Designing Immersive Virtual Reality for Geometry Education

Hannes Kaufmann[1]

Institute of Software Technology and Interactive Systems
Vienna University of Technology

Dieter Schmalstieg[2]

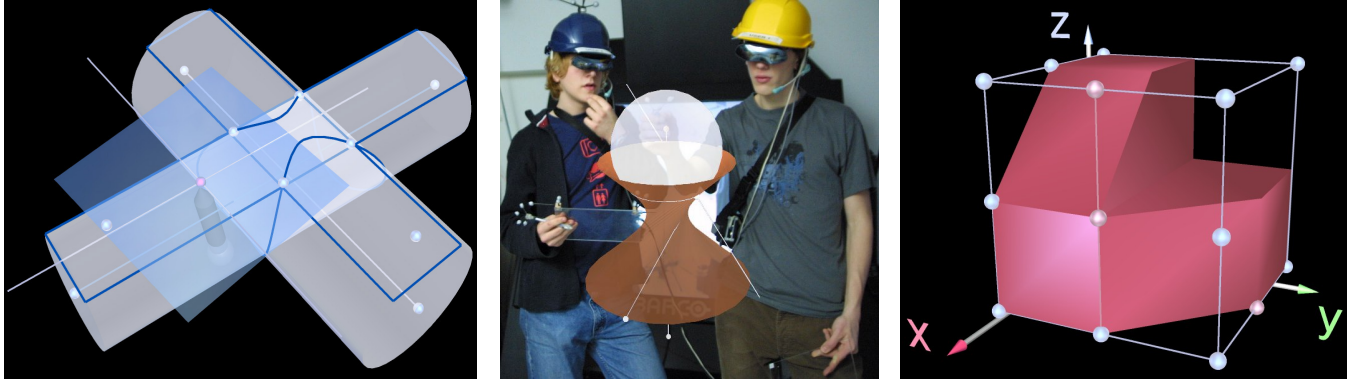Institute for Computer Graphics and Vision
Graz University of Technology

Figure 1: Geometric constructions done with Construct3D. Left: 2nd order intersection curve. Right: Tschupik-cube demonstrating basic Boolean operations on solids. Middle: Collaborative work in Construct3D.

ABSTRACT

Our work introduces immersive collaborative learning to geometry education. More specifically, we present a system that uses collaborative augmented reality as a medium for teaching, and uses 3D dynamic geometry to facilitate mathematics and geometry education. Both these aspects are novel to geometry education. We describe improvements in the user interface and visual design of such an application. We also report on practical experiences with using our system for actual teaching of high school students, and present initial quantitative data on the educational value of such an approach.

CR Categories: H.5.1 Multimedia Information Systems, H.5.2 User Interfaces, K.3.1 Computer Uses in Education

Additional Keywords: Geometry education, computer supported collaborative work, augmented reality, user interface design

1 INTRODUCTION

In this paper, we describe our experiences in developing and using Construct3D, a three-dimensional dynamic geometry construction tool which has been designed to fill the gap of next-generation user interfaces for mathematics and geometry education, and can be used in high school and university education. To provide a natural face-to-face setting for teachers and students, this system uses an immersive setup, more specifically a collaborative augmented reality (AR) setup, based on see-through head-mounted displays (HMD). The main advantage of using AR is that students actually see three dimensional objects which they until now had to calculate and construct with traditional - mostly pen and paper - methods (Figure 1 middle). The face-to-face setting allows for traditional pedagogic communication. Our ultimate pedagogic goal is to verify if working directly in 3D space allows better and faster comprehension of complex spatial problems and relationships than traditional teaching methods.

It is important to note that while geometry education software shares many aspects with conventional CAD software at a first glance, its aims and goals are fundamentally different. Geometry education software is not intended for generating polished results, but puts an emphasis on the construction process itself. While relatively simple geometric primitives and operations will suffice for the intended audience of age 10 to 20, the user interface must be both intuitive and instructive in terms of the provided visualizations and tools. Commercial CAD software offers an overwhelming variety of complex features and often has a steep learning curve. In contrast, geometry educators are interested in simple construction tools that expose the underlying process in a comprehensive way. In accordance to that our aim with Construct3D was not to create a professional 3D modeling package but a simple and intuitive 3D construction tool in an immersive environment for educational purposes.

The system features support for 3D dynamic geometry. A fundamental property of dynamic geometry software is that the dynamic behavior of a construction can be explored by interactively moving individual defining elements such as corner points of a rigid body. It can be seen what parts of a construction change and which remain the same. Experiencing what happens under movement allows better insight into a particular construction and geometry in general.

In this paper we describe the motivation, iterative design process, usability improvements and evaluation results of Construct3D with respect to its target audience of educators and learners. It summarizes lessons learned in the last three years since we reported on the original version of the system [10-12]. We will focus on aspects of the user interface intended to support collaborative geometric construction by multiple users in face-to-face learning situations. Our contribution is the introduction of collaborative augmented reality and 3D dynamic geometry as tools for computer-supported geometry education, insight gained from actual teaching with such a tool and some initial quantitative data on usefulness and educational value of such an approach.

2 RELATED WORK

A large body of work has been done on user interface design for desktop 3D modeling applications. Although 3D input devices with six degrees of freedom (6DOF) have been used to enhance modelers, little work has been done on studying usability for immersive virtual reality modeling. In the following we limit the discussion to HMD based and fishtank VR systems, but not pure desktop systems.

One of the few HMD based modeling systems called 3DM was presented by Butterworth et al. [5]. It includes several grid and snap functions, an extrusion tool for surface creation and some other interesting features. It lacks, however, many of the other aids and constraints that since have been found necessary for accomplishing precise work, as rated by Mine [17], who presented the Chapel Hill Immersive Modeling Program CHIMP. Like its precursor ISAAC [17, 18], it is a test bed for various interaction techniques.

The following 3D modeling and design systems focus on user interaction techniques. They are also used in university education with design or architecture students. DesignSpace by Chapin [6] is one of them. Another example is Bowman's Conceptual Design Space [3] – a real-time, interactive virtual-environment application which attempts to address the issue of 3D design in general and immersive design in particular. The Virtual Reality Aided Modeler (VRAM) by Regenbrecht et al. [21] is a test bed for the application of 3D user interface techniques on a conceptual design tool for architects and industrial designers that runs with VRML97. SmartSketches [23] is a conceptual modeler that combines multimodal user interfaces, combining sketches, gestures and speech in hybrid user interfaces. SeamlessDesign by Kiyokawa [13] is a collaborative augmented reality system for conceptual design. DIVEdit [25] is a collaborative object modeler for virtual environments. It is implemented as an application in the DIVE system [9] for research on distributed virtual environments. DIVEdit combines collaborative solutions and immersive shaping, making an interactive, collaborative object modeler. Preliminary user tests have been conducted in order to identify some questions and problems related to teamwork in virtual worlds.

A very comprehensive overview of nearly all existing 3D modeling systems and virtual reality supported conceptual design tools can be found on the web page of Ernst Kruijff [15]. For a comprehensive overview of interaction techniques in virtual environments we refer to [2] and [4].

In summary immersive modeling systems were primarily used for conceptual design, most likely because of two reasons. (1) Conceptual design has less stringent accuracy requirements than computer aided geometric design (CAGD). Therefore it is easier to support in VR/AR than CAGD. (2) Computer aided conceptual design is not as established in its workflow as CAGD, and therefore offers more room for experimental computer based

tools. To this date, no commercial-grade immersive collaborative modeling system, and no immersive geometry education system exist. We are studying the latter.

The next section introduces dynamic geometry as used in Construct3D, explains its importance for teaching, and its differences from traditional modeling and from traditional geometry education. A brief description of Construct3D and a basic hardware setup will follow. The motivation, the process and the outcome of our design of the virtual environment and the objects in it to support collaboration and teaching will be given in the following section. Finally we provide evaluations based on a standardized usability test which gives evidence of learners' intuitive use of Construct3D and its well designed user interface.

3 STATIC VERSUS DYNAMIC GEOMETRY

One of the fundamental choices in the development of Construct3D is based on the insight that exact construction by coordinates is difficult to accomplish in 3D space by direct manipulation in 6 degrees of freedom. Tracking inaccuracies (in the order of 2-20mm), lack of hand-eye coordination, hand tremor, and difficulties in precisely locating a 3D point presented with a fixed-focus stereoscopic HMD make rapid and precise input difficult. In fact, Bowman [4] and other studies suggest that for direct input in 3D space six degrees of freedom are not expedient most of the time. Therefore it is very reasonable to restrict the user's input to two dimensions for instance. Commonly grids and snapping functions are used to enable exact modeling.

Exact coordinate input is mandatory for all CAGD applications, because the exact geometry of the final result is important. In contrast, educational software can take a different route by employing dynamic geometry.

A fundamental property of dynamic geometry software is that a user can explore dynamic behavior of a construction by moving its defining parts, such as corner points of a rigid body. Derived objects, which are created by referring to other objects, retain this link throughout the construction process. Consequently, moving an object will implicitly update all dependent object, and it becomes very easy to experiment with "what-if" scenarios through direct manipulation. Such a learning environment encourages experimentation and complies better with pedagogic theories such as constructivism than traditional 3D modelers.

At the same time exact construction by coordinates looses its importance. It is still important to have powerful snapping functions to construct objects in correct relations to each other but in contrast to traditional education, coordinates as indications of position in space are of very low importance in a dynamic construction environment.

2D dynamic geometry applications are emerging in schools since the beginning of the 90's and provide an excellent way to explain geometric principles and ideas [14, 16]. Construct3D is the first geometry education software that uses 3D dynamic geometry.

4 COLLABORATIVE AUGMENTED REALITY SETUP

The setup used for Construct3D supports two collaborating users wearing stereoscopic see-through HMDs (Sony Glasstron D100BE) providing a shared virtual space. The users interact with the system using pen and pad props (Figure 2). Both users see the same virtual objects as well as each others' pens and menu systems, therefore a user can help the other one if desired. Head and hands are tracked using an ARTTrack1 tracking system. We also use a stationary camera to provide an additional augmented

viewpoint for bystanders or video documentation. Construct3D uses the Studierstube software platform [24] as a runtime environment and for multi-user synchronization.

The current version of Construct3D offers functions for the construction of 3D points and geometric objects. It provides planar and spatial geometric operations on these objects, measurements, and structuring of elements into layers. It supports generation of and operation on these basic object types: Points (either freely positioned in space or fixed on curves and surfaces), lines, planes, circles, ellipses, cuboids, spheres, cylinders, cones, B-Splines curves, NURBS surfaces up to 8x8 control points and variable degree, and surfaces of revolution. To mention just a few, the following geometric operations are implemented: Boolean operations (union, difference, intersection) on 3D objects, intersections between all types of 2D and 3D objects resulting in intersection points and curves as first class objects, planar slicing of objects, rotational sweeps, surface normals, tangential planes, tangents and many more.

Construct3D promotes and supports exploratory behavior through dynamic geometry without any restrictions, i. e., all geometric entities can be continuously modified by the user, and dependent entities retain their geometric relationships. For example, moving a point lying on a sphere results in the change of the sphere's radius.

Figure 2: Immersive multi-user setup with two HMDs and large screen showing augmented camera view.

5 USER INTERFACE DESIGN

The improvements described in this section were conducted with the intention of improving collaborative learning and teaching. As usability can only be improved in accordance with users' needs and application specific strengths and weaknesses, the guidelines mentioned here cannot be applied directly to other applications without careful adaptation.

Construct3D's menu system is mapped to a hand-held pen and panel interface, the Personal Interaction Panel (PIP) [26]. The panel allows straightforward integration of conventional 2D interface elements like buttons, sliders, dials etc. as well as 3D interaction widgets. Passive haptic feedback from the physical props guides the user when interacting with the panel, while the overlaid graphics allows the props to be used as multi-functional tools. The pen is used for operating on the panel as well as for 3D direct manipulation of the scene. Pen and panel resemble familiar tools from traditional drafting.

Desktop CAD systems offer an abundance of features in typically deeply nested menus that have a very steep learning curve. For Construct3D we clearly need a simpler menu system, which is easy to learn and intuitive to use. We also needed to

accommodate the fact that menu widgets seen in a HMD need a certain size in order to be usable. Organizing the functions proved difficult under these conditions as the number of program functions increased over time. We finally organized the menu – according to experts' knowledge and experience by logic grouping of functionality – into five submenus accessible via tabs (see Figure 3), with certain important functions being visible all the time. This provides relatively quick access to all program functions and is similar enough to traditional desktop CAD menu systems known by many students, while avoiding excessive interface modes. Teachers are also able to disable features which are not suitable for specific learning tasks. In addition widgets are automatically disabled if not applicable to current input elements.
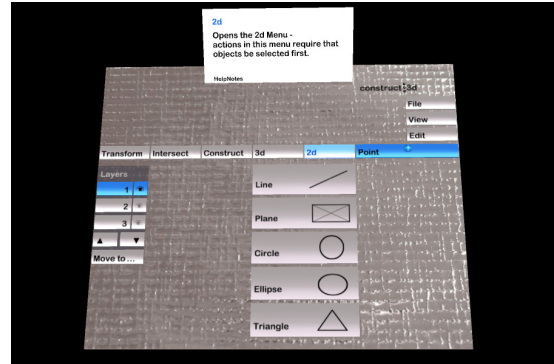

Figure 3: Final design of the menu system (2D submenu; blue color scheme). In a help-box (on top) further details and help on application features are provided.

The main reason for putting a lot of time and effort into interaction and user interface design is that we want students to focus on the actual task and to reduce cognitive overhead needed to use the application. As Rizzo [22] points out "Even if novice users are capable of using a less natural interaction method (or interface) at a basic level, the extra non-automatic cognitive effort required to interact/navigate could serve as a distraction and limit the value of a VR application for accomplishing a specified set of goals." A number of studies report that cognitive overhead in mastering the interface can hinder training and learning of the task [1, 7, 20].

6 VISUAL DESIGN OF GEOMETRIC OBJECTS

The purpose of the visual design of geometric objects constructed by the user is to support the user's understanding of the construction as well as the necessary context to carry out additional tasks. Unlike desktop visualization of the same content, using stereoscopic see-through HMDs requires to deal with limited contrast, resolution and viewing angle. Moreover, the visualization should try to present scenes of high depth complexity in a clear way that improves the user's understanding of the construction. Among the techniques employed in Construct3D to support these goals are the use of transparency to improve the user's understanding of the construction, color coding to allow distinguishing between multiple users' contributions, separation into layers to support semantic structuring of a construction, and automatic previewing of new objects. These techniques make scene handling and graphical rendering quite expensive despite the simple look of the application, but we feel that the usability improvements we observed after introducing these features are worth the additional computational load.

6.1 Transparency

Technical drawings, blue-prints and geometric constructions on paper in general all conform to certain stylistic requirements. Important aspects of constructions are visually enhanced, unimportant parts are de-emphasized. Often it is desirable to show internal, hidden parts too. In modern technical drawings as well as in computer generated images transparency is frequently used to show hidden parts such as parts inside an engine.

The importance of transparency in technical illustrations is documented by Diepstraten et al. [8]. They note "A major advantage of technical illustrations is that they provide a selective view on important details while extraneous details can be omitted. Technical illustrations are better suited to communicate the shape and structure of complex objects and they provide an improved feeling for depth, occlusion, and spatial relationships." Further on they state that "It is quite remarkable that transparency is widely neglected in computer-based illustrations because books on traditional manual illustrations do provide effective techniques and rules for handling transparency in order to communicate the location of occluding and occluded objects."

We are using transparency for geometric primitives to enable users to see inside other objects. Direct manipulation of points inside other objects is only possible if these points can be seen.

In the first version of Construct3D we implemented a slider to give users the option to modify the transparency of objects themselves. This was not satisfactory since after a number of transparency changes many objects had different transparencies which caused confusion. It did not present a consistent look to learners. In order to get a consistent learning environment, a professional graphics designer designed fixed transparency values for all objects and color schemes in general as described in the next section.

Correct transparent real-time rendering is not trivial and computational expensive. Figure 4 (left) shows a computationally less expensive transparency mode that we used initially. Certain geometric primitives produced visually irritating artifacts when rendered transparent. Figure 4 (right) demonstrates the best transparency mode we are currently using combined with our color scheme and 6 lights in the scene that give visually nice spotlights. The midpoint of the sphere is clearly visible in both cases. We are using a fully hardware accelerated transparency mode based on OpenGL 1.5 extensions (fragment programs) by depth peeling. It provides correct rendering of transparencies.
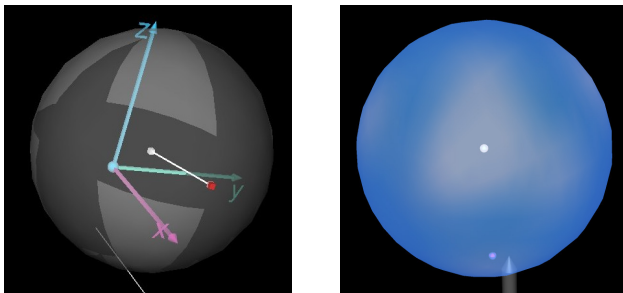

Figure 4: Left: A transparent sphere with rendering artifacts. Right: Rendering of a transparent sphere in a fully correct computational expensive rendering mode.

One disadvantage of using transparencies is that shading or color differences are hard to see on objects. Models that are too transparent appear as blobs and in complex models it is difficult to see all edges. Therefore a useful transparent value must be found that allows seeing through multiple layers of nested objects and still enables the user to see the shape of the model. The goal was to design all transparencies in a way that objects behind more than two other overlapping 3D objects can still be seen. Figure 5 serves as a proof of successful application of our design of transparencies.
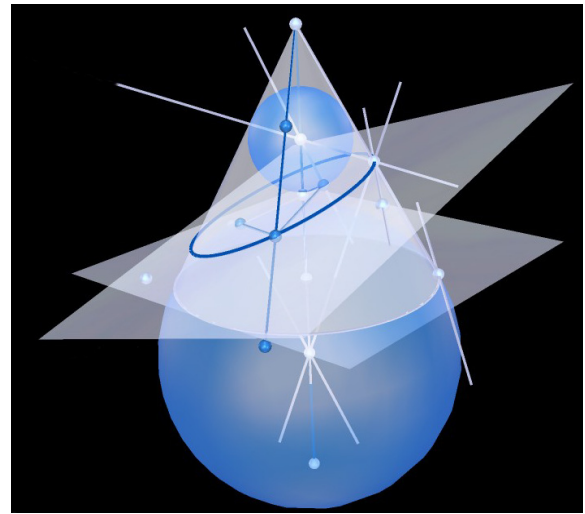

Figure 5: Proof of DANDELIN: A proof that the intersection of plane and cone are either ellipse, hyperbola or parabola. This image shows the elliptic case only. By dynamically changing the intersecting plane all cases can be visualized.

For the above mentioned reasons we do not use transparency for complex objects such as Boolean objects, NURBS surfaces or sweep surfaces. These objects are drawn opaque and users have the option to switch them individually to wireframe mode. It allows to see inside or behind these objects. Points for instance, which are inside other objects must be accessible by the user at all times in order to be able to modify them. Figure 6 shows a rotational sweep surface in normal and wireframe mode.
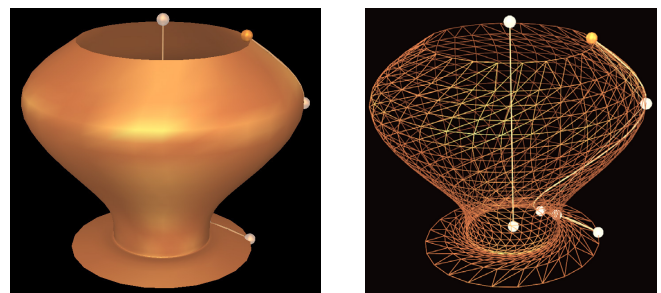

Figure 6: Left: The lower point of the axis and points on the B-Spline curve are hidden behind and inside the surface of revolution. Right: In wireframe mode all points are visible and easily accessible.

6.2 Color Coding

A graphics designer helped to develop a color scheme for Construct3D in order to structure geometric content. Two aspects have been considered as most important by us for educational applications:

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. Therefore user information is encoded in object colors. Each user
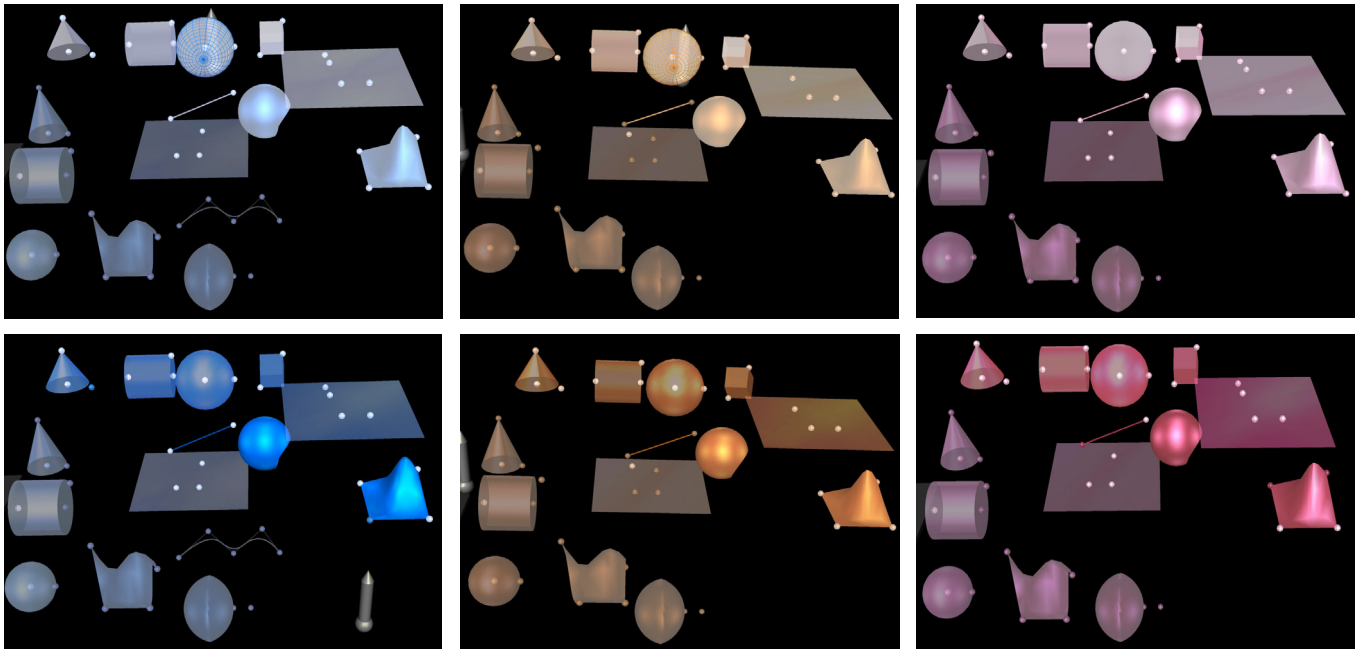
Figure 7: Three Construct3D color sub-spaces for selected / deselected / active layer and inactive layer colors.

(up to four) is working within a distinct color scheme: blue, orange, green and red.

In addition to encoding user information in the color scheme, it was also considered important to have visual information about active and inactive layers (see below) present at all times. This is done by rendering content assigned to inactive layers in a desaturated style.

6.3   Layers

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. Only one layer is active at any time but multiple layers can be visible. New objects are always drawn into the active layer. Inactive layers can show objects of previous stages of the construction. The layer feature is particularly powerful in conjunction with multi-user operation, where every user has a personal display for which visibility of layers can be controlled independently.

In traditional education teachers use colors and different drawing styles (e.g. dashed or dotted lines) to visually enhance complicated constructions and to structure construction steps. Using layers, a similar presentation style can be achieved and 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 Figure 7 we show objects in active and inactive layers. In the lower left corner of each individual image inactive objects are displayed. They are desaturated and darker than "active colors". In the upper right region active objects can be seen. All screenshots in the first row of Figure 7 show a comparison between *deselected* and inactive objects, the second row compares *selected* and inactive layers of the blue, orange and red color schemes. Active and inactive objects are clearly distinguishable as well as selected and deselected objects which was the main goal of this design.

Implementing the color scheme for per-user color and display of layers proved to be more troublesome than initially anticipated. Since each geometric primitive is internally rendered in a slightly different way, we had to assign each primitive a different material so that they all look the same. In total we had to design more than 140 different materials for our objects in order to generate a unique look and feel. 6 lights were added to the scene to produce uniform lightning conditions in the virtual environment independent of the user's position. Texture-based lightmaps were applied to planes and cubes to make them appear like being lit. We designed these colors specifically for the virtual environment when viewing a scene with head mounted displays. They look different on monitors and appear less bright when viewed with see-through HMDs than on paper printouts.

6.4   Improving User Interaction: Highlighting and Preview

Highlighting is used as a method to indicate if a user's pen is nearest to an object. If an object is highlighted the user knows that it can be selected. We performed extensive experiments on how to highlight objects and developed an efficient method. We tried using an additional color for highlighting but it proved to be not distinguishable from other colors in the same color scheme any more, no matter how we chose it. Especially when wearing see-through HMDs colors are not as bright as on monitors and are more difficult to distinguish.

Finally we chose a form of highlighting where we use a wireframe grid of the same model that we superimpose on it. This "highlighting grid" gives the impression of capturing and catching an object with a web which fits to the idea of selecting. Figure 8 (top right) shows how a highlighted point looks like. Various internal tests of this selection method showed that it is very convenient and intuitive to use. During our whole evaluation there were no difficulties with it. Users constantly see which objects are nearest - indicated by highlighting - and they can differ between selected and deselected objects.

Points which are very small objects are also highlighted by superimposing a wireframe grid on them. In addition points can

be dragged. If a users gets very close to a point – within a "dragging area" of 5 centimeters in diameter – the point changes its color to a blazing blue, orange, red or green (depending on the color scheme used) indicating that it can be dragged. It is important to note that all these colors can easily be distinguished from short and large distances.

The preview feature enables a user to see a preview of an object before she actually generates it. While the user moves her pen over a widget on the panel a preview of the object is shown. Figure 8 (left) demonstrates how this works. Immediate visual feedback is given if an operation works with the given input elements. A preview is generated for all possible operations, including intersections and Boolean operations, no matter how complex the resulting model is.
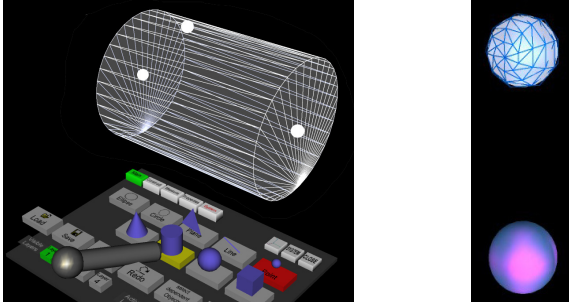

Figure 8: Left: Preview of a cylinder. Right top: A highlighted point. Right bottom: A point that can be dragged.

Not a single one but the combination of all these usability improvements has big impact on the general look and feel when working with Construct3D.

7 EVALUATIONS

Based on feedback from many trials with real high school students, we continuously improved Construct3D over a course of 3 years. Some of the details of these improvements have been mentioned in the previous sections. In early 2004, we conducted a larger study based on interviews and the standardized ISONORM 9241/10 usability questionnaire [19] after systematic exposure of 15 students (9 male, 6 female) to Construct3D. A number of exercises that fit the students' 12th grade curriculum were designed by us and then worked through by students from two high schools with the aid of their teachers. All students attended to geometry classes (descriptive geometry) since the beginning of grade 11. Each of them participated in 5 training sessions lasting 6 hours. Our main objective was to at least informally assess the usability and potential of our system and method for real high school work.

7.1 Exercises for Students

To illustrate the type of geometric examples we wanted to realize and structure with our design, an example is given that students had to construct in one of the training sessions: Given an axis, students must construct a surface of revolution by rotating a B-Spline curve (cubic, 5-6 control points) around the axis. The control points can be dynamically modified at any time resulting in a change of the surface of revolution.

As a next task students have to construct the tangential plane in a point of the surface. Therefore they have to construct a meridian curve through the point which they get by intersecting the surface with a plane through the axis. They also have to rotate the point around the axis to get its circle of latitude on the surface of revolution. The tangential plane is defined by the two tangents (a)

to the circle of latitude and (b) to the meridian curve. Figure 9 shows the result of this construction.
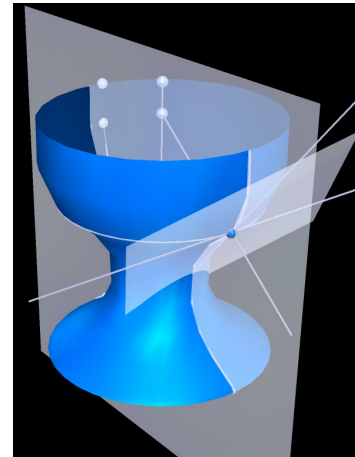

Figure 9: Surface of revolution with the tangential plane in a surface point.

Another example concerns the intersection curve of two cylinders. The goal is to learn about a *geometric method* how to find common points of two cylinders. Therefore the built-in functionality to calculate the intersection curve is disabled. Instead the "helper planes"-method is taught which uses planes to intersect both cylinders to find common points. In Figure 10 the result of the construction is displayed.
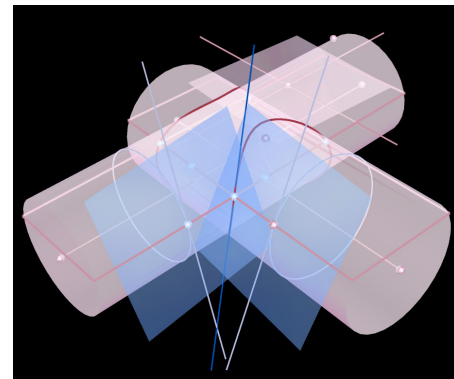

Figure 10: Intersection curve (in red) between two cylinders and a tangent in an intersection point are constructed.

7.2 Results and Discussion

At the end of all training sessions students had to fill out an ISONORM usability questionnaire [19]. Two questions regarding self-descriptiveness of the application had to be removed since they were related to desktop applications only. Afterwards students answered general questions regarding user acceptance, user behaviour, technical requirements and organisational aspects.

Table 1 summarizes the results of the ISONORM usability questionnaire in the 6 categories "suitability for the task", "self-descriptiveness", "controllability", "conformity with user expectations", "error tolerance" and "suitability for learning". A closer look at the data reveals that the categories "suitability for learning" and "suitability for task" receive the highest grading which is very important in this context.

Table 1: Mean values in the 6 categories of the ISONORM usability questionnaire on a scale from -3 to +3.
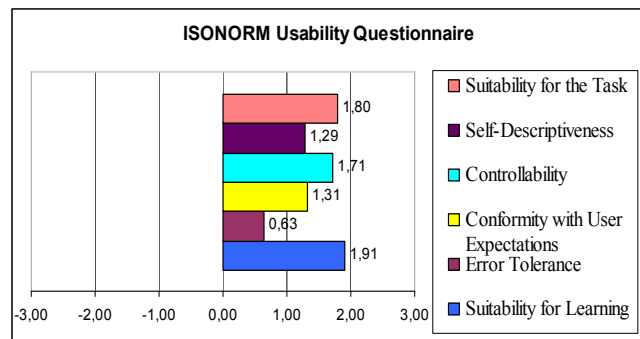
To summarize all ratings above 2,00 (highest grades) the majority of students think: Construct3D is "easy to use", "offers all functions to solve tasks efficiently", "uses comprehensible terms in menus", "offers good opportunities to stop the task and continue at the same point later on", "allows to change easily between menus", "can be used in a consistent way" (highest overall grade!), "requires in case of errors low effort to correct them", "requires little time to learn", "encourages to try new functions", "does not require the user to remember many details" and "is designed in a way that things you learned once are memorized well". All of this confirms our main projections for the development of Construct3D. In our opinion the highest priorities for an educational application that complies with pedagogic theories such as constructivism are that it (1) is easy to use and requires little time to learn, (2) encourages learners to try new functions and (3) can be used consistently and is designed in a way that things you learned once are memorized well. These are exactly the items that students rated very high.

The categories "self-descriptiveness" and "conformity with user expectations" got lower grades than the rest. In 2004 at the time of the evaluation, the final design of the menu system (as shown in Figure 3) was not implemented yet. Therefore the results reflect an earlier state of the menu system. All other design aspects as described in this paper were fully implemented and in use at the evaluation study though. After the study and because of these results, self-descriptiveness of Construct3D was improved by adding a help-box and better labeling of the widgets in order to explain menu items in a concise way. In addition we restructured the menu system to provide a better overview of the functionality (as reported in section 5).

An obvious problem during the evaluation was that the software's response time was unpredictable. The reasons for slow rendering and unpredictable processing time are mainly founded in our complex evaluation setup. During our pre-tests of the setup with 2 or 3 people we did not notice reduced performance. The problems got obvious when all 6 students plus their teacher used Construct3D in our distributed Studierstube setup. Such a large scale test of our distributed system was not done before.

The additional general questions about Construct3D that we asked revealed that students also rated the work style as fun, and subjectively suitable for learning geometry. It was interesting to hear that nearly all students prefer to collaborate with a teacher and a second student. We thought students would prefer to work alone or with a colleague but without their teacher.

Some of the students reported negative side effects after working in the virtual environment. One female student reported headache and eye strain after 20 minutes of work in the virtual environment but did not stop working and wanted to use

Construct3D again (in total she worked for 3 hours with the system). In retrospect we know that our one hour lessons were simply too long for continuous work with an HMD. Since negative side effects are a general problem when working with HMDs and influence the user's subjective experience of a VR/AR environment considerably they are relevant to all VR/AR applications that use HMDs. We identified some possible reasons of cybersickness that may be relevant to our virtual environment such as accommodation problems, low frame rate, lag or bad fitting helmets. We will work with top priority on reducing negative side effects.

Nearly all students reported that they can imagine using the current version of Construct3D in high school or university education. For the complete results of the evaluation we refer the reader to [10].

8   CONCLUSIONS AND FUTURE WORK

We have reported on the current state of Construct3D, which to our knowledge is both the only collaborative augmented reality application specifically aimed at high school education, and the only geometry education software that uses 3D dynamic geometry. This unique combination makes direct comparison to related work rather difficult. However, we believe that the current iteration of user interface and visual design of Construct3D, which we have described in some detail in this paper, as well as the underlying AR technology is sufficiently mature for use in educational practice. The evaluation described in this paper, although limited and informal, supports this claim.

In a related research project we are investigating the hypothesis that students' spatial abilities can be improved by training with Construct3D in absolute terms as well as relatively better than with traditional pencil and paper training. Therefore an extensive psychological evaluation is planned. We will address the following research questions in our study: effects of training on performance in tasks similar to the training; transfer of the training effect to more distant spatial tasks; effects of training on strategy use; dependency of individual training effects on pre-test spatial ability, verbal ability, and reasoning ability and gender differences.

REFERERNCES

[1]   B. Barnett, K. Helbing, G. Hancock, R. Heininger, and B. Perrin, "An Evaluation of the Training Effectiveness of Virtual Environments," presented at Proceedings of the Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), November 27-30, 2000. Orlando, Florida., 2000.

[2]   D. Bowman, E. Kruijff, J. J. LaViola Jr., and I. Poupyrev, *3D User Interfaces: Theory and Practice*: Addison/Wesley, 2004.

[3]   D. A. Bowman, "Conceptual Design Space - Beyond Walk-through to Immersive Design," in *Designing Digital Space*, D. Bertol, Ed. New York: John Wiley & Sons, 1996.

[4]   D. A. Bowman, "Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application," in *Ph.D. thesis, Georgia Institute of Technology*, 1999.

[5]   J. Butterworth, A. Davidson, S. Hench, and T. M. Olano, "3DM: A Three Dimensional Modeler Using a Head-Mounted Display," *Computer Graphics. Proceedings 1992 Symposium on Interactive 3D Graphics*, vol. 25, pp. 135-138, 1992.

[6]   W. L. Chapin, T. A. Lacey, and L. Leifer, "DesignSpace: A Manual Interaction Environment of Computer Aided Design," *Proceedings of the ACM's SIGCHI 1994 Conference: CHI'94 Human Factors In Computing Systems*, 1994.

[7]   C. Dede, M. C. Salzman, and R. B. Loftin, "ScienceSpace: Virtual Realities for Learning Complex and Abstract Scientific Concepts," *Proceedings of IEEE VRAIS '96*, pp. 246-252, 1996.

[8]   J. Diepstraten, D. Weiskopf, and T. Ertl, "Transparency in Interactive Technical Illustrations," *Computer Graphics Forum*, vol. 21, pp. 317-326, 2002.

[9]   E. Frécon and M. Stenius, "DIVE: A scaleable network architecture for distributed virtual environments," *Distributed Systems Engineering Journal (DSEJ)*, vol. 5, pp. 91-100, 1998.

[10]   H. Kaufmann, "Geometry Education with Augmented Reality.," in *Ph.D. Thesis*. Vienna: Vienna University of Technology, 2004, pp. 179.

[11]   H. Kaufmann and D. Schmalstieg, "Mathematics and geometry education with collaborative augmented reality," *Computers & Graphics*, vol. 27, pp. 339-345, 2003.

[12]   H. Kaufmann, D. Schmalstieg, and M. Wagner, "Construct3D: a virtual reality application for mathematics and geometry education," *Education and Information Technologies*, vol. 5, pp. 263-276, 2000.

[13]   K. Kiyokawa, H. Takemura, and N. Yokoya, "SeamlessDesign for 3D object creation," *IEEE Multimedia*, vol. 7, pp. 22-33, 2000.

[14]   U. H. Kortenkamp, "Foundations of Dynamic Geometry," Swiss Federal Institute of Technology, Zürich, Switzerland, PhD Dissertation 1999.

[15]   E. Kruijff, "An overview of virtual reality supported conceptual design tools," http://viswiz.imk.fraunhofer.de/~kruijff/cvde.html, 2003.

[16]   C. Laborde, "Designing tasks for Learning Geometry in a computer based environment," in *Technology in Mathematics Teaching - a bridge between teaching and learning*, L. Burton and B. Jaworski, Eds. London: Chartwell-Bratt, 1995, pp. 35-68.

[17]   M. Mine, "Working in a Virtual World: Interaction Techniques Used in the Chapel Hill Immersive Modeling Program," UNC Chapel Hill, Computer Science Technical Report TR96-029, 1996.

[18]   M. R. Mine, "ISAAC: a meta-CAD system for virtual environments," *Computer Aided Design*, vol. 29, pp. 547-553, 1997.

[19]   J. Prümper, "Der Benutzungsfragebogen ISONORM 9241/10: Ergebnisse Zur Reliabilität und Validität," in *Software-Ergonomie '97*, R. Liskowsky, Ed. Stuttgart, 1997.

[20]   J. Psotka, "Immersive training systems: Virtual reality and education and training," *Instructional Science*, vol. 23, pp. 405-431, 1995.

[21]   H. Regenbrecht, E. Kruijff, D. Donath, H. Seichter, and J. Beetz, "VRAM - a Virtual Reality Aided Modeler," in *Proceedings of eCAADe2000*. Weimar/Germany, 2000.

[22]   A. A. Rizzo, G. J. Kim, S.-C. Yeh, M. Thiebaux, J. Hwang, and J. G. Buckwalter, "Development of a Benchmarking Scenario for Testing 3D User Interface Devices and Interaction Methods," presented at The Proceedings of the 11th International Conference on Human Computer Interaction, Las Vegas, Nevada, USA., 2005.

[23]   P. Santos, H. Graf, T. Fleisch, and A. Stork, "3D Interactive Augmented Reality in Early Stages of Product Design," *HCI International 2003, 10th Conference on Human - Computer Interaction*, pp. 1203-1207, 2003.

[24]   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, 2002.

[25]   M. Stenius, "Collaborative object modelling in virtual environments," KTH, School of Computer Science and Engineering, Royal Institute of Technology, Stockholm, Sweden, Master Thesis 1996.

[26]   Z. S. Szalavári and M. Gervautz, "The Personal Interaction Panel - A Two-Handed Interface for Augmented Reality," *Computer Graphics Forum*, vol. 16, pp. 335-346, 1997.