This is a preprint of the chapter to be published in L. Jain, et al eds. *Fusion of Smart, Multimedia and Computer Gaming Technology: Research, Systems and Perspectives*. The final version is available in the Springer Digital Library: http://link.springer.com/chapter/10.1007%2F978-3-319-14645-4_7.

3D Virtual Worlds as a Fusion of Immersing, Visualizing, Recording, and Replaying Technologies

Mikhail Fominykh[1], Andrey Smorkalov[2], Mikhail Morozov[2] and Ekaterina Prasolova-Førland[1]

[1] Program for Learning with ICT, Norwegian University of Science and Technology, Norway
{mikhail.fominykh, ekaterip}@ntnu.no
[2] Multimedia Systems Laboratory, Volga State University of Technology, Russia
{smorkalovay, morozovmn}@volgatech.net

Abstract. This chapter discusses 3D virtual worlds as a fusion of technologies for serious applications. A variety of new trends in the area is considered along with original findings, exemplified with particular approaches and features designed. The goal of this chapter is to evaluate and envisage what benefits can be acquired if such trends and features are fused under a single platform. Specific trends considered in this chapter include increase of the sense of immersion with virtual reality devices, simplification of control with joysticks and mobile devices, working with large amounts of multimedia content, and capturing activities using virtual recording. The discussion is supported with the literature review and evaluation of the features developed for vAcademia virtual world. A particular focus of the chapter is on the discussion of how the new properties of 3D virtual worlds can increase their efficiency for serious applications, enable new scenarios of use, and improve user experience.

Keywords: 3D virtual worlds, technology, collaboration, motion capture, head-mounted displays, natural navigation, virtual recording, vAcademia.

1 Introduction

The great potential of three-dimensional virtual worlds (3D VWs) for collaboration and other serious applications such as education, healthcare, military, and business have been investigated and reported [1-6]. Most of them are considered to exploit advantages of the 3D VWs, such as low cost and high safety, 3D representation of users and objects, and interaction in simulated contexts with a sense of presence. At the same time, this technology has certain limitations that led to a number of challenges for applying it in various domains.

One of the most serious challenges in adapting 3D VWs for serious applications is the lack of features that professionals use in their everyday work [7]. Such applications often involve cooperation and co-construction, which need to be supported and often requires tools that do not exist in most 3D VWs [8].

Advanced human-computer interfaces can increase the value and transferability of virtual experience, as human mind is closely connected to the sensory-motor experience [9]. The theory of Embodied Cognition suggests and stipulates that human mind is defined by the body and specific context [10]. Therefore, embodied experience of activities in a 3D VW is closer to the real-world experience. This makes involvement of the body in the process of interaction with a 3D VW important.

In addition to the 3D VWs, various other Virtual Reality (VR) technologies that involve human body in the interaction are being developed or adapted serious applications. However, the cost is becoming a limiting factor in most cases. Industry, military, and healthcare are named as the only areas where these technologies are starting to be used [11,12]. Thus, the development of low-cost and off-the-shelf solutions is necessary.

This chapter describes an original approach that addresses the challenges above mentioned. This approach is based on a fusion of several technological solutions, schemes, and devices in a single system. Three particular components suggested for fusion fall in the following topics: natural navigation with high level of immersion, displaying large amount of meaningful media content, and virtual recording and replay. The main application domain of the fusion of technologies proposed in the chapter is collaboration. Section 2 of the chapter contains the overview of literature for each of the topics. Section 3 provides the original design of several features in each of the topics. Section 4 contains the results and discussion of the attempt to fuse the features with the goal of improved user experience or enabling new use cases and applications. Section 5 provides an example of such new scenarios. Section 6 concludes the chapter, outlining the directions for future work.

2 Background

2.1 Immersion and Tracking Technologies

Modern 3D VWs are usually built on a type of VR technology – a desktop VR, as they can be accessed on a standard desktop computer. In the past, they have been built on various technologies and accessed through different interfaces [13]. Another type of VR technology that is used in 3D VW can be called Immersive VR, as it goes beyond desktop and extends audio-visual experience with more natural navigation and control, thus increasing the sense of presence and immersion [14].

Various VR systems have a number of unique features and applications. The major change in the development of the VR in the recent years is the appearance of affordable devices. Therefore, their application area is becoming much wider. A number of studies have been done in this area outlining advantages and limitations of these technologies, for example, as working and training environments [15,16]. The main arguments for its use are that 3D environments are engaging as media, and that the use of 3D rather than 2D media facilitates comprehension, exploiting the natural capabilities of humans to interact in 3D space [17-19].

Relatively few simulations have been built for immersive VR, though the evidence of use in very high-value training (vehicle simulation, such as aircraft simulators or military training) is compelling [20]. There has been extensive study of the impact of 3D immersive visualization on user behavior, lower-level task performance, and comprehension of data. A key characteristic, arguably the one that motivates the use of immersive VR in high-value training, is that participants tend to react as if the scene they were seeing were real. That is they behave in a way that is similar to their behavior in a comparable real situation [21].

This behavioral response itself distinguishes an immersive VR from other media, even desktop VR, as when immersed the participant can actually act to a limited extent, as they would in the real world (e.g., by moving away from a threat). In desktop VR or other media, this capability is severely limited by the form and structure of the interface. There have been several studies of the relative capabilities of immersive VR and desktop VR systems on comprehension. Generally, the results indicate that for survey-like tasks a desktop VR may be preferred, but for interactive, exploratory tasks, immersive VR is preferred [22,23]. However, these and other studies have not provided a complete requirement analysis that can predict tasks for which immersive VR environments are superior. This work is continuing with larger studies showing for constrained tasks, which features of immersive VR are contributing to performance differences [24].

In the following, three particular technologies are presented as a related work for this chapter.

Motion-Tracking. Low-cost motion-sensing technologies such as Microsoft Kinect, Nintendo Wii Remote, and Playstation Move provide researchers and practitioners with new opportunities for improving virtual experience. One of the first affordable motion-tracking devices was Microsoft Kinect. It is a low-cost motion sensing input device that is able to capture one or two humans [25]. The device consists of a video camera, depth camera, and an IR camera.

Multiple examples of applications and systems built with motion tracking devices can be found. They include a low-cost alternative for interactive whiteboards and multi-touch teaching stations [26], automatic camera control system for presentations [27], anthropometric measurement system [28], and natural user interfaces in volume visualization [29].

Head-Mounted Displays. Head-mounted display (HMD) is a type of on-body VR devices that is worn on the head and has a display in front of the user's eyes [30]. Most of these devices consist of a display and a tracking system. It allows much greater immersion, as the user can control the direction of the view in a 3D VW in exactly the same way as in the physical world – by turning the head. The displays of HMDs have larger field of view and provide a stereoscopic image, making the experience more believable.

The Oculus Rift is an HMD device that has a 7-inch diagonal viewing area and 1280 to 800 resolution split between both eyes, yielding 640 to 800 per eye [31]. The device has a three-axis gyroscope, which senses angular velocity, three-axis magnetometer, which senses magnetic fields, and three-axis accelerometer, which senses accelerations, including gravitational [31]. A good field of view, stereoscopic vision, and fast tracking that are promised by the developers created huge expectations [32].

HMDs are also used together with motion-tracking devices, improving the accuracy of tracking and the overall performance of VR systems [33].

2.2 Media Content in 3D Virtual Worlds

This section contains the necessary background on the topic of using large amount of graphical content in 3D VWs and related work.

Large Amount of Graphics in 2D Platforms. Nowadays, many examples can be found of large amounts of textual and graphical information that needs need to be on the user's display at a time. This is often required in synchronous collaborative activities, when users have to be aware of what the others are doing. Such activities are typical for collaborative work. However, in practice software limitations do not allow implementing them to the full extent on either 2D web-conferencing platforms or in 3D VWs.

Modern web-conferencing platforms provide a shared workspace, means for content presentation, and various forms of interaction [34,35]. However, it should be noted that in most cases, this technology is used for simple delivering of information (e.g., a presentation). Well-organized web conferences are often interactive (e.g., questions, comments, and discussion) and collaborative (e.g., mind-mapping, posting notes on slides, and using a shared workspace) but, a single screen (with the content provided by the presenter) is used in most cases. The example of using breakout rooms illustrates synchronous collaborative activities in web-conferencing platforms, "allowing small groups to undertake tasks and bring outputs back to a plenary full group discussion" [36]. In this case, each group of users still works with a single shared workspace/screen, and the results are discussed one by one.

The implementation of such scenarios in 2D does not require large amounts of textual and graphical information on the user's display. The reason, however, is not in the lack of demand, but most likely in the technological limitations. When a collaborative scenario requires displaying multiple workspaces side by side, they can be located as tiles, but even with large monitors, this approach is rarely convenient. Alternatively, multiple workspaces can be set to different sizes (e.g., a moderator can enlarge them when required [37]).

Large Amount of Graphics in 3D Virtual Worlds. 3D VWs allow building environments that can accommodate multiple workspaces or virtual screens and convenient switching between different points of view (i.e. observing multiple screens or focusing on one). Natural 3D navigation or camera navigation and zoom can be used in such settings. This allows installing multiple virtual screens, for example, one with presentation slides, another with video or text notes. These affordances of the 3D space allow implementing active collaborative scenarios too, providing each user with personal virtual screen or several groups with screens.

However, in practice, existing 3D VWs allow using very limited numbers of virtual workspaces for collaboration on graphical content. For example, on some platforms, an environment may accommodate 2–5 virtual screens within the visibility area. This allows conducting several types of collaborative activities, including meetings and presentations. However, conducting activities that require active collaborative work with media content remains problematic. Another common practice is creating many screens with static images. This allows creating virtual galleries which can be explored collaboratively and discussed by a group of people, but without active involvement in creating or modifying the content displayed. In such a way, the possibilities for conducting collaborative activities are also limited.

Many 3D VWs offer the "virtual screen" functionality. However, the features of such screens often have inferior implementation because of performance limitations related to the use of CPU for processing images. For example, the 3D VW "Sametime 3D" built on OpenSimulator platform has a tool for working with sticky notes. However, the notes can only be placed on special square slots, their size is constant, and there is no possibility to use any other tools on the same screen (e.g., drawing). These limitations obstruct the effective use of the tool.

Technological Approaches to Processing Large Amounts of Graphics. Processing large amounts of images in 3D VW is mostly required when working on serious tasks, such as collaborative work. In other tasks, displaying images, video or animation is also often required; however, the amount of content is smaller. Usually, an image is calculated on a CPU on client side (e.g., in Second Life™ and Blue Mars™) or server side (e.g., in Open Wonderland™) and then loaded into the stream-processor memory as a texture.

Thus, the use of dynamic 2D images in existing 3D VWs is very limited. One of the most popular platforms – Second Life even has a hard limit of five dynamic images that can be rendered for one user at a time. This fact can be technically explained by the use of CPU for generating images. This technical limitation does not allow implementing some of the collaborative scenarios to their full potential.

Stream Processors. Stream Processors (SPs) are specialized processors characterized by a very high data parallelism [38]. SPs are most widely applied in graphics adapters, and therefore, their main tasks are related to processing 3D graphics, such as a high-efficiency rasterization of polygons with texture mapping, fast affine transformations of the vertex data flow, interpolation of scalars, vectors and matrices on the surface of polygons, and calculation of lighting.

Due to the focus on the highly parallel computing tasks in 3D graphics, these devices have many hardware constraints [39,40] and their cores have relatively simple architecture [41]. Therefore, most of the classical algorithms that can be executed on CPU cannot be executed on SPs without modification.

2.3 3D Virtual Recording and Replaying

Capturing Virtual Activities. Capturing such activities is a complex task from the technological point of view, as part of the information is usually lost. Using other technologies, such as video recording of face-to-face activities or recording of web conferences is not as complex. However, these technologies change the context of activities. They do not provide the same level of immersion, a possibility for collaborative work, or a method for further developing the 'crystallized activities', except for annotating them.

Currently, 3D VWs are also used for capturing activities, but usually only as 'flat' 2D video, which eliminates many advantages of the technology, such as sense of presence [42]. For example, this approach is used in Machinima – collaborative film making using screen capture in 3D VW and games [43].

Related Work on Virtual Recording. The demand for methods supporting asynchronous activities in 3D VWs and creating content out of synchronous activities was acknowledged as early as in the late 90s, for example, by developers of CAVE and MASSIVE systems. MASSIVE-3 supported a mechanism called 'temporal links', which allowed "real-time virtual environments to be linked to recordings of prior virtual environments so that the two appear to be overlaid" [44]. CAVE Research Network soft system had an application called Vmail which supported recording of an avatar's gestures and audio together with surrounding environment [45].

Another example is the system called Asynchronous Virtual Classroom or AVC. The developers were focused on solving the problem of time-sharing in distance learning. AVC allowed users to watch a video image of a certain lecture and to control it, while software agents were playing some of the displayed participants and created a presence effect [46].

Later, N*Vector (Networked Virtual Environment Collaboration Trans-Oceanic Research) project was focused on developing a virtual reality technology for overcoming time-zone differences and time management problems. Within the project, there were developed three approaches to support annotations for asynchronous collaboration in virtual reality. These approaches included: VR-annotator – an annotation tool that allows collaborators to attach 3D VR recordings to objects; VR-mail – an email system built to work entirely in virtual reality; VR-vcr – a streaming recorder to record all transactions that occur in a collaborative session [47].

More recently, an Event Recorder feature was implemented (however, not developed further) within the Project Wonderland (later, Open Wonderland™). Event recorder provides an implementation of the recording and playback of the 'events' caused by activities of users or agents in such a way that during playback a user is able to view the activities that those events caused. It achieves this by recording the 'events' in the world into an external form that can then be replayed to all the users that are engaged in the world. The invention also includes a mechanism to record the current state of a VW to use it as the starting point to play back the recorded events. In addition, a specialized plugin mechanism records the state and transitions of objects, which cannot be represented in terms of events, for example, the audio conversations.

A recovering feature is being developed for the new project by High Fidelity. Their new 3D VW is under alpha testing, but the fact that a new and ambitious project implements such a feature demonstrates its potential utility for the modern community.

All mentioned projects were contributing towards developing technological solutions for capturing activities in 3D VWs or other similar environments. As presented above, creating content out of synchronous activities is an important in-demand task. However, there is a clear possibility for improvement, as the application potential was not yet realized.

3 Fusion Design of a 3D Virtual World

This section presents the design of several features that enhance the technology of 3D VWs to improve user experience and enable new application domains. vAcademia 3D VW has been used as a platform for experimental design, implementation, and evaluation. Several main general-purpose design and development concepts of the vAcademia platform are summarized below, while more details are invent in an earlier work [48].

Most of the vAcademia components were developed from scratch. However, some well-known libraries and components were used for programming the networking, image processing, and sound. The graphical engine of vAcademia was developed specially for the project based on OpenGL. For implementing interactivity in vAcademia, a Jscript-based scripting language vJs was developed [49].

vAcademia has a client-server architecture (Fig. 1) and works on Windows operating system. The main components of the system are briefly described below.

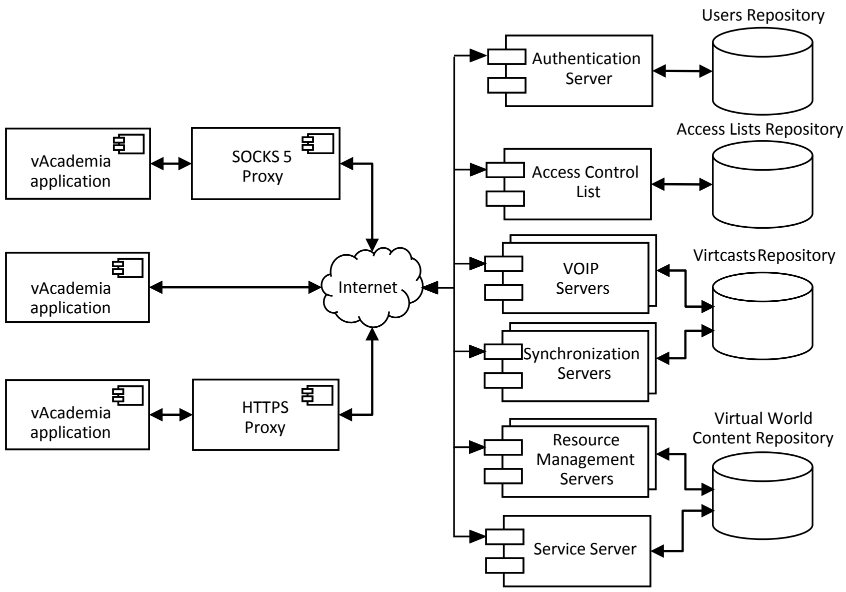

Fig. 1. vAcademia system architecture

- vAcademia application – client software that visualizes the VW and provides all the functionality.
- vAcademia servers – a scalable system that supports network interaction and consists of multiple servers, including Authentication Server, Resource Management Servers, Synchronization Servers, VOIP servers, and Service Server [48].
- Access Control List (ACL) – a system that manages user access permissions for the resources, such as scheduling and accessing virtual sessions.
- Virtual World Content Repository – a storage system for the 3D world data (geometry and textures) and user objects.
- Virtcast Repository – a storage system for 3D recordings.

3.1 Large Amounts of Graphical Content in a 3D Virtual World

In this section, an original approach towards displaying large amount of graphical content in a 3D VW is presented. It is based on the Stream Processors Texture Generation Model that includes a mathematical and a programming models.

Mathematical Model. In order to formalize the domain, a mathematical model of image processing on SPs has been developed, based on the specifics of the SP architecture and hardware constraints. The mathematical apparatus of processing 3D graphics on SPs was simplified to focus only on processing images [50,51].

The model introduces the basic terms, objects, and their transformations. An image is represented in the RGBA format.

$$U(x, y) = \{f_R(x, y), f_G(x, y), f_B(x, y), f_A(x, y)\} \tag{1}$$

In this equation, $f_R(x, y)$, $f_G(x, y)$, $f_B(x, y)$, $f_A(x, y)$ are discrete functions defined by tabular procedure and corresponding to the color channel with values in the range [0, 1].

The result of transformation G of image A based on image B is a modification of the image function (1):

$$R = G(A, B, x, y) \tag{2}$$

A geometrical figure is defined as a set of two-dimensional vectors of vertices V, a set of indexes of vertices F, and a color in {r, g, b, a} format.

$$S = \{V, F, \{r, g, b, a\}\} \tag{3}$$

Rasterization is a transformation of a geometrical figure that has an image as a result.

$$U(x, y) = G_R(G_P(S, M_P)) \tag{4}$$

In this equation, G_R is a rasterizing transformation, M_P is a projective matrix, and G_P is a projection transformation.

The result of a projection transformation G_P is a projected figure.

$$S_P = G_P(S, M_P) \tag{5}$$

In addition, the mathematical formalization was applied to the configurable functionality (of the SPs) that was suitable for image processing tasks. As one of the specifics, SPs have some configurable (not programmed) functionality, which was formalized and linked to the programmed functionality. This included a mechanism for texture sampling, color mask, hardware cut of the rasterization area, hardware-based blending of the source image and the rasterized image.

The suggested model allows defining the color of each pixel of the resultant image separately as the resultant image is the function of pixel coordinates. This makes it possible to calculate parts of an image or even single pixels instead of the whole image. The general nature of the model allows comparing the efficiency of different approaches to any specific image processing task, such as dynamic texture generation, using the formula for image generation time:

$$T = T_C + T_{TR} + T_1 * W * H \tag{6}$$

T – time of processing the whole image

T_C – compilation time of the image processing program (shader) that performs a transformation

T_{TR} – time of preparation for transformation

T_1 – time of calculating the color of one pixel, i.e. calculating (2)
W and H – width and height of the processed image

Programming Model. The mathematical model presented above was used as a base for the programming model and architecture based on four main objects (Texture, Drawing Target, Filter, and Filter Sequence) and a limiting condition <β>.

Texture is an image in format (1) stored in the SP memory. The image can be loaded to a texture and obtained from it asynchronously (using extension GL_ARB_pixel_buffer_object [52]) with the possibility to check data availability, reducing the expenses of the communication through the data bus.

Drawing Target is an object that defines the resultant image, color mask, and settings of other configurable features of SPs.

Filter is a subroutine with the main function GetColor that defines the image transformation (2) and returns the color of a point for the given coordinates. This function has predefined and custom (user-defined) parameters. The GetColor function is defined in GLSL-like language (OpenGL Shading Language) extended with additional functions for image processing. Its parameters are strongly typed, have unique names, and are defined in the form of XML. GetColor provides the multipurpose model, allowing, for example, to program texture generation as a method of procedural materials and as a method of full periodical texture regeneration.

Filter Sequence (FS) is a sequence of Filters with parameters to be used for complex transformations.

<β> is a limitation introduced to the model for securing independent parallel processing of any image blocks.

The models presented above have been used for the original modification of the Discrete Wavelet Transformation (DWT) algorithm for SPs and an algorithm for rasterizing attributed vector primitives on SPs [53].

Implementation in vAcademia: Methods and Tools. Interactive virtual whiteboard (VWB) is the main tool (or a container of tools) for collaborative work on 2D graphical content in vAcademia. The rationale behind the design of the VWB was the following. The system should remove the current technological limitations on the number of virtual displays, and allow new working scenarios to be realized in a 3D VW. In addition, it should allow displaying images from multiple sources on the same virtual display at the same time. The following requirements were set. The system should support up to 50 VWBs in the visibility scope of a user. Each VWB should be able to display images from two sources simultaneously and independently. Users should be able to setup VWBs and use them VWBs collaboratively. These major requirements are based on a preliminary evaluation of the needs of common practice.

The design of vAcademia VWB and the tools for collaborative work on graphical content is based on using a dynamic texture with two independent layers which are combined in one static texture when rendering. Generally, the lower level is used for permanent content, while the upper level – for temporary content that can be changed or erased. This allows having a temporary dynamic figure above several different lower-layer contents (such as slides or sticky notes). In other situations, the lower layer may contain a dynamic image, while the upper layer may remain unchanged (such as for commenting on a video when the comments must remain visible above all the frames).

A number of collaborative tools have been developed for the VWB in vAcademia [54]. The tools can be classified into three groups by the method of generating the resultant image. In the following, the methods and associated groups of vAcademia tools are presented, including technical requirements, design, and implementation. The requirements for the methods are based on the results of preliminary testing.

Sharing Changing Blocks. The following requirements were set for this method. The system should be able to support up to three simultaneously working image-sharing processes in any location, with up to 1024 to 1024 pixels resolution each, and with a frame rate at least 5 FPS (frames per second). The frame rate is considered as a requirement for the performance of the system only. However, it also depends on the Internet connection speed, and therefore meeting the requirements does not guarantee the desired frame rate (in case of low connection speed).

Design: The Sharing Changing Blocks method of generating the resultant image is based on an algorithm for DWT for image compression with quality reduction, which is adapted for SPs. It identifies the changing rectangular parts of the image by using occlusion query [55]. The algorithm is based on the filter cascade scheme that allows implementing the forward DWT in one pass. The method uses the lower layer of the VWB and processes the dynamic image. Quality may be adjusted by DWT coefficients.

Implementation 1: Sharing an application window allows to share the content of any window. The window is translated even if minimized or overlaid. The control over the window can be given to any user in the location. This feature can support various scenarios when a live demonstration of third-party software is required. For example, a hands-on tutorial on a specific application can be conducted with sharing its window. Another example is a synchronous work with a specific software application (Fig. 2).

Fig. 2. Sharing application window (Scratch programming language in the center)

Implementation 2: Sharing screen area allows sharing any square area on the desktop, which may contain any working applications. This tool can be used in a session when the use of multiple software applications is required to be shared. For example, a part of the screen can be shared to demonstrate contents using different applications, e.g., a web browser or a video player.

Implementation 3: Sharing a web-camera image allows sharing the web-camera image and adjusting its size. This tool can be used for building trust in the beginning of a distance collaborative session, e.g., for introducing the participants. Each of them can use a personal VWB to share the image from the web-camera. Another example is a scenario when hand gestures or facial expression need to be shared.

Sharing Attributed Vector Figures. The system was required to support simultaneous drawing or typing on up to 25 VWBs with the average performance degradation less than 15% and peaking performance degradation less than 25%.

Design: The Sharing Attributed Vector Figures method of generating the resultant image is based on triangulation of vector primitives with attributes for one- or two-way rasterization. Displaying (both typed and inserted) text is implemented by using font textures, rasterizing each symbol on demand. The method uses the upper layer of the VWB and processes the dynamic image.

Implementation 1: Drawing figures and typing text allows to draw, erase, copy, and paste several types of geometric figures and text on a VWB. Drawing and typing actions from the time of cleaning the VWB can be undone. The undo depth is unlimited, which would be a performance prohibitive operation if using CPU only. This tool can be used in multiple collaborative scenarios when drawing and typing is required, for example, taking notes during brainstorming session, creating mind maps, annotating slides, and sketching prototypes.

Implementation 2: Inserting text tool allows inserting text from the clipboard to a VWB as long as it fits its size. This feature can be used together with drawing figures and typing text, allowing participants to work in third-party applications and share results on the VWBs.

Processing Static Images. The requirements for the method were the following. The system should support simultaneously changing static images on five VWBs within the visibility area. The average performance degradation should be less than 10% and peaking performance degradation less than 15%.

Design 1: The Processing Static Images method of generating the resultant image is based on resizing the source image, applying Filters, asynchronous unpacking, and uploading it. The method uses the lower layer of the VWB and processes the static image.

Implementation 1: Slideshow allows opening a PDF or PPT presentation on a VWB and navigating the slides forward and back. This tool allows replicating meeting settings that are commonly used for presentations. As opposed to the typical physical environment, a presenter can use two or more screens for slides, and other participants can upload their slides to personal VWBs. The main advantages of vAcademia's VWB over similar tools in other 3D VWs are that a large number of screens with slides can be deployed in a single virtual location and that the tool is integrated into the system.

Implementation 2: Area print screen allows displaying any square area on the desktop. Image quality can be adjusted. This tool can be used in collaborative activities when a part of the desktop needs to be presented to all the participants. For example, this can be the result of an individual assignment which needs to be discussed. Another common situation can be the need to present a quick snapshot that fixes a status of a working application, enabling to compare with the next status later.

Implementation 3: Image insert allows inserting from the clipboard, from Resource Collection of vAcademia, and from applications (drag-and-drop). Image quality can be adjusted. This tool provides flexibility that is necessary for selecting working material on demand and on the fly.

Design 2: Processing static images from 2D scene is based on a FS of a high number of alpha-blending Filters with different source images, blending settings, and hardware scissors. A 2D image or a rasterized image of a letter is taken as an input parameter.

Implementation 4: Backchannel allows displaying text-chat messages on a VWB. An additional tab appears in the text chat of all users in a location when a backchannel tool in used on a VWB. Messages in this tab immediately appear on a VWB. Backchannel is a powerful mechanism for improving the contact between the speaker and the audience that is becoming increasingly popular and taking multiple technological implementations [56]. Such a simple interaction mechanism can be considered more valuable in a virtual environment, as the non-verbal interaction and awareness mechanisms are very limited in such environments.

Implementation 5: Sticky Notes tool allows placing text or graphical notes on a VWB. It is a full-scale tool for collaborative work and can be used in a wide range of activities, including brainstorming, project-work discussions, and SCRUM sessions [57]. Sticky notes in vAcademia can be placed on any VWB after activating the special mode (Fig. 3). Any user can create notes, edit them, move on the surface and between VWBs, remove own notes, and see the actions of other users in real time. The design of the tool has been presented earlier [58].

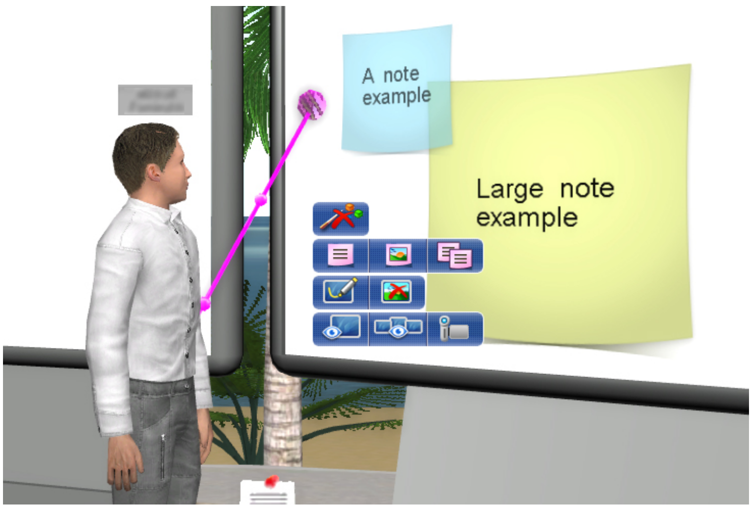

Fig. 3. Examples of sticky notes and the tool's context menu

3.2 Beyond desktop – Immersive technologies

Capturing Real-life Activities with Motion Tracking for a 3D Virtual World. The general motivation for integrating a motion-tracking system as an input device for a 3D VW is providing users with a possibility to control their avatars with natural gestures. Our specific motivation for designing the vAcademia-Kinect prototype lies in making the users able to conduct meetings, presentations, and lectures in the physical and in the virtual world at the same time, controlling their avatars and media content on the screen with natural gestures [59].

vAcademia-Kinect first prototype. The virtualizing real-life activities mode interface is implemented using *vAcademia Scripts* (Fig. 4). The Scripts initiate the beginning and the end of this mode. In addition, the scripts provide the Kinect-related information, including the skeleton in 2D and the recognition status of each body part.

The Scripts interact with the Animation library of the graphic engine through the Script Executing Library. The Animation library interacts with the Kinect plugin, and based on the data from it, controls the user's avatar using Cal3D (Fig. 4).

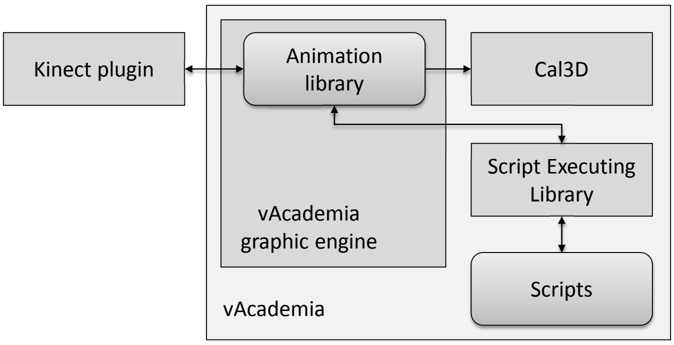

Fig. 4. Virtualizing real-life activities mode interface of vAcademia

vAcademia uses Cal3D library for implementing the skeleton avatar animation [60]. It allows to control the skeleton of an avatar by constantly and automatically recalculating its polygonal 3D model to match the current state of the skeleton. In the first prototype of the system [61], the Animation library of vAcademia requested a set of key points of the user's body from the Kinect-support plugin. If the user was recognized by Kinect, Cal3D bones of the vAcademia avatar are oriented according to the key points. In the general case, the user considered to be is standing, which is defined as a 'standing mode'. The system also supported a 'sitting mode', when the motion-capture data were used only for orientating arms and head.

In addition, Kinect provides the status of each key point of the presenter's body (not recognized, presumably recognized or recognized) as well as indicates if the body is cut in one of four directions by the camera visibility area. In order to eliminate unnatural poses, the skeleton was divided into five major parts (arms, legs, and head) to process the Kinect data for them separately. If a part of the skeleton remains unrecognized adequately for 0.2–0.5 seconds, the system sets it into the default state.

vAcademia requires and actively uses one CPU core. Other cores may be also used, but less actively (10–30%), except for the process of the first load of the VW data. Kinect requires a dual-core CPU, but uses only one core, as the second is reserved for the application that uses Kinect data. These settings define the requirements for the computer to run the system designed.

Although the evaluation has not been systematic so far, it allowed us to improve many characteristics of the system. Several evaluation sessions were conducted in several physical auditoriums with different configurations and lightning. The major evaluation data source has been an interview with the presenter. The following feedback on the technological setup is the most common.

The system applies too many restrictions on the movements of the presenter. First, the presenter has to stay in the area in front of the Kinect device, and the space between them should be clear. Second, the presenter has to face the device and not turn. Third, the presenter has to use only one hand for pointing at the screen. Fourth, the presenter has to avoid gestures that are difficult to recognize by the users of the 3D VW.

Increasing the sense of immersion with Head-mounted displays. The motivation for integrating an HMD with a 3D VW is to improve the user experience, allowing a more natural navigation in the environment and increasing the sense of immersion. vAcademia-Oculus Rift HMD prototype improves the immersion to the virtual environment and therefore increases the engagement by providing a more sophisticated visual and navigation experience. A particular key difference such a fusion makes is the mobility of the HMD user in comparison to the stationary PC.

vAcademia-Oculus Rift prototype. The development of a mode that supports Oculus Rift HMD has been done according to the Software development kit (SDK) provided by Oculus VR. However, an original approach has been adopted to increase the level of system performance. The standard scheme for Oculus Rift support contains three rendering passes. First, the image for the left eye is rendered, then the image for the right eye, and then the images are combined by a Filter transformation when displayed on the screen (Fig. 5).

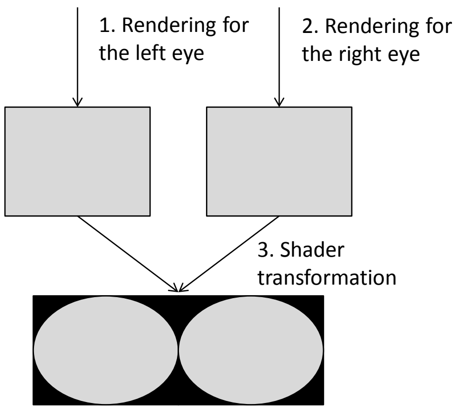

Fig. 5. Standard Oculus Rift support scheme

The new approach applied in vAcademia reduces the number of rendering passes to two at the cost of applying a more complex Filter (see the Programming model in Section 3.1) that transforms two images simultaneously for both eyes. Rendering of left-eye and right-eye images is conducted in parallel on the same surface. Then, the transformation is done using a Filter that is more complex than in the standard scheme. It enables transformation of the two areas for both eyes simultaneously and to application of a post effect (Fig. 6).

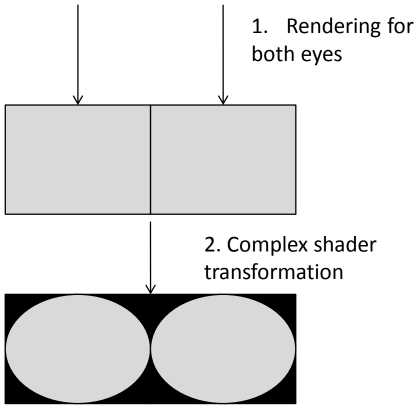

Fig. 6. Oculus Rift support in vAcademia

In the process of implementing the transformation, the shader suggested by Oculus Rift SDK was found to produce visual distortions on high objects when turning the head (with the viewpoint). The principle of a better realization of the shader has been acquired from the source code of a viewer provided by CtrlAltStudio.

In order to provide a convenient mechanism for avatar control while using the Oculus Rift HMD, xBox controller has been employed. One of the controller's joysticks is used for moving the avatar forward, back, strifing left, and strifing right. The other joystick is used for turning the avatar left and right along the vertical axis.

3.3 3D Virtual Recording and Replay

The technology of 3D virtual recording and reply allows a new type of media content that might have its own set of practical applications. A 3D recording stores rich data that in a format that preserves an original course of activities with the details. This allows accessing 'raw' virtual experience when replaying (or re-visiting) a 3D recording. Assuming that information and knowledge may be stored not only in tangible artifacts (such as written instructions and databases), but also in activities, practices, relations between participants, and in their shared experiences, 3D recording and replay provide a better access to the latter category [62,63].

3D Virtual Recording Concept. The most distinctive feature of vAcademia is 3D recording or virtual recording. It allows capturing everything in a given location in the VW in process, including positions of the objects, appearance and movement of the avatars, contents on the whiteboards, text and voice chat messages [48]. Similar functionalities were realized earlier in few VWs or desktop virtual reality systems. However, 3D recording was never developed into a convenient tool and never adopted for specific use as in vAcademia. In addition, no convenient tools for working with the resultant recordings were developed.

3D recording allows creating a new type of media content and new types of collaborative activities. The new media can be called ‘virtcast’ – a 3D recording of activities in virtual reality or a series of such recordings [64]. This type of content is user-generated, since the process of creating and sharing 3D recordings is fully automated and simple.

A user can attend and work at a recorded virtual session, not just view it as a spectator. In addition, any recorded session can be attended by a group of users. A new group can work within a recorded session and record it again, but with their participation. Thus, there is an opportunity to build up content of recorded sessions, and layer realities on top of each other.

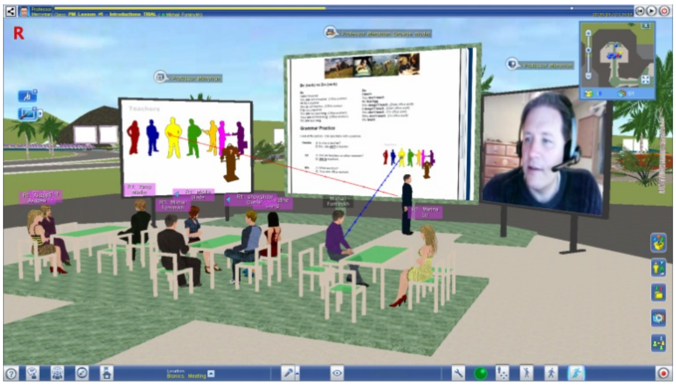

Fig. 7. 3D recording in vAcademia

From the user point of view, 3D recording control is very similar to the regular video player (Fig. 7, top). A 3D recording can be fast-forwarded and rewound, paused, and played again from any point of time. A replayed 3D recording looks exactly like a real virtual session. Of cause, the recorded avatars will always act the way they were recorded. However, it is possible to use all the functionality of the VW inside a recording. Moreover, a replayed 3D recording can be recorded again together with new actions. In such a way, new recordings and new content can be created based on the same original event.

Storing 3D Virtual Recordings and their Data. A 3D virtual recording may contain different resources, such as media content. Resources can appear and disappear multiple times during a virtual session, and have to be managed and stored in the 3D VW and linked to states of the world in the obtained virtual recording.

vAcademia has a storage for such resources called *Resource Collection* [48]. It can be used for uploading, storing, and sharing resources in the 3D VW. The Resource Collection has three sections with different access rights including a public collection (for all users), a private collection, and a virtual session collection (for the participants of the session). Each user can store presentations in PPT format, documents in PDF, pictures in PNG and JPEG, and 3D objects in Collada and 3DS. Some of the resources need to be converted into internal formats (e.g., presentations and 3D objects).

Mathematical Model for Storing 3D Virtual Recordings. A 3D virtual recording is described by the following formula. It represents that components of a 3D recording can change in parallel i.e. any number of object can change any number of their properties at any point of time.

$$P(t) = \{f_i(O_i, t), f_{i+1}(O_{i+1}, t), \ldots, f_n(O_n, t)\} \tag{7}$$

t – time from the beginning of the recording,
O_i – object in a recorded location,
$f_i(O_i, t)$ – discrete function of the object extended state transformation over time.

In order to make the mathematical model represent the data required for 3D replay, formula (7) can be written as follows.

$$P(a) = \{ B_1, B_2, \ldots B_n \} \tag{8}$$

a – identification number of the author of the 3D recording
B_i – data set required for replaying i-th object in the 3D recording P(a)
In its turn, B_i can be described by the following formula.

$$B_i = \{ O_i, A(t), R_i \} \tag{9}$$

O_i – object in a recorded location
A(t) – function of time for a synchronized object
R_i – data set of resources required for replaying object O_i in a 3D recording
If a resource $r \in R_i$, Resource Collection can be described by the following formula.

$$R = \{ R_l, R_a, R_g \} \tag{10}$$

R_g – public collection data set
R_a – private collection data set
R_l – virtual session collection data set

In order to manage resources of 3D recording remote fragments, a counter variable is attached to each resource. When a resource is used in any 3D recording, the associated counter is incremented. When the counter is equal to zero, the associated resource can be removed to the server archive. Thus, the notion of a resource is extended.

$$r_i' = \{r, J_i\} \tag{11}$$

r_i' – extended resource
J_i – counter of links to the resource in 3D recordings
A 3D recording in its turn can be described by the following formula.

$$J_i = \sum_{k=1}^{k<=N_R} \sum_{m=1}^{m<N_S(k)} H(P_K, B_M) \tag{12}$$

N_R – number of all 3D recordings

$N_S(k)$ – size of data set P_k,
The function $H(P_K, B_M)$ is defined by the following formula.

$$H(P_K, B_M) = \begin{cases} 1, \textit{if } \exists \mathrm{R}: r \in R \textit{ and } R \in B_M \\ 0, \text{otherwise} \end{cases} \tag{13}$$

Thus, a resource can be removed from the system if it is removed from all Resource Collections, and the counter of links is equal to zero i.e. if the following set of equations if fulfilled.

$$\begin{cases} J_i = 0 \\ \forall R, r \notin R \end{cases} \tag{14}$$

If only the first condition is fulfilled, the resource is moved to the server archive.

3D Virtual Replay System Design. The system of 3D virtual replay is closely integrated with the synchronization system. The latter implements the identical appearance of the 3D VW for all users at any point in time. It also saves the data necessary for recreating the state of the world during 3D virtual replay. Synchronization of the VW state is achieved by securing equivalence of the objects' extended states. In vAcademia, an extended state of an object defines a set of parameters required for the visualization of the object and, most importantly, parameters of the actions performed by the object to recreate the behavior of the object over time and visualize it for each user.

For each object functionality type, there is a model of possible extended states and transitions between them. Changes in object extended states happen much more rarely than changes of its visual state. Therefore, the synchronization system operates extended states. A model of some avatar extended states is described as an example (Fig. 8).

Complex substates 'Stand', 'Walk', 'Sit', and 'Run' represent the fact that some of the states are mutually exclusive. However, an avatar can be in other substates 'Speak into microphone' and 'Type into text chat' simultaneously with others. All complex substates contain data describing the state. It allows the system to recreate the same state when the avatar exit and re-enter the VW. Therefore, there is no exit from the state "Avatar in the virtual world" on the diagram (Fig. 8).

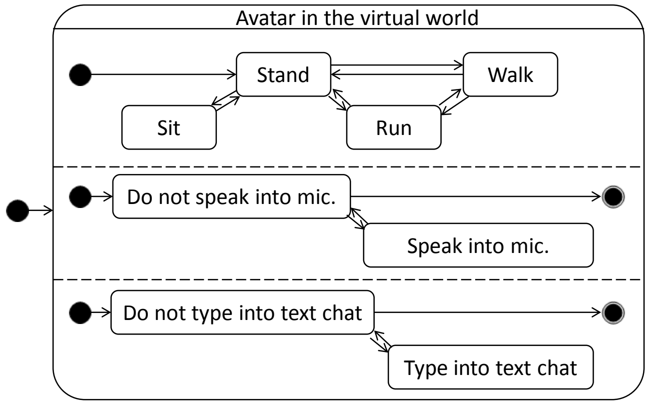

Fig. 8. State diagram of some avatar extended states

3D virtual replay differs from the video playback. In a video recording, frame is the elementary unit. However, in a 3D virtual recording, object action is the elementary unit, and it can be instant or continuous. The continuity of some of the actions makes it difficult to recreate the state of objects at any point in time (e.g., in the middle of a continued action). However, it is required for important tasks, including fast-forwarding and editing 3D virtual recordings [65].

Control Points and Complete State of 3D Virtual Recordings. While a 3D virtual recording is being created, *control points* are created at certain intervals. Control point of a 3D virtual recording is a set of *complete states* of all objects in a recorded location. Complete state of a virtual object is a set of all its synchronized properties. Continuing the parallel to the video, a control point of a 3D virtual recording is similar to a key frame of a video recording.

When fast-forwarding during virtual replay, the closest to the lower side in time control point is found first. Next, the complete states of all objects are sent to the software clients of all users. Finally, all changes in synchronized properties that occur from the time of control point to the sought point are sent.

Keeping the performance level acceptable in such tasks is challenging. First, the system has to apply all changes in extended states of objects that occur from the last control point of the 3D recording. Second, synchronization data that are stored in a 3D virtual recording often contain links to resources (e.g., media content). The network traffic and the computational power required for these processes are high.

Processing Continued Actions. In order to process continued actions, each of them received a variable for time shift from its beginning. This variable is processed by client software. Formula (9) can be written as follows.

$$B_i' = \{B_i, T\} \tag{15}$$

B_i' – extended data set of synchronized object properties
T – time shift

The requirement of time invariability of continued action execution has been set to ensure the correct work of this approach. It means that when replaying a synchronized action, object audio-visual properties will coincide for all users (i.e. software clients) independently of any circumstances, such as processing or network delays.

A different approach is used for synchronizing the continued actions of drawing on a VWB. The drawings are represented as a set of primitives with their properties [53]. All these data for each VWB are stored as a single binary storage. There are two actions defined over the binary storage: clear and write into the end. Thus, when the system needs to re-create the drawing during a virtual replay, the binary storage that existed on the server at the required moment can be applied.

3D Virtual Recording Use Cases. In this section, several 3D virtual recording use cases are elaborated to illustrate its functionality. 3D recording is a new trend in 3D VW technology, and therefore its potential is not yet fully discovered. This feature raises the use of recording to a new level, providing a possibility for active work inside a recording. Collaborative activities with 3D recordings or virtcasts can be conducted in two basic formats: first – visiting a 3D recording, and second – creating a new 3D recording while conducting activities being inside one.

Visiting a 3D virtual recording is similar to working with a video recording or a recorded webinar. It is possible to watch a 3D recording at any convenient time focusing on some parts and skipping the other ones. However, 3D recording differs significantly from recording methods. First, 3D recordings can be visited by a group of users who can actively work inside the recording. Second, a 3D recording is a copy of a live session, but it is not different from the original. Inside the 3D recording, at any point of time, all the objects are on the same positions as they were in the live session. All the interaction that happens in the 3D recording can be observed in the same way as in the live session. However, the visiting avatars cannot interact with the recorded ones.

The second format – creating a new 3D recording, being inside one is even more different form the other types of recordings. A 3D-recorded session can be visited with the same or a different group of users and recorded over again. As a result, there will appear another 3D recording, containing new discussions, questions, and comments. Working inside a 3D recording, the users can use all the functionality of the VW.

A simple procedure is required to access the second format of working with 3D recordings.

1. Loading existing 3D recording (see R_1 in Fig. 9)
2. Starting recording by pressing the corresponding button (R_2 is being recorded now)
3. Playing recording R_1
4. Conducting activities in the environment
5. Fast forwarding, pausing, and stopping R_1 when necessary
6. Stopping recording R_2
7. Saving R_2 in the database

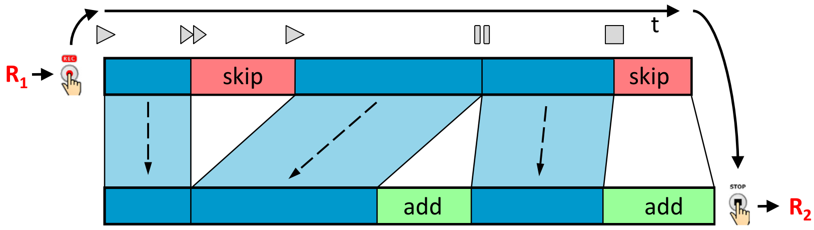

Fig. 9. Creating a new 3D recording, being inside one

It is possible to pause (see the “add” pieces in Fig. 9) and fast-forward/stop (see the “skip” pieces in Fig. 9) the original 3D recording, extending some of the parts and cutting the others. In such a way, the skipped parts of the original recording will not appear in the new one. At the same time, some new actions, recorded while the original recording was paused, will appear (Fig. 9).

Recording new activities being inside 3D recordings allows creating additional virtual content. Activities in the new layers of 3D recordings can focus on specific parts extending and improving the original one or providing multiple variations of an activity. In such a way, vAcademia allows editing content of asynchronous recordings of synchronous activities. 3D recording creates new possibilities for combining synchronous and asynchronous collaboration in a single platform. This feature is especially useful in the cases when a certain activity has to be repeated frequently.

4 Results and Discussion

This section provides the results of the evaluation of the fusion of the main components described above: natural interfaces, large amounts of graphics, and virtual recording. The advances and challenges for further development and practical use are discussed.

4.1 Fusion of Motion Capture, Graphical Content, and Virtual Recording

Applying Kinect for Motion Capture in vAcademia. The evaluation of the first prototype (see Section 3.2) revealed three major limitations of Kinect in the given context and three corresponding challenges for the system implementation.

- C1: Kinect does not capture the gestures accurately enough. Therefore, an attempt to build a reliable avatar model that excludes unnatural poses was not successful.
- C2: Kinect does not recognize the turn of the presenter. If the presenter turns away from the device, it mixes up the left and the right.
- C3: An obvious, but serious challenge for applying Kinect is its inability to capture parts of the body that are covered by other body parts or foreign objects.

Challenge C1 has been addressed by introducing several mechanisms. The distance between the Kinect device and the presenter has been limited. In addition, the Kinect device has been fixed at a 0.5 meters distance from the floor. An additional filtration mechanism for sorting out unnatural positions has been introduced, separating hands as distinct body parts.

Challenge C2 has been addressed by implementing the following algorithm. The turn is recognized relatively as a function of the position of the pelvis end points. The resultant value is valid within the range from -110 to 110 degrees against the “facing Kinect device” direction. Colored markers have been introduced for better recognition of the turns (Fig. 10).

The video data acquired from Kinect is analyzed using SPs. The mathematical model presented in Section 3.1 has been extended with a function of the projected figure template:

$$B_R(x,y) = \begin{cases} 0, \ if \ G_R(x,y) = \phi \\ 1, if \ G_R(x,y) \neq \phi \end{cases} \tag{16}$$

In this equation, G_R is a rasterizing transformation defined in (4). The value of B_R shows if the rasterized figure covers the pixel defined by x and y.

Using (16), another function was defined for the number of pixels in the projected figure on the rasterized image:

$$N_m(S_P) = \sum_{x=1}^{W} \sum_{y=1}^{H} B_R(x,y) \tag{17}$$

In this equation, S_P is a projected figure defined in (5).

The image is processed by a filter that returns the color of the pixel if the pixel does not correspond to the color of the marker with specified error. If (17) returns the value higher that defined, the system considers that the markers are recognized and the avatar is within the allowed turn range. If the returned value is lower, the system stops using Kinect data until the markers are recognized again.

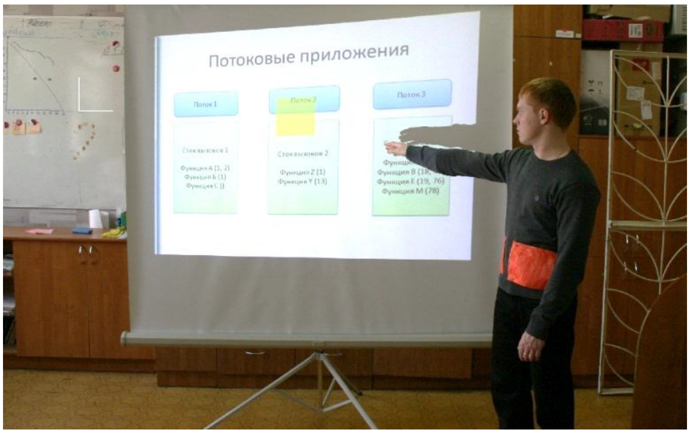

Fig. 10. Presentation capturing process

Challenge C3 may be addressed by applying multiple Kinect devices. It may also contribute to improving accuracy and turn recognition. At the same time, new challenges may appear, such as the increase of the price, complexity in setting up the system, and complexity in merging the data from multiple sources.

Applying Kinect and iPad for Controlling Virtual Whiteboards in vAcademia. Based on the results of the system's first prototype evaluation, the following challenges for controlling virtual whiteboards and their contents have been identified.

- C4: The position and movement of the presenter against the physical whiteboard should match the position and movement of his/her avatar against the virtual one.
- C5: A physical pointer needs to be captured and represented in the 3D VW, where it is often even more important. However, Kinect cannot capture a physical pointer.
- C6: Switching the slides requires interaction with the computer or a remote control, which does not convey any meaning when captured and translated into the 3D VW.

Addressing C4, a Setup mode has been introduced for achieving the precise match between the physical and the virtual whiteboard (horizontally) and installing Kinect at 0.5 meters from the floor (vertically). Instead of trying to capture the physical pointer, the virtual pointer is directed based on the position of the presenter's hand, addressing C5. If the half line that extends from the presenter's hand crosses the physical whiteboard, the avatar in the 3D VW directs a virtual pointer to the same point (Fig. 11). In order to keep the presenter aware of his or her hand being captured, a semi-transparent yellow area on the physical whiteboard is displayed on top of the slides (Fig. 10).

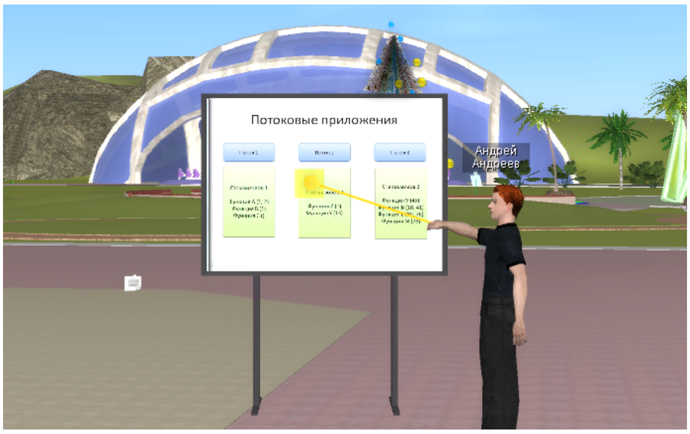

Fig. 11. Presentation streaming process

Addressing C6, the switching slides functionality has been developed to recognize standard Kinect gestures Swipe Left and Swipe Right. In addition, a tablet mobile device (iPad) has been included into the system as an optional interface extending the possibilities of controlling media contents. Using vAcademia Presentation Remote app, the presenter can stream handwriting and drawing to a VWB and control other content on it without going back to the computer. The tablet is connected to the vAcademia client software through the vAcademia communication server using access code [61].

4.2 Fusion of HMD, Multiple Virtual Screens, and Virtual Recording

The fusion of an HMD technology with using multiple VWBs and 3D virtual recording is promising. Even though the technologies are implemented under a single platform, the evaluation of them working together is still underway.

Working with Multiple Virtual Screens wearing a Head-Mounted Display. The fusion of an HMD interface and the multiple virtual screens has been technologically successful. It solved the problem of switching the view between multiple screens. As was indicated in the related work, there are both 2D and 3D technologies that can facilitate remote collaborative work with multiple screens that display various media content. In a 2D system, the display can be either split into sections or contain several screens that can be minimized and restored. In a 3D system, the virtual screens can be installed in any order (e.g., around the avatar of the user). However, the amount of computational and networked resources required to support such a setup is large.

The two technological components of the discussed fusion allow first – displaying a large amount of virtual screens in a 3D space keeping the performance level acceptable and second – switching the view between these screens in the most natural way i.e. by turning the head with an HMD.

The current realization of this fusion of technologies is being tested only internally, and a full-scale user evaluation is needed to confirm the benefits. However, the first serious challenge has been already discovered. The Oculus Rift HMD has a much smaller image resolution in comparison to regular modern desktops. The current realization includes the first Oculus Rift Developer kit that has the resolution of 1280x800 which means only 640x800 per eye. In addition, the Filter transformation (see the Programming model in Section 3.1) further reduces the amount of visual information on the screen (Fig. 6, step 2). This technical limitation does not allow using this fusion of technologies fully. One of the key characteristic of the approach for displaying large amounts of graphical content in vAcademia is high quality of graphics. The approach was meant for high-resolution graphical content that is viewed on a low-resolution display in the system.

Still, the technology of HMD is now rapidly developing, and Oculus VR released Developer kit 2 with the screen resolution 1920×1080 (960×1080 per eye) and a possible increase in the consumer version. These resolution values will move the visual experience closer to the desired level and allow a user evaluation of the system.

3D Virtual Recording and Replay with a Head-Mounted Display. This fusion of technologies constitutes one of the most promising directions for future research work. The current realization of the 3D recording processes users with Oculus Rift HMD in the same way as users that use regular desktop client software. In the resultant 3D recording, all participants will appear as regular avatars regardless of the use interface their users have been using during the recording. However, the viewpoint data of users can be visualized.

In most cases, the viewpoint data in 3D VWs provide little or no valuable information for the users or the system. These data are not collected or visualized in 3D VWs and similar systems. This fact can be partly explained by the low accuracy in matching the viewpoint with the focus of the user's attention. In addition, visualizing such data can obstruct the view. However, using an HMD, users control their viewpoint in a way that is much closer to the one in the physical world. This can be exploited by the system, but visualized in a virtual replay instead of the live activity. This would allow avoiding the obstruction of the users' view, but provide valuable information visualized and easily accessible through the virtual replay.

In the current realization of the system, users can take any position and viewpoint as well as change them while replaying a 3D recording. However, visiting users will always be third-party observers. The fact of wearing an HMD will only change the observer's experience. This raises a question whether a 3D recording can be experience through the eyes of each participant. The fact that changing the viewpoint with an HMD in a 3D VW is similar to the physical world makes this use case viable and promising. It will allow a more detailed analysis of virtual recordings and actions of particular participants.

Cybersickness. One of the challenges associated with the use of HMDs and Oculus Rift in particular include so-called 'cybersickness' [66]. Testers of early prototypes/development kits of Oculus Rift exhibit some problems such as motion sickness/nausea, loss of peripheral vision and latency when moving the head [67], especially when there is a mismatch between body position in the simulation (walking) and in reality (sitting). Prolonged and safe usage requires that these problems are solved in the coming versions. It also requires adjustment of usage modes, for example limiting time intervals with the Oculus, switching between different usage modes during the working session [68].

4.3 Fusion of Large Amount of Graphical Content, 3D Space and Virtual Recording

This section provides the results of testing the tools available on a VWB (see Section 3.1). In particular, the discussion considers the combination of 3D space and large amounts of meaningful 2D graphical content.

Such combination required a new approach. The 3D VW had to display large amounts of 2D graphics without overloading the CPU with computations. In order to achieve this, some of the core image processing algorithms (such as, discrete wavelet transformations and rasterization of attributed vector figures) have been adopted to work on SPs instead of CPU. The results of the evaluation demonstrated significant improvement in performance. The implementation of image processing algorithms on SPs excels the performance using only CPU computational power tenfold and more [53].

Next, the general efficiency of the system has been evaluated, i.e. its performance degradation when using many VWB tools simultaneously, measuring the average and peak values. In the following, the average results of the evaluation are presented.

Using 3D Virtual Recording for Evaluation. The process of evaluation of the system efficiency relied on another component of the technological fusion suggested in the chapter – 3D virtual recording. It was used to ensure the equivalency of the virtual evaluation scene for each test run and to simplify the process. The data were collected by replaying 3D recording on different hardware configurations. For each configuration, the number of simultaneously working tools was regulated by excluding the required number of VWBs from a re-played 3D recording (Fig. 12).

The data were acquired by running the system on 20 different hardware configurations with Intel CPU and NVidia / ATI graphics adapters from the same price range. On each hardware configuration, 10 runs were conducted [53].

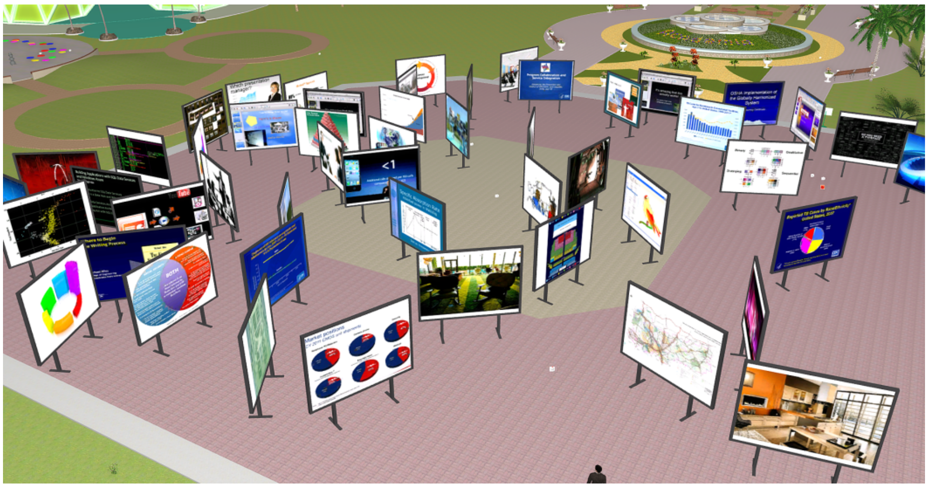

Fig. 12. The process of testing performance degradation as a function of the number of VWBs

General Efficiency of the System with Large Amounts of Graphics in 3D Space. The results are demonstrated by the ratio of the average and peaking performance degradation to the number of simultaneously working VWBs, actively used VWBs, and simultaneous changes of images on VWBs [53]. The *average* values are calculated as the arithmetic mean of all measurements took in all runs. The *peaking* values are calculated as the arithmetic mean of the maximum measurements took in each run. The analysis of data reveals that 50 simultaneously working VWBs reduce the performance of the client software by no more than 7%, while 4% is the average value. The peaking performance degradation reaches 22% when the number of actively used VWBs is 25. The average performance degradation reaches 13%. Five simultaneous changes reduce the performance of the system by 14% in the worst case and by 8% on average.

The results acquired during testing indicate that all the performance requirements for the tools for collaborative work on 2D graphical content were satisfied [53]. However, the quality of the visual content displayed on the VWBs is an essential characteristic for the most of the collaboration use cases. Therefore, the method is designed to ensure the quality of the generated image. Generally, a higher image resolution provides a better visual quality. However, increasing the resolution leads to significant losses in performance.

The proposed approach was designed to mitigate performance loses with the increase of the resolution of the processed images. These design principles were evaluated by comparing the performance of typical algorithms for images of different resolutions implemented on SPs and CPU. The algorithms selected for the evaluation are the packing (Fig. 13) and unpacking (Fig. 14) of images using forward DWT. These algorithms are used for the implementation of the Sharing Changing Blocks method (Section 3.1). The indicative parameter measured in the test is the time of preparation for transformation T_{TR}, defined in (6). The SPs-based algorithms were run on four computers with NVidia GF8800 GT, while the CPU-based algorithms were run on four computers with Core 2 Quad 2.4 G (all four cores were used). For each image size, 1000 runs were conducted on different images.

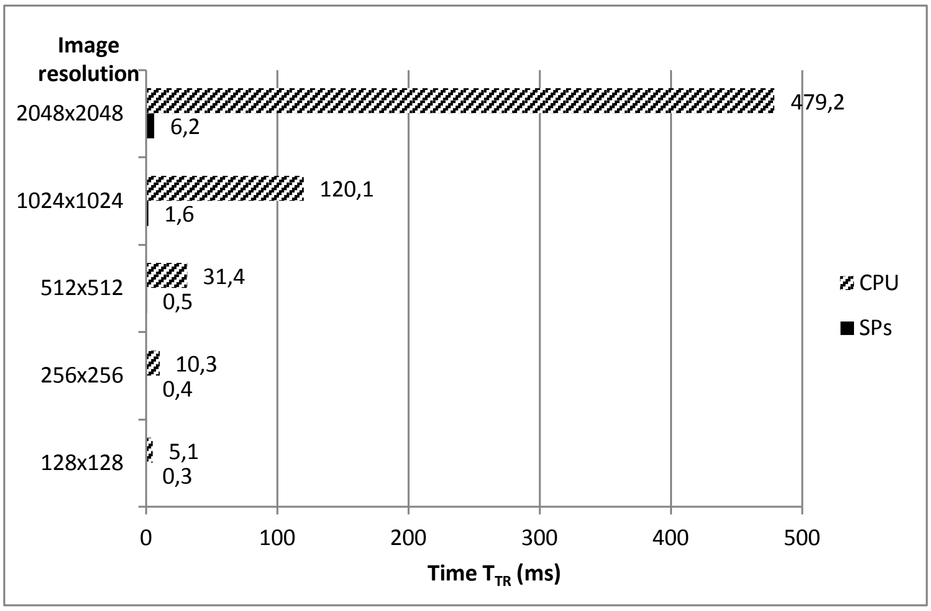

Fig. 13. Image packing preparation time

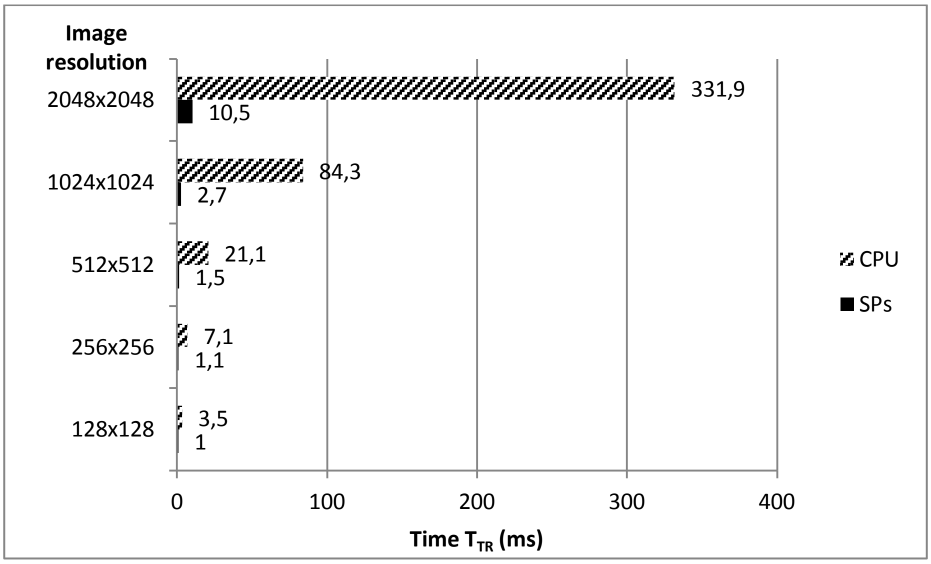

Fig. 14. Image unpacking preparation time

The data demonstrates the advantage of SPs over CPU for given algorithms riches 77 times for packing and 44 times for unpacking the image. However, the most valuable outcome of this test is that the maximum advantage is achieved for images of higher resolutions. The processing time for both algorithms does not grow exponentially or even linearly with the increase of image resolution. This means that the system can operate images of high resolution and quality without losing much of the performance.

Lower performance of the unpacking algorithm on SPs is caused by the higher number of rendering passes used. The non-linear relation between T_{TR} and image resolution is caused by the significant time values of the rendering pass in these algorithms.

The results acquired in both performance and quality tests demonstrate that large amounts of meaningful 2D graphical content can technically be displayed in a 3D VW. This means that the fusion of 3D space and large amount of 2D graphics has been successful. It enables new scenarios of use for the 3D VW and improves the user experience. In addition, 3D virtual recording proved to be a useful fusion component.

5 Use Case Scenarios

In this section, the fusion of technologies described above is discussed in light of how it could support various collaborative activities in the business settings, for example, meetings and collaborative teamwork. In order to structure the discussion, a collaboration model is used.

Collaboration is a complex activity that requires support at all stages to be effective and successful. Three main components of the collaborative process should be considered according to the collaboration standard [69]. The first component is *collaborative workplace* – an instantiated independent entity, consisting of two other components *collaborative group* that performs collaborative activities according to roles by means of a *collaborative environment* consisting of tools [69]. The technological support of these components in vAcademia allows conducting various collaborative activities that are illustrated by the scenarios below.

In the following scenarios, a fictional multinational company A that develops and sells medical equipment is used.

5.1 Scenario 1: Annual Shareholder Meeting

The company is holding its annual shareholder meeting in its head office. The collaborative group assigns the roles of presenters and attendees. The collaborative environment is enhanced with several vAcademia tools. In particular, the fusion of motion capture, graphical content, and virtual recording (Section 4.1) is used. The movements and slides of the presenter are captured with Kinect and recreated in designated virtual offices of company A in vAcademia. The presentation can therefore be followed both by co-located audience and by shareholders, employees, and customers in other parts of the world. The highlights from the financial results are presented on multiple VWBs in the virtual office. The virtual recording of the meeting is stored in the database on the company's own server for later reference. This can be seen as an asynchronous enhancement of the collaborative workplace.

5.2 Scenario 2: Working with Customers

A potential customer considers ordering medical equipment from company A. The collaborative group consists of a customer and a company representative. The collaborative environment used is the fusion of HMD, multiple virtual screens, and virtual recording (Section 4.2). The customer logs into vAcademia using Oculus Rift. He or she is met online by a sales representative from one of the company's offices, not necessarily in the same geographic region. The representative leads the customer to one of the virtual demo halls exhibiting equipment used during neurosurgical operations. The collaborative environment is a virtual exhibition hall that has a section designed as a surgery room. The customer is using Oculus Rift to examine the equipment in full 3D mode. The sales representative might invite the customer to a virtual recording showing a simulation of an operation, where the use of the equipment is demonstrated. The customer might want to update the recording using other tools of the collaborative environment, for example comments and additional specifications (recorded with voice, sticky notes, and written comments on VWBs). The new recording will then be used further by the sales and development teams to deliver the final product to the customer in accordance with his specifications. The customer can also get a copy of 3D recording that contains the entire experience of the meeting with company representative including the comments made during the demonstration. Such a recording can be seen as a different collaborative workplace when it is used later by different collaborative groups with additional roles both in the customer company and in company A (e.g., Scenario 3).

5.3 Scenario 3: Working in Distributed Teams

When receiving an order from the customer in Scenario 2, one of the international teams is assigned to the task. The team holds a meeting inside the virtual recording with the customer's comments and discusses how the customer's specifications might be met from a technical perspective. In this scenario, the collaborative group consists of various experts from company A. At the same time, the collaborative group from Scenario 2 is also a part of the activity. The meeting and brainstorming is recorded again with additional discussions and comments from the experts (including written comments on sticky notes and diagrams on VWBs). It appears that meeting the customer's specifications requires involvement of the second group of specialists in a different office of the company. These specialists review the virtual recording with the comments from the customer and their colleagues. The quality of the content on the VWBs is high enough to see all the details, and the system performs without visible delays. Based on the outcomes, the second group of specialists develops a set of alternatives solutions that are presented on multiple VWBs that are accessible to the other team and the customer.

6 Conclusions and Future Work

This chapter outlines the possibilities for fusion of technologies on the base of 3D VWs. Prospective benefits of such fusion and technological challenges are discussed, and possible scenarios of use in a business context are suggested. The realization of these scenarios requires, however, a seamless integration of technologies and addressing challenges identified in Section 4 in the future.

Another direction for future work will be exploring additional possibilities for fusion of technologies, combining 3D VWs with low-cost VR components and interface elements. For example, paraphernalia used with Oculus Rift already include a large range of devices. Apart from keyboard, standard game consoles, and motion sensor controllers such as Kinect and Razer Hydra are used. Possible paraphernalia include mobile phones and tablets, exoskeletons, haptic interfaces, and omnidirectional treadmill devices, such as Virtuix Omni. Developing realistic simulations of different working modes and business processes might require additional/dedicated tools as well as flexible combinations of different usage modes.

Finally, the adoption of new technologies will inevitably affect usage patterns and collaboration modes. A more throughout exploration of different scenarios and work patterns that could be supported by the fusion of technologies in different areas, such as business, public service, and entertainment are necessary for further development of the field.

References

1. de Freitas S.: Serious Virtual Worlds report. Joint Information Systems Committee, Bristol / London, UK, (2008)
2. Messinger P.R., Stroulia E., Lyons K.: A Typology of Virtual Worlds: Historical Overview and Future Directions. Journal of Virtual Worlds Research 1 (1), 1–18 (2008)
3. Sharma G., Qiang Y., Wenjun S., Qi L.: Communication in virtual world: Second life and business opportunities. Inf Syst Front 15 (4), 677–694 (2013)
4. Lukosch H., Van Nuland B., Van Ruijven T., Van Veen L., Verbraeck A.: Building a virtual world for team work improvement. In: 44th International Simulation and Gaming Association Conference (ISAGA), Stockholm, Sweden, June 24–28, pp. 60–68. Springer Verlag, New York (2014)
5. Surface E.A., Dierdorff E.C., Watson A.M.: Special Operations Language Training Software Measurement of Effectiveness Study: Tactical Iraqi Study Final Report. Special Operations Forces Language Office, Tampa, FL, USA, (2007)
6. Ghanbarzadeh R., Ghapanchi A.H., Blumenstein M., Talaei-Khoei A.: A decade of research on the use of three-dimensional virtual worlds in health care: A systematic literature review. Journal of Medical Internet Research 16 (2), 1–20 (2014)
7. Kluge S., Riley E.: Teaching in Virtual Worlds: Opportunities and Challenges. The Journal of Issues in Informing Science and Information Technology 5 (1), 127–135 (2008)
8. Warburton S.: Second Life in higher education: Assessing the potential for and the barriers to deploying virtual worlds in learning and teaching. British Journal of Educational Technology 40 (3), 414–426 (2009)
9. Barsalou L.W.: Grounded cognition. Annual Review of Psychology 59 (0), 617–645 (2008)
10. Gallagher S.: How the body shapes the mind. Clarendon Press, Oxford, UK (2005)
11. Tůma Z., Tůma J., Knoflíček R., Blecha P., Bradáč F.: The Process Simulation Using by Virtual Reality. Procedia Engineering 69 (0), 1015–1020 (2014)
12. Novak-Marcincin J., Janak M.: Design and realization of robot workplaces with virtual and augmented reality application. Advanced Materials Research 853 (0), 613–618 (2014)
13. Bartle R.: Designing Virtual Worlds. New Riders, NJ, USA (2003)
14. Sanchez-Vives M.V., Slater M.: From presence to consciousness through virtual reality. Nat Rev Neurosci 6 (4), 332–339 (2005)

15. Pasco D.: The Potential of Using Virtual Reality Technology in Physical Activity Settings. Quest 65 (4), 429–441 (2013)
16. Huang H.-M., Rauch U., Liaw S.-S.: Investigating learners' attitudes toward virtual reality learning environments: Based on a constructivist approach. Computers & Education 55 (3), 1171–1182 (2010)
17. Roussou M., Oliver M., Slater M.: The virtual playground: an educational virtual reality environment for evaluating interactivity and conceptual learning. Virtual Reality 10 (3-4), 227–240 (2006)
18. Napieralski P.E., Altenhoff B.M., Bertrand J.W., Long L.O., Babu S.V., Pagano C.C., Davis T.A.: An evaluation of immersive viewing on spatial knowledge acquisition in spherical panoramic environments. Virtual Reality, 1–13 (2014)
19. Djorgovski S.G., Hut P., McMillan S., Vesperini E., Knop R., Farr W., Graham M.J.: Exploring the Use of Virtual Worlds as a Scientific Research Platform: The Meta-Institute for Computational Astrophysics (MICA). In: Lehmann-Grube F, Sablatnig J (eds.) Facets of Virtual Environments, vol 33. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 29–43. Springer Berlin Heidelberg (2010)
20. Molka-Danielsen J., Savin-Baden M., Steed A., Fominykh M., Oyekoya O., Hokstad L.M., Prasolova-Førland E.: Teaching Threshold Concepts in Virtual Reality: Exploring the Conceptual Requirements for Systems Design. In: Fallmyr T (ed.) Norsk konferanse for organisasjoners bruk av informasjonsteknologi (NOKOBIT), Stavanger, Norway, November 18–20, pp. 93–106. Akademika forlag, Trondheim, Norway (2013)
21. Slater M.: Place Illusion and Plausibility Can Lead to Realistic Behaviour in Immersive Virtual Environments. Philos Trans R Soc Lond 364 (1535), 3549–3557 (2009)
22. Demiralp C., Jackson C.D., Karelitz D.B., Song Z., Laidlaw D.H.: CAVE and fishtank virtual-reality displays: a qualitative and quantitative comparison. Visualization and Computer Graphics, IEEE Transactions on 12 (3), 323–330 (2006)
23. Prabhat F.A., Katzourin M., Wharton K., Slater M.: A Comparative Study of Desktop, Fishtank, and Cave Systems for the Exploration of Volume Rendered Confocal Data Sets. Visualization and Computer Graphics, IEEE Transactions on 14 (3), 551–563 (2008)
24. Ragan E.D., Kopper R., Schuchardt P., Bowman D.A.: Studying the Effects of Stereo, Head Tracking, and Field of Regard on a Small-Scale Spatial Judgment Task. IEEE Transactions on Visualization and Computer Graphics 19 (5), 886–896 (2013)
25. Shotton J., Fitzgibbon A., Cook M., Sharp T., Finocchio M., Moore R., Kipman A., Blake A.: Real-Time Human Pose Recognition in Parts from Single Depth Images. Machine Learning for Computer Vision. Springer, Berlin Heidelberg (2013)
26. Cheong S.N., Yap W.J., Logeswaran R., Chai I.: Design and Development of Kinect-Based Technology-Enhanced Teaching Classroom. In: Park JJ, Jeong Y-S, Park SO, Chen H-C (eds.) Embedded and Multimedia Computing Technology and Service. Lecture Notes in Electrical Engineering, vol 181, pp. 179–186. (2012)
27. Winkler M.B., Hover K.M., Hadjakos A., Muhlhauser M.: Automatic camera control for tracking a presenter during a talk. In: Kellenberger P (ed.) International Symposium on Multimedia (ISM), Irvine, CA, USA, December 10–12, pp. 471–476. IEEE, New York (2012)
28. Espitia-Contreras A., Sanchez-Caiman P., Uribe-Quevedo A.: Development of a Kinect-based anthropometric measurement application. In: Coquillart S, Kiyokawa K, Swan JE, Bowman D (eds.) International conference on Virtual Reality (VR), Minneapolis, MN, March 29–April 2, pp. 71–72. IEEE, (2014)
29. Angelopoulou A., García-Rodríguez J., Psarrou A., Mentzelopoulos M., Reddy B., Orts-Escolano S., Serra J., Lewis A.: Natural User Interfaces in Volume Visualisation Using Microsoft Kinect. In: Petrosino A, Maddalena L, Pala P (eds.) New Trends in Image

Analysis and Processing – ICIAP 2013, vol 8158. Lecture Notes in Computer Science, pp. 11–19. Springer Berlin Heidelberg (2013)
30. Cakmakci O., Rolland J.: Head-worn displays: a review. Display Technology, Journal of 2 (3), 199–216 (2006)
31. Antonov M., Mitchell N., Reisse A., Cooper L., LaValle S., Katsev M.: Oculus Software Development Kit. Oculus VR, Inc., CA, USA (2013)
32. Rubia E., Diaz-Estrella A.: ORION: One More Step in Virtual Reality Interaction. In: Penichet VMR, Peñalver A, Gallud JA (eds.) New Trends in Interaction, Virtual Reality and Modeling. Human–Computer Interaction Series, pp. 45–61. Springer London (2013)
33. Reus V., Mello M., Nedel L., Maciel A.: Correcting User's Head and Body Orientation Using a Comfort Pose Function. In: Guerrero JE (ed.) Virtual and Augmented Reality (SVR), 2013 XV Symposium on, Cuiabá, Mato Grosso, Brazil, May 28–31, pp. 167–174. IEEE, (2013)
34. Mavridis A., Tsiatsos T., Tegos S.: Exploiting Web Conferencing to Support Collaborative Learning. In: Angelidis P, Michalas A (eds.) 15th Panhellenic Conference on Informatics (PCI), Kastoria, Greece, September 30–October 2, pp. 78–82. (2011)
35. Kear K., Chetwynd F., Williams J., Donelan H.: Web conferencing for synchronous online tutorials: Perspectives of tutors using a new medium. Computers & Education 58 (3), 953–963 (2012)
36. Cornelius S., Gordon C.: Facilitating learning with web conferencing recommendations based on learners' experiences. Educ Inf Technol 18 (2), 275–285 (2013)
37. Gronn D., Romeo G., McNamara S., Teo Y.H.: Web conferencing of pre-service teachers' practicum in remote schools. Journal of Technology and Teacher Education 21 (2), 247–271 (2013)
38. Marroquim R., Maximo A.: Introduction to GPU Programming with GLSL. In: Tutorials of the XXII Brazilian Symposium on Computer Graphics and Image Processing, pp. 3-16. IEEE Computer Society, New York (2009)
39. Fatahalian K., Houston M.: A closer look at GPUs. Communications of the ACM 51 (10), 50–57 (2008)
40. Kirk D.B., Hwu W.-m.W.: Programming Massively Parallel Processors: A Hands-on Approach. Morgan Kaufmann, New York, USA (2012)
41. Fatahalian K.: From Shader Code to a Teraflop: How a Shader Core Works. Beyond Programmable Shading Course. ACM SIGGRAPH, New York, NY, USA (2010)
42. Mckerlich R., Riis M., Anderson T., Eastman B.: Student Perceptions of Teaching Presence, Social Presence, and Cognitive Presence in a Virtual World. Journal of Online Learning and Teaching 7 (3), 324–336 (2011)
43. Barwell G., Moore C., Walker R.: Marking machinima: A case study in assessing student use of a Web 2.0 technology. Australasian Journal of Educational Technology 27 (Special issue, 5), 765–780 (2011)
44. Greenhalgh C., Flintham M., Purbrick J., Benford S.: Applications of Temporal Links: Recording and Replaying Virtual Environments. In: International conference on Virtual Reality (VR), Orlando, FL, USA, March 24–28, pp. 101–108. IEEE, New York (2002)
45. Leigh J., Ali M.D., Bailey S., Banerjee A., Banerjee P., Curry K., Curtis J., Dech F., Dodds B., Foster I., Fraser S., Ganeshan K., Glen D., Grossman R., Heil Y., Hicks J., Hudson A.D., Imai T., Khan M.A., Kapoor A., Kenyon R.V., Park K., Parod B., Rajlich P.J., Rasmussen M., Rawlings M., Robertson D., Thongrong S., Stein R.J., Tuecke S., Wallach H., Wong H.Y., Wheless G.: A Review of Tele-Immersive Applications in the CAVE Research Network. In: International conference on Virtual Reality (VR), Houston, TX , USA, March 13–17, pp. 180–187. IEEE, New York (1999)
46. Matsuura K., Ogata H., Yano Y.: Agent-based Asynchronous Virtual Classroom. In: Cumming G, Okamoto T, Gomez L (eds.) 7th International Conference on Computers in Education (ICCE), Japan, pp. 133–140. IOS press, (1999)

47. Imai T., Qiu Z., Behara S., Tachi S., Aoyama T., Johnson A., Leigh J.: Overcoming Time-Zone Differences and Time Management Problems with Tele-Immersion. In: 10th Annual Internet Society Conference (INET), Yokohama, Japan, July 18–21. (2000)
48. Morozov M., Gerasimov A., Fominykh M., Smorkalov A.: Asynchronous Immersive Classes in a 3D Virtual World: Extended Description of vAcademia. In: Gavrilova M, Tan CJK, Kuijper A (eds.) Transactions on Computational Science XVIII, vol 7848. Lecture Notes in Computer Science, pp. 81–100. Springer Berlin Heidelberg (2013)
49. Smorkalov A.: Design and architecture of a run-time system for a virtual reality programming language (in Russian). Programnye sistemy i vichislitelnye metody 1 (1), 7–22 (2014)
50. Smorkalov A., Fominykh M., Morozov M.: Stream Processors Texture Generation Model for 3D Virtual Worlds: Learning Tools in vAcademia. In: Lin Q, Muhlhauser M, Sheu P (eds.) 9th International Symposium on Multimedia (ISM), Anaheim, CA, USA, December 9–11, pp. 17–24. IEEE, New York (2013)
51. Smorkalov A.: Realization of educational tools in virtual 3D environments using stream processors (in Russian). Journal of Educational Technology & Society 14 (3), 409–425 (2011)
52. Harrison O., Waldron J.: Optimising data movement rates for parallel processing applications on graphics processors. In: Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: parallel and distributed computing and networks, Innsbruck, Austria, pp. 251–256. ACTA Press, Waltham, MA (2007)
53. Smorkalov A., Fominykh M., Morozov M.: Collaborative Work and Learning with Large Amount of Graphical Content in a 3D Virtual World Using Texture Generation Model Built on Stream Processors. International Journal of Multimedia Data Engineering and Management (IJMDEM) 5 (2), 18–40 (2014)
54. Smorkalov A., Fominykh M., Morozov M.: Collaborative Work with Large Amount of Graphical Content in a 3D Virtual World: Evaluation of Learning Tools in vAcademia. In: 16th International Conference on Interactive Collaborative Learning (ICL), Kazan, Russia, September 25–27, pp. 303–312. IEEE, New York (2013)
55. Wimmer M., Bittner J.: Hardware Occlusion Queries Made Useful. In: Pharr M (ed.) GPU Gems 2: Programming Techniques for High-Performance Graphics and General-Purpose Computation. pp. 91–108. Addison-Wesley Professional, Boston, Ma, USA (2005)
56. Atkinson C.: The Backchannel: How Audiences are Using Twitter and Social Media and Changing Presentations Forever. New Riders, Berkeley, CA (2010)
57. Gray D., Brown S., Macanufo J.: Gamestorming: A Playbook for Innovators, Rulebreakers, and Changemakers. O'Reilly Media, Sebastopol, CA (2010)
58. Morozov M., Smorkalov A., Fominykh M.: Sticky Notes - a Tool for Supporting Collaborative Activities in a 3D Virtual World. In: 14th International Conference on Advanced Learning Technologies (ICALT), Athens, Greece, July 7–10, pp. 683–687. IEEE, New York (2014)
59. Fominykh M., Prasolova-Førland E., Morozov M., Smorkalov A., Molka-Danielsen J.: Increasing Immersiveness into a 3D Virtual World: Motion-Tracking and Natural Navigation in vAcademia. In: International Conference on Applied Computing, Computer Science, and Computer Engineering (ICACC), Saipan, USA, December 28–29. Elsevier, Amsterdam, Netherlands (2013)
60. Jingtang L., Gang Z., Dunming T., Junjie X.: Research of skeletal-based virtual human animation using Cal3D. In: 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization (ICSEM) Chengdu, China, October 20–21, pp. 269–273. IEEE, New York (2012)
61. Smorkalov A., Morozov M., Fominykh M., Prasolova-Førland E.: Virtualizing Real-Life Lectures with vAcademia, Kinect, and iPad. In: Stephanidis C (ed.) HCI International 2014 -

Posters' Extended Abstracts, vol 435. Communications in Computer and Information Science, pp. 156–161. Springer International Publishing (2014)
62. Fominykh M.: Creating and Retrieving Knowledge in 3D Virtual Worlds. In: Gehmann U, Reiche M (eds.) Real Virtuality: About the Destruction and Multiplication of World. pp. 337–368. Transcript (2014)
63. Fominykh M., Prasolova-Førland E., Hokstad L.M., Morozov M.: Repositories of Community Memory as Visualized Activities in 3D Virtual Worlds. In: Ralph H. Sprague J (ed.) 47th Hawaii International Conference on System Sciences (HICSS), Waikoloa, HI, USA, pp. 678–687. IEEE, New York (2014)
64. Morozov M., Gerasimov A., Fominykh M.: vAcademia – Educational Virtual World with 3D Recording. In: Kuijper A, Sourin A (eds.) 12th International Conference on Cyberworlds (CW), Darmstadt, Germany, September 25–27, pp. 199–206. IEEE, New York (2012)
65. Ryzhenkov M.E.: Editing three-dimensional educational content (In Russian). Programnye sistemy i vichislitelnye metody 1 (1), 95–105 (2013)
66. Strauss S.: Cybersickness: the side effects of virtual reality. Technology Review 98 (5), 14–16 (1995)
67. Kot T., Novák P.: Utilization of the Oculus Rift HMD in Mobile Robot Teleoperation. Applied Mechanics and Materials 555 (0), 199–208 (2014)
68. Prasolova-Førland E., Taşdemir S.A., Fominykh M. (2014) Simulating Extreme Environments with Virtual Reality. Paper presented at the 6th Extreme Conference on communication and Computing (ExtremeCom), Galapagos islands, Equador, August 11–16, in press
69. Information technology — Learning, education and training — Collaborative technology — Collaborative workplace. ISO/IEC JTC1 Standard 19778 (2008)