EUROGRAPHICS 2002 / I. Navazo Alvaro and Ph. Slusallek
(Guest Editors)

Short Presentations

Automatic Data Acquisition and Visualization for Usability Evaluation of Virtual Reality Systems

R. Holm[1], M. Priglinger[2], E. Stauder[1], J. Volkert[2], R. Wagner[1]

[1]FAW, Johannes Kepler University of Linz, Austria
[2]GUP, Johannes Kepler University of Linz, Austria

Abstract
Evaluating the usability of a Virtual Reality system is a common practice nowadays. However, gathering and analyzing the necessary data are time-consuming tasks and sometimes involve inconvenient changes to the system. The VR-based safety training system SAVE suffered from this problem. Therefore, independent modules have been developed to acquire and visualize usability relevant data to allow extensive evaluation sessions to be performed efficiently. Interaction and simulation data is acquired automatically and joined with external physiological data to be visualized inside a complete 3D representation of the training scenario. Visualizations can be performed in real-time while acquiring the data during a session, or from recorded data after the session has been completed. Because the visualization is coupled with a VR authoring tool, it is also possible to analyze the usability of a VR scenario and perform changes instantly.

Categories and Subject Descriptors (according to ACM CSS): I.6.6 [Simulation Output Analysis]; I.3.7 [Three-Dimensional Graphics and Realism]: Virtual reality; K.6.1 [Project and People Management]: Systems analysis and design

1. Introduction

Designing usable and effective interactive virtual worlds is a new challenge for system developers and human-factors specialists. In order to determine the effectiveness of a virtual environment, a means of assessing human performance efficiency in virtual worlds is first required[1].

SAVE (Safety Virtual Environment)[2,3,4] is a Virtual Reality system which is mainly intended for safety training and employee instruction and is in use at the training department of the OMV refinery in Schwechat near Vienna. It is a Head Mounted Display (HMD) based simulator comprising a motion platform on which the user stands while performing safety critical training tasks in a virtual refinery. The user may experience virtual hazards such as leaking gas, fire and explosions, upon which he/she has to react in a proper way.

Virtual environments provide great potential for training, but designing virtual environments, especially interaction techniques, is a challenging task[5,6,7,8]. To investigate the usability of SAVE, test sessions where people of varying skill levels performed typical training sessions have been conducted. Our main goal was to obtain reliable information about SAVE's human computer interface. Therefore, an automatic data acquisition and visualization module has been developed. The resulting methodologies of collecting, visualizing and analyzing data can also be applied to other VR systems.

2. Related Work

Duchowski et al.[9] describe a setup for collecting and analyzing eye movements using a 3D binocular eye tracker and discusses the application to aircraft inspection training. The recorded data provide valuable human factors process measures complementing performance statistics used to gauge training effectiveness.

Pelz et al.[10] developed a virtual reality laboratory in which head mounted displays are instrumented with infrared video-based eyetrackers to monitor subjects' eye movements while they perform a range of complex tasks such as driving, and manual tasks requiring careful eye-hand coordination.

Training issues relating to Virtual Reality systems are discussed in [11,12,13]. The design of virtual environments for usability is discussed in [5,6,7,8,14,15]. Health and safety issues of Virtual Reality systems are examined in [1,16,17,18,19,20]. In the area of safety training there are various other systems [21,22,23].

3. SAVE

Evaluating the usability of a VR system requires detailed knowledge about the topology of its hardware and software. Figure 1 shows all parts of SAVE, it comprises the following hardware components:

- An assembly and construction environment to build virtual environments. It is implemented in Java running on a standard PC.
- The main simulation and image generation. The simulation is written in C++ and runs on an Onyx2 Reality.
- An electromechanical motion platform for haptic feedback
- An instructor PC to observe and control the simulation. This PC may also be used as the sound server and the motion platform server.
- Virtual Reality devices for the interaction and the navigation in the virtual environment (a HMD and a modified two-button joystick).

SAVE consists of the following software modules:

SAVEace is the assembly and construction environment. It provides several editors to build a 3D scenario from objects, connect these objects in a logic network and design the GUI (graphical user interface) for the instructor application.

SAVEsim constitutes the simulation application. It processes user input from the tracker and the joystick. Objects within the simulation exchange events using a dependency network. The visual part of the simulation (image generation) is provided by the simulation machine.

SAVEdesk is a desktop application to control and supervise the training session. The human instructor can change the state of various parts of the simulation, trigger events, trap the trainee and manage training sessions. Its look and feel is based on industrial control panels which can be found in control rooms.

SAVEbase is the steering server for the motion platform. The base on which the trainee stands while performing her/his training tasks enhances the immersion by adding physical feedback. Motion patterns include shaking ground, vibrations and slope.

SAVEcontrol offers a user interface to control global parameters of the simulation (e.g. turning collision detection on/off, changing the maximum navigational speed). It also deals as a receiver for simulation data (user input and simulation state and events) and controls data recording.

SAVEprobe allows the visualization of user data in a 3D representation of the scenario. It is a module of SAVEace which allows simultaneous online visualizations, and comprises an avatar which performs the same movements like the trainee in the virtual environment. Offline visualization for repeated investigation is enabled by the simulation's capability to record all relevant session data.

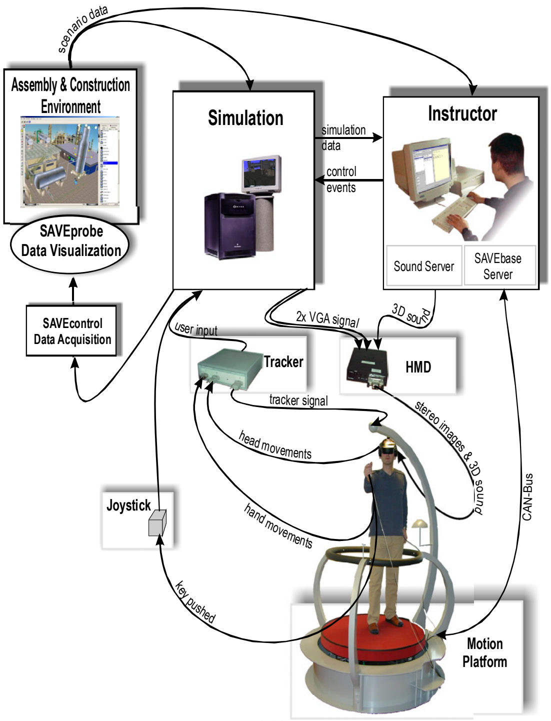

Figure 1: *Topology of SAVE*

SAVE uses a tracking system[24] to determine the trainee's position and orientation in the virtual environment. The tracking system comprises an emitter and two sensors. The emitter is mounted above the trainee's head at a fixed location. One sensor is attached to the HMD, the second sensor is mounted inside the joystick.

Because SAVE was originally intended as a safety training simulation in a virtual refinery, the common navigation metaphor is walking. Since the tracker range is limited and the HMD is connected to the simulation machine by wires, the area in which the trainee may actually move is limited to a sphere with a radius of 1.2 meters. For the trainee's safety the area is surrounded by handrails. Within that area the trainee is able to turn around, to bend or to make some steps.

The trainee wearing the HMD sees her/his virtual hand like she/he would see her/his real hand in reality. The integrated tracker sensor in the joystick corresponds to the trainee's hand position and orientation in the virtual environment. The joystick provides two buttons. One button is pushed by the trigger finger and is intended for walking through the virtual environment. The walking direction corresponds to the hand's orientation. The second button is used for user interaction and is pushed by the thumb. Objects in the virtual environment can be touched, a green semi-transparent bounding box will enclose the object. Interaction is initiated by pushing the thumb button while the bounding box is visible.

Walking through dangerous environments requires the trainee to be very careful. Obstacles lying on the ground may cause serious accidents. Since SAVE uses a HMD to display the virtual environment, its limited field of view (60° diagonal) constitutes a technical restriction[25]. If the user looks straight ahead s/he will not perceive obstacles lying on the ground close to her/him. Therefore, the trainee cannot always be made responsible for overlooking obstacles. The instructor has to tell the trainee that s/he could have fallen.

Haptic devices contribute to immersion. Therefore we designed a motion platform on which the trainee stands. The platform, steered by SAVEbase, responds automatically to changes of the trainee's vertical position in the virtual environment. A discussion of haptic feedback issues can be found in [26,27].

4. Usability Evaluation

During SAVE's development voluntary candidates tested new features. We observed them performing given tasks and they reported us a lot of impressions and suggestions (see Figure 2). But in order to obtain objective and repeatable results, we conducted a usability evaluation by acquiring data automatically from the simulation and the VR devices.

4.1. Usability of VR Systems

Human Computer Interaction (HCI) requires devices which enable humans to interact with computers. In Virtual Reality systems like SAVE such devices comprise amongst other things HMDs and joysticks. Since these devices show technical limitations, VR developers have to cope with that [7,8,14,25]. Consider a training task like the assembly of a machine from individual parts. In the real world, the trainee might use his hands and tools, he feels force feedback when touching a surface. The same task performed in a virtual environment requires devices for interaction, e.g. a joystick. The way to operate components in virtual environments differ from the real world, metaphors have to be developed [5,15]. Being aware of that, a major objective of a usability investigation is to pinpoint the shortcomings of the interaction metaphors used and to improve them.

4.2. Usability of VR Scenarios

Assembling virtual environments using SAVEace turned out to be very easy and intuitive for the scenario author, arbitrary objects may be arranged in a convenient way. After completing scene assembly, the scenario has to be checked in terms of usability. Some training tasks have to be fulfilled in time, e.g. escaping from burning buildings. Other tasks, e.g. performing inspection rounds in industrial areas, demand the trainee to read information from gauges or signs, usually displayed as textured objects. In the virtual environment textures may appear different from desktop applications and have to be adapted to look as intended. Sometimes technical limitations may cause unexpected problems. While testing the scenario it turned out that interaction with some objects became hard. The range of the applied tracker is limited to 1.2 meter to work accurately, so picking up little objects lying on the floor became rather difficult.

Any shortcoming detected as result of a usability evaluation contributes to better immersion and faster development cycles.

4.3. Data for Evaluating the Usability

Before performing an evaluation it is important to know what kind of data will have to be collected. For a usability evaluation it is important to collect data which relates to the interaction and the state of the simulation (i.e. about what is happening in the scenario). A question in this respect is what levels of abstraction should be considered: Is it useful to collect basic input data (tracking data, button clicks etc.)? Is higher level data (events, actions etc.) available and independently meaningful?

For our system we decided to gather all input data and internal simulation states/events which allow us to replay an entire session. This kind of data is also suitable for automatic acquisition and visualization, because raw input data is hardly affected by changes of the VR system, the scenarios or higher level interaction techniques.

Usability evaluations usually involve the measurement of physiological parameters, too. Gathering cardiovascular parameters (e.g. heart rate, blood pressure) is a common practice, but electroencephalogram recordings [10] or various endocrine parameters (e.g. stress hormones like cortisol) are also valuable indicators for physiological stress or psychological strain relating to the usability and ergonomics of a system, especially in Virtual Reality.

Every data sample must be accompanied by a time stamp, or it must be possible to reconstruct a time stamp for a sample (e.g. when a fixed sampling rate is used and the time of the first sample is known). Different kinds of data are most easily set in relation by matching their time stamps which can be performed automatically in real-time. Parameters with a single value or "before and after" values for a whole session (e.g. heart rate variability [28], hormone samples) are not useful for automatic acquisition because they cannot be set in relation to values of data streams.

A lot of information can be extracted from these data sets with rather simple calculations: response times, task latency, areas of intense interaction and navigation, excessive collisions, ergonomics (e.g. problems with reaching objects), areas and tasks which stress the user and a lot more which has not been thought about when performing the test session.

Figure 2: *A test performance in progress*

5. Automatic Data Acquisition and Visualization

An important goal of every system evaluation is the acquisition of as much data as possible in a convenient way. It should be possible to combine different data sets to gain new insights into interrelations. A good solution is to acquire the data generated during a test session with minimal interference from test personnel. Furthermore, it is useful to have the data available instantly for a preliminary review of the evaluation process.

5.1. Controlling Data Recording

SAVEcontrol has been developed as a dedicated data acquisition module which receives data over the network from the simulation machine in real-time. The SAVE simulation is capable of recording all of its input data onto its local hard disk, or of sending it across the network to a remote application like SAVEcontrol. Recording and sending do not interfere with the simulation itself. It is performed in real-time on a per-frame basis and can be scaled down for less powerful network connections or simulation hosts by changing the sampling rate, or processing only changes in data above a specified maximum derivation from the previously processed data set.

The simulation can record or send all of its input data including data from the tracking system (spatial position and orientation of the user's head and hand relative to a fixed point in the real world), interaction data (clicks with the joystick) and navigation data (clicks on the navigation button, position and orientation of the user in the coordinate system of the virtual environment) – see Figure 3. It also records or sends a subset of its internal simulation state: collisions of the user's virtual body and hand with objects in the virtual environment, a protocol of events in the simulation (e.g. user opens valve nr. 12 at time 362712; tank 123A catches fire at time 647332) and network traffic (commands from the instructor panel, state of the motion platform, triggering of sounds in the sound server, etc.). These data sets are not necessary for the trainer to control the session and rate the trainee's performance. Combining them with the session time, they deliver valuable information about user actions and allow the measurement of response times.

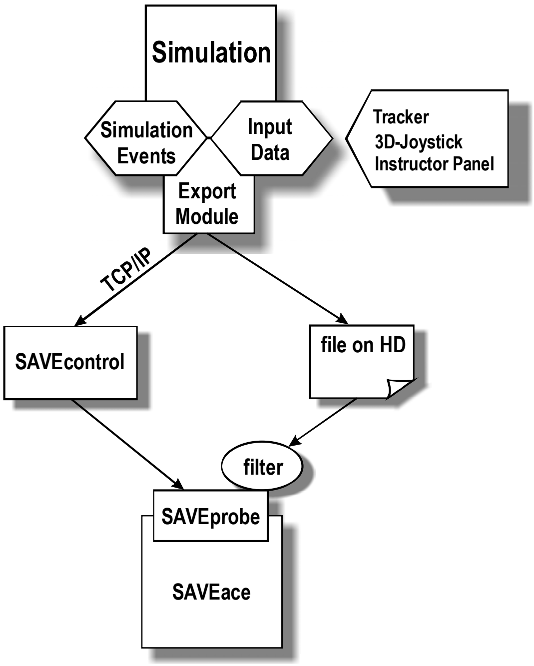

Figure 3: *Data Acquisition Scheme*

SAVEcontrol can start, pause and stop the data acquisition, select the destination for the data (local hard disk or a network client) and synchronize the simulation clock with an external clock (e.g. with the heart rate monitor or the video camera) and assign a user identification number to the data for later unification with external data.

The simulation can be any application which can send data using the custom SAVEcontrol protocol. The protocol is simple: a data package consists of a type identification flag (e.g. position, time etc.) and the value (e.g. a vector of three floats in the case of a position identification flag). The time is the most important data package because it is used to fuse or combine all the different data packages. Every incoming data package is interpreted as a sample which has been taken at the time of the last time package value.

5.2. Data Visualization

To visualize the data, we have developed the data import module *SAVEprobe* for our scenario authoring tool SAVEace. The authoring tool shows the same scenario used in the simulation, viewed from an observer's position. The observer can examine the scenario by rotating it and zooming into it, or by flying through it. SAVEprobe receives data on the fly from SAVEcontrol, or can read a complete set from the hard disk. The data can be visualized inside the authoring tool's copy of the scenario. Visualizations can either be static, or animated if permitted by the type of data (e.g. motion data can be used to animate an avatar). The observer can select filters for the data to display only the data s/he's interested in (e.g. only navigational data as shown in Figure 4).

Different types of visualizations can be selected. SAVEprobe can currently visualize all data types of the SAVEcontrol package protocol which include:

- position of the user in the virtual environment
- sensor positions (head and hand)
- sensor orientations (head and hand)
- collisions (collision events and collision points)
- interaction events (button clicks)
- external data (e.g. heart rate)

The visualization of a single data set is placed at the according position of the user in the scene, i.e. at the location where the user was in the scene at the time of the sampling of the data set.

There are 6 different visualization styles used in SAVEprobe:

1. continuous lines to visualize position changes
2. single vertical lines of value dependent length to visualize single absolute values (e.g. heart rate)
3. single vertical vectors of value dependent length and color to visualize value changes (e.g. a vector of length 8 upwards depicts a rise in heart rate of 8 beats, a vector of length 7 downwards depicts a fall in heart rate of 7 beats)
4. direction vectors to visualize orientations (e.g. the view direction)
5. spheres of value dependent size and color to visualize extrem values (e.g. a green sphere for the lowest heart rate)
6. spheres of fixed size and value dependent color to visualize occurrences of events (e.g. user button clicks, events of the motion platform)

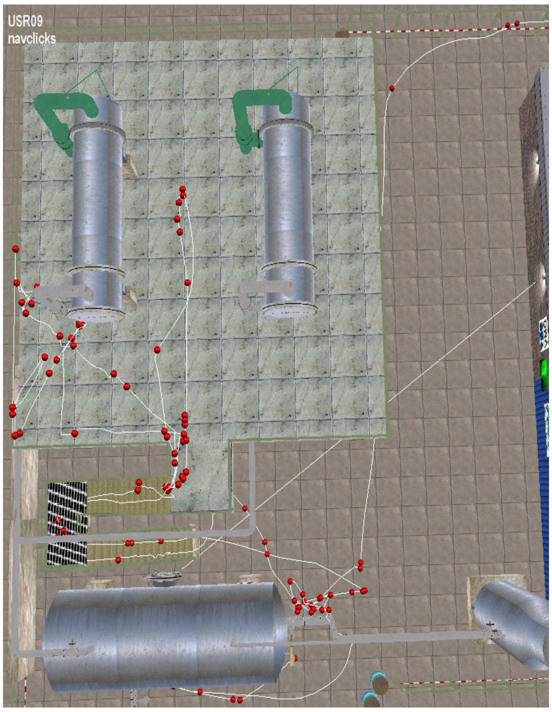

Figure 4: *Visualization of the navigation path and stop positions*

The visualization styles can be switched for data types at runtime during an online visualization or a visualization of recorded data. SAVEprobe calculates and visualizes some more data from the existing data sets, e.g. the speed of the user, accelerations and decelerations.

In the case of the SAVE system, virtual environments and scenarios which are evaluated have been created with the SAVEace authoring tool. Therefore they are especially suitable for the visualization because their components are known to SAVEace and SAVEprobe and their states can be queried, too. E.g. if the user opens a valve in the simulation, it will appear open in the SAVEace scenario visualization as well.

5.3. Visualizing External Data

Besides data from the simulator, external data sources can be included and combined with the simulator data. In the case of a usability evaluation, it can be useful to monitor physiological data of testers. Today's heart rate monitors offer a convenient way of gathering data and exporting it for use in a computer.

However, it is also possible to visualize scenes of other VR systems if the scene is available in a format which can be loaded by SAVEace. Because SAVEace is built on top of Java3D it supports all formats for which a geometry loader is available. Data does not have to come online via SAVEcontrol. SAVEprobe can read data from files which have to comply to the SAVEcontrol data package protocol in a recorded form, or a filter component class (Java Bean) must be available to convert the data upon loading.

SAVEprobe treats this as external data and visualizes it at the corresponding location in the virtual environment where the user was at the time of acquisition of the heart rate value. Data sources are automatically combined using the sampling time of each value. The set of data containing the position of the user in the virtual environment is the most important reference for the visualization. Every data sample is visualized at the corresponding position in this data set by trying to match the time stamp of the sample with the nearest possible time stamp contained in the position data.

If the external data source does not produce time stamps, they can be generated by specifying a fixed sampling rate, or calculating the time from interpreting the data itself. The heart rate monitors of an evaluation of SAVE recorded the current heart rate in a 1/100 seconds accuracy at the time of an occurrence of a heart beat. As the heart rate is a function of the space of time between two heart beats, it's easy to recalculate the time stamp for each sample.

5.4. Avatar Animation

For reviewing, SAVEprobe can present a visualization of the tester's path through the virtual environment, the viewing directions, the hand and head movements, as well as external data (the heart rates) at each task in the session or at each position in the scenario (see Figure 8). The application is able to replay the test session with an avatar representing the tester. The avatar is an example for an animated visualization of the recorded data. It performs the same movements like the user according to the tracker simulation data in real-time. However, because the SAVE simulation does not track legs or body movements, body motions are extrapolated from head and hand data while leg movements (walking) are missing completely. The avatar animation can be combined with other visualizations, e.g. a trace of the path through the virtual environment, vectors indicating the viewing direction etc. If available, external data (e.g. the current heart rate) can be displayed above the avatar. Figure 6 and Figure 7 show a review session with the current heart rate above the avatar. Figure 5 shows an on-the-fly visualization where the trainee is guessing the distance to a telephone cabin [29].

To find extremes in the data, SAVEprobe computes the mean, minimum and maximum values in every data set, as well as the amount and direction of change between two samples. This is especially helpful in detecting areas and times of stress in the scenario and the training session, respectively. Sample values above the mean are visualized in red color, values below the mean in green color and extreme values (e.g. occurrence of the maximum recorded heart rate value) are marked with an additional sphere. Alternatively, rising values can be visualized differently from lowering values.

It is still possible to query component states, display various visualizations and navigate freely during an avatar animation which would not be possible in VCR-style recordings of the visual simulation.

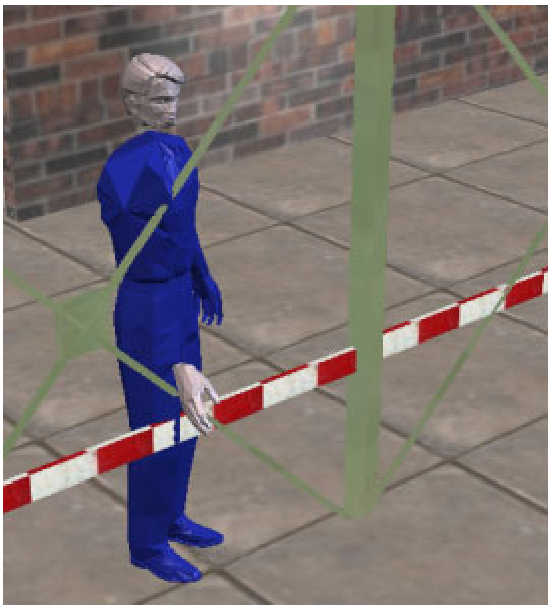

Figure 5: *Online avatar animation where the user is guessing the distance to an outside telephone cabin*

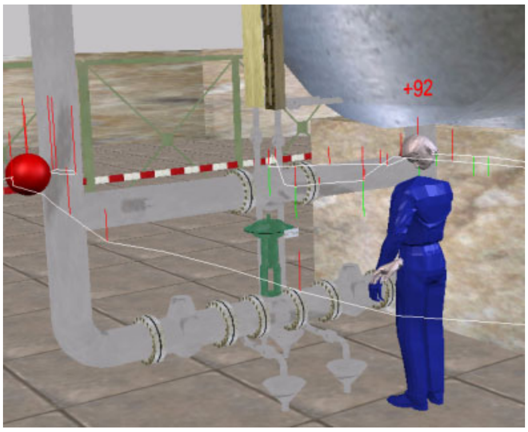

Figure 6: *Avatar animation with trajectory and heart rate visualization*

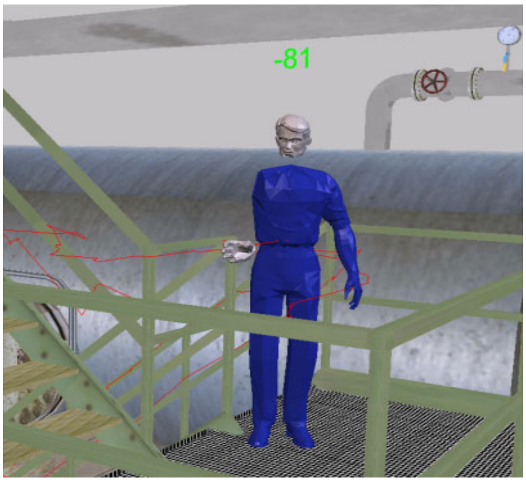

Figure 7: *Replayed avatar animation*

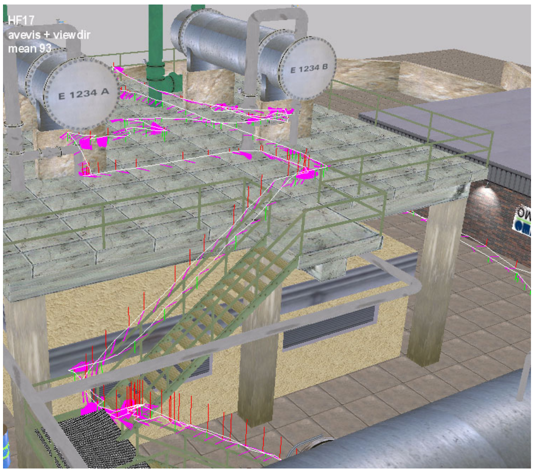

Figure 8: *Visualization of the path, heart rate changes and view directions*

5.5. Unification of Multiple Sessions

The unification of data sets of multiple sessions produces a huge amount of data which usually cannot be displayed in real-time. Currently SAVEprobe offers only two ways to overcome this problem:

- reduction of the sampling rate and visualization of multiple data sets at the same time (in the same scene)
- calculation of mean and extreme values and visualization of these statistical values

However, the software architecture allows the integration of filter components to process the data before visualizing it. An example would be the implementation of an algorithm to calculate the convex hull of the paths through the scenario which can be regarded as the orbital of the users (the most probable path).

6. Conclusions

The original intention of SAVEprobe is its application as a tool for gathering and visualizing as much data as possible during evaluation sessions. In a five day usability and ergonomics evaluation of the SAVE safety training system we performed sessions with 41 testers. A typical session lasted for about 18 minutes and produced an average of 4MB of raw data at a rate of 5 frames per sample. Samples were recorded and a visualization and avatar animation was instantly available to review the session. This was helpful when we were interested in the opinion of the testers about a certain incident or usability problem encountered during the session. E.g. a high heart rate during a task or a disoriented path through some parts of the scene helped us ask the tester in more detail about his/her subjective feelings or problems at this special time of the session. An animated replay also provided a common base for discussing the session and made it easier for the testers to remember what they felt or what problems they had during a task.

Different strategies to overcome shortcomings of the VR hardware have been evaluated, e.g. jitter of the electromagnetical tracking system (see Figure 9). An important question is the influence of the SAVEbase motion platform on the user's physiology, which we are investigating. The collected data and the visualization of heart rate data in combination with what is going on at the same time in the scenario, offer a valuable tool to researchers who are not experts in Virtual Reality – e.g. the physiological effects of simulated hazards become visible if the straight path and relaxed heart rate before the hazard can be seen in direct opposition to the disoriented path with high heart rate of the escape (see Figure 10). They are enabled to regard tasks in the virtual environment in the same way as tasks in the real world when watching the avatar's actions or the visualizations of data at the according locations in the scenario.

Another important application is testing and tuning of a scenario. Analyzing the recorded data may offer insight into shortcomings of the scenario: e.g. virtual appliances which are hard to reach for the user, new objects in the scenario which are hard to use, problems with navigating, hotspots in the scenario where the user spends a lot of time (and therefore should be improved in terms of visual appearance or higher frame rate).

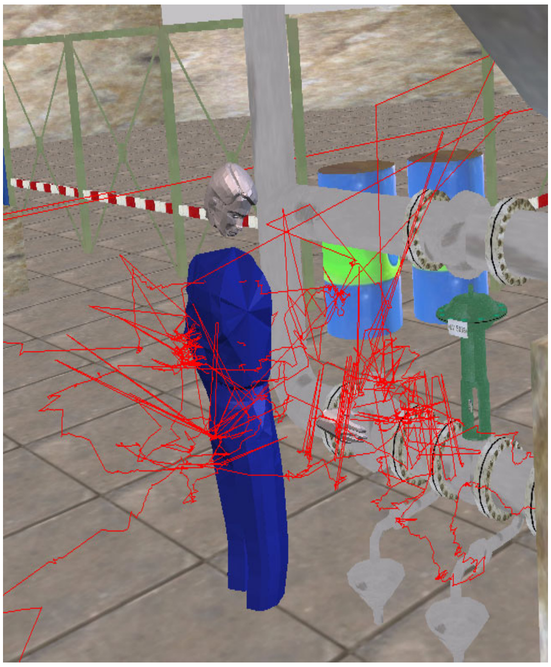

Figure 9: *Jitter in hand movements*

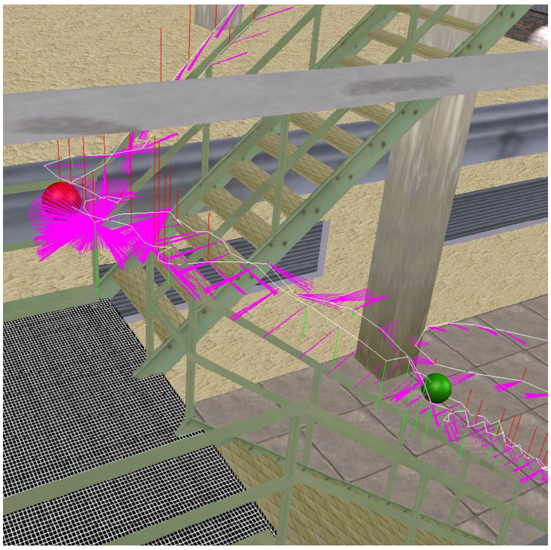

Figure 10: *Heart rate and view direction before and after a simulated fire hazard*

Because the visualization and avatar animation is an integral part of the SAVE system without interfering with regular training sessions, it can be used by the instructor as a tool for observing the actions of the trainee. It also offers a tool for debriefing when the instructor shows the trainee what she/he has done wrong.

The visualization module proved to be a valuable tool for investigations of the recorded data and as a means for reviewing training sessions. It will be included in the final simulator as a tool for debriefing.

Special visualizations for unified data from multiple sessions have to be developed (e.g. walking volumes rather than individual lines). More visualizations of data extracted from existing data sets have to be done, e.g. a visualization of volumes depicting areas the user has never seen during his/her navigation through the scenario.

7. Acknowledgements

The authors would like to thank the project manager Anton Dunzendorfer, Friedrich Kreuzriegler, Dieter Kranzlmüller and the project team of OMV AG Schwechat for their support: Wolfgang Ginzel, Franz Grion, Alois Mochar and Günther Schwarz.

References

1. K.M. Stanney, R.R. Mourant, R.S. Kennedy. Human Factors Issues in Virtual Environments: A Review of the Literature. *Presence, Vol 7, No. 4*, August 1998, 321-351.
2. R. Holm, M. Priglinger, E. Stauder, J. Volkert, R. Wagner. A Combined Immersive and Desktop Authoring Tool for Virtual Environments. *IEEE Virtual Reality 2002*, Orlando, March 2002.
3. M. Haller, R. Holm, M. Priglinger, J. Volkert, R. Wagner. Components for a virtual environment. *Workshop on Guiding Users through Interactive Experiences: Usability Centred Design and Evaluation of Virtual 3D Environments*, Paderborn (Germany), April 2000.
4. M. Haller, R. Holm, J. Volkert, R. Wagner. A VR based safety training in a petroleum refinery. *20th Annual Conference of the European Association for Computer Graphics (Eurographics '99)*, Milano (Italy), September 1999.
5. D.A. Bowman. Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application. Ph.D. thesis, Virginia Polytechnic & State University, 1999.

6. H.-J. Bullinger, W. Bauer, M. Braun. Virtual environments. *In G. Salvendy, Handbook of human factors and ergonomics (2nd ed.),*. New York et al.: Wiley, pp. 1725-1759.
7. K. Kaur. Designing Virtual Environments for Usability. *In S. Howard, J. Hammond, and G. Lindgaard, editors, Human-Computer Interaction: INTERACT'97*, Chapman and Hall, 1997, pp. 636-639.
8. R.W. Lindeman, J.L. Sibert, J.K. Hahn. Towards usable VR: an empirical study of user interfaces for immersive virtual environments. *Proc. CHI '99*, Pittsburgh, 1999.
9. A.T. Duchowski, E. Medlin, A. Gramopadhye, B. Melloy, S. Nair. Binocular Eye Tracking in VR for Visual Inspection Training. *Proceedings of the ACM symposium on Virtual reality software and technology*, November 15-17, 2001, Banff, AB, Canada.
10. J.B. Pelz, M.M. Hayhoe, D.H. Ballard, A. Shrivastava, J.D. Bayliss, M. von der Heyde. Development of a Virtual Laboratory for the Study of Complex Human Behavior. *Proc. SPIE-The International Society for Optical Engineering*, Vol. 3639B, The Engineering Reality of Virtual Reality, San Jose, 1999.
11. J.K. Caird. Persistent issues in the application of virtual environment systems to training. *IEEE HICS '96*, Dayton, 1996.
12. C.R. Hall, R.J. Stiles, C.D. Horwitz. Virtual Reality for Training: Evaluating Knowledge Retention. *Proc. VRAIS '98*, Atlanta, 1998.
13. D.A. Philbin, W. Ribarsky, N. Walker, C.E. Hubbard. Training in Virtual Environments: Analysis of Task Appropriateness. *Proc. VRAIS '98*, Atlanta, 1998.
14. C. Esposito. User interface issues for virtual reality systems. *Proceedings of the CHI '96 conference companion on Human factors in computing systems: common ground*, Vancouver, 1996, pp. 340 - 341.
15. I. Poupyrev, S. Weghorst, M. Billinghurst, T. Ichikawa. A framework and testbed for studying manipulation techniques for immersive VR. *Proceedings of the ACM symposium on Virtual reality software and technology*, p.21-28, September 1997, Lausanne, Switzerland.
16. S.V.G. Cobb, S.C. Nichols, A.R. Ramsey, J.R. Wilson. Virtual Reality-Induced Symptoms and Effects (VRISE). *Presence: Teleoperators and Virtual Environments*, 8(2), 1999, pp. 169-186.
17. P.A. Howarth. Empirical Studies of Accommodation, Convergence, and HMD Use. *Proceedings of the Hoso-Bunka Foundation Symposium*, Tokyo, 1996
18. P.A. Howarth, K.J. Hill. The maintenance of habituation to Virtual Simulation Sickness in Human-Computer Interaction: Ergonomics and User Interfaces. *Proceedings of HCI '99*, August 22-26, 1999,Vol 1. pp 137-141, Munich, Germany.
19. J.J. LaViola. A Discussion of Cybersickness in Virtual Environments. *SIGCHI Bulletin* 32(1), January 2000.
20. K.M. Stanney. Realizing the full potential of virtual reality: Human factors issues that could stand in the way. *Proc. VRAIS '95,* Research Triangle Park, 1995, pp. 28-34.
21. R. Bukowski, C. Sequin. Interactive simulation of fire in virtual building environments. *Proc. ACM SIGGRAPH*, Los Angeles, 1997, pp. 28-35.
22. R.B. Loftin, P.J. Kenney. Training the Hubble Space Telescope Flight Team. *IEEE Computer Graphics and Applications*, Vol. 15, No. 5, September 1995, pp. 31-37.
23. D. Tate, L. Sibert, T. King. Virtual Environments for Shipboard Firefighting Training. *Proc. VRAIS '97*, Albuquerque, 1997, pp. 61-68.
24. B.D. Allen, G. Bishop, G. Welch. Tracking: Beyond 15 Minutes of Thought. *SIGGRAPH 2001, Course 11*, Los Angeles, August 2001.
25. K. Arthur. Effects of field of view on task performance with head-mounted displays. *Proceedings of the CHI '96 conference companion on Human factors in computing systems: common ground*, Vancouver, 1996, pp. 29-30.
26. G. Burdea. Haptics Issues in Virtual Environment. *Proc. IEEE Computer Graphics International*, Geneva (Switzerland), June 2000.
27. W. Mark, S. Randolph, M. Finch, J.V. Werth, R. Taylor. Adding Force Feedback to Graphics Systems: Issues and Solutions. *Proc. ACM SIGGRAPH*, New Orleans, 1996, pp. 447-452.
28. M.P. Tulppo, T.H. Mäkikallio, T. Seppänen, J.K. Airaksinen, H.V. Huikuri. Heart rate dynamics during accentuated sympathovagal interaction. Am J Physiol, 274(3 Pt 2), 1998, H810-816.
29. R.G. Eggleston. Virtual Reality System Effects on Size Distance Judgments in a Virtual Environment. *Proc. VRAIS '96*, Santa Clara, 1996.