

INTUITIVE VIRTUAL REALITY - CAD ASSEMBLY SYSTEM

M. Fechter, J. Miehl, T. Stangl and S. Wartzack

Keywords: virtual reality, computer-aided design, gesture interaction

1. Immersive Human Computer Interaction

The human computer interaction (HCI) has continuously been developed since the invention of the computer mouse in 1964 by Douglas Engelbert. Nowadays the increasing appearance of touchscreens in mobile devices confirms the importance of innovative interaction technologies. Swipe or zoom gestures to flip pages or scale maps are highly graspable for users.

Wendrich insists that creativity is affected by the way a user interacts with the computer. Being able to transfer shapes into the computer, that are handmade of malleable material, offers great potential for designers. Thus it is shown, that the user's creativity is encouraged, when doing things by hand [Wendrich 2013].

A similar effect occurs when a product developer is interacting with digital mock-ups in virtual reality (VR). The more the handling of the current model matches the handling of the product in reality, the more creative and efficient the working process will evolve. Virtual reality has the great benefit of illustrating products in head-tracked stereoscopic vision, whereas common CAx software displays the three-dimensional (3D) model just in a two-dimensional projection. Hence VR offers possibilities for a completely new way of intuitive interaction with and a more realistic impression of the virtual product. To benefit from these advantages with minimal influence on the engineers working process, we propose to introduce a "hybrid workplace". This means a combination of an established desktop PC and devices that provide high level user immersion. The product developer is still going to have the same equipment, like mouse and keyboard as input devices, whereas operations that are more intuitive and efficient in virtual reality are executed with the help of the additional gesture-based interaction.

In the field of computer-aided design some 3D-related workflows, like part modelling and assembling components, exhibit disadvantages that result from the two-dimensional vision and the limitations of common HCI with traditional input devices. These processes would gain from contemporary methods like intuitive gestural interaction and immersive visualisation of VR environments.

The purpose of this contribution therefore is the development of a framework for combining the benefits of highly intuitive interaction and stereoscopic visualisation techniques with the advantages of common workflows leading to an integrated hybrid engineering environment.

The usability of this framework is demonstrated by a VR-CAD assembly system. This system enables the product developer to insert constraints by selecting a face or part in space, harnessing the power of gestures and movements, instead of cumbersome selection of face-to-face constraints. The system is not intended for assembly planning or evaluation, but to facilitate the process of imbedding constraints for the product developer.

Therefore, we first discuss and evaluate state-of-the-art interaction techniques and approaches of CAD and VR coupling. Thereafter our framework is described including a concept for the gestural hand interaction with the virtual product as well as a case study of a gear box assembly trial. The present paper ends with a conclusion, future prospects and possible extensions.

2. The Basics of Hybrid Workplaces

The proposal of establishing a hybrid workplace brings along a few notable requirements. These can be assigned to three different groups: the VR environment, the influence on the product developer's workflow and the feasibility with regard to the asset costs of additional devices. Below these requirements are analysed and after that existing approaches are considered with a view to the applicability in a hybrid workplace. At the end of this chapter the resulting design of the hybrid workplace used is presented.

2.1 Requirements of Hybrid Workplaces

For providing a spatial impression of the displayed models and a high level of immersion, stereoscopic vision and head tracking have to be integrated into the system. Common environments for immersive visualisation are cave automatic virtual environments (CAVE), powerwalls and head-mounted displays (HMD). Head tracking can be achieved through optical systems that utilise image data for triangulation of a position or through non-optical systems that use magnetic and inertial effects or mechanical motion [Cruz-Neira et al. 1993].

To achieve an easy to handle system with high efficiency, the nature of the interaction with the virtual product has to be as intuitive as possible. The most widely used VR devices are flysticks that were originally designated for virtual design reviews in the automotive industry and gloves that are predestined for element grabbing operations [Knöpfle and Voß 2000]. But considering the flysticks lack of direct intuitive handling, the application in a hybrid workplace seems not very promising. Additionally the costs for the tracking system are quite high. Assets and drawbacks arising through the usage of glove devices are discussed in the following section.

For the integration of VR and CAD there are three alternative approaches: linking a CAD system and VR system, linking a CAD kernel and a VR system or using voxel models for geometry description. In the present paper, voxel methods are not going to be examined in more detail. An important point in combining VR and CAD components is minimizing the implementation costs to gain the opportunity of reaching the majority of engineering workplaces. Thereby the plurality of common functionalities of CAD systems should be available.

Of course the novel working environment represents an intervention that could potentially disrupt the product developer's habitual working processes. There must not be any restrictions during the workflow that may cause a non-use of in fact helpful and efficient facilities, like impeding keyboard entries through wearing interaction devices.

2.2 Approaches to Integrate CAD and VR

First of all a literature search was carried out to investigate available system architectures, as the idea of integrating VR and CAD already appeared some time ago [Berta 1999]. The approaches presented below were selected, because they demonstrate the use of a wide range of virtual environments and peripheral devices. The different techniques have advantages and drawbacks to be considered with regard to the utilisability in a hybrid workplace.

The first approach examined is the virtual assembly design environment (VADE) [Jayaram et al. 1999]. This system was especially developed for assembly planning and evaluation. Initially a digital mock-up is designed using parametric CAD software. This model is exported to the VADE system, where parts can be assembled with virtual tools, like for example screwdrivers. During the procedure there is a link between the CAD software and VADE. Key design parameters of features like the radius of a hole can be changed by the user through a three-dimensional menu. During this procedure the resulting part is updated in VADE. All provided tools are utilised by virtual hands that follow the actual movements of the user who is wearing glove devices. After introducing the system's working method, the suitability of the solution and their applicability in a hybrid workplace is examined. The integration of CAD and VR is reasonable, because the basic modelling functions can be adopted by the CAD software, so there are no needless implementation costs. VADE provides a high level of immersion through the use of a head-mounted display. Interacting via hand movements corresponds to the handling of real world objects due to finger tracking and haptic response, provided by sensors and actuators attached to the used gloves. Transferring the motion to the virtual hands gives feedback to

the user and increases the level of immersion. Using HMDs brings along the disadvantage of being cut off from the outside world and also infringes the requirement of still being able to handle traditional input devices, like typing on a keyboard. Thereby this approach interferes with the condition of not negatively influencing the developer's workflow.

Another quite different approach is the Navigation Interface for Modeling (NavIMode). This is a system, where complete commercial CAD software is available in a CAVE environment. The intention thereby is the simplification of exploring complex digital mock-ups. Therefore the model is displayed in the CAVE while the developer utilises a position and orientation tracked tablet PC as virtual magnifying glass for command insertion [Weidlich et al. 2007]. Thus the stereoscopic viewing environment is separated from the two-dimensional interaction interface, in this case the common graphical user interface (GUI) of the CAD system, which is shown on the screen of the freely movable tablet PC. The great benefit of the CAVE with respect to the requirements mentioned above is its stereoscopic vision. This provides a high level of immersion combined with the possibility of nearly unrestricted movability around the displayed model in the environment. Concurrently the full scope of functions of the commercial CAD system is available and therefore time and effort for implementing them is omitted. An important drawback for the engineer's daily labour situation that arises through this configuration is the spatial distance between the normal workstation and the CAVE system. Due to space requirements, direct integration of the CAVE into the workplace would also be not feasible. By reasons of spatial distance, space requirements and asset costs, the integration of this system with respect to the hybrid workplace is quite difficult.

Construct3D, presented as the third approach, enables modelling with accurate 3D geometries [Kaufmann et al. 2000]. The user wears a semi-transparent stereoscopic HMD where any kind of virtual content, like digital models or menus, can be faded in overlaying the real environment. Controlling the system is realised through a pen and the personal interaction panel (PIP) that is operated with two hands. Pen and panel are equipped with position and orientation trackers. Menus are projected onto the PIP and elements can be selected by pointing with the pen. In contrast to the previous VR-approaches this one uses Augmented Reality (AR), where the product developer's environment is overlaid with virtual models. Handling the input with pen and PIP is well suited for interacting with close-by virtual models and menus. There is no real CAD and VR integration, but the OpenCascade software development platform is used for a few necessary geometric operations. The implementation cost for creating separate CAD functions is extensive. From the ergonomic point of view the engineer's workflow does not interfere with the integration of these additional devices into the novel workplace. On the other side the asset costs for the semi-transparent stereoscopic HMD and the tracking system are probably not acceptable.

After considering the three approaches and discussing the strengths and weaknesses in visualisation, interaction and ergonomic circumstances, in the following the design of the hybrid workplace is specified.

2.3 Design of the Hybrid Workplace

According to the integration approaches discussed above, the options for stereoscopic visualisation are an AR-HMD, a VR-HMD and a CAVE. AR-HMD's often exhibit pretty small field of views, whereas VR-HMD's cut off the user from the outside world and arouse simulator sickness. CAVE systems induce the highest level of presence, but the integration into a product developer's workplace causes the already mentioned problems and high asset costs. For these reasons currently a large stereoscopic screen seems to be the most promising alternative.

Integrating either the pen and PIP or the tablet PC is not an acceptable approach due to the lack of applicability and impairing the developer's process of work. Gloves offer an efficient and intuitive way of gestural interaction, but in combination with the requirement of not interfering with common actions, e.g. keyboard inputs, they also cause inconveniences. Based on these considerations an approach combining the benefits of intuitive gestural interaction with the requirement of not negatively affecting the normal working processes appears to be an efficient solution. A promising option for this is the application of markerless tracking methods. Therefore the Microsoft Kinect sensor is utilised as an input device for detecting corporal gestures and movements. It was originally

intended for the usage on the Microsoft Xbox 360 game console, but due to its potential and success, the 2012 published Kinect for Windows was especially developed for the application on PCs. Capturing depth images works through applying a technique named structured light. Thereby an infrared (IR) sensor captures the arrangement of points that are generated by sending IR light from a diode through a special grid [Zeng 2012]. With the knowledge about the reference arrangement of the points and some other data, the depth image can be calculated. The sensor offers the potential for efficient markerless HCI in combination with not restricting the product developer's workflow and additionally causes just low expenses. To assure the availability of CAD functions and to keep the implementation costs down at a reasonable level, the appropriate solution is linking an open source CAD system and an open source VR framework.

3. Intuitive Stereoscopic Assembly System

3.1 Linking CAD and VR systems

The structure of the intended system should be oriented towards the major tasks – separated in three software modules. There is one module for CAD modelling, one for visualisation together with interaction handling and one for providing interaction data. The basic architecture, its modules and the process of data transfer is shown in Figure 1.

Due to the easy accessibility of graphical representations and modelling functions, the use of an open source CAD system appears to be the most efficient and viable solution. Utilising FreeCAD provides a lot of advantageous features. The program can handle the STEP format for exchanging models with commercial CAD software. FreeCAD's API is responsive to Python, enabling the access to nearly all available functions, without necessarily starting any GUI. An important feature for visualisation is the method for exporting the model to a tessellated 3D geometry. Unfortunately there are no assembly functions and no assembly structures included in the software, resulting in extra implementation cost. The topology of a FreeCAD document is a straight forward hierarchical graph structure. All elements of the geometry, like faces or edges, can be identified by their name plus superordinate component. That information is essential for the connection between an element in the visualised projection and the corresponding element in the CAD representation [Collette 2012].

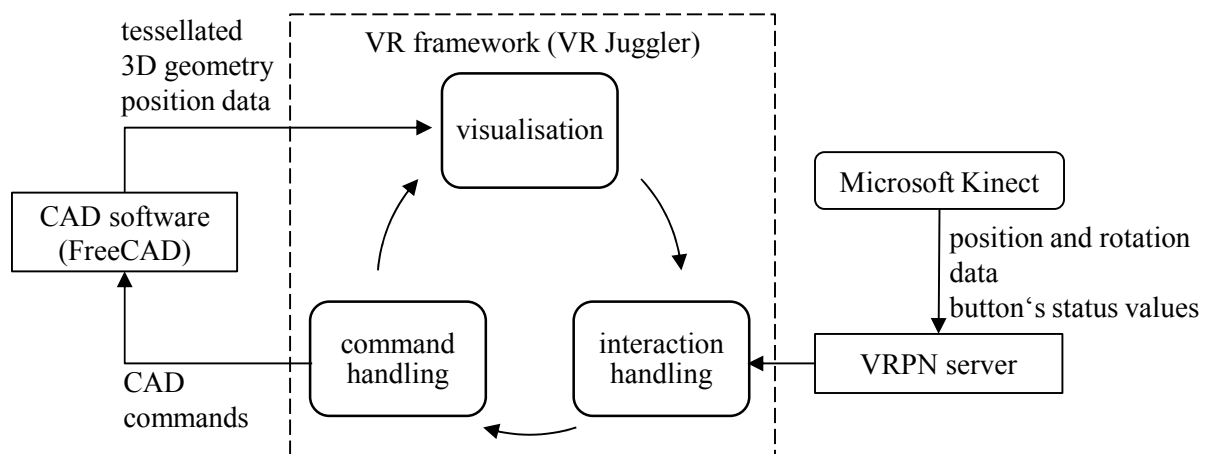


Figure 1. System architecture of the CAD and VR link

It is decided to use VR Juggler together with OpenSceneGraph (OSG) as a powerful development platform. A decisive role for this choice plays the ability to use Python for developing VR Juggler applications. OSG offers the crucial advantage of providing an algorithm to convert the tessellated data from FreeCAD to a data type which can directly be interpreted by the scene graph [Pavlik and Vance 2012].

The calculation of the interaction data by the Kinect sensor runs parallel to the VR application and is provided through a Virtual Reality Peripheral Network (VRPN) server in the local area network.

The communication between CAD software and VR application consists of several components. Common CAD commands like importing assembly groups must be transferred. There is a need for methods that ensure the consistency of the geometrical, kinematical and topological representation of the model in the CAD system and the VR application. For example, one important parameter while assembling a design group is the position and the orientation of the various components. This data has to be equilibrated if any object in either the CAD model or the VR model is moved before any calculation is performed.

3.2 Gestural interaction

After combining CAD and VR, the next step is the development of an intuitive interaction method for handling digital mock-ups. The interaction in virtual environments can be divided into three basic components: selection/manipulation, navigation and system control [Bowman et al. 2001]. Generally selection is the most important component that is partly utilised by the other components. Navigating functions with the meaning of “movement in the virtual space” are not needed for steering the assembly application. The navigation is done via tracking the user’s head movement and simultaneously adapting the perspective in correspondence with changing the parts position and orientation through gestures and hand movements. System control encapsulates managing the VR application together with the CAD system. It cannot be clearly delimited from the other interaction components, because normally it utilises menus in conjunction with selection.

Selecting an object can have different intentions. Some operations, like changing a subassembly’s position and orientation, consist of the selection and an additional manipulation. Other processes, like picking an item from a menu, only consist of selection. Therefore the concept must at least provide one gesture for procedures like object manipulation and one gesture for item picking.

Generally the selection is divided into two constituents: the “pre-selection” that could be imagined as dragging the mouse pointer over the required object and the “selection action”, substituting the mouse click then.

The present concept for object selection was influenced by Rodrigues, who utilises the tracked position of the index finger for operation on a virtual touch screen. The two axes of the projection screen are used for moving the mouse cursor and the value of the axis that is vertical to the screen is used for the clicking event [Rodrigues et al. 2011].

Figure 2 shows an overview of the interaction setting, which is described in the following.

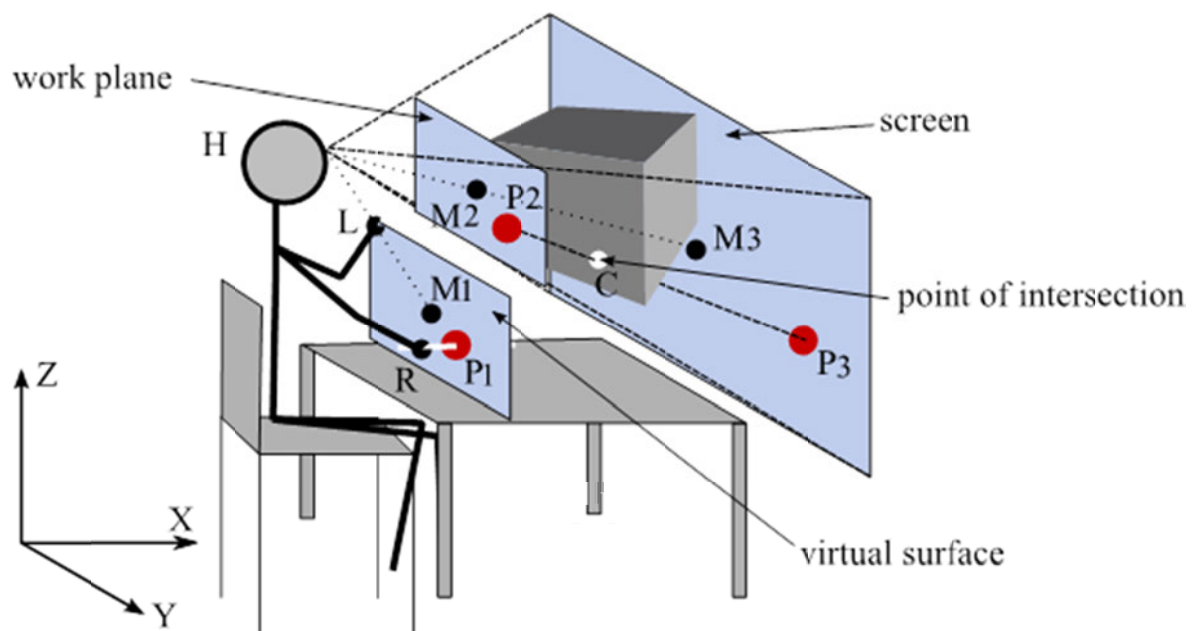


Figure 2. Concept for object selection in a virtual environment

Pre-selection

Tracking the head position and the position of the hands happens through the Kinect sensor. The head position H (see Figure 2) and the vector $\overrightarrow{HM}$ between the head and the centre point of the screen $M3$ are determined for further calculations. First a virtual surface with the centre point $M1$ is created in a defined distance that depends on fixed values and the Z -rotation of $\overrightarrow{HM}$. The orientation of this surface corresponds to the orientation of the screen around the X - and Y -axis, whereas the orientation around the Z -axis is the same as the value of the vector $\overrightarrow{HM}$. The middle of a straight line (white) with the length of 20 centimetres is placed to the position of the right hand R . This line is perpendicular to the virtual surface.

Afterwards it must be determined, if the straight line and the virtual surface intersect. If that condition is true, the intersection point $P1$ is identified. This point is then projected to the respective point $P2$ on the work plane. The edges of this plane are located on the connection line between the head H and the edges of the screen. The distance between H and the centre point of the work plane $M2$ is set to 40 centimetres. By the use of a second projection, the position of the point $P3$ on the screen is calculated. With the help of the positions $P2$ and $P3$ the vector between these points is determined. To provide feedback to the user and tell him in which position on the virtual surface his hand currently is, a small sphere is displayed in the position of $P2$. The line between $P2$ and $P3$ is used to control if it intersects with the actual subassembly. If there is an intersection, the concerned surface is marked by a different colour for visual feedback. The face or the object is the pre-selection of the user.

Selection action

Subsequently the second part of the selection, the “selection action”, must be performed. The recognition should be realised through hand gestures that are captured by the Kinect sensor. Recognising finger gestures with the Kinect currently is a popular research approach [Ren et al. 2011], [Oikonomidis et al. 2011], [Keskin et al. 2013]. The gestures can either consist of individual finger movements and postures or hand states, as explained later. An essential requirement is that the concept shall be sufficiently stable, so the product developer is not dissuaded by inefficient gesture recognition.

Three different gestures are needed: one for picking an element, one for displacing and one for rotating a model. An applicable and stable method that has the ability of simultaneously performing position and orientation changes was not found during research.

The first of the two presented approaches is tracking the position of the finger with an algorithm that identifies the contour of the hand with depth data [Ryan 2012]. Afterwards the contour is searched for curves that represent fingers. With the aim of creating preferably easy performable gestures, three postures were chosen, as seen in Figure 3.

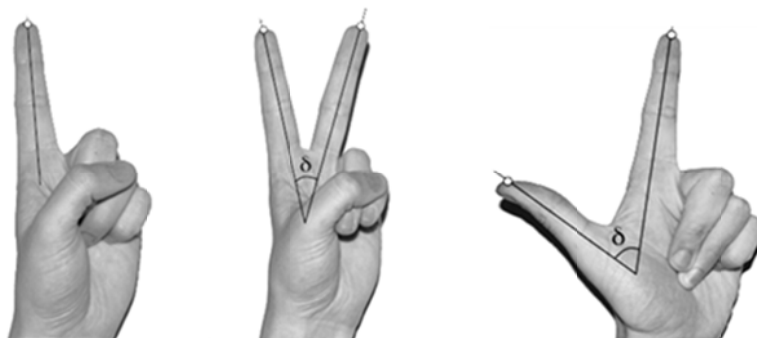


Figure 3. Gestures for picking an element, changing position and translation

The first one, where the index finger is outstretched, is made for picking an element, like a face of the 3D geometry or a menu item. The second gesture, where index and middle finger form a “V”, enables the developer to displace a pre-selected part and the third one, where thumb and index finger form an “L”, analogously makes rotations possible. The distinction between the two subsequent gestures happens through the angle δ between the fingers. For picking an element no motion is necessary,

whereas moving or rotating a subassembly may contain components in all three spatial directions. The hand motion is transferred to the pre-selected model as soon as the V-gesture is performed. The rotation is also controlled by the movement of the hand which performed the gesture and works similar to rotating in common CAD software.

The second alternative approach builds on the “Kinect Interactions” interface that was brought along by the “Kinect for Windows SDK 1.7” [Tashev 2013]. This contains the methods “press” and “grip” that were designed for markerless interaction. The three required interaction elements from Figure 3 are all performed solely with the grip function of both hands. This is triggered, whenever the user forms a fist. Picking an element is done through closing and opening the hand in an interval smaller than 0.4 seconds. When a part is pre-selected and a fist is performed with the right hand, the motion of the hand is transferred to the model analogous to the prior approach. The rotation is modifiable, if both hands are clenched to fists at the same time. The rotation angle is calculated through the change of the vector between the two hands in space. Therefore the difference of this vector between two frames is transmitted to the pre-selected model.

Both finger and hand gesture concepts were implemented and the consequential results were tested by a small group of mechanical engineers. The most important criteria are the stability of the gesture recognition and the ergonomics of the interaction.

Tracking the fingers works pretty well in case of keeping a constant distance and holding the fingers perpendicular to the sensor. Once the hand is moved too fast or too far away from the Kinect (> 140 centimetres), the recognition gets unreliable. Thus discrepancies could come up while performing movements that are necessary for common CAD operations. Without going into details, the algorithm for searching the contour of the hand and the request for stable interaction reveal conflicts of interests.

Regarding the second approach, the detection of the hand state (open or closed hand) is highly reliable in a distance up to two metres. The velocity of the movement and the alignment of the hand to the sensor only have little incidence on the stability of the recognition. The grip and move gesture is quite intuitive and the test persons understood and internalized the procedure rapidly. The selection action and also the displacement of elements also work quite well. Rotating around the Z- and X-axis (see coordinate system in Figure 2) functions properly, while twisting around the Y-axis disclosed some difficulties due to occlusion of one hand.

Summing up the results from the two approaches mentioned above, there are considerable differences in stability and ergonomics. This simplified the decision for using the second concept with the grip gesture. Clenching one’s right fist is intended for displacing, while clenching both hands is made for rotating a subassembly. Performing a short fist gesture picks an element (e.g. a menu item).

Menu navigation

An efficient way to control the stereoscopic assembly system is the induction of pie menus depicted in Figure 4. This is a frequently used technique in virtual environments [Gebhardt et al. 2013]. The handling of a pie menu happens analogous to the face selection of the digital mock-ups utilising the pre-selection via the virtual surface together with the rapidly closing and opening hand gesture. If the hand is positioned over an item of the menu, the developer gets a visual feedback through a coloured highlight. If there is more than one menu level, the submenus can be displayed above the previous.

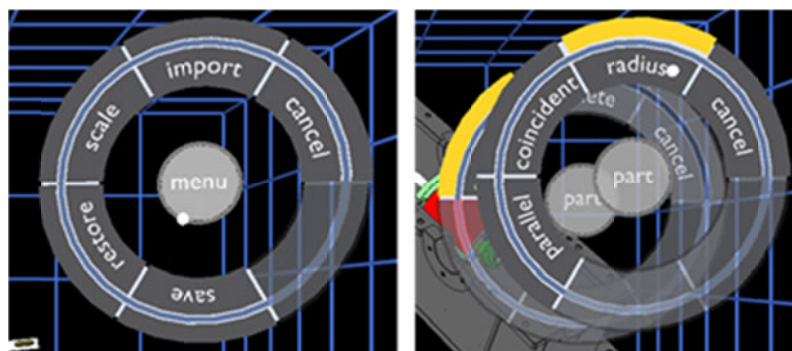


Figure 4. Controlling the system through pie menus

4. Case Study: Gearbox Assembly

To demonstrate the functionality of the CAD-VR integration and the intuitive interaction concept in practical use, the assembling of a gearbox serves as example. At the beginning of the procedure the lower housing section is fixed to the point of origin in the scene (illustrated in Figure 5 on the left side). The next part or module being assembled to the existent assembly can be chosen from an import menu (see Figure 5 on the right side). Subsequently the selected component is integrated to the consisting structure with constraints until its position and orientation is fully defined. Altogether the virtual prototype of the gearbox consists of five subassemblies that are connected by three different constraints.

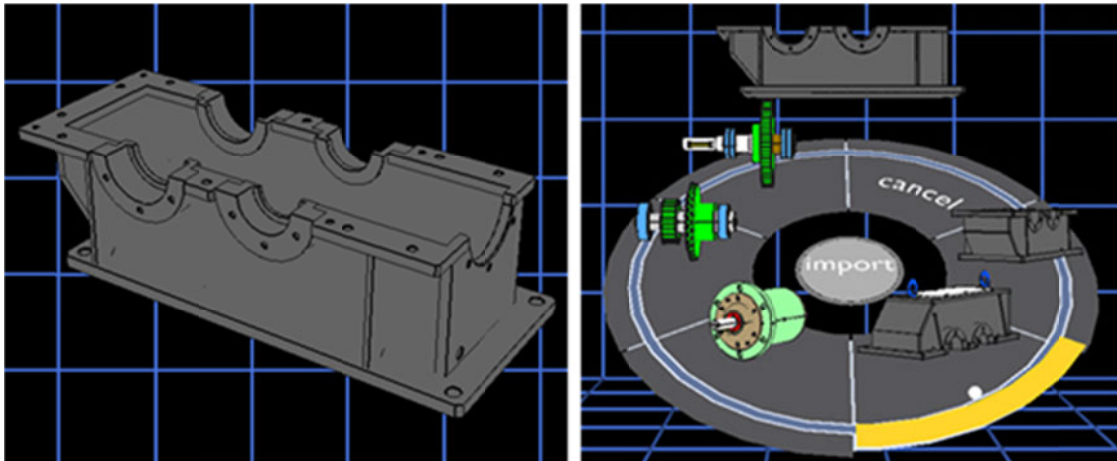


Figure 5. The lower housing section of the gearbox (left) and the import menu (right)

The CAD constraints necessary for the assembly process were implemented prototypically: coincidence, parallelism and coaxiality which are shown in Figure 6. A component always has six degrees of freedom, three for translation and three for rotation. The insertion of a constraint restricts one or more of these degrees of freedom. This means that the part is not movable in the appropriate direction(s) or around certain axes. The movability of the subassemblies in the virtual environment correlates to these restrictions.

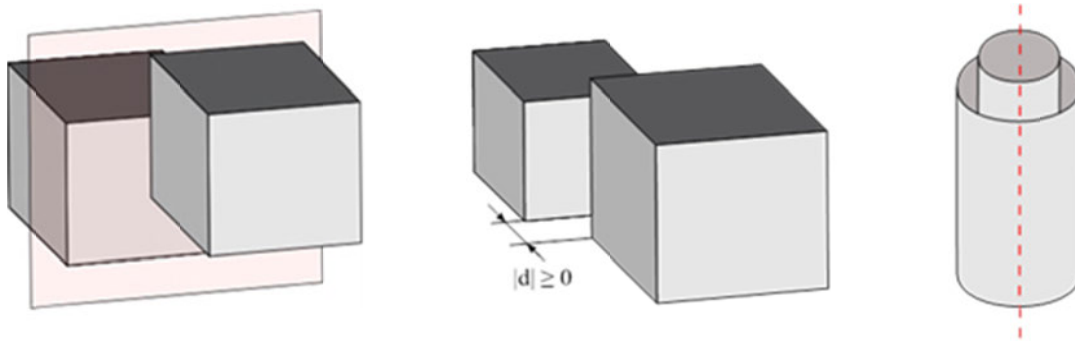


Figure 6. Constraints for assembling the case study: coincidence, parallelism and coaxiality

In engineering systems there are often mounting situations, where a cylindrical surface should be applied to another cylindrical surface, that has a very similar or even the same radius. Therefore an assembly procedure has been designed that provides an easy and intuitive method to insert this kind of constraint. The user first picks the desired cylindrical face and tells the system that a coaxial constraint should be inserted. Then the entire assembly, except the new parts, is scanned for potential surfaces that have similar radii (e.g. $\pm 3\%$) as the selected. Afterwards the selected surfaces' position and orientation is permanently compared to the found counterparts. Model elements that should be installed can freely be moved and rotated by the user. During this process a red semi-transparent bounding box is displayed around the subassembly with the intention of providing visual feedback to

the user (see Figure 7). As soon as the difference in position and orientation undercuts a certain threshold, the component will automatically dock to the appropriate position and the bounding box switches to green colour. Simultaneously a pie menu appears where the constraint can be confirmed, cancelled and the orientation of the subassembly can be flipped on the central axis.

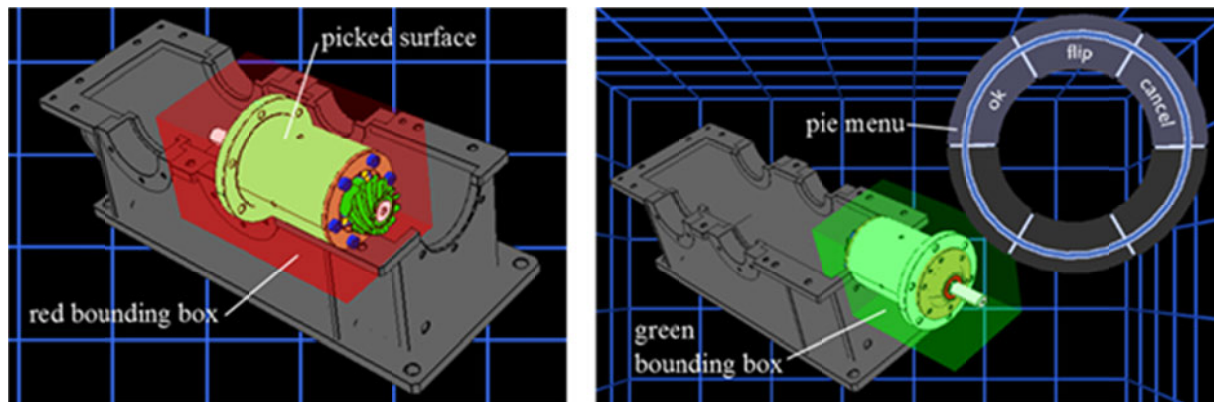


Figure 7. Inserting a coaxial constraint

Coincident and parallel constraints can be inserted through the common procedure of selecting the surfaces of the appropriate part and the counterpart. Integrating a coaxial constraint between cylindrical faces with a different radius works similar. Finally, when the design group is fully assembled (see Figure 8), it can be saved for further consideration.

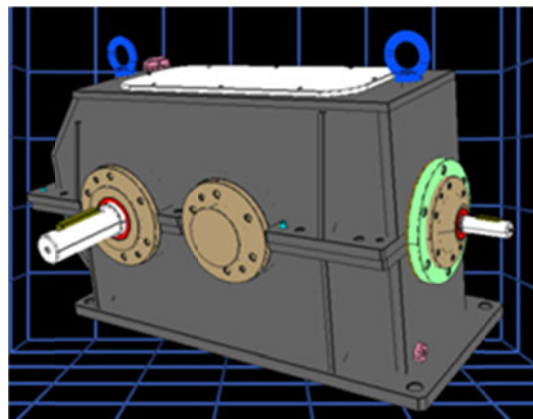


Figure 8. Fully assembled gearbox

5. Conclusion and Outlook

The idea presented in this paper is to combine the advantages of a virtual reality environment with the product developer's common workplace by using a new VR-CAD framework. This framework has the inherent potential for decisive improvements, both in visualisation and interaction with digital mock-ups. Especially the novel interaction concept – in combination with the stereoscopic view – could simplify workflow operations by intuitively modelling a part and assembling it through the utilisation of gestures.

In a first step several approaches for creating CAD-VR environments and concepts for interacting with the displayed content are considered. These approaches are evaluated with a view on the application in a hybrid workplace and subsequently refined to a CAD-VR environment. To demonstrate the usability and the efficiency of the system a gearbox is assembled from subassemblies exclusively by using gestures. The result is a hybrid environment that enables the product developer to utilise the benefits of stereoscopic head-tracked visualisation while putting together an assembly without disturbing usual workflows.

Further improvements in gestural interaction could be achieved through the utilisation of the enhanced precision of the Leap Motion sensor. It enables the tracking of finger positions and the detection of tiny finger movements [Weichert et al. 2013]. The sensor's relatively high accuracy is well suited for an application in a desktop configuration. In addition the forthcoming new generation of the Kinect sensor could have the potential to track the users hand movements more precisely. An approach that utilises direct handling for the interaction with the virtual product could gain from using an AR-/ VR-head-mounted display, where a virtual hand model is faded in. In conjunction with the new emerging opportunities for interaction, additional intuitive methods for inserting constraints could be developed.

References

- Berta, J., "Integrating VR and CAD", *Computer Graphics and Applications*, Vol. 19, No. 5, 1999, pp. 14-19.
- Bowman, D. A., Kruijff, E., LaViola Jr, J. J., Poupyrev, I., "An introduction to 3-D user interface design", *Presence: Teleoperators and virtual environments*, Vol. 10, No. 1, 2001, pp. 96-108.
- Collette, B., "FreeCAD (how-to)", Falck, D., Birmingham, 2012.
- Cruz-Neira, C., Sandin, D. J., DeFanti, T. A., "Surround-screen Projection-based Virtual Reality: The Design and Implementation of the CAVE", *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques - SIGGRAPH '93*, ACM New York USA, 1993, pp. 135-142.
- Gebhardt, S., Pick, S., Leithold, F., Hentschel, B., Kuhlen, T., "Extended Pie Menus for Immersive Virtual Environments", *Visualization and Computer Graphics*, Vol. 19, No. 4, 2013, pp. 644-651.
- Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K., Hart, P., "VADE: a virtual assembly design environment", *Computer Graphics and Applications*, Vol. 19 No. 6, 1999, pp. 44-50.
- Kaufmann, H., Schmalstieg, D., Wagner, M., "Construct3D: A Virtual Reality Application for Mathematics and Geometry Education", *Education and Information Technologies*, Vol. 5, No. 4, 2000, pp. 263-267.
- Keskin, C., Kırac, F., Kara, Y. E., Akarun, L., "Real Time Hand Pose Estimation Using Depth Sensors", Springer London, 2013.
- Knöpfle, C., Voß, G., "An intuitive VR user interface for design review", *Proceedings of the working conference on Advanced visual interfaces*, ACM New York USA, 2000, pp. 98-101.
- Oikonomidis, I., Kyriazis, N., Argyros, A. A., "Efficient model-based 3D tracking of hand articulations using Kinect", *British Machine Vision Conference*, Dundee UK, 2011, pp. 1-11.
- Pavlik, R. A., Vance, J. M., "VR JuggLua: A framework for VR applications combining Lua, OpenSceneGraph, and VR Juggler", *Software Engineering and Architectures for Realtime Interactive Systems (SEARIS)*, Orange County CA, 2012, pp. 29-35.
- Ren, Z., Meng, J., Yuan, J., Zhang, Z., "Robust Hand Gesture Recognition with Kinect Sensor", *Proceedings of the 19th ACM International Conference on Multimedia*, ACM USA, 2011, pp. 759-760.
- Rodrigues, P. G., Raposo, A. B., Soares, L. P., "A Virtual Touch Interaction Device for Immersive Applications", *The International Journal of Virtual Reality*, Vol. 10, No. 4, 2011, pp. 1-10.
- Ryan, D. J., "Finger and Gesture Recognition with Microsoft Kinect", *Master's thesis in Computer science*, University of Stavanger Norway, 2012.
- Tashev, I., "Kinect Development Kit: A Toolkit for Gesture- and Speech-Based Human-Machine Interaction", *Signal Processing Magazine*, Vol. 30, No. 5, 2013, pp. 129-131.
- Weichert, F., Bachmann, D., Rudak, B., Fisseler, D. "Analysis of the accuracy and robustness of the leap motion controller", *Sensors*, Vol. 13, No. 5, 2013, pp. 6380-6393.
- Weidlich, D., Cser, L., Polzin, T., Cristiano, D., Zickner, H., "Virtual reality approaches for immersive design", *CIRP Annals-Manufacturing Technology*, Vol. 56, No. 1, 2007, pp. 139-142.
- Wendrich, R. E., "The creative act is done on the hybrid machine", *Proceedings of the International Conference on Engineering Design 2013 - ICED13*, Seoul South Korea, 2013, pp. ICED13/443.
- Zeng, W., "Microsoft Kinect Sensor and Its Effect", *MultiMedia*, Vol. 19, No. 2, 2012, pp. 4-10.

Marius Fechter, M.Sc., Research Associate
 University of Erlangen-Nuremberg, Chair of Engineering Design
 Martensstr. 9, 91058 Erlangen, Germany
 Telephone: +49-(0)9131/85-23217
 Telefax: +49-(0)9131/85-23223
 Email: fechter@mfk.fau.de
 URL: <http://www.mfk.fau.de>