Christian Diaz
Helmuth Trefftz*
Grupo de Investigación I+D+I en TIC
Virtual Reality Lab
Universidad EAFIT
Medellin, Colombia

Lucia Quintero
Grupo de Investigación de
Modelado Matemático
Universidad EAFIT

Diego A. Acosta
Grupo de Investigación DDP
Universidad EAFIT

Sakti Srivastava
Department of Surgery
Stanford University

Collaborative Networked Virtual Surgical Simulators (CNVSS): Factors Affecting Collaborative Performance

Abstract

Stand-alone and networked surgical simulators based on virtual reality have been proposed as a means to train surgeons in specific surgical skills with or without expert guidance and supervision. However, a surgical operation usually involves a group of medical practitioners who cooperate as team members. To this end, CNVSS have been proposed for the collaborative training of surgical procedures in which users with different surgical roles can take part in the training session. To be successful, these simulators should guarantee synchronicity, which requires (1) consistent viewing of the surgical scene and (2) a quick response time. These two variables are affected by factors such as users' machine capabilities and network conditions. As far as we know, the impact of these factors on the performance of CNVSS has not been evaluated. In this paper, we describe the development of CNVSS and a statistical factorial design of experiments (DOE) to determine the most important factors affecting collaboration in CNVSS. From the results obtained, it was concluded that delay, jitter, packet loss percentage, and processor speed have a major impact on collaboration in CNVSS.

1 Introduction

Laparoscopic surgery is an innovative technique performed by using multiple, small incisions a few centimeters in length. The surgeon inserts long instruments and a tiny camera to perform the surgical procedure. The camera allows the surgeon to see the surgical area, and the long instruments allow the surgeon to interact with anatomical structures and perform operations such as cutting or attaching. Over the last 20 years, the advantages of these procedures (i.e., (1) decreased risk of infection for the patient and (2) shorter post-surgery recuperation time) have increased their clinical usage (Pugh et al., 2009; Székely et al., 2000). However, laparoscopic surgery demands more skillful surgeons because of the reduced workspace and lack of 3D perception of the anatomical structures not perceived by hand. Recently, advances in virtual reality technology, deformable objects modeling, and machine capabilities have allowed for the development of surgical simulators using virtual reality (Liu, Tendick, Clearly, & Kaufmann, 2003). These systems have been seen to provide better training for laparoscopic surgeons compared to traditional

Presence, *Vol. 22, No. 1, Winter 2013,* 54–66
doi:10.1162/PRES_a_00133

*Correspondence to htrefftz@eafit.edu.co.

methods (Aggarwal, Grantcharov, Moorthy, Hance, & Darcy, 2006; Lamata et al., 2006; Woodrum et al., 2006).

So far, three kinds of surgical simulators using virtual reality have been developed: stand-alone, networked, and collaborative networked virtual surgical simulators. Stand-alone virtual surgical simulators allow the student to receive training on virtual surgical procedures without expert mentoring or supervision. These kinds of simulators are able to record the surgical performance of the student, and depending on his or her skills, provide him or her with feedback and curriculum guidance. Networked virtual surgical simulators were developed due to the growing availability of computer networks and the decreased availability of expert laparoscopic surgeons in remote regions. This second kind of surgical simulator was created to allow a student to be trained remotely by an instructor. In such a system, the instructor can perform the procedure remotely while the student not only watches, but also feels (haptic feedback provided) what the instructor is touching without actually participating in the execution of the surgical procedure. Finally, considering that a surgical procedure involves teamwork among medical practitioners, a third kind of virtual surgical simulator, the collaborative networked virtual surgical simulator (CNVSS), was proposed by Tang et al. (2007). CNVSSs allow for the collaborative training of users located remotely with each member playing a role during the training session. Therefore, CNVSSs are not only useful for training basic surgical skills, but also for training team members to work collaboratively in a surgical room. These skills are required to perform a successful surgical procedure.

CNVSS have to guarantee effective collaboration among the users in order to successfully train participants. The quality of collaboration in CNVSS depends on the consistency of the shared surgical scene and the response time during the training to each user. Consistency in the surgical scene is possible when each user can view and interact with the same shared surgical scenario, and the response time is the time elapsed between a user command and the user receipt of an action, result, or feedback from the CNVSS.

In CNVSS, factors such as users' machine capabilities and network conditions can affect the response time and consistency of the shared state, affecting collaboration during the training session. However, as far as we know, none of the impacts of these factors on collaboration performed in CNVSS have been evaluated. Moreover, knowing which factors have a major impact on the level of collaboration achieved in CNVSS allows one to propose appropriate strategies and methods in order to mitigate the lack of consistency and provide a faster response time. For these reasons, in this paper we describe the development of CNVSS and a statistical design experiment in order to determine which factors affect collaboration in CNVSS.

The paper is structured as follows: Section 2 describes similar projects and opportunities of research. Sections 3 and 4 show the hardware, software, and methods required to perform the experiments. These sections also lay out the changes necessary in order to make simulation open framework architecture (SOFA) networked and collaborative (Allard et al., 2007), and statistical designs of experiments to determine the effect of factors on the performance of the collaboration in CNVSSs. Section 5 shows the results obtained and Section 6 describes the conclusions and future work.

2 Literature Review

Early works have evaluated how different factors such as jitter, delay, and packet loss degrade simulation and consistency in CNVSS (Liberatore, Cavusoglu, & Cai, 2003; Montgomery et al., 2002). Gunn, Hutchins and Adcock (2005) and Gunn (2007) describe how the performance of CNVSS is affected by jitter and network latency. They report that latency produces vibrations in the force affected by the haptic device and degenerates physical simulations of organs and tissues. However, they do not determine in which latency and jitter values these issues arise. To compensate for the effect of network latency in collaborative surgical simulations, a pseudo-physical approximation is proposed. This solution decreases the realism of the deformable calculation, but guarantees the stability of the simulation. In a similar manner, Dev and Heinrichs (2008) describe an experiment to determine the effect of network latency on touch perception of virtual organs using the *SPRING* framework. From the experiments it is concluded that subjects performing a virtual surgical task with a delay longer than 50 ms are not able to perceive different forces of different magnitudes. Additionally, it is reported that force feedback becomes unstable when there are latencies on the order of 100 ms. Hamza, Santhanam, Fidopiastis, and Rolland (2005) assessed how the shared-state consistency of a surgical augmented reality environment is affected by the variation of the network delay. They found that when the delay of the network is longer than 50 ms, the consistency of the shared state is considerably affected.

A review of CNVSS by Qin, Choi, Pang, Yi, and Heng (2010) shows the challenges which characterize these collaborative virtual environments (CVE) and a detailed explanation of the techniques used to address them. Finally, several collaborative surgical environments developed for different medical applications are described. The works described above have developed different collaborative and networked surgical simulators. However, as far as we know, there are no reports available on which factors affect collaboration in these kinds of virtual environments the most, and the major part of the works have focused on evaluating the effect of latency and jitter on force feedback perception and simulation stability of CNVSS.

Qin, Choi, Poon, and Heng (2009), Tang et al. (2007), and Qin, Choi, and Heng (2010) developed middleware to provide collaboration services to standalone surgical simulators. The middleware performance was tested by evaluating the effect of the number of users and collaboration strategy (coupling and token control) on the average frame rate of each user machine and the latency measured in the network. However, in the reported experiments, neither the impact of network parameters nor the impact of machine capabilities on collaboration performance in CNVSS was evaluated. It is also reported that the middleware guarantees consistency, but no quantitative evidence is offered to prove such a claim.

Other researchers evaluated which network conditions and machine capabilities affected collaboration the most in other types of collaborative environments. In Park and Kenyon (1999) and Allison, Zacher, Wang, and Shu (2004), factorial design of experiments is proposed to evaluate the effect of delay, jitter, and complexity of the task on human performance in CVE. It is concluded that all of these factors greatly impact collaboration. For example, the task completion time and the number of errors increase by approximately 40% for jitter values of 263 ms and for delay values of 200 ms. However, these experiments were not conducted for virtual surgical procedures.

Some researchers have evaluated collaboration in specific tasks, such as the handshake task, under different network conditions (jitter, delay, bandwidth, and percentage of packet loss; Dev, Harris, Gutierrez, Shah, & Senger, 2002; Gutierrez, Shah, & Harris, 2002). They report that delay, jitter, bandwidth and packet loss percentage larger than 20 ms, 1 ms, 128 kps, and 10%, respectively, are unacceptable for collaboration. Using the same task but only evaluating the effect of the delay on collaboration, Alhalabi, Horiguchi, and Kunifuji (2003) report that a delay longer than 600 ms deteriorates haptic perception. In addition, the task completion time increases more than 50% for delays over 1800 ms. Dev et al. (2002) and Gutierrez et al. (2002) report shorter delays compared to those reported by Alhalabi et al. (2003), because the first one considered the effect not only of the delay, but also the effect of the jitter, bandwidth, and packet loss. Thus, the combined effects of these factors have a major impact on collaboration.

Souayed, Gaiti, Yu, Dodds, and Marshall (2004) evaluated how the network factors affect the haptic interaction in distributed virtual environments. Through the use of a qualitative assessment of the haptic perception, they report that delay, jitter, and percentage of packet loss above 30 ms, 3 ms, and 10%, respectively, are unacceptable for effective collaborative interaction. Jay, Glencross, and Hubbold (2007) examined the impact of delayed haptic and visual feedback from the partner in a collaborative virtual environment with two operators. They found that both visual and haptic delay hinder task performance in terms of loss of contact with the target

Table 1. *Upper and Lower Limits of the Network Factors Evaluated in the Literature*

Reference	Jitter (ms)	Delay (ms)	Packet loss (%)
Park and Kenyon (1999)	12–163	10–200	NA[a]
Dev et al. (2002)	0–25	0–150	0.001–100
Alhalabi et al. (2003)	NA	0–2000	NA
Souayed et al. (2004)	1–15	0–50	0.1–50
Allison et al. (2004)	NA	0–200	NA
Hamza et al. (2005)	NA	0–50	NA
Jay et al. (2007)	NA	0–50	NA

[a] NA: Not analyzed.

object and acquisition time. However, haptic delay had a larger impact on performance than visual latency. In the aforementioned study of Jay et al., continuous haptic delay could be perceived to be starting from around 50 ms in a CVE.

Norman and Hamza-Lup (2010) present a survey of works which reviews how network conditions affect collaboration in CVE involving haptic perception. They conclude that the haptic channel is affected by small amounts of jitter, packet loss, and latency. Table 1 reviews the magnitude of the upper and lower network factors evaluated in the literature.

The research works mentioned describe how network conditions affect collaboration in CVE. However, as confirmed in Park and Kenyon (1999), Dev et al. (2002), and Gutierrez et al. (2002), network conditions affect collaboration depending on the evaluated application and collaboration task performed. Therefore, in order to determine the effect of these factors on collaboration in CNVSS, an experimental test involving surgical tasks and a surgical scenario including the simulation associated engine are required.

On the other hand, few research projects have considered the evaluation of the impact of machine factors in CVE. In Trefftz (2002), the impact of heterogeneity of user machines on the frame rate of a networked virtual environment is evaluated. They conclude that the inequity of the machines in a networked virtual environment (NVE) session makes the machines with lesser resources vulnerable to large amounts of information generated by high-end machines (Trefftz, Marsic, & Zyda, 2003). However, how the differences in user machine capabilities impact collaboration in CVEs involving rich simulation behavior (Marsh, Glencross, Pettifer, & Hubbold, 2006), such as in CNVSS, is not evaluated.

Some works evaluating the impact of network factors have used different machine capabilities in their experiment configurations (Hamza et al., 2005; Jay et al., 2007), but they do not conclude how these factors impact collaboration in the virtual environment (Table 2).

Yet, to the best of our knowledge, the following aspects have not been considered so far in the literature:

1. No one has evaluated how all network and machine factors affect performance in CNVSS.
2. No one has evaluated the effect of the interactions between factors in collaboration.
3. Research experiments performed up to now have not followed a systematic experimental design (i.e., they have evaluated each variable individually).

Considering that 2 and 3 are statistically invalid, we propose a fractional factorial experiment design (DOE) to determine which factors affect performance in collaborative virtual surgical environments the most.

3 Materials

3.1 CNVSS

SOFA is a new open source framework primarily targeted at medical simulation research. Based on advanced software architecture, it allows developers to create complex and evolving medical simulations by using a large set of algorithms and by simply editing an XML file (Allard et al., 2007). Additionally, SOFA introduces the concept of a multiresolution model that allows developers to use different resolution data structures for the deformation modeling, visual and haptic rendering,

Table 2. *Upper and Lower Factors of the Machine Capabilities Evaluated in the Literature*

Reference	Processor speed (GHz)	RAM capacity (MB)	Graphic card	Network card speed (Mbps)
Trefftz et al. (2003)	0.4–1.5	256–1024	T1[a]–T2[b]	NA[c]
Hamza et al. (2005)	1.5–2.8	512–1024	T3[d]–T4[e]	100
Jay et al. (2007)	2–3.2	512–1024	NA	NA

[a] Intense 3D, 16 MB.
[b] GeForce 2, 32 MB.
[c] Not analyzed.
[d] GeForce 4, Ti4200.
[e] GeForce 4, Ti4600.

and collision detection algorithms. SOFA, however, does not provide components in order to implement CNVSS because the framework lacks networking capabilities. For this reason, the functionality added to SOFA only considers the extension of collaborative functionality. This functionality is added first to the framework to determine (1) factors affecting collaboration in CNVSS and (2) which and how heterogeneity conditions affect collaboration in CNVSS.

The framework groups components in several packages. For example, the *Controller* package contains all of the components that permit interaction between surgical instruments and anatomical structures in the simulator using human–computer interface devices, such as haptic devices. The *Collision* package contains the pipeline and all the algorithms used to detect collisions among virtual models in the simulator.

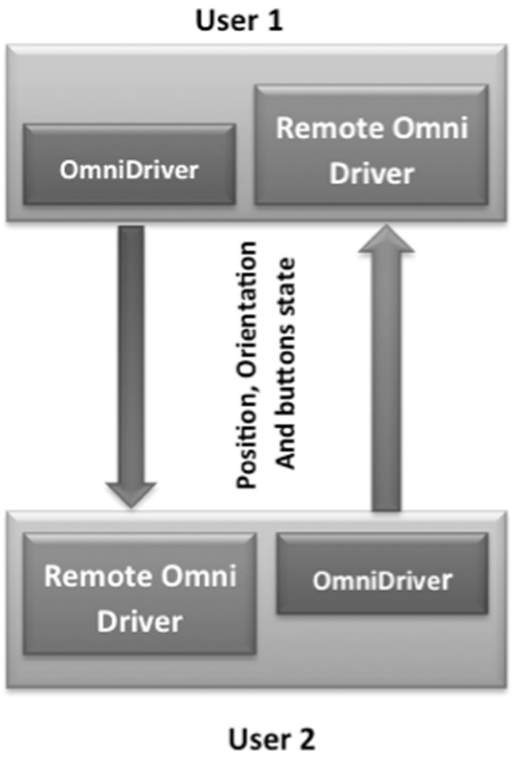

Figure 1. *Scheme describing how the* OmniDriver *and* RemoteOmni Driver *components are distributed when two users are collaborating.*

The *OmniDriver* component was modified in the *Controller* package. This component controls the movement of a surgical instrument in the simulation using the PHANToM Omni haptic device. The capability to send the state (position, orientation, and buttons state) of the haptic device to a remote simulation of the collaborative virtual surgical environment was added. Also in this package, the *RemoteOmniDriver* component, which is a local representation of the state of a remote haptic device, was added. For example, as Figure 1 shows, if two users, $U1$ and $U2$, are collaborating, the *OmniDriver* component located in the $U1$ machine sends the state of the local haptic device using UDP protocol. The *RemoteOmniDriver* component, located in the $U2$ machine, receives the state of the remote haptic device, and binds the state with a graphic representation of the surgical instrument. Neither the *OmniDriver* nor the *RemoteOmniDriver* implement mechanisms to handle network impairments, such as packet loss, jitter, or delay.

The *CarvingManager* component was modified in the *Collision* package to allow for cutting and attaching operations using a local or a remote haptic device. This is achieved by binding the *CarvingManager* component to *OmniDriver* or *RemoteOmniDriver* components. The user pressing the first and second buttons of the haptic device performs the cutting and attaching operations, respectively. When the user presses the cutting button, *CarvingManager* determines whether there is a collision between the graphic representation of the haptic device and an anatomical structure. If there is a collision, using the collision pipeline of the SOFA framework, the *CarvingManager* component determines the primitives colliding with the instrument and, using the topology changing functionality of the framework, executes the cutting operation. On the other hand, when the user presses the attaching button, *CarvingManager* again determines whether there is a collision between the graphic representation of the haptic device and an anatomical structure. If there is a collision, knowing the colliding primitives, the *CarvingManager* component creates an invisible spring between the primitives and the surgical instrument. The force exerted by this spring creates the effect of attaching the structures.

3.2 Surgical Scenario

In the surgical education field, a procedure that is frequently used as a first step in training is the cholecystectomy (Liu et al., 2003). The procedure provides several advantages for developing the skills of a trainee, such as:

- The surgeon needs basic anatomical and physiological knowledge of the anatomical structures that are involved in the procedure.
- The procedure allows for the manipulation of the organs and tissues using several instrument types. It demands the trainee to become familiar with the surgical instrument.
- The workspace in which the trainee moves the instruments is spacious when compared to those used in other procedures.
- The step sequence to carry out the procedure is comparatively simple.
- In the laparoscopic surgery field, this procedure is the most frequently executed and consistently improved upon.

In order to simulate a cholecystectomy, the 3D models of the gallbladder, liver, cystic conduct, and ligaments joining the gallbladder to the liver are used. The 3D models of these structures were created using the method described in Diaz, Trefftz, and Bernal (2009) and Diaz, Trefftz, Bernal, and Eliuk (2010). However, SOFA uses different data structures for visual and haptic rendering, collision detection, and deformable modeling. For visual and haptic rendering, models created by Diaz et al. (2009) are used. For collision detection, models created by Diaz et al. (2009) were decimated using the Blender open source application (Blender, 2012), to avoid affecting the real-time performance of the simulator. For the deformable modeling, SOFA needs a tetrahedral structure, and by implementing the *MeshTetraStuffing* component included in the SOFA framework, we create a tetrahedral data structure using a triangular mesh as an input. Figure 2 shows the various data structures used for the simulation of the gallbladder, and Table 3 summarizes the characteristics of the data structures used for each one of the anatomical structures involved in the surgical procedure.

3.3 Experimental Setup

An experimental test was developed in order to determine which factors affect collaboration in CNVSS. In the following sections we provide details about the experimental test.

3.3.1 Subjects. Sixteen subjects from 18 to 25 years old took part in the experiment. All subjects selected for the experiment were right-handed and had normal visual acuity. None of the subjects selected had previous surgical experience. Two teams were randomly assigned from the participants' pool in order to form

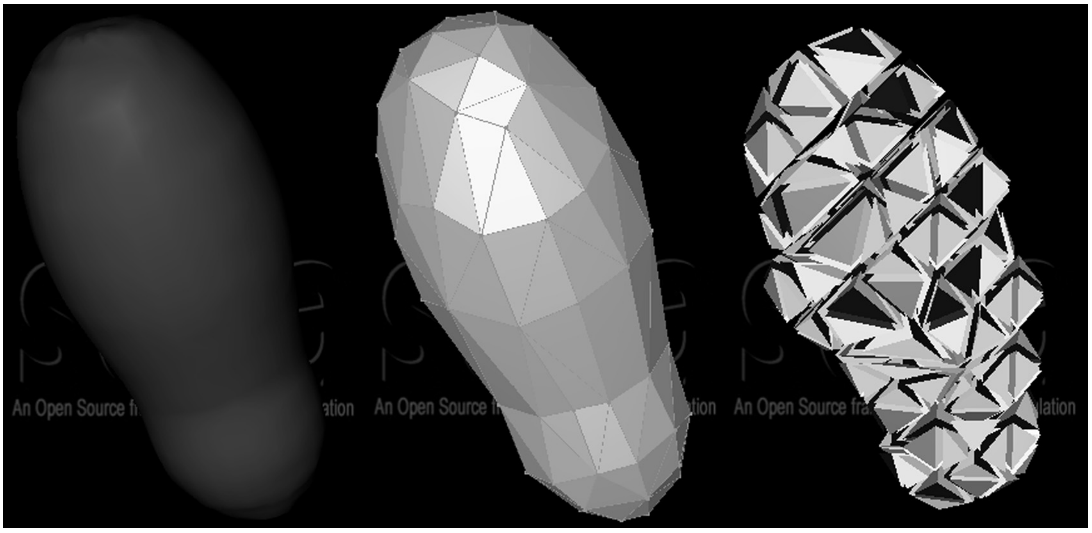

Figure 2. *Data structures used for the simulation of the gall bladder. Left: visual data structure. Center: collision data structure. Right: deformable modeling data structure.*

Table 3. *Characteristics of the Deformable, Visual, and Collision Data Structures Used in the Simulation*

Anatomical structure	Visual[a]	Collision detection[a]	Deformation[b]
Liver	4,384	2,054	1,382
Gall bladder	1,800	980	1,155
Cystic conduct	NA[c]	NA	279
Ligament	NA	NA	840
Surgical instruments	416	245	NA

[a] Number of polygons.

[b] Number of tetrahedrons.

[c] Not applicable. (Some anatomical structures use the same data structure for visual rendering, collision detection, and deformation computation).

Table 4. *Capabilities of the Machines Used in the Experimental Test*

Machine	Processor speed (GHz)	RAM capacity (Gb)	Graphic card	Network card speed (Mbps)
1	2.66–3.44	1–2	T1[a]	100–1000
2	2.66–3.44	1–2	T1	100–1000
3	2.8	1	T2[b]	1000

[a] Nvidia GeForce 8800 GTX.

[b] Nvidia Quadro FX 3000.

eight teams. All subjects were uninformed about the task and the purpose of the experiment.

3.3.2 System Configuration. Two workstations were configured to allow for collaboration between two persons in each experimental session. Each workstation consisted of a PHANToM Omni haptic device, which allowed users to interact with the surgical environment, a version of the CNVSS running locally in each machine, an XML file describing the surgical scenario, and 3D models used for the SOFA framework to load collision, visual, haptic and deformable modeling data structures. The capabilities of each machine are described in Table 4. The first and second machines' speed processor, RAM capacity, and network card speed were changed, but the same graphic card for each treatment

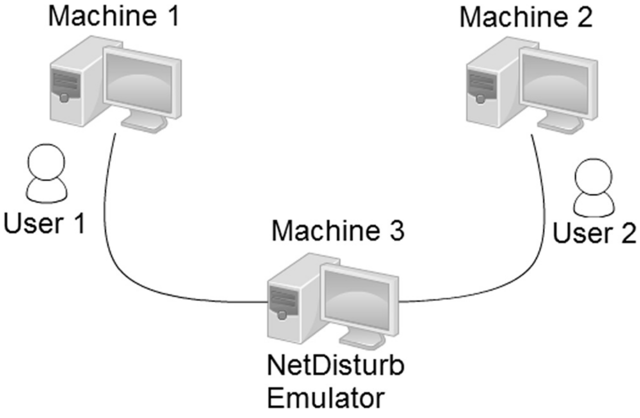

Figure 3. *System setup used for the experimental test. The users interact with the collaborative surgical simulation using Machine 1 and Machine 2. In Machine 3, the network impairments are simulated using NetDisturb.*

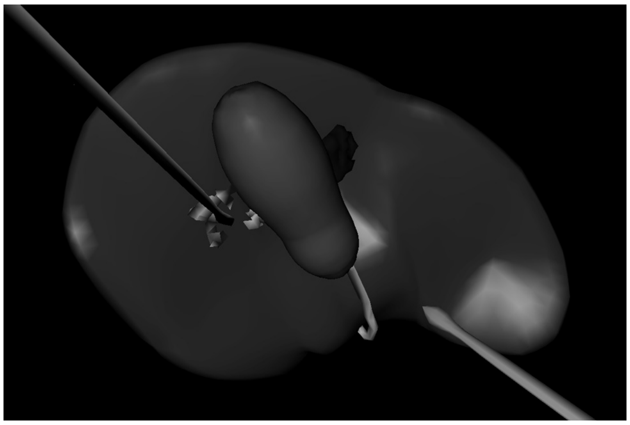

Figure 4. *Collaborative removal of the gallbladder. The light gray instrument is the representation of the local haptic device and the dark gray one is the representation of the remote haptic device.*

was used because its effect was not included in the analysis. Additionally, for each treatment of the experiment, the machines taking part in the collaborative session had the same capabilities.

For this experiment, the computation required in the surgical simulator (deformation, collision detection, and topological changes) was only performed in the CPU (serial processing) in order to simplify the experimental analysis. Including different kinds of processing strategies (serial processing in the CPU, parallel processing in the CPU, and parallel processing in the GPU) is beyond the scope of this paper. For example, if parallel processing strategies are included, it is required to not only include the processor speed of the CPU as a factor, but also the number of cores of the CPU.

The workstations were connected using a crossover cable. In order to add network impairments, a machine running the NetDisturb network software emulator was included in the cable path (NetDisturb, 2012), as shown in Figure 3. NetDisturb allows for control of network conditions such as: network delay, jitter, bandwidth, and packet loss. The machine capabilities are described in Table 4. The network conditions for each treatment of the experiment are the same from Machine 1 to Machine 2 and from Machine 2 to Machine 1. Additionally, direct voice communication was established using headset microphones and speakers.

4 Methods

4.0.1 Collaborative Surgical Task. The collaborative surgical task carried out by each team was a cholecystectomy (i.e., gallbladder removal). In order to achieve this, the surgeon has to cut the two ligaments connecting the gallbladder to the liver, and cut the cystic conduct while it is being stretched with the other instrument. Referring to the surgical virtual environment shown in Figure 4, the collaborative surgical task is executed by the users as follows:

1. Each user configures the point of view of the simulation, depending on the task executed by each one.
2. User 1 cuts the ligament on the right (blue on the screen) and User 2 cuts the ligament on the left (green on the screen) simultaneously.
3. User 1 cuts the conduct while User 2 attaches his or her instrument to the conduct and stretches it.
4. Finally, Users 1 and 2 remove the gallbladder using the surgical instruments.

4.0.2 Experimental Design. Fractional factorial DOE is applied when the number of factors is large and the effect of these factors, over a response variable, is to be determined. This DOE minimizes the

Table 5. *Factors Considered in the Experimental Design with Their Respective Lower and Higher Levels*

Factor	Lower level	Higher level
Delay (ms)	0	300
Packet loss (%)	0	70
Jitter (ms)	0	50
Bandwidth (Mbps)	100	1,000
Processor speed (GHz)	2.6	3.4
RAM memory (Gb)	1	2
Network card speed (Mbps)	100	1,000

number of experimental runs without losing important features of the problem studied. Therefore, a $2^{7\text{-}4}$ fractional factorial DOE was applied to determine which of the seven factors considered (network delay, jitter, bandwidth, packet loss, processor speed, RAM memory capacity, and network card speed) affect collaboration in CNVSS (Montgomery, 2008). Table 5 shows the levels considered for each factor.

Measuring the consistency of the shared state and response time of a CNVSS is a difficult task, and no one, so far, has proposed methods to quantify these variables in this kind of virtual environment. However, there are other variables which are easier to measure and which provide an indirect measurement of the consistency of the shared state and response time in CNVSS. These variables are the frames per second (FPS) of each machine and the task completion time (TCT) of the surgical task. For this reason, we considered these two variables as the response variables to be measured while the users executed the collaborative task. The R language, version 2.14.1 R-Project, was used for the design of the experiments and the data analysis (R-Project, 2012).

5 Results and Discussion

Figure 5 shows the normal plot, indicating which factors have a major effect over the response variable FPS. This figure shows that the most significant factors affecting the variable FPS are processor speed and RAM memory capacity.

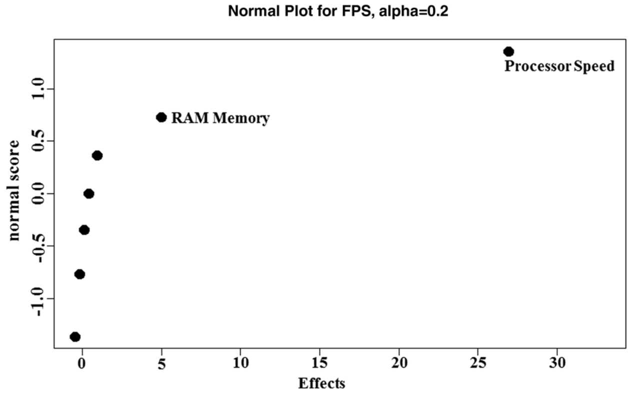

Figure 5. *Normal plot for the variable FPS (frames per second).*

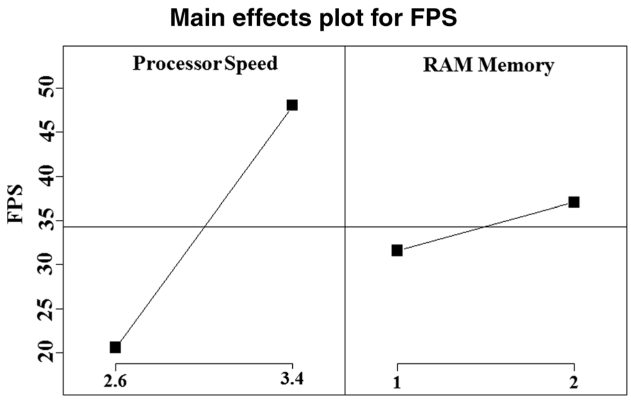

Figure 6. *Main effects plot for the variable FPS (frames per second).*

Figure 6 shows, in more detail, the effect of processor speed and RAM memory capacity factors over the response variable FPS. The highest increment achieved by the variable FPS is present when there is an increment in the processor speed. Similarly, but to a lesser magnitude, FPS increases incrementally as RAM memory capacity increases.

Figure 7 shows the normal plot, indicating which factors have a major effect on the variable TCT. The most important factors affecting the variable TCT are delay, jitter, packet loss percentage, and processor speed.

Figure 8 shows, in more detail, the effect of delay, jitter, packet loss percentage, and processor speed on the TCT variable. The delay factor most significantly

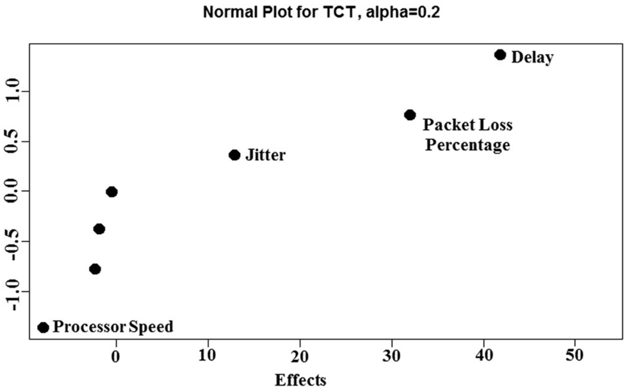

Figure 7. *Main effects plot for the variable TCT (task completion time).*

impacts TCT. When the delay value increases from 0 ms to 300 ms, TCT increases by approximately 40 s. The second largest effect is produced by the packet loss percentage factor. When the packet loss percentage increases from 0 to 70%, TCT increases by approximately 30 s. The third effect is the jitter factor. An increase of over 10 s is observed for TCT when the value of jitter increases from 0 to 50 ms. Finally, a decrease in TCT is observed in the presence of an increase in processor speed.

From the fractional factorial DOE analysis, it was found that the processor speed and RAM memory capacity have a larger effect over the response variable FPS; that is, the FPS of the simulation largely depends on local machine capabilities. Our CNVSS run locally on each user machine and the performance of the simulation depends on the local capabilities of the user machine. Similarly, the jitter, delay, percentage packet loss, and processor speed have a considerable effect on the response variable TCT and the largest effect on TCT is produced by the delay, which is consistent with the results in Dev et al. (2002), Alhalabi et al. (2003), and Jay et al. (2007). Thus, TCT is mainly affected by the network conditions and this is supported by the user comments. The users who execute the task with the higher levels of network conditions agree that Step 3 of the surgical task is quite difficult. This is because the execution of Step 3 of the surgical task requires tightly coupled collaboration between the users, and this modality of collaboration only takes place with a high-consistency shared state of the CVE (Shirmohammadi & Georganas, 2001). However, the higher level of network conditions evaluated and the peer-to-peer architecture of the developed system fail to guarantee the high-consistency shared state of the CVE without implementing mechanisms such as those proposed in Delaney, Ward, and Mcloone (2006a, 2006b).

6 Conclusions

From the analysis of the experiment it can be concluded that processor speed and RAM memory capacity have a larger effect on the response of variable FPS, whereas jitter, delay, percentage packet loss, and processor speed have a larger effect on the response of variable TCT. In brief, machine capabilities mainly affect the local performance of the simulation, and network conditions mainly affect the performance of the collaborative networked system. These results are important because by knowing which factors affect collaboration in CNVSS, it is possible to perform a second experimental test, called the surface response DOE, in order to determine, in more detail, the effect of the important factors on collaboration. The results of both experiments will be used to formulate a mathematical model able to maintain the collaboration of the user under heterogeneous network conditions and machine capabilities.

Finally, the SOFA framework was extended to support collaborative training of surgical tasks. The component-based architecture of the framework allowed for the easy extension of the framework, including components to handle remote surgical instruments, and the remote cutting and attaching of anatomical structures. However, peer-to-peer architecture used for the proposed CNVSS has encountered serious difficulties in maintaining the shared-state consistency, specifically when users perform tightly-coupled collaborative tasks and network conditions are not the best. For this reason, we will explore the implementation of a hybrid architecture in which the computation load will be distributed between a server and each client. The deformation computation will be calculated by the server machine in order to guarantee the consistency of the physical simulation, and the col-

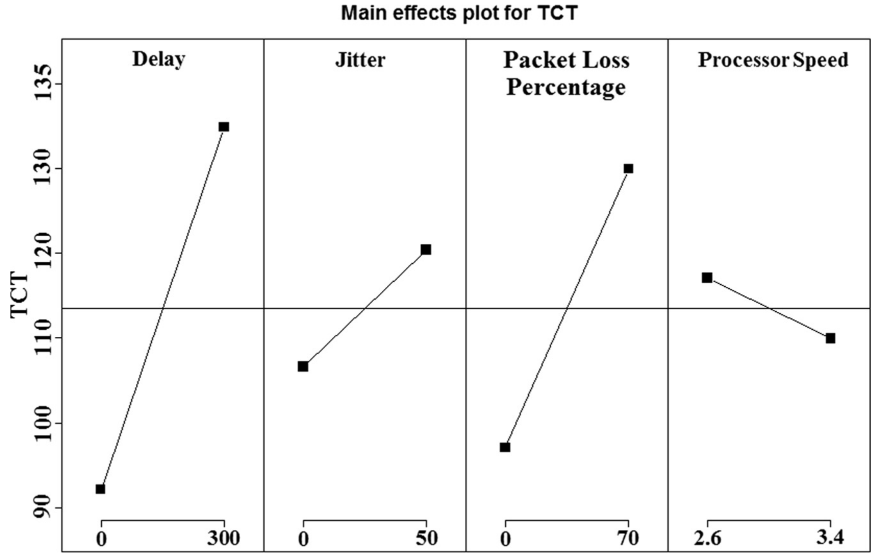

Figure 8. *Main effects plot for the variable TCT (task completion time).*

lision detection and haptic and visual rendering will be performed by each client.

Additionally, the performance of a computer running an application depends on its hardware components (network card, graphic card, processor, and RAM), and its particular architecture, or the way in which these components interact with each other. Thus, it is not suitable to define when a computer is better than another only by using a metric measuring the theoretical performance of each component and not using a metric measuring the real performance of the computer components as a whole. In order to avoid problems which arise when categorizing the users' machines' capabilities with different characteristics, future research will use a benchmarking algorithm to determine which computer is best suited for optimal performance.

References

Aggarwal, R., Grantcharov, T., Moorthy, K., Hance, J., & Darcy, A. (2006). A competency-based virtual reality training curriculum for the acquisition of laparoscopic psychomotor skill. *The American Journal of Surgery, 191*, 128–133.

Alhalabi, M. O., Horiguchi, S., & Kunifuji, S. (2003). An experimental study on the effects of network delay in cooperative shared haptic virtual environment. *Computers & Graphics, 27*(2), 205–213.

Allard, J., Cotin, S., Faure, F., Bensoussan, P.-J., Poyer, F., Duriez, C., … Grisoni, L. (2007). SOFA: An open source framework for medical simulation. *Medicine Meets Virtual Reality, MMVR '07*, 13–18.

Allison, R. S., Zacher, J. E., Wang, D., & Shu, J. (2004). Effects of network delay on a collaborative motor task with telehaptic and televisual feedback. *Proceedings of the 2004 ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry*, 375–381.

Blender. (2012). Open Source 3D Content Creation Suite. Retrieved from http://www.blender.org/

Delaney, D., Ward, T., & McLoone, S. (2006a). On consistency and network latency in distributed interactive applications: A survey—Part I. *Presence: Teleoperators and Virtual Environments, 15*(2), 218–234.

Delaney, D., Ward, T., & McLoone, S. (2006b). On consistency and network latency in distributed interactive applications: A survey—Part II. *Presence: Teleoperators and Virtual Environments, 15*(4), 465–482.

Dev, P., Harris, D., Gutierrez, D., Shah, A., & Senger, S. (2002). End-to-end performance measurement of internet based medical applications. *AMIA Fall Symposium 2002*, 205–209.

Dev, P., & Heinrichs, W. (2008). Learning medicine through collaboration and action: Collaborative, experiential, networked learning environments. *Virtual Reality, 12*, 215–234.

Diaz, C., Trefftz, H., & Bernal, J. (2009). Development of a laparoscopic surgical simulator to train a cholecystectomy procedure. *Proceedings of the IV Iberoamerican Symposium in Computer Graphics*, 121–126.

Diaz, C., Trefftz, H., Bernal, J., & Eliuk, S. (2010). General algorithms for laparoscopic surgical simulators. *Revista Ingeniería Biomédica*, 4(8), 57–70.

Gunn, C. (2007). *Using haptics in a networked immersive 3D environment.* Doctoral dissertation, University of Western Australia, Perth.

Gunn, C., Hutchins, M., & Adcock, M. (2005). Combating latency in haptic collaborative virtual environments. *Presence: Teleoperators and Virtual Environments*, *14*(3), 313–328.

Gutierrez, D., Shah, A., & Harris, D. A. (2002). Performance of remote anatomy and surgical training applications under varied network conditions. *Proceedings of ED–MEDIA World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 662–667.

Hamza, F. G., Santhanam, A., Fidopiastis, C., & Rolland, J. P. (2005). Distributed training system with high-resolution deformable virtual models. *Proceedings of the 43rd Annual Southeast Regional Conference (ACM–SE 43)*, 268–273.

Jay, C., Glencross, M., & Hubbold, R. (2007). Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment. *ACM Transactions on Computer–Human Interaction*, *14*(2), 8.

Lamata, P., Gomez, E., Bello, F., Kneebone, R., Aggarwal, R., & Lamata, F. Conceptual framework for laparoscopic VR simulators. *IEEE Computer Graphics and Applications*, *26*(6), 69–79.

Liberatore, V., Cavusoglu, M. C., & Cai, Q. GiPSiNet: An open source/open architecture network middleware for surgical simulations. *Medicine Meets Virtual Reality MMVR '03*, 316–321.

Liu, A., Tendick, F., Cleary, K., & Kaufmann, C. (2003). A survey of surgical simulation: Applications, technology, and education. *Presence: Teleoperators and Virtual Environments, 12*(6), 599–614.

Marsh, J., Glencross, M., Pettifer, S., & Hubbold, R. (2006). A network architecture supporting consistent rich behavior in collaborative interactive applications. *IEEE Transactions on Visualization and Computer Graphics, 12*, 405–416.

Montgomery, D. C. (2008). *Design and analysis of experiments.* New York: John Wiley & Sons.

Montgomery, K., Bruyns, C., Brown, J., Sorkin, S., Mazzella, F., Thonier, G., … Menon, A. (2002). Spring: A general framework for collaborative, real-time surgical simulation. *Proceedings of Medicine Meets Virtual Reality '02 (MMVR)*, 23–26.

NetDisturb. (2012). Impairment emulator software for IP networks. Retrived from http://www.zti-telecom.com/EN/NetDisturb.html

Norman, J., & Hamza-Lup, F. G. (2010). Challenges in the deployment of visuo-haptic virtual environments on the internet. *Proceedings of the 2010 Second International Conference on Computer and Network Technology, ICCNT '10*, 33–37.

Park, K. S., & Kenyon, R. V. (1999). Effects of network characteristics on human performance in a collaborative virtual environment. *Proceedings of the IEEE Conference on Virtual Reality, VR '99*, 104–109.

Pugh, C., Watson, A., Bell, R., Brasel, K., Purcell, G., Weber, S., & Kao, L. (2009). Surgical education in the internet era. *Journal of Surgical Research, 156*(2), 177–182.

Qin, J., Choi, K.-S., Poon, W.-S., & Heng, P.-A. (2009). A framework using cluster-based hybrid network architecture for collaborative virtual surgery. *Computer Methods and Programs in Biomedicine, 96*(3), 205–216.

Qin, J., Choi, K.-S., & Heng, P.-A. (2010). Collaborative simulation of soft-tissue deformation for virtual surgery applications. *Journal of Medical Systems, 34*, 367–378.

Qin, J., Choi, K.-S., Pang, W.-M., Yi, Z., & Heng, P.-A. (2010). Collaborative virtual surgery: Techniques, applications and challenges. *The International Journal of Virtual Reality, 9*, 1–7.

R-Project. (2012). For statistical computing. Retrieved from http://www.r-project.org/

Shirmohammadi, S., & Georganas, N. D. (2001). An end-to-end communication architecture for collaborative virtual environments. *Computer Networks, 35*(2–3), 351–367.

Souayed, R. T., Gaiti, D., Yu, W., Dodds, G., & Marshall, A. (2004). Experimental study of haptic interaction in distributed virtual environments. *Proceedings of EuroHaptics '04*, 260–266.

Székely, G., Brechbühler, C., Dual, J., Enzler, R., Hug, J., Hutter, R., … Bajka, M. (2000). Virtual reality-based simulation of endoscopic surgery. *Presence: Teleoperators and Virtual Environments, 9*(3), 310–333.

Tang, S.-W., Chong, K.-L., Qin, J., Chui, Y.-P., Ho, S.-M., & Heng, P.-A. (2007). ECiSS: A middleware based development framework for enhancing collaboration in surgical simulation. *IEEE International Conference on Integration Technology, ICIT '07*, 15–20.

Trefftz, H. (2002). *System-wide constraints and user preferences in collaborative virtual environments.* Doctoral dissertation, Rutgers University, New Brunswick, NJ.

Trefftz, H., Marsic, I., & Zyda, M. (2003). Handling heterogeneity in networked virtual environments. *Presence: Teleoperators and Virtual Environments, 12*(1), 37–51.

Woodrum, D. T., Andreatta, P. B., Yellamanchilli, R. K., Feryus, L., Gauger, P. G., & Minter, R. M. (2006). Construct validity of the LapSim laparoscopic surgical simulator. *The American Journal of Surgery, 191*(1), 28–32.