

INTERACTIVE VIRTUAL TRAINING ENVIRONMENT WITH WEB-CAMERA FOR SCIENCE AND ENGINEERING EDUCATION

Hassan Rajaei and Arsen Gasparyan
Department of Computer Science
Bowling Green State University

Abstract

Web-based interactive environments exhibit impressive potentials for educational purposes. Web-based simulation (WBS) offers such a surrounding. Numerous courses in science and engineering can employ WBS techniques to explain and demonstrate complex concepts for students through a simple and pleasant setting. Virtual laboratories can be built with simulation techniques to guide our students learning complex subjects. A major difficulty, however, concerns whether the simplicity may come at the expense of effectiveness of the learning surroundings. One remedy to this problem suggests use of special equipments often used for e.g. pilot training. Another example includes use of digital gloves and head-mounted devices in virtual reality applications. Nevertheless, these equipments are expensive and often not available to most students. In contrast, our solution suggests use of inexpensive tools like web-cameras as an interface to interact with the simulated objects in cyberspace, and WBS as a core building block of the virtual laboratories.

In this paper we propose a paradigm for virtual training environment (VTE) that uses web-cameras as the main interface to the platform. The interactive setting enables the students to move the simulated objects with their hands through the web-cam in cyberspace. In addition, the platform allows the users to collaborate with other students in the virtual world to perform their training tasks in virtual laboratories. We present a proof of the concept, examine the limitations and discuss the opportunities this paradigm can offer.

Introduction

Advances in web-based technologies have provided a spectrum of options for educational purposes. Virtual classrooms and laboratories are speculated to become an integral part of our future educational settings. Various courses in science and engineering can use such settings to effectively demonstrate and describe complex contents to students. While educational spectrum is broad, we limit us to virtual laboratories and trainings. Nevertheless, the concept with some modifications can be applied to the broader virtual classrooms as well. Whereas education and training can have different meanings, we interchange them in this paper to simplify the description of the platform. Furthermore, we focus on the web-camera interface to the environment while we provide an overview of Web-Based Simulation (WBS)[1,2,3,4,5] as a core component of the platform.

Simulation and animation are proven methods widely used to demonstrate capabilities of new concepts in education, industry, and other sectors. Engineering Education exhibits varieties of courses that are potentially difficult for many of our students to observe, analyze, and digest the complexities of the content. Modeling and simulation can alleviate some of the difficulties by providing meaningful models which are capable of showing the details to the students and engage them actively in their learning. The simulated platforms, however, need be relatively simple to project the models to their targeted audiences. Augmented methods and interactions can be used to furnish a pleasant experience for the learners similar to computer games where most of the participants have true engagement in their activities. Distributed web-based simulation can expand

the targeted audience in cyberspace and deliver them with a joyful virtual learning environments[6,7,8,9].

A true learning setting often requires live interactions between the participants as well as the learning objects which aim to ease the learning task. We use web-cam to add video and live interactions to the virtual space where the students and the instructor can not only see each other, but also to interact with the simulated models to examine and experience with them in order to better achieve the learning objectives of the course.

To make the interaction possible, we explore and present methods which allow estimating the position of user's head and hands via image recognition techniques. The data is then mapped onto the virtual 3D avatar allowing the student to control and interact with the simulated objects as if managing them in real life. Moreover, we target to develop a cost efficient system using only cross-platform and freely available software components, establishing the interaction via commodity hardware, i.e. a computer, a web-cam, and an internet connection. Consequently, we aim increasing accessibility and cost efficiency of the system and avoidance of expensive instruments such as gloves and cave system for interaction in cyberspace. We use two test cases to explore the capabilities of our proposed system. One test case illustrates a ball that is passed between several users who interact with a single object, the ball. The second test case illustrates a chemistry lab where two students are performing their tasks with two simulated objects in a virtual lab. Albeit our examples and their implementations may seem simple, they clearly demonstrate the proof of concept that cost-efficient video- and simulation based interactions over the internet can furnish a virtual training environment for educational purposes. An important limitation of current prototype regards the scalability of the system. This issue, which is not addressed in this paper, will be taken care of by employing distributed

web-based simulation similar to a distributed gaming environment.

This paper describes a new method to interact with web-based Virtual Training Environment (VTE) using web-cameras. The next section describes the virtual learning environment including the web-based simulation. To better illustrate the web-cam interactions with the virtual world, two case studies are developed and then presented. The details of the web-cam interaction are sketched are next. Followed by related work is examined in and finally some discussions, future work, and concluding remarks.

An Advanced Learning Environment for Engineering Education

This section briefly describes key components of the virtual learning environment[6,9]. The system is distributed and is accessed by students remotely or locally. An overview of the system is presented below together with some of its key modules.

System Description

The overview of our Virtual Training Environment (VTE) is illustrated in Figure 1. The system comprises several components one of such is the Simulation Service Center (SSC). This center synchronizes all activities of the learning environment and provides needed services to the students. The VTE system aims to offer a user friendly interface. The current prototype however, has a simple and an easy-to-use interface, including simple web forms and a secure login. Security is handled with user authentications, currently maintained by a database. A navigation menu offers quick access to the different features of the VTE. Students are able to select various types of simulation models which may run sequentially or in parallel on a multiprocessor system. For each selected scenario, the users are prompted with the default input values which can be modified according to the training session's need. The output results are delivered to the users with

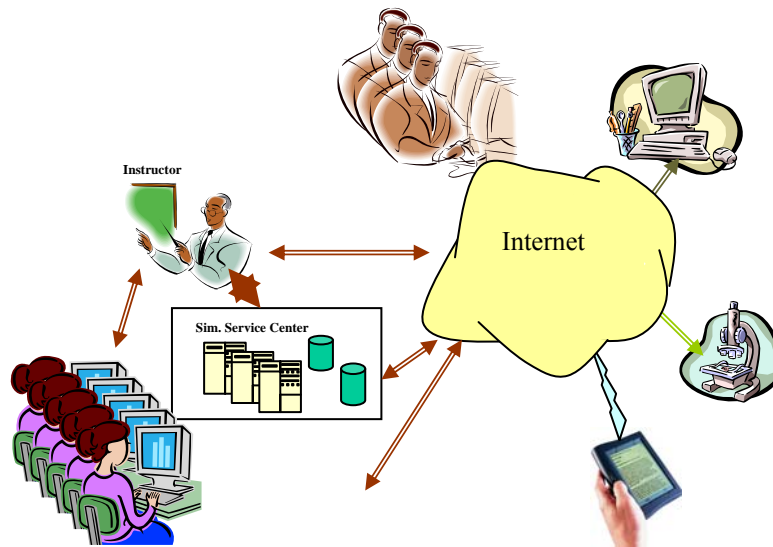


Figure 1: The Virtual Training Environment (VTE) system overview.

various graphs, tables, figures, or animated audio-visual objects depending on the type of selected simulation. This VTE system is synchronous which requires all participants of a course or training activities to login to the system during a classroom session. Clearly the scalability of the system becomes a major challenge.

Simulation and Animation for Interactive Education

The VTE system supports multiple sessions and classrooms. Each course needs to have its own package of model components to be integrated into the syllabus. To facilitate describing this section we use a pilot course like Data Communications and Networking as an example. When a message is sent from point A to B, the simulated model can project animation of the flow between the two points. Depending which portion of the course is under focus, the details of the communication pattern are projected. For example, the routing mechanism in the network and the congestion handling process when part of the network becomes congested due to overflow of packets. The simulation model should allow the students to change the configuration and add complexities to demonstrate how issues such as network management, security, error detection and correction, point-to-point vs. peer-to-peer, etc

are handled with protocols. The complexity will get momentum when teaching wireless communications. Albeit the topic is hot and attracts many students to register to the course, the amount of details in such a course can easily be overlooked if simulation and animation is not employed.

Web-Based Simulation

Web-Based Simulation emerged in the mid-90s, pioneered by Fishwick[3] and Nair and Miller[10]. It rapidly gained attention from the research communities, industry, defense, and many other potential users[4,5,7,9]. Advances in web technology and integration with emerging multimedia have furnished interesting applications and services such as VTE, which is regarded as the next generation of web-based simulation. The interactions between participants as well as with the environment bring about new levels of experience for the users. As a distributed system, WBS allows multiple users to run the same simulations simultaneously.

The system employs multiple, distributed servers to quickly respond to the needs of the students. This VTE facilitates group work and collaboration. The platform meets the users' needs for computational power necessary for complex simulation. The simulation repositories

reside on distributed servers. A client activates a simulation model and provides the needed input data. The inputs are transferred to the server which accommodates the model. The system identifies the targeted server and activates the requested task. That server runs the requested simulation and sends the relevant output results to all students in the group. In our system, all students share the same view as they normally expect in a typical classroom.

The VTE system integrates the web-based simulation techniques to make it possible to the target audience to access the simulated and animated models. Our WBS system is implemented in a shared view fashion such that any change by anyone is viewed in real-time by all participants. The simulation engines reside in the main server at Simulation Service Center while all interactions between the clients and the servers are seamless for the students. Figure 2 demonstrates a snapshot of current prototype.

Cost Efficient Video Interaction using Web-Cam

Virtual environment allows the distributed people to meet and collaborate with each other

via the Internet. For distance learning, however, there is no ultimate solution yet to allow the users to communicate between their groups effectively while sustain scalability (i.e. accommodating larger groups) while maintaining the quality. There are different accessible applications which solve small parts of this problem (as VoipBuster[11] for voice, Skype[12] for video, and Second Life[13] for virtual interaction). One main problem in current systems regards the limitation which does not allow people to see each other's faces and gestures in the virtual space at the same time. The issue become more critical when the users desire to interact with the virtual objects using their hands similar way as they do in real life (or in physical laboratories).

On the other hand there are many research projects which target to enhance the interaction in the virtual environment or to increase the scalability of videoconferencing (CAVE[14] and AccessGrid[15]). Nevertheless, they require expensive equipment (sensors, stereo cameras, special routers with wide channels, etc) which normally is not available for common users.

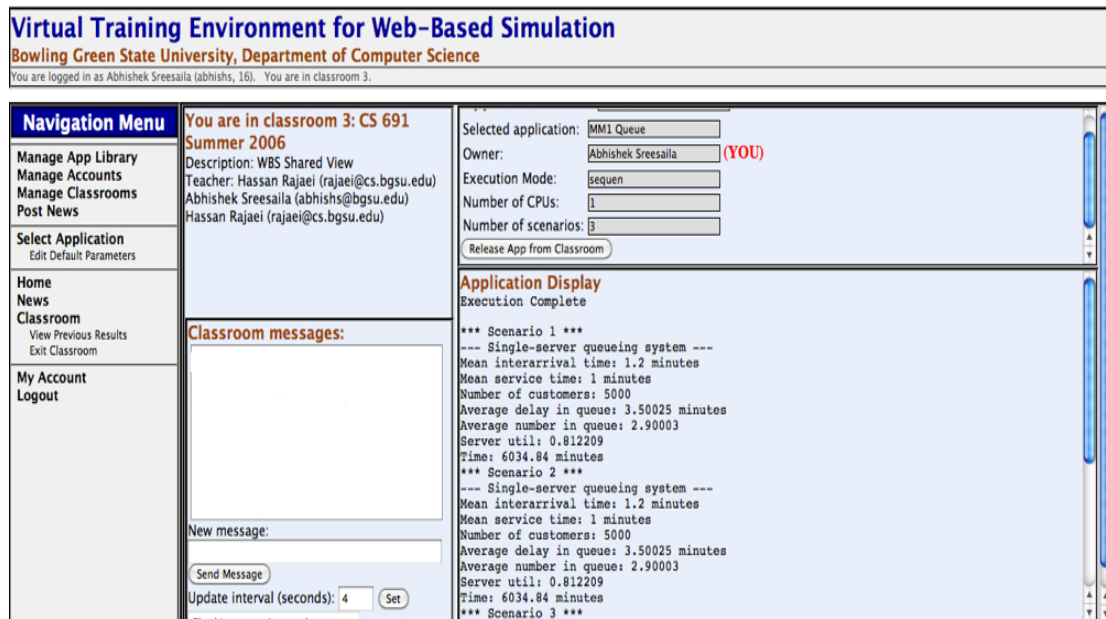


Figure 2: A snapshot of a WBS shared view at a client site.

The hypothesis of this study is to check whether it is possible to interact with the virtual objects using hand gestures acquired by a monocular web-camera; and whether the computational power of a contemporary PCs (or laptops) is sufficient to perform all required image recognition and simulation procedures. The obtained results furnished a proof of the concept and provided the evidence that our hypothesis is correct.

The goal of this work is to examine the potentials of a web-camera interface could offer to a virtual learning environment. Furthermore, to propose a solution for interactions to the system that uses only commodity hardware. The obtained result should also provide reductions in the bandwidth of the communications while increasing the scalability of the system.

Case Studies

In order to demonstrate how web-cam interaction works we developed two test cases to present the idea, identify the basic problems, and indicate whether the interaction problems are solved. The test cases present special categories, therefore different implementation constraints may require.

The first test case is designed to test the very basic and simple functionality of the interaction by following the laws of physics with a rough estimate of hand gesture detection. The

expected results of this test should indicate whether hand-tracking is a viable option as an interface.

The second test case involves more accurate and complicated interaction, simulating a potential real application related to a virtual lab. The expected result of the second test case should indicate whether or not the hand-tracking method is feasible as a main interface to the virtual training environment.

Test Case 1: Simple Ball Game

Real world gaming uses the laws of physics as a boundary to the game objects. That is, in a football or any other ball games, the game object is an elastic ball. The players interacts with the ball, kicking it, moving his/her extremity towards it, experiencing collision forces which try to move the extremity back, and the ball – forward. The ball receives the impulse, accelerates and moves respectively. Friction with air, collisions with ground and other objects affect its speed and rotation.

The interaction between players in real life happens via passing the ball, which is in fact the same as kicking the ball in the direction of the recipient. This observation demonstrates that the real world with its physical laws and properties is a mediator for the interaction between the players as shown on the Figure 3.

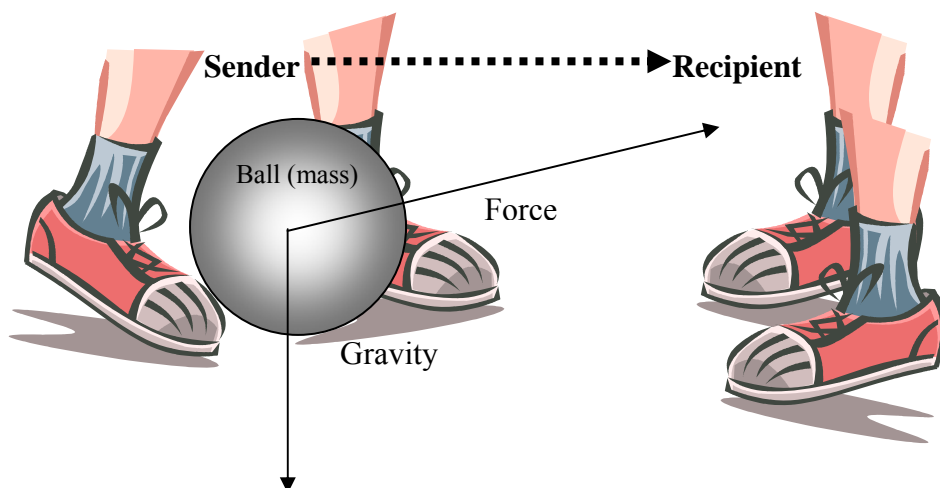


Figure 3: Simplistic representation of the environment-mediated interaction with a ball.

In our work we target to implement the first test case allowing a user to control the extremities of the virtual body with his/her hands and pass the virtual ball to another user. As mentioned above, the images from the monocular web-cam are going to be acquired and processed to allow the interaction.

Test Case 2: Virtual Chemistry Lab

In the test case for chemistry lab we want a chemistry student to pour the chemicals from his/her pot to the pot hold by another student in the cyber lab as illustrated in Figure 4. Unlike the previous case with passing the ball, more actions are required here.

- 1) Student A grabs the empty pot with the help of his fingers (causing small collisions and friction, which do not allow the pot to fall).
- 2) Student B grabs the pot with chemical a similar way.
- 3) Student B moves his pot towards the pot of Student A and flips it (the pot just follows the trajectory of the hand since pot's movement is constrained by hand's fingers)
- 4) The content of the pot falls to the empty pot due to the forces of gravity (Figure 4).

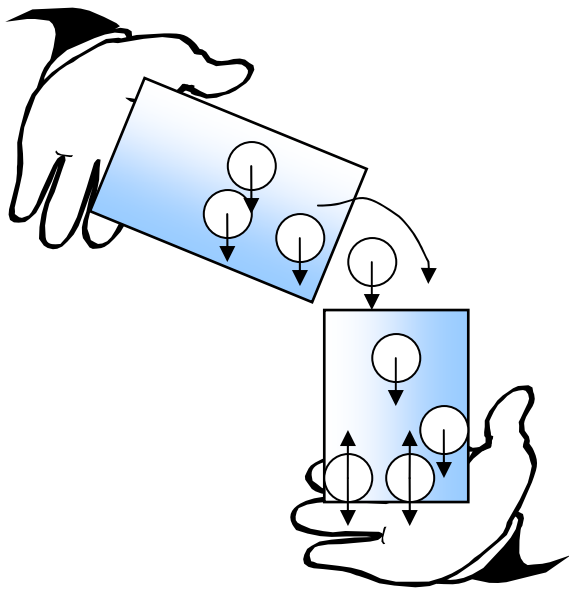


Figure 4: Chemicals pouring from one pot to another following the gravity law.

Observations

Based on several preliminary examples, we made an observation that a small amount of fundamental laws of physics should be enforced to allow establishing variety of different interactions (potentially all interactions we can observe in real life). Having understood how they are actually performed in real world, we can model them in the virtual world as well. Potentially it allows us to find a general solution for realistic and natural interaction in virtual environment instead of developing specialized interfaces for each possible type of user action and object (as is commonly done in most virtual reality and gaming projects).

Web-Cam Interaction Techniques

Design Constraints

Since the subject of this study is video input in the virtual environment, we had to find either the existing virtual environment or software modules could provide the functionality and flexibility necessary for the project considering the learning curve and the limited timeframe. One of the first issues we faced was to decide whether it would be better to use the existing system, or to develop the prototype of new one using freely available engines and libraries.

Before we made a selection, we identified our basic requirements for the system:

1. The system should run on commodity hardware.
2. The system should be capable of 3D-rendering of dynamic scenes on the fly.
3. Unlike other systems, the interaction in the system should be implemented similar to the real world – follow the laws of physics.
4. The system should use the network efficiently considering the real-time nature of the application.

Notice that most of these requirements are common for computer games. Usage of the gaming techniques could be regarded “non-

scientific”, however gaming is in fact a very developed sector of economy which moves forward the whole computing industry and gets as much as it can from the existing computer hardware and user interfaces. Virtual Reality field is in fact very close to gaming, which is confirmed by the usage of VR-equipment in computer games and the growing popularity of WII remote (a relatively cheap 3D manipulator) which appeared on the market recently.

Prototype Implementation

This section presents two prototypes for the two cases studies: the ball game and chemistry lab.

Prototype 1: Ball Game Interaction

Goals

The general goal for this prototype is to implement the first test case (“Ball game”). The objectives are the following:

- To integrate the project with OpenCV[16] library and to see if it is able to get the image from the web-cam.
- To integrate the face detection algorithm provided from OpenCV to the project, and to map the detected face to the arbitrary 3D objects (later should be mapped to the head of the virtual body only).
- To adapt and enhance the algorithm developed in the preliminary research for hand tracking in 3 dimensions and to kick the ball with hand gesture.

Design

The main design innovation (except adding OpenCV calls) was about embedding and improvement of the hand tracking algorithm. The hand angles were estimated based on the average angles of pixels on the filtered difference-maps relative to the estimated position of the body. That algorithm worked well only during intense and frequent motions

with both hands, and it allowed estimation of only two angles in one plane.

When we have integrated Haar Classifiers Cascade[17] into the implementation, we got an opportunity to improve the estimation of the human pose using the knowledge acquired by the face detection shown in Figure 5. This improvement increases the accuracy of the method and makes the obtained images more stable by reducing the role of some evenly distributed pixels between the hand motions as illustrated in Figure 6.

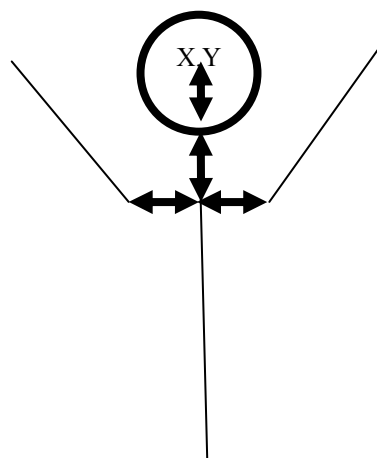


Figure 5: Detected face coordinates and size allow to approximate position of body center and shoulders easily.

In addition, the algorithm could be slightly modified to roughly estimate the angles for the new dimension based on the visual hand deformation due to the motion along the horizontal plane, as shown in Figure 7 and Figure 8.

Prototype 1 Implementation

OpenCV[16] library was integrated into the project successfully without any serious problems. Code from the face detection example also was integrated easily. However it took some time to figure out how to update the Irrlicht’s[18] textures on the fly and how to set camera’s fps (frame per second). OpenCV library is still in development and lacks this feature. The first attempt to implement the

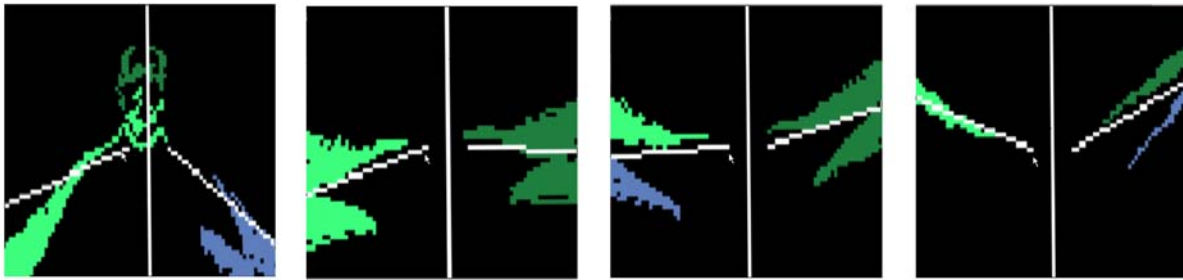


Figure 6: Hand tracking based on information from face detection (estimation of the human pose based on detected information as shown in Figure 8).

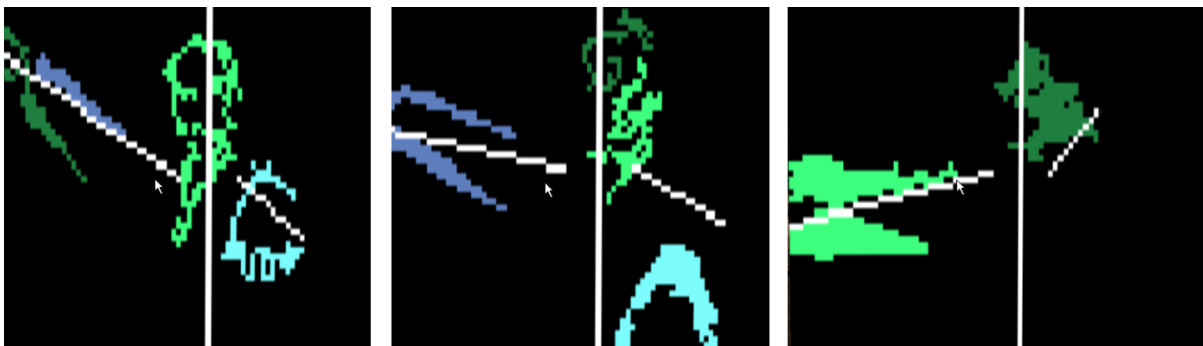


Figure 7: Visual hand deformation due to the movement in the horizontal plane. The detected information is projected to estimation human pose as shown in Figure 8.

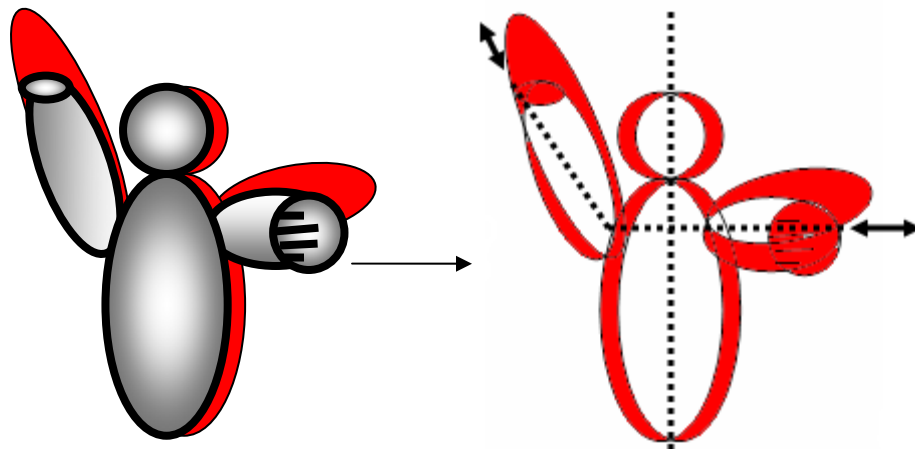


Figure 8: The rough guess of angle in a horizontal plane can be achieved via estimation of the visual hand deformation due to the movement in that dimension.

prototype caused drastic decreasing of the frame rate down to 2-4 fps, which was obviously unacceptable for the project and could result in its failure. However moving the camera-related

routines out to a separate thread solved the issue and the fps came back to normal about 30 fps.

Later processing of hand motion was moved to an individual thread since it is important to get this information as fast as possible and hands normally move much quicker than the head. This prototype showed that it is in fact possible to integrate image recognition with word (re)-simulation and rendering on an ordinary computer and even with existing primitive algorithm of hand tracking (which probably could be improved or replaced with other computer vision techniques) it is possible to kick the virtual ball with real hand gesture.

The secondary conclusion is that mapping the low resolution (~20x20 pixels) image of the detected face appeared to work pretty stable, i. e. not depending on the position of the face in the original image retrieved from a web-cam, it always appear on the object it is mapped to. At the same time the mapped low-resolution image of the face seems to be sufficient to recognize its owner and to guess its expression. However further research with human subjects involved could give a more accurate data on this issue.

Prototype 2: Chemistry Lab Interaction

Goals

The goals of the second prototype are the following

- To create and import the graphical 3D models for hands and the pot.

- To implement the simplified second test case (“Chemistry lab”) adding the pot to the virtual environment and attempting to grab it with hands.

Design

Design of prototype 2 does not differ much from prototype 1 since we use exactly the same principles for hand control and for object interaction. Only 2 changes need to be done:

- To import designed graphical models (3D-meshes) from Blender
- To add the physical model of the pot from the primitives – cylinders and a box.

Prototype 2 Implementation

The new feature of this prototype is the use of graphical models which differ from their physical representation: for simulation purposes hands are still considered to be stretched spheres whereas 3D meshes are being used for their visualization as shown in Figure 9.

Similar approach was used for pot visualization of Figure 11, which contributes to another new feature – a new object of the special shape shown in Figure 10 makes it possible to grab it with hands using the laws of physics (gravity and friction). Despite the fact that this simplified object looks more like a



Figure 9: Visual representation of the avatar (on the left) and its shape for simulation (on the right).

chalice, we intentionally call it a pot to be consistent with our “Chemistry lab” example.

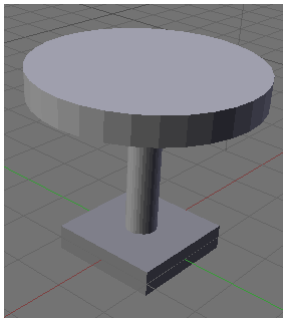


Figure 10: The physical shape of the pot.

Our experiments proved that it is possible to grab the pot via one or two hands (Figure 11, and 12). However it was much harder to do it before we added a vertical attractor to pot to prevent it from tilting. The drawback of using the vertical attractor is that it limits the area of applications of the pot: it is not possible to use it for pouring the liquid from it the way we proposed to do for the second test case.



Figure 11: The pot being grabbed with the stretched hand of the user whose face is projected on the avatar body.

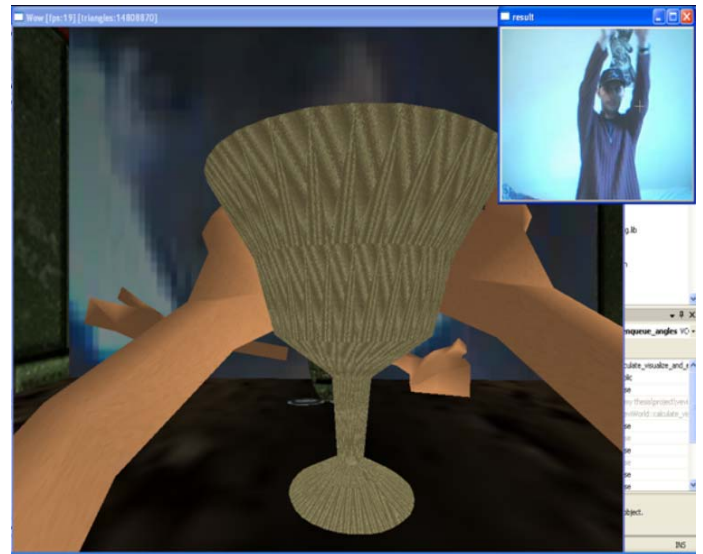


Figure 12: The pot being grabbed with 2 hands (first person view).

We expect that when the accuracy of hand tracking algorithm is improved and more degrees of hands’ freedom are added, it will become easier to grab the pot and to pour the liquid from it without vertical attractor. It will also require more detailed physical model of hands, pot, and liquid.

Related Work

The literature contains thousands of articles related to virtual environments. We have selected, studied, reviewed quit large number of publications. In this paper, however, we present a few cases only. The following aspects of virtual environments were of our interest during this work:

- What is a virtual reality and what is a virtual environment
- Visual representation
- Image recognition or other methods of gesture input
- Physics simulation
- Simulated sound distribution
- Scalability & Networking
- Usability issues
- Related projects

Virtual Reality vs. Virtual Environment

There is noticeable misunderstanding about what Virtual Reality (VR) is and what Virtual Environment (VE) is. Wikipedia (the free encyclopedia) shows the same article for terms claiming that VR is a technology which allows a user to interact with computer-simulated environment, but not clarifying what is VE[19]. Columbia Electronic Encyclopedia states that VE and VR mean the same thing: computer-generated environment with and within which people can interact[20]. According to Burdea and Coiffet[21], Virtual Reality is high-end user-computer interface that involves real-time simulation and multimodal interactions, but at the same time they state that Telepresence and Augmented reality are not Virtual Reality in its strictest sense since they incorporate real images. With this specification it looks like the system we aim to build will not be “strict” Virtual Reality. For propose of current work, the reality is augmented in the VTE platform and hence the augmented VR and simulated based VE becomes closer to each other.

Visual Aspects

Graphical representation is crucial for virtual environments since vision is agreed to be the most important sense[22] which is confirmed by the fact that visual cortex is the largest sensory area in the human brain[23]. Many techniques and algorithms have been developed in the field to make VR look more realistic. We conducted an investigation to obtain an overview of the existing projects to get a better picture of what is already done.

VR²S – a generic VR software system which is an extension of the high-level rendering system VRS, focused real-time human-computer interactions and multimedia responsiveness of the system, for example provides virtual shadows for the real image of the head.

Online Model Reconstruction[24] – is a system for generating real-time 3D reconstructions between the user and other real objects in an immersive virtual environment for

visualization and interaction. This system bypasses an explicit 3D modeling stage, does not use additional sensors or prior object knowledge, but uses a set of outside-looking-in cameras instead. Visual hull technique (proposed by Benjamin Lok[24]) was used to rapidly determine which volume samples lay within the visual hull, and then to build object reconstruction from any given viewpoint at interactive rates. Unlike our project, Online Model Reconstruction utilizes specialized hardware such as five wall-mounted NTSC cameras (720x486) and one additional camera (640x480) to extract the visual hull of human body. Obviously, this type of system is not portable and has limited accessibility.

Second Life[13], There[25], and massively-multiplayer games provide scalable 3D worlds with interactive virtual objects and means of interaction. However they do not allow audio/video conferencing, nor do they provide means for realistic and essential interaction: users have to use keyboard and mouse to control their avatars and deal with simplistic icons and menu items to perform interaction.

Voice-over-IP applications allow real-time audio communications where some of them (such as Skype[2,] Access Grid[15]) allow video conferencing. These projects are specialized in teleconferencing and do not provide 3D-space and virtual objects for interaction.

CAVE[14] and similar projects create illusion of immersion into the virtual world; some of them even use physics for interaction[26]. However unlike our project, they require special hardware and can not become globally accessible.

Potentially the synthesis and further development of existing ideas and technologies could bring us to scalable and universal immersive virtual environment with essential interface, which could integrate audio/video communication with interaction in 3D world. Our contribution to essential interface lies in the field of video-based interaction via common

hardware. We aim to provide the essential way of interaction in the virtual environment without utilizing uncommon or expensive hardware. We plan to empower the computing needs by virtualization (e.g. grid computing) to reduce the necessity in special hardware, replacing it with combination of usual web-camera and computer vision techniques.

Computer Vision

Most of the existing systems (which aim using cameras for control) require special hardware such as stereo cameras. However, there are some schemes which use monocular camera for user input[27,28] as well as necessary computer vision techniques[17,29].

One of the related systems uses Intel's OpenCV library to detect user's face thus locating his/her position in order to allow pointing via user's fingertip. The library is Open Source and free for non-commercial use. It utilizes cascade of boosted classifiers working with Haar-like features originally proposed by Paul Viola[30].

Viola and Jones used four types of features and introduced the term of Integral Image each pixel (X, Y) of which is the sum of brightness of all pixels with $x \leq X$ and $y \leq Y$. After the Integral Image is built, it takes only a constant time to extract image features[30]. A cascade of features (preliminary selected via special learning algorithm) is being used to detect faces or other objects. Later, Wilson and Fernandez[17] enhanced the algorithm to diagonal features.

Discussions and Concluding Remarks

In this work we examined the potentials and proposed solutions for distributed virtual environment with video-based interface. We proved the concept and showed that common hardware is sufficient to allow users' interaction in virtual 3D space via their hand movements acquired from the web-cam and tracked with the help of computer vision techniques. We identified the challenges, developed hand

tracking algorithm, explored the limitations of the used solutions, answered the research questions, proposed a framework and suggested some topics for future research.

This study showed that the video-based interface to the virtual world is feasible by using commodity hardware like web-camera, internet connection, and simple algorithms. As a result, we conclude that the proposed interaction paradigm using web-cameras is a viable solution. Furthermore, the hand movements acquired from the web-camera, mapped to the virtual body, and mediated via modeled physics can be used to e.g. kick a virtual ball or grab a virtual pot. Consequently, we can postulate that through a web-camera we can manipulate virtual objects in cyberspace in the same fashion as we could interact and manipulate real objects in the real world.

The proof of concept is significant since it demonstrates that when we filter out few issues, the hand-tracking interactions can become a very useful technique. When platform is integrated with the simulated physics and 3D graphics, it can provide an essential and globally accessible interface for virtual environments.

While the obtained proof of concept is very impressive, this study also disclosed a few difficulties which we had overlooked in the beginning. One as such concerns the physical avatars. It is significantly different from our point of view with many others (e.g. Second Life) which either do not use physics at all, or they use it in a primitive fashion to enable realistic look of the virtual world, but not trying to use it for the actual interaction.

Another problem concerned mapping movements to virtual body. It appears to be hard to control simultaneously 3-dimensional motion of two extremities via keyboard and mouse in real time. Simulating physical behavior in the virtual environment does not make much sense if we do not map real movements onto the virtual body. However it is not a trivial task to do since it can require using non-traditional input device to track the user's movements.

Even though there are few devices on the market that allow 3D-hand tracking (such as wired gloves or Wii remote), they have limited range of applications. It does not mean that they should not be used in the virtual environment, but this topic could be a subject of other research.

The advantages of using web-camera as an input device are the following:

- it is inexpensive and commonly available
- theoretically it allows not only hand tracking, but tracking of the whole body as well
- it also offers the full-pledged audio visual communications between the participants.

Even though the field of computer vision has various techniques for intelligent image processing, the problem of precise real-time 3D-human pose estimation based on the images acquired from monocular camera has not been solved yet. This work is an attempt to develop such interface by combining open source algorithms and techniques, and to demonstrate that the concept is feasible. Further development of the physical model and image processing algorithms should make it possible to control virtual objects from any location in cyber space using a laptop with embedded web-camera.

Bibliography

1. Bluemel, E., Hintze, A., Schulz, T., Schumann, M., & Stuering, S. (2003). Virtual environments for the training of maintenance and service tasks. In Proceedings of the 2003 Winter Simulation Conference, USA, 2001-2007.
2. De Lara, J., & Alfonseca, M. (2001). Constructing simulation-based web documents. *IEEE MultiMedia*, 8, 42-49.
3. Fishwick, P. A. (1996). Web-based simulation: Some personal observations. Proceedings of the 1996 Winter Simulation Conference, USA, 772-779.
4. Manojlovich, J., Prasithsangaree, P., Hughes, S., Chen, J., & Lewis, M. (2003). UTSAF: A multi-agent-based framework for supporting military-based distributed interactive simulations in 3D virtual environments. Proceedings of the 2003 Winter Simulation Conference, USA, 960-967.
5. Narayanan, S. (2000). "Web-Based Modeling and Simulation." In Proceedings of the 2000 Winter Simulation Conference, Orlando, FL.
6. Rajaei, H. (2004). Distributed Virtual Training Environment. In Proceedings of 2nd Swedish-American Workshop on Modeling and Simulation (SAWMAS 2004), Feb, FL
7. Rajaei, H. & Barnes, A. (2006). "A Real-Time Interactive Web-Based Environment for Training." In Proceedings of SummerSim 2006, Calgary, Canada.
8. Rajaei, H. & Dieball, A. (2007). "A Shared-View Web-Based Environment for Training." In Proceedings of CNS 2007, Norfolk, Virginia.
9. Seibt, F., Schumann, M., & Beikirch, J. (1998). Concept and components for a web-based simulation environment (WBSE). In International Conference on Web-Based Modeling & Simulation, San Diego CA, 189-194.
10. Nair, R. S., Miller, J. A., and Zhang, Z. (1996). Java-based query driven simulation environment. Proceedings of the 1996 Winter Simulation Conference, USA, 786-793.
11. *VoipBuster* (n. d.). Retrieved October 2, 2006 from <http://voipbuster.com>
12. *Skype* (n. d.). Retrieved October 2, 2006 from <http://skype.com>
13. *Second Life* (n. d.). Retrieved October 2, 2006 from <http://secondlife.com>
14. *CAVE User's guide* (n. d.). Retrieved June 08, 2007 from <http://www.evl.uic.edu/pape/CAVE/prog/CAVEGuide.html>
15. *AccessGrid.Org* (n. d.). Retrieved October 2, 2006 from <http://www.accessgrid.org/>

16. *OpenCV Library Wiki* (n. d.) Retrieved February 16, 2007 from <http://opencvlibrary.sourceforge.net>
17. Wilson P. I. & Fernandez J. (2006, April). Facial feature detection using Haar classifiers. *Journal of Computing Sciences in Colleges, Volume 21 Issue 4*, 127-133
18. *Irrlicht Engine – A free open source 3D engine* (n. d.) Retrieved October 30, 2006 from <http://irrlicht.sourceforge.net>
19. Virtual Environment. (2006, September 26). In *Wikipedia, The Free Encyclopedia*. Retrieved October 02, 2006, from http://en.wikipedia.org/wiki/Virtual_environment
20. Virtual reality. (n.d.). In *The Columbia Electronic Encyclopedia, Sixth Edition*. Retrieved October 02, 2006, from <http://www.answers.com/topic/virtual-reality>
21. Burdea, G. & Coiffet, P. (1994). *Virtual reality technology*. New York: Wiley
22. Auyang S. Y. (1999, May) Cognitive and neural processes that make possible vision. Retrieved June 4, 2007 from <http://www.creatingtechnology.org/papers/vision.htm>
23. Mather, G. (2006). *Foundations of Perception*. Psychology Press, 2006 Retrieved June 4, 2007 from <http://www.psypress.com/mather/resources/topic.asp?topic=ch01-tp-0>
24. Lok, B. (2001). Online model reconstruction for interactive virtual environments. In *Proceedings of the 2001 Symposium on interactive 3D Graphics I3D '01*. ACM Press, New York, NY, 69-72
25. *There – the online virtual world that is your everyday hangout* (n. d.). Retrieved June 09, 2007 from <http://www.there.com/>
26. Lee, S., Kim, I., Ahn, S. C., Lim, M., and Kim, H. (2005). Toward immersive telecommunication: 3D video avatar with physical interaction. In *Proceedings of the 2005 international Conference on Augmented Tele-Existence* (Christchurch, New Zealand, December 05 - 08, 2005). ICAT '05, vol. 157. ACM Press, New York, NY, 56-61.
27. Lee, S., Kim, I., Ahn, S. C., Lim, M., and Kim, H. (2005). Toward immersive telecommunication: 3D video avatar with physical interaction. In *Proceedings of the 2005 international Conference on Augmented Tele-Existence* (Christchurch, New Zealand, December 05 - 08, 2005). ICAT '05, vol. 157. ACM Press, New York, NY, 56-61.
28. *SourceForge.net: Jirr* (n. d.). Retrieved July 8, 2006, from <https://sourceforge.net/projects/jirr/>
29. Kjeldsen R. (2005, August). IBM Head Tracking Pointer User's Manual, Retrieved June 8, 2007 from http://dl.alphaworks.ibm.com/technologies/headpointer/Head_Tracking_Pointer_Users_Manual.pdf
30. Viola, P. & Jones, M. (2001). Rapid object detection using boosted cascade of simple features. *IEEE Conference on Computer Vision and Pattern Recognition*.

Biographical Information

Hassan Rajaei is a Professor of Computer Science at Bowling Green State University, Ohio. His research interests include computer simulation, parallel and distributed processing, high performance computing, distributed simulation of communication networks, wireless communications, and virtual training environments. Dr. Rajaei received his Ph.D. from Royal Institute of Technologies, KTH, Stockholm, Sweden and he holds a MSEE from Univ. of Utah.

Arsen Gasparyan has been a graduate student in Computer Science at Bowling Green State University. His research interests included computer gaming and computer graphics. He conducted his thesis work with Dr. Rajaei as partial fulfillment of his master degree.