomed. **72**(3), 257–268 (2003)
20. Schmid, J., Magnenat-Thalmann, N.: MRI bone segmentation using deformable models and shape priors. In: Metaxas, D., Axel, L., Szekely, G., Fichtinger, G. (eds) MICCAI 2008, Part I. LNCS, pp. 119–126 (2008)
21. Schmid, J., Nijdam, N., Han, S., Kim, J., Magnenat-Thalmann, N.: Interactive segmentation of volumetric medical images for collaborative telemedicine. In: Modelling the Physiological Human, Proc. 3D Physiological Human Workshop 5903, pp. 13–24 (2009)
22. Simmross-Wattenberg, F., Carranza-Herrezuelo, N., Palacios-Camarero, C., Casaseca-de-la-Higuera, P., Martín-Fernández, M., Aja-Fernández, S., Ruiz-Alzola, J., Westin, C., Alberola-López, C.: Group-Slicer: A collaborative extension of 3D-Slicer. J. Biomed. Informatics **38**(6), 431–442 (2005)
23. Singhal, S., Zida, M.: Networked Virtual Environments: Design and Implementation. Addison-Wesley, Reading (1999)
24. Snel, J., Venema, H., Grimbergen, C.: Deformable triangular surfaces using fast 1-D radial Lagrangian dynamics-segmentation of 3-D MR and CT images of the wrist. IEEE Trans. Med. Imaging **21**, 888–903 (2002)
25. Volino, P., Magnenat-Thalmann, N.: Implementing fast cloth simulation with collision response. Comput. Graph. Int. **2000**, 257–266 (2000)
26. Wootton, R., Craig, J., Patterson, V.: Introduction to Telemedicine, 2nd edn. The Royal Society of Medicine Press Ltd, London (2006)
27. Zhang, J., Stahl, J.N., Huang, H.K., Zhou, X., Lou, S.L., Song, K.S.: Real-time teleconsultation with high-resolution and large-volume medical images for collaborative healthcare. IEEE Trans. Inf. Tech. Biomed. **4**(2), 178–185 (2000)



**Seunghyun Han** is a senior researcher at MIRALab. He obtained his Ph.D. in computer science from the Information and Communications University, South Korea. During his academic years, he worked on ATLAS which is a network framework for networked virtual environments. He also has actively worked on how to support groups of users in shared ubiquitous computing environments. One of the significant outcomes of his research efforts is Active Surroundings which is a community-aware middleware for ubiquitous computing environments. Currently, he is participating in the EU project InterMedia. His research interests include the support of interactions among groups of users in both physical and virtual spaces.

**Niels A. Nijdam** is a Ph.D. candidate at MIRALab, University of Geneva. He obtained his M.Sc. in Computer Science at the Human Media Interaction group of The University of Twente in the Netherlands in 2007. His research interests include programmable graphics, distributed network rendering, and augmented reality.

**Jérôme Schmid** is an engineer in Computer Science specialized in imaging and virtual reality. He obtained his master degree in 2003 at the Superior National School of Computer Science and Applied Mathematics of Grenoble (ENSIMAG), France. He worked more than three years in the field of medical imaging, at IRCAD Institute, France, and at the Minimally Invasive Surgical Skills Centre of the Prince Of Wales Hospital, Hong Kong. Currently he is Ph.D. candidate at Miralab and works on medical imaging. His interests include image processing, segmentation, registration, and augmented reality.

**Jinman Kim** is a senior researcher at MIRALab. He received his B.Sc. (Honours1) degree in computer science and technology in 2001 and continued his study into a Ph.D. (Science) degree in 2005 working in the Biomedical and Multimedia Information Technology (BMIT) research group, University of Sydney, Australia. From 2006 until 2007, he was in charge of a research commercialization project, together with the Royal Prince Alfred (RPA) Hospital, which involved the research and development of a biomedical visualization system that is designed to work with large and complex image sets. In 2008, he was awarded a prestigious Australian research council postdoctoral (APD) fellowship. He is also a unit coordinator for a Master of IT unit of study—Project Management in IT. Currently, he is participating in the EU project 3dAH and FocusK3D. His research interests includes: Scientific visualization and complex data presentation, Multidimensional biomedical image processing; multimodal image representation, and m-health.

**Nadia Magnenat-Thalmann** has pioneered research into virtual humans over the last 25 years. She obtained her Ph.D. in Quantum Physics from the University of Geneva. From 1977 to 1989, she was a Professor at the University of Montreal, where she founded the research lab MIRALab. She was elected Woman of the Year in the Grand Montreal for her pioneering work on virtual humans and presented Virtual Marilyn at the Modern Art Museum of New York in 1988. Since 1989, she is Professor at the University of Geneva, where she recreated the interdisciplinary MIRALab laboratory. With her 15 Ph.D. students, she has authored and coauthored more than 500 research papers and books in the field of modeling virtual humans, interacting with them, and living in augmented worlds. She is presently the Coordinator of several European Research Projects such as the Network of Excellence (NoE) INTERMEDIA and the European Research training Network Marie Curie 3D ANATOMICAL HUMANS. She is the editor-in-chief of the Visual Computer Journal published by Springer Verlag and coeditor-in-chief of the journal Computer Animation and Virtual Worlds published by Wiley.