A Generic Virtual Reality Flight Simulator

Turgay Aslandere*, Daniel Dreyer†, Frieder Pankratz⋆,Renê Schubotz†

* German Aerospace Center (DLR) † Airbus Group Innovations
38106 Brunswick 81663 Munich
E-Mail: Turgay.Aslandere@dlr.de

⋆ Technische Universität München (TUM)
D-85748, Munich

Abstract: Flight simulators are used for different purposes, such as pilot training, aircraft design and development. Full-scale flight simulators have high costs and dependency on aircraft type due to hardware constraints. Hence, virtual reality flight simulators are designed. On the other hand, they are generally created only for specific applications, such as helicopter simulators. As a result, these tools can hardly be used as a generic tool which can work with various aircraft simulations. Although, there are certain generic virtual reality applications that can be used for virtual prototyping and ergonomics, they lack realistic flight simulation and environment. In this paper, we present a generic aerospace application which brings a solution to these problems. The architecture of the application is described and a calibration method which, makes the application independent of the physical mock-up and the flight simulator compatible with different aircraft types, is presented. The preliminary results of the first prototype are given as the generic virtual reality flight simulator is used by the aerospace industry for research and development purposes.

Keywords: System Architecture and Intelligent environments, 3D Interaction Devices and Interaction Techniques, Flight Simulator

1 Introduction

Full flight simulators have drawbacks such as high costs since they have a fixed installation because of the hardware constraints. Several low cost flight simulators have been employed such as [LHH07], so far they are dependent on the aircraft type. The adaption of a mock-up to aircraft types is expensive and difficult with this kind of flight simulators. It is necessary to create a different replica cockpit for each aircraft type to use this kind of simulators as a generic flight simulator. Hence, an alternative flight simulator concept compatible with all kind of aircraft is needed.

In the industry, various virtual reality tools are used for virtual prototyping and ergonomy analysis. The tools which are created for virtual prototyping and evaluation of the ergonomics as Lijing et al. suggested [LWX+09] can be used with different aircraft types without a cockpit replica. However, these tools only provide an internal view of the aircraft and lack a flight environment and simulation in general. Therefore, the pilots who use the virtual prototyping software, are not able to see the world including airports, cities as they look through the aircraft window. Therefore, their evaluation of the aircraft is not satisfying with virtual prototyping tools. As a result, a generic virtual reality flight simulator which is compatible with all kind of aircraft and has complete flight physics and environment is required. The flight simulator can also be used for pilot training purposes along with the aircraft development and testing purposes.

In this paper, we create a generic Virtual Reality Flight Simulator (VRFS). The most central components of the VRFS are discussed introducing a modular approach. Our approach makes the VRFS usable with various scenarios from virtual reality flight training to testing cockpit ergonomics. This approach also makes the VRFS capable of working with various aircraft. However, compatibility with different aircraft requires a perfect alignment of virtual and real world. Therefore, a calibration method is described. The preliminary results with regard to the interaction in the VRFS are discussed.

Our work has two main contributions. First, we describe how to implement a generic VRFS exploiting existing tools, such as desktop flight simulators, using an extensible and adaptable system architecture. Second, we show that the system brings new challenges with regard to interaction techniques. We evaluate interaction with the virtual buttons.

Virtual reality applications have been developed to overcome the high costs of the full-scale flight simulators and dependency on aircraft type. Yavrucuk et al. present a virtual reality helicopter simulator [YKT11]. This simulator is a low cost solution. On the other hand, the application is highly based on the Flight Gear [Per04] software and does not present a generic software architecture. Unlike [YKT11], an interactive real time application with a modular architecture which is not dependent on the virtual environment is demonstrated by Wolff et al. [WPG11]. It is designed for satellite servicing and can also be used for astronaut training and maintenance [WPG11]. Courter et al. proposed an extensible aerospace tool which aims more at walk-through scenarios [CSN$^+$10]. A walk-through property is not necessarily needed for flight simulators since pilots usually control the aircraft while they are sitting on a cockpit seat.

2 System Overview

VR (Virtual Reality) applications are complex systems which consist of various components. Zachmann describes the general architecture and basic software components of VR systems [Zac00]. The VRFS also contains basic software components which will be called core modules. These are virtual environment, communication and object handler module. The virtual environment module is responsible for visualization and flight simulation. The communication module provides data exchange with other modules and the object handler is the module that controls the human computer interaction. These are the essential components which are required to run the VRFS with the hardware.

The hardware of VRFS can vary as the system is generic and independent of specific input and output devices. The modules of the VRFS run in high performance workstations with multi-core processors and GPUs. The first prototype consists of hardware components depicted in Figure 1. An optical tracking system (ART Optical Tracking System [ART12]) for finger and head tracking, a joystick, throttle and pedals with other desktop input devices such as keyboard and mouse are employed as input devices. An optical tracking system is chosen for the first prototype, since it provides precise tracking with minimal latency. Besides, it does not have any cables for connection which allows users to move more easily. It is not only employed for finger and head tracking but also to track the seating buck. Actually, the seating buck is a simplified mock-up with low cost constraints. It includes a cockpit seat which is surrounded by the basic flight hardware, such as joystick and pedals in the first prototype. Most of the interaction is achieved by using virtual hand metaphor [BKLP04] with virtual objects without a more complex physical mock-up. This is necessary to reduce the hardware used in the virtual reality flight simulator and to make it capable of working with all kinds of aircraft including airplanes and helicopters. On the other hand, the interaction with virtual objects is challenging and has some drawbacks such as missing haptic feed-back and real time constraints.

A Head Mounted Display (Nvisor SX60) and stereo headphones are used as output devices. The first prototype is also usable with projection based output. However, projection based displays have active view problem where head tracking cannot be used for more than one person, whereas HMDs do not have this problem since each user has to wear an independent HMD with a head tracking target. Therefore, the projection based systems are supported for future development but they are not used for the first prototype of VRFS. Stereoscopic views which enable binocular oculomotor cues and increase the feeling of presence in the synthetic 3D world of the VRFS are provided for both HMDs and projection based output devices.

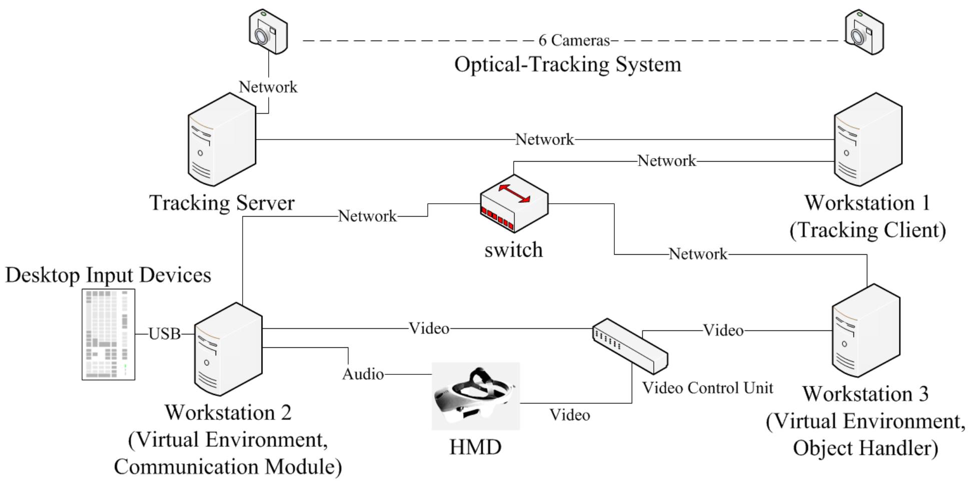


Figure 1: A possible configuration for data flow through the hardware components of the VRFS. In the first prototype, the virtual environment does not allow users to create more than one view port. Therefore, two instances of the virtual environment module are necessary for stereoscopic views.

3 System Architecture

The VRFS is a distributed system as the modules are generally located on different computing environments due to requirements of high computing power. Also, network connected computers are needed where multiple users are involved. The communication among the modules is provided by the Local Area Network (LAN). The data flow through the modules would be different where more users are involved or additional modules are executed.

Each module can be located on a different workstation (Figure 1) where high performance is needed. However, end-to-end latency of the distributed system increases with this configuration due to network latency.

3.1 Communication Module

The communication module is the core of the VRFS. Besides the communication, it is also responsible for spatial transformations, calibration of the system and sensor fusion. These tasks are achieved by the communication module to isolate modules of the VRFS from each other. In this manner, additional modules should be only connected to the communication module for further development.

The communication module exploits the Ubitrack framework which makes use of the SRG (Spatial Relationship Graph) concept to setup a dataflow graph. SRG is a graph where the nodes represent coordinate systems or orientation free points on real or virtual objects. The edges of SRG represent transformations between the coordinate systems [EHP+08]. The spatial relationship graph specifies all relevant properties of the tracking setup. SRGs enable the automated analysis and synthesis of complex tracking scenarios. Therefore, they improve the usability of VRFS and reduce the sources of potential error related to the tracking setup. A more detailed description of Ubitrack and SRG concept can be found in [PHW+11].

Ubitrack can dynamically adapt to changes in the sensor infrastructure. In other words, it incorporates all hardware devices and software components that deliver raw spatial data. The generic sensor API of Ubitrack provides an abstraction from the specific hardware devices while providing sufficient information to make use of existing tracking devices [NWB04]. Therefore, exploiting this capability makes the VRFS independent of the virtual reality input devices. It also offers a tool called Trackman with an editor for graphical planning and analysis of the tracking setup. This editor makes the tracking setup of VRFS accessible by engineers and designers who have no programming experience.

3.2 Virtual Environment Module

The virtual environment module is responsible for the flight simulator engine and visualization of virtual scenarios. The virtual environment is internally represented as a scene graph. It includes virtual world objects and their behavior. Accordingly, the objects such as airports and aircraft are a part of the virtual environment. These objects are expected to look realistic and act according to the law of physics. The virtual environment of the

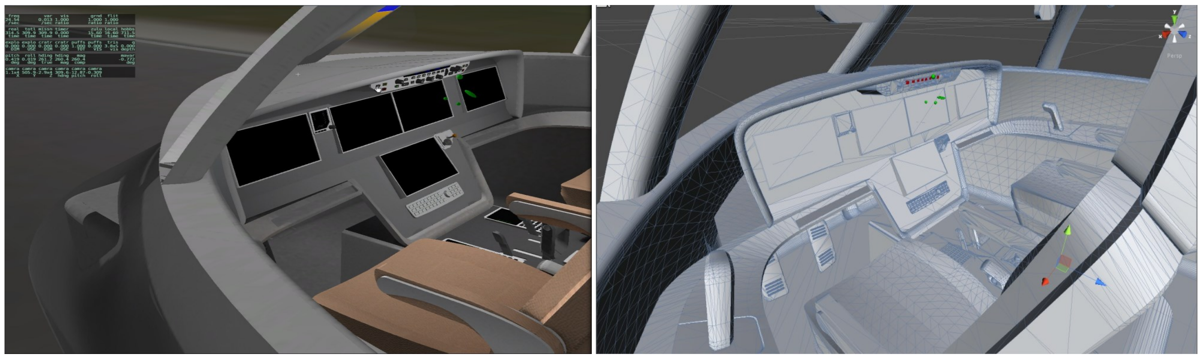

a) Virtual Environment Module (X-Plane)
b) Object Handler Module (Unity3D)

Figure 2: Aircraft geometry in the virtual environment and object handler module.

first prototype can be seen in Figure 2 a) where all properties of the virtual environment such as textures and lighting are assigned. The virtual environment module has a control and interaction sub-module which receives the data from the object handler module through the communication module and assigns it to the virtual world. The flight simulator engine is also generally a sub-module of the virtual environment module. Its responsibility is the simulation flight physics and conditions. The sub-modules of the virtual environment can be plugins or classes depending on the virtual environment that is employed.

The virtual environment module is generally a desktop flight simulator such as X-Plane[xpl12] , Flight-Gear[Per04]. In this case, the control and interaction plugin bridges the components of VRFS with the desktop flight simulator. On the other hand, in-house-developed software which consists of flight simulator and rendering engine and might be preferred for commercial purposes. We use X-Plane 9 in the first prototype. Actually, the full flight simulators (which imitate whole aircraft with motion systems) exploit the similar software. X-Plane is a commercial engineering tool that is used to predict performance of aircraft with high accuracy. It also has Federal Aviation Administration (FAA) certification to be used for pilot training with valid hardware. It provides the flight simulator engine within the virtual environment in the VRFS.

3.3 Object Handler Module

The object handler module is responsible for the human-computer interaction and the physics in the aircraft interiors. In other words, what happens in the cockpit or aircraft cabin, is under control of the object handler module. A physics simulation is necessary to simulate correct behavior of virtual objects inside of the cockpit.

The object handler module receives input from the communication module and produces collision data using the collision detection module which is usually embedded into the object handler. The object handler module is required to have an editor that can be used for interaction design. This editor employs the same CAD data of the aircraft as the virtual environment module. However, this geometry does not have textures and colors since it is only used for collision detection and physics (Figure 2 b)).

The object handler was independent of any other software packages during the design of the VRFS. However, the advantages of commercial software, such as an editor to configure scene graph for the interaction management push us to make the architecture of the object handler compatible with them. In the first prototype of VRFS, we employed Unity3d. It is a commercial software package which offers an editor that can be used for interaction design. Unity3d contains NVIDIA PhysX built-in physics engine that is used to simulate the physics inside the cockpit.

The object handler module makes the interaction independent of the aircraft geometry. We implement predefined aircraft objects (particular instance of a class) without any geometry to achieve this. The CAD data of aircraft must be assigned to predefined objects in the scene graph by the designers and engineers who are going to test their virtual setup in the VRFS. This process does not require any programming knowledge, it is achieved by dragging a CAD object onto the predefined object by the mouse cursor using the graphical user interface.

3.4 Data Flow

The data flow among the modules of the first prototype is shown in Figure 3. In this figure, the input device client collects data from tracking devices, sends it to the communication module, which then computes required coordinate transformations for the tracking data. After that, the communication module sends the transformed tracking data to the object handler and virtual environment module. The computation of the coordinate transformations is a necessity since coordinate systems in the object handler and the virtual environment often have a different orientation and origin. The object handler processes transformed tracking data to compute collisions with the collision detection module. The computed collision data is sent back to the communication module by the object handler. The virtual environment module receives the processed data from the object handler and provides the visual and audio output to the hardware. The processed data includes collision data, transformed tracking data and additional data such as aircraft ID.

The virtual environment module also directly receives sensor data from desktop input devices and assigns it to flight simulator engine. Afterwards, the flight data including location, speed, altitude of the aircraft is synchronized with the main virtual environment, if there is more than one instance of virtual environment module. The sensor data from desktop input devices could be also distributed via communication module. On the other hand, this would increase latency. The latency of the aircraft controls such as a joystick and pedals can result in incorrect navigation of the aircraft.

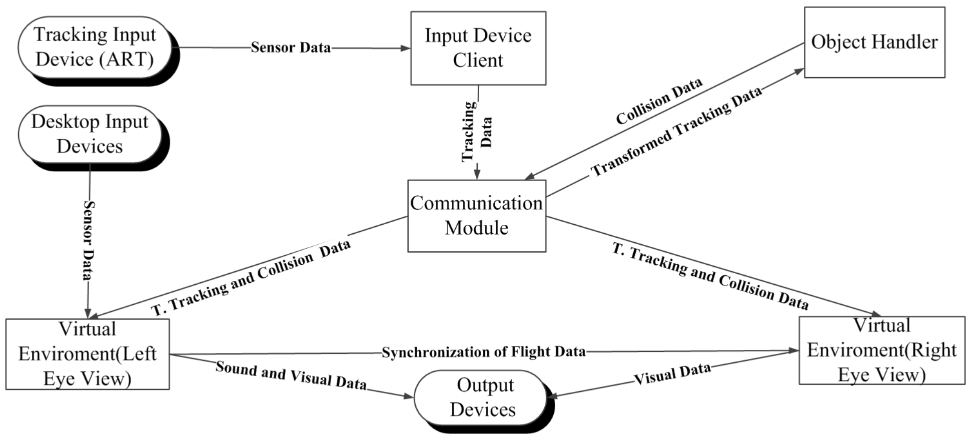

Tracking Input Device (ART)
Sensor Data
Input Device Client
Object Handler
Desktop Input Devices
Tracking Data
Collision Data
Transformed Tracking Data
Communication Module
Sensor Data
T. Tracking and Collision Data
T. Tracking and Collision Data
Virtual Enviroment(Left Eye View)
Synchronization of Flight Data
Virtual Enviroment(Right Eye View)
Sound and Visual Data
Output Devices
Visual Data

Figure 3: A possible configuration for data flow through the software components of VRFS. This diagram will differ as new modules are added. The flight data is synchronized without using communication module to make the VRFS independent of the virtual environment.

4 Calibration and Validation of the System

Virtual seating bucks are commonly used for different purposes such as the investigation of ergonomics during product development [SF08]. However, they require careful calibration to ensure alignment between the virtual and physical world the user interacts with. In this section, we describe our calibration method: "Virtual Seating Buck Calibration (VSBC)" for the first prototype of VRFS.

4.1 Virtual Seating Buck Calibration (VSBC)

The cockpit objects in both virtual and real world must precisely match with each other. Otherwise it would be difficult to provide a feeling of presence in the VRFS. And this can have a negative effect on interaction. For this reason, the VSBC is provided to align the virtual environment and real world for exploiting the seating buck. The VSBC is performed in three steps. First of all, the alignment of the virtual and real objects is performed considering geometry and location. Second, the alignment of coordinate systems of virtual and real world is provided. This assures that real and virtual world objects have the same scale and move in the same orientation. Finally, the display system is calibrated. This provides the correct image of the virtual world from the first person point of view.

The alignment of the real and virtual objects: The geometry and location of the objects must be precisely the same in the virtual environment and real world. Non-functional geometry, colors, material and the rest of the properties related to cosmetics of the physical mock-up do not have to be similar since, HMD blocks the real world. For this process, a cockpit seat is chosen, although an arbitrary seat may be chosen on which users can be located. The cockpit seat is scanned with a calibrated 3-D scanner and point clouds are obtained. Then, the mesh of the seat is created using open source software which provides marching cubes algorithms. The reconstructed seat is modified by the aircraft designers according to the design of the aircraft. The usage of a different aircraft seat for each virtual cockpit can be extremely expensive. Therefore, this process (Figure 4) reduces the costs of physical mock-up. Moreover, different aircraft types are tested constantly in virtual reality laboratories and providing a physical mock-up for each aircraft type is difficult.

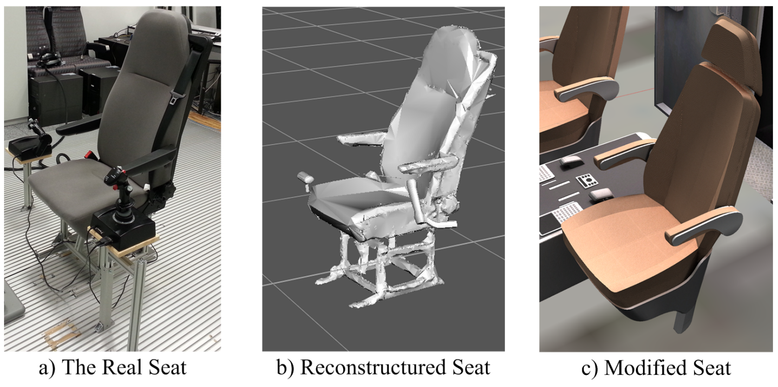


Figure 4: Alignment of the real and virtual seat object.

The alignment of the coordinate systems: The orientation and scaling of the coordinate systems of the virtual environment and real world must be identical or additional transformations must be implemented. A point on the seat in the both virtual environment and real world is chosen for this purpose (Figure 5). A tracking target is located on the point to get the exact position of it in the real world. In the virtual environment, the location of this point (which is represented by virtual marker) is determined by measuring the position of the real tracking target on the real seat. Afterwards, transformations are performed in the communication module to match the position and orientation of the virtual marker and the real tracking target. The transformations are carried out using SRGs. In Figure 5, the virtual and real tracking targets are circled in red and the orientation of the virtual environment and real world coordinate systems are illustrated on the backs of both seats. The position and orientation of the head tracking target is computed relative to the tracking target to make the VRFS independent of the room coordinate system. In other words, regardless of orientation and location of the real seat in the room, the user will be sitting on the virtual seat and looking at the same direction in the virtual environment.

Calibration of the HMD: Uncalibrated HMD systems can cause distorted perspective-related cues and lead to wrong distance judgments. The method which [KTCR09] suggested for calibration of HMDs in virtual reality environments is used in this work. This method is based on the perception of users that they can estimate correctness of the FOV by repeatedly putting the HMD on and off while they are comparing real objects and virtual objects displayed in the HMD. The FOV in the VRFS system is easily determined since all the measurements necessary for the calibration are known including the location of the user and the size of the objects in both virtual and real world.

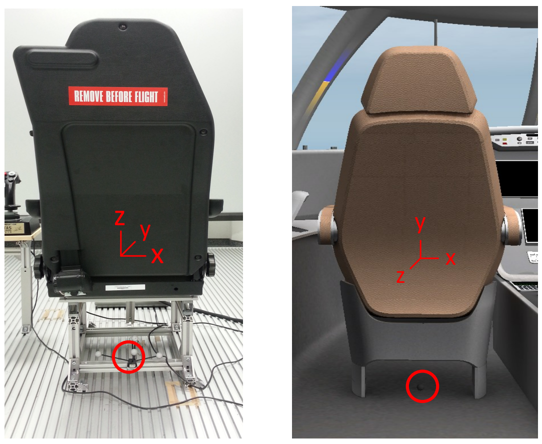
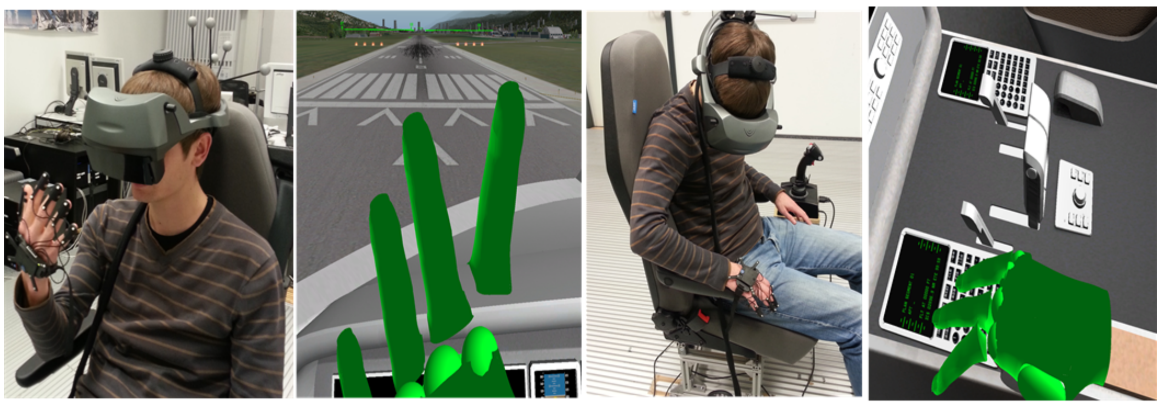


Figure 5: Alignment of the coordinate systems.

4.2 Validation

The validation was performed by four male lab members whose ages were between 25 and 35. The seat model was a proper validation parameter to test, since the similar geometry was used in the real and virtual world. The subjects were asked to touch different points on the virtual seat (Figure 6). None of the subjects reported any mismatches between the seats in the real and virtual world.



a) The Mock-up and the Hardware b) The View of the User c) The Mock-up and the Hardware d) The View of the User

Figure 6: Validation of the Virtual Seating Buck Calibration

5 Preliminary Results

We evaluated the effectiveness of virtual hand-button interaction in the VRFS. The experiments were performed with 16 right handed persons whose ages were between 23 and 44. Two of the participants were female. A brief introduction to the VRFS and test procedure was given to participants before they started the experiments. The aim of the introduction was to make sure that the participants would follow the standard behavior. For example, the participants were supposed to put their hands on their knees after touching a button, and wait for the next instruction. The hand avatar was a scanned hand geometry of a male hand which almost had 50 percentile male hand size (BS EN ISO 7250-1:2010).

The participants touched 7 virtual buttons (The button size: 15x10x10 mm^3) on an FCU (Flight Control Unit) panel with two runs. Each button was pressed once after an instruction. In total, they touched 14 buttons. The order of the buttons that the participants touched, was pseudorandom. Feed-back of the selection is provided visually by changing the color of the virtual hand as haptic feed-back cannot be provided. The color of the hand avatar was red during the collision with a button. Otherwise, it was green. All participants had short breaks between the runs to decrease effect of tiredness. The hits such as touching the same target button twice, touching another buttons while target button was instructed, missing the target button were counted as a miss. When the target button was enabled, it was lit. The results were recorded as a percentage of successful hits. The elapsed time for touching the target button was recorded. Since elapsed time was biased by instructor's latency, this information was not used for measuring the hit rate.

The participants achieved a mean value of 0.77 ($\sigma = 0.19$) hit rate. Some of the participants achieve over 0.90 hit rate. Misses occurred randomly. The participants generally reported that the HMD was heavy or the finger tracking device did not fit them well. One of the participant reported that he was unable to converge stereo images. The missing haptic feed-back was the biggest shortcoming of the VRFS during the experiments. Most of the participants hit a target button more than one time which increased the miss rate during the interaction. Because, they were not aware that they were touching buttons in spite of the visual feed-back and their finger was going through the buttons in the real world. The solution for this miss rate is to use a physical mock-up which gives users more haptic feed-back. On the other hand, adapting the physical mock-up to different cockpits is problematic and expensive. This would also make the VRFS partly dependent on the aircraft type which is an inadmissible property.

It has been observed that the frame rate of the virtual environment is faster than 25 fps during our tests. On the other hand, the frame rate relies on many variables. For example, enabling weather conditions in the virtual environment or increasing the number of the objects in the virtual cockpit for the collision detection would slow down the application. Also, the end-to-end system latency is very dependent on the hardware where modules are located. The latency would increase when additional modules are used for multiple users. Therefore, a detailed latency measurement is not given in this paper. A latency analysis for various configurations of the VRFS and the discussion for the real-time rendering challenges are left as future work.

6 Conclusion

This paper presented an outline of a generic distributed virtual reality application which is aimed to meet the needs of the aerospace industry. The most central components of the modular system were discussed. Furthermore, the advantages of the VRFS over its counterparts, such as its independence from the tracking devices, virtual environment and aircraft type were explained. The calibration and validation of the generic virtual reality application was demonstrated.

The preliminary results show that the VRFS is a promising flight simulator concept in spite of the real time constraint. The VRFS is used as an engineering flight simulator for testing new aircraft concepts at the moment. The virtual hand-button interaction might be sufficient for virtual prototyping but it is not ready for pilot training yet. In the future work, we will evaluate the effect of the hand avatar and the button size on the interaction. Also, instead of the visual feed-back, electrotactile feed-back might be an alternative during the virtual hand-button interaction. Additionally, we will investigate the interaction with other virtual cockpit objects, such as sliders and touch-screens which will be needed for pilot training.

7 Acknowledgements

This work has been supported by Airbus Group.

References

[ART12] ART Advanced Realtime Tracking GmbH. *http://www.ar-tracking.com*, 2012.

[BKLP04] Doug A. Bowman, Ernst Kruijff, Joseph J. LaViola, and Ivan Poupyrev. *3D User Interfaces: Theory and Practice.* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 2004.

[CSN+10] D. Courter, J.P. Springer, C. Neumann, C. Cruz-Neira, and D. Reiners. Beyond desktop point and click: Immersive walkthrough of aerospace structures. In *Aerospace Conference, 2010 IEEE*, pages 1–8, 2010.

[EHP+08] F. Echtler, M. Huber, D. Pustka, P. Keitler, and G. Klinker. Splitting the scene graph - using spatial relationship graphs instead of scene graphs in augmented reality. In *Proceedings of the 3rd International Conference on Computer Graphics Theory and Applications (GRAPP)*, January 2008.

[KTCR09] Scott A. Kuhl, William B. Thompson, and Sarah H. Creem-Regehr. Hmd calibration and its effects on distance judgments. *ACM Trans. Appl. Percept.*, 6(3):19:1–19:20, September 2009.

[LHH07] Zhang Lei, Jiang Hongzhou, and Li Hongren. Pc based high quality and low cost flight simulator. In *Automation and Logistics, 2007 IEEE International Conference on*, pages 1017–1022, 2007.

[LWX+09] Wang Lijing, Xiang Wei, He Xueli, Sun Xiaohui, Yu Jinhai, Zhou Lin, and Sun Gaoyong. The virtual evaluation of the ergonomics layout in aircraft cockpit.

In *Computer-Aided Industrial Design Conceptual Design, 2009. CAID CD 2009. IEEE 10th International Conference on*, pages 1438–1442, 2009.

[NWB04] Joe Newman, Martin Wagner, and Martin Bauer. Ubiquitous tracking for augmented reality. In *Proc. IEEE International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 192–201, Arlington, VA, USA, Nov. 2004.

[Per04] Alexander R. Perry. The flightgear flight simulator. In *Proceedings of the annual conference on USENIX Annual Technical Conference*, ATEC '04, pages 31–31, Berkeley, CA, USA, 2004. USENIX Association.

[PHW$^+$11] D. Pustka, M. Huber, C. Waechter, F. Echtler, P. Keitler, J. Newman, D. Schmalstieg, and Gudrun Klinker. Automatic configuration of pervasive sensor networks for augmented reality. *Pervasive Computing, IEEE*, 10(3):68–79, 2011.

[SF08] H. Salzmann and B. Froehlich. The two-user seating buck: Enabling face-to-face discussions of novel car interface concepts. In *Virtual Reality Conference, 2008. VR '08. IEEE*, pages 75–82, 2008.

[WPG11] R. Wolff, C. Preusche, and A. Gerndt. A modular architecture for an interactive real-time simulation and training environment for satellite on-orbit servicing. In *Distributed Simulation and Real Time Applications (DS-RT), 2011 IEEE/ACM 15th International Symposium on*, pages 72–80, 2011.

[xpl12] *Laminar Research. X-Plane Manual*, 2012.

[YKT11] I. Yavrucuk, E. Kubali, and O. Tarimci. A low cost flight simulator using virtual reality tools. *Aerospace and Electronic Systems Magazine, IEEE*, 26(4):10–14, 2011.

[Zac00] Dipl.-Inform. Gabriel Zachmann. *Virtual Reality in Assembly Simulation Collision Detection, Simulation Algorithms, and Interaction Techniques.* PhD thesis, der Technischen Universitaet Darmstadt, 2000.