A Collaborative and Immersive VR Simulator for Education and Assessment of Surgical Teams

Paulo Vinícius de Farias Paiva*, Liliane dos Santos Machado§, Thiago Vinícius Vieira Batista‡

*§‡LabTEVE / Department of Statistics
Federal University of Paraíba, João Pessoa/PB, Brasil
e-mails: {*paulo.fariaspaiva, ‡thiagovvb}@gmail.com, §liliane.machado@pq.cnpq.br

***Abstract*— Traditionally, the evaluation of surgical procedures in VR simulators have been restricted to their individual technical aspects, and disregarded the procedures carried out by teams. However, some decision models have been proposed in order to be incorporated and support the collaborative training evaluation process of surgical teams in Collaborative Virtual Environments (CVEs). This paper aims to discuss some possibilities and advantages of using the CVEs for the training process and assessment of surgical teams as well as presenting the steps of planning and development of a new simulator with these features, called SimCEC. The discussion is held in a computational modeling and implementation optic.**

***Keywords*— surgical simulation, collaborative virtual environments, user's assessment, SimCEC, VR.**

I. Introduction

In recent years, Virtual Environments (VEs) are increasingly being developed and used as technological support for education and medical training [8, 9]. However, training and assessment aspects are often devoted for just single user, while important aspects of collaborative procedures executed by surgical teams remain disregarded [13]. A number of studies have shown that educational in the medical field presents some deficiencies in the sense of being developed and evaluated not just individual technical competencies, but also interpersonal skills (e.g. cooperation, leadership considered of great importance in the procedures performed by teams [1, 2, 3]. According to Aggarwal [1], health students work in groups but are rarely trained together so that these skills can be developed, and also traditional teaching strategies are based on training in the operating room are dependent on hospital volume of surgical procedures.

One solution offered by RV is the use of VEs in a computing distributed architecture which is denominated as Collaborative Virtual Environments (CVs). The CVEs are systems in which multiple users can interact with one another in real-time in order to perform a given task, in a collaborative manner. Medicine has largely benefited from CVEs, which are specially useful for surgical training applications. CVEs enable the practice of medical procedures by users who can share experiences with a remote tutor or other students [5]. Several CVEs have been developed in order to provide support to surgical team training [7, 12, 14]. Nevertheless, few emphasize aspects of teams evaluation.

According to Machado and Moraes [13], VR simulators are frequently required for educational purposes, but they are still developed with little attention to the medical curriculum. As a natural consequence of the lack of attention to these important requirements, many simulators do not include the possibility of providing an user's assessment tool. In this sense, user's assessment feature becomes essential for CVEs that are focused on surgical education and training [11]. It's important to emphasize that unless a training simulator provides automatic evaluation mechanisms, it will not match the needs of students [13]. In order to achieve this goal, some efforts have been made during the last decades inside VR area, and some user's Assessment Systems (ASs) can be found in literature [3, 10, 9, 16]. Usually, these intelligent systems classifies users' knowledge by using their interaction data (e.g. position, speeds, applied forces) and diverse I.A techniques. Once ASs are computer-based reasoning systems, more accurate evaluation criteria can be provided, as well as immediate performance feedback.

This work aims to present important issues related to the design and implementation of a CVE that supports collaborative evaluation of a group of students in basic surgical procedures, named SimCEC. The SimCEC simulator has a assessment module which enables automatic feedback of individual and collective performances of all connected users. The paper also discuss important issues in respect to networking performance, by proposing a simple model for adapting latency, assessment data security and VE replication.

II. Related Works

Initially, to perform the planning of SimCEC and its collaborative evaluation architecture, a survey was conducted to find similar applications in the academic and industry fields. As inclusion criteria to our study, we seek VR simulators that addressed user's collaboration and/or assessment features. In the specific case of assessment systems, there was found simulators which could classify one user and/or multiple users at the same time. For this work, we adopt Machado and Moraes [13] assessment system's classification: Individual Assessment System Systems (IAS) and Collaborative Assessment System

978-1-4673-7204-6/15 $31.00 © 2015 IEEE
DOI 10.1109/SVR.2015.33

(CAS). In this sense, five simulators were studied and its main features was analyzed: (1) 3DiTeams simulator (Figure 1) which simulates a room of military operations [9], (2) Emergency Medical Technical Training [21]; (3) Virtual ED I (Emergency Department) that simulates a hospital environment with the presence of a virtual team controlled by users [13]; (4) The Pre, Trans and Postoperative Virtual Environment (OPVIR) [12] that is a pre-training education tool for the teaching the basics surgical principles in medical school; (5) Simulator for team training in procedures of the Advanced Support protocol Cardiovascular Life (ASCL) [14]. A comparative analysis between the simulators surveyed regarding the evaluation and nature of trained skills can be seen in Table 1.

Table 1. Comparative analysis of assessment requirements of collaborative simulators in the literature.

CVE	IAS ?	CAS ?	Nature of trained skills
3DiTeams [9]	Yes	Yes	Teams coordination
TEM Simulator [21]	No	No	Communication skills
Virtual ED I [13]	No	No	Clinical skills
SAVC Simulator [12]	Yes	No	Technical skills
OPVIR [14]	Yes	Yes	Technical skills

As noted of some cases among the collaborative applications surveyed, the majority support the single user's assessment focusing on improving just technical skills. Given the above, as well as based on more detailed literature reviews on collaborative surgical simulators [12, 13, 14, 16] it is pointed out the need for CVEs development to enable the training and joint evaluation of collaborative procedures.

III. TECHNICAL ASPECTS ON COLLABORATIVE MEDICAL SIMULATORS DESIGN

This topic will address some of the main aspects related to CVEs which supports collaborative evaluation, such as networking infrastructure, used protocols, synchronism between distributed VEs instances, collaboration modes, among other aspects.

A. User's Training Evaluation

According Cosman et. al. [16], VR simulators offers two great advantages. First of them, improves surgical skills and the second, objectively measures the evaluation of the training through performance reporting generation. The collected data from these reports turns possible the measurement and analysis to quantify the evaluation performance and the student knowledge level about such procedures. Thus, a Assessment System (AS) coupled to the VR simulator should monitor and verify in real time, the level of success of the performed procedures [13].

Virtual Objects placed into 3D space, applied forces, speed and sounds can exemplify interaction data which are monitored from simulators and used for evaluation [28]. Regarding the response time of SAs, these can be classified as *Online* (real-time:. ~ 1.0 sec) and *Offine* (non-real time) [10]. The evaluation in these systems can be deployed according to different decision models (e.g. A.I techniques, visual and comparative analysis). As presented before, the SAs may also be classified about the number of evaluated users ISA (Individual System of Assessment) or CSA (collaborative) [11].

B. Synchronism

In early research related to graphic simulators with multiple users, Singhal and Zyda [15], suggest some alternatives to solve the lack of synchronization between different instances of the same VE spread by network , such as: active replication states and prediction algorithms (dead-reckoning). Both help in sharing the exact location of a virtual object in real-time. Thus, the exact path of medical tools and surgical team members' positions can be precisely measured.

The active replication it is the most trivial mechanism and more costly, where each host is responsible for sending periodic messages containing the current state of their local copies, to be updated in the remote clients [15]. This mechanism is also known as heartbeat, as an analogy to the human cardiologic system. The dead-reckoning algorithms, in turn, are based on the calculation process of the current position of the given object from a predetermined position, advancing along the time and route. To this end, data from its last known cinematic state are used (e.g. position, velocity, acceleration) [6].

C. Data distribution

Different existing network communication mechanisms may be used for transmission of data through the network, keeping updates stability in all CVEs existing instances. Among the mechanisms used, Remote Procedure Calls (RPCs), Remote Method Invocation (RMI) or the use of different network sockets APIs. In the first cases, it becomes possible to call a function or method in a remote location with addressing major abstractions of low-level code. All these methods are frequently used in distributed systems [15].

D. Network Architectures and Data Storage

- ***Authoritative Dedicated Server (Client-Server):*** In this approach, the server creates a map where all the virtual objects are instantiated without graphics rendering. Final users send their updates only to the server, which in turn performs the communication validation, redirecting it to other remote users. This model presents greater stability's connectivity. As expected downside, we can mention the extended response time added by communication links between clients and server.

- ***Non-authoritative Server (Peer-to-Peer)***: Each client is responsible for updating its CVE's local instance and for synchronism maintenance. This is the active replication method presented before. In this approach, one of the hosts acts like a server, and is responsible for establishing connections as well as intermediating all communications. Differently from the other approach, the server is a final user's machine and it do have graphical rendering. Thus, all users controls their logical flow locally without any processing on the server side. One disadvantage from this model is the increased volume of data by unnecessary updates that must be continuously received and processed (e.g. position of invisible avatars) by all hosts. Therefore, it is recommended for CVEs of LAN scope and/or fewer users.

E. Collaborative Modes

Importantly to observe, the collaborative tasks in CVEs usually occurs in two ways with different names found in literature: a) simultaneous (concurrent or synchronous) collaboration or b) sequential (asynchronous) collaboration [23]. In the first case, remote users operate modifications on the same or different properties of the same virtual objects at real-time, such as when the surgeon and surgical assistants operates on the same operative field or vital organ on the patient. In the second collaborative approach, user actions occur sequentially. For an example, when the anesthetist induces the patient to the state of unconsciousness through general anesthesia so that the surgeon can start the operation, among other possibilities.

F. Reliability on data transmission for CASs

To implement a CAS, it is important to emphasize that it is essential that the interaction data is not lost and remain intact along the transmission, as these data represent the variables that are relevant to the team evaluation process. Another important issue to be considered by decision models for assessment teams is the network latency, since the Internet conditions become unpredictable. It's also important because in a real operating room, procedural delays during execution of collaborative tasks can cause damage to the patient [28].

A collaborative evaluation model adaptable to network latency is proposed in [17], which is incorporated and adapted to our project. In this model, a Local Assessment System (LAS) performs the preprocessing of the individual evaluations for a set of collaborative tasks performed between two or more users. Thus, when a user "A" interacts with the other "user B", this interaction is measured locally, using the minimum amount of time that this interaction took to complete, in their respective LAS (users A and B). Thus, the CAS should identify the relative positions of users and the time spent between each user actions (d) for each collaborative task, being discounted the differences in latencies between all subnets present in the simulation in order to provide adaptability of the evaluation model:

$$\Delta t \leftarrow |\ t - (-d)\ | \qquad (1)$$

Where t = start time of a collaborative task (tc) computed locally by LAS tool; d = network latency (in milliseconds) which is represented by a negative value; Δt = adaptation of latency computed by CAS. A case illustrating its use can be seen in Figure 1. In this case, it's made the sharing of one important global variable for team assessment (Tp). This variable represents the surgical preparation time and comprehends all moments between the general anesthesia induction (made by anesthesiologist) and beginning of surgical procedure (the incision made by the surgeon). For computing Δt in this case, t assumes the value of Tp. In this case, while Tp increases on the anesthesiologist side (Tp = 0.0 ... 15.0 ms), he/she should send the update messages to other team members.

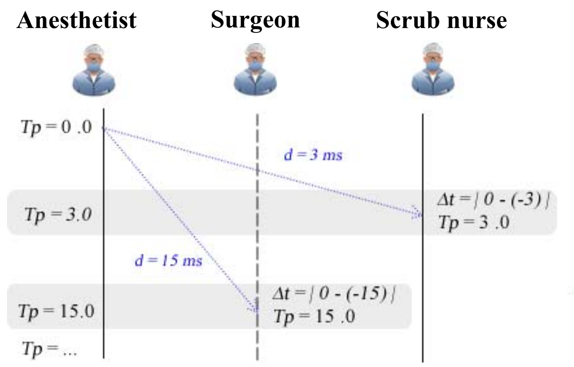


Figure 1. Sharing of the global assessment variable Tp (surgical preparation time) among all participants in the simulation.

Thus, each update packet sent should contain, in addition to the value of the variable Tp, the value of the network latency (d), so that Δt value can be computed in the remote users. This way, the state of this and all others global variables, must be synchronized in all user's CAS. Thus, everyone has synchronized variable locally, so they can use it in the general equation that computes the final team score Finally, with this simple method, the network latency can be adjusted without the need of more complex mechanisms, such as bandwidth control and others.

IV. PLANING AND DEFINITION OF SIMCEC ARCHITECTURE

According to Machado and Moraes [17] the project of a medical VR simulator project should start with a full detailed revision of the requirements and procedures involved, especially for those that enable the evaluation of the user's training. Thus, during the planning phase, it was defined a systematic methodology for analyzing SimCEC requirements, already addressed in previous studies [11, 16] in which were defined (not necessarily in this order):

- Main objectives;
- Definition of the target audience;
- Nature of trained skills;
- Simulated surgical stage (pre-, intra-, post- or perioperative);
- Individual and Colaborative tasks;
- Graphical realism degree;
- Evaluation model and its metrics;
- Detailed error's prediction;
- Collaborative approach (concurrent or sequential);
- Detail of tasks: levels and stages;

The *SimCEC* ***main goal*** is to assist students in the process of education and evaluation inside different health courses about the basic tasks and issues involved in surgical procedures. The ***surgical stage*** that was chosen to be simulated was the pre-, and intra-operative phases, ranging from patient preparation to the begining of incision.

The ***covered specialist content*** includes basic preoperative procedures and concepts, such as: verification of X-rays of the patient, selection of anesthetic drugs as well as surgical tools, asepsis and antisepsis, handwashing protocol, user's paramentation, biosafety control, among others. For intra-operative stage, it was chosen three surgical cases of mandibular fractures to be simulated. The cases are randomly generated by the simulator (variating severity and region of fractures), so that the surgeon can make his/her best decisions. More information about SimCEC's contents and its requirements can be found in previous works [11, 16] as well as in project official webpage [21]. The system architecture that contains four main modules and the relationship between them can be seen in Figure 2.

A. Graphic and Immersive Module

This is the module responsible for the graphical rendering, i.e., the process of drawing 3D scene and updating modifications on this virtual world in real time, allowing the visualization and interaction with the VE. Among the key requirements identified for this module, in order to facilitate collaborative interactions, stand out: a) Need to provide a sense of mutual presence to users who interact within the CVE; b) Guide users along the simulation about the tasks to be performed individually and collaboratively; c) Promote the sense of immersion and allow natural interaction (NI) capabilities; d) Loadable 3D virtual objects and graphics rendering in real-time; e) Provide texturing, lighting, use of shaders and animations.

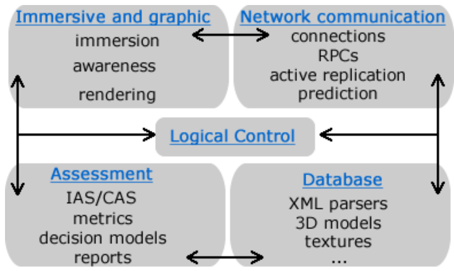


Fig. 2. System architecture of SimCEC and the relationship among its five main modules.

Thus, in order to approach these requirements, the module was sub-divided into three: awareness (a-b), Immersion (c), rendering (d-e).

Awareness sub-module: The visual graphics returns are generated through techniques to keep the users' attention and knowledgeable (sense of awareness), guiding them along the interaction with virtual objects and/or other virtual characters. Therefore different interactive techniques can be used and usually they're divided into sub-steps: selection, engagement, manipulation (rotation direction) and release. Thus, different types of visual feedbacks can be mapped for each action inside the VE, such as the change of colors and sizes of the objects, use of lighting and 2D messages identifying labels. In the context of collaborative activities, some researches have been developed in order to enable the cooperative manipulation of virtual objects (concurrent collaboration mode). For SimCEC, some visual feedback techniques have been predefined to be triggered by each type of user's action and default system alerts (Table 2).

Table 2. Predefined visual feedback techniques, according to user's input or system alerts.

Trigger events	Technique Description
Identify Avatar position	2D Label
Identify interactive objects	2D Label
Error's execution	2D Informative messages
Definição de áreas para os postos individuais de trabalho na sala ciúrgica	Illuminated floor area
Notificação de nova mensagem no chat	Illuminated button

Immersive sub-module: This module is responsible for manage the features of user's interaction through the use of different types of VR devices, which may be supported. In this way, one important requirement is the software modularity and possibility of expansion of previous interfaces and classes. Thus, the sub-module enables the addition of new API and plugins¸ that support the implementation of different types of drivers, as well as supports different visualization methods (e.g. active and passive stereoscopy, multiprojection). Depending on the chosen execution platform and the procedures performed, tracking devices for Motion Capture (MoCap) users become more or less important. For example, in a system of multiple projection like a CAVE[1], the motion tracking becomes important once it informs the correct location of each member inside the operating room. Also the way surgical tools are handled can be detected and compared to predefined patterns considered to be assertive. Initially it was implemented two immersive techniques for the SimCEC: stereoscopic viewing by anaglyph and motion tracking by infra-red.

Rendering sub-module: VR systems and computer graphics in general, have as fundamental requirement to provide the optical illusion of continuity by maintaining the refresh rate of the graphics scene (> 30 fps or 60 Hz) [21]. In that way, realism and interaction with the VE do not be compromised. Thus, this module uses computer graphics techniques such as rasterization, movement interpolations, transformation in graphical matrices, among others.

B. Network Communication Module

Responsible for establishing the connections between the participants of the collaboration and sharing data across the network, as well as managing its synchronization between the different environment instances. The module contains the sockets interfaces and implementation of RPCs that are used, and others implementation mechanisms which provides security, data integrity, particularly of the evaluation global variables. The RPCs methods were used for actions performed infrequently inside the CVE, as in the case of opening doors, sending text messages and updating of global variables of the surgical team. The prediction method, on the other hand, was used for handling updates and positioning of the characters while they walk in the virtual space, which is an event that needs to be constantly checked. Finally, the module also provides an instant messaging sharing system so that the team of students can communicate each other and perform their collaborative tasks more easily (Figure 3). Two versions of the simulator were developed according to the desired scope of network: **Local SimCEC** and **SimCEC in Cloud**.

[1] *CAVE* - Interactive and immersive system, where AV is projected around the user covering much (or almost all) of their visual field.

C. Logical Control Module

This module is responsible for executing the logic control of the system, i.e. the simulation flow, performing all the necessary updates. Among its main functions are: reception of user interaction data, collision detection between avatars, movement of characters in virtual space, monitoring the current state of the simulation, through reception of various events, among others.

Fig. 3. Three students connected to SimCEC Cloud.

Currently, this module uses an usual classic logic rule-based system containing elements such as antecedents, consequents and logical operators (e.g. and, or, not). However, the module is also expandable so that in the future, the system can be added with more sophisticated techniques of A.I. These will enable the control of virtual actors, in a similar way of how non personal characters (NPCs) entities works in video games. In that way, the presence of other remote students for performing collaboration can be avoided, when just one single user wants to practice the surgical procedures of a specific member.

D. User's Assessment Module

This module is coupled to the VR simulator, which means there is located internally to the system. This module contains the IAS and CAS, and are presented therefore in all instances of the simulator. These systems store all relevant interaction data for evaluating the group of users. Addressing to some issues related to the computational requirements of assessment systems in VR simulators, Machado and Moraes (2013) pointed out that the main problems related to ASs are the accuracy and complexity [21]. Thus, the evaluation models used by the ASs should not be computationally complex so that compromise the execution of the simulation, and should also evaluate the user in a precise way so that judgment errors can be avoided [11]. Upon completion of the simulation, the CAS is responsible for sending the collaborative report for the whole team, also including the received individual reports.

The system also enables that N remote students can connect to the CVE and observe the simulation.

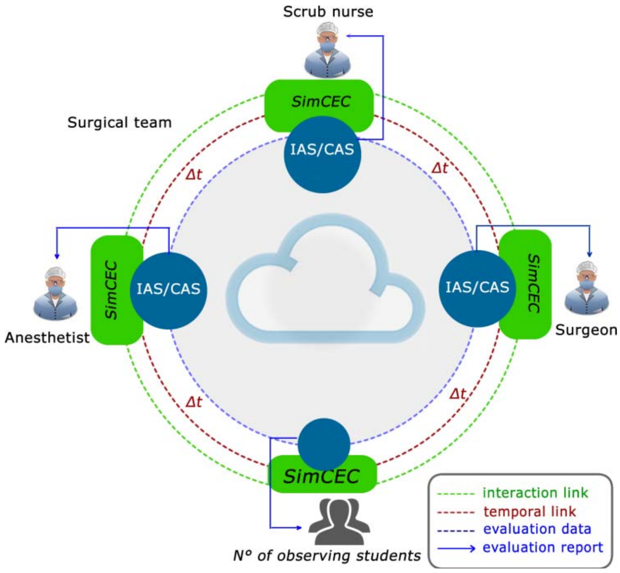


Fig. 4. AVC architecture that supports collaborative assessment (SimCEC in Cloud).

Thus, the system can be used in school classes with lot of students which can observe the performances of other student groups. The Local SimCEC architecture has been already presented in a previous study [24].

E. Data Storage Module

Since the objective of the simulator is to train graduating students, with special attention to the procedures' evaluation metrics as well as the courses' curriculum, the necessity of keeping the reports of the students' performances for posterior analysis was observed. In this context, the SimSEC brings the possibility of storing the reports. That way, students and professors are able to acess the performance reports created from previous simulations, to compare them to the current results, being able to verify the learning progress. Among the different data storaging techniques available, the XML (eXtensible Markup Language) was chosen. The XML is a marking language capable of describing many kinds of information in a simple and structured manner.

V. Development: Implementation Issues

The development phase of the simulator involved some other phases, namely: scenario modeling, data structure modeling, collaborative evaluation algorithm modeling, and design of the animations, among others. Initially, a comparison between the different game engines was made, aiming to analyze and compare their main functionalities, in special: 1) Network communication, 2) Supported VR devices, 3) Development programming languages and 4) Running platforms.

The following 3D engines were compared: Unreal Engine 4.0, Source Engine, Unity3D 4.6 and Vizard 5.0, as shown by Table 3. In this work, the Unity3D game engine was chosen, because of its wide array of tools for network communication. This platform is particularly known for being capable of building its projects for multiple platforms. Another advantage is the possibility of an integrated IDE for the development of scripts in the Unityscript (an special Javascript adaptation), CSharp and Boo programming languages. Regarding the basic data structure used by the the Unity3D, all the virtual objects (GameObjects) are derivated from the Object class. Game objects are static objects that execute no action unless they are commanded to. For that to happen, scripts and Unity components (e.g. colliders, audio sources, etc) need to be attached to these objects. Scripts are routines that will make that the virtual object present some behavior or execute some action given the expected conditions.

Table 3. Comparison between the 3D systems development tools.

	Unreal 4.0	Source	Unity 4.6	Vizard 5.0
1.	Yes	Yes (SDS)	Yes (PUN, uLink,Bolt)	Yes
2.	HMD	HMD	HMD, MoCap, Haptics, CAVE	HMD, MoCap, Haptics, CAVE
3.	SteamOS, Web, OS X, iOS, xBox, Mac, Android	Win, OS X, Unix, PS, Xbox 360, Android	Win, Mac OS, X, Unix, Mobile, Games, swf, Web	Web, iOS, Android, Win.
4.	UnrealScript, C++, Flathead	Souce Code	Unityscript, Boo, C#	Python

A. SimCEC versions

1. Local SimCEC: This application utilizes the standard network packages from Unity UN (Unity Networking). The basic classes of the UN package offer all the functionalities and fundamental parameters for the development of distributed applications. The network class can be considered the most fundamental class of the UN, containing essential network attributes. On this version of the application, a non-authoritative and non-dedicated architecture is used, as explained previously. Therefore, it was necessary that both codes responsible for the initialization of the server and client connection were present in all instances of the CVE, for that purpose the class comboConnect was developed. It has the following function: store the informations of the local member, avatars' data of all members of the surgical team, stabilishment of the network connection, main camera configuration, among other operations. Initially the host applications makes the following call, defining the parameters:

```
Network.InitializeServer(50, port, false);
Network.Connect(ipServer, port);
```

It is important to note that the translation of the addresses isn't necessary since the application scope is local (LAN).

2. SimCEC on the Cloud: In this case, the package for development of distributed application PUN (Photon Unity Network) was chosen because it showed numerous advantages over the other solutions considered, like Bolt, uLink, and others. Among the advantages observed, we can highlight the connection uptime assurance through many dedicated servers in the cloud, at low cost. The PUN utilizes the concept of multiplayer game rooms, where the players connect themselves and play. However, the SimCEC only provides one room for the players to coonect to. Being this room created randomicaly:

```
PhotonNetwork.JoinRandomRoom();
```

The rooms can be created with different configurations, being some options (RoomOptions) defined in the process, like the maximum number of possible connections (maxHosts):

```
PhotonNetwork.CreateRoom(null,new
RoomOptions() { maxHosts = 10 }, null);
```

As observed, the SimCEC on the cloud doesn't require any definitions of IP addresses, ports and the use of protocols like NAT during the connection. Thus, each participant that connects has their informations stored locally by the SurgicalMember class, such as: role on the team and the data structure containing the native network informations (NetworkPlayer) and the player's 3D avatar model.

B. Movimentation and character control

Each *SurgicalMember* type instance possesses a control class *CH_Controller*, that has as it goal to allow movements (interpolation and prediction) according to data from the users interactions, update the animations from the avatars and monitor the collisions. For attenuating any delay consequences on the graphic scene, a prediction and interpolation algorithm was developed based on its cinematic data (position and speed). Part of the algorithm code is shown on Algorithm 1. In this case, the position vector and the current speed are transmitted (lines 4 and 5). On the remote clients, the data is received (lines 8 and 9), the network delay (d) is computed by the difference between the current time and the time in which the last update was performed (lines 10 and 11). Finally, the prediction of the next position (pos1) is calculated, based on the received speed and the calculated delay (line 14).

```
1.  void        OnPhotonSerializeView(PhotonStream
    stream, PhotonMessageInfo info)          {
2.
3.     if (stream.isWriting)         {
4.         stream.SendNext(rigidbody.position);
5.         stream.SendNext(rigidbody.velocity);
6.     } else {
7.
8.      posSin = (Vector3)stream.ReceiveNext();
9.      velSin = (Vector3)stream.ReceiveNext();
10.     T = 0f;    d = Time.time - lastT;
11.     lastT = Time.time;
12.
13.     pos0 = rigidbody.position;
14.     pos1 = posSin + (velSin * d);
15. }
16. }
```

Algorithm 1. Piece of code containing the prediction algorithm of avatars movement based on their kinematic data.

```
1.  public void Update()                       {
2.
3.      if (photonView.isMine)
4.          defaultInputMov ();
5.      else
6.          interpolateMov ();
7.  }
8.
9.  public void interpolateMov()               {
10.
11.    sincT += Time.deltaTime;
12.    rigidbody.position = Vector3.Lerp(pos0,
13. pos1,T/d);
14.
15. }
```

Algorithm 2. Piece of code containing the graphical loop and interpolation of movement functions.

On the main graphic loop on the same script CH_Controller (Algorithm 2), two functions for the character movements are defined: the defaultInputMov() and the interpolateMov(). The first one is called when the code is executed locally, i.e., directly from the input data, without the need of prediction to cover the network latency (lines 3 and 4). The second is executed only for renderization of the remote characters, where it interpolates the initial (pos0) and the final (pos1) previously calculated, resulting in a smooth and synchronized motion effect (lines 6,11 and 12).

C. Scenario Modeling

Some of the 3D objects were modeled using the Blender software, and the others were obtained as free models which were downloaded from different sources on the web. More complex models were also composed, containing many layers, like the patient's head model that is used at the moment of the intra-oral incision. This model contains the complete jaw with the dental arch, tongue and the mouth opener (Figure 5-A).

Regarding the avatars, they were obtained from the *Extreme Medical Characters* package and the stereoscopic method utilized was provided by the package *Stereoskopix FOV2GO,* both these packages can be found at the official Unity Asset Store [18]. This package allows the creation of immersive VEs through the utilization of stereoscopy techniques, like the anaglyph and the side-by-side (used in HMD's). Also, it prevents the programmer from studying about stereoscopic synchronization techniques and the optimization of stereoscopic scenes. Aiming to simulate a realistic surgical environment, metallic shaders were also used to simulate the tools and surgical table's textures. Rasterization algorithms and Bezier curves were also implemented for the simulation of strings on medical equipments and the trajectory of the electrical scalpel.

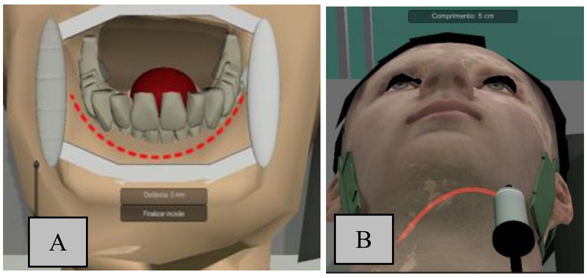


Figure 5. Some of the virtual objects representing the patient's head during the incisions of intra-oral type (A) and extra-oral (B).

D. Collaborative Assessment Model

The evaluation algorithm implemented by both assessment systems (CAS and IAS) contain a set of equations that compute the evaluation scores (e_j) based on relevance weights for each individual procedure (t_j) or sequential collaborative tasks (*tc*) located in a weights vector $W = \{ w_1, ... , w_j \}$. Thus, based on the objects of the type $M[m_k][t_j][e_j]$, individual performance scores (m_k) are generated for each member of the surgical team, as shown in (1).

$$m_k = e_1 w_1 + e_2 w_2 + _{...} + e_j w_j \qquad (1)$$

Where $M[m_k][t_j][e_j]$ is the multidimensional vector, containing the individual task objects (t_j) and their corresponding performance classification algorithms (e_j). Each member is identified by m_k , being k the total number of members on the team and j the total number of tasks from each member. Therefore, each weighted equation similar to the equation (1) is computed for all the member and stored in *Ei,* which represents the individual performances. In addition to individual activities, a number of variables that also emphasize the collaborative interaction skills were defined, such as: movimentation (*mov*) based on the members collisions (*c*) and the total of leaves from the work post, the level of collaborative activity (*Col*), ordering of tasks (*O*), level of biosecurity (*B*) and the surgical preparation time (*Tp* that goes from the induction to the incision). Finally, as recommended in [15], it is important that the possible mistakes made by the students be predicted by the evaluation model, with the special attention for the most serious errors (E) that could compromise the patient's life.

$$Eq = Ei.w_1 + B.w_2 + Col.w_1 + O.w_3 - [\, s(Tp.w_4) + mov.w_5 + E.w_6 \,]$$

$$onde \begin{cases} s = 0, se\ Tp \geq 20' \\ s = 1, se\ Tp < 20 \end{cases}$$

It is important to note that before the evaluation model could be incorporated into the SimCEC, it was necessary to perform a calibration process of the importance weights vector (*W*). During this process, a health care professional was consulted to help balancing all weights under different simulated cases. In this sense, all evaluation input parameters were varied in different levels by the expert, being observed each obtained gradation and its influence on the final evaluation. The level of best match between the simulated situations and the obtained scores was then achieved.

VI. RESULTS: SIMCEC-USER INTERACTION AND PERFORMANCE ANALYSIS

The SimSEC and its evaluation systems (IAS/CAS) are the results of the study performed about the use of CVEs and the possibility of integrating decision models as auxiliary tools on the evaluation process. Thus, this section aims to present how users can interact with the simulator in a practical way, as well as to present initial results. Initially, the initial screen of the simulator is presented (Figure 6) to the users, containing the title of the simulator and the menu with the following options:

- *Configurações (Configurations)*: Allow the users to configure the system's interaction parameters.
- *Instruções (Instructions):* Contains interaction informations.
- *Iniciar Simulação (Start Simulation)*: Asks for some information (Function on the team and name) so that the user can be connected to the system.
- *Créditos (Credits)*: Contains information about the development team and the Project.
- *Relatórios (Reports):* Option that allows the user to load previously stored reports.

The default avatar interaction occurs via the mouse (rotation) and keyboard (translation). The system provides three viewing modes (1st, 2nd and 3rd person). The camera on the 1st person mode gives the possibility to use the scroll mouse button to zoom in or out for better viewing the objects inside the surgical room.

Fig. 6. SimCEC main menu.

After the system implementation conclusion, some running tests were performed (Figure 7), aiming to analyze its main functions, like interaction, network communication (delay levels), and online evaluation processing, among others. The following values of rtt (round-trip time) were achieved, according to the used network type: ~295ms with variance of ~84ms (Ethernet 100Mbps); and: ~414ms with variance of ~26ms (Wireless 54Mbps). The local SimCEC version was executed on standalone mode on the Fedora 18 OS only on locally connected PC’s via Ethernet cables. Locally, it was observed that the synchronism between the three executed copies of the VE kept itself stable.

For verification of the evaluation model classification outputs, three group of simulated situations were defined: 1) Best cases, 2) intermediate cases and 3) worst cases. According to their own classification terminology, the cases vary from the best cases, where the participants made none or very little mistakes to grave errors that were committed on purpose. Based on the initial observations, for each simulated case, the evaluation model behaved like it was expected, i.e., according to the predetermined weights, and the obtained scores revealed accordance with the individual and collaborative actions. The system response time was satisfactory, being carried out with the considered ideal time in all three copies, with mean of ~1 ms (online evaluation). An example of the generated report can be seen on Figure 8.

Initial tests with the immersion module were also conducted, with the stereoscopy by anaglyph being used and an interactive manual tracking technique by infrared provided by the Leap Motion® device. One of the reasons for this test was to verify the software modularity requirement in the immersion module, i.e., the ability to insert new devices with low coding efforts. The Leap Motion is a small device with an USB peripheral controller, designed to be placed on a physical desktop, facing up. Using two monochromatic cameras and three infrared LEDs, the devices observes a hemispherical area at a distance of approximately one meter [25]. The device was used to include a hand-tracking prototype for the identification of the correct positioning of the hands on the cleaning procedure (Figure 9-a) and the avatar movimentation on the virtual scenario, via manual gestures of natural interaction.

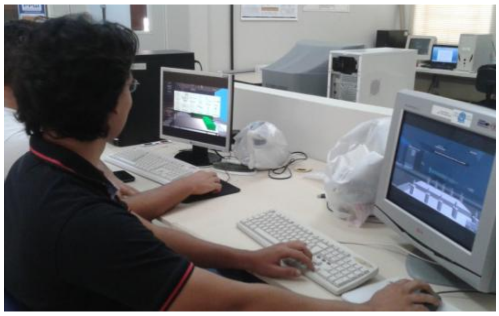
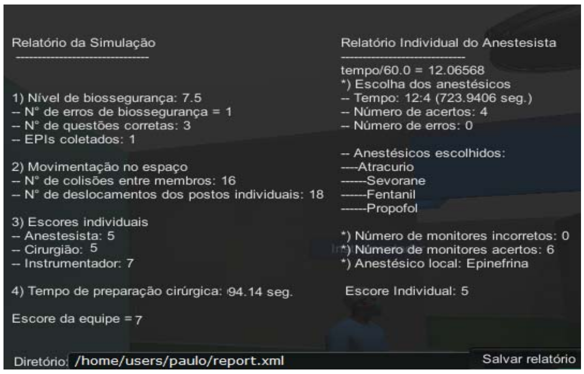


Fig. 7. Local SimSEC execution. The user at left performs the procedure of the surgical tools' selection and user at right select the general anesthesics.


Relatório da Simulação

1) Nível de biossegurança: 7.5
-- N° de erros de biossegurança = 1
-- N° de questões corretas: 3
-- EPIs coletados: 1

2) Movimentação no espaço
-- N° de colisões entre membros: 16
-- N° de deslocamentos dos postos individuais: 18

3) Escores individuais
-- Anestesista: 5
-- Cirurgião: 5
-- Instrumentador: 7

4) Tempo de preparação cirúrgica: 94.14 seg.

Escore da equipe = 7

Relatório Individual do Anestesista

tempo/60.0 = 12.06568
*) Escolha dos anestésicos
-- Tempo: 12:4 (723.9406 seg.)
-- Número de acertos: 4
-- Número de erros: 0

-- Anestésicos escolhidos:
---Atracurio
------Sevorane
------Fentanil
------Propofol

*) Número de monitores incorretos: 0
*) Número de monitores acertos: 6
*) Anestésico local: Epinefrina

Escore Individual: 5

Diretório: /home/users/paulo/report.xml
Salvar relatório

Fig. 8. Evaluation report model. The report contains all the evaluation variables and the obtained scores.

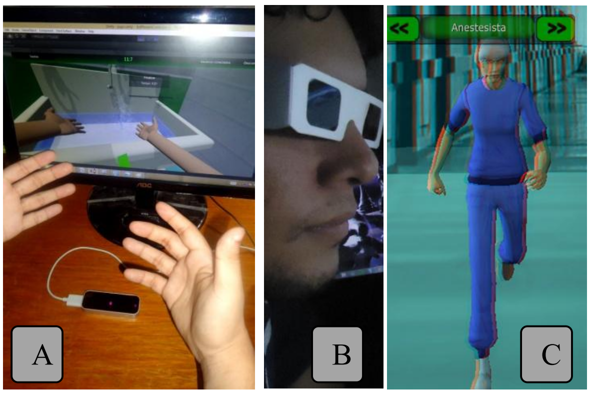


Fig. 9. Interation tests with the graphical and immersive module: A- Manual tracking with the leap motion; B- User with 3D Glasses; C- Stereoscopic Visualization on the VE.

VII. Final Considerations And Future W orks

As observed, many application of simulation with VR does not take into account the assessment of team interaction skill, often focused only on individual technical aspects. As studies show, the same is truth for much of the collaborative surgical simulators [9, 11, 3], and the use of collaborative assessment is often disregarded. According to some authors [30, 36, 13], a possibility for the occurrence of this fact is the little or no rigor established during the process of requirements selection. Among the main contributions of this work, we highlight: design and implementation of the SimCEC, definition of a collaborative evaluation model, the possibility of collaboration between students from different areas of surgical knowledge, among others. Among the specific advantages of the SimCEC evaluation model, we highlight: generation and storage of performance report, real-time feedback (online evaluation), adaptability to network latency. For possibilities of future works, we can highlight the use of the simulator with the target audience and healthcare educators and the execution of the system in other platforms, like CAVEs. Finally, we intent that the simulator can be used in specific moments on graduation level courses, as an auxiliary tool in the process of teaching and learning of the future professionals. The SimCEC-1.0 is already available for navigation at: http://www.de.ufpb.br/~labteve/download/simcec/www.html.

Acklowledgments

This Project was financiated by CAPES and the CNPq (Processes 181813/2010-6, 310561/2012-4, 310470/2012-9) and it is part of the activities of National Institute of Science and Technology – Medicine assisted by Scientific Computing.

References

[1] Aggarwal, R.; Ph, D. Simulation for ward processes of surgical care. The American Journal of Surgery, v.206, n.1, pp. 96–102, 2013.

[2] Marr, M. *et. al.* Team play in surgical education: a simulation-based study. Journal of surgical education, v. 69, n. 1, pp. 63–9, 2012.

[3] Palter, V. N.; Grantcharov, T. P. Virtual reality in surgical skills training. The Surgical clinics of North America, v.90, n.3, pp. 605–17, 2010.

[4] Paiva, P.V.F.; Gadelha, R.; Machado, L.S.; Santos, S.R. Potencialidades da Educação em Saúde Baseada em Realidade Virtual. Revista Eletrônica de Comunicação, Informação & Inovação em Saúde (RECIIS) vol. 7, n. 806, 2013.

[5] Paiva, P.V.F.; Machado, L.S.; Oliveira, J.C. A peer-to-peer Multicast Architecture for Supporting Collaborative Virtual Environments (CVEs) in Medicine In: Symposium on Virtual and Augmented Reality 2012. Niterói/RJ - Brazil. pp. 165-173, 2012.

[6] Machado, L.S.; Moraes, R.M. Evaluation of Collaborative Activities in Medical Simulators based on Virtual Reality In: Symposium on Virtual and Augmented Reality 2013. Cuiabá/MT - Brazil. pp. 244-247, 2013.

[7] Moraes, R.M.; Machado, L.S. A New Architecture for Assessment of Multiple Users in Collaborative Medical Training Environments Based on Virtual Reality . In: 10th International FLINS Conference 2012. Istanbul - Turkey. Uncertainty Modeling in Knowledge Engineering and Decision Making. New Jersey: World Scientific, v.7, pp. 1119-1124, 2012.

[8] Cosman, P.H. *et. al.* Virtual Reality Simulators: Current Status in Acquisition and Assessment of Surgical Skills. ANZ Journal of Surgery, v. 72, 30–34, 2002.

[9] Taekman, J.M. *et al.* 3DiTeams – Healthcare team training in a virtual environment. The Journal of the Society for Simulation in Healthcare, v.3, n.5, p. 112, 2008.

[10] Machado, L.S.; Moraes, R.M. Multiple Assessment for Multiple Users in Virtual Reality Training Environments. Lecture Notes in Computer Science, v. 4756, p. 950-956. Springer, 2007.

[11] Paiva, P.V.F.; Machado, L.S.; Valença, A.M.G. A Virtual Environment for Training and Assessment of Surgical Teams In: Symposium on Virtual and Augmented Reality 2013. Cuiabá/MT - Brazil. pp. 17-26, 2013.

[12] Reis, R.J. *et. al.* Virtual Reality Environments in Surgical Training – Introducing the Pre, Trans and Postoperative Virtual Environment (OPVIR). Brazilian Journal of Videoendoscopic Surgery, v.3, n.3, p.126-133, 2010.

[13] Heirinrichs, W.L. *et al.* Simulation for team training and assessment: case studies of online training with virtual worlds. World Journal of Surgery, v.32, n.2, p. 161-170, 2008.

[14] Vankipuram, A. et al. Design and Development of a Virtual Reality Simulator for Advanced Cardiac Life Support Training. IEEE journal of biomedical and health informatics, 2013. doi:10.1109/JBHI.2013.2285102.

[15] Singhal, S; Zyda, M. Networked Virtual Environments: Design and Implementation, ACM Press/Addison-Wesley Publishing Co, Nova Iorque, Estados Unidos, 1999.

[16] Paiva, P.V.F.; Machado, L.S.; Valença, A.M.G.; Moraes, R.M. (2014) Simulação Cirúrgica: Aplicando uma Arquitetura de Avaliação Colaborativa de Equipes. In: Symposium on Virtual and Augmented Reality 2014. Salvador/BA - Brazil. p. 307-310.

[17] Moraes, R.M.; Machado, L.S. (2013) Simultaneous assessment of teams in collaborative virtual environments using Fuzzy Naive Bayes. In: 2013 Joint IFSA World Congress and NAFIPS Annual Meeting (IFSA/NAFIPS). Edmonton - Canada. p. 1343-1348.

[18] Unity3D. Online: <http://portuguese.unity3d.com/>. Accessed in: 08/2014.

[19] Leap motion. Online: <https://www.leapmotion.com/>. Accessed in: 08/2014.

[20] Chodos, D. et al Healthcare education with virtual-world simulations. In Proceedings of the 2010 ICSE Workshop on Software Engineering in Health Care - SEHC '10, New York, New York, USA: ACM Press, p. 89–99, 2010. doi:10.1145/1809085.1809097.

[21] SimCEC. Official web page of SimCEC project. Online: http://www.de.ufpb.br/~labteve/projects/simcec_e.html. Accessed in: 02/2015.