Christian Diaz
Helmuth Trefftz*
Grupo de Investigación
I+D+I en TIC
Universidad EAFIT

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) Implementing Hybrid Client–Server Architecture: Factors Affecting Collaborative Performance

Abstract

Currently, surgical skills teaching in medical schools and hospitals is changing, requiring the development of new tools to focus on (i) the importance of the mentor's role, (ii) teamwork skills training, and (iii) remote training support. Collaborative Networked Virtual Surgical Simulators (CNVSS) allow collaborative training of surgical procedures where remotely located users with different surgical roles can take part in the training session. To provide successful training involving good collaborative performance, CNVSS should guarantee synchronicity in time of the surgical scene viewed by each user and a quick response time which are affected by factors such as users' machine capabilities and network conditions. To the best of our knowledge, the impact of these factors on the performance of CNVSS implementing hybrid client–server architecture has not been evaluated. In this paper the development of a CNVSS implementing a hybrid client–server architecture and two statistical designs of experiments (DOE) is described by using (i) a fractional factorial DOE and (ii) a central composite DOE, to determine the most influential factors and how these factors affect the collaboration in a CNVSS. From the results obtained, it was concluded that packet loss, bandwidth, and delay have a larger effect on the consistency of the shared virtual environment, whereas bandwidth, server machine capabilities, and delay and interaction between factors bandwidth and packet loss have a larger effect on the time difference and number of errors of the collaborative task.

1 Introduction

Medical surgical training is changing worldwide, mainly due to: (i) a change in a training paradigm that highlights the mentor role and also the importance of teamwork besides basic surgical skills training before entering a surgical room, and (ii) the low number of expert surgeons located in distant regions or with time available to provide face-to-face surgical training (Gawande, 2011; Moller, Karamichalis, Chokshi, Kaafarani, & Santry, 2008;

Presence, *Vol. 23, No. 4, Fall 2014,* 393–409
doi:10.1162/PRES_a_00208

---

*Correspondence to htrefftz@eafit.edu.co.

Rombeau, Goldberg, & Loveland-Jones, 2010). Virtual reality-based tools, such as collaborative networked virtual surgical simulators (CNVSS), have been proposed to allow teamwork training and remote mentoring. These simulators create a shared virtual environment of the surgical scene that allows the collaborative training of users located remotely, with each member playing a role during the training session (Diaz, Trefftz, Quintero, Acosta, & Srivastava, 2013).

CNVSS must guarantee good collaboration among users to provide successful training. The quality of collaboration in a CNVSS can be measured by user-dependent or user-independent metrics, depending on whether users' skill level or time trained affect or do not affect the value of the measurements. For example, task completion time and number of errors are considered to be user-dependent measurements while consistency and response time are user-independent measurements.

In CNVSS, factors such as the user's machine capabilities and network conditions influence user-dependent and -independent measurements, affecting collaboration during the training session. However, to the best of our knowledge, the impact of these factors on CNVSS collaboration has not been evaluated. Moreover, knowing which factors have a major impact in CNVSS collaboration allows devising appropriate strategies and methods to mitigate the lack of consistency, provide a shorter response time, and decrease task completion time and number of errors caused by a deteriorated state of the system, not by users' skill levels. In this paper we describe the development of a CNVSS based on a hybrid client–server architecture and two statistical designs of experiments in order to determine which and how these factors affect the collaboration in a CNVSS.

The paper is structured as follows: the literature review section describes similar projects and opportunities of research. The materials section discusses the hardware, software, and methods required to implement a hybrid client–server architecture using the Simulation Open Framework Architecture (SOFA), and two statistical designs of experiments (DOEs) (fractional factorial and central composite) to determine the effect of factors on CNVSS collaboration. The last two sections discusss the results, conclusions, and future work.

2 Literature Review

Early works have evaluated how jitter, delay, and packet loss degrade the simulation and consistency in CNVSS (Liberatore, Cavusoglu, & Cai, 2006; Montgomery et al., 2002). Gunn, Hutchins, and Adcock (2005) and Gunn (2007) describe how the performance of a CNVSS is affected by jitter and network latency. They report that latency produces vibrations in the force effected by the haptic device and degenerates physical simulations of organs and tissues. However, they do not report at which latency and jitter values these issues arise. To compensate 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. Similarly, Dev and Heinrichs (2008) describe an experiment to determine the effect of network latency in perception by touch of virtual organs using the *SPRING* framework. From the experiments it was concluded that subjects performing a virtual surgical task, with a delay longer than 50 ms, are not able to perceive differently forces of different magnitude. Additionally, it was reported that force feedback becomes unstable when there are latencies of 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 network delay variation. They found that when the network delay is longer than 50 ms, consistency of the shared state is considerably affected.

A review of CNVSS by Qin, Choi, Pang, Yi, and Heng (2010) shows the challenges characteristic of 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 have developed different collaborative and networked surgical simulators. However, as far as we know, there are no reports available on which factors affect collaboration the most in this kind of virtual environment, and the major part of the works has focused on evaluating the effect of latency and jitter in the 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 evaluating the effect of the number of users and collaboration strategy (coupling and token control) over 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 machine capabilities on collaboration performance in CNVSS were evaluated. It is also reported that the middleware guarantees consistency, but no quantitative evidence is offered to prove this 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 was concluded that all of these factors have a major impact over the collaboration. For example, task completion time and 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 surgical applications.

Some researchers have evaluated the collaboration of a specific task, handshaking 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 a delay, jitter, bandwidth, and packet loss percentage longer than 20 ms, 1 ms, 128 Kbps, and larger than 10%, respectively, are unacceptable for collaboration. Using the same task but evaluating only the effect of the delay in collaboration, Alhalabi, Horiguchi, and Kunifuji (2003) report that a delay larger than 600 ms deteriorates the haptic perception and task completion time increased more than 50% for delays longer than 1800 ms. Dev et al. (2002) and Gutierrez et al. (2002) reported shorter delays compared to those reported by Alhalabi et al. (2003), because they considered the effect not only of the delay but also the effect of the jitter, bandwidth, and packet loss. This leads us to conclude that the combined effects of these factors have a major impact on collaboration.

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

Author	Jitter (ms)	Delay (ms)	Packet Loss (%)
Park and Kenyon (1999)	12-163	10-200	NA*
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

*NA: Not Analyzed

Souayed, Gaiti, Yu, Dodds, and Marshall (2004) evaluated how network factors affect haptic interaction in distributed virtual environments. Using a qualitative assessment of the haptic perception, they report that delay, jitter, and percentage of packet loss longer than 30 ms, 3 ms, and larger than 10%, respectively, are unacceptable for effective collaborative interaction. Jay, Glencross, and Hubbold (2007) examined the impact of delayed haptic and visual feedback 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 object and acquisition time. However, haptic delay had a larger impact on performance than visual latency.

Norman and Hamza-Lupp (2010) present a review of works studying how network conditions affect the collaboration in CVEs involving haptic perception. From this study 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.

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

Author	Processor Speed (GHz)	RAM Capacity (MB)	Graphic Card	Network Card Speed (Mbs)
Trefftz et al., 2003	0.4-1.5	256-1024	†T1-‡T2	*NA
Hamza et al., 2005	1.5-2.8	512-1024	•T3-**T4	100
Jay et al., 2007	2-3.2	512-1024	NA	NA

*NA: Not Analyzed
†T1: Intense3D,16 MB
‡T2: GeForce2,32 MB
•T3: GeForce 4Ti4200
**T4: GeForce 4Ti4600

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

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 machine capabilities differences in a Networked Virtual Environment (NVE) session makes the machines with lesser resources vulnerable to be flooded with large amounts of information generated by high-end machines (Trefftz, Marsic, & Zyda, 2003). However, how differences in user machine capabilities impact the collaboration in CVEs involving a rich simulation behavior (Marsh, Glencross, Pettifer, & Hubbold, 2006), such as a CNVSS, were not evaluated.

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

To summarize, to the best of our knowledge, the following issues have not been considered thus far in the literature:

1. No one has evaluated how all network and machine factors affect the performance in CNVSSs.
2. No one has evaluated interactions between factors in the collaboration.
3. Research experiments performed until now have not followed a statistical design of experiments (only one variable has been evaluated at a time).

To fill these voids in the literature we have performed two statistical DOEs to determine which factors affect the performance in collaborative virtual surgical environments the most.

3 Materials

3.1 CNVSS

SOFA, by Allard et al. (2007), is a new open source framework primarily targeted at medical simulation research. Based on an advanced software architecture, it allows developers to create complex medical simulations either by using a large set of algorithms and by simply editing an XML file, or by developing and adding new algorithms and components to the framework. SOFA, however, does not provide components in order to implement a CNVSS, because the framework lacks networking capabilities (Diaz et al., 2013).

In Diaz et al. (2013), the authors reported the extension of the framework to provide networking capabilities implementing a peer-to-peer architecture with replicated databases at each client, but, as they concluded, this architecture showed serious difficulties in maintaining the shared state consistency, specifically, when users perform tightly coupled collaborative tasks and network conditions are not the best. To address this problem, a hybrid client–server architecture, based on the one proposed by Lin, Narayan, and Lee (2010), as well as the components developed to apply this architecture in the SOFA framework, are described in the next section.

3.2 Hybrid Client–Server Architecture

The network architecture describes the functional relationship existing between network elements that compose a CNVSS. Client–server and peer-to-peer are the most commonly implemented network architectures, each one providing different advantages and disadvantages for the development of a CNVSS. In peer-to-peer architecture, the surgical simulation environment state is stored and computed locally by each client and user input events are transmitted to other clients in order to update the status of each client's simulation. This architecture is commonly implemented when the collaborative application requires a quick response time and tightly coupled collaborative tasks are not involved. Unfortunately, most of the tasks performed during a surgery by a team demand a tightly coupled collaboration and require highly consistent shared state. By contrast, in client–server architectures, the surgical simulation environment status is stored and computed by a host computer called the server and transmitted to each of the connected clients. A copy of the simulation state is stored at each client but only for viewing purposes. This architecture is able to guarantee a better shared state consistency of the surgical simulation environment, because the simulation is centrally computed by one computer. However, the server can become a bottleneck due to the high computing load as well as the volume of data that needs to be communicated through the server (Marsh et al., 2006). Additionally, updating the simulation status, when an event occurs on a client side, requires a round-trip to the server. It is worth mentioning that client–server and peer-to-peer refer to terms defined at the network level, and are equivalent to centralized and replicated databases, respectively, defined at the application level.

As a result of the work described in this paper, a hybrid client–server architecture was implemented. This implementation is loosely based on the one proposed by Lin et al. (2010) and is described in this section. This architecture allows the system to mantain the consistency of the collaborative virtual surgical environment, centralizing the computation of the surgical simulation on a server, and also preventing the server from becoming a bottleneck by distributing the computational load of collision, visual, and haptic rendering algorithms among each client.

In the proposed hybrid client–server architecture, one host, among those participating in the collaborative session, is selected to act as server and client at the same time, while the others act just as clients. In this context, the server role consists of computing the deformation of anatomical structures and the client role consists of running, locally, the visual rendering, collision detection, and haptic rendering algorithms. Figure 1 shows an example of the architecture where two users are collaborating. One of the hosts acts as client/server and the another one acts as a client. Considering the numbers appearing in this scheme and showing the execution sequence, the following steps are performed for each cycle of the simulation process, at the client–server machine:

1. Position and orientation data of the surgical instrument are sent from the module that reads the human–computer interface to the collision detection module.
2. In case of a collision between an anatomical structure and a surgical instrument, colliding primitives (triangles, points, or lines) are determined and sent to the surgical operation module.
3. After that, the surgical operation module determines what type of operation is enabled for the surgical instrument controlled by the user (i.e., cutting, carving, attaching, probing, or clip attaching).

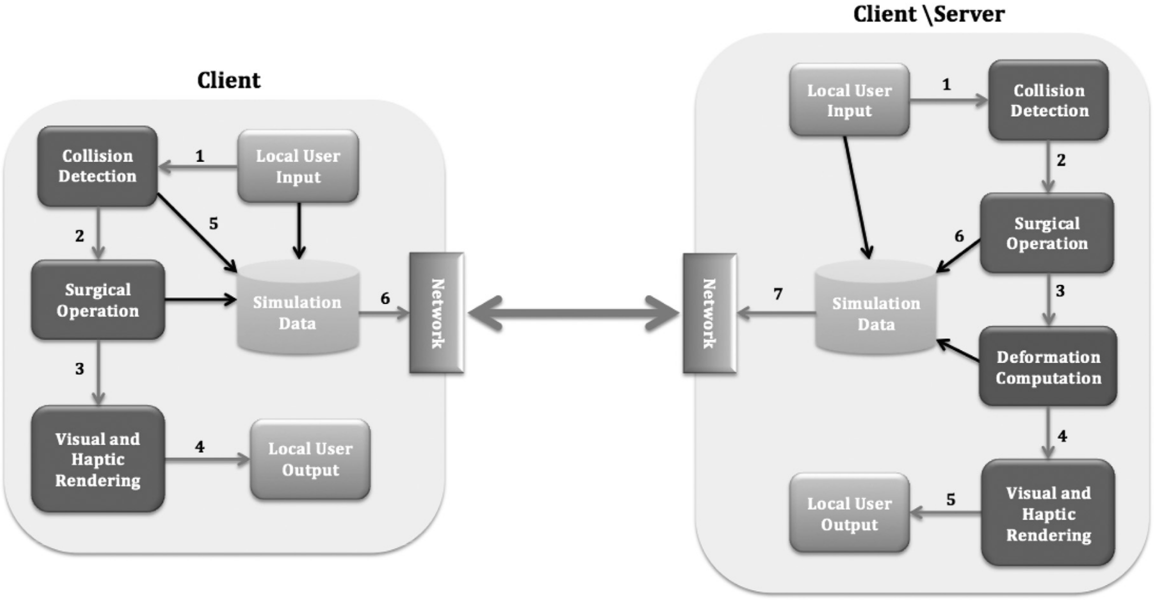


Figure 1. *Network elements and the functional relationships that compose the architecture implemented by our CNVSS.*

Based on that, it applies the appropriate algorithms for each operation and transmits the new state of the anatomical structure to the deformation computation module. In this step, the new state of the anatomical structure (i.e., topological changes or changes in the forces exerted on the model) is computed. This takes place, for instance, when a probing operation is performed.

4. Considering the new state of the anatomical structure, deformation is computed to determine the new displacements and positions of each of the points composing the updated anatomical structure.
5. Subsequently, haptic and visual rendering algorithms use the updated anatomical structure to compute the visual and force feedback provided to the user through the local user output module.
6. In this step, the simulation database is updated with the following information: (i) topological changes, (ii) position and orientation of the surgical instrument controlled by the user, and (iii) X, Y, and Z coordinates of each point composing the updated anatomical structure.
7. Finally, the state of the simulation database is translated into messages (see Table 3) and sent to the client machine. The data sent represents the changes that take place during the simulation.

A similar process is performed on the client machine side; it differs in the following aspects: (i) the deformation computation is not performed locally for the anatomical structures and (ii) the simulation database is updated with colliding primitives, topological changes, and position and orientation of the surgical instrument controlled by the user. The data are transmitted to the client–server machine. The proposed architecture does not implement any strategy to guarantee consistency based on time or information management techniques as described in Delaney, Ward, and McLoone (2006[a], 2006[b]), because the goal of the experiment described in this paper is to evaluate how the collaboration of users is affected using a CNVSS implementing the proposed network architecture. The data shown in the results section will make it possible, as future work, to evaluate which strategies for maintaining collaboration are appropriate under specific conditions, and propose an adaptive mechanism aimed at maximizing collaboration among users. Additionally, events occurring in the client and client–server machines are not synchronized, so that each machine has its own simulation and,

Table 3. *Description of the Messages Composing the Application Protocol of the CNVSS*

Message Type	Description
MSG ATTACHING, MSG CLIPATTACHING and MSG PROBING	Each one of these messages contains the primitives of the anatomical structure and the surgical tool intersecting each other. In order to simulate the phenomenon of attaching a clip or a surgical instrument, a set of virtual springs are used to connect the tool or the clip to the anatomical structure.
MSG CARVING	The algorithm implemented for carving operations uses a destruction strategy. The primitives that must be destroyed compose the message.
MSG CUTTING	The algorithm implemented for cutting operations uses a separation strategy. The primitives that must be divided compose the message.
MSG INSTRUMENT	It contains the position and orientation of the instrument controlled by the user.
MSG DEFORMATION	It contains the X, Y, and Z coordinates of the points composing the deformed anatomical structure.

possibly, a different data refresh rate. The data refresh rate of each machine defines the responsiveness provided by the system to each user considering the visual and haptic feedback.

Using the proposed architecture: (i) the computation of surgical simulation is centralized to avoid the divergence of the shared state and (ii) the load computation required by collision detection, as well as by the graphics and haptics rendering, is distributed among each user's machine. The components developed to implement the hybrid client–server architecture in the SOFA framework are detailed as follows:

- *TetrahedralCFEMServer* calculates the deformation of the anatomical structures associated using a tetrahedral co-rotational finite elements method (CFEM) and makes the deformation state available to be distributed by *CNVSS-Middleware* (MW) to the clients.
- *TetrahedralCFEMClient* receives the deformation state at the client side, which is computed by the *TetrahedralCFEMServer* component at the server side. When this component receives the deformation state, it communicates it to the local components which perform visual and haptic rendering and collision detection in order to update the corresponding data structures.
- *AttachingControllerClient*, *CarvingControllerClient*, and *AttachingClipControllerClient* determine whether a local surgical instrument is colliding with an anatomical structure and if it is, all the information related with the collision and the basic surgical operation performed (attaching, carving, clip attaching, among others) are stored to be sent to the server by the CNVSS-MW component.
- *AttachingControllerServer*, *CarvingControllerServer*, and *AttachingClipControllerServer* receive all the collision and basic surgical operation data sent by each client and apply them modifying the simulation state at the server side.
- *OmniDriver* and *RemoteOmniDriver* functionality is described in Diaz et al. (2013).
- *NetworkController* is a very important component in our architecture that runs at the client and server side as an independent thread. Its function is to read the state of the components described above and to determine whether there is an event to be transmitted by CNVSS-MW to the clients or to the server.
- *CNVSS-MW* is a middleware layer which provides three main networking capabilities to our CNVSS: (i) organizing the event data in messages that can be transmitted using a specific application level com-

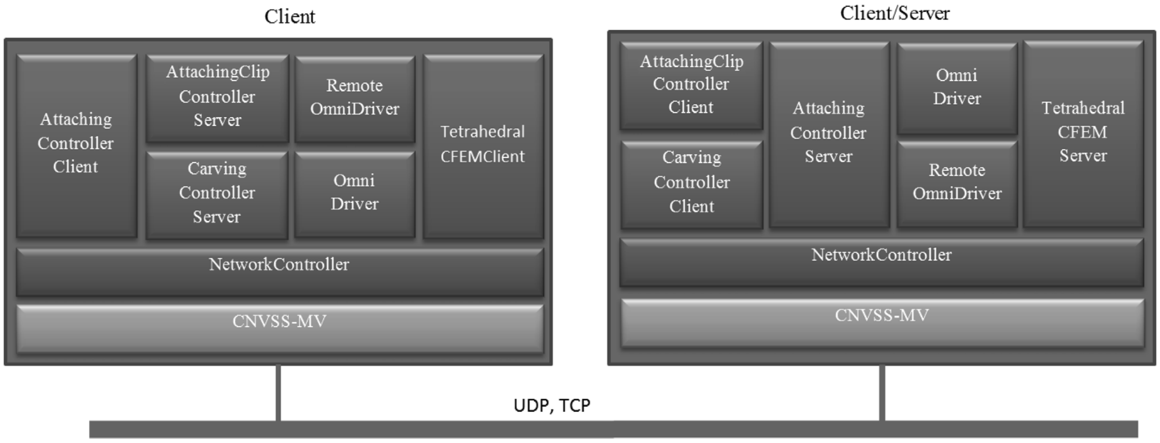


Figure 2. *Components developed to implement the hybrid client-server architecture.*

munication protocol developed for this purpose, (ii) defining, depending on the type of message, whether it needs to be sent using UDP (User Datagram Protocol) or TCP (Transmission Control Protocol), and (iii) controlling the connection state and managing the session data between the clients and the server.

Figure 2 shows the components developed in order to implement the hybrid client–server architecture using the SOFA framework. In this example, two users are collaboratively performing the surgical procedure, in which the client plays the role of attaching and the client–server plays the role of clip attaching and carving the anatomical structures.

3.3 Surgical Scenario

The surgical scenario developed to test which factors affect the proposed hybrid client–server architecture was the cholecystectomy. This procedure involves the removal of the gall bladder for treating symptomatic gallstones and is frequently used as first step of surgical training (Liu, Tendick, Cleary, & Kaufmann, 2003). In order to simulate a cholecystectomy, the tridimensional models of the gall bladder, liver, cystic duct, and artery and structures joining the gallbladder to the liver are required. The three-dimensional models of these structures were provided by the Clinical Anatomy Lab affiliated with Stanford University School of Medicine. Additionally, SOFA uses different data structures for visual and haptic rendering, collision detection, and deformable modeling. The process to generate these data structures is described in detail in Diaz et al. (2013). Figure 3 shows the surgical scenario created to simulate a cholecystectomy procedure and Table 4 summarizes the characteristics of the data structures used for each one of the anatomical structures involved in the surgical procedure.

4 Methods

4.1 Experimental Set-Up

Two experimental tests, a fractional factorial and a central composite DOE, were developed in order to determine which factors and how these factors affect the collaboration in CNVSS implementing the hybrid client–server architecture.

4.1.1 Subjects. Forty-six subjects took part in the two experiments, with ages ranging from 19 to 39. Ninety-five percent of the subjects selected for experiments were right-handed and the remaining 5% were left-handed. All the subjects selected had normal visual acuity and none of them had pre-existing surgical experience. Two teams were randomly assigned from the

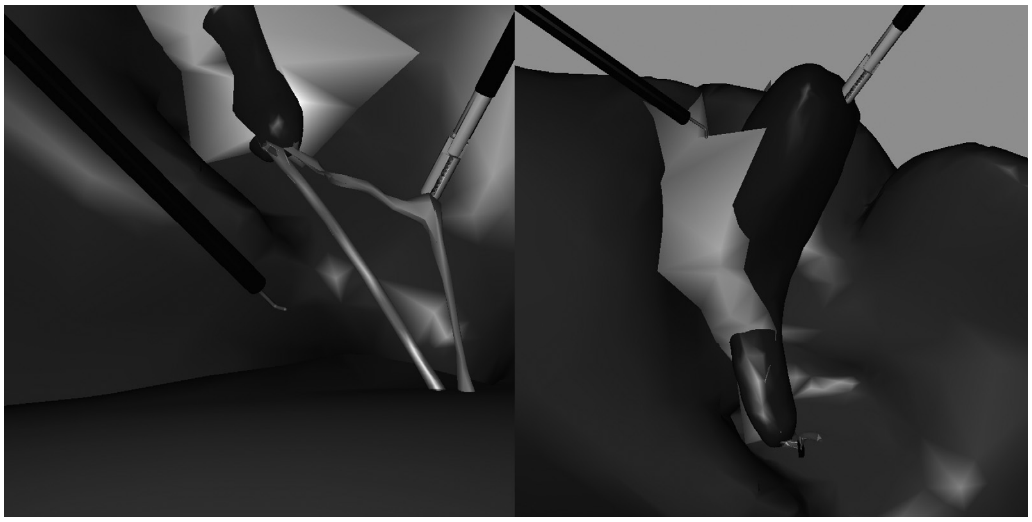


Figure 3. *Virtual reality cholecystectomy surgical scenario.*

Table 4. *Characteristics of the Deformable, Visual, and Collision Data Structures Used in the Simulation*

Anatomical Structure	Visual*	Collision Detection*	Deformation†
Liver	8006	2014	711
Gall bladder	2040	506	1011
Cystic duct	NA	NA	334
Cystic artery	NA	NA	309
Tissues joining the gall bladder to liver	NA	NA	351
Surgical instrument for attaching	3385	695	NA
Surgical instrument for carving	976	760	NA

NA: Not Applicable (some anatomical structures use the same data structure for visual rendering, collision detection, and deformation computation)

*Number of polygons

†Number of tetrahedrons

participant pool in order to form eight teams for the fractional factorial DOE and fifteen teams for the central composite DOE. All subjects were uninformed as to the task and the purpose of the experiment.

4.1.2 System Configuration. The system configuration described next was used for both experiments and is similar to the system configuration reported in Diaz et al. (2013). Two workstations were configured to allow collaboration between two persons in each experimental session (Machine 1 and Machine 2). Each workstation consisted of a PHANToM Omni haptic device which allows 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 tridimensional 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 5.

The workstations were connected using a crossover cable. In order to add network impairments, a third machine running the Netdisturb network software

Table 5. *Capabilities of the Machines Used in the Experimental Test*

Machine	Processor Speed (GHz)	RAM Capacity (GB)	Graphic Card	Network Card Speed (Mbs)
1	2.66	1	†T1	100
2	3.20	3	‡T2	1000
3	1.86	2	†T1	•1000

Graphic cards used by each machine. †T1: Nvidia GeForce 310 and ‡T2: Nvidia GeForce 8800 GTX. •The third machine used two network cards with equal characteristics as is recommended by NetDisturb.

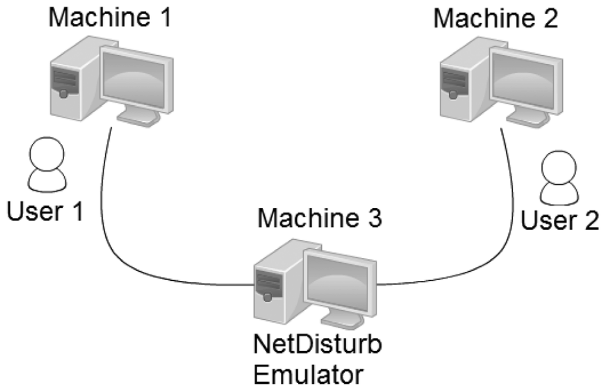


Figure 4. *System setup used for the experimental test. The users interact with the collaborative surgical simulation using Machine 1 and Machine 2. In Machine 3, the network impairments are simulated using NetDisturb.*

emulator NetDisturb (2013) was included in the cable path, as shown in Figure 4. NetDisturb allows control of network conditions such as network delay, jitter, bandwidth, and packet loss. However, NetDisturb allows only three parameters at a time to manipulate, so another application, developed by us, was used to control the jitter and NetDisturb was used to control the other parameters (delay, bandwidth, and packet loss). Additionally, a direct voice communication was established using Skype application and a headset, microphones, and speakers. The network impairments were applied only to the network data transmitted by the developed CNVSS.

4.1.3 Collaborative Surgical Task.

The collaborative surgical task carried out by each team was a cholecystectomy (i.e., gallbladder removal) (see Figure 5). During the collaborative task, two roles were performed by the users: User 1 attaches anatomical structures while User 2 cuts and carves the tissues and applies clips to the artery and cystic duct. The role played by each user was defined randomly. Steps involved to perform the collaborative surgical task are detailed as follows:

1. Each user configures the point of view of the simulation, depending on the task executed by each one.
2. Each user moves the surgical instrument to visualize it in the surgical area.
3. While User 1 attaches the artery and separates it from the cystic duct, User 2 applies clips to each one of the arteries and cystic duct extremes.
4. User 2 cuts the cystic duct while User 1 maintains the artery separated from cystic duct.
5. User 1 releases the artery, after which User 2 cuts the artery.
6. User 1 attaches the gall bladder and stretches it while User 2 carves tissues joining the gall bladder to the liver until the gall bladder becomes released.
7. User 1 removes the gall bladder.

Several factors can affect the collaborative performance of the users performing a virtual surgical task, such as systems conditions (network and machine capabilities), user skill level (individual and as a team), and type of collaborative task, among others. Since the objective of the experiment was to determine how system conditions affect the collaboration, the users performed individual and team training sessions, in order to become proficient in the collaborative task and evaluate only the impact of system conditions. So, each team carried out two training sessions, the first one with the purpose of learning how to handle the human–computer interface (PHANToM Omni) and how to perform the role played, carving or attaching, and the second one with the purpose of training the collaborative work of the virtual surgical procedure as a team. Each training session stopped when each user and team performed up

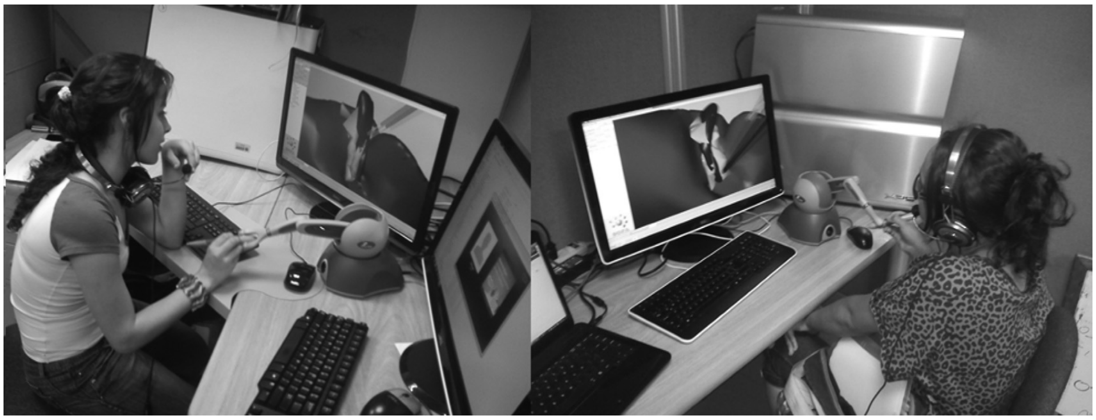


Figure 5. *Collaborative removal of the gall bladder. Right user handles the attaching instrument and Left one handles the carving and clip attaching instrument.*

to seven training sessions and the learning curve became almost flat. The training sessions used an ideal system configuration. Before users started the training sessions, a review and discussion, using a video, of the seven steps composing the collaborative cholecystectomy surgical procedure were performed.

4.1.4 Experimental Design. The purpose of this experiment is to determine which and how a group of factors affect the collaboration in a CNVSS implementing the hybrid client–server architecture proposed. In order to know which factors affect the collaboration, a fractional factorial DOE with eight runs was performed to establish which of the factors (delay, jitter, packet loss, bandwidth, server, and client benchmark) are the most influential on outputs (consistency, time difference, and number of errors) by using a Daniel plot (Daniel, 1959). The parameters whose distribution cannot be considered as normal standard are statistically relevant in the fractional DOE. Therefore, they are considered to affect the collaboration in a CNVSS. In order to determine how the relevant factors affect the collaboration, a surface response central composite DOE with 15 runs was performed afterward with delay, packet loss, and bandwidth as studied factors and using the same output variables. Input and output variables are described in detail in Tables 6 and 7, respectively.

In order to measure the difference in consistency between the client and server, the clock of each machine was synchronized using the Network Time Protocol (NTP). After that and while the users performed the collaborative surgical task, every 100 ms, the tridimensional positions of the points composing the deformable data structure of the gall bladder were stored in a file with a time label. Then, using the two files (stored in the client and server machines) the average Euclidean distance between points was calculated and considered as the measure of inconsistency. Data from the DOE were analyzed with the software for statistical computing R with Fractional Factorial Designs with 2-level factors -FrF2-, Wrapper for Design of Experiments Functionality -DoE.wrapper-, and Response Surface Method -rsm- add-on packages (Lenth, 2009).

5 Results and Discussion

5.1 Fractional Factorial DOE

5.1.1 Inconsistency. Daniel's plot for inconsistency indicates that the most influential factors are *packet loss*, *bandwidth* and *latency* which deviate the most from the normal distribution curve (see Figure 6a). Studying in more detail the effect of the factors, Figure 7a shows that *inconsistency* increases 0.8 cm when delay is increased from 0 ms to 300 ms. This result was expected, since the delay causes the state of the simulation in the server and the client to differ, in a given time,

Table 6. *Input Variables Considered in the Experimental Design*

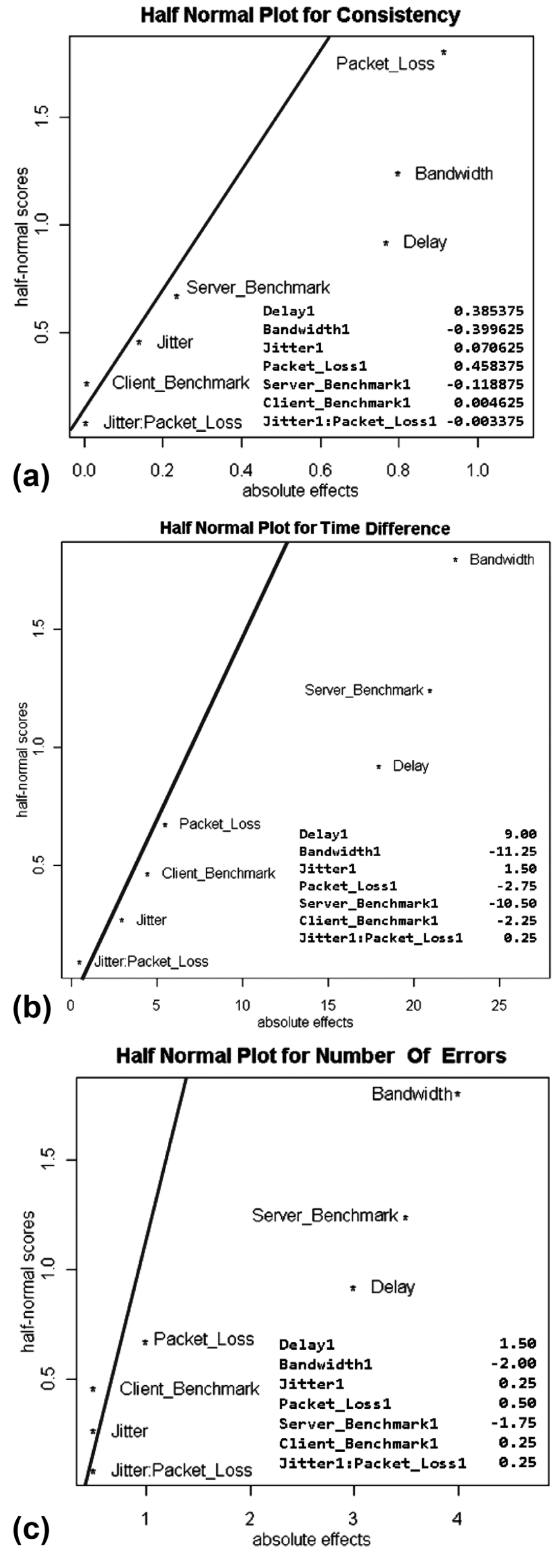
Input Variables Factor	Description	Lower Level	Higher Level
Delay (ms)	The time it takes for a packet to get from one end (Machine 1) to the other (Machine 2).	0	300
Packet Loss Percentage (%)	Percentage of how many data packets are lost during transmission due to congestion, link failures, or other problems.	0	50
Jitter (ms)	The statistical variance of the delay.	0	50
Bandwidth Available	Bandwidth is defined as the amount of data (megabits) that can be transmitted in a fixed amount of time (one second) between Machine 1 and Machine 2.		
Percentage (%)	So, this input variable is the percentage of amount of data that can be transmitted, considering that the bandwidth required by the application is 100%.	70	100
Server and Client Benchmark (ms)	Time taken by the client or server machine to execute one step of the cholecystectomy simulation.	59.47	34.53

Table 7. *Output Variables Considered in the Experimental Design*

Output variables Response variable	Description
Inconsistency (cm)	Average Euclidean distance between points of the deformable data structures storage in the server machine and that storage in the client machines. When inconsistency is 0, it means a fully consistent CNVSS.
Difference in Task Completion Time (s)	Difference in time between the time taken by a team to perform the collaborative task in the conditions defined by the experimental run and those taken in ideal conditions.†
Number of Errors	Number of mistakes made by a user. In our simulation the user makes a mistake in two cases: (i) when a user touches a prohibited anatomical structure and (ii) when User 2 incorrectly attaches a clip.

†Ideal conditions are when the benchmark of the user machines are the best and no network impairment is applied.

based on the value of the delay. Similarly, it is noted that, when available bandwidth is reduced from 100%, the minimum bandwidth required by our simulation, to 70%, there is an increase in the inconsistency of more than 0.8 cm. In this case the effect of the bandwidth reduction is similar to performing a queuing of the deformation data. This queuing causes the deformation data to be received at the client side after a time interval, increasing inconsistency. The influence of client benchmark is not noticeable, probably because the amount of computation performed by the client is much less than the server, since the client does not compute the deformation. This causes the client frames per second (fps) to be equal or not much less than the server fps, although



```
Delay1                 0.385375
Bandwidth1            -0.399625
Jitter1                0.070625
Packet_Loss1           0.458375
Server_Benchmark1     -0.118875
Client_Benchmark1      0.004625
Jitter1:Packet_Loss1  -0.003375
```

```
Delay1                   9.00
Bandwidth1             -11.25
Jitter1                  1.50
Packet_Loss1            -2.75
Server_Benchmark1      -10.50
Client_Benchmark1       -2.25
Jitter1:Packet_Loss1     0.25
```

```
Delay1                  1.50
Bandwidth1             -2.00
Jitter1                 0.25
Packet_Loss1            0.50
Server_Benchmark1      -1.75
Client_Benchmark1       0.25
Jitter1:Packet_Loss1    0.25
```

Figure 6. *Daniel's plots for determining the significance of factors.*

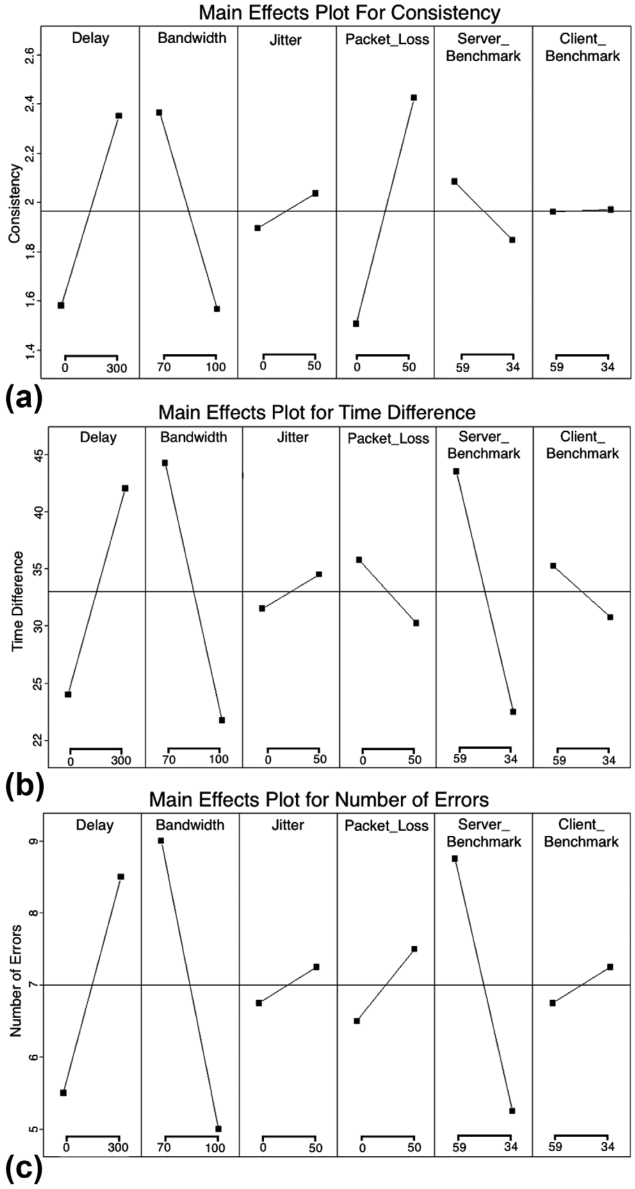


Figure 7. *Main effects plots for inconsistency, number of errors, and time difference response variables.*

the client benchmark is about half the server benchmark. For this reason the client is never going to be flooded by the simulation data it receives from the server, affecting performance on the client side. Finally, when server benchmark is increased, there is an increase in the inconsistency of more than 0.2 cm. A large server benchmark value causes the simulation on both server and client to slow; this does not affect the inconsistency significantly because, even though the simulation does not run above an appropriate refresh rate, 30–60 fps, at the client and server side, the state of the simulation at both sides is the same. Finally, packet loss, the most influential factor for this output variable, increases more than 0.9 cm when the percentage of packet loss is increased from 0% to 50%.

5.1.2 Time Difference and Number of Errors.

Figures 6(b) and 6(c) show the normal plots, indicating what factors have a major effect over the response variables *time difference* and *number of errors*, respectively. These figures show that the factors affecting both variables the most are bandwidth, server benchmark, and delay. Figures 7(b) and 7(c) show in more detail the influence of studied factors over these response variables. From these figures it can be observed that when delay is increased from 0 ms to 300 ms, time difference and number of errors increased more than 15 seconds and 3 errors, respectively. This is because an increased delay increases the difference between the shared virtual environment simulated on the server and the one displayed on the client, causing an asynchrony between the actions performed by users taking part of the collaborative task. For this reason, steps 3, 4, and 6 become the most difficult of the simulated collaborative surgical task. During these task steps the users usually and accidentally collide their instruments with other prohibited structures, increasing the number of errors. A similar effect is observed when bandwidth is decreased. As mentioned, a reduction in the bandwidth produces a data queuing, which causes data packets from the server to arrive at the client side with a higher delay, increasing the difficulty of performing tightly coupled collaborative tasks. Moreover, when server benchmark is increased, time difference and number of errors increases 20 seconds and nearly 3 errors, respectively. This occurs because the server benchmark determines the smoothness of deformation computation displayed on both the client and server machines. For this reason, if the server benchmark is considerably high, the deformation computation is executed slowly, forcing users to perform their movements slowly, because they try to synchronize with the simulation speed. All this increases the difference time and the number of errors, even though in this case there is no differences between the simulation displayed on the client and server.

5.2 Central Composite DOE

Figures 8(a), 8(b), and 8(c) show the tridimensional surface of each response variable as a function of two of the factors, taken simultaneously, while keeping the remaining one constant. From Figures 8(b) and 8(c) it can be observed that there is an interaction between bandwidth and packet loss. Time difference and the number of errors are decreased, when percentage packet loss increases and there is a reduction in the bandwidth. This may be contrary to what would be expected, that an increase in the percentage of packet loss, when bandwidth is reduced, deteriorates further collaboration. However, what actually happens is that the packet loss causes the bandwidth required by the application to decrease, and thus the effect of the bandwidth reduction is diminished. Additionally, as can be seen in Figures 6(b) and 6(c), the percentage of packet loss by itself does not have a significant effect on the users' performance.

Similarly, when percentage packet loss is increased and bandwidth is reduced, inconsistency is decreased but in smaller amount. Two causes explain this phenomenon: (i) the method used in NetDisturb to simulate packet loss guarantees that only one packet is lost at a time and (ii) the movements performed by users during the collaborative task are slow, causing small deformations of the anatomical structures. For these reasons, collaboration is maintained because the geometric shape of the anatomical structures is preserved even though some deformation packets are lost and points of the mesh at the client side are outdated. The phenomenon caused by the interaction between packet loss and bandwidth can be used as a strategy to improve collaboration when the available bandwidth is not enough. For example, the CNVSS can vary the send frequency of the data or the resolution of the deformable meshes transmitted to improve collaboration.

Additionally, when the available bandwidth was less than the required bandwidth and some time of the col-

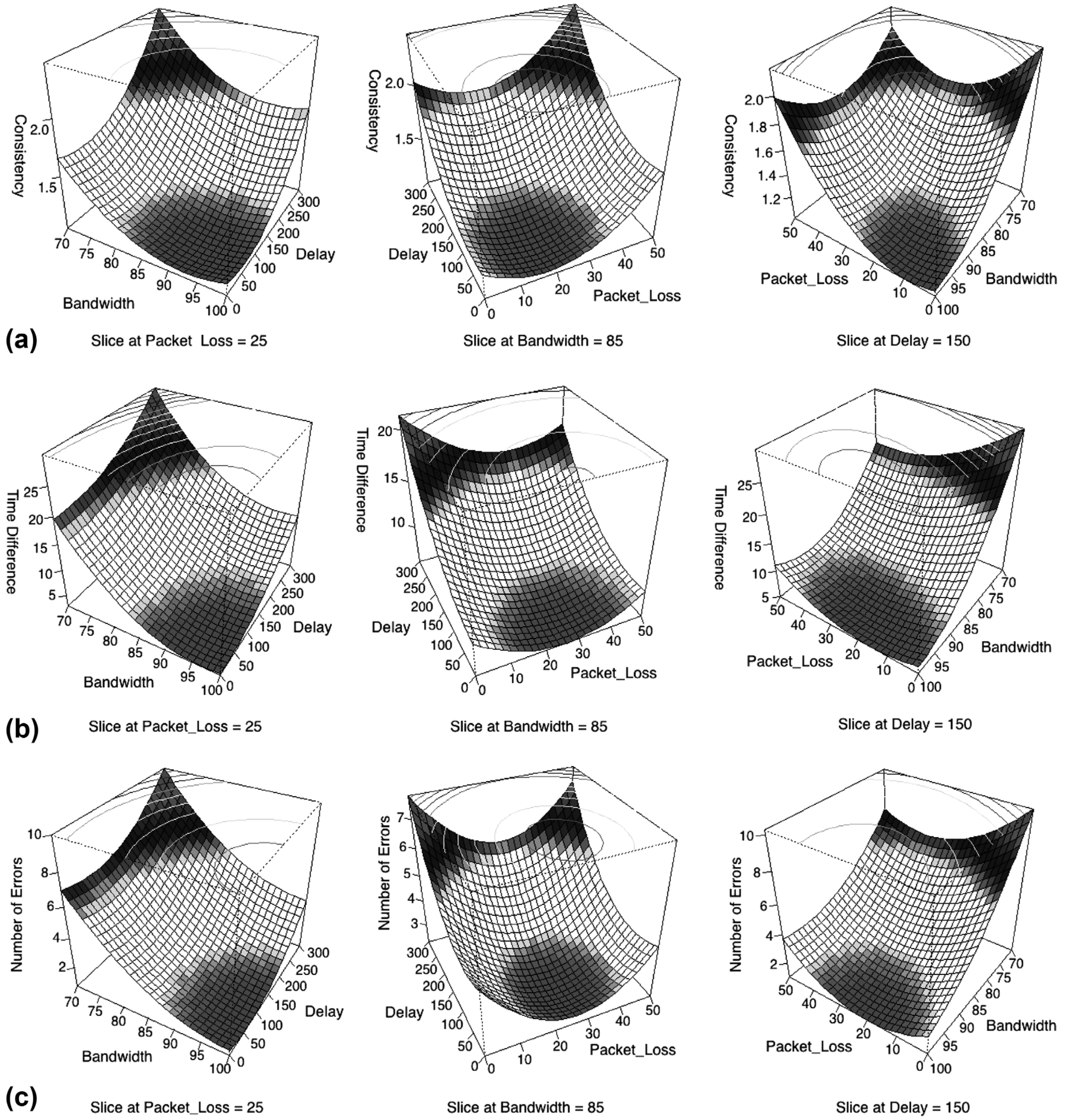


Figure 8. *Response Surface Plots for central composite DOE.*

laborative task has elapsed, users were able to adapt their performance to the unsynchronized surgical environment using communication by voice, because audio communication data were not exposed to the network impairments during the experiments. For this reason, the user working on the server machine was able to guide by voice instructions the movements of the other user, diminishing the potential impact on collaboration influenced by bandwidth factor; however, this kind of collaboration is not useful for surgical training.

6 Conclusions

From the analysis of the experiments it can be concluded that packet loss, bandwidth and delay have a larger effect on the response of variable inconsistency, whereas bandwidth, server benchmark, delay, and interaction between factors bandwidth and packet loss have a larger effect on the response of variables difference time and number of errors.

These results are important because by knowing which and how factors affect the collaboration in CNVSS implementing a hybrid client–server architecture, it is possible to develop an inference machine able to maintain the collaboration of the users under different network conditions and machine capabilities. Specifically, the results of fractional factorial and central composite DOE play two major roles in the formulation of the inference machine. The first one defines which factors must be expressed in the inference machine as input variables and the second one defines the range of values in which these factors minimize or maximize the performance of the collaboration. This range of values will be used to define the characteristics of the training set used by the inference machine, in order to choose the set of parameters that increase the collaborative performance.

Finally, the SOFA framework was extended to support collaborative training of surgical tasks implementing a hybrid client–server architecture which centralizes the computation of surgical simulation to avoid the divergence of the shared state and distributes the load computation, required by collision detection and graphic and haptic rendering, between each user machine.

References

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. *Studies in Health Technology and Informatics, 125*, 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* (pp. 375–381). Singapore: ACM.

Daniel, C. (1959). Use of half-normal plots in interpreting factorial two level experiments. *Technometrics, 1*(4), 311–341.

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* (pp. 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., Quintero, L., Acosta, D., & Srivastava, S. (2013). Collaborative networked virtual surgical simulators (CNVSS): Factors affecting collaborative performance. *Presence: Teleoperators and Virtual Environments, 22*(1), 1–29.

Gawande, A. (2011). Personal best: Top athletes and singers have coaches. Should you? *The New Yorker–Annals of Medicine*, 1–9.

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

Gunn, C., Hutchins, M., & Adcock, M. (2005). Combating latency in haptic collaborative virtual environments. *Presence: Teleoperators and Virtual Environments, 14*, 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* (pp. 662–667).

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

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).

Lenth, R. (2009). Response–surface methods in r, using rsm. *Journal of Statistics Software, 32*(7), 1–17.

Liberatore, V., Cavusoglu, M. C., & Cai, Q. (2006). GiPSiNet: An open source/open architecture network middleware for surgical simulations. *Studies in Health Technology and Informatics* (Vol. 119, pp. 316–321).

Lin, S., Narayan, R. J., & Lee, Y.-S. (2010). Hybrid client-server architecture and control techniques for collaborative product development using haptic interfaces. *Computers in Industry, 61*(1), 83–96.

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.

Moller, M., Karamichalis, J., Chokshi, N., Kaafarani, H., & Santry, H. P. (2008). Mentoring the modern surgeon. *Bulletin of American College of Surgeons, 93*(7), 19–25.

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. *Medicine Meets Virtual Reality '02 (MMVR)* (pp. 23–26). IOS Press.

NetDisturb. (2013). Impairment Emulator Software for IP Networks. Retrieved 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* (pp. 33–37). Washington, DC: IEEE Computer Society.

Park, K. S., & Kenyon, R. V. (1999). Effects of network characteristics on human performance in a collaborative virtual environment. *Proceedings of the IEEE Virtual Reality* (pp. 104–109). Washington, DC: IEEE Computer Society.

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.

Rombeau, J. L., Goldberg, A., & Loveland-Jones, C. (2010). *Surgical mentoring: Building tomorrow's leaders.* New York: Springer Verlag.

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* (pp. 260–266).

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, 2007, ICIT* (pp. 15–20).

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

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