

LIRKIS CAVE: Architecture, Performance and Applications

Štefan Korečko, Marián Hudák, Branislav Sobota

Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 041 20 Košice, Slovakia, stefan.korecko@tuke.sk, marian.hudak.2@tuke.sk, branislav.sobota@tuke.sk

Abstract: LIRKIS CAVE is a contemporary Cave Automatic Virtual Environment, developed and built at the home institution of the authors. Its walls, ceiling and floor are covered by stereoscopic LCD panels, user movement is tracked by OptiTrack cameras and scene rendering is carried out by a cluster of seven computers. The most unique feature is a portable design. It allows for disassembly of the whole CAVE to transport it to another location. The paper describes the hardware and software of the CAVE and presents results of several performance evaluation experiments. It also deals with current and future applications of the CAVE, which fall into the area of cognitive infocommunications and are primarily aimed toward impaired people.

Keywords: Virtual Reality; CAVE; Stereoscopy; Visualization; Wheelchair simulation

1 Introduction

Thanks to recent technological advancement virtual reality (VR) has become a hot topic, again. The most common types of devices that allow for an immersion into a virtual world are head-mounted displays, or VR headsets, and CAVE systems. It is the first type that is primarily responsible for the recent VR boom. The increase of mobile computing systems performance and display quality allowed to create head-mounted displays affordable for the general public. The price of VR headsets, such as Oculus Rift¹ and HTC VIVE², is around 500 €. And there are even cheaper solutions available. For example, Google Cardboard³ and derived products can create a stereoscopic display from a smartphone for about 10 €.

¹ <https://www.oculus.com/>

² <https://www.vive.com/>

³ <https://vr.google.com/cardboard/>

On the other hand, CAVE (Cave Automatic Virtual Environment) systems will not become so widespread any time soon, at the very least because of their size. A typical CAVE system has a form of a room, where the walls, and in some cases also the floor and the ceiling, are used to display the virtual world. When CAVE systems had been originally introduced in 1990s, they offered two features that the VR headsets of that era weren't able to deliver [1]: an unprecedented field of view and no need for a virtual representation of the user's body, because the user physically entered the virtual space. While some expensive contemporary headsets offer a wide field of view⁴, the second feature is still exclusive to CAVE systems. In addition, several persons can occupy a CAVE simultaneously and they can interact naturally, as in the real world. And it has been shown that CAVE systems cause less simulation sickness than VR headsets [2]. It should be also noted that CAVE systems have evolved significantly since their introduction, too [1]: High-performance computer clusters allow high-resolution graphical output rendering and multiple user input processing in real time. The original CRT projectors have been replaced by DLP, LCD or LCoS ones. And the introduction of large-size high-resolution LCD panels has offered an alternative to the projector screens.

One of the most recent CAVE systems that fully utilizes these technological developments is the LIRKIS CAVE. It has been designed and built at the home institution of the authors on the basis of their previous experience with virtual reality technologies [3], [4]. The LIRKIS CAVE is an LCD panel-based CAVE system of a cylindrical shape, which provides a 250 degree panoramic space. LCD panels cover the walls as well as the ceiling and the floor of the CAVE. The system supports various control devices such as a joystick, a gamepad, the MYO armband and an EEG headset. Users may also use hand gestures and head movements, which improve their immersion into the virtual scene. Maybe the most original feature of the CAVE is its compact and transportable design.

The LIRKIS CAVE is described in detail and evaluated in the rest of this paper, which is organized as follows. First, Section 2 lists other similar CAVE systems and compares them to our solution. It also relates the CAVE to the cognitive infocommunications research and development. Section 3 describes the LIRKIS CAVE and its software and hardware components. Section 4 reports results of several performance tests carried out in the CAVE, including a test of a newly developed thread-based scene computing. Section 5 outlines applications of the CAVE. Finally, Section 6 concludes with a summary of achieved results and plans for future development from the cognitive infocommunications point of view.

⁴ For a detailed comparison, please see <http://virtualrealitytimes.com/2017/03/06/chart-fov-field-of-view-vr-headsets/>

2 Related Work

There are several contemporary CAVE systems that share particular features with the LIRKIS CAVE. Probably the most related one is CAVE2 [5], which is similar in the overall shape and hardware configuration. Both CAVEs are of a cylindrical shape with LCD panels and optical, camera-based, motion tracking systems. CAVE 2 is a large one, with 7.5 meters in diameter and 72 LCD panels. The LIRKIS CAVE uses 20 LCD panels and has 2.5 meters in diameter. The panoramic space is larger in CAVE2. It offers 320 degrees, while the LIRKIS CAVE has 250 degrees. On the other hand, there are no ceiling and floor displays in CAVE2 and it is not transportable.

With 3 meters in diameter, StarCave [6] offers nearly the same internal space as our solution. The biggest difference is in the display technology where StarCave uses a backward projection and the LIRKIS CAVE the LCD panels. Both technologies have their advantages and disadvantages. Projectors can generate a continuous image for all walls of a CAVE, without any visible seams. On the other hand, they require considerable extra space outside the CAVE (about 2.6 m for each wall in StarCave). Because the StarCave designers didn't have the necessary space below the floor of their CAVE and considered the floor projection important, they used a down-projection. Therefore, StarCave doesn't have any ceiling projection. In addition, the image projected on the floor is imperfect because users stand in the way of the projectors. In the LCD panels-based LIRKIS CAVE no extra space is required and both the ceiling and the floor have the screens. However, the visible bezels of the LCD panels may disturb some users. Another difference between the CAVEs is the horizontal screens organization. In StarCave they form all 5 sides of a pentagon, while in the LIRKIS one seven sides of a decagon.

The space requirements of the backward projection-based CAVEs are also evident in the Zvolen CAVE [7]. It is situated at the Technical University in Zvolen, Slovakia and its primary purpose is a forestry-related visualization. It has a block shape with 3 m width 3 m length and 2.5 m height. But the room where it is situated is about three times bigger to make the space for the projectors. The stereoscopic image is projected directly on three horizontal walls and by means of mirrors on the floor and ceiling. In addition to the similar usable space, the visualization software of the Zvolen CAVE is also based on the same graphics library as our CAVE, i.e. on OpenSG⁵.

Compared to the aforementioned solutions, the LIRKIS CAVE offers an original combination of a compact and transportable design, a self-supporting construction, a high image resolution provided by full HD LCD panels, a wide viewing angle and a presence of both the floor and ceiling displays. In addition, the system is

⁵ <https://sourceforge.net/projects/opensg/>

designed as modular with a possibility to change or extend both the hardware and software components. This is also true for the displays, provided that new ones will be of the same size as the currently used ones.

With respect to the cognitive infocommunications [8], the LIRKIS CAVE can be related to the VirCA [9], [10] collaboration VR platform, which later evolved into MaxWhere⁶. In MaxWhere, a 3D virtual scene serves as a space, where users share documents, multimedia and other resources. The collaboration is possible thanks to multiple web browser panels, included in the scene. The browser panels allow accessing the resources directly or by running corresponding web applications. Recent experiments [11], [12] proved that MaxWhere is an effective platform for collaborative information and workflow sharing. The platform has been also used for other interesting tasks, such as an evaluation of a 2D advertising in 3D virtual space [13], an assessment of the role of VR in communication and memory management [14] and a virtual laboratory system [15]. The LIRKIS CAVE can be used for such collaboration and experimentation, too. It can be achieved by adapting MaxWhere or developing a similar software platform. Being a fully immersive VR installation with rich peripherals, the LIRKIS CAVE can serve as a home for multiple cognitive infocommunications – related experiments and applications: A wheelchair simulation, with goals similar to [16], is under development now (section 5). The CAVE can be also used for so-called exergames [17], utilizing its OptiTrack motion tracking system or other sensors, such as Microsoft Kinect or the Myo armband. Another application area is a virtual reconstruction of historical sites, in a way similar to [18].

3 LIRKIS CAVE

The LIRKIS CAVE (Fig. 1) consists of two standalone components: a rack holding a computing cluster of the CAVE (the white rack in Fig. 1 a) and the CAVE itself (the rest of Fig. 1 a). The CAVE is situated inside a self-supporting steel frame, which is 2.5 m wide, 2.5 m long and 3 m high. The frame holds all the LCD panels and audio and tracking systems of the CAVE. Twenty stereoscopic LCD TV sets with diagonal 55” are used as the panels. They are distributed vertically along the sides (14 panels) as well as horizontally (3 panels in the ceiling and 3 panels under the floor). The 14 vertical panels form 7 sides of a decagon. This can be seen in Fig. 1 b), where the solid lines represent the panels and the dashed rectangle is the steel frame. The position of the user is the same as in Fig. 1 a). The floor panels are installed under a safety glass, which can support five adults. The total weight of the CAVE is about 2000 kg. The frame doesn’t need to be fixed to the floor or walls of the room where it is situated by any

⁶ <https://store.maxwhere.com/>

means. The whole CAVE, including the frame, can be disassembled and transported to another place.

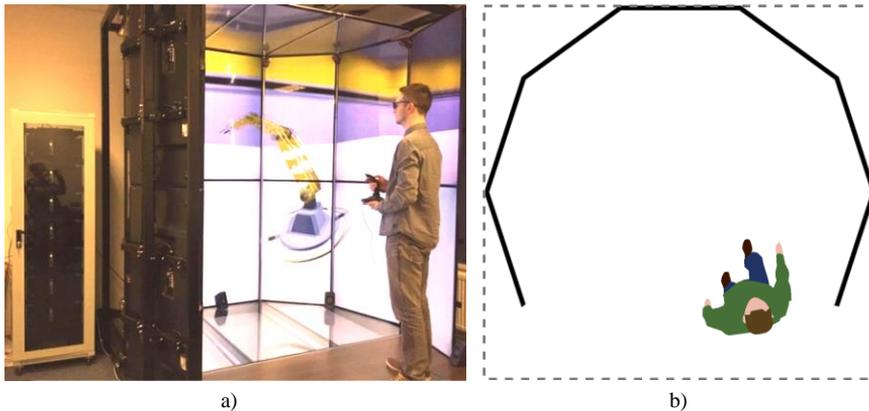


Figure 1

LIRKIS CAVE with a user controlling a hydraulic arm in the virtual scene by a joystick (a) and a schema showing placement of its vertical LCD panels from above (b)

3.1 Hardware

The LIRKIS CAVE hardware consists of a computing cluster, user input devices, LCD panels and an audio system. The audio system is a THX-Certified 6 channels speaker system by Logitech, which noticeably contributes to the immersion in a virtual scene.

3.1.1 Computing Cluster

The computing cluster is responsible for the user input processing, audiovisual output rendering and control over the whole system. Clusters are popular in CAVE systems as they support variability of an attachment and configuration of display units for computing [19]. In the cluster structure, each computing unit controls a portion of the three-dimensional environment. The number of computers depends on the number and resolution of the displays, the complexity of the virtual scene and required performance.

The LIRKIS CAVE cluster contains 7 computers, 1 master and 6 slaves (Fig. 2). The master computer manages the communication between all the computers in the cluster and also supplies the slaves with the data necessary for the 3D scene rendering. The slave computers carry out the rendering itself and related tasks. Each slave renders several parts of the scene, one for each LCD panel attached to it. To provide sufficient graphical performance, the slaves are equipped with NVIDIA Quadro graphics cards. The configuration of the cluster computers is given in Table 1.

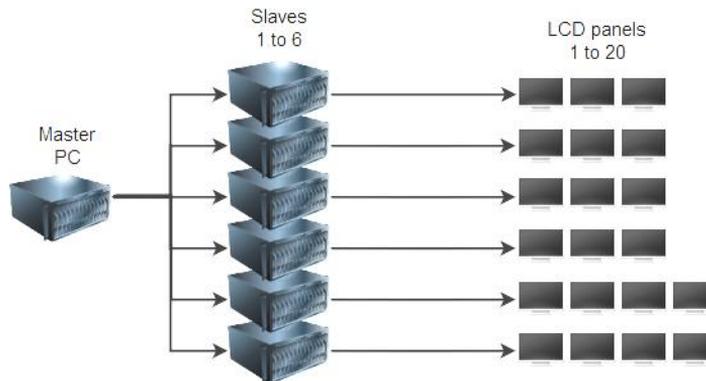


Figure 2

Cluster rendering with master and slave computers

The scene rendering is synchronized exclusively via the master computer; the slaves do not communicate with each other. The master also provides a basic level of control over individual slaves for the user. In the current configuration of the CAVE, each slave renders the image for 3 to 4 LCD panels (Fig. 2). However, it is possible to change the configuration in the control software of the CAVE.

Table 1

Configuration of the LIRKIS CAVE computing cluster computers

| PC | Processor | Graphic Card | RAM capacity | Drive type/ capacity |
|---------------|-----------------------|-------------------------|--------------|-------------------------|
| Master | Intel® Core™ i7-7700K | integrated | 16GB | SSD / 500GB |
| Slave | Intel® Core™ i7-7700K | NVIDIA Quadro K5000 4GB | 16GB | SSD / 500GB |

3.1.2 Input Devices

Input devices of the LIRKIS CAVE fall into two categories. The first one is a real-time user tracking and it is solely occupied by a system of eight “OptiTrack Flex 13” cameras. To provide the best capturing performance, the cameras are arranged along the top of the CAVE with 7 cameras in the upper corners of the vertical LCD panels and one behind the user, on the metal frame (Fig. 3 a).

The user tracking is necessary for providing faithful representation of the virtual environment: While the images for all screens are rendered from the same point in the scene, each of them is under a different angle. And these angles are changing when the user moves inside the CAVE. The position of the point in the scene is changing, too. Therefore, to maintain the illusion of the presence in the virtual world, the visualization engine of the CAVE reads the user position from the OptiTrack system and adjusts the position and angles before each frame rendering.

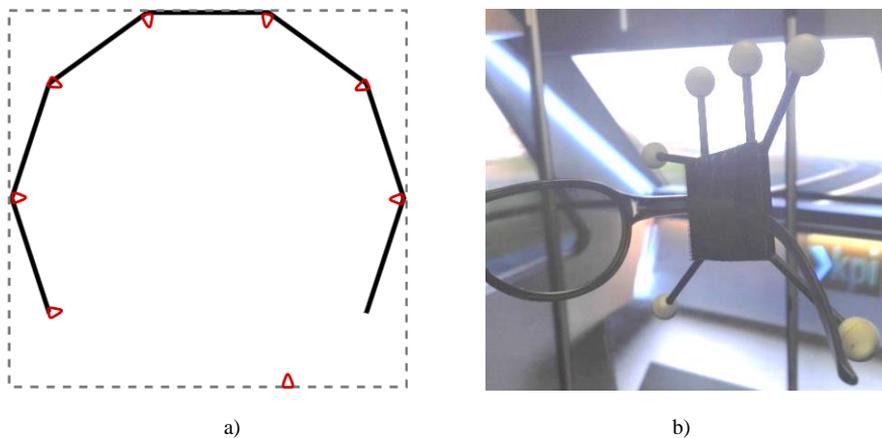


Figure 3

Placement of OptiTrack cameras (red triangles) in the LIRKIS CAVE (a) and an OptiTrack marker located on 3D glasses (b)

The OptiTrack system scans the user position by means of a marker, fixed to the user's 3D glasses (Fig. 3 b). To understand the need of the user tracking one may compare Fig. 3 b) and Fig. 11 a) (Section 5). In Fig. 3 b) the rendered image continues correctly from one LCD panel to another. However, there is an observable deformation between the panels in Fig. 11 a), because the camera taking the image was far from the marker position. Other CAVEs, e.g. [6], [7], use this approach, too. Its slight disadvantage is that only one person, the one with the marker, gets the perfect immersion.

The second category contains devices used to control the rendered scene and objects in it. Multiple devices can be used at once, simultaneously with the OptiTrack system. A wide range of devices is currently supported by the CAVE: from the traditional devices such as a mouse and a keyboard, through gaming devices (joystick, gamepad) to very specific ones, e.g. a 3D mouse, the Emotive Epos⁷ EEG headset and the Myo⁸ gesture control armband. The current status of the support is in more detail described in [20].

3.1.3 LCD Panels

The choice of LCD panels as display devices was a necessary one considering the desired compactness and transportability. The panels used are 55" LCD TV sets manufactured by LG, each with the full HD resolution (1920 x 1080 pixels). They produce stereoscopic image, utilizing passive 3D technology and circular

⁷ <https://www.emotiv.com/epoc/>

⁸ <https://support.getmyo.com>

polarization. Therefore, it is needed to wear 3D glasses in order to experience 3D illusion of the displayed scene. The organization of the vertical displays into the decagon (Fig. 1 b) is not a typical one, but was selected for two practical reasons. The first one was our intention to provide as natural viewing angles as possible, considering the small size of the cave. Second, we tried to keep the number of displays forming a single wall to a minimum in order to make their bezels as unobtrusive as possible. Now a single wall consists of only two displays, organized vertically in the portrait position (Fig. 1 a). Unfortunately, it was impossible to use bezel-less displays as they were not commercially available and the limited CAVE development budget didn't allow any customization. During the acquisition of the TVs we encountered a strange issue: the stereoscopy settings varied noticeably from set to set. Because it is impossible to change these settings, it was necessary to inspect about fifty units before twenty with acceptable differences have been selected.

3.2 Software

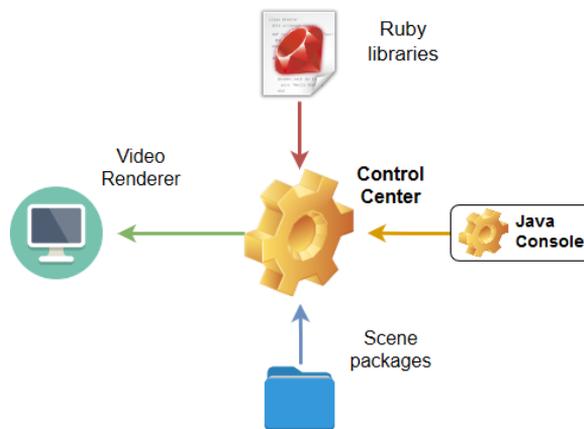


Figure 4

Modules of the LIRKIS CAVE control and visualization software

The LIRKIS CAVE visualization software can be divided into five modules (Fig. 4). The *Control Center* is the main one and provides the communication between all other modules. It also allows a user to control the system. It is located on the master computer and its other responsibility is to deliver scene and user input data to the *Video Renderer* modules. These run on the slave computers and render a scene to the LCD panels of the CAVE. What data to send to which renderer is decided by the *Control Center* on the basis of a dedicated configuration file. The number of *Video Renderer* instances that run on a slave computer is equal to the number of the panels connected to the slave. The *Video Renderer* is based on the OpenSG 3D graphics library.

The third module, the *Java Console*, can be seen as a graphical user interface of the Control Center. It communicates with the Control Center via a local network and allows a user to control individual computers in the cluster. Namely, the user is able to run or stop all instances of the Video Renderer and load, start or stop scene visualization. In addition, it allows configuring in-scene cameras of video renderers. This means that the whole CAVE can be rearranged to another shape and the displays can be added or removed. The console also displays a customized control panel for each loaded scene.

A *scene package* contains all the content necessary for the corresponding scene visualization (execution). The content consists of three parts: Ruby scripts, a graphic content and sounds. The scripts provide dynamic interaction between users and the scene. They are written in the Ruby scripting language (version 1.8.6). At least one script has to be present in each package. It is the *main scene script*, which serves as an entry point of the scene. Its task is to load all necessary elements and start the scene. The graphic content may consist of files representing various 3D objects, textures, 2D animations, transparent billboards and so on. The software supports several 3D formats, including 3ds, obj, vrml, and fbx. All files must be logically arranged in the folders of the package and the texture files must be stored in the same folder as the 3D model files. All sounds have to be stored in one folder and the allowed formats are wav, wma and ogg. The loading of the graphic content and sounds is managed by the scripts. Available scene packages can be accessed via the Java Console. The fifth software module is a set of *Ruby libraries*, necessary for the scripts execution.

To make a scene available in the CAVE, one must upload its package to a corresponding folder on the master computer. Then, the main script of the scene can be launched from the Java Console. After the launch, the Control Center copies the scene package to each slave computer for rendering. Each Video Renderer on a slave computer renders a different part of the scene from a different angle, according to the configuration of the CAVE and the position of the user with the OptiTrack marker.

3.2.1 Thread-based Scene Computing of 3D Objects

3D scenes and virtual environments may contain a large number of 3D objects with a high number of polygons. In addition, many of them have the dynamics (behavior) described by scripts. In the case of the LIRKIS CAVE, the scripts are written in Ruby and Ruby is an interpreted programming language. During a visualization of highly detailed dynamic scenes in the CAVE a noticeable latency has been observed between a command from an input device and the corresponding response in the virtual environment. Similarly, there were visible delays in an object behavior when collisions of the objects and changes in their movement had to be computed. To improve the response of the virtual environment, the *Thread - Based Scene Computing 3D* (TBSC 3D) has been

implemented. TBSC 3D distributes the execution of the scripts of the 3D objects into concurrently running threads. In each 3D scene script, threads are used to control different types of dynamic and static properties of 3D objects. These threads are divided into four categories (Fig. 5).

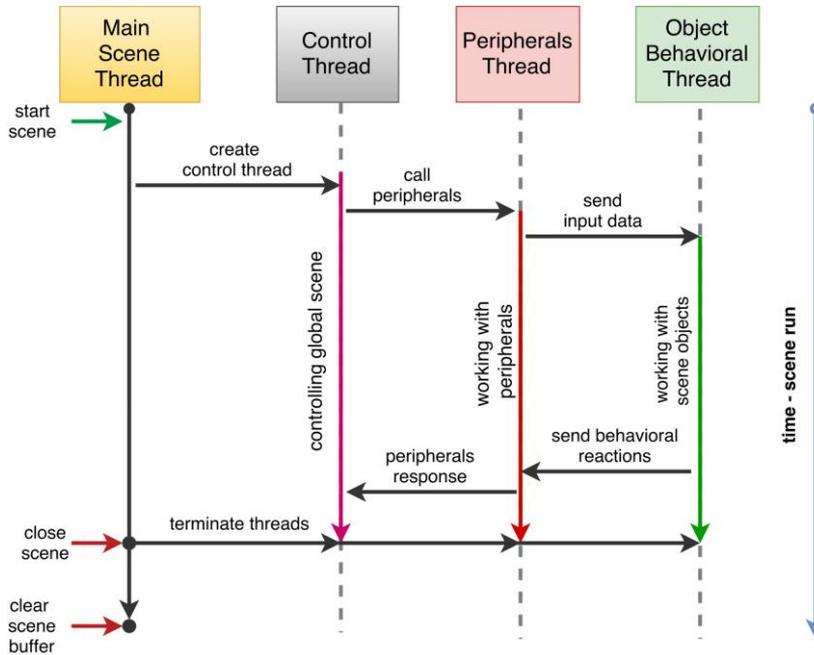


Figure 5

Parallel processing of 3D objects behavior in the LIRKIS CAVE

The first one is the *main scene thread*. It is created after a scene is started. Its role is to control and distribute tasks to other threads. After the scene is started, the thread works with global scene data such as the location of the objects in the scene, their size and visibility. It monitors all other threads, calls them and terminates them.

The second one is the *control thread*, which is created by the main thread. Its primary task is to manage input peripherals and assign them to 3D objects. The control thread calls the necessary number of peripherals threads and sends them the global information of the virtual scene. The number of the called threads depends on the number of connected input devices. When an input device connection is terminated, the control thread terminates its peripherals thread.

The *peripheral thread* is the third one and is called and controlled by the control thread. Its main task is to send data from an input device to the scene and to control it. Its significant feature is the ability to receive force feedback commands,

e.g. vibration signals, from object behavioral threads and to send the signals to the corresponding peripheral.

The last one is the *object behavioral thread*. It receives global information about scene objects and input signals for controlling 3D objects from other threads. After receiving the data, it deals with the behavior of 3D objects in the virtual environment, acquiring the data from the main scene thread.

As the results in Section 4.3 show, TBSC 3D noticeably increased the LIRKIS CAVE performance.

4 Performance Evaluation

Several experiments with various test scenes have been performed to evaluate the performance of the LIRKIS CAVE. Here, we present results concerning the impact of different 3D model and texture formats, a model complexity, lighting methods and the impact of the TBSC 3D utilization.

4.1 Model Format and Complexity

Because the user experience in a CAVE system depends significantly on the quality of the rendered scene content, the first two sets of experiments measured the influence of 3D model-related properties on the frame-rate.

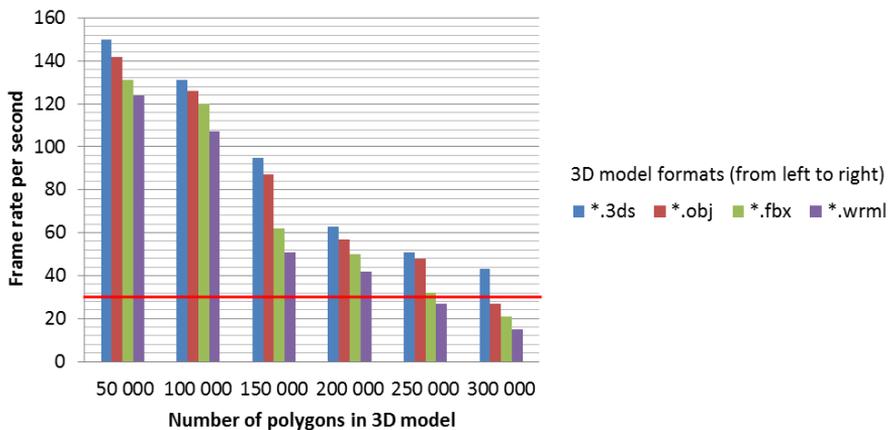


Figure 6

Influence of 3D model format and complexity on the frame-rate per second during visualization

The goal of the first set was to observe the impact of the 3D model format and complexity. By the complexity we mean the number of polygons of the model. The experiments were conducted on a scene with a corridor. First, a hollow corridor containing 50 000 polygons was used. Then, more details were added gradually, up to 300 000 polygons. The results can be seen in Fig. 6. Considering 30 frames per second as the lowest acceptable frame-rate, the models up to about 250 000 polygons can be used, but only in 3ds or obj formats. The differences between the formats were a bit surprising, but the success of 3ds and obj can be explained by a simpler structure of the 3ds format and obj being the native binary format of OpenSG.

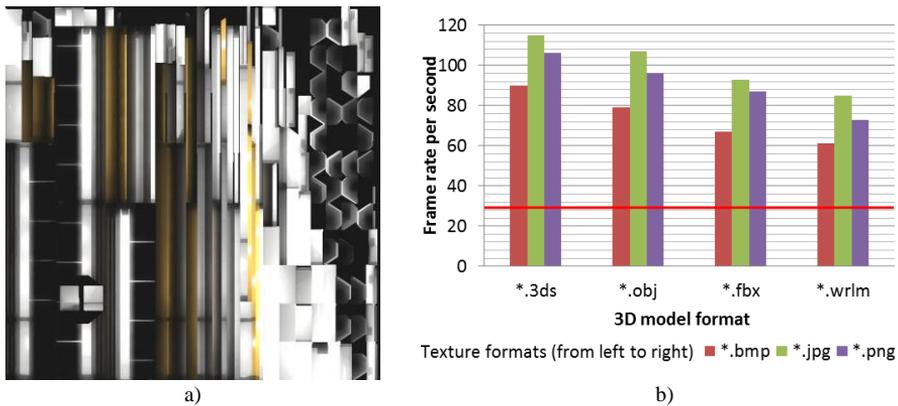


Figure 7

Texture used for texture format impact evaluation (a) and a graph showing the impact of various combinations of texture and 3D model formats in a 100 000 polygons scene on the frame-rate (b)

The second set tested the impact of texture format and was performed on a scene with a 3D object of 100 000 polygons. The used texture can be seen in Fig. 7 a). Its resolution was 1024 x 1024 pixels. With the resolution fixed, the primary factors influencing the frame-rate were the size of the texture file and the used compression method. Therefore, 3 formats were included: an uncompressed format (bmp), a format with lossless compression (png) and a format with lossy compression (jpeg). As expected (Fig. 7 b), the best scene fluency was achieved with the jpeg textures. However, the difference between the jpg and png is not significant, so png textures can be used when the high quality of the visual output, without compression artifacts, is required.

4.2 Lighting Effects Rendering

VR scenes combine visual effects and program logic for a more realistic user experience in a virtual environment [21]. The most common effects are lighting effects, which are applied to the surface of objects in the scene.

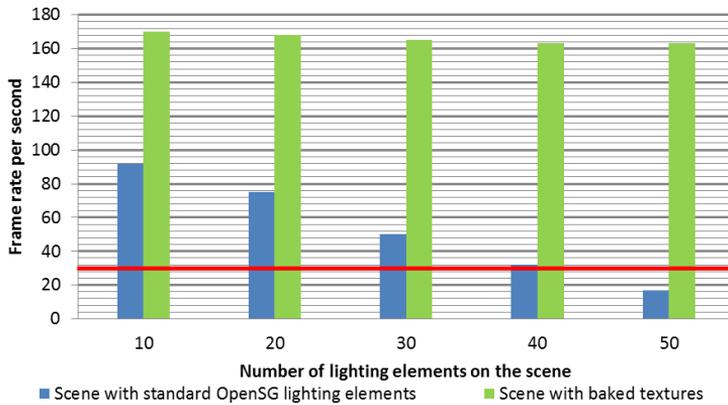


Figure 8

Rendering performance with real time light rendering and baked textures

Real time rendering of lighting effects, such as shadows and reflections, can cause significant drops in frame-rate. Fortunately, an alternative approach exists, where all lighting effects are generated beforehand, in a 3D modelling tool, and saved as a part of the 3D model texture. Such textures are then called baked textures. A typical scene with baked textures uses only the diffuse lighting. The performance impact of the diffuse lighting is minimal because it is constant in all parts of the scene. As Fig. 8 shows, there was only a small latency when the number of lights increased in the scene with baked textures. On the other hand, the real time use of the OpenGL lighting components affected the frame-rate significantly. The results in Fig. 8 were obtained during a visualization of a scene with 40000 polygons and 3D objects in 3ds format.

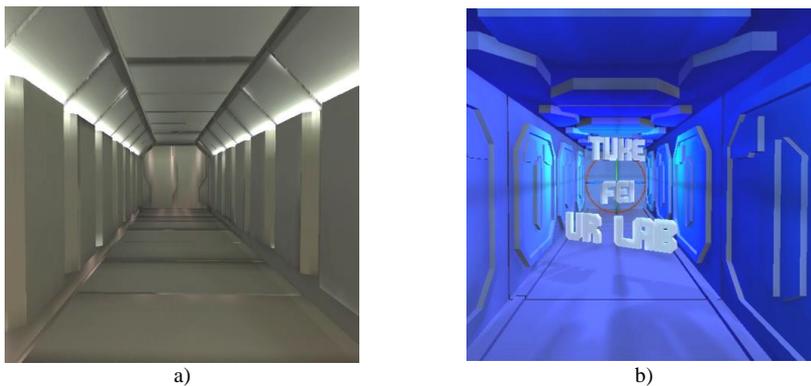


Figure 9

Visual difference between baked textures (a) and OpenGL standard lighting components (b) in similar scenes

However, the baked textures also have a significant disadvantage: They allow static lighting only. For example, lights affecting only a static surface, such as walls and ceilings, without any interaction with dynamic objects can be pre-rendered into baked textures (Fig. 9 a). But a light interacting with moving objects, such as the rotating text “TUKE FEI VR LAB” in Fig. 9 b), has to utilize the real-time light rendering.

4.3 TBSC 3D Performance Impact

The improvement achieved thanks to the thread-based scene computing 3D (TBSC 3D) has been measured using scenes with 3D models in 3ds and obj formats. These were chosen because of their performance in previous tests. The test was performed on the same scene as the first set in Section 4.1.

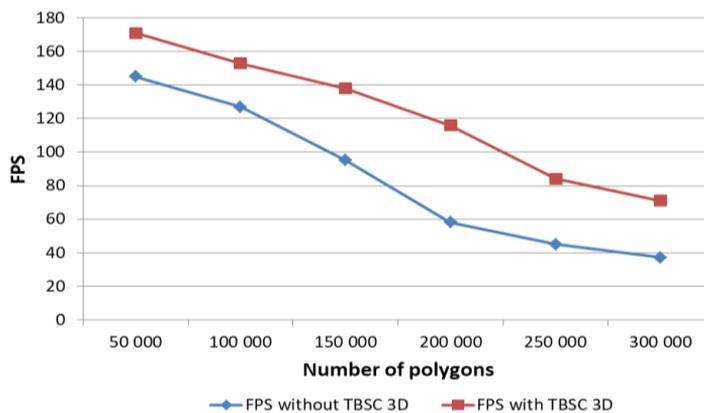


Figure 10

Rendering performance using 3ds models from 50 000 to 300 000 polygons with and without TBSC 3D

As the results in Fig. 10 show, the improvement is significant. FPS is noticeably higher, and the system response to the 3D object behavior is much more accurate. The scene does not produce duplicate data, which need high performance processing. Every problem is split to small tasks and only the necessary ones are computed. The results in Fig. 10 are for 3ds format, the ones for obj format are very similar.

5 Applications

Similarly to other CAVE systems, the LIRKIS CAVE is suitable for applications where a virtual environment is a satisfactory and cost-effective replacement of a real one. An example of an application for which the LIRKIS CAVE is fully prepared is a virtual inspection (Fig. 11 a) of vehicles, machinery or architecture under development. Thanks to its support of standard 3D formats a 3D model can be easily imported from the corresponding CAD software and visualized by the CAVE. Another advantage is the ability to host up to 5 inspectors at once.



Figure 11

Applications of the LIRKIS CAVE: a virtual inspection of a bus undercarriage and engine (a), and a wheelchair simulator prototype tested by a manual wheelchair user (b)

A major application currently under development in the CAVE is a wheelchair simulator, which will provide training for both manual and electric wheelchair users. Its development is divided into four phases. The first one is a modification of a real manual wheelchair, which will represent both manual and electric types. This phase includes lifting up the rear wheels just enough to rotate freely, and an installation of a gamepad and sensors. The gamepad will emulate the joystick of the electric wheelchair and the sensors will measure rear wheels revolutions for the manual wheelchair simulation. While other solutions, such as [22] or [23], place wheelchairs on a platform with rollers and measure the rollers revolutions, we decided to measure directly from the wheels and put all the sensors between the rear wheels. This is because any platform with rollers will block the floor LCD panels and cause a significantly elevated position of the chair with respect to the other persons inside the CAVE. The second phase, carried out simultaneously with the first one, is a development of a dedicated virtual environment, which will

resemble a real location and implement the wheelchair physics. The third phase will be a testing of the simulator by wheelchair users and domain experts. In the fourth phase we plan additional modifications of the simulator, such as an installation of motors to the rear wheels to emulate uphill and downhill movement of the manual chair or a replacement of the gamepad with an actual electric wheelchair joystick. Other modifications will be carried out according to the results of the third phase. The development is in its first and second stage now. A prototype of the simulator (Fig. 11 b) has been already implemented and evaluated by a wheelchair user.

The simulator will not only provide virtual training for local wheelchair users, but also combine and enhance features of other existing solutions. For example, [22] focuses on manual wheelchair users and utilizes a real wheelchair as in our case. On the other hand, [22] uses a VR headset and 27% of its users reported a motion sickness. We expect the motion (simulation) sickness to be less an issue, because CAVEs perform better as VR headsets in this aspect [2]. In the simulator [23] the users use their own wheelchairs, so both types can be simulated, but the VR environment is rather basic with the image backward-projected on just tree walls. Another study, [24], which compares the use of a classic LCD display and a VR headset, points out the importance of seeing the representation of the user's body during the simulation. This is provided naturally in the CAVE as the user sees himself.

6 Conclusion

The LIRKIS CAVE is an up-to-date immersive virtual reality environment with a unique compactness and portability. Next to the walls and ceiling, it also provides floor displays, which are often lacking in contemporary LCD panel-based virtual reality installations [1].

While the tests presented in this paper confirmed its ability to visualize fairly complex interactive scenes, the OpenSG software core is showing its age. This is particularly evident in scenes involving real time lighting effects. The most promising candidate for the new visualization software of the CAVE is the Unreal Engine 4⁹ (UE4) 3D game engine. This is because it provides two features the current LIRKIS CAVE software lacks: a support of the newest 3D graphics functionality and sophisticated tools for the scene preparation. In addition, it is free for non-commercial use and open source. The last feature comes in very handy as it is necessary to modify UE4 to be usable in the CAVE. The work on the modification is under way and we already tried to run multiple synchronized

⁹ <https://www.unrealengine.com>

UE4 instances in the CAVE. The approach is very promising; however, there are observable delays between renderings on individual displays, which have to be eliminated. We also consider adaptation of virtual reality collaboration platforms, such as VirCA [9] or its successor MaxWhere.

The future applications of the CAVE are in the context of cognitive infocommunications [8], primary in the inter-cognitive communication mode utilizing the sensor-sharing and sensor-bridging communication. They will primarily focus on the area of VR-based rehabilitation, which is considered in many contexts, e.g. the Parkinson disease [25]. It has been also proven more effective than traditional rehabilitation programs in cases related to the physical outcome development [26]. The aforementioned wheelchair simulator is only one of them. These applications will aim at different impairments and will implement gamification elements to motivate the trainees to reach planned goals. The interaction will take place between the trainee and the CAVE software (inter-cognitive mode), which will collect data from multiple sensors to assess the progress achieved (sensor-sharing) and to adapt the training process if needed (sensor-bridging). Their development will be based on the previous practical experience [27], [28], [29], gained during a collaboration with Pavol Sabadoš special united boarding school children with mental and physical disabilities in Prešov, Slovakia. Another interesting area is a visualization of programming-related theoretical concepts, such as linear logic [30]. And, as the CAVE is a power-hungry installation, we also plan to measure how different coding practices affect its power consumption. Here, we consider adapting approaches used for other devices, for example [31].

Acknowledgment

This work has been supported by the APVV grant no. APVV-16-0202 “Enhancing cognition and motor rehabilitation using mixed reality”.

References

- [1] T. W. Kuhlen, B. Hentschel: Quo vadis CAVE: does immersive visualization still matter?, *IEEE computer graphics and applications*, Vol. 34, No. 5, 2014, pp. 14-21
- [2] K. Kim, M. Z. Rosenthal, D. J. Zielinski, R. Brady: Effects of virtual environment platforms on emotional responses, *Computer Methods and Programs in Biomedicine*, Vol. 113, No. 3, 2014, pp. 882-893
- [3] Cs. Szabó, Š. Korečko, B. Sobota: Data Processing for Virtual Reality, *Advances in Robotics and Virtual Reality, Intelligent Systems Reference Library*, Vol. 26, Berlin Heidelberg: Springer-Verlag, 2012, pp. 333-361
- [4] B. Sobota, F. Hrozek, Š. Korečko, Cs. Szabó: Virtual reality technologies as an interface of cognitive communication and information systems,

- Proceedings of 2011 2nd IEEE International Conference on Cognitive Infocommunications, Budapest, 2011, pp. 1-5
- [5] A. Febretti *et al.*: CAVE2: a hybrid reality environment for immersive simulation and information analysis, Proceedings of SPIE - The International Society for Optical Engineering, 8649, 2013, art. no. 864903
- [6] T. A. DeFanti *et al.*: The StarCAVE, a third-generation CAVE and virtual reality OptIPortal, Future Generation Computer Systems, Vol. 25, No. 2, 2009, pp. 169-178
- [7] P. Valent: Forest visualization as a tool for decision support and forestry education, Acta Facultatis Forestalis, Vol. 56, 2014, pp. 49-64
- [8] P. Baranyi, Á. Csapó: Definition and Synergies of Cognitive Infocommunications, Acta Polytechnica Hungarica, Vol. 9, No. 1, 2012, pp. 67-83
- [9] D. Vincze *et al.*: A Novel Application of the 3D VirCA Environment: Modeling a Standard Ethological Test of Dog-Human Interactions, Acta Polytechnica Hungarica, Vol. 9, No. 1, 2012, pp. 7-17
- [10] P. Galambos, C. Weidig, P. Baranyi, J. C. Aurich, B. Hamann and O. Kreylos: VirCA NET: A case study for collaboration in shared virtual space, Proceedings of 2012 IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom), Košice, Slovakia, 2012, pp. 273-277
- [11] B. Lampert *et al.*: MaxWhere VR-learning improves effectiveness over classical tools of e-learning, Acta Polytechnica Hungarica, Vol. 15, No. 3, 2018, pp. 125-147
- [12] I. Horváth, A. Sudár: Factors Contributing to the Enhanced Performance of the MaxWhere 3D VR Platform in the Distribution of Digital Information, Acta Polytechnica Hungarica, Vol. 15, No. 3, 2018, pp. 149-173
- [13] B. Berki: 2D Advertising in 3D Virtual Spaces, Acta Polytechnica Hungarica, Vol. 15, No. 3, 2018, pp. 175-190
- [14] A. Csapo *et al.*: VR as a Medium of Communication: from Memory Palaces to Comprehensive Memory Management, Proceedings of 2018 9th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Budapest, 2018, pp. 389-394
- [15] T. Budai, M. Kuczmann: Towards a Modern, Integrated Virtual Laboratory System, Acta Polytechnica, Hungarica, Vol. 15, No. 3, 2018, pp. 191-204
- [16] C. S. Lanyi, S. M. Tolgyesy, V. Szucs and Z. Toth: Wheelchair driving simulator: Computer aided training for persons with special need, Proceedings of 2015 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Győr, 2015, pp. 381-384

-
- [17] N. Katajapuu et al.: Benefits of exergame exercise on physical functioning of elderly people, Proceedings of 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Debrecen, 2017, pp. 85-90
- [18] A. Gilányi, G. Bujdosó, M. Bálint: Virtual reconstruction of a medieval church, Proceedings of 2017 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Debrecen, 2017, pp. 283-288
- [19] T. Ogi, T. Uchino: Dynamic load-balanced rendering for a CAVE system, Proceedings of the ACM symposium on Virtual reality software and technology, ACM, 2006, pp. 189-192
- [20] M. Hudák, Š. Korečko, B. Sobota: Peripheral Devices Support for LIRKIS CAVE, Proceedings of 2017 14th International Scientific Conference on Informatics (Informatics'2017), Poprad, 2017, pp. 117-121
- [21] M. Wahlström et al.: CAVE for collaborative patient room design: analysis with end-user opinion contrasting method, Virtual Real, Vol. 14, London, 2010, pp. 197-211
- [22] L. Y. Sørensen, J. P. Hansen: A low-cost virtual reality wheelchair simulator, Proceedings of 10th International Conference on Pervasive Technologies Related to Assistive Environments (PETRA '17), Island of Rhodes, ACM, 2017, pp. 242-243
- [23] H. P. Mahajan et al.: Assessment of wheelchair driving performance in a virtual reality-based simulator, J Spinal Cord Med, Vol. 36, No. 4, 2013, pp. 322-332
- [24] A. Alshaer, H. Regenbrecht, D. O'Hare: Immersion factors affecting perception and behaviour in a virtual reality power wheelchair simulator, Applied Ergonomics, Vol. 58, 2017, pp. 1-12
- [25] A. Gilányi, E. Hidasi: Virtual reality systems in the rehabilitation of Parkinson's disease, Proceedings of 2016 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Wroclaw, 2016, pp. 301-306
- [26] M. C. Howard: A meta-analysis and systematic literature review of virtual reality rehabilitation programs, In: Comput. Hum. Behav., Vol. 70, 2017, pp. 317-327
- [27] B. Sobota and Š. Korečko: Virtual Reality Technologies in Handicapped Persons Education, In: Advanced in Information Science and Applications, Vol. 1, WSEAS, 2014, pp. 134-138
- [28] B. Sobota, Š. Korečko, P. Pastornický and L. Jacho: Education Process and Virtual Reality Technologies, A journal for information technology,

- education development and teaching methods of technical and natural sciences., Vol. 1, No. 1, 2016, pp. 286-291
- [29] B. Sobota, et al.: Development of handicapped children communication skills using touch user interface; Proceedings of Information Technology and Development of Education ITRO 2015 - Zrenjanin : University of Novi Sad, 2015, pp. 49-52
- [30] V. Novitzká, D. Mihályi, V. Slodičák: Linear logical reasoning on programming, Acta Electrotechnica et Informatica, Vol. 6, No. 3, 2006, pp. 34-39
- [31] J. Saraiva, M. Couto, Cs. Szabó, D. Novák: Towards Energy-Aware Coding Practices For Android, Acta Electrotechnica et Informatica, Vol. 18, No. 1, 2018, pp. 19-25