

9-2007

Combining Geometric Constraints With Physics Modeling for Virtual Assembly Using SHARP

Abhishek Seth
Iowa State University

Judy M. Vance
Iowa State University, jmvance@iastate.edu

James H. Oliver
Iowa State University, oliver@iastate.edu

Follow this and additional works at: http://lib.dr.iastate.edu/me_conf



Part of the [Computer-Aided Engineering and Design Commons](#)

Recommended Citation

Seth, Abhishek; Vance, Judy M.; and Oliver, James H., "Combining Geometric Constraints With Physics Modeling for Virtual Assembly Using SHARP" (2007). *Mechanical Engineering Conference Presentations, Papers, and Proceedings*. 39.
http://lib.dr.iastate.edu/me_conf/39

This Conference Proceeding is brought to you for free and open access by the Mechanical Engineering at Iowa State University Digital Repository. It has been accepted for inclusion in Mechanical Engineering Conference Presentations, Papers, and Proceedings by an authorized administrator of Iowa State University Digital Repository. For more information, please contact digirep@iastate.edu.

Combining Geometric Constraints With Physics Modeling for Virtual Assembly Using SHARP

Abstract

This research combines physics-based and constraint-based approaches for virtual assembly simulations where geometric constraints are created or deleted within the virtual environment at runtime. In addition, this research provides a solution to low clearance assembly by utilizing B-Rep data representation of complex CAD models for accurate collision/physics results. These techniques are demonstrated in the SHARP software (System for Haptic Assembly and Realistic Prototyping). Combining physics-based and constraint-based techniques and operating on accurate B-rep data, SHARP can now assemble parts with 0.001% clearance and can accurately detect collision responses with 0.0001mm accuracy. Case studies are presented which can be used to identify the suitable combination of methods capable of best simulating intricate interactions and environment behavior during manual assembly.

Keywords

VRAC, Physics, Modeling, Virtual assembly

Disciplines

Computer-Aided Engineering and Design

DETC2007-34681

COMBINING GEOMETRIC CONSTRAINTS WITH PHYSICS MODELING FOR VIRTUAL ASSEMBLY USING SHARP

Abhishek Seth

Department of Mechanical Engineering
Virtual Reality Applications Center
Iowa State University
Ames, IA 50011
abhiseth@iastate.edu

Judy M. Vance

Division of Civil, Mechanical and Manufacturing
Innovation
National Science Foundation
Arlington, VA 22230
jmvance@nsf.gov
ASME Fellow

James H. Oliver

Department of Mechanical Engineering
Virtual Reality Applications Center
Iowa State University
Ames, IA 50011
oliver@iastate.edu
ASME Fellow

ABSTRACT

This research combines physics-based and constraint-based approaches for virtual assembly simulations where geometric constraints are created or deleted within the virtual environment at runtime. In addition, this research provides a solution to low clearance assembly by utilizing B-Rep data representation of complex CAD models for accurate collision/physics results. These techniques are demonstrated in the SHARP software (System for Haptic Assembly and Realistic Prototyping). Combining physics-based and constraint-based techniques and operating on accurate B-rep data, SHARP can now assemble parts with 0.001% clearance and can accurately detect collision responses with 0.0001mm accuracy. Case studies are presented which can be used to identify the suitable combination of methods capable of best simulating intricate interactions and environment behavior during manual assembly.

Keywords: Virtual Reality, Virtual Prototyping, Human Computer Interaction, Virtual Assembly, Constraint-Based Modeling, Physics-Based Modeling.

1.0 INTRODUCTION

Assembly processes constitute a majority of the cost of a product [1]. Thus it is crucial to establish a comprehensive assembly planning process which anticipates actual assembly situations including assembly sequences, ergonomics and

operator safety. A well designed assembly process can improve efficiency and quality; reduce cost and a product's time to market. Computer aided assembly planning focuses on developing algorithms to automatically generate assembly sequences. Challenges in formalizing the extensive amount of expert knowledge involved limit the effectiveness of such algorithms. Commercial CAD programs on the other hand generate geometric constraint relationships among models to develop assembly simulations. Once created, these assembly sequences can be recorded and visualized as 3D simulations.

However, neither of these approaches account for the effect of human interaction involved in the assembly process. For example, they do not allow direct manipulation of 3D objects and do not take into account human factors. The result is that problems with the assembly process are found late in the product design process, on the assembly line, when the first physical prototype is built.

Virtual reality technology offers a solution to this problem by providing a three dimensional immersive environment where users can interact using natural human motions. Virtual reality technology produces human computer interaction through multiple senses, such as visual, haptic, and auditory, to create a sense of presence in the computer generated world. Developing virtual reality simulations for manual assembly is difficult due to the need to simulate constant and subtle human interactions that are involved. Other challenges include

handling large and complex CAD data sets and real time simulation constraints.

Virtual assembly in this paper is defined as assembling virtual representations for physical models through simulating realistic environment behavior and part interaction thus reducing the need of physical assembly prototyping by providing the ability to make more encompassing design/assembly decisions in an immersive computer generated environment.

The goal of this paper is to develop and identify methods to perform accurate simulation of manual assembly tasks in a virtual environment. Specific attention is paid to modeling realistic part behavior and complex human interactions.

2.0 CHALLENGES AND RELATED WORK

Mechanical Assembly: Human in the Loop

In this section we will analyze interactions involved in a simple assembly task of inserting a pin into a hole. The pin diameter is 2.5mm and the hole diameter is 2.6mm. The task can be divided into three separate steps (Fig. 1). These steps are described here to highlight the challenges involved in developing an interactive simulation to emulate this process.

Step 1: Approach the worktable on which the two parts are placed and grasp the pin.

Step2: Manipulate the pin and align it roughly with the hole.

Step3: When aligned, push the pin into the hole to complete assembly.

Simulating simple assembly tasks such as the one described above in virtual environments present several complications. Analyzing the above steps in detail, it is evident that to accomplish the first step, the system should provide the ability to the user to interactively select any part present in the environment. Collision detection is frequently used to select parts in a virtual scene. A virtual hand model is constructed to place the user's hand into the computer generated environment. Position trackers are used to coordinate the movement of the virtual hand model with the user's hand. Collisions are detected between the virtual hand model representation and other complex part models present in the environment. Once the part colliding with the hand model is identified, the user presses a button or makes a gesture to grab the colliding part which is then attached to the virtual hand model. High collision detection accuracy is not critical to this step.

After the user grabs the part, the second step is to simulate realistic part manipulation in the virtual environment. This requires modeling complex hand-part interactions which will allow the user to be able to rotate and translate the virtual part similar to the real world. Different grasping techniques are explored by researchers to allow for dexterous manipulation of virtual parts [2, 3]. One important consideration in modeling realistic manipulation of parts is that the user should be able to rotate the part based on the grab location. For example, when holding a long shaft, the user should be able to rotate it about its center of mass when it is grabbed at the center, and about the end when it is grabbed at the end.

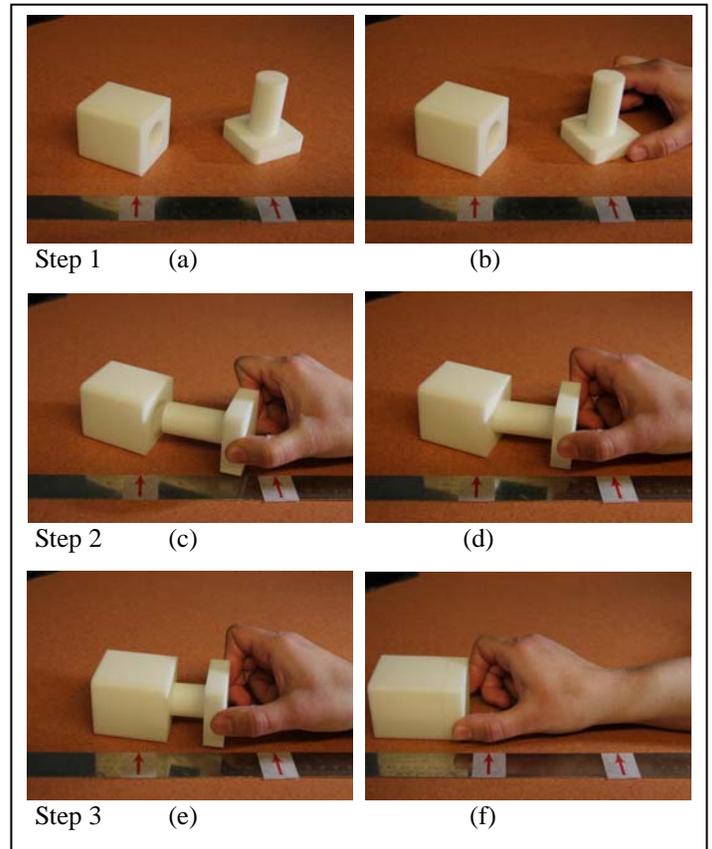


Figure 1: Assembly sequence of pin and hole

During the third step, when the user is inserting the pin into the hole his/her hands feel friction and the collision force exerted by the parts. Consider the hole part to be freely resting on the table and the pin roughly aligned with the hole. When trying to assemble, the pin will go into the hole until their cylindrical surfaces collide with each other (Fig 1d). In the presence of sufficient friction, the freely resting hole part will then move by the force exerted by the user's hand and align itself to facilitate assembly (Fig. 1e). It is evident from the ruler markings (Fig. 1f) that once the pin part is completely inserted into the hole, the user can push the entire assembly. If instead, the hole part is held in a fixture, once the cylindrical surfaces collide and the user pushes the pin, the hole surface will exert an appropriate reaction force on the pin part which can be felt by the user which helps him/her to align the pin properly to facilitate assembly. Another way of performing this assembly task is using two hands as described in [3]. In these scenarios the user is not able to see the collisions occurring inside the hole part and thus relies solely on haptic feedback to complete the assembly task.

Simple assembly tasks like inserting a pin into a hole consist of complex interactions which require depth perception for grabbing and proper alignment, precise part manipulation, haptic perception, and realistic part behavior. Simulating such behavior requires the system to be capable of detecting collisions between the pin and the hole surfaces with very high

accuracy. Once collisions are detected physical responses need to be modeled to reproduce realistic behavior of the rigid bodies. These responses then need to be passed to the user through haptic devices to allow the user to feel the physical (collision and tactile) response from virtual parts.

2.1 BACKGROUND

Initial attempts for virtual assembly simulations used part snapping both for selecting parts and to place them in the assemblies. Several virtual assembly applications relied on snapping parts to predetermined positions using pre-defined transformation matrices.

Kuehne and Oliver [4] developed IVY (Inventor Virtual Assembly) system with the purpose of being used by designers interactively during the design process to verify and evaluate the assembly characteristics of components directly from a CAD package. Once the assembly was completed, the application rendered a final animation of assembly steps. Parts were selected using assembly hierarchy as collision detection was not supported by the system.

Pere et al. [5] used "World Toolkit" to develop a PC-based system for virtual assembly called Vshop. The system used bounding box collision detection for object selection and to avoid object interpenetration. Gesture recognition was used for various tasks like switching on and off navigation and selecting parts in the environment.

Ye et al. [6] developed a virtual assembly system to investigate the potential benefits of VR in assembly planning. A non-immersive desktop VR environment and an immersive CAVE (Computer Aided Virtual Environment) [7, 8] environment were evaluated. The experiment compared assembly operations in a traditional engineering environment and immersive and non-immersive VR environments. The results concluded that the subjects performed better in VEs than in traditional engineering environments in tasks related to assembly planning.

Dewar et al. [9-11] developed a virtual assembly system at Heriot-Watt University which focused on generating assembly sequences and methods of joining components together. A head mounted display (HMD) was used for immersive visualization and a 3D mouse was used for interaction. The system relied on predefined final part positions to complete assembly tasks. Two methods - collision snapping and proximity snapping were developed for joining parts in the virtual environment.

A virtual assembly system using a three layer (scene graph layer, scripting layer and application layer) framework for abstraction was developed at BMW [12]. The system used Cyber Touch glove device for gesture recognition (for holding parts) and for providing tactile force feedback. The system used proximity detection to trigger part snapping for assembly. The interaction with the VE was assisted by voice input. Results from the user study indicated that use of VR for virtual prototyping will play an important role in the near future.

Researchers have attempted to model physical behavior of parts in virtual environments to facilitate realistic interaction and environment response for assembly tasks. Once collisions

were detected, these applications used physics-based algorithms for simulating environment responses. VEDA (Virtual Environment for Design for Assembly) a desktop VE developed by Gupta et al. [13, 14] used physics-based modeling for assembly. The application used two PHANTOM[®] haptic devices from Sensable Technologies [15] for interacting with virtual models. Being one of the initial attempts at using physics-based modeling for assembly, VEDA's capabilities were limited to handling 2D models for assembly.

Coutee et al. [16, 17] used similar desktop based dual PHANTOM[®] system setup for developing virtual assembly application called HIDRA (Haptic Integrated Dis/Re-assembly Analysis). HIDRA expanded the capabilities of VEDA by simulating collision and physics interactions among 3D objects. Because HIDRA treated 'fingertip' as a point rather than a surface, it lacked in providing realistic interaction and created difficulties when manipulating complicated geometries. Also, the application had limitations when handling non-convex CAD geometry and thus was only suitable for simulating assembly operations among simple models.

Fröhlich et al. [2] developed an interactive virtual assembly system using CORIOLIS[™] [18] physics-based simulation package. The system used the Responsive Workbench [19] for simulating bench assembly scenarios. Various spring configurations were developed for simulating realistic interaction with virtual objects. The system encountered problems in providing interactive update rates when several hundred collisions occurred simultaneously. To avoid numerical instabilities that arose while assembling low clearance models, at least five percent clearance was necessary.

Kim et al. [20, 21] investigated several collision detection and physics-modeling software applications and found VPS [22] (Voxmap Point Shell) software from The Boeing Company to be most appropriate for assembly operations. The application expanded the capabilities of VEGAS [23] by implementing physics-based modeling for simulating realistic part behavior. Networked capabilities were later added to the application to facilitate collaborative assembly through the internet [24]. Although realistic part behavior was simulated, the volume based approach of VPS, used coarse model representations to maintain interactive update rates of the simulation and thus did not allow low clearance parts to be assembled.

The above literature review shows that earlier applications aimed at modeling physical behavior were limited to 2D model representations. Later applications successfully integrated point-surface collision detection however the complex tri-mesh to tri-mesh collisions and physics responses are still challenging to perform. Large CAD assemblies consisting of hundreds of thousands of triangles present challenges in successfully and accurately modeling collision and physics responses. While simulating assembly tasks like pin and hole assembly, several hundreds/thousands of collisions occur simultaneously among the colliding parts resulting in numerical instabilities in the system and making simulations non-interactive [2]. Another approach involves developing

volumetric representations[22] of CAD models from tri-mesh data for faster collision and physics results by sacrificing accuracy. Although these approaches are successful in simulating physical behavior for suitably complex scenes interactively, the coarse model representations used for collision and physics computations do not allow CAD parts to be assembled with actual clearances [2, 3]. Thus performing collision and physics computations among complex models with tight clearances interactively is still a major challenge.

Another approach for virtual assembly simulations attempted previously by researchers helps bypass complications involved in physics-based modeling. This approach relies on utilizing inter-part geometric constraints (predefined and imported from a CAD system or defined on-the-fly) for performing assembly. Once the constraints are defined and applied among the parts, the geometric constraint solver calculates the new (generally fewer) degrees-of-freedom available to the object thus simplifying assembly.

VADE (Virtual Assembly Design Environment) developed by Jayaram et al. [25-29] used Pro/Toolkit to import assembly data (transformation matrices, geometric constraints, assembly hierarchy etc.) to simulate assembly operations in a virtual environment. Predefined geometric constraints imported from the CAD system were activated when related parts were in proximity to simulate constrained motion. Parts were then snapped to their final position to complete the assembly task. Stereo vision was provided in VADE using HMD or an Immersadesk [30] system. A physics-based algorithm with limited capabilities was added to VADE for simulating realistic part behavior [31]. Ergonomic software was later integrated into VADE to perform ergonomic evaluation for assembly tasks [32, 33].

A geometric constraint manager system was developed by Marcelino et al. [34] at University of Salford, for simulating interactive assembly/disassembly tasks in VEs. The system supported multi-platform operation, multiple constraint recognition and automatic constraint management. The constraint manager was capable of handling simple planar and cylindrical surfaces for defining and validating constraints, determining broken constraints and solving constrained motion in a system. The D-Cubed constraint engine was later used by the constraint library to perform assembly and maintenance operations using complex CAD models [35, 36].

MIVAS (A Multi-Modal Immersive Virtual Assembly System) a CAVE-based system for virtual assembly system was developed at Zhejiang University by Wan [37]. Similar to VADE, MIVAS used Pro/Toolkit for importing CAD geometry and predefined geometric constraints from Pro/Engineer CAD software. The application performed hand to part collision detection using VPS [22] software, while part to part collision detection was implemented using RAPID [38].

Liu et al. [39] used constraint-based modeling for assembly and tolerance analysis. The “assembly ports” concept imports information about the mating part surfaces; for example geometric and tolerance information, assembly direction, and type of port (hole, pin, key etc.) from different CAD systems

for assembly. The system used assembly port information for analyzing if new designs can be re-assembled successfully once parts were modified. Different criteria (proximity, orientation, port type and parameter matching) were used for applying constraints among parts. Gesture recognition was implemented using a CyberGlove device.

Chen et al. [40] developed VECA (Virtual Environment for Collaborative Assembly) which allowed collaborative assembly tasks to be performed by engineers at geographically dispersed locations. Similar to VADE and MIVAS, VECA also used Pro/Toolkit for extracting geometry (Multigen OpenFlight) and constraint data from Pro/Engineer CAD software.

Most virtual assembly applications using constraint-based methods rely on importing pre-defined geometric constraints for assembly. Instead of freezing all degrees-of-freedom of the part as implemented by snapping methods, this approach reduces the degrees-of-freedom of parts depending on the geometric constraints among them. By reducing degrees-of-freedom of parts, constraint-based methods proved useful in achieving precise part motion in virtual environments that is not achievable when unconstrained parts are manipulated with current VR input hardware. However, for every assembly scenario, specific metadata requirements (transformation matrices, geometric constraints, material properties, assembly hierarchy, etc.) resulted in time consuming and cumbersome model preprocessing requirements whenever a new assembly scenario was imported into the virtual environment. As most of these applications relied on Pro/Toolkit for generating data required for assembly simulation, these systems did not allow possibilities for importing assembly scenarios modeled in other CAD systems. In addition, most applications imported geometric-constraints from CAD systems and did not allow changing constraint relationships within the virtual environment.

Thus we see that different approaches (part snapping, physical constraint modeling and geometric-constraint modeling) have been utilized for facilitating assembly. However, none of the approaches has been proven to be successful in simulating all aspects of the complex interactions that occur during a manual assembly task. The motivation of this research is to come up with a solution which can simulate complex interaction details that are involved, and provide appropriate feedback to the user in performing manual assembly tasks in a virtual environment. The idea is to bring virtual assembly simulations closer to real world manual assembly experience. Thus, it is important to identify which method, or combination of methods will provide an encompassing solution to the problem.

3.0 THE SHARP VIRTUAL ASSEMBLY SYSTEM

Over the years, a significant amount of work has been done in the area of virtual assembly by researchers at the Virtual Reality Applications Center (VRAC) at Iowa State University. Several virtual assembly applications have been developed and various techniques for virtual assembly have been reported providing details about their usefulness and limitations. The

newest system, “**SHARP**”, System for **H**aptic **A**ssembly & **R**ealistic **P**rototyping [3] presented a dual handed haptic approach to virtual assembly. The SHARP took advantage of previous knowledge [16, 17, 24, 25, 37, 41] and utilized collision detection and physics-based modeling techniques for simulating realistic environment behavior and providing haptic force feedback during assembly. SHARP utilizes the VRJuggler [42] software toolkit for controlling the virtual environment. The system provides the capability of being ported to different VR system configurations including low-cost desktop configurations, Barco Baron [43], Power Wall as well as four-sided and six-sided CAVE systems. The network display module of the system allows it to communicate with multiple VR systems (such as CAVEs) at geographically dispersed locations. SHARP also supported swept volume generation and visualization. Direct data transfer from CAD to VR was implemented such that files made in any CAD system can be imported into VR using generic CAD formats with no preprocessing requirements.

In SHARP, collision detection and physics modeling were implemented using the VPS [22] software from The Boeing Company. VPS is a volumetric-based algorithm that accepts tri-mesh data from CAD systems using .stl file format and represents it using a set of cubic elements called voxels. A pointshell is created for the moving object which consists of points located at the centers of each voxel element. When two objects collide with each other, VPS calculates and returns the contact forces which are proportional to the amount of penetration of the pointshell of the moving object, into the voxmap of the static object. Utilizing VPS software, SHARP has successfully simulated realistic part behavior while handling complex industrial assembly scenarios at interactive frame rates.

VPS relies on approximated tri-mesh representations of B-Rep data from CAD models for generating voxel representations for collision and physics computations. Thus, the accuracy of a cubic voxel-based model representation is inversely proportional to the voxel size i.e. the smaller the voxel size, the greater the accuracy. However, small voxel size results in larger number of voxels for the same model increasing memory requirements exponentially. Also, a large number of voxels results in large computational loads as more point-voxel interactions occur when low clearance mating parts are assembled.

Figure 2 shows the voxel representation of pin and hole parts loaded in the VPS based version of SHARP. It is evident that the pin’s effective diameter is increased and the hole’s effective diameter is decreased as cubic voxel elements are used for generating the physical representations of the pin and hole model. When trying to assemble the pin through the hole, the system will not allow the user to assemble tight fitting parts because of the coarse representation of models used for collision detection and physics responses. Assembly tasks generally required 8-10% clearance between parts for successful completion. Although using VPS proved to be a successful solution for simulating realistic part behavior and

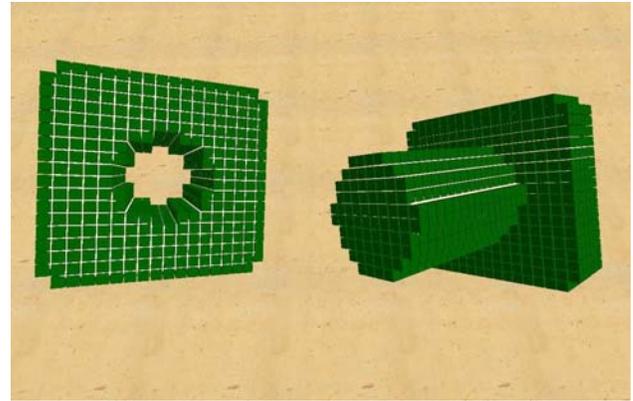


Figure 2: Pin and Hole Voxel Representations

haptic feedback, voxel-based approximation used by VPS was not accurate enough for performing low clearance assembly.

Thus the current problems with SHARP can be summarized as

- Low clearance assembly not possible because of geometry approximation
- Large memory and computation requirements
- Limited number of parts in an environment
- Collision and physics responses are insensitive to features smaller than the voxel size

3.1 New Solution to Accomplish Low Clearance Assembly

The motivation behind this research is to develop a virtual assembly application where CAD models of complex parts can be imported and assembled together in a manner closely analogous to manual assembling their physical prototypes. The user should be able to collide parts together, visualize physical constraints such as parts sliding on surfaces, and a peg sliding into a hole with a very high accuracy.

It is important to note that most of the virtual assembly applications developed previously used triangular mesh representations of complex CAD models for performing collision detection. Some methods utilized triangle information directly to perform collision queries [2, 31, 41]; while other methods generated approximate volumetric representations based on the polygonal geometry to compute collisions [3, 20, 44]. However, such representations do not provide a successful solution when low-clearance assembly operations have to be performed solely based on collision and physics responses. Low clearance assembly simulations need highly accurate collision detection among part surfaces which is not possible when approximate model representations are used.

B-Rep model representations consisting of accurate part surfaces and topology could possibly provide a solution to this problem. Figure 3 shows voxel-based, tri-mesh, and B-Rep representations of a CAD model. It is possible to get highly accurate collision and physics computation results if collision detection and physics modeling algorithms use B-Rep data models for computation. By using a new B-Rep data model for collision and physics computations SHARP can now detect collisions with an accuracy of 0.001mm.

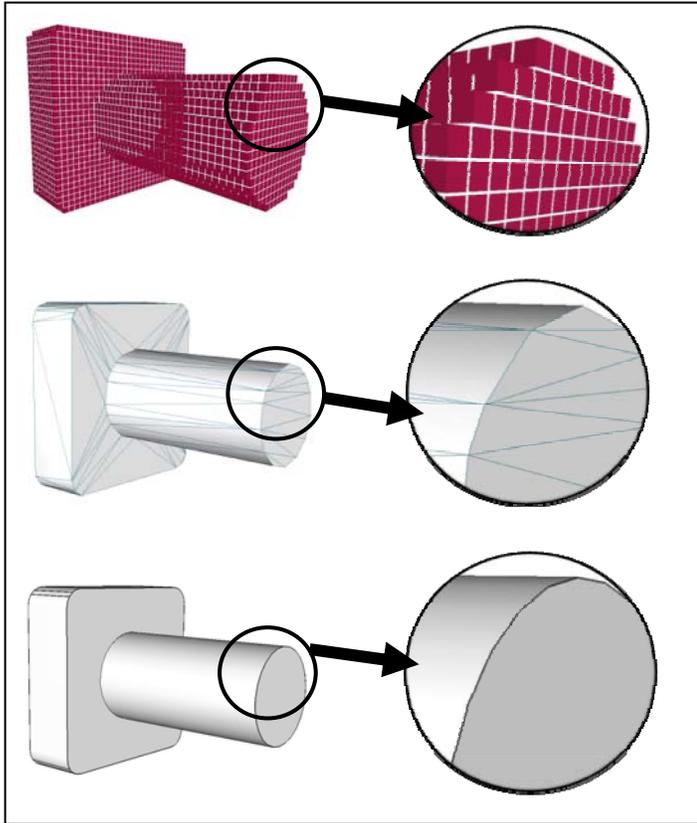


Figure 3: Voxel, Tri-mesh & B-Rep representations of a part

3.2 Runtime Physical/Geometric Constraint Solving in SHARP

Realistic environment behavior in SHARP is simulated using physically based modeling, and to prevent unnecessary collision/physics computation load for low clearance assembly, geometric constraint-based modeling is used. OpenSceneGraph, an open-source scene graph library is used for visualization. Assembly models made in any CAD system can be imported into SHARP with minimal preprocessing. SHARP requires a graphic model file and a B-Rep model file for importing a part into the virtual environment. Graphic model files are used for visualization and B-Rep model files are used by the application for performing collision/physics computations and for defining geometric constraints among models present in the environment. Thus for each model loaded in the environment, the designer has to export a graphics file and a collision model file. For graphics, *.wrl, *.iv, *.3ds, *.osg and several other generic CAD formats are accepted by the system. For collision, physics and geometric constraints, a Parasolid transmit file format (.x_t) is used. It is important to note that SHARP system operates only on CAD model files for generating geometric and physical constraints and no specific data such as assembly hierarchy, part positions, pre-defined constraints are needed for assembly.

SHARP uses the D-Cubed family of software components from UGS® for collision detection, physics and constraint

behavior simulation in the virtual environment. Three different components of the D-Cubed family are currently used by SHARP for different purposes. The Collision Detection Manager (CDM) module is used for calculating and querying collision/interference information, and the Dimensional Constraint manager (DCM) module is used for defining and solving for geometric constraints. The Assembly Engineering Manager (AEM) module is used for manipulating solid parts in the virtual environment. AEM integrates mass and inertia properties of the geometric model with a physics engine for performing realistic physical simulation.

Figure 4 shows the applications flowchart. The application first reads a configuration file which contains data about the

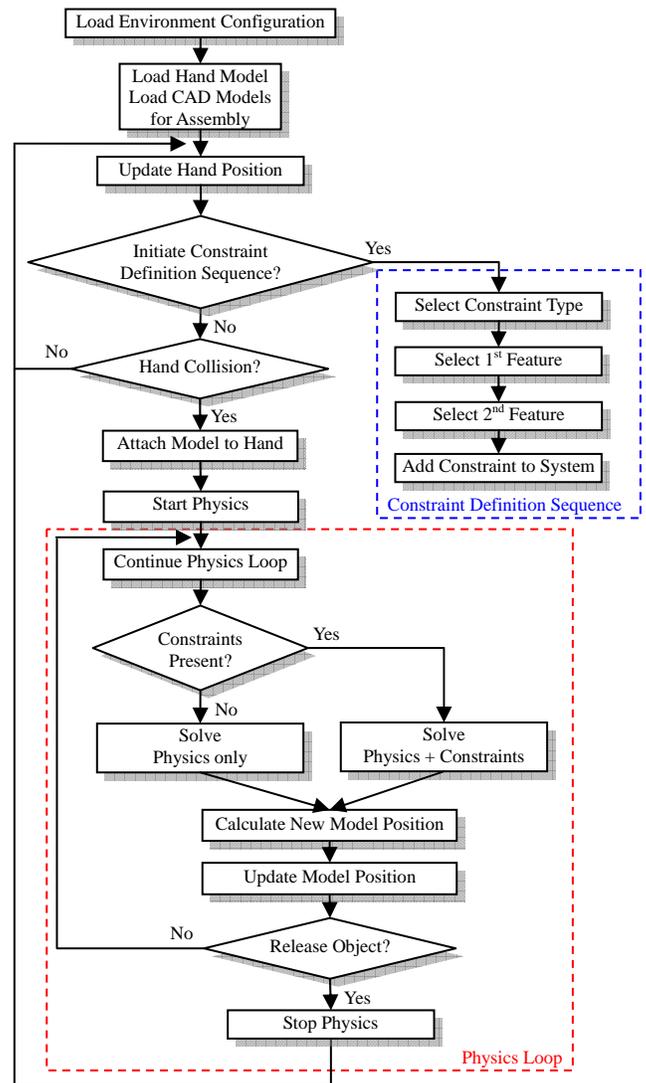


Fig.4 Application Flowchart

initial assembly environment setup such as number of parts, initial positions etc. Once B-Rep and graphic data models are loaded, the user can reach and grab models in the virtual environment and start the assembly process. The application relies on collision detection for selecting parts in the scene.

Once a part is selected by the user, an AEM based physics modeling sequence is initiated. This allows the user to manipulate the model, move it freely in space and place it in its final desired position. The system detects collisions between the models present in the scene and allows the user to guide the part into its position using simulated physical constraints. Collision detection and physics modeling allows the user to collide parts together, push other parts realistically, and visualize gravitational and interaction forces.

After trying to assemble low clearance parts using only physics modeling we realized that when clearance between parts is small, precise movement and alignment is required to complete the assembly task. Current VR hardware (trackers and 3D input devices) lack the accuracy necessary to perform precise manipulation of parts in the virtual space. In practice, the noise associated with the input signals causes unnecessary collisions among objects when trying to perform low clearance assembly tasks. To address this challenge, SHARP allows user to specify geometric constraints among part surfaces. B-Rep model data used for collision and physics computations is also utilized by the application to define constraint relationships between geometric features of different CAD parts present in the environment. A constraint definition sequence can be initiated using virtual menus or voice commands. The system uses voice-based directions to assist the user in completing the three step constraint definition sequence. Once geometric constraints are defined, the solver takes into account both physical and geometric constraints for computing part trajectories. The defined constrained can be deleted at any time by the user by voice or menu command.

This application is one of the first attempts to successfully demonstrate a combination of physics-based and constraint-based behavior for virtual assembly where both physical and geometric constraints are dynamically created and deleted at run-time. Previous attempts [25, 37, 39, 40] required geometric constraints to be predefined and imported from a CAD system before assembly could be performed. Also, these systems do not allow the user to change these geometric constraint relationships within the virtual environment.

4.0 Pin and Hole Assembly: Finding the Right Method

As discussed in the literature review section, several techniques (collision detection, physics modeling, constraint modeling) were previously used for assembling parts in a virtual environment. In the SHARP system all these capabilities are now integrated as various modules. Using menus and voice commands the users can switch On/Off different modules in SHARP. This allows SHARP to run in a reduced capacity mode i.e., using collision detection only, constraints only, or collision detection and physics modeling for assembling virtual parts. In this section we will consider these different techniques for assembling a pin into a hole as described in section 2.0. This will help identify which technique best facilitates assembly and at the same time realistically simulates complex part interactions. The virtual pin and hole models are modeled with

the same dimensions as ones used in the real world assembly demonstration and have 1mm clearance (Fig. 1).

4.1 Case I: Collision Detection Only

In this condition, only collision detection is available to assist the user in assembly. SHARP only detects collisions among models to prevent interpenetration. The user picks up the pin part and aligns the pin direction with the hole. While inserting the pin into the hole, the pin stops as soon as it collides with the hole part (Fig. 5). In this case the system does not provide any intuitive help to the user to facilitate assembly, e.g., there is no physical “self-aligning” response of the hole part to the force exerted by the mis-aligned pin. All parts are inherently stationary so the user must align the pin precisely to complete the assembly, which is extremely difficult with the precision of today’s interface hardware.

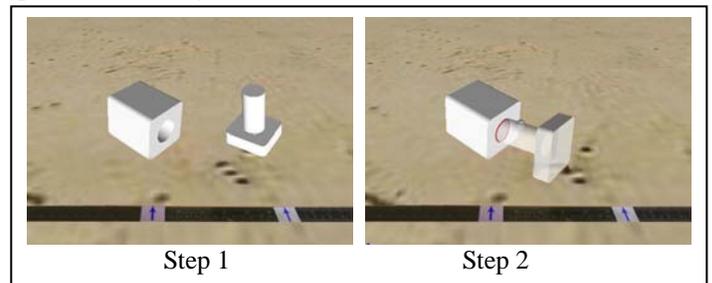


Figure 5: Assembly using collision detection only

4.2 Case II: Constraint Based Modeling

In this case constraint based modeling is used for assembling components. During the first step, the user manipulates and roughly aligns the model (Fig. 6b). Then the user starts the constraint definition sequence in which he/she selects the cylindrical surface of the hole then the cylindrical surface of the pin. Next, the user instructs the application to apply a concentric constraint between these two surfaces and the part positions are updated such that the pin and hole are properly aligned with each other (Fig. 6c). In SHARP, using the new Voice interaction module, users can define, apply and delete geometric constraints on-the-fly as well as launch other system commands. Red arrows passing through the models (Fig. 6c) depict concentric constraint acting between the models

The system reduces the degrees-of-freedom of the pin part such that it can only move in and out of the hole and rotate about its axis. Without the presence of collision detection (among the parts), the parts can interpenetrate each other making the simulation unrealistic (Fig. 6d). No physical behavior among parts (such as the pin pushing the hole model) are simulated.

4.3 Case III: Collision Detection + Physics Based Modeling

SHARP uses capabilities of the AEM module to simulate physical behavior among models present in the scene. Once collisions are detected, subsequent part trajectories are calculated by the system based on the interaction forces between models. Thus, when the user tries to insert the pin into

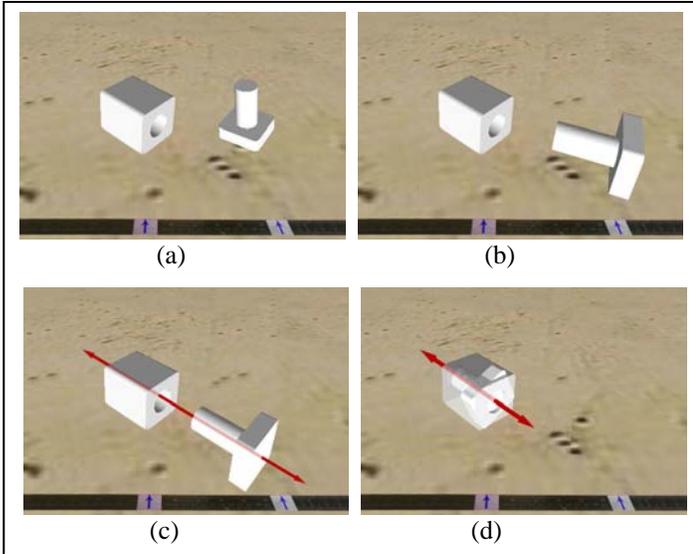


Figure 6: Assembly using Constraint Based Modeling Only

the hole, physical constraints (among the colliding surfaces) facilitate in guiding the pin. Physics based algorithms provide a realistic part behavior simulation such as pin pushing the hole part. Once the end of the pin part enters the hole, interaction forces move the hole part such that part surfaces are aligned to facilitate assembly. This behavior is similar to what we observed while performing assembly in the real environment.

In this case, however we observe that although collision and physics calculations are very accurate; the noise in the input signal (from tracker and other 3D input devices) cause vibrations in the moving pin part. These vibrations create difficulties for the user when trying to manually restrict the part motion such that it follows the insertion trajectory with the required precision. Thus, several trials were required before proper alignment was successfully achieved to complete the assembly task.

4.4 Case IV: Collision + Physics + Constraints

In this case, the user is allowed to utilize collisions, constraints and physics capabilities together to assemble parts. The user reaches and grabs the pin part (using collision detection) and aligns it roughly to the hole part (Fig. 7b, 7c). When pin and hole parts are close, the user starts a concentric constraint definition sequence (Fig. 7d). Once a constraint is defined and applied, the solver allows the user to move the pin into the hole smoothly (Fig. 7e). When fully inserted, collisions are detected between the flat face of the pin head and the hole part which collide, preventing part interpenetration. It is important to note that if the user keeps applying force on the pin part, the system will calculate the interaction forces at the colliding surfaces and would simulate realistic physical behavior (Fig. 7f). Thus, geometric constraints in this case facilitate the assembly task by ensuring proper alignment between parts while physical constraints help simulate realistic part behavior.

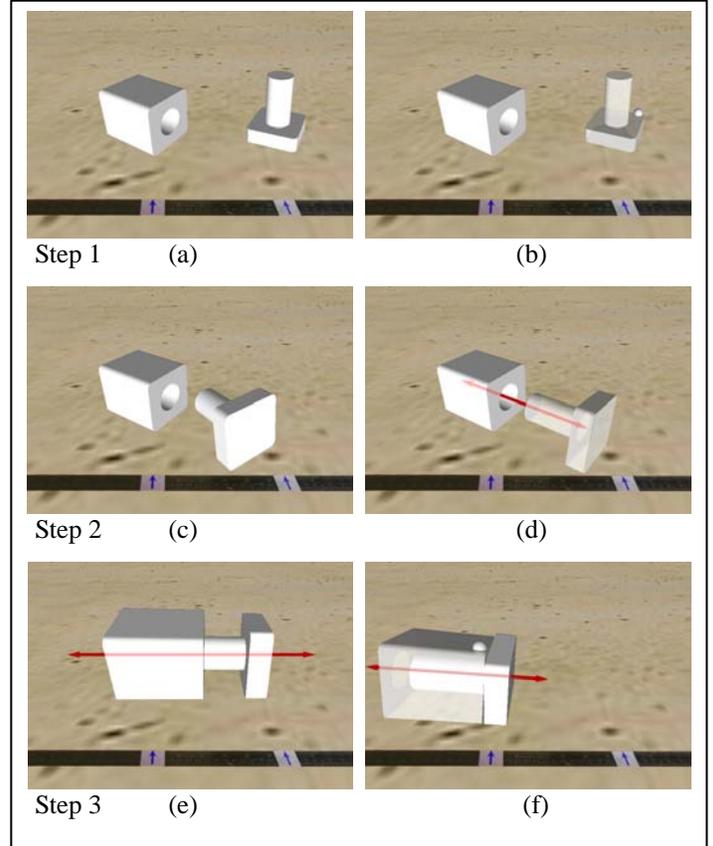


Figure 7: Collision, Physics and Constraint facilitating assembly in SHARP

4.5 Discussion

The SHARP system showed promising results for implementing realistic physical behavior into virtual assembly simulations. VPS software initially used by the SHARP system provided a robust solution for realistic simulation; however, model approximations used by VPS created problems when part clearances were small. Accuracy of collision detection is established to be a critical factor when assembling parts only on the basis of physical constraints in the environment.

Theoretically, it is possible to assemble parts using only physics based modeling if collision and physics results obtained from the virtual environment are as reliable and accurate as their real counterparts. Based on the four cases analyzed above, it has been established that even if collision and physics results are accurately determined, it is very difficult to align and move parts with the precision possible in the real world when assembling low clearance parts. Collision detection avoids model interpenetration but does not provide help from the system to facilitate assembly. Lastly, although physics based methods successfully simulate part behavior, they present high computation requirements that are difficult to perform at interactive frame rates with the required accuracy.

An assembly task has different requirements at different stages. Reaching out and grabbing only requires coarse level of collision detection. Realistic behavior modeling requires

simulations to calculate collisions between dynamic parts and calculate subsequent part trajectories based on the physics laws related to rigid-body dynamics. When assembling low clearance models, the system must provide help to the user to constrain part movements to avoid unnecessary collisions among mating surfaces which tend to slow down the simulation. Thus, none of these methods alone provides a complete solution to the virtual assembly problem. A complete solution is a combination of all of the above mentioned techniques which takes advantages of different methods during different stages of the simulation to render the best possible results.

5.0 CONCLUSIONS & FUTURE WORK

This paper presents the results of research efforts focused on providing a method of human computer interaction to facilitate evaluation of assembly sequence planning. The paper analyzes complex interactions involved while performing a simple assembly task of inserting a pin into a hole. Challenges involved in simulating such complex interactions are identified. Detailed examples are presented which illustrate the inadequacies of using either collision detection, constraint-based modeling or physics-based modeling as the only interaction method. None of the methods alone are found to be capable of simulating all aspects of the complex assembly process. It is concluded that a combination of different methods and techniques is required to realistically simulate complex interactions and facilitate assembly of complex parts in a virtual environment. The ability to combine different methods has been implemented in the SHARP software program.

The paper also outlines problems with volumetric collision detection and physics modeling while performing low-clearance assembly. A new B-Rep based collision and physics algorithm is integrated into SHARP. The system is now capable of computing highly accurate collision and physics responses among complex CAD models.

The new SHARP system demonstrates one of the first attempts in which both physical and geometric constraints are generated and deleted at runtime for performing assembly tasks in a virtual environment. Different methods (collision, physics and constraints) are successfully integrated into SHARP and can now be used independently or in combination to complete the assembly task at hand. Using only existing CAD model data, SHARP allows the user to define, apply and delete constraints at runtime. Geometric constraints are automatically taken into account by the physics algorithm when models are manipulated by the user.

Future work will include automatic geometric constraint recognition which will allow the system to automatically define the necessary constraint based on the predicted assembly intent of the user. Thus geometric constraints will be added and deleted automatically into the system resulting in more intuitive interaction with the environment by making geometric constraints transparent to the user.

6.0 ACKNOWLEDGEMENTS

We are grateful for the technical assistance of Rakesh Aggarwal of UGS. This work was funded by Deere & Company. We are also thankful to Virtual Reality Applications Center for the use of computational resources and hardware. This work was performed at Iowa State University and publication does not represent endorsement by the National Science Foundation.

7.0 REFERENCES

1. Boothroyd, G., and Dewhurst, P., *Product Design for Assembly*. 1989, New York: McGraw-Hill, Inc.
2. Fröhlich, B., Tramberend, H., Beers, A., Agarawala, M. and Baraff, D. *Physically-Based Modeling on the Responsive Workbench*. in *IEEE Virtual Reality Conference*. 2000.
3. Seth, A., Su, H. J., and Vance, J. M. *SHARP: A System for Haptic Assembly & Realistic Prototyping*. *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2006/CIE-99476)*. 2006. Philadelphia, PA, USA.
4. Kuehne, R., and Oliver, J. *A Virtual Environment for Interactive Assembly Planning and Evaluation*. *Proceedings of ASME Design Automation Conference*. 1995. Boston, MA., USA.
5. Pere, E., Langrana, N., Gomez, D., and Burdea, G. *Virtual Mechanical Assembly on a PC-Based System*. *ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC1996/DFM-1306)*. 1996. Irvine, CA.
6. Ye, N., Banerjee, P., Banerjee, A., and Dech, F., *A Comparative Study of Virtual Assembly Planning in Traditional and Virtual Environments*. *IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Review*, 1999. **29**(4): p. 546 - 555.
7. Cruz-Neira, C., Sandin, D., and DeFanti, T., *Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE*. *Proceedings of SIGGRAPH 93*, 1993: p. 135-142.
8. Cruz-Neira, C., Sandin, D.J., DeFanti, T.A., Kenyon, R., and Hart, J.C., *The CAVE, Audio Visual Experience Automatic Virtual Environment*. *Communications of the ACM*, 1992: p. 64-72.
9. Dewar, R.G., Carpenter, I.D., Ritchie, J.M., and Simmons, J.E.L. *Assembly Planning in a Virtual Environment*. *Proceedings of Portland International Conference on management of Engineering and Technology (PICMET 97)*. 1997. Portland, OR: IEEE Press.
10. Ritchie, J.M., Dewar, R.G. and Simmons, J.E.L., *The Generation and Practical use of Plans for Manual Assembly using Immersive Virtual Reality*. *Proceedings of the I MECH E Part B Journal of Engineering*, 1999. **213**(5): p. 461-474.
11. Ritchie, J.M., Simmons, J.E.L., Carpenter, I.D. and Dewar, R.G. *Using Virtual Reality for Knowledge Elicitation in a Mechanical Assembly Planning Environment*. *Proceedings*

- of 12th Conference of the Irish Manufacturing Committee. 1995.
12. Gomes de sa, A. and Zachmann, G., *Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes*. Computers and Graphics, 1999. **23**: p. 189-403.
 13. Gupta, R., and Zeltzer, D., *Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments*. Proceedings of ASME Computers in Engineering Conference and the Engineering Database Symposium. 1995. Boston, MA.
 14. Gupta, R., Whitney, D., and Zeltzer, D., *Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments*. Computer Aided Design (Special issue on VR in CAD), 1997. **29**(8): p. 585-597.
 15. *Sensible Technologies* (<http://www.sensible.com/>).
 16. Coutee, A.S., McDermott, S. D., and Bras, B., *A Haptic Assembly and Disassembly Simulation Environment and Associated Computational Load Optimization Techniques*. ASME Transactions - Journal of Computing & Information Science in Engineering, 2001. **1**(2): p. 113-122.
 17. Coutee, A.S., and Bras, B., *Collision Detection for Virtual Objects in a Haptic Assembly and Disassembly Simulation Environment*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2002/CIE-34385). 2002. Montreal, Canada.
 18. Baraff, D., *Interactive Simulation of Solid Rigid Bodies*. Computer Graphics and Applications, 1995. **15**(3): p. 63-75.
 19. Krüger, W., and Fröhlich, B., *The Responsive Workbench*. Computer Graphics and Applications, 1994. **14**(3): p. 12-15.
 20. Kim, C.E., and Vance, J.M., *Using Vps (Voxmap Pointshell) As The Basis For Interaction in a Virtual Assembly Environment*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2003/CIE-48297). 2003. Chicago, IL.: ASME.
 21. Kim, C.E., and Vance, J.M., *Collision Detection and Part Interaction Modeling to Facilitate Immersive Virtual Assembly Methods*. ASME Journal of Computing and Information Sciences in Engineering, 2004. **4**(1): p. 83-90.
 22. McNeely, W.A., Puterbaugh, K. D. and Troy, J. J., *Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling*. SIGGRAPH 99 Conference Proceedings, Annual Conference Series. 1999. Los Angeles, CA.
 23. Johnson, T.C., and Vance, J. M., *The Use of the Voxmap Pointshell Method of Collision Detection in Virtual Assembly Methods Planning*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/DAC-21137). 2001. Pittsburgh, PA.
 24. Kim, C.E., and Vance, J.M., *Development of a Networked Haptic Environment in VR to Facilitate Collaborative Design Using Voxmap Pointshell (VPS) Software*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2004/CIE-57648). 2004. Salt Lake City, UT.
 25. Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K. and Hart, P., *VADE: A Virtual Assembly Design Environment*. Computer Graphics and Applications, 1999. **19**(6): p. 44-50.
 26. Jayaram, S., Connacher, H. I., and Lyons K.W., *Virtual Assembly using Virtual Reality Techniques*. Computer Aided Design, 1997. **29**(8): p. 575-584.
 27. Jayaram, U., Tirumali, H. and Jayaram, S. A *Tool/Part/Human Interaction Model for Assembly in Virtual Environments*. ASME Design Engineering Technical Conferences 2000 (DETC 2000/CIE-14584). 2000. Baltimore, MD.
 28. Taylor, F., Jayaram, S. and Jayaram, U., *Functionality to Facilitate Assembly of Heavy Machines in a Virtual Environment*. ASME Design Engineering Technical Conferences (DETC 2000/CIE-14590). 2000. Baltimore, MD.
 29. Jayaram, S., Jayaram, U., Wang, Y., and Lyons, K., *CORBA-based Collaboration in a Virtual Assembly Design Environment*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2000/CIE-14585). 2000. Baltimore, MD.
 30. Czernuszenko, M., Pape, D., Sandin, D., DeFanti, T., Dawe, G. L., and Brown, M. D., *ImmersaDesk and Infinity Wall Projection-Based Virtual Reality Displays*. Computer Graphics, 1997. **31**(2): p. 46-49.
 31. Wang, Y., Jayaram, S., Jayaram, U., and Lyons, K., *Physically Based Modeling in Virtual Assembly*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/CIE-21259). 2001. Pittsburgh, PA.
 32. Shaikh, I., Jayaram, U., Jayaram, S., and Palmer, C., *Participatory Ergonomics Using VR Integrated with Analysis Tools*. 2004 Winter Simulation Conference. 2004. Washington D.C.
 33. Jayaram, U., Jayaram, S., Shaikh, I., Kim, Y., and Palmer, C., *Introducing Quantitative Analysis Methods into Virtual Environments for Real-Time and Continuous Ergonomic Evaluations*. Computers in Industry, 2006. **57**(3): p. 283-296.
 34. Marcelino, L., Murray, N., and Fernando, T., *A Constraint Manager to Support Virtual Maintainability*. Computers & Graphics, 2003. **27**(1): p. 19 - 26.
 35. Murray, N., and Fernando, T., *An Immersive Assembly and Maintenance Simulation Environment*. Eighth IEEE International Symposium on Distributed Simulation and Real-Time Applications. 2004.
 36. Murray, N., and Fernando, T., *Development of a an Immersive Assembly and Maintenance Simulation Environment*. International Conference on Advanced Research in Virtual and Rapid Prototyping. 2003. Leiria, Portugal.

37. Wan, H., Gao, S., Peng, Q., Dai, G and Zhang, F. *MIVAS: A Multi-Modal Immersive Virtual Assembly System*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2004/CIE-57660). 2004. Salt Lake City, UT.
38. Gottschalk, S., Lin, M. C. and Manocha, D. *OBBTree: A Hierarchical Structure for Rapid Interference Detection*. 23rd Annual Conference on Computer Graphics and Interactive Techniques. 1996.
39. Liu, Z., and Tan, J. *Virtual Assembly and Tolerance Analysis for Collaborative Design*. in 9th International Conference on Computer Supported Cooperative Work in Design 2005. 2005. Coventry, United Kingdom.
40. Chen, X., Xu, N., and Li, Y. *A Virtual Environment for Collaborative Assembly*. Second International Conference on Embedded Software and Systems (ICES'05). 2005.
41. McDermott, S.D., and, Bras, B. *Development of a Haptically Enabled Dis/Re-Assembly Simulation Environment*. ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC1999/CIE-9035). 1999. Las Vegas, NV.
42. Just, C., A. Bierbaum, A. Baker, and, C. Cruz-Neira, *VR Juggler: A Framework for Virtual Reality Development*, 2nd Immersive Projection Technology Workshop (IPT98) CD-ROM. 1998: Ames, IA.
43. *Barco Baron*
(<http://www.barco.com/entertainment/en/products/product.asp?element=1192>).
44. Seth, A., Su, H. J., and Vance, J. M. *A Desktop Networked Haptic VR Interface for Mechanical Assembly*. ASME International Mechanical Engineering Congress & Exposition (IMECE2005-81873). 2005. Orlando, FL, USA.