Iowa State University
Digital Repository

9-2006

SHARP: A System for Haptic Assembly and Realistic Prototyping

Abhishek Seth
Iowa State University

Hai-Jun Su
Iowa State University

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

Recommended Citation

SHARP: A System for Haptic Assembly and Realistic Prototyping

Abstract

Virtual Reality (VR) technology holds promise as a virtual prototyping tool for mechanical assembly; however, several developmental challenges still need to be addressed before virtual prototyping applications can successfully be integrated into the product realization process. This paper describes the development of SHARP (System for Haptic Assembly & Realistic Prototyping), a portable VR interface for virtual assembly. SHARP uses physically-based modeling for simulating realistic part-to-part and hand-to-part interactions in virtual environments. A dual handed haptic interface for realistic part interaction using the PHANToM® haptic devices is presented. The capability of creating subassemblies enhances the application's ability to handle a wide variety of assembly scenarios. Swept volumes are implemented for addressing maintainability issues and a network module is added for communicating with different VR systems at dispersed geographic locations. Support for various types of VR systems allows an easy integration of SHARP into the product realization process resulting in faster product development, faster identification of assembly and design issues and a more efficient and less costly product design process.

Keywords

VRAC, Manufacturing, Haptics

Disciplines

Computer-Aided Engineering and Design

Proceedings of the DETC'06
ASME International Design Engineering Technical Conferences and
Computers and Information in Engineering Conference
September 10-13, 2006 Philadelphia, PA

DETC06/CIE-99476

SHARP: A SYSTEM FOR HAPTIC ASSEMBLY & REALISTIC PROTOTYPING

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

Hai-Jun Su
Department of Mechanical Engineering
Virtual Reality Applications Center
Iowa State University
Ames, IA 50011
haijunsu@iastate.edu

Judy M. Vance
Department of Mechanical Engineering
Virtual Reality Applications Center
Iowa State University
Ames, IA 50011
jmvance@vrac.iastate.edu

ABSTRACT

Virtual Reality (VR) technology holds promise as a virtual prototyping tool for mechanical assembly; however, several developmental challenges still need to be addressed before virtual prototyping applications can successfully be integrated into the product realization process. This paper describes the development of SHARP (System for Haptic Assembly & Realistic Prototyping), a portable VR interface for virtual assembly. SHARP uses physically-based modeling for simulating realistic part-to-part and hand-to-part interactions in virtual environments. A dual handed haptic interface for realistic part interaction using the PHANToM® haptic devices is presented. The capability of creating subassemblies enhances the application's ability to handle a wide variety of assembly scenarios. Swept volumes are implemented for addressing maintainability issues and a network module is added for communicating with different VR systems at dispersed geographic locations. Support for various types of VR systems allows an easy integration of SHARP into the product realization process resulting in faster product development, faster identification of assembly and design issues and a more efficient and less costly product design process.

Keywords: Haptics, Virtual Reality, Virtual Prototyping, Human Computer Interaction, Virtual Assembly, Swept Volumes, Physically-Based Modeling.

INTRODUCTION

Virtual reality technology is gaining popularity as an engineering design tool and is increasingly used in the product realization process because of its ability to provide an immersive and intuitive environment which can be used as a digital test-bed for early prototypes.

Wang[1] defines Virtual Prototyping (VP) as "a computer simulation of a physical product that can be presented, analyzed, and tested from concerned product life-cycle aspects such as design engineering, manufacturing, service, and recycling as if on a real physical model". VP is used as a tool during the design process to evaluate design alternatives for assembly, manufacturability, maintainability etc. However, in order to use digital product models for advanced evaluations, a virtual prototype must exhibit behavior that is very similar to physical models. For instance, the digital environment should provide the same level of human/product interaction, allow for similar testing scenarios, and accurately reflect the evaluations that would have been obtained when using physical models. Sensory evaluations of a product such as visual, haptic (force feedback), and auditory are also important to accurately evaluate the performance of the product.

Virtual Prototyping techniques are used throughout the design process to simulate different components of the product realization process, i.e. design evaluation, manufacturing process evaluation, development of assembly techniques, etc. This paper focuses on the current human computer interaction

problems in the area of virtual assembly, a specific subset of virtual prototyping. Kim and Vance [2] define *virtual assembly* (VA), as the "ability to assemble CAD models of parts using a three-dimensional immersive user interface and natural human motions". In the past decade, many VA applications have been developed to help engineers identify product/process design errors early in the product development process in order to save time, effort and money. Reducing the number of physical prototypes needed to perform assembly evaluations results in substantial cost saving in the overall design process [3].

BACKGROUND

Several research groups have attempted to address the challenges of virtual assembly using existing technologies. Stereo viewing, head tracking, and instrumented glove interaction are all common components of many virtual assembly applications [2, 4-7]. Efforts have also been directed at interacting with complex CAD models. Recently, haptic interaction has been integrated into many of these applications [8-13]. Haptic interaction provides force feedback to the user as an additional sensory input to aid in evaluating the suitability of the assembly process represented in the virtual environment.

Gupta et al. [14, 15] developed a desktop based virtual assembly application called VEDA (Virtual Environment for Design for Assembly) which used physically based modeling (PBM) for modeling part behavior. Dual PHANToM® haptic devices were used for providing force feedback and auditory and stereo cues were provided to augment part interaction.

Jayaram et al. [10] developed VADE (Virtual Assembly Design Environment) at Washington State University. VADE used a CyberGrasp haptic device for interacting with virtual objects. Pro/E CAD models were directly imported and assembly was performed using constraint methods. Stereo vision was provided by a Head Mounted Display (HMD) or a Barco Baron. Using the CyberGrasp device for haptic interaction provided force feedback for grasping but not for part collisions. VADE also supported swept volume generation for addressing maintainability issues.

Johnson and Vance [16] developed VEGAS (Virtual Environment for General Assembly), in 2001. Using Voxmap Point Shell (VPS)[17] software from Boeing Corporation, users could assemble full scale models with high polygon counts. Collision detection was implemented; however, the program lacked any kind of part behavior simulation and haptic interaction.

Kim and Vance [2, 4] investigated several collision detection and part behavior algorithms and further modified VEGAS to include physically based modeling to simulate part behavior in virtual environments. Though the application could handle large model data for collision detection and part behaviors, it did not support haptic interaction.

Kim and Vance [12] also developed NHE (Networked Haptic Environment) where users from geographically dispersed locations could share the same assembly environment. Interaction was provided using PHANToM®

haptic devices which can be used to grab and manipulate virtual objects. Realistic part behavior was simulated using the Voxmap Point Shell (VPS) [17] library from Boeing Corporation. Immersion was provided using a multi-pipe projection screen VR system. However, the need of a dedicated PC for force rendering at each network-node made the system expensive and provided no possibility for dual handed haptic interaction.

Coutee and Bras [8, 9, 13] developed HIDRA (Haptically Enabled Dis/Re-Assembly Simulation Environment) which used a dual PHANToM® configuration for haptic interaction. The application lacked in providing stereo visual feedback and did not support physical modeling of complex CAD geometry. HIDRA used virtual finger tip interaction to hold and manipulate virtual objects.

A virtual assembly system was developed at BMW for performing assembly simulations using virtual prototypes [18]. The system used a three layer framework which provided abstraction. A Cyber Touch glove device was used for gesture recognition and tactile force feedback. Voice commands and gestures were used for interacting with the virtual environment. The user study found that grasping interaction alone was insufficient and concluded that force feedback was crucial for performing virtual assembly tasks.

Wan et al. [11] developed a multimodal CAVE-based virtual assembly system called MIVAS (A Multi-Modal Immersive Virtual Assembly System) at Zhejiang University. Immersion was provided by a four wall projection screen system and assembly was performed using constraint methods. Hand-part collision detection was implemented using VPS [17] software while part-to-part collision detection was implemented using RAPID. Haptic feedback was provided using the CyberGrasp haptic device. Like VADE, MIVAS could only simulate grasping and not part collisions.

Ye et al. [19] developed a virtual assembly system to identify potential benefits of virtual reality in assembly planning. The experiment compared assembly performance in traditional, non-immersive, and immersive virtual environments. The three conditions differed in ways in which assembly was presented and handled. The paper concluded that subjects performed better in virtual environments than in traditional engineering environments in tasks related to assembly planning.

Jun et al. [20] at Beijing Institute of Technology proposed a hierarchical assembly task list (HATL) model where different assembly tasks are organized into a hierarchical list for Virtual Assembly Process Planning (VAPP). The desktop version of the system was developed using WTK (WorldToolkit9.0) and was capable of automatic constraint recognition and collision detection. Although part behavior was implemented using constraints, the application did not provide haptic feedback and an immersive assembly environment.

The goal of the work presented here is to advance the state-of-the-art in virtual assembly by developing an application capable of providing dual handed force feedback

and realistic simulation of part behavior among complex CAD models while performing assembly tasks in virtual environments.

SHARP: A SYSTEM FOR HAPTIC ASSEMBLY & REALISTIC PROTOTYPING

Over the years, researchers at the Virtual Reality Applications Center (VRAC) at Iowa State University have investigated various virtual assembly techniques and reported on their usefulness and limitations. The newest system, SHARP, **S**ystem for **H**aptic **A**ssembly & **R**ealistic **P**rototyping, takes advantage of previous knowledge [8-13] and expands the functionality of virtual assembly to include dual handed haptics, swept volume representation, subassembly modeling and more realistic part behavior through the use of physically based modeling.

SHARP has been tested on Windows, Linux and Irix platforms and supports different types of VR systems (4 and 6 sided multi-screen projection systems, Barco Baron, HMD and desktop stereo environments) and a variety of haptic feedback devices from Sensable Technologies (PHANToM® 3.0 Premium, PHANToM® 1.5, PHANToM® Desktop and PHANToM® Omni). These devices all provide six degree-of-freedom motion but only three degree-of-freedom force feedback. Figure 1 shows a user sitting in front of a Barco Baron and manipulating the two PHANToM® Omni haptic devices. Table 1 lists the different software libraries used in developing SHARP.
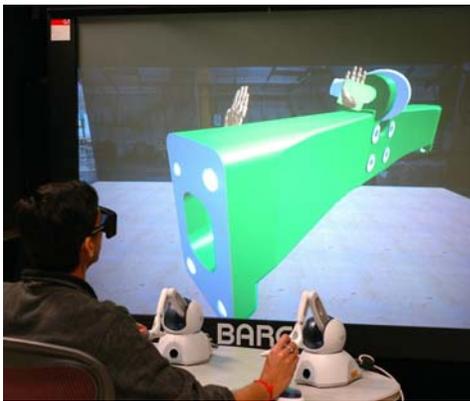

Figure1: SHARP being used with Barco Baron and dual PHANToMs

Table 1: Software libraries used in SHARP

Purpose	Software Library
Virtual Reality Infrastructure	VR-Juggler
Visualization Toolkit	OpenGL Performer
Haptic Device Control	Open Haptics Toolkit
Network Capability	TCP/IP
Collision Detection & PBM	VPS

Graphical Visualization

The SHARP application infrastructure is based on VR Juggler, an open source software toolkit developed at ISU. VR Juggler provides a platform for VR applications enabling them to run on different VR systems (HMD, 4 & 6 sided CAVE, Barco Baron and Desktop). Reconfiguring the application for different systems is performed easily by changing a configuration file. The VR Juggler Portable Runtime library provides an operating system abstraction layer that simplifies the process of creating cross-platform software. In this application, the graphical rendering is performed using SGI OpenGL-Performer scene graph library.

Realistic Object Behavior

When developing a virtual environment which supports interactive manipulation and assembly of complex CAD objects, the greatest challenge to achieving realistic part behavior is managing the tradeoff between object complexity and computational burden. Most often, an approximate geometric model is used for collision detection and force calculations. An approximate model which is coarsely defined allows for fast, but inaccurate collision and force calculations. Similarly, an approximate model which closely approximates the real model may contain so much detail that the collision detection and force calculations cannot be performed fast enough to support interactive manipulation in the virtual environment.

Researchers have found that a haptic interface is desirable for performing assembly tasks in virtual environments [18]. In an assembly task, a haptic force can help designers feel and better understand the geometry of virtual objects. Research has shown that the addition of force feedback to virtual environments increases task efficiency times [21, 22]. Also, testing on subjects has verified that operators feel more secure and can relate better to the real world processes when trained on a simulator with haptic feedback than those trained on a simulator with no haptic feedback [23]. Since most haptic devices require a high update rate to guarantee force continuity, the real challenge is to maintain haptic update rate especially when interacting with large CAD models. In addition, generating a feedback part-to-part collision force that is natural to the operator is also non-trivial.

In SHARP, realistic object behavior modeling is implemented using the Voxmap Point Shell (VPS) software from Boeing Corporation. VPS is especially suited for virtual assembly applications for three reasons: 1) VPS can operate on CAD models of complex geometry; 2) VPS works well when there are a small number of moving objects in the virtual environment; and 3) VPS is optimized for maintaining the haptic force update rate as high as 1000Hz[24].

In SHARP, each CAD model is discretized into a set of voxels (cubic elements) creating a "voxmap" which is used for collision detection and physics computation. 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 returns the contact force which is proportional to the amount of penetration of the pointshell of the moving object into the voxmap of the static object. This force must then be translated to the haptic device.

When a user grasps a part, a virtual spring-damper system is attached between the part and the virtual hand (Fig. 2). The distance between the virtual hand and the manipulated object determine the spring force $\vec{\mathbf{F}}_{spring}$ and torque $\vec{\boldsymbol{\tau}}_{spring}(t)$ exerted on the object. Note that the spring force and torque also include the viscosity force of the damping system. The collision force $\vec{\mathbf{F}}_i$ is proportional to the amount of penetration that one object is into the other object in the environment. The manipulated object is dynamic in nature and its motion is subject to physics law, more specifically rigid body dynamics. That is, given the dynamic state of a rigid body at time t, its motion must satisfy the following equation:

$$\frac{d\vec{\mathbf{P}}(t)}{dt} = \vec{\mathbf{F}}_{total}(t), \frac{d\vec{\mathbf{L}}(t)}{dt} = \vec{\mathbf{M}}_{total}(t)$$

where $\vec{\mathbf{F}}_{total}(t) = \vec{\mathbf{F}}_{spring}(t) + \sum \vec{\mathbf{F}}_i + \vec{\mathbf{F}}_{brake}$ and $\vec{\mathbf{M}}_{total}(t) = \vec{\boldsymbol{\tau}}_{spring}(t) + \sum \vec{\mathbf{r}}_i \times \vec{\mathbf{F}}_i$ are the total external force and moment exerted on the body respectively. For our case, they are given by the sum of the force/torque applied by the virtual spring, collision force applied by other objects, damping and braking force. And $\vec{\mathbf{P}}(t), \vec{\mathbf{L}}(t)$ are linear and angular momentums of the rigid body respectively given by

$$\vec{\mathbf{P}}(t) = m\vec{\mathbf{v}}(t), \vec{\mathbf{L}}(t) = [I]\vec{\boldsymbol{\omega}}(t)$$

where $\vec{\mathbf{v}}(t), \vec{\boldsymbol{\omega}}(t)$ are the linear and angular velocity respectively, m is the total mass, and I is the inertia tensor determined by the geometry of the part. The rigid body dynamics equation is solved using the VPS function "VpsPbmEvolve". See [17] for more details concerning the VPS method.

The spring force is sent to the haptic device for rendering. Hence, what the user feels is really the spring force between the part and the hand model.
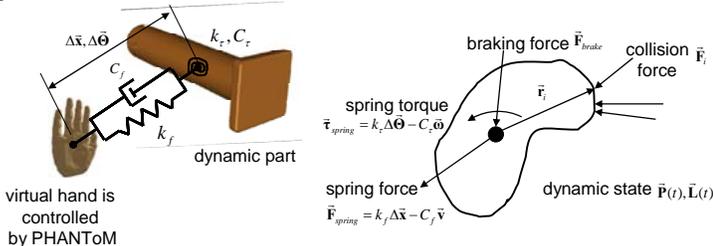

Figure 2: Physics modeling of object using VPS

Careful selection of the amount of discretization and the number of offset layers of the VPS haptic model is needed in order to produce a representation which is sufficiently modeled so that tight tolerance parts can be assembled. This enables

large CAD models can be viewed in the environment without significant computational delays. Offset layers are used in VPS to insure that penetration does not occur between colliding parts. SHARP allows for individual models in the scene to have different voxel sizes and number of surface offset layers. In addition, SHARP provides for interactive re-voxelization of models during runtime of the application. Implementation of this feature has allowed us to assemble a bolt into a hole within a complex CAD part. Future work will involve investigation of selective voxelization of a subspace within a given part which will provide even more versatility.

Dual PHANToM® Haptic Interface

Several assembly processes require two hands. A dual handed haptic interface has successfully been developed and integrated into SHARP. Open Haptics Toolkit (v.2.0) library is used for communicating with the PHANToM® haptic devices from SensAble Technologies (Fig. 3). The dual handed interface with haptic feedback provides a very efficient and intuitive interaction for virtual assembly tasks. Interacting with two hands and getting force feedback, an operator can more realistically perform assembly tasks with the same dexterity as he/she has in the real world.

Figure 3: PHANToM® Desktop, PHANToM® 1.5, PHANToM® 3.0 and PHANToM® Omni, by SensAble Technologies (Images courtesy of Novint Technologies)

An illustration of the difference between two handed and single handed manipulation will highlight the significance of this additional capability. For example, if a user wants to assemble a peg into a block using single handed haptic interaction, the user can only manipulate one part at a time. Thus, the assembly steps using a single handed haptic interface are shown in Figure 4.
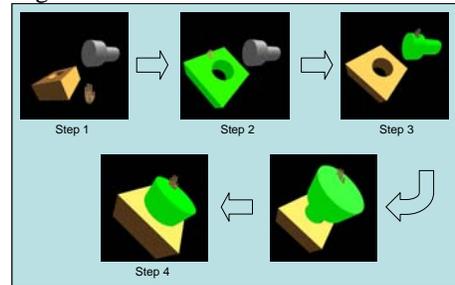

Figure 4: Assembly steps using single haptic hand CAD models were made using Pro/Engineer

Step 1: Grab the Block model – position and orient it suitably.
Step 2: Release the Block model.
Step 3: Grab the Peg and try to orient and insert it into the stationary Block model.

Step 4: Try Re-orienting the Block model if assembly is cumbersome.

Step 5: Perform Step 2 – 4 as necessary.

Using dual handed haptic interaction the user can manipulate both parts simultaneously, orient them with respect to each other and assemble them. Assembly steps using dual handed haptic interface (Fig. 5) will be as follows:

Step 1: Grab the Block model with one hand and the Peg with the other hand.

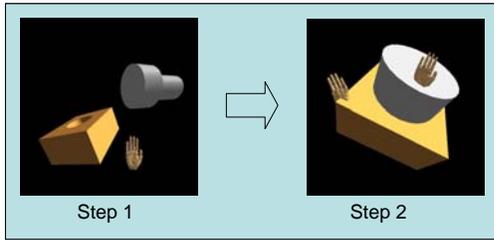Step 2: Orient them simultaneously and assemble together. See Figure 5.

Figure 5: Assembly steps using dual handed assembly

Thus we see that a dual handed interface not only reduces the number of assembly steps to almost half but also makes the assembly simulation more realistic, by closely replicating real world interactions.

SHARP loads voxelized models of the virtual hand for both hands during initialization and detects collision between the hand models and each of the voxelized CAD models present in the environment. The user can grab a CAD model by intersecting his/her hand with the desired CAD model and pressing the stylus button on the respective PHANToM® haptic device. SHARP is capable of simulating scenarios of simultaneous manipulation of parts/subassemblies grabbed in each hand and is capable of performing collision detection and physically based modeling while assembling objects. Two hands can also hold and manipulate the same object.

Swept Volume Generation

Modeling of swept volumes plays a critical role in resolving issues that may arise while servicing or inspection of complex mechanical assemblies. In SHARP, VPS is used for swept volume generation and SGI Performer for swept volume visualization. For calculating the volume swept by a model, we track and record the position and orientation of the model during a given time period, which is needed by VPS for Swept volume computations. To start monitoring the part for swept volume generation the user has to switch the swept volume button state to "SWEPT VOLUME ON". Then as the user moves the part from its initial position to the desired final position SHARP records the transformation matrices of the moving model at every frame. The "SWEPT VOLUME OFF" button stops the part monitoring process.
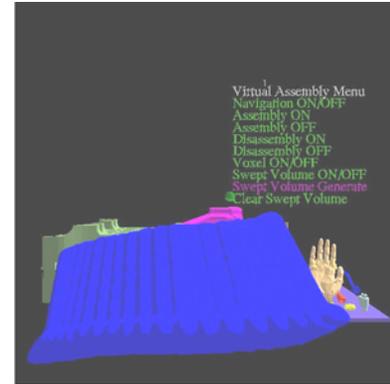

Figure 6: Illustration of the generated Swept Volume

The swept volume is formed by a Boolean union of VPS object models transformed according to each motion frame. To visualize the swept volume generated by VPS, we use a tessellation function to generate the triangulated data which is then displayed using OpenGL Performer (Fig. 6). Note that the swept volume represents the area of the voxelized models and therefore is an approximation to the model geometry.

Support for Subassemblies

Subassemblies are an integral part of a mechanical assembly process. A mechanical assembly task can be any of the following:

- Assembling two separate parts
- Assembling a part with another subassembly
- Assembling two subassemblies

Thus, in order to simulate a mechanical assembly process realistically, the ability to assemble subassemblies is important. One of the major improvements to SHARP is the ability to support interaction with subassemblies in a virtual assembly process simulation.

Performing dynamic assembly/disassembly operations in virtual environments requires modification of the underlying scene graph, or object hierarchy tree in order to maintain consistent object motions. When two or more parts are assembled together, their VPS data and display nodes need to be rearranged so that they behave as a single entity in the digital world.

For building a subassembly, the user assembles parts together and places them in their final relative positions in the subassembly. Then the user has to inform the application that these parts should be treated as a single object in the virtual environment. This requires calculating the mass, center of mass, moment of inertia and other physical properties of the subassembly for future physics computations and rearranging the visualization scene graph structure such that the graphic position of the subassembly correspond to that of the respective physics model in the virtual environment. This requires storing all properties and current states of models that are assembled together. This information is later used for restoring the individual models to their current state when the subassembly is disassembled.

Providing capabilities for building a subassembly using two or more subassemblies (instead of parts) made the problem even more complex. The data structure in SHARP is designed such that each individual part contains information about its current state, i.e. if it is a single part or a member of a subassembly, whether it is assembled to another part, or whether other parts are assembled to it.

A new thread called "Assembly Thread" is designed to accomplish the subassembly process. (Fig. 7)
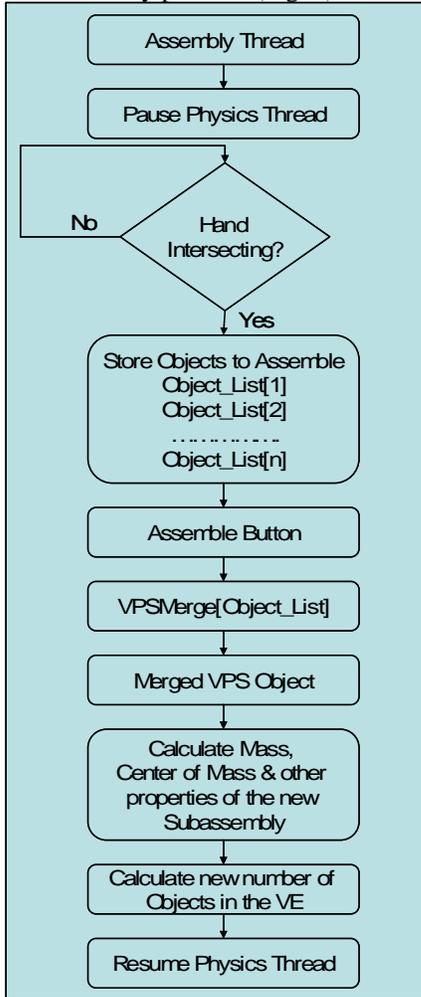

Figure 7: Operations performed by the Assembly thread

All part manipulation operations like grabbing and moving the parts in the environment are suspended. After placing the parts/assemblies together, the user selects the parts to be sub-assembled by intersecting his/her hand with the part/assembly to be sub-assembled. The "VPSMerge" function is used for returning a merged VPS object as output which will be used as a merged voxmap and/or pointshell in the virtual environment for physically based modeling and collision calculations. The OpenGL Performer scene graph structure is changed and parts to be sub-assembled are removed from the root node and attached to the part node to which they are sub-assembled. Figures 8 and 9 show the changes in data structure while

assembling parts 2 and 3 to part 1. Parts 2 and 3 are removed from the root node in the scene graph and attached to part 1 node. Also the data structure for part 1 is updated with the information that it has parts 2 and 3 assembled to it and the data structured of parts 2 and 3 are updated with information that they are now assembled to part 1. Now calculations for the new number of models (2 in this case i.e. model 1 and model 4) in the environment are done. Also, calculations for mass, center of mass, moment of inertia and other properties of the assembly are executed before the assembly thread is terminated.
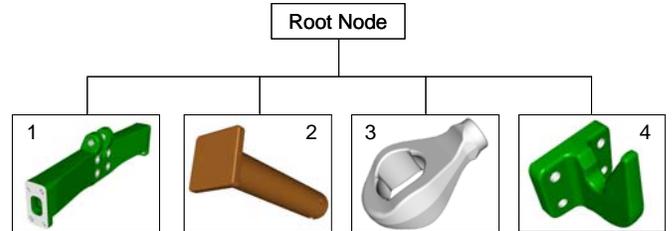

Figure 8: Data structure before assembly

This completes the subassembly process and all selected parts are now joined together and are treated as a single part for collision detection and physically based modeling in the virtual environment.

For disassembling an assembly, the users have to first press the "DISASSEMBLY ON" button and select the subassembly to be disassembled. Now pressing the "DISASSEMBLY OFF" button restores the parts in the subassembly to their respective original states.
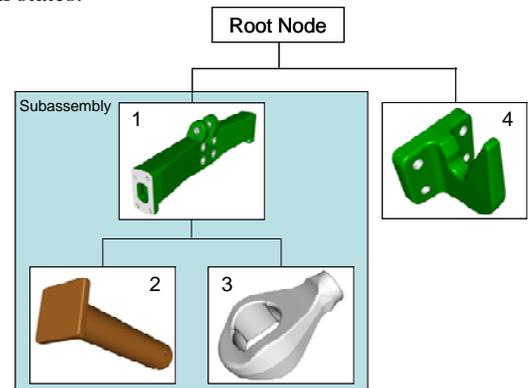

Figure 9: Data structure after assembling Part 1, 2 and 3

ASSEMBLY TASK

SHARP has been tested using several assembly scenarios of complex industrial CAD models. This section describes assembling parts of a hitch assembly from John Deere. The assembly task demonstrated here consists of five parts. CAD models of parts to be assembled are imported into the virtual environment (Fig.10) The assembly task involved inserting the hydraulic cylinder between the lower holes of the lift arm and locking it in place using the large pin part. Completion of the assembly task required inserting the upper lift link between the front holes of the lift arm locking it in place using the small pin

part. Representing CAD models in the form of voxels (Fig. 11) for collision detection and physically based modeling, low clearance assembly is not feasible. Thus, in order to assemble these parts, the two pins were scaled to 90% of their original sizes. Part statistics are shown in Table 2.

Model Name	No. of Triangles	No. of Voxels
Lift Arm	17399	106,773
Large Pin	888	5011
Upper Lift Link	6616	11214
Small Pin	682	2678
Hydraulic Cylinder	10615	47804

Table 2: Part Statistics

Figure 10: Parts to be Assembled

Voxelized representation of parts can bee seen in Fig. 11. SHARP supports different parts to have different voxel sizes.
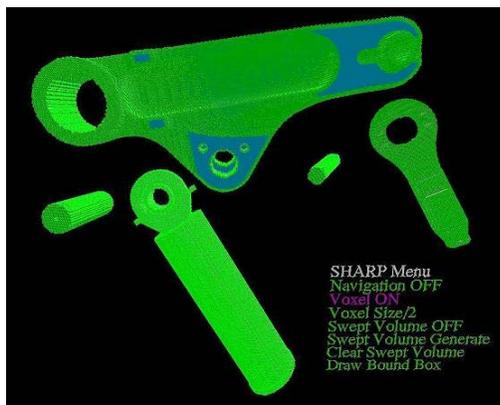

Figure 11: Voxelized View

This method saves memory and facilitates handling of large data-sets for collision detection and PBM while performing assembly. In this scenario as all parts have low clearance assembly features, the parts were voxelized using very small voxel size. The dual handed haptic interface provided simultaneous part manipulation and made the assembly task at hand easier to perform. Figure 12 shows successful assembly of CAD models.
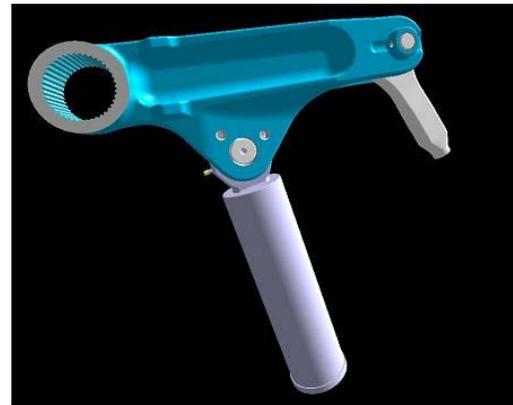

Figure 12: Assembled Parts

CONCLUSIONS & FUTURE WORK

In this paper, a platform independent application, SHARP, has been presented which uses physically based modeling for simulating realistic part behavior and provides an intuitive dual handed PHANToM® haptic interface for mechanical assembly in an immersive virtual reality environment.

SHARP is capable of assembling complex CAD geometry and supports a vast variety of VR systems for increased portability. A unique approach for assembly/disassembly operations is presented to handle more complex assembly scenarios. Swept volumes are integrated to generate information for addressing maintainability issues. SHARP also includes a record and play module for assembly sequence verification and operator training purposes and a network module to support collaborative development [21].

Although SHARP shows promising results, the virtual assembly process can be still be improved. Physically based interaction methods provide total user control over part movements and therefore seem very realistic; however, the lack of full six degree-of-freedom haptic feedback restricts the user to experiencing only three degree-of-freedom forces, i.e. no torque feedback, when objects collide. In many assembly operations, torque feedback is an important factor. Physically based modeling also depends on the underlying haptic model to generate collisions and contact forces. This haptic model represents an approximation of the surface geometry and introduces dimensional error in tight fitting assembly operations. We have addressed this issue in SHARP by providing the ability to have multiple parts with multiple degrees of voxelization and the ability to re-voxelize during run time. However, in the future we will be examining methods to have different voxel sizes on one individual part and more accurate collision detection algorithms. We will also be examining a combination of constraint-based methods, where mating parts "snap to" their correct positions when in close proximity to one another, and physically-based modeling to provide the optimum interaction paradigm for assembly prototyping.

REFERENCES

1. Wang, G.G., 2002, "Definition and Review of Virtual Prototyping," ASME Journal of Computing and Information Science in Engineering, **2**(3), pp. 232-236.
2. Kim, C.E., and Vance, J.M., 2003, "Using Vps (Voxmap Pointshell) As The Basis For Interaction in a Virtual Assembly Environment (DETC2003/CIE-48297)," *ASME Design Engineering Technical Conferences*, Chicago, IL.
3. Savall, J., Borro, D., Gil, J.J., Matey, L., 2002,"Description of a Haptic System for Virtual Maintainability in Aeronautics," *IEEE International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland.
4. Kim, C.E., and Vance, J.M., 2004, "Collision Detection and Part Interaction Modeling to Facilitate Immersive Virtual Assembly Methods," ASME Journal of Computing and Information Sciences in Engineering, **4**(1), pp. 83-90.
5. Kuehne, R., and Oliver, J., 1995,"A Virtual Environment for Interactive Assembly Planning and Evaluation," *ASME Design Engineering Technical Conferences*, Boston, MA.
6. Yuan, X., and Sun, H.,1997,"Mechanical Assembly with Data Glove Devices" *IEEE 1997 Canadian Conference on Electrical and Computer Engineering*, St. Johns, Newfoundland, Canada.
7. Jayaram, S., Connacher, H. I., and Lyons K.W., 1997, "Virtual Assembly using Virtual Reality Techniques," Computer Aided Design, **29**(8), pp. 575-584.
8. Coutee, A.S., McDermott, S.D., and Bras, B, 2001, "A Haptic Assembly and Disassembly Simulation Environment and Associated Computational Load Optimization Techniques," ASME Journal of Computing & Information Science in Engineering, **1**(2), pp. 113-122.
9. Coutee, A.S., Bras, B., 2002,"Collision Detection for Virtual Objects in a Haptic Assembly and Disassembly Simulation Environment (DETC2002/CIE-34385)," *ASME Design Engineering Technical Conference/Computers in Information Engineering*, Montreal, Canada.
10. Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K. and, Hart, P., 1999,"VADE: A Virtual Assembly Design Environment," Computer Graphics and Applications, **19**(6), pp. 44-50.
11. Wan, H., Gao, S., Peng, Q., Dai, G and Zhang, F., 2004, "MIVAS: A Multi-Modal Immersive Virtual Assembly System (DETC 2004/CIE-57660)," *ASME Design Engineering Technical Conferences*, Salt Lake City, UT.
12. Kim, C.E., and Vance, J.M., 2004, "Development of a Networked Haptic Environment in VR to Facilitate Collaborative Design Using Voxmap Pointshell (VPS) Software (DETC2004/CIE-57648)" *ASME Design Engineering Technical Conferences,* Salt Lake City, UT.
13. Scott, D.M., Bert Bras., 1999, "Development of a Haptically Enabled Dis/Re-Assembly Simulation Environment (DETC99/CIE-9035)", *ASME Design Engineering Technical Conferences*, Las Vegas, NV.
14. Gupta, R., and Zeltzer, D., 1995, "Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments," *ASME Computers in Engineering Conference and the Engineering Database Symposium*, Boston, MA.
15. Gupta, R., Whitney, D., and Zeltzer, D., 1997, "Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments," Computer Aided Design (Special issue on VR in CAD), **29**(8) pp. 585-597.
16. Johnson, T.C., and, Vance, J. M., 2001,"The Use of the Voxmap Pointshell Method of Collision Detection in Virtual Assembly Methods Planning (DETC2001/DAC-21137)," *ASME Design Engineering Technical Conferences*, Pittsburgh, PA.
17. McNeely, W.A., Puterbaugh, K. D. and, Troy, J. J., 1999, "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," *SIGGRAPH 99 Conference Proceedings, Annual Conference Series*, Los Angles, CA.
18. Gomes de sa, A. and Zachmann, G., 1999, "Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes" Computers and Graphics, **23**(3)**,** pp. 389-403.
19. Ye, N., Banerjee, P., Banerjee, A., and Dech, F., 1999, "A Comparative Study of Virtual Assembly Planning in Traditional and Virtual Environments," IEEE Transactions on Systems, Man, and Cybernetics - Part C: Applications and Review, **29**(4), pp. 546-555.
20. Jun, Y., Liu, J.,Ning, R. and Zhang,Y., 2005, "Assembly Process Modeling for Virtual Assembly Process Planning," International Journal of Computer Integrated Manufacturing, **18**(6), pp. 442-451.
21. Seth, A., Su, H.-J., and Vance, J. M., 2004, "A Desktop Networked Haptic VR Interface for Mechanical Assembly (IMECE2005-81873)," *ASME International Mechanical Engineering Congress & Exposition*, Orlando, FL.
22. Burdea, G.C., 1999, "Haptic Feedback for Virtual Reality," *Virtual Reality and Prototyping Workshop*, Laval, France.
23. Volkov, S.A., Vance, Judy M., 2001, "Effectiveness of Haptic Sensation for the Evaluation of Virtual Prototypes (DETC2001/DAC-21135)," *ASME Design Engineering Technical Conference*, Pittsburgh, PA.
24. Burdea, G.C., 1999, "Invited Review: The Synergy Between Virtual Reality and Robotics," IEEE Transactions on Robotics and Automation, **15**(3), pp. 400-410.
25. Balijepalli, A. and T. Kesavadas, 2004, "Value-addition of Haptics in Operator Training for Complex Machining Tasks" ASME Journal of Computing and Information Science in Engineering, **4**(2), pp. 91-97.