

# Multiple Head Mounted Displays in Virtual and Augmented Reality Applications

H. Kaufmann, M. Csisinko

Institute of Software Technology and Interactive Systems, Vienna University of Technology

**Abstract**—With the recent introduction of low cost head mounted displays (HMDs), prices of HMD-based virtual reality setups dropped considerably. In various application areas personal head mounted displays can be utilized for groups of users to deliver different context sensitive information to individual users. We present a hardware setup that allows to attach 12 or more HMDs to a single PC. Finally we demonstrate how a collaborative, educational, augmented reality application is used by six students wearing HMDs on a single PC simultaneously with interactive framerates.

**Keywords**—Augmented Reality, Head Mounted Displays, Multi-user applications, Virtual reality.

## 1. INTRODUCTION

In previous years Virtual or Augmented Reality (VR/AR) applications using head mounted displays (HMDs) supported only a low number of users. Very few applications [21] and setups are described in literature using more than two head mounted displays simultaneously. In a number of application areas multi-user setups with head mounted personal displays would be beneficial though. In contrast to stereoscopic VR multi-display setups used for large audiences (which are typically projection systems nowadays), personal displays allow to deliver specific context or user dependent information [16] to the individual.

HMDs support collaborative VR/AR and allow personalized viewing of data at the same time. In education and training applications [10], for instance, this is of specific interest. In training (e.g. medical training) groups of users with personal displays can study a dataset together with a tutor whereas each participant is still able to choose his own visualization mode of the data (or a subset), depending on the own knowledge and preferences. Context sensitive rendering is also useful in teaching scenarios where a teacher can be enabled to see a solution of a problem whereas (some) students in the same VR/AR environment cannot see it [8]. Entertainment is obviously another interesting application area for the usage of head mounted personal displays on a larger scale [21].

## 2. BACKGROUND AND RELATED WORK

Considering the image generation process to provide multiple users in VR/AR with individual graphical output, three different types and approaches are known in literature:

- Private or individual screen(s): Each participating

user is associated with an individual display and frame buffer(s), where the “personal” image is generated. One HMD per user is a common example of this.

- Frame interleaving: Users share the same set of screens, but a time-based multiplexing/slicing algorithm separates the images of individual users. Rendered images are displayed time-sequentially on the output device. Special hardware (e.g. glasses as in [3]) is needed to synchronize correctly for each user. With an increasing number of participants this method suffers from bad brightness caused by long time slices of blank images after de-multiplexing on a per-user basis [3].
- Screen partitioning: Rendered images are written into a single frame-buffer and displayed on the same output device. They have to be separated by additional optics to allow each user to see his specific portion of the screen only. Typical examples are the Virtual Showcase [2] or the IllusionHole [12].

With a growing number of users (more than four) the frame interleaving as well as the screen partitioning approaches both become impracticable. There are multiple reasons why applications utilizing large numbers of HMDs are rare:

High costs are a major factor why the acquisition of multiple HMDs has been infeasible in previous years. With the introduction of low cost HMDs such as the Emagin Z800 3DVisor or the Icuity DV920 non-see-through HMDs became affordable. Optical-see-through HMDs for AR applications are currently still not available at a low price. Only few large scale HMD setups can be found in literature. One example are the Virtual Theaters [21] built for edutainment purposes and medical training during the phase of the VR hype in the mid 1990’s until the end of the decade. The usage of multiple (up to four) HMDs has also been considered in neuroendoscopy [22] to allow the principal surgeon, as well as other members of the surgical team to view the surgical site concurrently.

In the following we focus on the use of stereo-capable HMDs only. Depending on the stereo mode supported by the HMD, one (field-sequential/quad-buffered) or two (dual-head) graphic card outputs are needed per HMD to generate a stereoscopic image on the device. Therefore the number of available VGA/DVI ports is an important factor when building

multi-HMD setups. On professional PC graphics cards with two output ports, two HMDs supporting quad-buffered stereo can be connected – one to each output. Alternatively expensive high-end workstation hardware which provides multiple outputs supports setups with multiple HMDs too.

Obviously *distributed* VR/AR frameworks can be used to build large scale HMD setups as well. Cluster-based approaches for multi-display rendering have been studied in the past [6, 15, 17, 20] (and by many more).

As mentioned in related work [17, 20] an advantage of cluster rendering to single workstation/PC solutions is higher computing power per graphics port. With the advent of multi-core processors being now widely available to the public rendering load can be distributed more efficiently within a single PC though (see section 5).

Maintenance costs and efforts of distributed VR/AR setups with multiple clients are considerably higher than those of single workstation solutions. In addition hardware costs of small cluster-PC solutions are nowadays higher than single PC solutions with multi-core processors assuming comparable performance. There is also no network traffic on physical media when running on a single workstation and in case of a single instance solution there is even no need for networking and data synchronization at all.

Multi-display setups in general have been studied by many research groups in previous years. For instance, Schmalstieg et al. [18] describe a distributed multi-user system combining various display technologies in an augmented reality environment: projectors, HMDs and desktop monitors. Nowadays typical multi-display setups serving large groups of users are based on passive stereo technology. For instance, in VR theaters and similar display environments multiple high resolution projectors are driven by workstations or high end PC hardware (e.g. NVIDIA Quadro Plex). These setups do not allow delivering specific content to a specific viewer in the audience.

We present a hardware setup that supports groups of users with personal head mounted displays (Figure 1). Initially a single PC solution is proposed which is easy to maintain, though the setup can be extended to PC clusters.

In the next section we describe the hardware setup that was used in our lab to attach six HMDs to a single graphics card. Tracking all users was another challenge that will be mentioned briefly. Section 4 focuses on software specific aspects and describes relevant features of the Studierstube VR/AR framework that we used. In section 5 extensions and other feasible hardware scenarios which become possible with the proposed hardware are presented. Finally results are presented in section 6. An educational AR application for geometry education, Construct3D, is used to demonstrate practicability of the proposed setup.



Figure 1: Six users equipped with HMDs working on our lab setup on an education AR application.

### 3. HARDWARE SETUP

High-end graphics cards such as those in the NVIDIA Quadro series, are equipped with two 400Mhz digital-analog signal converters (RAMDAC chips) which allows them to output very high resolutions (such as 3840x2400) on both DVI/VGA outputs. For example, choosing horizontal span mode on an NVIDIA Quadro card permits selecting a resolution of 4800x600 (2400x600 on each output).

#### 3.1 Matrox TripleHead2Go

A recently introduced hardware device, the Matrox TripleHead2Go plays a key role in our setup. It is somehow the successor of the Matrox DualHead2Go device, which is able to drive a pair of displays, arranged next to each other (in screen space) without the need for a second graphics card or VGA output. Consequently, the TripleHead2Go allows splitting one VGA input signal of large resolution (up to 3840x1024) into three individual VGA signals representing three screens, each with one third of the (horizontal) resolution of the original image. The technical functionality of the device is described in the specifications by Matrox [13] “Inside the TripleHead2Go, the monitor signal from the computer is first converted to digital data using various techniques to ensure the best possible conversion. These techniques include gain compensation to normalize the signal and phase adjustment to properly interpret the analog input signal. After the input is converted to digital data, TripleHead2Go divides the display information into 3 display outputs. The first third of the image is prepared to be sent to a left monitor, the middle third of the image is prepared to be sent to a middle monitor and the rightmost third of the image is prepared to be sent to a right monitor. Three separate CRTCs inside the TripleHead2Go are used to generate the three timings and then each one of these three separate images are converted into a separate analog output using Matrox's signature high quality analog output technology.”

Therefore it is possible to split a resolution of 2400x600 into three 800x600 signals. For example three monitors or three HMDs with a resolution of up to 1280x1024 (at 60Hz) can be

attached to a TripleHead2Go box. By using two Matrox TripleHead2Go boxes, it is possible to attach 6 HMDs to a single graphics card with two VGA/DVI output ports. Figure 2 shows a TripleHead2Go with three HMDs attached to it. By doubling the number of PCIe graphics cards in a PC (using two) we can drive up to 12 HMDs on a single workstation. For enthusiasts there is still the possibility to go even further by getting a mainboard with more fast PCIe slots (e.g. from Gainward featuring 4 PCIe slots) or to plug in additional PCI graphics cards in order to support more HMDs. For instance, for parallel computation on the GPU a PC has been equipped with 6 PCI graphic cards and 1 AGP card recently [5].



Figure 2: One input and 3 output connectors. Three HMDs (two Emagin Z800 and one Sony Glasstron) are attached to the Matrox TripleHead2Go. In our 6-user setup three more Sony Glasstron HMDs were attached to another TripleHead2Go box.

In our setup we use two Emagin Z800 HMDs and four optical-see-through Sony Glasstron HMDs, all supporting a resolution of 800x600 at 60Hz (32bpp) and stereoscopic viewing (quad-buffered stereo). Two Matrox TripleHead2Go devices receive the input signal (2400x600 each) from an NVIDIA Quadro 4400. This graphic card supports quad-buffered stereo even at a high resolution of 2400x600 on both DVI outputs. In order to enhance immersion in the virtual world, stereoscopic viewing is of major importance. It was our goal from the beginning to enable stereo viewing for all HMDs in our setup. The Sony Glasstron HMD natively displays stereo when receiving the 800x600 input signal after the larger 2400x600 signal has been processed/split by the Matrox TripleHead2Go.

The stereo mode of the Emagin HMDs was originally tailored towards NVIDIA graphic cards featuring automatic and correct assignment of stereoscopic images to the left and right eye. Unfortunately, after the signal has been processed by the TripleHead2Go device, some information about this image-eye assignment is lost and the Emagin HMDs do not automatically switch to stereo mode anymore. Although we could not get clear insight into this issue, it seems that an undocumented signal property is responsible for this

behaviour. Obviously, the TripleHead2Go is not able to deal with this VGA signal extension, which is probably of rather proprietary kind.

Help was provided by Emagin a short time ago, when they offered the possibility to circumvent the constraints regarding switching to stereoscopic mode. After releasing an updated version of the Z800 firmware and extending the programming interface by a method to manually switch into stereo mode it is easily possible to force the Emagin Z800 HMDs into stereoscopic mode. After carrying out the proper command, the Z800 displays perfect stereo too. We developed a simple, tiny software tool to enable and disable stereo mode manually. It utilizes the Emagin SDK and communicates with the Z800 control box via USB.

The large number of HMDs - given the fact that HMDs have to be connected by cables - require proper mounting of the wiring. For example one possibility is to mount control boxes plus cables on the ceiling to omit users stumbling over cables lying on the floor and to minimize cable crossing occurrences. Considering typical group applications, where users usually do not move around to such an extent, the problem of cables is not crucial. In our applications no observable issues occurred.

### 3.2 Tracking

In order to provide groups of users with an immersive experience exactly tailored to their point of view, all users (respectively their HMDs) must be tracked in space. Tracking a large number of users in such an environment is a problem on its own. It is obvious that any wireless form of tracking is preferred over a wired alternative because of the number of cables, limited freedom of movement leading to a rather tethered experience and other practical problems that would arise due to the cables. Optical tracking is usually not ideal in environments with many users because of occlusions. The big advantage in our setup is that we are mainly interested in tracking the displays of all users which are attached to their heads. Markers/targets that are used for optical tracking are easily visible by cameras mounted to the ceiling and hardly occluded by any other objects. We used our own low-cost optical tracking system [14] which is very similar to commercial system such as the ones from ART or VICON but available for 1/10<sup>th</sup> of the costs. In terms of tracking our system is cable-free and therefore the total amount of cables in the setup is equal to the number of HMDs.

In addition to tracking displays, it is possible to track supplementary interaction devices of 1 or 2 users in a centered interaction area by adding extra cameras directly above that hotspot. We provide user interfaces (a pen and tablet) for 1-2 users, who can collaborate simultaneously while others watch. In order to break a strict role assignment, actively collaborating users and passive watchers can exchange user interfaces on the fly. That way passive observers become active participants and vice-versa. For educational purposes (teacher-students), training or demonstrations, where it is not feasible to equip each user with the same full set of interface devices, this is an ideal collaboration scenario.

A side effect of AR/VR environments, which is usually seen as deficiency, turns out to be advantageous in multi-user environments, especially if the number of co-located users increases: Without further processing virtual scenes are always displayed in (see-through) HMDs “on top” of the real environment. As long as users are not rendered or represented by avatars in the virtual world there will be no occlusions of the virtual content. In other words: Users can never occlude a virtual scene. No matter how many users are located in a room, the whole VR scene, all virtual objects, can be seen by each user regardless where the colleagues are located. The focus of the user assembly can not get lost by line-of-sight issues. This also proves to be important in educational scenarios, during marketing presentations or demonstrations where people often suffer from not being able to fully see at each instant what is being shown.

#### 4. SOFTWARE IMPLEMENTATION

From a software perspective the problem remains of how to generate graphical output fast enough for such a large number of displays on a single PC. Thanks to the advances of multi-core processors, server mainboards with 2-4 CPUs, supporting quad-core processors are available these days. Even in simple VR environments, using the latest high-end graphics cards available, there is the absolute need to split the computational load. The classical approaches are either to do multi-threaded rendering or to run multiple instances of a distributed VR environment.

Our Studierstube [19] framework supports both modes. Studierstube is a collaborative multi-user VR/AR software development toolkit. It uses the scene graph library Open Inventor [24] with additional distribution capabilities (Distributed Open Inventor (DIV) [6]) and supports rendering on various VR/AR display devices as well as acquiring data input from 3D tracking devices. For computer graphics generation we utilize Coin, which is based on SGI’s original Open Inventor implementation. The render traversal of the scene graph is single-threaded as in the original implementation; therefore it can not benefit from multi-core processor architectures directly. In section 5 we will illustrate how load balancing can be achieved by running several instances simultaneously with enabled distribution features though.

With DIV features enabled, the whole or some parts of the scene graph can be shared among multiple instances running on different machines connected to a network. Network communication on standard Internet protocols (TCP/IP and multicast UDP) keeps the distributed scene graph on each participating host up-to-date. Studierstube applications are encapsulated in the scene graph (represented as nodes) and are therefore synchronized as well.

We are running multiple instances of the Studierstube application on the same PC and distribute data between single instances. Although it is possible to share the scene graph as a whole, it is often more efficient to distribute a more compact

representation of high level data especially in long distance distribution as it was shown in [9]. Network transparency features also allow distribution on a single host. As a convenient side effect of this distribution strategy, a system composed of multiple rendering instances balances work load on modern multi-core processor architectures.

#### 5. DISTRIBUTED MULTI-USER SCENARIOS

As already mentioned in section 1, there are several options of how to implement multi-user scenarios with multiple HMDs:

- Single instance: A single instance application used in conjunction with Matrox TripleHead2Go devices opens several windows, each of them being the render target of a HMD. With this configuration each graphics card output can drive up to 3 HMDs. Balancing work load between multiple processors has to be done manually in the single application instance.
- Single host with multiple instances: Multiple instances of an application allow implicit load balancing on multi-core CPUs. Each instance is able to render output for one or more HMDs, but distribution and synchronization with other instances has to take place. Again, using Matrox TripleHead2Go devices triples the number of possible HMDs.
- Multiple hosts: In true distributed systems multiple hosts are involved, but synchronization between these hosts has to be physically communicated over network lines. The distribution of our Studierstube application was demonstrated in [9]. In order to reduce the number of hosts, making use of TripleHead2Go devices typically allows to run 6 HMDs per machine instead of just 2. Using TripleHead2Go splitting devices on multiple hosts, this configuration is able to run a huge number of HMDs, while keeping the number of machines low. Groups of HMDs (typically consisting of 6 display devices) share system resources on a particular host.

#### 6. EDUCATIONAL APPLICATION

In order to use VR/AR applications in realistic, educational settings, a large group of students must be able to participate either actively or passively in the activities taught in VR/AR.

##### 6.1 Construct3D

Our work is based on the educational AR application Construct3D [8, 10, 11]. This system deploys AR to provide a natural setting for face-to-face collaboration of teachers and students. The main advantage of using AR is that students actually see three dimensional objects. With traditional methods students have to rely on 2D sketching or calculating and constructing objects using pen and paper or CAD software. By working directly in 3D space, complex spatial problems and

spatial relationships may be comprehended better and faster than with traditional methods.

According to pedagogical theories [23], collaboration is a fundamental social process that supports learners' development of capabilities. In a collaborative AR environment multiple users may access a shared space populated by virtual objects, while remaining grounded in the real world. This approach is particularly powerful for educational purposes when users are co-located and can use natural means of communication (speech, gestures etc.), but can also be mixed successfully with immersive VR [1] or remote collaboration [7]. Supporting natural collaboration in the mathematics domain opens new possibilities to the educational process.

Direct manipulation and dynamic interaction with virtual 3D objects using tangible interaction devices are key features of Construct3D. In our standard lab setup users are wearing an optical-see-through head mounted display; a pen and a panel are used for direct interaction in 3D space (Figure 3). Head, pen and panel are fully tracked in 3D which allows users to walk around objects and to view them from different perspectives.

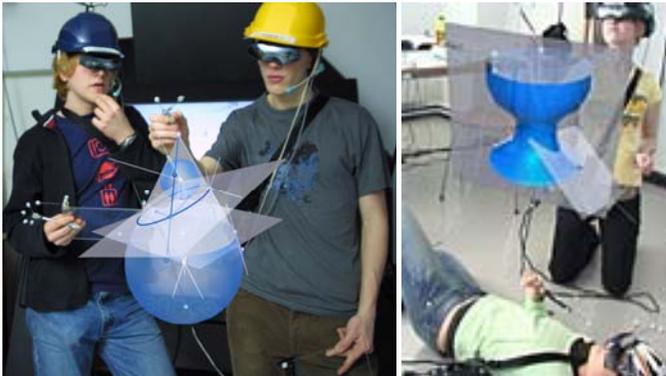


Figure 3: Students working with Construct3D.

For students, teachers and spectators it must be possible to distinguish between the work done by each single user. This is especially important in distributed, remote teaching scenarios, but also in co-located setups. Therefore user information is encoded in object colors. Each user is working within a distinct color scheme: blue, orange, green and red. This color coding is consistently visible on the panel (representing the menu system) and on all geometric objects, indicating the user being responsible for object creation.

Complex constructions involving many objects and work steps can quickly lead to a loss of overview. Therefore, we enable additional structuring by introducing layers. A layer is a simple user-controlled grouping mechanism for geometric objects. Using layers, visual complexity can be managed when teaching with Construct3D. A teacher can switch off irrelevant parts of a construction, or prepare future steps and alternatives as invisible hidden layers to guide students through complicated steps. In conjunction with multi-user operation this feature is particularly powerful. Each user has a personal display for which visibility of layers can be controlled independently [10]. Context sensitive scene graph traversal

[16] is used to generate private views of public shared data.

Construct3D has been evaluated in more than 500 teaching lessons with more than 100 high school students and is under development since 2000.

## 6.2 Teaching Scenarios

A typical application scenario may include several groups of students consisting of six persons. If the total number of students is low, all students can be co-located in a room - preferably in a special lab, permanently equipped with the required hardware and with professional wiring solutions. They collaborate and communicate directly and share a set of interaction devices.

In order to serve a larger number of students there is the possibility to locate smaller groups of users in different labs. Collaboration between these groups has to take place by making use of distribution features to share a common workspace. Within each group members are aware of each other, as the other's physical presence is guaranteed. Supplementary communication and coordination are possible in a very natural way. For communication between the distributed groups, voice chat is suggested. All interaction devices (e.g. pen and panel) have 3D representations in the virtual world and can therefore be shared as part of the distributed VR/AR environment. Their movements can be observed by all participants if desired to see and better understand what other groups are currently working on.

The latter grouping scenario prevents a room from getting too crowded by participants.

## 6.3 Teaching Content

In our lab setup with six users we used two different teaching examples to study the hardware and software issues that might arise in such a multi-user environment. In both cases a tutor presented the examples and asked the other users to participate, requesting feedback and interaction like in a teacher-student scenario. We will briefly describe the teaching content in the following.

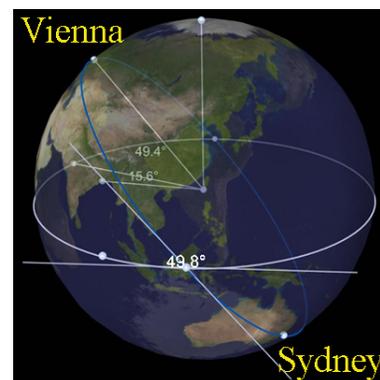


Figure 4: Finding the shortest flight connection between two points on earth.

*Flight Route:* The task is to construct the shortest flight route from Vienna to Sydney, which in ideal case is an orthodrome. The given virtual scene in Construct3D (Figure 4) shows a

model of earth (with texture) to help pupils find the correct places on earth and to immerse them further into the problem. Figure 6 shows this content being used in the six-user setup (image composition).

*Milling Cutter:* Surfaces of revolution are omnipresent in everyday life. This task should be an incitement to learn more about their geometric properties. The perpendicular axis of a surface of revolution is given. Students have to rotate an arbitrary B-Spline curve around this axis. The result is a surface of revolution  $\Phi$  (Figure 5). Construct3D allows to modify the original B-Spline curve dynamically, which results in the modification of the surface of revolution in real time, too. This is a unique feature of Construct3D.

In addition a spherical milling cutter is given (transparent sphere in Figure 5 left) whose radius can be modified. The task is to find a spherical cutter which is suitable to mill the given surface. Students soon notice that if the radius of the spherical cutter is too big, it will intersect with the surface and therefore cut off too much. The example provides an excellent opportunity to discuss curvature of surfaces.

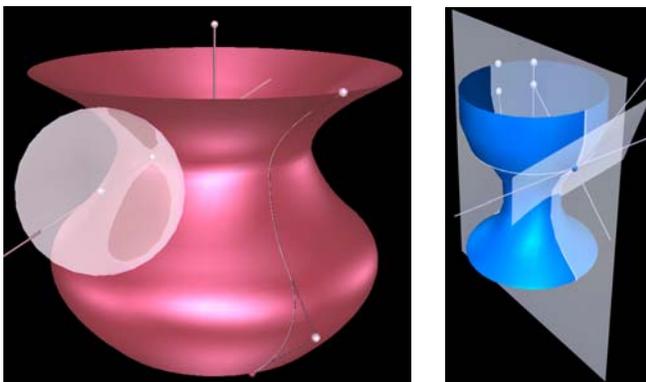


Figure 5: Surfaces of Revolution. Left: Spherical cutter intersects the given surface and is inappropriate for milling.

## 7. RESULTS

Our results were generated on PCs running Microsoft Windows and various NVIDIA graphics cards were used with the latest drivers.

### 7.1 Low cost setup

A single-core AMD Athlon 64 FX-57 processor provided sufficient computing power in our setup. Using a publicly available free software tool (RivaTuner), we reset the PCI Device-ID of an ordinary Geforce 6800 GT graphics card in order to enable high end features of the card, a process which is publicly known and frequently used. Due to this change, the low cost consumer card becomes equivalent to an NVIDIA Quadro FX 4000 card, which allows stereoscopic rendering on both DVI outputs.

We used low cost HMDs - Emagin Z800 3DVisors - in our setup in addition to optical see-through Sony Glasstron HMDs that were already available in our lab. Stereoscopic rendering on both outputs was verified with this configuration and

initially a dual user setup was built. Since the card is equipped with two RAMDACs, we set up a horizontal span resolution of 4800x600. A Matrox TripleHead2Go device was attached to each output, supporting six users with HMDs on a Geforce 6800 graphics card (Figure 6). In dual view display mode two outputs with a resolution of 2400x600 each are also possible. This mode offers more flexibility in running different resolutions on each VGA/DVI port.

Our approach is not limited to a specific graphics card model or brand. Recent models, such as the Geforce 8800 support high resolutions too and are also equipped with two RAMDACs (two dual-link DVI outputs supporting two 2560x1600 resolution displays).

In case a newer Geforce model cannot be modified to support quad-buffered stereo, the NVIDIA stereo driver can be used to achieve stereo rendering on one output on low-cost consumer cards. Since the NVIDIA stereo driver was designed to offer stereo rendering in games, *only* applications running in full screen mode are supported yet. There is no windowed stereo support with the NVIDIA stereo driver. In case a higher screen resolution is split in various smaller parts (as done by the TripleHead2Go) to provide multiple users with stereo graphics, the software which renders the scene must be adapted to generate a full screen stereo window. With the driver constraint regarding stereoscopic mode it is not possible to render stereo images in smaller windows (e.g. three windows with 800x600 as we do in case of the TripleHead2Go) for each partition of the screen. Instead only one full screen window (e.g. 2400x600) can be used as stereo render target and all application output must be drawn accordingly for each user in his visible section of this large window. This can only be done if all sections of the output image are rendered by the same instance of the application; therefore no stereo setup “single host with multiple instances” (as described in section 5) is possible in such a low cost scenario with consumer graphics cards.



Figure 6: Six students working on the flight route task in Construct3D.

In our test we used a single instance setup with six users on a single-core Athlon 64 FX-57 PC. Even in this configuration it is possible to achieve interactive frame rates. During the flight

route task (Figure 4) the frame rate was about 15 fps which allows real time interaction. All HMDs were connected to one graphics card (exactly as described above) and stereo 3D graphics were generated for all six users.

## 8. CONCLUSION

Finally we got one step closer to our aim to provide a low cost setup for immersive VR/AR systems. Our recent work was focused on single or dual user educational settings, but in real classroom use groups of students must be able to follow the teacher. We demonstrated how to use low cost components such as the Emagin Z800 3DVisors, which is affordable for the masses. In addition an affordable optical tracking system was chosen. We used the Matrox TripleHead2Go devices in an uncommon, yet undocumented, but still creative and efficient manner: The ability to drive up to 6 HMDs with a single graphics card plugged into one computer helps saving money and reduces maintenance effort. Thinking of quad-core and 8-core PCs, such a configuration is most suitable for efficiently utilizing current high performance graphics and processing hardware without wasting too many resources, which might remain unused otherwise.

Apart from keeping the expenses low, this system configuration has other implications. As the limit is raised, where a distributed system with multiple hosts becomes practically indispensable, unnecessary system complexity can be effectively prevented.

## 9. FUTURE WORK

Improvements are obviously possible by adapting scene-graph toolkits such as Open Inventor to multi-threaded rendering (this is already included in the commercial version of Mercury Open Inventor). Single instance applications without the need for inter-host synchronization should be optimized to take advantage over modern multi-core CPUs by running several processes or threads. Synchronization features of these single instance processes can be implemented without the need for networking overhead.

For future extensions, the TripleHead2Go device might be potentially interesting in a number of VR/AR applications. For example a low cost CAVE such as the DAVE [4] (Definitely Affordable Virtual Environment) could be built using a single PC. Three walls with passive stereo projection (2 projectors per wall) and a resolution of up to 1280x1024 could be served by the six outputs that can be generated with the help of the TripleHead2Go. Synchronization issues would be easier to handle, maintenance and hardware costs were lower. Even high-end notebooks with two VGA/DVI outputs (which are available too), could be used to drive such a projection environment.

## ACKNOWLEDGMENT

This work was funded in part by the Austrian Science Fund FWF project P19265.

## REFERENCES

- [1] M. Billinghurst, H. Kato, and I. Poupyrev, "The MagicBook - Moving seamlessly between reality and virtuality", *IEEE Computer Graphics and Applications*, vol. 21, pp. 6-8, May-Jun 2001.
- [2] O. Bimber, B. Fröhlich, D. Schmalstieg, and L. M. Encarnacao, "The Virtual Showcase", *IEEE Computer Graphics and Applications*, vol. 21(6), pp. 48 - 55, 2001.
- [3] J. Ehnes, C. Knopfle, and M. Unbescheiden, "The Pen and Paper Paradigm Supporting Multiple Users on the Virtual Table", *Proceedings of IEEE Virtual Reality Conference 2001 (VR'01)*, pp. 157-164, 2001.
- [4] D. Fellner, S. Havemann, and A. Hopp, "DAVE – Eine neue Technologie zur preiswerten und hochqualitativen immersiven 3D-Darstellung", Institute of Computer Graphics, University of Technology Braunschweig, Technical Report TUBS-CG-2003-08, 2003.
- [5] J. Fung, S. Mann, and C. Aimone, "OpenVIDIA: Parallel GPU Computer Vision", *Proceedings of the ACM Multimedia 2005*, pp. 849-852, 2005.
- [6] G. Hesina, "Distributed collaborative augmented reality", in *PhD Thesis: Vienna University of Technology*, 2001, p. 102.
- [7] T. Höllerer, S. Feiner, T. Terauchi, G. Rashid, and D. Hallaway, "Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system", *Computers & Graphics*, vol. 23, pp. 779-785, 1999.
- [8] H. Kaufmann, "Geometry Education with Augmented Reality", in *Ph.D. Thesis: Vienna University of Technology*, 2004, p. 179.
- [9] H. Kaufmann, M. Csisinko, and A. Totter, "Long Distance Distribution of Educational Augmented Reality Applications", *Proceedings of Eurographics 2006 (Educational Papers)*, pp. 23-33, 2006.
- [10] H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality", *Computers & Graphics*, vol. 27, pp. 339-345, Jun 2003.
- [11] H. Kaufmann and D. Schmalstieg, "Designing Immersive Virtual Reality for Geometry Education", in *Proceedings of IEEE Virtual Reality Conference 2006*, Alexandria, Virginia, USA, 2006, pp. 51-58.
- [12] Y. Kitamura, T. Konishi, S. Yamamoto, and F. Kishino, "Interactive stereoscopic display for three or more users", *Proceedings of the 28th Annual Conference on Computer graphics and Interactive Techniques*, pp. 231 - 240, 2001.
- [13] Matrox, "TripleHead2Go technology brief", [http://www.matrox.com/graphics/en/gxm/products/th2go/tech\\_brief.php](http://www.matrox.com/graphics/en/gxm/products/th2go/tech_brief.php), last visited: Dec. 18th, 2006.
- [14] M. Mehling, "Implementation of a Low Cost Marker Based Infrared Optical Tracking System", in *Master Thesis: Fachhochschule Stuttgart & Vienna University of Technology*, 2006, p. 101.
- [15] E. Olson, "Cluster Juggler: PC Cluster Virtual Reality", in *Master Thesis: Iowa State University*, 2002.
- [16] G. Reitmayr and D. Schmalstieg, "Flexible Parametrization of Scene Graphs", *Proceedings of IEEE Virtual Reality Conference 2005 (VR'05)*, pp. 51-58, 2005.
- [17] B. Schaeffer and C. Goudeseune, "Syzygy: Native PC Cluster VR", *Proceedings of IEEE Virtual Reality 2003*, pp. 15-22, 2003.
- [18] D. Schmalstieg, A. Fuhrmann, and G. Hesina, "Bridging Multiple User Interface Dimensions with Augmented Reality", *Proceedings of ISAR 2000*, pp. 159-164, 5-6 October 2000.
- [19] D. Schmalstieg, A. Fuhrmann, G. Hesina, Z. S. Szalavári, L. M. Encarnacao, M. Gervautz, and W. Purgathofer, "The

- Studierstube augmented reality project", *Presence - Teleoperators and Virtual Environments*, vol. 11, pp. 33-54, Feb 2002.
- [20] O. Staadt, J. Walker, C. Nuber, and B. Hamann, "A Survey and Performance Analysis of Software Platforms for Interactive Cluster-Based Multi-Screen Rendering", *Proceedings of the Workshop on Virtual Environments 2003*, pp. 261-270, 2003.
- [21] StrayLight Corp., "'Incredible Voyage' At International Stroke Conference", in <http://www.strayvr.com/ISC2004.htm>, 2004. Last visited Dec. 19th, 2006.
- [22] E. J. van Lindert, J. A. Grotenhuis, and T. Beems, "The use of a head-mounted display for visualization in neuroendoscopy", *Computer Aided Surgery*, vol. 9(6), pp. 251-256, 2004.
- [23] L. S. Vygotsky, *Thought and language. (Original work published 1934)*. Cambridge, MA: MIT Press, 1962.
- [24] J. Wernecke, *The Inventor Mentor: Programming Object-Oriented 3D Graphics with Open Inventor*, 2nd ed.: Addison-Wesley, 1993.



**Hannes Kaufmann** finished his PhD thesis in 2004 on "Geometry Education with Augmented Reality". After his postdoc work in the EU-IST project Lab@Future he got an assistant professor position at the Interactive Media Systems Group, Institute of Software Technology and Interactive Systems at Vienna University of Technology in 2005. He is now leading the VR/AR group at that institute at Vienna University of Technology.

He is the main developer of Construct3D which is being used in a number of national and international research projects and initiated development of the optical tracking system at the Interactive Media Systems Group. He was involved in projects on computational geometry, geometry education, augmented reality and psychological research in AR/VR. His research interests include augmented and virtual reality, education in AR/VR, optical tracking, psychological topics in AR/VR (Cyberpsychology), user interface design for collaborative AR and educational applications, AR/VR and CAD Integration, mobile and ubiquitous computing and computational geometry.



**Mathis Csisinko** is a research assistant and PhD candidate at the Interactive Media Systems group at Vienna University of Technology, Austria. His research interests include 3D user interaction techniques, augmented and virtual reality and especially distribution in VR/AR systems. After finishing his master thesis on "Long Distance Distribution of Virtual and Augmented Reality Applications" where he also worked on Construct3D he received his MS in computer science in 2006 from Vienna University of Technology, Austria.