

Assembly Simulation on Collaborative Haptic Virtual Environments

R. Iglesias, E. Prada
Fundación Labein - Tecnalia
Parque Tecnológico Zamudio
Edificio 700
Spain E-48160, Derio

{riglesias,eprada}@labein.es

A. Uribe, A.Garcia-Alonso
U. of the Basque Country
Facultad Informática
Manuel de Lardizabal 1
Spain E-20018, San Sebastián

alex.galonso@ehu.es

S. Casado, T. Gutierrez
Fundación Labein - Tecnalia
Parque Tecnológico Zamudio
Edificio 700
Spain E-48160, Derio

{sara, tere}@labein.es

ABSTRACT

Currently, Virtual Environments (VEs) are used within engineering industry: physical prototypes or mock-ups are replaced by virtual prototypes. Increasingly, these VEs also allow designers and engineers to carry out assembly/disassembly processes and assembly sequences before any physical prototype is built. Moreover, different designers or engineers, which may be situated in the same or in geographically dispersed locations, often collaborate in the design of products. This allows developing more complex products within a shorter time scale and lowered costs. On the other hand, the utilization of haptic feedback has been found to significantly enhance task performance, for instance, in assembly tasks. In this paper we describe an assembly simulation application on a collaborative haptic virtual environment, where several users interact with virtual models to perform assembly operations within the same virtual scene. The paper also summarizes results achieved with experiments which evaluated different collaborative architectures. Furthermore, it reports on the goals that can be achieved and the limitations for haptic collaborative interaction in each case.

Keywords

Virtual reality; Collaborative virtual environment; Virtual prototyping; Haptic feedback; Assembly simulation.

1. INTRODUCTION

The traditional design systems (CAD, CAM and CAE) allow generating 3D designs and simulating the behavior of the product and part of the manufacturing process, such as assembly/disassembly (A/D) processes and sequences for training. However, they do not integrate all the physical processes of the real world. The use of haptic devices (sense of touch) is a powerful technology that can enhance and solve some of the limitations of the traditional simulation systems.

Haptic devices are generally used to indicate a class of mechanical system that is intended to replicate forces and local continuous stimuli on specific areas of human body: finger, hand or body. Today, several

haptic devices with different specifications can be available, such as, PHANToM Premium (Figure 1), PHANToM Omni and GRAB. PHANToM is developed and distributed by SensAble Technologies (Cambridge MA, USA). GRAB, which provides a larger workspace and two points of contact, was developed by PERCRO (Scuola Superiore Sant'Anna, Italy).

These days, haptic technology is an emerging field, that is being successfully applied to a wide range of applications, for instance, training simulators [Bas01a], visually impaired people applications [Igl04a], entertainment and gaming [Zho04a], as well as industrial design and maintenance [Bor04a], [Pet04a], [Igl06a]. In this latter application, the utilization of haptic devices has been found to significantly improve operation effectiveness in assembly tasks [Bas00a], [Sal00a], [Pet04a].

On the other hand, nowadays, products are increasingly being developed by geographically dispersed design teams. These may be located in different partner companies, or different offices of the same company, perhaps even in different countries. On large projects, different design teams meet regularly for preliminary reviews, design reviews, defect reviews and so on. It is becoming

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
Copyright UNION Agency – Science Press, Plzen, Czech Republic.

strategically important to be able to link distributed design teams, to permit concurrent access and modification of design models from different geographical locations.

There are different types of distributed collaborative virtual environments (CVEs), for instance, some provide collaborative visualization while, others, not so many, also consider haptic interaction. The developments of Hagsand [Hag96a], Borro [Bor00a] and Greenhalgh [Gre00a] are focused on distributed visualization: a user interacts with a virtual model while others can watch it. Other examples about distributed VEs can be found in Mclaughlin [Mcl02a] and Burdea [Bur03a].



Figure 1. A user interacting with PHANToM Premium.

The work presented in this paper provides a helpful tool for the design and maintenance teams. The designers would be able to check the validity of the design, simulate A/D processes in order to simplify the process, avoid interferences and define A/D procedures with haptic interaction. The procedures defined by the design team could also be the base for training operators in new or complex assembly and maintenance (A/M) tasks.

This paper describes an application to simulate assembly operations on a collaborative virtual environment (CVE). This application allows different users to analyze in real-time new products, A/M operations without using physical models, by means of realistic navigation, visualization and interaction with the virtual models by means of traditional devices (i.e. keyboard, mouse) or haptic devices.

Sections 2 and 3 explain the toolkits in which the assembly application is based on. The main challenges on collaborative haptic virtual environment are presented in Section 4. It also includes one example and the results that lead to the conclusions.

2. TOOLKITS: DATum AND ASSEMBLY SIMULATOR

The research described later on in this paper requires the integration of three toolkits: *DATum*, an *Assembly Simulator* and a *Haptic Assembly Simulator*. The next section deals with the third toolkit. The three toolkits described between this section and the following one allows analyzing new products in real-time and simulating virtual A/M operations via keyboard, mouse or haptic devices.

DATum is an object oriented variational non-manifold geometric modeler developed by LABEIN, with a STEP translator compliant with ISO 10303-AP203 (International Standard for the representation and the exchange of product data between different CAD systems).

DATum uses a hybrid representation scheme between the two most common representations within the field of the Geometric Modeling: the Constructive Solid Geometry [Req77a] and the Boundary Representation [Man84a], exploiting the advantages of each one of these representations. In this way, a model can be created through Boolean operations (union, intersection and difference) between other two models, and it has always associated a boundary representation. The boundary representation of a model provides both geometric and topological information. In relation to the geometry, *DATum* supports both basic geometry (conics and quadrics) and complex geometry (NURBS curves and surfaces). Its topological structure is based on the Weiler structure [Wei86a].

DATum is a non-manifold modeler [Wei86a]. This capability allows to represent solid, surface and wireframe models in a unified and simultaneous way and to deal with the “region” concept. In this way, a model can be composed by several regions associated, for example, to different materials.

It is also a variational modeler. The variational geometry is based on the definition and modification of geometric models through a set of functional restrictions, instead of the classical parameters. In this way, the model can include the design intent.

On top of *DATum*, LABEIN has developed an *Assembly Simulator* that combines direct manipulation techniques, collision detection, automatic assembly constraint recognition and management within a unified framework to allow undertaking maintenance operations of mechanical assemblies interactively. It consists mainly of two modules: collision detection and assembly constraint recognition.

The collision detection module detects any collision between an object that is being moved by the user (through translations and/or rotations) and any other object of the working space. In this way, the object can not penetrate into another and only real feasible movements are allowed. The implemented algorithm is based on the ‘RAPID’ library implemented by Gottschalk [Got96a]. This is a polygon interference detection library for large environments. The degree of performance of this algorithm is quite related to the quality of the triangles, so the more regular the triangles are a higher performance will be achieved. Algorithms to get suitable triangles were developed using the basic functionality of *DATum*.

When a collision is detected the user is not allowed to move in the collision direction, but he must change the movement direction in order to avoid the collision. Sound aids and changes of colors have been implemented to warn the user when a collision is produced.

The *Assembly Simulation* module allows the automatic recognition of the potential constraints between a model that is being moved by the user and the rest of the components moved of the mechanical assembly. Once the system detects a constraint, the movement of the model will be constrained to satisfy the active constraint detecting also the collisions with the other model of the workspace [Gut98a], [Bar98a], [Bar99a]. This module makes use of the collision detection algorithms explained above and it is based on the information about the adjacency relationships of the topological entities of the models, for example how the faces are connected to detect if a face generates a hole or a protrusion. The following A/D constraint methods are depicted in Figure 2: objects along a common axis (pin-hole), and objects along coincident planar faces. Other constraints are pin-multiple holes, hole-pin, pin-pin and hole-hole.

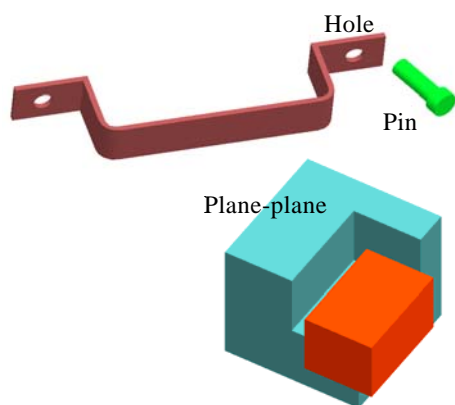


Figure 2. Assembly/Disassembly: pin-hole and plane-plane.

3. TOOLKITS: HAPTIC ASSEMBLY SIMULATOR

The integration of a haptic device within *DATum* allows the user to interact with 3D designs in a new and more realistic way than the traditional systems. The user can not only view the objects designed, but also interact with them: touching, grasping and moving them within the virtual scene, detecting and feeling the possible collisions and assemblies among models. Experiments used different haptic devices: the PHANToM device, PHANToM Omni and GRAB device (see Section 1). Haptic devices were used for the following manipulation tasks: touch, move and collide, and Assembly/Disassembly operations.

The user can touch any 3D model and move itself along its external surface, detecting its edges and corners. The algorithm to touch and interact with a virtual object by means of a haptic device is based on the analysis of the position of the user’s finger (recovered by the haptic device) with respect to the object to check if the point is inside or outside of the object. In this case, the force sent by haptic device will be proportional to the penetration depth of the user’s finger into the virtual object and normal to the object surface.

Any object, that can be touched, can also be *grasped and moved* (through translations and rotations, called transformations in general) by the user along the virtual workspace.

This utility also detects any collision between the object that is being moved and any other object of the workspace. A model cannot penetrate into another and only real feasible movements are allowed.

The workflow repeats the following steps:

- 1) Calculate the transformation described by the movement of the user’s finger (movement of the end-effector of the haptic device): translation and/or rotation.
- 2) Study if, in the new position, the object is colliding with any other object of the workspace.
- 3) If there is not collision, apply the transformation of the object. Calculate the force to be sent to the user, depending on the result of the collision detection. In order to provide a more realistic interaction, the object weight has also been implemented. This has required the implementation of some algorithms to calculate the volume and the area of any 3D object. If there is not any collision, the force sent by the haptic device corresponds to the object weight. Whereas if the object is colliding, a force

opposing to the movement direction to come back to the last position.

Assembly/Disassembly allows simulating A/M operations of a mechanical assembly and it is based on the automatic recognition of constraints provided by the Assembly Simulator explained above. Most of the previous utilities (touch, move, detect collisions) are enclosed here.

The workflow repeats the following steps:

- 1) Calculate the transformation described by the movement of the user's finger (movement of the end-effector of the haptic device): translation and/or rotation.
- 2) This transformation is sent to the *Assembly Simulator* to study if, in the new position, the object is colliding with any other object of the workspace, breaking an existing constraint or satisfying a new one.
- 3) The *Assembly Simulator* calculates the adequate movement to be applied to the object and information about the constraints that are satisfied. With this information the force to be sent to the user is calculated.

Three types of sliding forces were implemented to support assembly methods: on a surface, on a line and on a point.

- 1) *Sliding forces on a surface* allow two objects to remain in contact along a common surface. An external force can compel the user to move his finger, and so the object fixed to it, on a given surface. The user can move freely along the surface but she can not move away from it (unless the user exerts an incremental force to break the active constraint). The algorithm to calculate this force is based on the computation of the point of minimum distance from a point to a surface. In this case, the force corresponds to the vector defined by the user's position and the point of minimum distance on the surface.
- 2) *Sliding forces on a line* allow two objects to remain in contact along a common line. As in the previous case, the user can move his finger, and so the object fixed to it, on a given line. The user can move freely along the line but he can not move away from it (unless the user exerts an incremental force). The algorithm to calculate this force is based on the computation of the point of minimum distance from a point to a line. In this case, the force corresponds to the vector defined by the user's position and the point of minimum distance on the line.

- 3) *Sliding forces on a point* allow two objects to remain in contact along a common point. With this type of force, the user can only rotate the object fixed to her finger about its own axis, unless the user exerts an incremental force. In this case, the force corresponds to the vector defined by the user's position and the point.

4. CVEs FOR ASSEMBLY SIMULATION

There are different types of *distributed collaborative virtual environments* (CVEs), for instance, some only provide collaborative visualization whereas, others, not so many, also consider haptic interaction. This section analyses the problem of assembly simulation on CVEs where users can simultaneously interact within the same scene using traditional or haptic devices. First, CVEs are considered, and then a practical case is analyzed. The last subsection summarizes different system architectures that were experimented extending the toolkits described in sections 2 and 3.

4.1 Collaborative assembly application

A CVE implies a distributed system that allows geographically separated users (computers) to communicate and/or interact within the same virtual scene through connected networks such as, LAN or the Internet.

In our application, mechanical assemblies can be designed within *DATum*, or imported from another CAD system through STEP files.

Each client can interact within the virtual scene, either moving freely and touching models or grasping a model. In this latter case the user feels collisions with other models along their movement and the system may guide the user to undertake an assembly method. Users can interact with the virtual scene using different devices: mouse, keyboard or haptic devices (see Section 1).

Each client replicates the same virtual scene, managing their visualization, from a different point of view. It is not necessary that all users connect themselves simultaneously to the work session. A user can be joined the work group when it is considered opportune; when a new user connects to the server, this user will receive the state of the environment upon connection.

Several clients may undertake different actions on the same CVE. For instance, one client can freely move, while other can move a model satisfying a constraint (assembly), and another different client can collide with a different object. During these tasks, *consistency*, that is virtual scene synchronization, must be guaranteed for all clients.

Whenever a user tries to move the grasped object, the system must validate the movement considering potential collisions with the rest of the objects within the scene. Constraints must also be considered depending on the type of assembly method chosen by the client. It is important the way in which communications, data and processes are distributed between server and clients to provide a realistic interaction among users and to compute immediate responses as in the case of haptic devices with high frequencies.

Forces applied to the user by the haptic device must provide an adequate sensation. This topic has been deeply addressed by other authors [Bur03a].

User interaction on a distributed assembly application must be similar to the standalone (not networked). However, some restrictions necessarily appear, i.e. opening/closing virtual environments, an grasped by one user can not be simultaneously grasped by another user.

Problems regarding network communications also play an important role in the design of specific implementations. However, the inevitable conditions of the network (e.g. latency, jitter, background traffic) affect distributed applications in a different way as it is considered in the next subsection.

4.2 Practical case

This subsection describes a specific experiment, where the virtual scene was an aeronautical assembly (Figure 3) provided by an engineering company, SENER.

This CVE allowed the simulation of assembly tasks with simultaneous interaction of several users. This was achieved by adopting a client-server architecture, as Figure 4 shows. The distributed components were developed with ORBacus (an Object Request Broker that is compliant with CORBA specification).

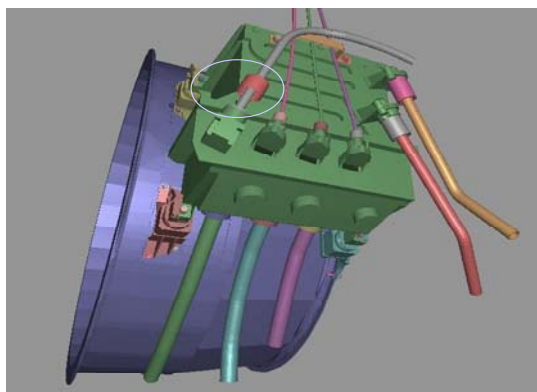


Figure 3. Aeronautical assembly (an electrical box for an aircraft engine) provided by SENER.

A server administrates all data received from clients: new virtual scene, request of grasping a model, request of transformation of a model, new client and so on. The server manages the selection of models to avoid that two different users simultaneously grasp the same model.

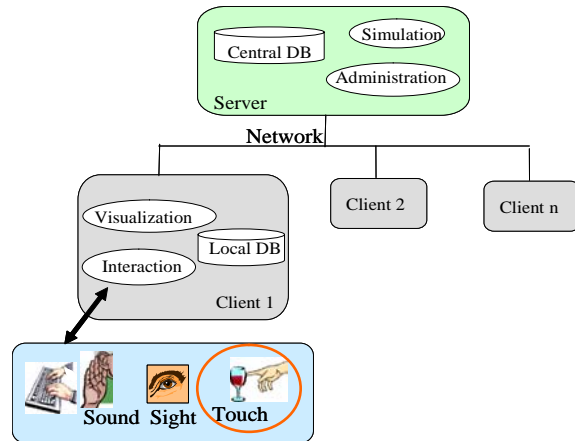


Figure 4. Client-Server architecture.

The original assembly was done in Pro-E and read into *DATum* through STEP. A distributed session was run between two users through a local area network simulating the assembly process. A user could interact using a haptic device: grasping a model, feeling its weight, detecting the collisions with other objects along their movement and simulating the assembly methods explained above. Meanwhile, the other user could interact with the shared scene with the keyboard.

Previously, with this design an assembly problem was found [Car03a]. During the assembly path a collision with the green box does not allow to finish the assembly process (Figure 3). Therefore, a re-design was needed in order to avoid this problem.

Using this architecture the application worked properly and consistency between clients was guaranteed all the time.

4.3 Architectures for distributed environments

Some research has been done on CVEs where virtual scene synchronization (consistency), effective and compelling haptic feedback (quality of force feedback) and scalability continue to be enormous challenges. There are different architectures to support distributed systems: peer-to-peer [Cla01a], client-server [Sin99a] or a mixture of them [Bor00], [Mar06a]. Marsh et al. [Mar06a] analyze different architectures supporting haptic interaction on CVEs and provide an updated review of the research performed to deal with the challenges described in the Section 4.1.

Iglesias et al. [Igl05a] compare three different client-server architectures for assembly simulation on a CVE described in this paper. That research studied how could be distributed the simulation workload between server and clients. Two architectures provided interesting results. In architecture A3 clients deal with the validation of movement and consistency maintenance is managed at server side. On the other hand, in architecture A1 consistency is trivially achieved because server sequentially validates all movements. When there are more than two users, the architecture A3 had a better performance.

With client-server architectures consistency is easily achieved, it is calculated once either at server side or trivially guaranteed if server validates all movements. No complex synchronization mechanism is required. However, in such architectures, haptic interaction may specially be affected due mainly to network conditions (i.e. delay, jitter). This plays an important role in situations where force feedback at a user depends on the actions of another user (*dependent interaction*). This case happens when two users grasp a different object and for instance, a user assemble the grasped object into another one being held by a different user (*remote assembly*). Not only performing an assembly, but when both grasped objects collide and users should feel the corresponding collision force feedback (*dependent collision*).

As a consequence, these client-server architectures limit the use of haptic interaction to extremely good network conditions between the client and server. If that previous condition is not achieved, users should avoid working in the vicinity of objects grasped by other users. The haptic interaction may be affected in case of dependent interaction. Marsh et al. [Mar06a], which used the same assembly application proposed in this paper, reported on a hybrid architecture that only supports haptic interaction for simultaneous cooperative haptic tasks over a low delay network (between the client and server) and in other case, users collaborate by taking turns.

As a result of the impact of network conditions on haptic interaction with client-server architectures, in Iglesias [Igl06a] a peer-to-peer architecture is presented. This paper aims to achieve a higher degree of collaboration, at least between two users: to achieve nearby collaboration, such as, to carry out remote assemblies and maintain consistency even when network conditions get worse. A new consistency-maintenance scheme was proved to maintain consistency. Results were satisfactory with different network conditions.

5. CONCLUSIONS

In recent years, VEs are used within engineering industry: physical mock-ups are being replaced by virtual prototypes. Increasingly, these VEs allow designers and engineers to carry out assembly/disassembly processes before any physical prototype is built. On the other hand, several users often collaborate in the design of a product, evaluating assembly sequences, which may be situated in the same or in geographically dispersed locations.

Moreover, the utilization of haptic devices allows users to have physical interaction with digital mock-ups. And the haptic feedback (sense of touch) has been found to significantly enhance task performance, such as these assembly tasks.

We describe an application to carry out assembly operations on a collaborative virtual environment with the use of keyboard, mouse or haptic devices. The Haptic Assembly Simulator recognizes automatically collisions, assembly constraints and replicates properly forces on user's fingers to provide an effective interaction.

Section 4.3 shows which collaborative interaction goals can be achieved using different network topology architectures and strategies. Client-server architectures provide good results if network conditions are good enough and objects managed by users are sufficiently separated. A peer-to-peer architecture has been proposed in order to support a collaborative assembly task with certain network delay. With this architecture, a remote assembly (a user assembles an object into another object grasped by a different user) can be performed even with the worsening of network conditions. A consequence is that, although there is not a global solution to the problem yet, different network topologies may be adopted to build applications with specific goals.

6. ACKNOWLEDGMENTS

This work has been partly supported by ENACTIVE Interface, European project IST-2002-002114; Gipuzkoako Foru Aldundia project (13/2005) with FEDER funds; Spanish Ministry of Education and Science, grant TIN2006-14968-C02-01; and AMIGUNE Basque Government project (2005-2007).

7. REFERENCES

- [Bar98a] Barbero, J.I., Gutiérrez, T. and Eguidazu, A., Haptic Virtual Prototypes for Assembly Simulation, in ISATA Conference No.31, pp. 109-117, 1998.
- [Bar99a] Barbero, J.I., Gutiérrez, T., Alvarez, A. and Carrillo, A., Assembly Simulation Tools:

- Tolerance Analysis and Haptic Virtual Environment, in EAEC European Automotive Congress, 1999.
- [Bas00a] Basdogan, C., Ho, C.H., Srinivasan, M. A. and Slater, M., An Experimental Study on the Role of Touch in Shared Virtual Environments. *ACM Transactions on Computer-Human Interactions*, 7, pp. 443-460, 2000.
- [Bas01a] Basdogan, C., Ho, C., Srinivasan, M.A., Virtual Environments for Medical Training: Graphical and Haptic Simulation of Common Bile Duct Exploration. *IEEE/ASME Transactions on Mechatronics*, 6, pp. 267-285., 2001.
- [Bor00a] Borro, D., Matey, L., Sánchez, H., Recio, I., Garcia-Alonso, A., CSCW for foundry design using Java 3D, demonstration at ACM 2000 Conference on Computer Supported Cooperative Work, 2000.
- [Bor04a] Borro, D., Savall, J., Amundarain, A., Gil, J.J., García-Alonso, A., and Matey, L., Large Haptic Device for Aircraft Engine Maintainability. *IEEE Computer Graphics and Applications*, 24, 70-74, 2004.
- [Bur03a] Burdea, G.C., and Coiffet, P., *Virtual Reality Technology*, John Wiley and Sons, 2003.
- [Car03a] Carrillo, A.R., Beloki, O., Casado, S., Gutierrez, T., Barbero, J.I., Virtual Assembly and Disassembly Simulation on a Distributed Environment, in *Proceedings of Virtual Concept*, 2003.
- [Cla01a] Clark, D., Face-to-Face with Peer-to-Peer Networking. *Computer*, Vol. 34, No. 1, pp. 18-21, 2001.
- [Got96a] Gottschalk, S., Lin, M.C. and Manocha, D., OBBTree: A Hierarchical Structure for Rapid Interference Detection, in *Proc. of ACM Siggraph*, pp. 171-180, 1996.
- [Gre00a] Greenhalgh, C., Purbrick, J., and Snowdon, D., Inside MASSIVE_3: Flexible support for data consistency and world structuring, in *ACM Conference on Collaborative Virtual Environments*, pp. 119-127, 2000.
- [Gut98a] Gutiérrez, T., Barbero, J.I and Eguidazu, A., Virtual Assembly and Disassembly, in *IFAC workshop on intelligent Assembly and Disassembly*, pp. 35-40, 1998.
- [Hag96a] Hagsand, O., Interactive Multiuser VEs in the DIVE System. *IEEE Multimedia*, 3, pp. 30-39, 1996.
- [Igl04a] Iglesias, R., Casado, S., Gutiérrez, T., Barbero, J.I., Avizzano, C.A., Marcheschi, S., Bergamasco, M., Computer graphics access for blind people through a haptic and audio virtual environment, in *IEEE International Workshop on Haptic Audio Visual Environments and their Applications*, pp. 13-18, 2004.
- [Igl05a] Iglesias R., Casado S., Gutierrez T., Barbero J.I. , Sanchez E., Garcia-Alonso A. “Architectures for distributed multimodal virtual environments”, *International Conference on Manufacturing Research, England (2005a)*.
- [Igl06a] Iglesias, R., Carrillo, A., Casado, S., Gutiérrez T. and Barbero, J.I., Virtual Assembly Simulation in a Distributed Haptic Virtual Environment, in *International conference on Advanced Design and Manufacture*, 2006.
- [Igl06b] Iglesias R., Casado S., Gutiérrez T., García-Alonso A., Meng K., Yu W. and Marshall A. “A Peer-to-peer Architecture for Collaborative Haptic Assembly”, *Proceedings of the 10-th IEEE International Symposium on Distributed Simulation and Real Time Applications*, pp. 25-34, 2006.
- [Man84a] Mantyla, M., A Note on the Modelling space of Euler operators. *Computer Vision, Graphics and Image Processing*, 26, 45-60, 1984.
- [Mar06a] Marsh J., Glencross M., Pettifer S. and Hubbard R., “A network architecture supporting consistent rich behaviour in collaborative interactive applications”, *IEEE Transactions on visualization and computer graphics*, Vol. 12, No. 3, pp. 405-416, May 2006.
- [Mcl02a] Mclaughlin, M.L., Hespanha, J.P., Sukhatme, G.S., *Touch in Virtual Environments: Haptics and the Design of Interactive Systems*, Prentice Hall, 2002.
- [Pet04a] Petzold, B., Zaeh, M.F., Faerber, B., Deml, B., Egermeier H., Schilp J. and Clarke, S., A Study on Visual, Auditory, and Haptic Feedback for Assembly Tasks. *Presence: Teleoperators & Virtual Environments*, 13, pp. 16-21, 2004.
- [Req77a] Requicha, A.A.G. and Voelcker, H.B., *Constructive Solid Geometry*. Tech. Memo 25, Production Automation Project, University of Rochester, 1977.
- [Sal00a] Sallnäs, E.L., Supporting Collaboration in Distributed Environments by Haptic Force Feedback. *ACM Transactions on Computer-Human Interaction*, 7, pp. 461-476, 2000.
- [Sin99a] Singhal, S. and Zyda, M., *Networked Virtual Environments Design and Implementation*, ACM Press SIGGRAPH Series, Addison Wesley, 1999.
- [Wei86a] Weiler, K., Topological structures for Geometric Modeling. PhD thesis, Dept. Comput. Syst. Engr., Rensselaer Polytechnic Institute, 1986.
- [Zho04a] Zhou, S., Wentong, C., Lee, B. and Turner, S. J., Time-Space Consistency in Large-Scale Distributed Virtual Environments. *ACM Transactions on Modeling and Computer Simulation*, 14, pp. 31-47, 2004.

