A Shooting Simulator from Boats

Pablo Figueroa[1], Carlos Francisco Rodriguez[2], Jose Tiberio Hernandez[1], Juan Camilo Blanco[2], Raul Oses[1] and Luis Ballesteros[1]

[1]*Systems and Computing Department, Los Andes University, Cra 1 N 18A- 12, Bogota, Colombia*

[2]*Mechanical Engineering Department, Los Andes University, Cra 1 N 18A- 12, Bogota, Colombia*

Keywords: Virtual Simulation, Shooting Trainer, Military Training.

Abstract: We present a simulator for militar fluvial boats, in which we concentrate in the task of shooting. Shooting from a boat is particularly challenging, due to rapid movements, river conditions, and special considerations that should be taken into account for a shooters security. We present a hardware platform that simulates boat movements under different scenarios. On top of such platform, a user is able to freely move around, aim a target, and shoot by using a handle that holds a display and replicates a large gun. Our system can also report metrics related to a users performance. This paper presents the main hardware and software components, the main tasks we would like to address in our simulator, the information in users reports, and some preliminary tests of the implemented functionality.

1 INTRODUCTION

The Virtual Reality (VR) community has been carrying out research in shooting simulators for a long time. The advantages that VR shooting simulators have for training over traditional methods in real life are unquestionable, and there are several commercial products in this field. One area that has not been studied enough is shooting from rapid boats, where a shooter has to aim at targets at both sides of a river while another person drives. This task requires coordination between people inside the boat in order to avoid hazards and facilitate the shooter's task. It also requires a lot of skills from a shooter, since boat movements could be very fast and it could be very difficult to aim at the targets. A simulator for this situation could be very useful not only for River-based Armed Services but also for entertainment purposes in large video arcades.

This work concentrates on a simulator for a shooter on a fast boat. We simulate the boat's movements at high speed, some stimuli from the river, and people at the river's banks, both foes and friends. Since our main purpose is to provide a simulator for shooting, we do not include a driver. Instead, we use pre-recorded data from a real boat in order to move our simulator as similar as possible to a real situation.

This paper is divided as follows. First, we cite some related work. We then describe the main components of our simulator, such as the motion platform, the shooter interface, the simulation software, and the reporting system. Later we present some preliminary results of tests with experts, and finally we present some conclusions and future work.

2 RELATED WORK

Simulation Systems in the Military field have been broadly used. Here we first mention some general references to this topic, then we introduce some examples of related simulators, and finally we mention some shooting simulators and how they relate to our work.

There are several works that describe specific topics in any military simulation. In (Smith, 1998) , for example, a small taxonomy and a global architecture for simulators are presented. (Page and Smith, 1998) presents an introduction to military training simulation with a very useful glossary that has been used in many other works. This article enumerates the following problems that developers of simulators face, which will be addressed by our work also:

- Domain architectures, or how to re-use standard simulator architectures to accelerate the development process and maintainability.
- Terrain, or how terrain data will be represented

Figueroa P., Rodriguez C., Hernandez J., Blanco J., Oses R. and Ballesteros L..
A Shooting Simulator from Boats.
DOI: 10.5220/0004662200880095
In *Proceedings of the 9th International Conference on Computer Graphics Theory and Applications* (GRAPP-2014), pages 88-95
ISBN: 978-989-758-002-4

among a wide variety of options.

- Behavior representation, or how artificially inteligent entities are represented within a simulator.
- Natural environment, or how to replicate natural environment data in a simulation.

Some boat simulators have been developed in recent years. For example, the Motion Based Sailing Simulator (Avizzano et al., 2010) includes fundamental interaction stimuli and environmental phenomena such as wind and sea conditions, vessel configuration, and actions performed by drivers on real maneuvers. This simulator takes into account a dynamic model for water and wind that considers 6 degrees of freedom (DOF). (Yang, 2008) proposes a mode for evaluation of sailor's skills. In such a mode, formulas based on fuzzy logic assess a driver's steering maneuvers and measure deviations from a desired boat's heading, course plan, and secure operating thresholds. (Perez et al., 2006) presents a Simulink tool for rapid prototyping of Maritime systems. It includes models for ships, underwater vehicles, and floating structures. Overall, it provides models for guidance, navigation, and control of real time ship simulations.

(Bruzzone et al., 2012) presents a simulator for both ship pilots and port operators. An operator can inform a pilot about location of other ships and port congestion status. This simulator can track how a pilot behaves in his entrance to a port, the time spent in a port, docking time, and his interaction with other ships and vessels around. This work is mainly aimed to the simulation of large vessels in their process of approaching to a port. (McKenna and Little, 2000) presents a Submarine simulator where the simulation focus on submarine engagement. This task can take several hours in real life due to the complex process of information gathering before any move can be taken. This simulator replicates the data gathering process in real life, and allows users to make decisions based on a situation. One of its main features is its capability of speed up the simulation time in order to concentrate users in the crucial decision making points. (Mertens, 1993) describes an application for High Commander Officers. In such a simulator, Officers are involved in a battlefield situation that force them to make decisions and communicate to their superiors and subordinates.

There are not many references to shooting simulators in the academic field. Most of the references we found are commercial systems and some patents such as (Stephen P. Rosa, 2003; Lvovskiy, 2005; Seet et al., 2001; Takeda et al., 1999; Wang et al., 1996). Eurosimulator (Dahlberg, 2012) is a Web based simulator that offers a variety of weapons such as pistols, rifles, and revolvers in several indoor and outdoor scenarios. ShotPro 2000 (TROJAN, 2012) allows shooting animated targets in natural scenarios and uses quadraphonic sound. ShooterReady (Christensen, 2000) is a long range simulator designed to train how environment variables affect a bullet's course.

CAPS (Young, 2013) is a shooting simulator for training police officers which are challenged under special situations to use not only their decision making skills but also their guns. This simulator shows different scenarios on a projection surface, and users have to react to such situations by sometimes firing their real weapons with rubber bullets. Noptel 2000 (Supply, 2013) is a training system for army forces that provides two modes of operation: interior training for basic shooting and exterior training for combat situations. An interesting aspect of this simulator is the use of a small device that clearly shows targets by reflecting a laser light on them. Finally, a live fire,multiscreen simulator is used by the Cleveland and Durham Police Authority in the UK (Systems, 2013) in order to expose police officers to a controlled scenario where they can practice their shooting skills.

Most of these simulators train their subjects in the use of hand weapons of both short and long range by showing specific scenarios where subjects have to decide when and where to shoot at. These simulators place shooters in a closed virtual space with limited movements. Most subjects in these simulators are stationary. These characteristics differ from our proposal, in which we want to recreate the conditions that shooters have in a moving boat that follows a river course where targets could be moving.

3 DEVELOPMENT

We present in this section details of our shooting simulator for fast boats. We divide this description in four main components, the motion platform that simulates a boat's movements, the shooter interface that simulates a gun and measures shooter actions, the reporting system that gives information of a shooter's performance, and the simulation software that receives a user's movements, defines environment stimuli, and produces visual and auditory feedback.

3.1 Motion Platform

The motion platform is composed by two main elements: a shooter interaction space and a robot that produces simulated boat's movements. These elements are depicted in Figure 1

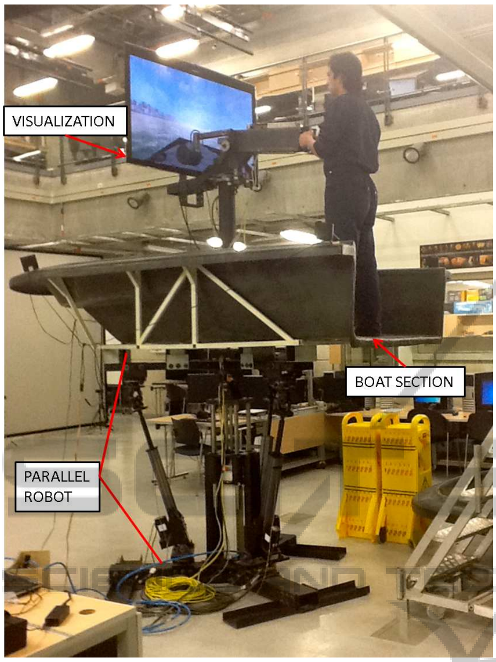

Figure 1: General Elements of Our Simulator.

We wanted feedback from our simulator to be as real as possible. For that reason, we designed a real replica of the gun's handle so users can hold it in the same way as the real one. We also allow the type of movements that such a gun has in a real boat, so it can be aimed to both banks of a river. Finally, a section of a real boat is replicated in order to allow trainees to have the same range of movements than in a real boat. All these elements are mounted on top of a robot that is capable of lifting all these equipment plus a trainee. We concentrate our description here in this robot, while the shooter interface is described in the next section.

Typical training maneuvers in a real scenario were measured with a 3-axis accelerometer placed in the shooter position. Measured accelerations were separated into two groups, depending on their frequency, and we classify them as low frequency accelerations and high frequency accelerations.

Low frequency accelerations are below 0.8 Hz and can be simulated by tilting our boat replica. In this way, low frequency accelerations in the longitudinal axis are simulated by pitch rotations, while lateral accelerations are simulated as roll rotations. Low frequency vertical accelerations such as the ones we measured would require a larger robot and space than the one we have available and in this case they were not cost effective, so we decided to leave them out of the scope of our robot's range of movements.

High frequency accelerations were defined as those between 0.8 and 3 Hz. These accelerations are simulated directly as linear movements over each axis. The measurement of yaw rotation was not considered as the boat didn't present representative dynamics on this type of movements.

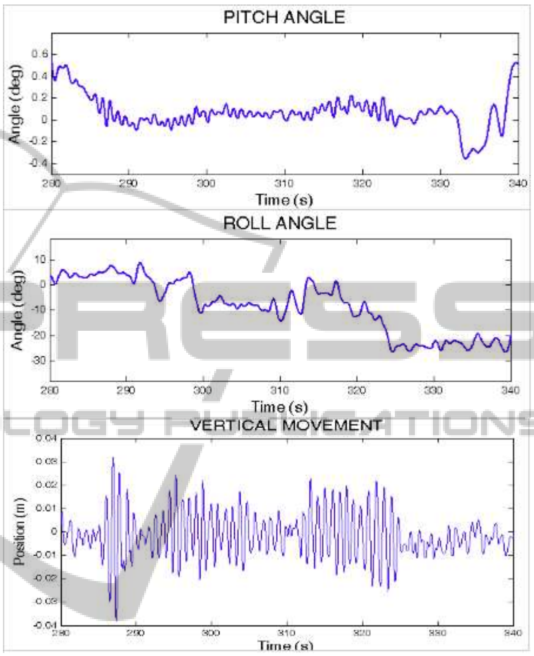

Figure 2: Typical Simulator Movements.

An analysis of the processed measurements led to a low requirement of longitudinal and lateral linear movements, as they presented high frequency, low amplitude displacements. With this result, it was determined that a 3DOF robot with roll, pitch and a vertical displacement could produce the required simulator dynamics for training. Figure 2 shows typical movements of our simulator in terms of angle variations during time. These movements are produced from filtering and processing the real boat signal.

Our 3DOF parallel robot is a variation of the robot presented in (Fattah and Kasaei, 2000), in which we add a central weight compensator. In terms of the standard robot nomenclature that describes the type of joints that fully constraint a robot's dynamics (Gao et al., 2002), we designed a 3 UPS + PU robot. The 3 UPS stands for 3 limbs, each consisting of a sequence of three joints: Universal (U) on the bottom, followed by a prismatic active joint (P), and a spherical joint at the top end (S). The PU stands for the central passive weight compensator. It has a prismatic joint to the ground (P) fixed to a universal joint at the top (U). A linear pneumatic actuator keeps a constant vertical force at the central prismatic joint, by means of a pneumatic circuit.

The geometry of this robot was optimized according to the simulation requirements, in order to minimize the actuator's nominal force requirements, while maintaining an adequate workspace and motion capabilities of the simulator. The optimization process and results are presented in (Blanco and Rodríguez, 2010a), (Blanco and Rodríguez, 2012), and (Blanco and Rodríguez, 2010b).

3.2 Shooter Interface

The shooter interface consists of the physical elements that surround a shooter, the hardware and software that monitor the gun's orientation and trigger readings, and the middleware that gives this information to the simulation process.

As we showed in Figure 1, on top of our 3DOF robot we mount a replica of a boat's bow, with a replica of a large, moving weapon. The weapon's tip is replaced by a 52 inches screen that gives a trainee a first person perspective of the most relevant part of the virtual scenario in front of him. The weapon has yaw and pitch rotations, although the screen does not follow pitch rotations, since shooters are mostly interested in the river bank in front and therefore pitch rotations are not highly required. Figure 3 illustrates how the weapon is connected to the simulation software. The weapon's instrumentation gives us the angle in which we have to show the virtual scenario, so the visualization is synchronized to user's movements. The weapon's movement ranges are approximately 180 degrees on the horizontal plane (yaw) and 90 degrees on the vertical one (pitch), which correspond to the ones that the actual weapon has. In order to assure a fast and accurately way to capture the weapon's movements, two heavy duty incremental encoders (Stegamnn, 2012) were used. Each one has 1000 pulses per revolution and an output frequency of 100 KHz. Additionally, the weapon has three pulse buttons; one is used to capture the fire event, and the other two capture the sequence of tasks when ammunition is loaded. Since encoders are incremental and therefore weapon's orientation should be computed in terms of a previous one, we defined a callibration process that a user should perform in order to define the initial position and orientation. Both encoders and pulse buttons are connected to a Wiring card (Barragan, 2013). Such a card contains a program writen in the Processing Programming Language (Fry and Reas, 2013) that receives signals from attached sensors —encoders and pulse buttons in our case— and produces an optimized output stream, which is transmitted to our simulator in our server.

On the server side we use VRPN (Taylor et al., 2001) to handle the weapon's data stream. We designed a VRPN custom device and protocol in order to minimize the data stream coming from Wiring, which is then translated to the standard VRPN data stream types, in this case quaternions. We first used a basic, text-based protocol between Wiring and our server which had delays of up to 400ms. Such delays generated uncomfortable jitter on our visualization if our weapon was abruptly moved. We avoided such a big delay by writing a binary based protocol with a smaller buffer size. In this way we could create packets of 8 bytes instead of 128, and reduce our overall delay for this channel up to 30ms. We also take advantage of VRPN's support for multiple clients, in order to provide a simple debugging channel from third party applications. Finally, a VRPN remote client is built inside our simulator software through a dynamic library, in order to read the required information and facilitate updates.

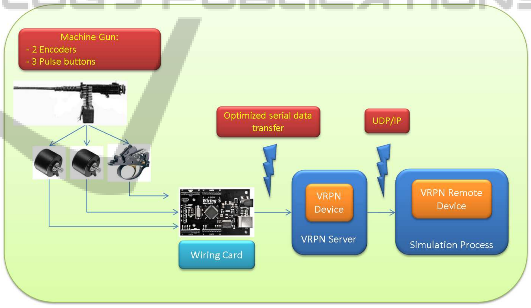

Figure 3: Weapon Connection to our Simulator.

Modifications in orientation and events from firing and reloading the weapon are recorded with a timestamp in a relational database. This information is used by the report system in order to summarize a user's session.

3.3 Simulation Software

Our simulation software loads the simulation scenario, defines the physical platform's movements, reads data from the shooter interface, computes the state changes in the simulation given by user's input or environmental changes, and produces visual and auditory feedback. We use The Unity Game Engine (Unity, 2013) as the software platform for our simulator, given its ease of development, rich functionality, and current widespread use.

A general overview of this component is showed in Figure 4 and it has a similar structure to the one proposed by (Smith, 1998).

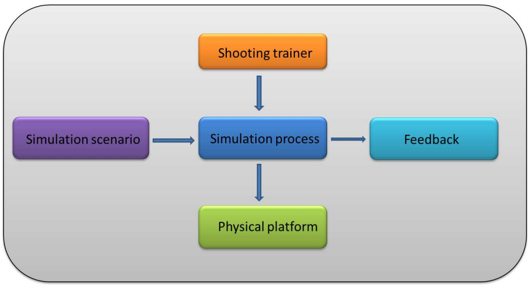

Figure 4: General Components in our Simulator.

3.3.1 Simulation Scenario

We define several simulation scenarios that allow trainees to test their skills under several conditions. Figure 5 shows some terrains we have developed, which offer different conditions about the location of friends and foes. Figure 6 presents several atmospheric conditions (i.e dry or rainy weather) and daylight intensity (i.e. sunny, sunset, cloudy, or night).

Figure 5: Terrain Shapes.

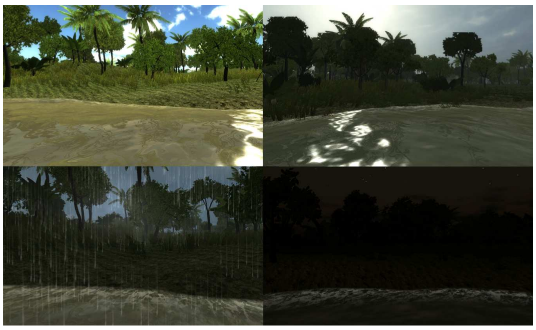

Figure 6: Different Environment Types.

Another important condition in our scenarios is the difficulty of the training session. We define three levels of difficulty:

- Training: It contains few static enemies that do not fire back, and it is used to introduce a trainee to a new terrain.
- Practice: It contains static enemies that fire back, and it is used to introduce a trainee to a combat situation.
- Combat: It contains moving enemies spread all over a terrain, and it is used to simulate a full combat scenario.

The scenario information, i.e. terrain maps, boat trajectories, and enemy locations, is created with the Unity built-in editor. Boat trajectories are computed by an external program that precomputes the specific boat dynamics and outputs a text file, which is used as input by the simulator and by the motion platform.

3.3.2 Motion Platform Software

Movements in our motion platform are predefined, since our simulator concentrates on the shooter behavior. A non realtime Matlab program creates a text file that contains trajectory data for both the motion platform and the visualization. An embedded VRPN client receives a stream with the trajectory data and moves the motion platform accordingly.

3.3.3 Shooting Interface Software

The shooting interface software is a VRPN server on its own computer that receives weapon's button events (i.e. loading and shooting) and movements (i.e. yaw and pitch) and sends such information to any interested client.

3.3.4 Simulation Process and Feedback

There are several tasks that are performed by the simulator at each step. First, it reads the next position from the boat's predefined trajectory and sends it to the motion platform. Then, it reads the weapon's events and updates its virtual state. If the weapon's trigger has been fired, it starts a bullet's simulation. At every step we compute ray - object collisions and find out possible targets to update.

Although there is visual feedback related to bullets, specially during night, the most significant feedback we generate in any scenario besides visualization changes is sound. Wind and boat motors are part of the constant background in this system, and it is modelled as stereo feedback. Weapon's firing sounds are located in the 3D space, and they are played both when enemies shoot and when a trainee fires his weapon. This last sound is the loudest in the system, which almost silences all other sounds. Visual feedback for bullets simulates tracer bullets, special bullets that are used by a shooter to correct his aim. They are simulated as smaller bullets that are more visible and do not produce collisions.

Our simulator keeps track of dynamic objects in a scene and records the creation and impact conditions of every bullet. It also keeps track of the weapon's events. This information is gathered for an off-line process that creates reports.

3.4 Reporting System

We develop a reporting system that collects the user actions and events generated by the simulation and subsequently compiles them into a report that shows the shooter's performance.

All data collected about the events of the weapon, timing and bullets whereabouts are dumped into a relational database at the end of each session. In this way, we register the following events:

- Readings from the weapon (changes in orientation and pressing or releasing buttons)
- Impact of a bullet over a relevant object
- The event of running out of ammo
- The start and end time of a simulation

Since the above events are too granular to determine the shooter's performance, we define tasks. Each task is a set of events that the user has to accomplish at some point along the execution time-line of the the training session. These are non-overlapping groups of events, as show in Figure 7.

Figure 7: Data Hierarchy Between Tasks and Events.

Examples of tasks can be:

- Load the weapon
- Aim and shoot a target
- Counterattack an ambush
- Deter enemies by shooting over a specific area

A task is defined by a collection of events that the shooter has to accomplish, a collection of events that could interfere with the normal termination, and the metric or metrics used to measure it. A back end system holds all these data and provides services to generate reports.

For instance, to accomplish the task of aim and shoot a target, the shooter has to change the flank of attack and shoot the enemies in a correct timing. If ammo runs out before hitting a target, a trainee must perform a weapon reload and resume shooting. If the training session ends without reloading, this task has failed. Two metrics are used to evaluate performance of this task: elapsed time and number of bullets used. Both of them are measured from the moment the enemy appears to the moment it gets the first hit.

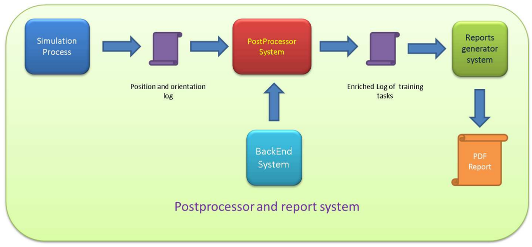

Figure 8: Postprocessing and Reporting System.

Figure 8 shows the process of generating performance reports.

The reporting system can generate a PDF file with performance information about a session. Some of the metrics we report are the following: number of bullets fired to valid targets, bullets impacted areas nearby a target, and rate of firing.

4 PRELIMINARY RESULTS

We conducted a test session with expert users in order to evaluate the current implementation of our simulator. Four experts in combat operations with ages between 28 and 34 and experience from 3 to 7 years in several rivers of our Country were invited to our lab. None of them had previous experience with this type of simulators. They received a short introduction about our simulator, its capabilities and limitations, and some instructions for safe use. Each subject had to go on top of the motion platform, hold the weapon at all times, identify the location of enemies that were shooting at the boat and fire at will. They had to pay attention to the stimuli from the motion platform, visual and auditory cues in order to identify enemies, and they were asked to move in order to cover their flanks.

Sessions last 7 minutes, the time for our prerecorded movements. We used the normal river terrain with rainy conditions and we defined a rough situation for river movements. At the end of a session subjects filled a questionnaire with questions related to the simulator and to the relationship between this experience and a real situation.

Figure 9 shows the average answers to the questions related to the simulator, in a scale of 1 (worst) to five (best), as follows (Letters in questions correspond to the original order in the questionnaire):

1. A. Do you consider that this simulator can improve the skills of an shooter?
2. B. How easy is to shoot in this simulator?
3. D. How easy is to reload ammo in this simulator?

4. F. How easy is to identify where the enemies are shooting from?
5. H. How easy is to move in this motion platform?
6. J. How fast is the response in this simulator?

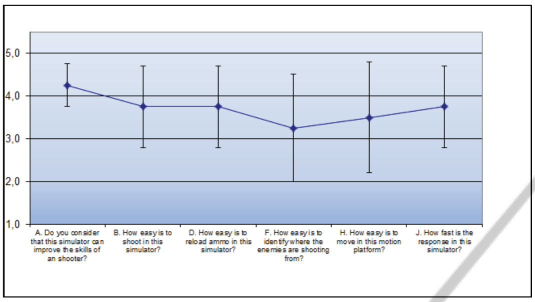

Figure 9: Questions Related to the Simulator.

In general, subjects see value on the use of a simulator like this for training (A). They found the weapon easy to use. Although simulator's response received a fair mark, it seems it can be improved in future versions. In the same way, enemy identification can be improved, as well as movements in the motion platform. In this regard, it is possible to think that enemy identification is also difficult in the real scenario, and in some way we correlate to that fact. They also mentioned an exagerated movement as a problem in the motion platform, and the lack of correlation between boat movements and enemy fire, which is part of a real situation.

We asked subjects to answer the following questions related to the similarities between our simulator and a real situation:

1. C. How similar is this shooting experience to the real situation?
2. E. How similar is to reload ammo in this simulator?
3. G. How similar is the visualization of enemies in this simulator?
4. I. How similar are your movements in this motion platform to the ones in real life?

Figure 10 shows answers for these questions. We can see we have to improve our similarity to the real weapon. In this sense, our mockup is not an exact copy of the real gun, therefore experts noticed that fact. We also have to improve on the motion platform movements. In this regard, since our movements are precomputed and for this test somewhat exagerated, experts felt they didn't correlate to the situation at hand. However, experts believe the visualization was good enough for this situation, which encourages us in our selection.

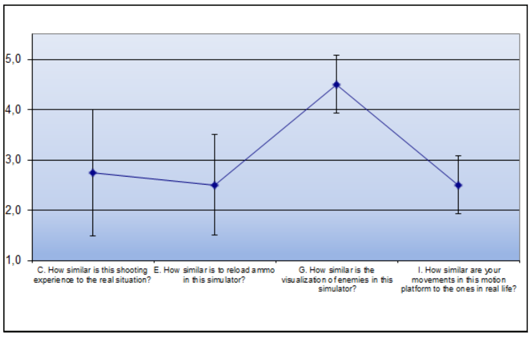

Figure 10: Questions Related to the Correlation Between Our Simulator and a Real Situation.

5 CONCLUSIONS AND FUTURE WORK

We present a simulator for shooting from fluvial boats, which consists of a 3DOF motion platform, a close to reality weapon mockup, a large screen for visualization, and speakers for sound effects. Movements for the motion platform are derived from the analysis of measurements in a real boat. We include a reporting system that shows a shooter's performance in terms of number of bullets fired to valid targets, bullets impacted areas nearby a target, and rate of firing. Unity, VRPN, and a Wiring embedded system were used as the basic infrastructure for our simulator. Our simulator has been tested by users with a considerable experience in fluvial combat. Their general perception is that our system is suitable for enhancing the shooting training program of shooters.

In the future we plan to improve on some features pointed out by experts. We also plan to include more people in the simulation, specially a boat driver.

ACKNOWLEDGEMENTS

The authors would like to thank to the Division of Research and Investigation of the Colombian Navy for their financial support, providing the personnel assistance and advice to carry out the design, construct and tests of the simulator. Also, we appreciate the assistance of the professors of the the departments of Mechanical Engineering and Systems and Computer Engineering, and administrative personnel of Colivri laboratory at Los Andes University in Bogotá Colombia.

REFERENCES

Avizzano, C., Tripicchio, P., Joale, L., and Bergamasco, M. (2010). Design of a motion based sailing simulator. In *RO-MAN, 2010 IEEE*, pages 1–7.

Barragan, H. (2013). Wiring open project, http://wiring.org.co.

Blanco, J. and Rodríguez, C. (2010a). Configuration Optimization of a Boat Simulation Platform for a Mobile User. In *Proceedings of the ASME 2010 International Mechanical Engineering Congress and Exposition*, IMECE 2010.Vancouver.

Blanco, J. and Rodríguez, C. (2010b). Optimized development of a simulation platform for fluvial combat. Master's thesis, Mechanical Engineering, Uniandes, Bogotá, Colombia.

Blanco, J. and Rodríguez, C. (2012). Optimal Design of a River Boat Simulator. DETC2012-70456. In *Proceedings of the ASME 2012 International Design Engineering Technical Conferences (IDETC) and Computers and Information in Engineering Conference (CIE)*, IDETC/CIE 2012. Chicago, IL.

Bruzzone, A., Longo, F., Nicoletti, L., and Diaz, R. (2012). Traffic controllers and ships pilots training in marine ports environments. In *Proceedings of the 2012 Symposium on Emerging Applications of M&S in Industry and Academia Symposium*, EAIA '12, pages 16:1–16:8, San Diego, CA, USA. Society for Computer Simulation International.

Christensen, K. (2000). The long range shooting simulation., http://www.shooterready.com.

Dahlberg, E. (2012). Eurosimulator, http://www.hlberg.dk/eurosimulator/shooting_uk.html.

Fattah, A. and Kasaei, G. (2000). Kinematics and dynamics of a parallel manipulator with a new architecture. *Robotica*, 18(5):535–543.

Fry, B. and Reas, C. (2013). Processing, http://processing.org.

Gao, F., Li, W., Zhao, X., Jin, Z., and Zhao, H. (2002). New kinematic structures for 2-, 3-, 4-, and 5-dof parallel manipulator designs. *Mechanism and Machine Theory*, 37(11):1395 – 1411.

Lvovskiy, M. (2005). Training simulator for sharp shooting. Patent No. 6942486 USA.

McKenna, I. H. and Little, S. (2000). Military concept development: developing tactics using low cost, accessible simulations. In *Proceedings of the 32nd conference on Winter simulation*, WSC '00, pages 991–1000, San Diego, CA, USA. Society for Computer Simulation International.

Mertens, S. (1993). The corps battle simulation for military training. In *Proceedings of the 25th conference on Winter simulation*, WSC '93, pages 1053–1056, New York, NY, USA. ACM.

Page, E. and Smith, R. (1998). Introduction to military training simulation: a guide for discrete event simulationists. In *Simulation Conference Proceedings, 1998. Winter*, volume 1, pages 53–60 vol.1.

Perez, T., Smogeli, Ø. N., Fossen, T. I., and Sørensen, A. J. (2006). An Overview of the Marine Systems Simulator (MSS): A Simulink Toolbox for Marine Control Systems. *Modeling, Identification and Control*, 27(4):259–275.

Seet, G., Lau, M., Low, E., and Cheng, P. (2001). A unified pilot training and control system for underwater robotic vehicles (urv). *Journal of Intelligent and Robotic Systems*, 32(3):279–290.

Smith, R. (1998). Essential techniques for military modeling and simulation. In *Simulation Conference Proceedings, 1998. Winter*, volume 1, pages 805–812 vol.1.

Stegamnn, I. (2012). Heavy duty incremental encoder, http://www.stegmann.com/product/incremental.

Stephen P. Rosa, Motti Shechter, J. C. (2003). Firearm laser training system and method employing an actuable target assembly. Patent No. 6575753 USA.

Supply, C. S. (2013). Champion shooters supply, llc, http://championshooters.com/noptel/index.htm.

Systems, A. I. (2013). Indoor live-fire multi-screen simulator, http://www.ais-solutions.co.uk/portfolio/cleveland-durham.php.

Takeda, H., Yamasaki, M., Moriya, T., Minakawa, T., Beniyama, F., and Koike, T. (1999). A video-based virtual reality system. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '99, pages 19–25, New York, NY, USA. ACM.

Taylor, II, R. M., Hudson, T. C., Seeger, A., Weber, H., Juliano, J., and Helser, A. T. (2001). Vrpn: a device-independent, network-transparent vr peripheral system. In *Proceedings of the ACM symposium on Virtual reality software and technology*, VRST '01, pages 55–61, New York, NY, USA. ACM.

TROJAN, A. (2012). Shotpro 2000., http://www.trojansim.com.

Unity (2013). Unity technologies, http://unity3d.com.

Wang, Z., Hu, Y.-M., and Xie, F. (1996). Optical fiber simulator for shooting and aiming practices. In Bennett, K. D., Kim, B. Y., and Liao, Y., editors, *Fiber Optic Sensors V*, volume 2895 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 475–478.

Yang, Y.-F. (2008). Evaluation of steering simulating for china coast guard ship in narrow waterway. In *ITS Telecommunications, 2008. ITST 2008. 8th International Conference on*, pages 95–98.

Young, D. (2013). Canadian academy of practical shooting., http://www.caps-inc.com/c-4-company.aspx.