2007

Combining physical constraints with geometric constraint-based modeling for virtual assembly

Abhishek Seth
Iowa State University

Recommended Citation

Seth, Abhishek, "Combining physical constraints with geometric constraint-based modeling for virtual assembly" (2007). *Retrospective Theses and Dissertations*. 15540.
http://lib.dr.iastate.edu/rtd/15540

Combining physical constraints with geometric constraint-based modeling for virtual assembly

by

Abhishek Seth

A dissertation submitted to the graduate faculty

in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Co-Majors: Mechanical Engineering; Human-Computer Interaction

Program of Study Committee:
Judy M. Vance (Major Professor)
James H. Oliver
Eliot Winer
Thomas J. Rudolphi
Chris Harding
Sankar Jayaram

Iowa State University

Ames, Iowa

2007

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Dedicated to my parents

to whom I will always be indebted.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 2. VIRTUAL ASSEMBLY: REVIEW & FUTURE DIRECTIONS

CHAPTER 3. DEVELOPMENT OF A DUAL-HANDED HAPTIC ASSEMBLY SYSTEM: SHARP

CHAPTER 4. COMBINING PHYSICAL CONSTRAINTS WITH GEOMETRIC CONSTRAINT MODELING FOR VIRTUAL ASSEMBLY USING SHARP

CHAPTER 5. FEATURE-BASED CONSTRAINT RECOGNITION FOR VIRTUAL ASSEMBLY

CHAPTER 1. INTRODUCTION

Virtual reality (VR) technology provides a new paradigm for human-computer interaction by immersing the user into the computer generated scene and allowing him/her to interact with computers using natural human motions. During the past two decades, the advances in computer processing and graphics hardware have evolved VR to a level where its power can be harnessed beyond scientific visualization, to aid engineering analysis and facilitate interactive mathematical modeling applications. The capability of VR to provide a human-centered simulation environment makes it a perfect tool for analyzing processes, such as assembly, that involve constant human interactions that prove difficult to account for when using traditional computer simulation techniques.

Assembly analysis is generally performed using physical prototypes of parts to identify potential problems that might arise much later during production stages. Design changes from the analysis results in costs that would be prohibitive. The creation of each physical prototype incurs tremendous time and cost allowing only a few of the several available design alternatives to be evaluated. Virtual prototypes on the other hand provide designers with similar testing scenarios without the time and cost commitments that are associated with physical prototyping. Flexible virtual prototypes also allow for instantaneous design changes making it possible to assess several design alternatives in a much shorter time span.

Virtual assembly simulations allow designers to import concepts into virtual environments during the early design stages and perform assembly/disassembly evaluations that would only be possible much later, when the first prototypes are built. Using haptics technology, designers can touch and feel complex CAD models of parts and interact with them using natural and intuitive human motions. Collision and contact forces calculated in real-time can be transmitted to the operator using robotic devices making it possible for him/her to feel the simulated physical contacts that occur during assembly. In addition, the ability to visualize realistic behavior and analyze complex human interactions makes virtual assembly simulations ideal for identifying assembly related problems such as awkward reach angles, insufficient clearance for tooling, and excessive part orientation during assembly, etc. They also allow designers to analyze tooling and fixture requirements for assembly. In addition to manufacturing, virtual assembly systems could also be used to analyze issues that might arise during service and maintainability operations such as inaccessibility to parts that require frequent replacement, etc. Expert assembly knowledge and experience that is hard to document could be captured by inviting experienced assembly workers from the shop floor to assemble a new design and provide feedback for design changes. Disassembly and recycling factors can also be taken into account during the initial design stages allowing for an environmentally conscious design. Virtual training is another application of assembly simulations which provide a platform for offline training of assembly workers which becomes important when assembly tasks are hazardous or specially complicated.

In order to reliably reproduce results from physical mockups, virtual assembly systems must be able to accurately simulate real world interactions with virtual parts, along with their physical behavior and properties. This research aims to create a virtual assembly environment capable of simulating the constant and subtle interaction (hand-part, part-part) that occurs during manual assembly, and providing appropriate feedback to the user in real-time. A virtual assembly system called SHARP “System for Haptic

Assembly and Realistic Prototyping” is created, which utilizes simulated physical constraints for part placement during assembly.

The first approach taken in this research attempt utilized Voxmap Point Shell (VPS) software from The Boeing Company for implementing collision detection and physics-based modeling in SHARP. A volumetric approach, where complex CAD models were represented by numerous small cubic-voxel elements was used to obtain fast physics update rates (500 – 1000 Hz). A novel dual-handed haptic interface was developed and integrated into the system allowing the user to simultaneously manipulate parts with both hands. However, coarse model approximations used for collision detection and physics-based modeling only allowed assembly when minimum clearance was limited to ~ 8-10%.

To provide a solution to the low clearance assembly problem, the second effort focused on importing accurate parametric CAD data (B-Rep) models into SHARP. These accurate B-Rep representations are used for collision detection as well as for simulating physical contacts more accurately. In this dissertation, a new hybrid approach is presented, which combines the simulated physical constraints with geometric constraints which can be defined at runtime. Different case studies are used to identify the suitable combination of methods (collision detection, physical constraints, geometric constraints) capable of best simulating intricate interactions and environment behavior during manual assembly. An innovative automatic constraint recognition algorithm is created and integrated into SHARP. The feature-based approach utilized for the algorithm design, facilitates faster identification of potential geometric constraints that need to be defined.

This approach results in optimized system performance while providing a more natural user experience for assembly.

Organization of Dissertation

The various aspects of the research introduced above are described in detail in four different papers that form the bulk of this dissertation. Chapter 2 presents an in-depth review of virtual assembly by categorizing previous research attempts based on the methods used for part placement in the virtual environment. The chapter elaborates on the challenges involved in different approaches and identifies future directions for research.

Chapter 3 provides the research challenges and implementation details of the initial volumetric approach used by the SHARP virtual assembly system. A detailed description of the dual-handed haptic interface is provided. The chapter describes the various modules (networking, subassembly creation, record and play) that form the SHARP system. System test results are presented which provide information about system performance during single and dual-handed interaction.

Chapter 4 of the dissertation presents a detailed analysis of complex interactions that are involved in completing a simple assembly task of inserting a pin into a hole. Implementation of a new hybrid approach to virtual assembly which combines physical constraints with geometric constraint-based modeling is then described. Various case-studies are used to identify the suitable combination of methods capable of best simulating intricate interactions and environment behavior during manual assembly.

Chapter 5 of the dissertation presents a novel feature-based constraint recognition algorithm that is developed and integrated into the SHARP system. The chapter provides implementation details of the algorithm and compares it with previous attempts for automatic constraint recognition for virtual assembly. Test results are presented to compare system performance with and without the use of the feature-based constraint algorithm in both simple and complex assembly scenarios. Finally, Chapter 6 presents conclusions and provides suggestions for future work.

CHAPTER 2. VIRTUAL ASSEMBLY: REVIEW & FUTURE DIRECTIONS

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

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

For submission to: *Computer Aided Design*

1. Introduction

Innovation is critical for companies to be successful in today’s global market. Competitive advantage can be achieved by utilizing futuristic and revolutionary approaches towards current engineering design practices. Such revolutionary approaches should encompass all aspects of product design (such as ergonomics, manufacture, maintenance, product life cycle etc.) during the early stages of product creation. Prototyping and evaluation are indispensable steps of the current product creation process. Although computer modeling practices such as CAD are currently being used at different stages, building one-of-a-kind physical prototypes makes the current prototyping process very costly and time consuming.

New technologies are needed which can empower industry with a faster and more powerful decision making process. The concept of virtual reality (VR) has evolved to a new level during the last two decades. VR has changed the ways scientists and engineers look at computers for performing mathematical simulations, data visualization, and decision making [1-4]. VR technology combines multiple human-computer interfaces to provide various sensations (visual, haptic, auditory, etc.) which give the user a sense of presence in the virtual world. This enables users to immerse in a computer generated scene and interact using natural human motions. The goal is to provide an “invisible interface” which allows the user to interact with the virtual environment as they would with the real world. This makes VR a perfect tool for simulating tasks that require frequent and intuitive manual interaction such as assembly methods prototyping. Several virtual assembly definitions have been proposed by researchers.

Jayaram et al. [5], in the year 1997, defined virtual assembly as “The use of computer tools to make or “assist with” assembly-related engineering decisions through analysis, predictive models, visualization, and presentation of data without physical realization of the product or supporting processes.” This definition emphasizes the use computer tools and visualization techniques for facilitating assembly related decisions however it does not specifically include immersive virtual environments and realistic interaction as an integral part of virtual assembly simulations.

A later definition by Kim and Vance [6] in 2003, described virtual assembly as the “ability to assemble CAD models of parts using a three-dimensional immersive, user interface and natural human motion". This advanced the definition to include a three-dimensional interface, VR and natural interaction as a critical part of virtual assembly.

Definitions of virtual assembly have evolved with VR technology; once visualization issues were resolved, the newer definition included natural interaction as a challenge for virtual assembly simulations. As VR continues to advance we would like to expand previous definitions to provide a more comprehensive description.

Virtual assembly in this paper is defined as the capability *to assemble virtual representations of physical models through simulating realistic environment behavior and part interaction to reduce the need for physical assembly prototyping resulting in the ability to make more encompassing design/assembly decisions in an immersive computer generated environment.*

2. Why Virtual Assembly?

Assembly process planning is a critical step in product development. In this process, details of assembly operations, which describe how different parts will be put together, are formalized. It has been established that assembly processes often constitute the majority of the cost of a product [7]. Thus, it is crucial to develop a proper assembly plan early in the design stage. A good assembly plan incorporates considerations for minimum assembly time, low cost, ergonomics and operator safety. A well designed assembly process can improve efficiency, quality, reduce cost and shorten product's time to market.

Expert assembly planners today still use traditional approaches in which they have to look at the three-dimensional (3D) CAD models of the parts to be assembled on two dimensional (2D) computer screens in order to examine part geometries and determining assembly sequences for a new product. Other methods for assembly planning include performing several trials by assembling physical prototypes and finding the best assembly sequence. As the assembly tasks get more complicated, such methods tend to be even more time consuming, costly and prone to errors.

Computer aided assembly planning is gaining popularity to solve these problems. A lot of research has been conducted for developing algorithms for generating suitable assembly sequences. However, there are several issues that still need to be addressed. For example, it is difficult to formalize the massive amount of expert knowledge utilized during assembly process design [8]. Also, as the number of parts in the assembly increase, the possible assembly sequences increase exponentially and thus it becomes more difficult to characterize criteria for choosing the most suitable assembly sequence for a given product [9].

Modern CAD systems also provide capabilities for building assemblies of CAD models. However, CAD systems use two-dimensional interfaces such as the keyboard and mouse and depend on the constraint information where the user has to manually select the mating surfaces, axes and edges to assemble the parts. Thus, these interfaces do not reflect human interaction with complex parts. Thus, it becomes difficult to foresee issues that appear during assembly, maintenance and service operations, for example ensuring accessibility for part replacement during maintenance and the effects of changing assembly sequences. Such computer-based systems also lack in addressing issues related to ergonomics such as awkward to reach assembly operations, etc.

VR technology plays a vital role in simulating such advanced 3D human-computer interactions by providing users with different kinds of sensations (visual, auditory and haptic) for creating an increased sense of presence in a computer generated scene. The aim for performing virtual assembly simulations is to generate the most suitable process for assembling a product to reduce the costly and time consuming physical prototyping process. Using such virtual prototyping applications, design changes can be incorporated easily in the conceptual design stage thus optimizing the design process towards Design for Manufacture and Assembly (DMA). Using VR technology, simulations can be performed and different assembly sequences can be analyzed by utilizing existing CAD data.

To replace/reduce the current prototyping practices a virtual assembly simulation should be capable of answering the following question:

- Can any given set of parts be assembled?
- Can we perform task-time and sequence analyses?
- Is it possible to perform disassembly procedures for service and maintenance?
- Can we identify fixture and tooling requirements for an assembly?
- Can we perform ergonomic studies for avoiding hard-to-perform assembly steps and analyze operator strain?
- Can we use the system for offline training?

Once successful, this capability will provide the basis for many useful virtual environments that will address various aspects of the product life cycle such as ergonomics, workstation layout, tooling design, off-line training, maintenance, and serviceability prototyping (figure 1).

3. Virtual Assembly - Challenges

An assembly operation in a virtual environment consists of two steps:

1) Detect collisions among parts to allow part grabbing and to prevent part interpenetrations
2) Place part in the final position using available cues which can be:
    a. Pre-defined final part positions - snapping
    b. Geometric Constraints
    c. Guiding part in the final position by simulating physical constraints that are computed real-time using physics-based modeling techniques.

There are several challenges that exist while performing assembly simulations using VR: data transfer between CAD VR systems, accurate collision detection, realistic physical simulation, inter-part constraint detection and management, intuitive object manipulation, support for complex CAD data etc. In the following section, a few of these

Figure 1: Applications of a Virtual Assembly/Disassembly Simulation

challenges are outlined and previous approaches in each area are summarized.

3.1 Collision Detection

Virtual assembly simulations present a bigger challenge than virtual walkthrough environments as they require frequent human-interaction and real time simulation involving complex models. Real world assembly tasks require extensive interaction with surrounding objects including grabbing parts, manipulating them realistically and finally placing them in the desired final position and orientation. Thus, for successfully modeling such a complex interactive process, the virtual environment not only needs to simulate visual realism, it also needs to model realistic part behavior of the virtual objects. For example, graphic objects should not pass through each other and should behave realistically when external forces are applied. The first step to accomplish this is implementing accurate collision detection among parts [10].

Collision detection algorithms provide useful information (contact points, number of contacts, minimum Euclidian distance between objects etc.) to the system when a collision has occurred or is going to occur. Several algorithms have been developed in the past, which detect collisions using different representations (polygon representation, algebraic surfaces or splines) of geometric models. Polygonal models are most commonly used in computer graphics applications because of their simplicity and hardware-accelerated rendering supported by graphics hardware manufacturers. Several algorithms that use polygonal data for collision detection were designed by researchers at University of North Carolina (I-collide [11], SWIFT [12], RAPID [13], V-collide [14], SWIFT++ [15], CULLIDE [16] etc.) among others V-Clip [17], VPS [18] etc. A comprehensive review of collision detection algorithms can be found in [19, 20] and a taxonomy of collision detection approaches can be found in [21].

Due to the highly complex nature of CAD geometry (which consist of high polygon counts or complex parametric curves and surfaces etc.) that is used for assembly simulations; detecting collisions among several such models accurately may take a long time. In virtual environments where interactive simulation is critical, fast and accurate collision detection among dynamic objects is a challenging problem.

Once implemented, collision detection prevents part interpenetration. However, the system with only collision detection, does not provide any feedback to the user about how to change position and orientation of parts to align them for completing the assembly operation [22]. Two techniques are used in the literature for implementing part positioning during an assembly. The first technique uses physics-based modeling which simulates realistic behavior of parts in a virtual scene. Parts are assembled together with the help of simulated physical constraints which are calculated in real-time. The second technique utilizes geometric constraints similar to those used by modern CAD systems. In this approach, geometric constraints such as concentric, coplanar etc. are applied between parts thus reducing the degrees-of-freedom and facilitating the assembly task at hand. Examples of virtual assembly applications using these techniques are provided in the next section.

3.2 Physics-Based Modeling

The physics-based modeling approach relies on simulating physical constraints for assembling parts in a virtual scene. It has been seen that realistic physical modeling can significantly enhance the user’s sense of immersion and interactivity, especially in manipulation intensive applications [23]. Physics-based algorithms simulate forces acting on bodies along with their physical properties in order to model realistic behavior. Such algorithms solve equations of motion of the objects at each time step, based on forces and torques that act upon the objects.

Physics-based modeling algorithms can be classified into three categories based on the method used, namely the penalty force method, the impulse method, and the analytical method. In the penalty force method, a spring damper system is used to prevent interpenetration between models. Whenever a penetration occurs, a spring damper system is used to penalize it [18, 24]. Penalty based methods are easier to use and computationally inexpensive, however they are characterized with problems caused by very high spring stiffness leading to stiff equations which are numerically intractable [25]. The impulse based methods [26-28] simulates interactions among objects using collision impulses. Static contacts in this approach are modeled as a series of high frequency collision impulses occurring between the objects. The impulse based methods are known to be more stable and robust than penalty force methods. However, these methods have problems handling stable and simultaneous contacts (such as stack of blocks at rest) and also in modeling static friction in certain cases like sliding [29]. The analytical method [30, 31] checks for interpenetrations. If found the algorithm backtracks the simulation to the point in time immediately before the interpenetration. Based on contact points, a system of constraint equations is solved to generate contact forces and impulses at every contact point [32]. The results from this method are very accurate however it requires extensive computation time when several contacts occur simultaneously.

Thus, although various algorithms for physics-based modeling have evolved over the years, simulating realistic behavior among complex parts interactively and accurately is still a challenging task.

3.3 Inter-Part Constraint Detection and Management

Due to the problems related to physics-based modeling (instability, difficult to attain interactive update rates, accuracy etc.), several approaches using geometric constraints for virtual assembly have been proposed. Constraint-based modeling approaches use inter-part geometric constraints (that are predefined and imported or defined on the fly) to determine relationships between components of an assembly. Once constraints are defined and applied, the constraint solver computes the new and reduced degrees-of-freedom of objects and the object's resulting motion.

A vast amount of research focused on solving systems of geometric constraints exists in the literature. Numerical constraint solver approaches translate constraints into a system of algebraic equations. These equations are then solved using iterative methods such as Newton-Raphson [33]. Good initial values are required to handle exponential number of possible solutions. Although solvers using this method are capable of handling large non-linear systems, most of them have difficulties handling over-constrained and under-constrained instances [34] and are computationally expensive which makes them unsuitable for interactive applications such as virtual assembly [35]. Constructive constraint approaches are based on the fact that in principle, most configurations of engineering drawings can be solved on a drawing board using standard drafting techniques [36]. In the rule-constructive method, “solvers use rewrite rules for discovery and execution of construction steps”. Although complex constraints are easy to handle, exhaustive computation requirements (searching and matching) of these methods make them inappropriate for real world applications [37]. Examples of this approach can be found in [38-40]. Graph-constructive approaches are based on analysis of the constraint graph. Based on the analysis, a set of constructive steps are generated. These steps are then followed to place the geometries relative to each other. Graph constructive approaches are fast, methodical and provide means for developing robust algorithms [36, 37, 41, 42]. An extensive review and classification of various constraint solving techniques can be found in [34].

4. Review of Virtual Assembly Applications

In this section, research in the area of virtual assembly simulations is reviewed. The literature is classified on the basis of the methodology used for assembling parts. Assembly applications are classified into two categories: constraint-based and physics-based systems.

The first category consists of systems that use constraints to place parts in their final position and orientation in the assembly. It has been recognized that geometric constraints prove to be useful in precise part positioning tasks in a virtual environment where physical constraints are absent [43]. Constraints in the context of this paper are of two types. The first are positional constraints which are pre-defined final part positions. The second type of constraints is geometric constraints which are relationships among part features which are applied when related objects are in proximity. Constraint based methods summarized in section 3.3 are used to solve for relative object movements.

The second category of applications includes assembly systems which simulate real world physical properties, friction and contact forces to assemble parts in a virtual environment. These applications allow users to move parts freely in the environment. When a collision is detected physics-based modeling algorithms, as described in section 3.2, are used to calculate subsequent part trajectories to allow for realistic simulation.

4.1 Constraint-Based Assembly Applications

4.11 Systems Using Positional Constraints

IVY (Inventor Virtual Assembly) at Iowa State University was developed by Kuehne and Oliver [44]. IVY used IRIS Open Inventor graphics library from Silicon Graphics and was used by designers to interactively verify and evaluate the assembly characteristic components directly imported from a CAD package. The purpose of IVY was to provide a design tool to support true design-for-assembly (DFA). Once, the assembly was completed, the application rendered a final animation of the assembly steps.

A PC-based system for virtual assembly "Vshop" (figure 2) was developed by Pere et al. [45]. The application used World Toolkit, a

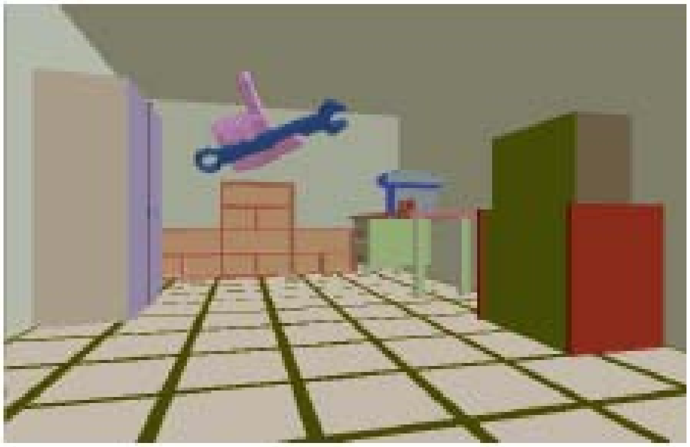


Figure 2: VShop User Interface
(obtained from [45])

commercial software toolkit for easy creation of interactive VR applications. Bounding box collision detection was implemented to avoid object interpenetration. The OpenGL graphics library was used for visualizations and 3D graphics models created in Autodesk Inventor and AutoCAD were imported in the environment. Hand position tracking was provided by Polhemus Insidetrac tracking system. Rutgers Mater II haptic exoskeleton was used for gesture recognition and to provide tactile feedback to the user. Hand gesture recognition was used for various tasks like switching on and off navigation and moving forward/backward in the environment. Collision forces were calculated using the exoskeleton's SIS driver and gravity was simulated graphically without using any physics-based methods.

Ye et al. [46] conducted an experiment for investigating the potential benefits of VR environments in supporting assembly planning. For virtual assembly planning, a non-immersive desktop VR environment and an immersive CAVE (CAVE Computer Aided Virtual Environment) [47, 48] environment were evaluated. The desktop VR environment consisted of a Silicon Graphics workstation. The CAVE environment used IRIS Performer CAVE interface and provided the subjects with a realistic sense of virtual assemblies and parts. The experiment compared assembly

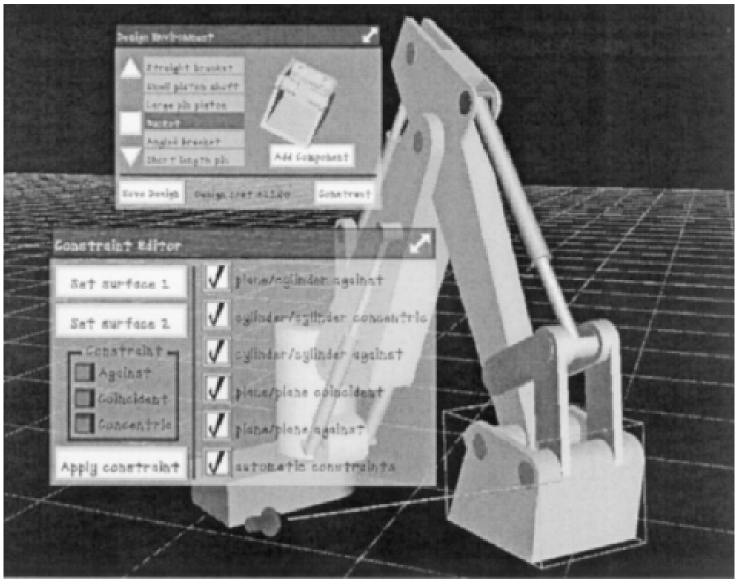
Figure 3: Presentation of Aircylinder assembly in Ye's Application
(obtained from [46])

operations in a traditional engineering environment and in immersive and non-immersive VR environments. The three environments differed in how the assembly was presented and handled. The assembly task was to generate an assembly sequence for an air-cylinder assembly (figure 3) consisting of 34 parts. The results from the human subject study concluded that the subjects performed better in VEs than in traditional engineering environments in tasks related to assembly planning.

A virtual assembly system was developed by Bullinger et al. [49] at Fraunhofer-Institute of Industrial Engineering. The application used VRANTHROPOS which utilized human models that were developed based on anthropometric data for addressing virtual ergonomic prototyping issues. A Head Mounted Display (HMD) was used for stereo viewing and a data glove was used for gesture recognition. Head and hand tracking was implemented using magnetic trackers. While performing assembly tasks, the users could see their corresponding human model in virtual environment. The system produced a script file describing the sequence of actions performed by the user for assembling the product.

BMW developed a virtual assembly system for performing assembly simulations using virtual prototypes [50]. The system used a three layer (scene graph layer, scripting layer and application layer) framework. A Cyber Touch glove device was used for gesture recognition (for holding parts) and for providing tactile force feedback. Proximity snapping technique was used for part placement and interaction with the system was assisted by voice input. Gestures from the glove device were also used for navigating the virtual environment. For immersive visualization, a HMD device was used. A user study was conducted in which five different groups with diverse backgrounds participated. The users found tactile feedback to be unrealistic. It was concluded that force feedback is crucial for performing virtual assembly tasks.

4.12 Systems Using Geometric Constraints



☑ plane/cylinder against
☑ cylinder/cylinder concentric
☑ cylinder/cylinder against
☑ plane/plane coincident
☑ plane/plane against
☑ automatic constraints

Figure 4: Marcelino's Constraint Manager Interface
(obtained from [51])

A geometric constraint manager system was developed by Marcelino et al. [51] at University of Salford, for simulating interactive assembly/disassembly tasks in VEs. The system design (figure 4) supported features like multi-platform operation, scene graph independence, multiple constraint recognition and automatic constraint management. The system defined surfaces using parametric equations and each surface had a specific bounding volume for defining surface limits. The constraint manager was capable of validating existing constraints, determining broken constraints, enforcing existing constraints, solving constrained motion and recognizing new constraints in a system. The system was optimized in order to make it capable of simulating assembly of industrial CAD geometry.

VADE (Virtual Assembly Design Environment) was developed by Jayaram et al. [5, 52-55] in collaboration with NIST, at Washington State University. VADE (figure 5) offered features like dual handed assembly and dexterous manipulation of objects in an immersive virtual environment. The CyberGrasp haptic device was used for tactile feedback. VADE used Pro/Toolkit import assembly data (transformation matrices, geometric constraints, assembly hierarchy etc.) to facilitate assembly operations in a virtual environment. A physics-based algorithm with limited capabilities was later added to VADE for simulating realistic part behavior [56]. Stereo vision was provided by a HMD or and Immersadesk [57] system. VADE was expanded to perform ergonomic evaluation for assembly tasks by integrating an ergonomic software package [58, 59].

Figure 5: Swept Volumes in VADE
(obtained from [52])

Another multimodal CAVE-based system for virtual assembly called MIVAS (A Multi-Modal Immersive Virtual Assembly System) was developed at Zhejiang University by Wan [60]. MIVAS used constraints to simulate part behavior in the virtual environment. The application performed hand-to-part collision detection using VPS software while part-to-part collision detection was implemented using RAPID. The users can feel the size and shape of digital CAD models using the CyberGrasp haptic device from Immersion Corporation. Since Haptic feedback was only provided in griping tasks, the application lacked in providing force information when parts collided.

Chen et al. [61] 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. For multimodal interaction, the system supported speech and gesture recognition.

Liu et al. [62] developed a virtual assembly system that used constraint-based modeling for assembly and tolerance analysis. The system used the concept of “assembly ports” which were comprised of information about the mating part surfaces for example geometric and tolerance information, assembly direction and type of port (hole, pin, key etc.). If parts were modified by a design team, the system used assembly port information to analyze if new designs could be re-assembled successfully. Different rules were used (proximity, orientation, port type and parameter matching) for applying constraints among parts. Gesture recognition was implemented using a CyberGlove device. A user study was conducted which confirmed that constraint-based modeling was beneficial for users when performing precise assembly positioning tasks [63].

A CAD-linked virtual assembly environment was developed by Wang et al. [64] which utilized constraint-based modeling for assembly. The desktop-based system ran as a standalone process and maintained communication with Autodesk Inventor® CAD software. Low level-of-detail (LOD) proxy representations of CAD models were used for visualization in the virtual environment. The assembly system required persistent communication with the CAD system using proprietary APIs for accessing information such as assembly structure, constraints, B-rep geometry and object properties.

Yang et al. [65] used constraint-based modeling for assembly path planning and analysis. Assembly tree data, geometric data of parts and predefined geometric constraints were imported from parametric CAD systems such as Pro/Engineer using a special data converter. A data glove device and a hand tracker were used for free manipulation of objects in the virtual environment. The automatic constraint recognition algorithm activated the pre-defined constraints when bounding boxes of the interrelated parts collided. The users were required to confirm the constraint before it could be applied. These capabilities were applied to the integrated virtual assembly environment (IVAE) system.

4.2 Physics-Based Modeling Applications

VEDA (Virtual Environment for Design for Assembly) a desktop-based assembly system was developed by Gupta [66, 67]. The system used physics-based modeling for modeling part behavior. The application used a dual phantom interface for interaction and provided haptic, auditory and stereo cues to the user for part interaction. However, VEDA was only limited to handling 2D models to maintain interactive update rate of the application.

Kim [6, 68] investigated several collision detection and part behavior algorithms at Iowa State University. VPS [18] software from Boeing Corporation was found to be most applicable for handling arbitrary CAD geometry while performing physics-based modeling and collision detection. The application (figure 6) expanded the functionality of VEGAS [69] and implemented physics-based modeling to simulate realistic part behavior. The system used SGI OpenGL Performer for visualization. A six-sided immersive projection screen VR system was used for performing assembly. Additional position trackers were placed on the user's wrist, forearm and upper arm to simulate a virtual arm model for collision detection while performing assembly. Dual handed assembly was supported and gesture recognition was done using wireless data glove device from 5DT corp. [70] VRJuggler [71] software library was used as a platform for VR application development.

NHE (Network Haptic Environment) was developed by Kim [72] to facilitate collaborative assembly through internet. A combination of peer-to-peer and server-client architecture was developed to maintain the stability and consistency of the system data. A "Release-but-not-released - RNR" method was developed for allowing computers with different performance capabilities to participate in the network. The system architecture required each virtual environment to be connected to a local PC machine that maintains the haptic device. This was done to assure 1Khz update rate for smooth haptic interaction.

Figure 6: Kim's Assembly Application
(obtained from [6])

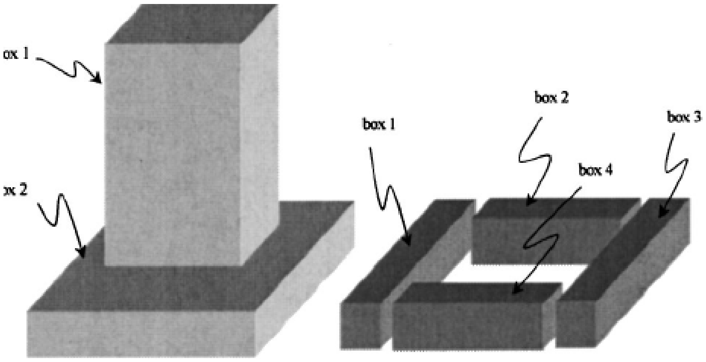


Figure 7: Geometry in HIDRA
(obtained from [73])

HIDRA (Haptic Integrated Dis/Re-assembly

Analysis), a desktop virtual assembly application was developed by Coutee and Bras [73, 74] at Georgia Institute of Technology. HIDRA used GHOST (General Haptic Open Software Toolkit) from Sensable Technologies [75] and two Phantom® devices for simulating physical behavior of parts in a desktop environment. OpenGL was used for visualization and V-Clip in conjunction with Q-hull and SWIFT++ were used for collision detection. Because HIDRA (figure 7) treated ‘fingertip’ as a point rather than a surface, it created difficulties while handling complicated geometries. Also using GHOST SDK for physical modeling and polygon soup based collision detection; HIDRA had limitations while handling non-convex CAD geometry.

Fröhlich et al. [76] used CORIOLISTM [77] physics-based simulation package to develop an interactive virtual assembly environment using the Responsive Workbench [78]. Different configurations of spring based virtual tools were developed to interact with objects. The system used the workbench in table top configuration and supported multiple tracked hands and users to manipulate an object. The system lacked in providing interactive update rates when several hundred collisions occurred simultaneously. At least five percent tolerance was necessary to avoid numerical instabilities which sometimes resulted in breaking the system.

Seth et al. [79, 80] developed SHARP: System for Haptic Assembly and Realistic Prototyping. The system presented a dual-handed haptic interface for virtual assembly which provided collision force feedback using two PHANToM® haptic devices (figure 8) Direct data transfer from CAD to VR was implemented by using generic CAD formats. SGI Performer was used for graphics rendering and Open Haptics Toolkit library was used for communicating with the haptic devices. VPS [18] was used for collision detection and physics-based modeling. The system provided the capability of being ported to different VR systems configurations including low-cost desktop configurations, Barco Baron [81], Power Wall, four-sided and six sided CAVE systems. The network display module of the system allowed it to communicate with multiple VR systems (such as CAVE) at geographically dispersed locations. SHARP also supported swept volume generation and visualization.

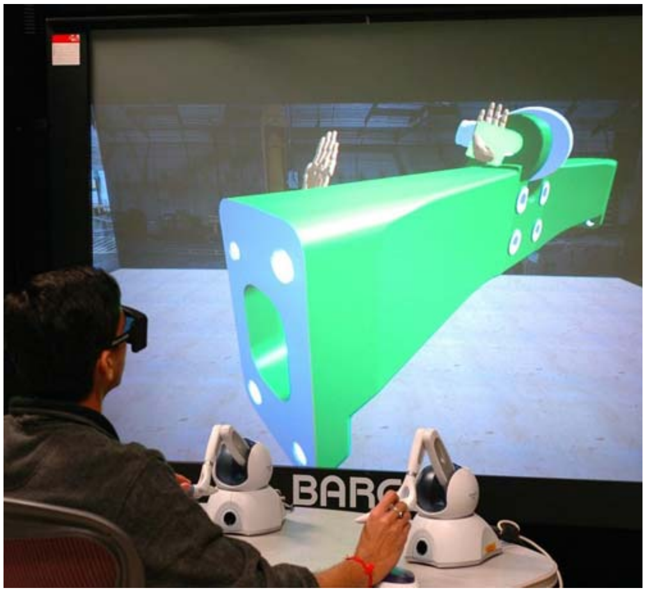


Figure 8: SHARP Haptic Interface
(obtained from [80])

Garbaya et al. [82] created a physics-based virtual assembly system which used spring-damper model to provide the user with collision and contact forces during mating phase of the assembly operation. An open source PhysX® toolkit was used for collision detection and physically-based modeling. Grasping force feedback was provided using a CyberGraspTM haptic device and collision force was provided using CyberForceTM haptic device from Immersion Corporation. An experimental study was conducted to check the system effectiveness and user performance in real and virtual environments. The study concluded that the user performed increased when inter-part collision forces were rendered as compared to when only grasping forces are rendered to the user.

4.3 Review Summary

The review in the above section shows that several approaches have been implemented over the years for performing assembly simulations. Initial efforts in simulating assembly used pre-defined transformation matrices of parts for positioning them in the virtual scene. Users moved parts in the environment which were snapped in place when close to their pre-defined final positions (figure 9). Most of the early applications did not implement collision detection among objects which allowed them to interpenetrate during the simulation.

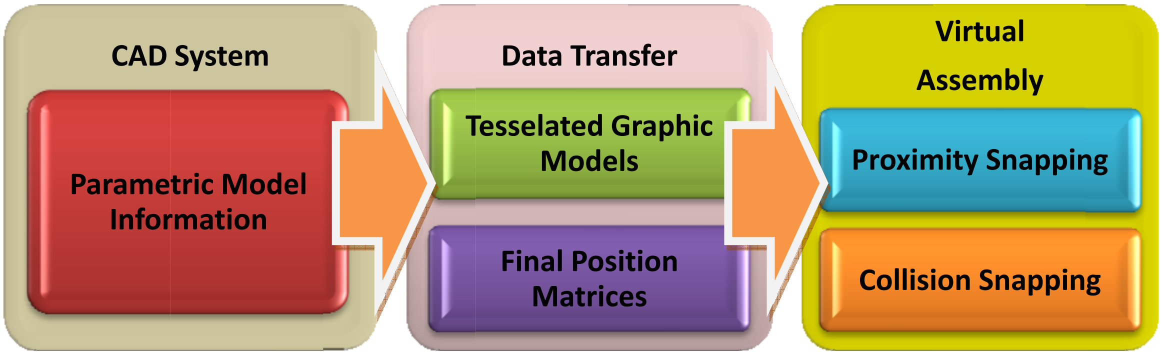


Figure 9: Data Transfer in Positional Constraint Applications

Another approach to virtual assembly used geometric-constraint relationships which were pre-defined and imported from a CAD system for assembling parts. Here, the pre-defined constraints were activated when related parts came in close proximity to each other. Once geometric-constraints were recognized, constrained motion could be visualized between parts which were then assembled using pre-defined final positions [52]. Such applications implemented collision detection between models for preventing model interpenetration during assembly. Constraint-based approaches have shown promising results in the past. They present less computation and memory requirements as

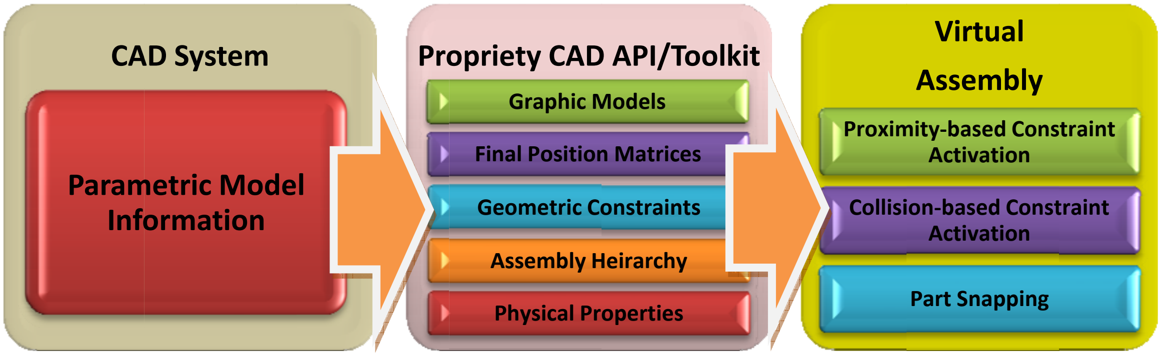


Figure 10: Data Transfer in Geometric Constraint-based Applications

compared to physics-based methods. At the same time, the use of accurate models (parametric representations, B-Reps) for calculations allows the users to position parts in an assembly with very high accuracy. However, some of these applications required special CAD toolkits to extract relevant CAD metadata (figure 10) that was required for preparing an assembly scenario [52, 60, 61]. Such special data requirements and dependence on specific CAD systems has hindered widespread acceptance of such applications. Advanced constraint-based methods were successful in identifying, validating and applying constraints on-the-fly and thus did not require importing predefined CAD constraints [51, 83]. Although, constraint-based modeling proved successful for assembling parts; this method alone does not support simulating realistic behavior of objects in a virtual simulation.

The third approach has been used for assembling CAD models relied on simulating real world physical constraints in a virtual simulation (figure 11). Physics-based modeling completes the definition of VR as defined by Jayaram et al. [84] by adding physical properties (such as mass, moment of inertia, center of mass etc.) to the geometry only model used by the simulation, thus enabling the virtual objects to behave realistically. Physics-based methods allow for testing scenarios similar to those possible

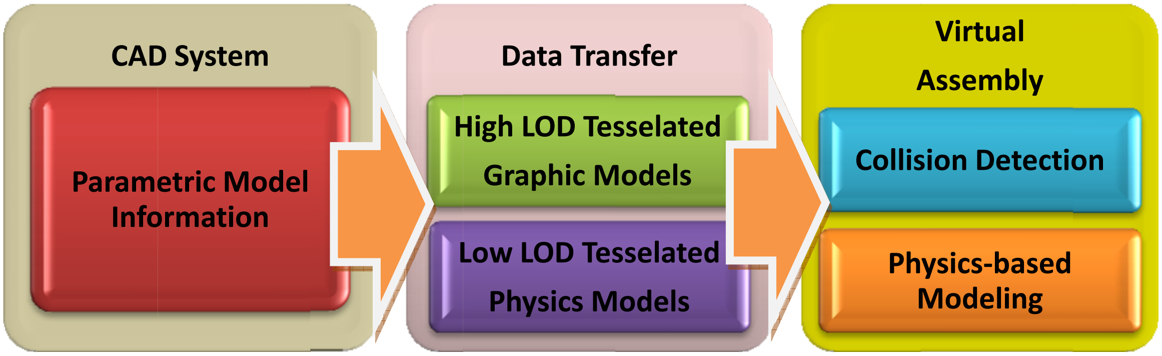


Figure 11: Data Transfer in Physics-based Applications

only by physical mock-ups by calculating subsequent part trajectories after collisions occur, based on collision, friction, gravity and other forces that act on the objects. In addition, physics-based methods also lay the foundation for the implementation of haptic interfaces for virtual prototyping applications. Such haptic interfaces allow users to touch and feel virtual models that are present in the simulation. In prototyping applications like virtual assembly, attempts have been made to provide collision and tactile forces to the users for more intuitive interaction with the environment [72, 73, 79, 80, 85]. Identifying physical constraints among an arbitrary set of complex CAD models in a dynamic virtual simulation is a very complicated problem. It generally consists of two main steps. First the application needs to detect collisions among the models. After a collision is detected, physics-based modeling techniques are used to predict realistic part behavior. Collision and physics responses need to be calculated at interactive rates to respond to user's actions in real time. Physics-based solvers generally sacrifice computational accuracy to keep the update rate of the visual simulation realistic [20]. Most previous efforts used a simplified and approximated polygon mesh representations of CAD models for faster collision and physics calculations. Other efforts generated representations by using cubic voxel elements for physics and collision calculations [18, 86, 87]. Assembly configurations like a tight peg in a hole caused several hundreds of collisions to occur which often resulted in numerical instabilities in the system [76]. High update rate requirements of ~1KHz, for haptic feedback interfaces makes such problems even more complex and hard to address. In order to complete assembly tasks with tight tolerances, parts needed to be reduced in size [77, 80]. However, because assembly operations require mating low clearance surfaces, it was not possible to assemble low-clearance parts with actual dimensions using physics-based methods. The demand for highly accurate physics/collision results while maintaining simulation interactivity is still a challenge for the community. In addition there are several generic challenges which affect all these approaches for virtual assembly.

4.4 CAD-VR Data Exchange

CAD-VR data exchange is one important issue facing the virtual prototyping community. CAD systems used by the industry to develop their product models are generally unsuitable for producing optimal representations for VR applications. Most VR applications use scene-graphs (Openscenegraph, OpenSG etc.) for visualization which require simplified polygonal geometry to ensure interactive frame rates. Translating existing parametric CAD data presents problems of “excessive number of polygons and number of objects that are created” [88]. The problem becomes even more challenging when incorporating pre-existing texture maps in these optimized model representations.

During this translation process, the parametric information of the CAD model generally does not get imported into the VR application. In virtual assembly simulations, geometric constraint-based applications that depend on parametric model definitions for defining inter-part constraint relationships generally have to deal with two representations of the same model: one for visualization and another for constraint modeling algorithms for performing assembly. Similarly, physics modeling applications also use dual model representations: high-fidelity model for visualization and a coarser representation used for interactive physics calculations [76, 79].

Also, the use of pre-defined assembly data (constraint relationships, transformation matrices, assembly hierarchy) require assembly applications to use special CAD toolkits [52] for importing CAD metadata (pre-defined geometric constraints, transformation matrices, assembly hierarchy, physical properties etc.) which makes them dependent on a particular CAD system thus hindering widespread acceptance.

Commercial service providers (AutoCAD, UGS, Dassault Systems, RealD Corp.) have made attempts to embed capabilities for immersive and desktop stereo visualization into available commercial software to some degree. Attempts have also been made by academia to provide haptic interaction and immersive visualizations for assembly/disassembly applications within commercial CAD systems [89]. Thus, although addressed to some degree by industry and academia, there is still no general non-proprietary way to convert CAD assemblies into a representation suitable for VR.

Additionally, today’s VR applications have matured to a level where they provide users with the ability to identify meaningful design changes however, translating these changes back to CAE applications (eg. CAD systems) is currently not possible. The efforts mentioned above represent a promising basis for this research, but as yet, it remains a major bottleneck to further industrial adoption of VR.

4.5 Haptic Interaction

Today's virtual assembly environments are capable of simulating visual realism to a very high level. The next big challenge for the virtual prototyping community is simulating realistic interaction. Haptics is an evolving technology that offers a revolutionary approach to realistic interaction in VEs. "Haptics means both force feedback (simulating object hardness, weight, and inertia) and tactile feedback (simulating surface contact geometry, smoothness, slippage and temperature)" [23]. Force cues provided by haptics technology can help designers feel and better understand the virtual objects by supplementing visual and auditory cues and creating an improved sense of presence in the virtual environment [90]. Research has shown that the addition of force feedback to virtual environments resulted in increasing task efficiency times [91, 92].

Highly efficient physics-based methods that are capable of maintaining high update rates are generally used for implementing haptic feedback in virtual assembly simulations. Various approaches for providing haptic feedback for assembly have been presented in the past which focused on developing new methods for providing tactile [45, 52, 60, 89, 93], collision [72, 79, 80] and gravitational force feedback [90, 94]. High update rate (~1KHz) requirements for haptics have always been a challenge while integrating this technology. As already stated earlier, most physics-based algorithms used highly coarse model representations to keep up with the update rate requirements. Lack of accuracy of such algorithms presents problems when detailed contact tasks are necessary. Simulating complex part interactions such as grasping is also demanding as it requires the simulation to detect collisions and generate contact forces accurately for each individual finger [60, 85, 89, 95]. Maintaining update rates for haptic interaction (~1KHz) while performing highly accurate collision/physics computations in complex interactive simulations such assembly is still a big challenge for the community.

In addition, there are several limitations of the haptics technology itself. Non-portable haptic devices such as Sensable Technologies' PHANToM® [75, 96], Immersion's CyberForceTM [97], Haption Virtuose [98], and Novint Falcon [99] devices [65] among others [100, 101] have workspace limitations which results in restricted user motion in the environment. Additionally, because these devices need to be mounted, their use in projection screen immersive virtual environments becomes difficult. On the contrary, wearable haptic gloves and exoskeleton devices such as CyberTouch^{TM}, CyberGrasp^{TM} [97], Rutgers Master II [102] among others [94] provide a much larger workspace for interaction. However, they only provide force feedback to fingers and palm and thus are only suitable for tasks that involve dexterous manipulations. A detailed discussion on haptics issues can be found in [10]. The challenges presented here among several others are needed to be addressed, before the community can explore the real potential that haptics technology brings to the task of virtual prototyping.

5. Discussion & Future Directions

Collision detection algorithms unquestionably form the first step towards building a virtual assembly simulation system. Although they add to simulation realism by preventing part interpenetrations; they do not provide any help for adjusting relative part orientations to facilitate assembly operations. Thus, the next question that arises is which technique should be used for part placement to make future virtual assembly applications a success.

Constraint-based approaches provide the ability to precisely position parts in VEs. Physics-based approaches on the other hand enable virtual mock-ups to behave as their physical counterparts. Both these approaches serve different purposes which are crucial in making a virtual assembly simulation successful.

An ideal approach would be to combine physics-based and constraint-based methods. The resulting virtual assembly application will be able to simulate realistic environment behavior for enhanced sense of presence and would also allow to position parts precisely in a given assembly. An attempt has been made in the past to implement physics-based algorithms with limited capabilities within an existing constraint-based assembly system [56]. However, limitations of the physics algorithm, part snapping and excessive metadata requirements using CAD system dependent toolkit prevented its widespread impact.

In the proposed approach, physics-based methods will be used for simulating realistic part behavior and haptic interaction. Constraint-based methods will come into play when low clearance assembly needs to be performed to allow for precise movement of parts into their final position. The challenge in this approach is that physics-based methods should be able to take into account the presence of a geometric constraint and the “hybrid solver” should be able to calculate part trajectories in such a way that both physical and geometric constraints are satisfied at any given point of time.

As the technology progresses, the cost of computing and visualization technology will continue to fall as their capabilities increase. It will soon be possible to utilize this power to integrate faster and more accurate algorithms into virtual assembly simulations that will be capable of handling large assemblies with hundreds of parts.

6. References

[1] S. Bryson, "Virtual Reality in Scientific Visualization," Communications of the ACM, vol. 39, pp. 62-71, 1996.

[2] J. Eddy, and Lewis, K. E., "Visualization of Multidimensional Design and Optimization Data using Cloud Visualization," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2002/DAC-34130), Montreal, Canada, 2002.

[3] F. Zorriassatine, Wykes, C., Parkin, R., and Gindy, N., "A Survey of Virtual Prototyping Techniques for Mechanical Product Development," Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, vol. 217, pp. 513-530, 2003.

[4] Y. Xianglong, Yuncheng, F., Tao, L., and Fei, W., "Solving Sequential Decision-making Problems Under Virtual Reality Simulation System," presented at Winter Simulation Conference Proceedings, Arlington, Virginia, 2001.

[5] S. Jayaram, Connacher, H. I., and Lyons K.W., "Virtual Assembly using Virtual Reality Techniques," Computer Aided Design, vol. 29, pp. 575-584, 1997.

[6] C. E. Kim, and Vance, J.M., "Using Vps (Voxmap Pointshell) As The Basis For Interaction in a Virtual Assembly Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2003/CIE-48297), Chicago,IL., 2003.

[7] G. Boothroyd, and Dewhurst, P., Product Design for Assembly. New York: McGraw-Hill, Inc., 1989.

[8] J. M. Ritchie, Simmons, J.E.L., Carpenter, I.D. and Dewar., R.G, "Using Virtual Reality for Knowledge Elicitation in a Mechanical Assembly Planning Environment," presented at Proceedings of 12th Conference of the Irish Manufacturing Committee, 1995.

[9] R. G. Dewar., Carpenter, I.D., Ritchie, J.M. and Simmons, J.E.L., "Assembly Planning in a Virtual Environment," presented at Proceedings of Portland International Conference on management of Engineering and Technology (PICMET 97), Portland, OR, 1997.

[10] G. C. Burdea, "Haptics Issues in Virtual Environments," presented at Computer Graphics International, Geneva, Switzerland, 2000.

[11] J. D. Cohen, M. C. Lin, D. Manocha, and M. K. Ponamgi, "I-COLLIDE: An Interactive and Exact Collision Detection System for Large-Scale Environments," The 1995 ACM International 3D Graphics Conference, pp.189-196, 1995.

[12] S. A. Ehmann and M. C. Lin, "SWIFT: Accelerated Proximity Queries Between Convex Polyhedra By Multi-Level Voronoi Marching," Technical report, Computer Science Department, University of North Carolina at Chapel Hill, 2000.

[13] S. Gottschalk, M. C. Lin, and D. Manocha, "OBB-Tree: A Hierarchical Structure for Rapid Interference Detection," ACM SIGGRAPH'96, pp. 171-180, 1996.

[14] T. Hudson, M. C. Lin, J. Cohen, S. Gottschalk, and D. Manocha, "V-COLLIDE: Accelerated Collision Detection for VRML," Proceedings of the second symposium on Virtual Reality Modeling Language, pp. 119-125, 1997.

[15] S. A. Ehmann and M. C. Lin, "Accurate and Fast Proximity Queries between Polyhedra Using Surface Decomposition," Eurographics. Computer Graphics Forum, vol. 20, 2001.

[16] N. K. Govindaraju, Redon, S., Lin, M. C. and Manocha, D., "CULLIDE: Interactive Collision Detection Between Complex Models in Large Environments using Graphics Hardware," In Proceedings of ACM SIGGRAPH/Eurographics Workshop on Graphics Hardware, pp. 25-32, 2003.

[17] B. Mirtich, "V-Clip: fast and robust polyhedral collision detection," ACM Transactions on Graphics, vol. 17(3), pp. 177-208, 1998.

[18] W. A. McNeely, Puterbaugh, K. D. and Troy, J. J., "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," presented at SIGGRAPH 99 Conference Proceedings, Annual Conference Series, Los Angles, CA., 1999.

[19] M. Lin, and Gottaschalk, S., "Collision Detection between Geometric Models: A survey," presented at Proceedings of IMA Conference on Mathematics of Surfaces, 1998.

[20] P. Jiménez, Thomas,F., and Torras, C., "3D Collision Detection: A Survey," Computers and Graphics, vol. 25, pp. 269-285, 2001.

[21] D. Borro, Hernantes, J., Garcia-Alonso, A., and Matey, L., "Collision Problem: Characteristics for a Taxonomy," presented at Proceedings of the Ninth International Conference on Information Visualization (IV'05), 2005.

[22] B. Frohlich, Tramberend, H., Beers, A., Agarawala, M. and Baraff, D., "Physically-Based Modeling on the Responsive Workbench," presented at IEEE Virtual Reality Conference, 2000.

[23] G. C. Burdea, "Invited Review: The Synergy between Virtual Reality and Robotics," IEEE Transactions on Robotics and Automation, vol. 15, pp. 400-410, 1999.

[24] K. Erleben, Sporring J., Henriksen, K. and Dohlmann, H., Physics-Based Animation, First ed. Hingham, MA: Charles River Media, Inc., 2005.

[25] A. Witkin, Gleicher, M. and Welch, W., "Interactive Dynamics," Computer Graphics, vol. 24, pp. 11-22, 1990.

[26] J. K. Hahn, "Realistic Animation of Rigid Bodies," Computer Graphics, vol. 22, pp. 299-308, 1988.

[27] B. Mirtich and J. Canny, "Impulse-based Simulation of Rigid Bodies," Symposium on Interactive 3D Graphics, 1995.

[28] E. Guendelman, Bridson, R. and Fedkiw, R.P., "Nonconvex Rigid Bodies with Stacking," ACM Transactions on Computer Graphics, vol. 22, pp. 871-879, 2003.

[29] B. Mirtich, "Impulse-based Dynamic Simulation of Rigid Body Systems," in Computer Science, vol. Ph.D. : University of California at Berkeley, 1996, pp. 246.

[30] D. Baraff, "Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies," Computer Graphics, vol. 23, pp. 223-232, 1989.

[31] D. a. W. Baraff, A., "Physically Based Modeling: Principles and Practice (Online Siggraph 97 Cource Note), www-2.cs.cmu.edu/~baraff/sigcourse/," 1997.

[32] D. Baraff, "Curved Surfaces and Coherence for Non-penetrating Rigid Body Simulation," Computer Graphics, vol. 24, pp. 19-28, 1990.

[33] R. Light, and Gossard, D., "Modification of Geometric Models through Variational Geometry," Computer Aided Design, vol. 14, pp. 209-214, 1982.

[34] I. Fudos, "Constraint Solving for Computer Aided Design," Computer Sciences, vol. Ph.D.: Purdue University, 1995, pp. 107.

[35] T. Fernando, Murray, N., Tan, K., and Wilmalaratne, P., "Software Architecture for a Constraint-Based Virtual Environment," presented at Proceedings of the ACM symposium on Virtual Reality Software and Technology, London, UK, 1999.

[36] J. C. Owen, "Algebraic Solution for Geometry from Dimensional Constraints," presented at ACM Symposium Foundations of Solid Modeling, Austin, TX, 1991.

[37] I. Fudos, and Hoffman, C. M., "A Graph Constructive Approach to Solving System of geometric Constraints," ACM Transactions on Graphics, vol. 16, pp. 179-216, 1997.

[38] A. Verroust, Schonek, F., and Roller, D., "Rule-Oriented method for Parameterized Computer-Aided Design," Computer Aided Design, vol. 24, pp. 531-540, 1992.

[39] G. Sunde, "Specification of Shape by Dimensions and Other Geometric Constraints.," presented at Geometric Modeling for CAD Applications, North Holland IFIP, 1988.

[40] H. A. H. Suzuki, and Kimura, F., "Variation of Geometries Based on a Geometric-Reasoning Method," Computers and Graphics, vol. 14, pp. 211-224, 1990.

[41] W. Bouma, Fudos, I., Hoffman, C. M., Cai, J., and Paige, R., "A Geometric Constrain Solver," Computer Aided Design, vol. 27, pp. 487-501, 1995.

[42] I. Fudos, and Hoffman, C. M., "Correctness Proof of a Geometric Constraint Solver," International Journal of Computational Geometry & Applications, vol. 6, pp. 405-420, 1995.

[43] Y. Wang, Jayaram, U., Jayaram, S., and Shaikh, I., "Methods and Algorithms For Constraint Based Virtual Assembly," Virtual Reality, vol. 6, pp. 229-243, 2003.

[44] R. Kuehne, and Oliver, J., "A Virtual Environment for Interactive Assembly Planning and Evaluation," presented at Proceedings of ASME Design Automation Conference, Boston, MA., USA, 1995.

[45] E. Pere, Langrana, N., Gomez, D., and Burdea, G., "Virtual Mechanical Assembly on a PC-Based System," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC1996/DFM-1306), Irvine, CA, 1996.

[46] N. Ye, 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, vol. 29, pp. 546 - 555, 1999.

[47] C. Cruz-Neira, Sandin, D., and DeFanti, T., "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," Proceedings of SIGGRAPH 93, pp. 135-142, 1993.

[48] C. Cruz-Neira, Sandin, D.J., DeFanti, T.A., Kenyon, R., and Hart, J.C., "The CAVE, Audio Visual Experience Automatic Virtual Environment," Communications of the ACM, pp. 64-72, 1992.

[49] H. J. Bullinger, Richer, M. and Seidel, K. A., "Virtual Assembly Planning," Human Factors and Ergonomics in Manufacturing, vol. 10, pp. 331-341, 2000.

[50] A. G. de Sa, Zachmann, G., "Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes," Computers and Graphics, vol. 23, pp. 189-403, 1999.

[51] L. Marcelino, Murray, N., and, Fernando, T., "A Constraint Manager to Support Virtual Maintainability," Computers & Graphics, vol. 27, pp. 19 - 26, 2003.

[52] S. Jayaram, Jayaram, U., Wang, Y., Tirumali, H., Lyons, K. and, Hart, P., "VADE: A Virtual Assembly Design Environment," Computer Graphics and Applications, vol. 19, pp. 44-50, 1999.

[53] U. Jayaram, Tirumali, H. and, Jayaram, S., "A Tool/Part/Human Interaction Model for Assembly in Virtual Environments," presented at ASME Design Engineering Technical Conferences 2000 (DETC 2000/CIE-14584), Baltimore, MD., 2000.

[54] F. Taylor, Jayaram, S. and, Jayaram, U., "Functionality to Facilitate Assembly of Heavy Machines in a Virtual Environment," presented at ASME Design

Engineering Technical Conferences (DETC 2000/CIE-14590), Baltimore, MD., 2000.

[55] S. Jayaram, Jayaram, U., Wang, Y., and Lyons, K., "CORBA-based Collaboration in a Virtual Assembly Design Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2000/CIE-14585), Baltimore, MD., 2000.

[56] Y. Wang, Jayaram, S., Jayaram, U., and Lyons, K., "Physically Based Modeling in Virtual Assembly," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/CIE-21259), Pittsburg, PA, 2001.

[57] M. Czernuszenko, Pape, D., Sandin, D., DeFanti, T., Dawe, G. L., and Brown, M. D., "ImmersaDesk and Infinity Wall Projection-Based Virtual Reality Displays," Computer Graphics, vol. 31, pp. 46-49, 1997.

[58] I. Shaikh, Jayaram, U., Jayaram, S., and Palmer, C., "Participatory Ergonomics Using VR Integrated with Analysis Tools," presented at 2004 Winter Simulation Conference, Washington D.C., 2004.

[59] U. Jayaram, 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, vol. 57, pp. 283-296, 2006.

[60] H. Wan, Gao, S., Peng, Q., Dai, G and Zhang, F., "MIVAS: A Multi-Modal Immersive Virtual Assembly System," presented at ASME Design Engineering

Technical Conferences and Computers and Information in Engineering Conference (DETC 2004/CIE-57660), Salt Lake City, UT., 2004.

[61] X. Chen, Xu, N., and Li, Y., "A Virtual Environment for Collaborative Assembly," presented at Second International Conference on Embedded Software and Systems (ICESS'05), 2005.

[62] Z. Liu, and Tan, J., "Virtual Assembly and Tolerance Analysis for Collaborative Design," presented at 9th International Conference on Computer Supported Cooperative Work in Design 2005, Coventry, United Kingdom, 2005.

[63] Z. Liu, and Tan, J., "Constraint Behavior Manipulation for Interactive Assembly in a Virtual Environment," International Journal of Advanced Manufacturing Technology, vol. 32, pp. 797-810, 2007.

[64] Q. H. Wang, Li, J. R., and Gong, H. Q, "A CAD-linked Virtual Assembly Environment," International Journal of Production Research, vol. 44, pp. 467-486, 2006.

[65] R. Yang, Wu, D., Fax, X., and Yan, J., "Research on Constraint-Based Virtual Assembly Technologies," Frontiers of Mechanical Engineering in China, vol. 2, pp. 243-249, 2007.

[66] R. Gupta, and Zeltzer, D., "Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments," presented at Proceedings of ASME Computers in Engineering Conference and the Engineering Database Symposium, Boston,MA., 1995.

[67] R. Gupta, Whitney, D., and Zeltzer, D., "Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments," Computer Aided Design (Special issue on VR in CAD), vol. 29, pp. 585-597, 1997.

[68] C. E. Kim, 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, vol. 4, pp. 83-90, 2004.

[69] T. C. Johnson, and, Vance, J. M., "The Use of the Voxmap Pointshell Method of Collision Detection in Virtual Assembly Methods Planning," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/DAC-21137), Pittsburgh,PA., 2001.

[70] "Fifth Dimension Technologies (http://5dt.com/)."

[71] C. Just, A. Bierbaum, A. Baker, and, C. Cruz-Neira, "VR Juggler: A Framework for Virtual Reality Development," 2nd Immersive Projection Technology Workshop (IPT98) CD-ROM. Ames, IA., 1998.

[72] C. E. Kim, and Vance, J.M., "Development of a Networked Haptic Environment in VR to Facilitate Collaborative Design Using Voxmap Pointshell (VPS) Software," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2004/CIE-57648), Salt Lake City, UT., 2004.

[73] A. S. Coutee, McDermott, S. D., and Bras, B., "A Haptic Assembly and Disassembly Simulation Environment and Associated Computational Load Optimization Techniques," ASME Journal of Computing & Information Science in Engineering, vol. 1, pp. 113-122, 2001.

[74] A. S. Coutee, and, Bras, B., "Collision Detection for Virtual Objects in a Haptic Assembly and Disassembly Simulation Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2002/CIE-34385), Montreal,Canada., 2002.

[75] "Sensable Technologies (http://www.sensable.com/)."

[76] B. Fröhlich, Tramberend, H., Beers, A., Agarawala, M. and Baraff, D., "Physically-Based Modeling on the Responsive Workbench," presented at IEEE Virtual Reality Conference, 2000.

[77] D. Baraff, "Interactive Simulation of Solid Rigid Bodies," Computer Graphics and Applications, vol. 15, pp. 63-75, 1995.

[78] W. Krüger, and Fröhlich. B., "The Responsive Workbench," Computer Graphics and Applications, vol. 14, pp. 12-15, 1994.

[79] A. Seth, Su, H. J., and Vance, J. M., "A Desktop Networked Haptic VR Interface for Mechanical Assembly," presented at ASME International Mechanical Engineering Congress & Exposition (IMECE2005-81873), Orlando, FL, USA., 2005.

[80] A. Seth, Su, H. J., and Vance, J. M., "SHARP: A System for Haptic Assembly & Realistic Prototyping," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2006/CIE-99476), Philadelphia, PA, USA, 2006.

[81] "Barco Baron (http://www.barco.com/entertainment/en/products/product.asp?element=1192)."

[82] S. a. Z.-C. Garbaya, U., "The Affect of Contact Force Sensations on User Performance in Virtual Assembly Tasks," Virtual Reality, 2007.

[83] Y. Zhang, Sotudeh, R., and Fernando, T., "The Use of Visual and Auditory Feedback for Assembly Task Performance in a Virtual Environment," presented at Proceedings of the 21st spring conference on Computer Graphics, Budmerice, Slovakia, 2005.

[84] S. Jayaram, Vance, J.M., Gadh, R., Jayaram, U. and Srinivasan, H, "Assessment of VR Technology and its Applications to Engineering Problems," Journal of Computing and Information Science in Engineering, vol. 1, pp. 72-83, 2001.

[85] Z. Zhu, Gao, S., Wan, H., Luo, Y., and Yang, W., "Grasp Identification And Multi-Finger Haptic Feedback For Virtual Assembly," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2004/CIE-57718), Salt Lake City, Utah, USA., 2004.

[86] A. Garcia-Alonso, Serrano, N., and Flaquer J., "Solving the Collision Detection Problem," IEEE Computer Graphics and Applications, vol. 14, pp. 36-43, 1994.

[87] A. Kaufman, Cohen, D., and Yagle, R., "Volume Graphics," IEEE Computer, vol. 26, pp. 51-64, 1993.

[88] P. Banerjee, "Data Interface Software for Windows PC-Compatible Virtual Reality Scene Graphs," ASME Journal of Computing and Information Science in Engineering, vol. 2, pp. 72-74, 2002.

[89] S. Jayaram, Joshi, H., Jayaram, U., Kim, Y., Kate, H., and Varoz, L., "Embedding Haptic-Enabled Tools in CAD for Training Applications," presented at ASME

Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2006-99656), Philadelphia, PA, 2006.

[90] A. S. Coutee, and, Bras, B., "An Experiment on Weight Sensation in Real and Virtual Environments," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2004-57674), Salt Lake City, Utah, USA., 2004.

[91] G. C. Burdea, "Invited Review: The Synergy Between Virtual Reality and Robotics," IEEE Transactions on Robotics and Automation, vol. v 15, pp. 400-410, 1999.

[92] S. Volkov, and Vance, J. M., "Effectiveness of Haptic Sensation for the Evaluation of Virtual Prototypes," ASME Journal of Computing and Information Sciences in Engineering, vol. 1, pp. 123-128, 2001.

[93] H. Regnbrecht, Hauber, J., Schoenfelder, R., and Maegerlein, A., "Virtual Reality Aided Assembly with Directional Vibro-Tactile Feedback," presented at Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia, Dunedin, New Zealand, 2005.

[94] H. Gurocak, Parrish, B., Jayaram, S., and Jayaram, U., "Design of a Haptic Device For Weight Sensation in Virtual Environments," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2002/CIE-34387), Montreal, Canada, 2002.

[95] G. Zachmann, and Rettig, A., "Natural and Robust Interaction in Virtual Assembly Simulation," presented at 8th ISPE International Conference on Concurrent Engineering: Research and Applications, Anaheim,CA, 2001.

[96] a. S. Massie T., K., "The PHANToM Haptic Interface: A Device for Probing Virtual Objects," presented at Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Chicago, IL, 1994.

[97] "Immersion Corporation (http://www.immersion.com/)."

[98] "Haption S.A. (http://www.haption.com/index.php?lang=eng)."

[99] "The Novint Falcon (http://www.novintfalcon.com/)."

[100] a. H. Buttolo P., B., "Pen-based Force Display for Precision Manipulation in Virtual Environments," presented at IEEE Virtual Reality Annual International Symposium, New York, NY, 1995.

[101] P. A. Millman, Stanley, M., and Colgate, J.E., "Design of a High Performance Haptic Interface to Virtual Environments," presented at IEEE Virtual Reality Annual International Symposium, Seattle, WA, 1993.

[102] M. Bouzit, Popescu, G., Burdea, G.C., and Boian, R., "The Rutgers Master II-ND Force Feedback Glove," presented at HAPTICS 2002: Haptic Interfaces for Virtual Environment and Teleoperator Systems, Orlando,FL, 2002.

[103] "JTOpen (http://www.jtopen.com/index.shtml)."

CHAPTER 3. DEVELOPMENT OF A DUAL-HANDED HAPTIC ASSEMBLY SYSTEM: SHARP

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
ASME Fellow

Submitted to: *The ASME Journal of Computing and Information Science in Engineering*

Abstract

Virtual reality (VR) technology holds promise as a virtual prototyping (VP) 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 categorizes and elaborates these challenges and then describes how SHARP (System for Haptic Assembly & Realistic Prototyping), addresses them for virtual assembly. SHARP uses physics-based modeling for simulating realistic part-to-part and hand-to-part interactions in virtual environments. A dual handed haptic interface is presented for realistic hand-part interaction. Additional modules are added to utilize the system to provide answers for maintenance issues, virtual training applications and collaborative design. 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. This has the potential to result 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, Physics-Based Modeling.

1. Introduction

VR 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 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 feedback are also important to accurately evaluate the performance of the product. VP 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 issues in the area of VA, a specific subset of VP.

VA 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” [2].

A VA system as proposed in this paper will empower future engineers with a platform which will allow them to visualize and realistically interact with design solutions during conceptual stages before physical prototypes are built. Such a system will facilitate identification of product/process design errors during early stages of product development where major changes are still feasible. Thus, such systems will reduce unforeseen problems that arise during later stages of the product life cycle, consequently saving both time and money while improving product quality [3].

2. Literature Review

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 [4-

8]. Efforts have also been directed at interacting with complex CAD models[4, 9-12]. Recently, haptic interaction has been integrated into many of these applications [9, 11, 13-16]. Haptic interaction provides force feedback to the user as an additional sensory input to aid in evaluating assembly tasks in the virtual environment.

Kuehne and Oliver [6] 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.

Gupta et al. [17, 18] developed a desktop virtual environment called VEDA (Virtual Environment for Design for Assembly) which uses physics-based modeling for modeling part behavior, dual PHANToM® haptic devices for force feedback interaction and auditory and stereo cues to augment part interaction. Coutee et al. [13, 14] developed a similar system for the desktop called HIDRA (Haptic Integrated Dis/Re-assembly Analysis). HIDRA uses the GHOST Software Toolkit from Sensable Technologies and two PHANToM® devices for simulating physical behavior of parts in a desktop virtual environment. Both VEDA and HIDRA are somewhat limited because of their inability to adequately handle complex CAD models.

Fröhlich et al. [19] developed an interactive virtual assembly system using CORIOLISTM [20] physics-based simulation package. The system used the Responsive Workbench [21] for simulating bench assembly scenarios. Various spring configurations were developed for simulating realistic interaction with virtual objects. The system did not provide any kind of haptic feedback and 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.

VADE (Virtual Assembly Design Environment) was developed by Jayaram, et al. [9, 22-24] for performing VA. This application advanced the state-of-the-art by providing the ability to directly input and interact with Pro/E CAD files. Two-handed assembly, using CyberGloves, was also developed. Constraint-based methods for modeling part behavior, demonstrated the ability for parts to slide and rotate with respect to each other. Because VADE uses constraint-based interaction methods, reaction forces are not generated when objects collide with each other and therefore, no haptic interface is available. Once close to their pre-defined positions, parts were snapped to complete the assembly task. A physics-based algorithm with limited capabilities was added to VADE for simulating realistic part behavior [25].

Bullinger et al. [26] developed an assembly planning system at Fraunhofer-Institute for Industrial Engineering (IAO) called VirtualANTHROPOS which uses ANTHROPOS, an anthropometric computer modeling software package, to place a virtual human in the assembly operation. Although, the application used Head Mounted Display (HMD) and Data Glove device for natural part interaction, it lacked in providing haptic feedback to the user.

Fernando[27] at University of Salford developed a virtual assembly application called IPSEAM (Interactive Product Simulation Environment for Assessing Assembly and Maintainability) that uses constraint based geometric modeling for interaction, however simulating part behavior is limited to lower pair joints interactions, such as constraints between surfaces, leaving out constraints involving vertices and edges. Also, there is no force modeling so haptic interaction is not present in the system.

Johnson and Vance [28] developed VEGAS (Virtual Environment for General Assembly), in 2001. Using Voxmap Point Shell (VPS) software from The Boeing Company, users could assemble full scale models with high polygon counts. Collision detection was implemented, however, the program lacked in providing part behavior simulation and haptic interaction. Kim and Vance [4, 5] further modified VEGAS to include physics-based modeling to simulate part behavior. NHE (Network Haptic Environment) was also developed by Kim and Vance [11] to facilitate collaborative assembly through the internet. The variety of computation capability of each node often caused inconsistency problem which produced unrealistic haptic forces. In addition, each network-node needed a dedicated personal computer for force rendering as well as a simulation machine for visualization using a projection screen VR system.

Wan et al. [15] developed a multimodal CAVE-based virtual assembly system called MIVAS (A Multi-Modal Immersive Virtual Assembly System) at Zhejiang University. MIVAS used constraints for simulating part behavior in a virtual environment. The application performed hand-to-part collision detection using VPS software while part-to-part collision detection was implemented using RAPID. The users could feel the size and shape of digital CAD models using the CyberGrasp haptic device from Immersion Corporation. Since Haptic feedback was only provided in griping tasks, the application lacked in providing force information when parts collided.

Liu et al. [29] 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. [30] 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.

Thus, we see that initial approaches for virtual assembly used snapping for parts positioning while more advanced applications focused on simulating assembly operations using a combination of geometric-constraint modeling and part snapping techniques. Collision detection and physics-based modeling is another approach that is used by some applications for simulating assembly. The next section describes the challenges involved in creating assembly/disassembly simulations capable of realistically simulating part-behavior and human-interactions involved in manual assembly tasks.

3. Research Challenges

During the last two decades, VR technology has evolved to a level where immersive virtual walkthroughs and data visualization simulations have become commonplace [31, 32]. Prototyping assembly/disassembly processes in virtual environments present a much more challenging problem because they require frequent, direct and intuitive human interactions with virtual product models. To simulate simple real world assembly tasks in a virtual environment, a VA system must include the following features (Table 1): graphical visualization which provides visual feedback, including depth-perception, to the worker; object behavior modeling which simulates the physical interaction (dynamics, collision and friction) between part-part and hand-part; haptic force feedback which allows the worker to feel contacts that occur between parts; and dual handed assembly. In addition, capabilities such as subassembly creation, part joining methods,

Table 1: VA Research Challenges

Features	Challenges
Graphical visualization	• High LOD product models • Low cost immersive VR systems • Support for multiple VR systems
Realistic object behavior of real CAD models	• Physics (dynamics, friction etc.) modeling of CAD models with complex topology • Real-time collision detection with high precision • Dynamic interaction between part-part and hand-part • Minimize data translation between CAD and VR
Haptic force feedback	• Haptic rendering rate • Feedback part-part collision force natural to the operator
Dual handed assembly	• Simulate natural part manipulation • Maintain physics and haptic update rate
Subassemblies/ disassemblies	• Update data structure, affect part interaction and haptic force calculation
Assembly planning	• Generate data (swept volume, assembly sequence etc.) useful for engineering practice

interaction with tools and fixtures also form core components of the simulation.

Once assembly/disassembly simulation is possible, additional capabilities are required to utilize this power for maintainability, training and collaboration purposes. Such capabilities include generating swept-volumes, recording assembly sequences and task timings, and networked VA environments, among others.

Several challenges exist in addressing the aforementioned requirements using current technology and computation power. Prominent challenges in this field are classified into four categories and elaborated below.

3.1 Graphic Visualization

Graphical visualization is the first important feature of a VA system. Tasks such as part picking and placement require the users to understand complex 3D spatial relationships among CAD models. Stereo visualization and high level-of-detail product models are critical to provide an accurate representation of the real world assembly scenarios. Most VR applications use scene-graphs (Openscenegraph, OpenSG etc.) for visualization which mostly require simplified polygonal geometry to ensure interactive frame rates. Translating existing parametric CAD data presents problems like “excessive number of polygons and number of objects that are created” [33]. The problem becomes even more challenging when incorporating pre-existing material properties and texture maps in these optimized model representations. Direct and lossless transfer of CAD data from CAD to VR systems is still a challenge. In addition, designing a VA system which can support multiple VR configurations that are available today (from low-cost desktop, single wall configurations to fully immersive CAVE environments) requires designing different interaction paradigms to best suit the VR system at hand.

3.2 Collision Detection

Another critical challenge in creating VA simulations is accurately modeling physical behavior of parts. Collision detection algorithms are frequently used for part selection and preventing part interpenetration during an assembly operation. Mechanical assembly scenarios demand accurate collision detection among arbitrarily complex (non-convex) CAD geometry. In VA simulations where real-time update rates are critical, performing fast and accurate collision detection among dynamic objects is a challenging problem. . A comprehensive review of collision detection algorithms can be found in [34, 35] and a taxonomy of collision detection approaches can be found in [36]. Although collision detection prevents parts from interpenetration, a system with only collision detection, does not provide any help to the user on how to change the position and orientation of parts to align them for facilitating the assembly operation[37].

3.3 Physics-Based Modeling

Once collisions are detected in the environment, physics-based modeling algorithms are needed to compute the subsequent part trajectories to simulate realistic physical behavior among CAD models. Physics-based algorithms simulate forces acting on bodies along with their physical properties to model realistic behavior. Such algorithms solve equations of motion of objects at each time step based on forces and torques that act upon the objects. Various methods for performing physics-based modeling have been proposed in the past [20, 38-43]. All these algorithms are have different limitations associated with them such as modeling accuracy, handling stable and simultaneous contacts, large computation time when many contacts occur simultaneously and system instabilities leading to stiff equations which are numerically intractable[44]. Approximated model representations are generally used for maintaining interactive update rates. Due to such problems, very few VA applications have attempted to model physical constraints among parts to perform assembly [4, 16, 19, 45].

3.4 Haptic Rendering

Another important aspect for VA systems is providing the user with haptic force cues allowing him to feel tactile/collision forces that are calculated using physics-based modeling algorithms. Such force cues supplement the visual and auditory cues and thus create an improved sense of presence in the virtual environment[46]. Research has shown that addition of force feedback to virtual environments increases task efficiency times[47, 48]. Especially in assembly task, haptic force can help a designer feel and better understand the geometry of virtual objects. Haptic devices require a high update rate (~1000Hz) to guarantee force continuity. Hence, the real challenge is to maintain such high update rates for the physics-modeling computations especially when interacting with large and complex CAD datasets. Further, handling multiple haptic devices simultaneously makes the problem even more complicated. Providing the user with both tactile and collision forces simultaneously is a challenge that is yet to be addressed completely.

Many of the challenges mentioned above are expected to be solved as increasing computing power and more effective algorithms (collision detection and real-time physics computation) become available in the future.

4. Motivation

The focus of the work presented in this paper is to create a system that can address the challenges outlined above and provide a successful solution to the VA problem. Once successful, the VA capability will provide the foundation for many useful virtual environments including virtual process planning, task timing, workstation layout, tooling design and integration of the immersive virtual environment with interactive discrete event programming. In addition, the results of this research will support further development of immersive off-line training, maintenance and serviceability prototyping.

Our intent is to develop and evaluate a system that spans various levels of VR hardware from desktop to full immersion in order to explore how all of these different VR interfaces might be used together to improve the design process. In this paper we present SHARP, System for Haptic Assembly & Realistic Prototyping, the newest virtual assembly system developed at Iowa State University. The following section describes the system configuration and methodology used for assembly/disassembly capability in SHARP. Next, the paper will describe additional components which expand SHARP’s capabilities to address problems related to maintainability, training and collaborative analysis in virtual environments. SHARP takes advantage of previous knowledge [9, 11, 13-16, 49] and expands the functionality of virtual assembly to include dual-handed haptics, swept volume representation, subassembly modeling and realistic part behavior.

5. SHARP: A System for Haptic Assembly & Realistic Prototyping

5.1 VR Software & Hardware Implementation

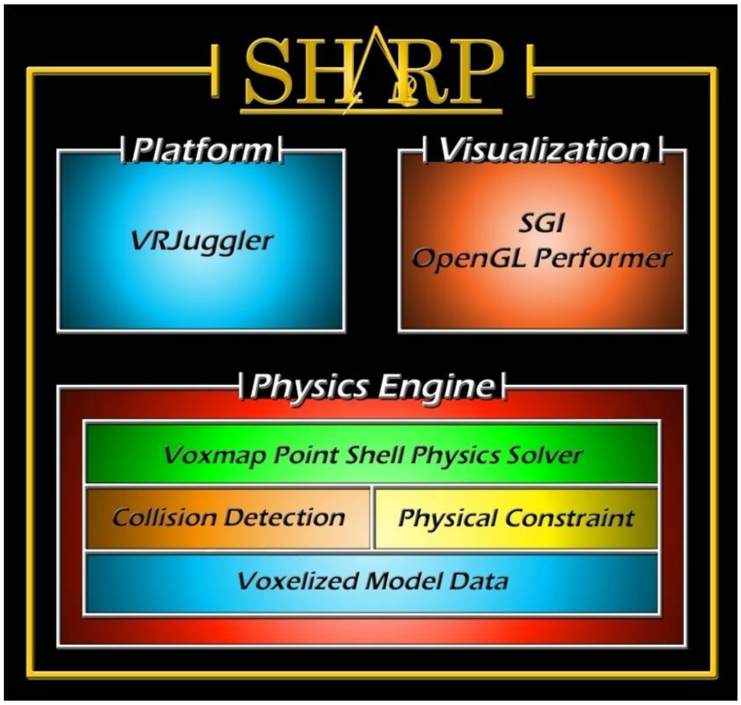


Figure 1: System Components

SHARP uses various open-source and commercially available software toolkits (Figure 1) which provide different functionalities to the system. C++ is chosen as the programming language and the open-source VRJuggler software toolkit is used for controlling the virtual environment (www.vrjuggler.org). VRJuggler provides a platform for VR application development and allows a user to run a single application on different VR systems by changing a configuration file [50]. The VRJuggler Portable Runtime (VaPoR) library provides an operating system abstraction layer that simplifies the process of creating cross-platform software.

VPS software [39] from The Boeing Company is used for collision detection and physics-based modeling. 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[51].

OpenGL Performer scene graph library from SGI is used for graphical visualization. Using VR Juggler as the platform and C++ as the programming language, the application currently runs on Windows, Linux and Irix platforms. For communication with the haptic devices, Open Haptics Toolkit from Sensable Technologies is used on Windows and Linux platforms.

Figure 2: Low-Cost VR Setup (User performing assembly with PHANToM® Omni devices while viewing parts in stereo using LCD shutter glasses and an emitter.

SHARP system could be ported to a wide variety of VR systems from single-pipe display systems such as head-mounted displays, single projection walls, and projection benches to multi-pipe stereo projection environments such as CAVE. The main application runs with the haptic device hooked up to a Windows or Linux workstation.

Figure 2 shows a low-cost hardware configuration of the system. The system is tested on Windows and Linux workstations. The workstations consist of dual 3.6 Giga Hz Intel Xeon processors with 3GB RAM and PCI Express Nvidia Quadro 4400 graphics card with 512 MB graphics memory. Active quad-buffered stereo and Crystal Eyes shutter glasses from Stereographics Corporation provide stereo viewing and PHANToM® haptic devices provide force feedback (Figure 3).

The multi-pipe stereo projection environment at VRAC is a 10 ft. x 10 ft. x 10 ft. room equipped with 6 rear projection surfaces, which serve as the walls, ceiling and floor. The system is the highest resolution CAVE in the world with hundred million pixels, as of today. The users wear stereo shutter glasses which are synchronized with the computer display to alternate the left and right eye views at a rate of 96 Hz in order to produce stereo images. An ultrasonic tracking system tracks the user's head, hand, and arm position. A 96-processor Hewlett-Packard cluster supplies the computational power and feeds images to 24 Sony digital cinema projectors to create a highly detailed virtual environment.

Figure 3: PHANToM® Desktop, PHANToM® 1.5, PHANToM® 3.0 and PHANToM® Omni, by SensAble Technologies.

5.2 Model Preprocessing and Representation

Seamless integration of VA applications into the design process requires frequent and efficient data exchange between CAD and VA systems. Several previous VA applications required specialized CAD toolkits to access proprietary CAD data necessary for simulating assembly [9, 15, 29]. Such requirements limited their impact as they could only simulate assemblies made in a specific CAD system. It also resulted in large preparation times for every assembly scenario that was imported.

SHARP system design supports direct data transfer from any standard CAD software to the virtual environment. For every model in the scene the system uses graphic model representation for visualization and haptic model representation for performing collision detection and physics-based modeling. Parametric data from CAD systems is tessellated and exported into standard file formats.

Graphics: For graphic model representation, (Figure 4) *.wrl, *.iv, *.3ds, *.pfb and several other generic CAD formats (which consist of high LOD tri-mesh data along with material properties) can be used. These files are used to construct a scene graph structure for model visualization. Every model node is assigned a transformation matrix which guides its position and orientation in the graphics world.

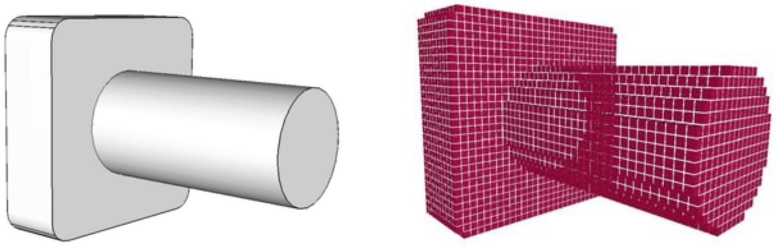


Figure 4: Graphic & Voxel model representations

Physics: For physics computations, a standard .stl file format is used. The *.stl file is parsed and the triangle and normal information is loaded into a data structure. During the voxelization step, the set of triangular polygons read from the file are converted to the VPS spatial representation called voxmap. VPS, a pair-wise collision detection algorithm detects collision between object pairs. Physical properties such as such as mass, center of mass, and moment of inertia for each CAD model are then calculated by the system completing the system initialization process.

5.3 Simulation Loops

There are four major simulation loops namely graphics, hand collision, physics and haptics loop in SHARP (Figure 5). The graphics loop is responsible for updating the scene graph model positions and handling all inputs from mouse, keyboard or wand. The hand collision loop updates the hand model position and orientation and also checks for hand/part collisions. The physics loop performs all computations for collision detection, calculates all reaction forces and computes the final position matrices for the dynamic objects at every frame. The haptic device communication loop reads the stylus position data and switch state from the haptic devices and sends the computed collision force back to the devices. In order to maintain high update rates, the physics-loop is assigned the highest priority among application threads.

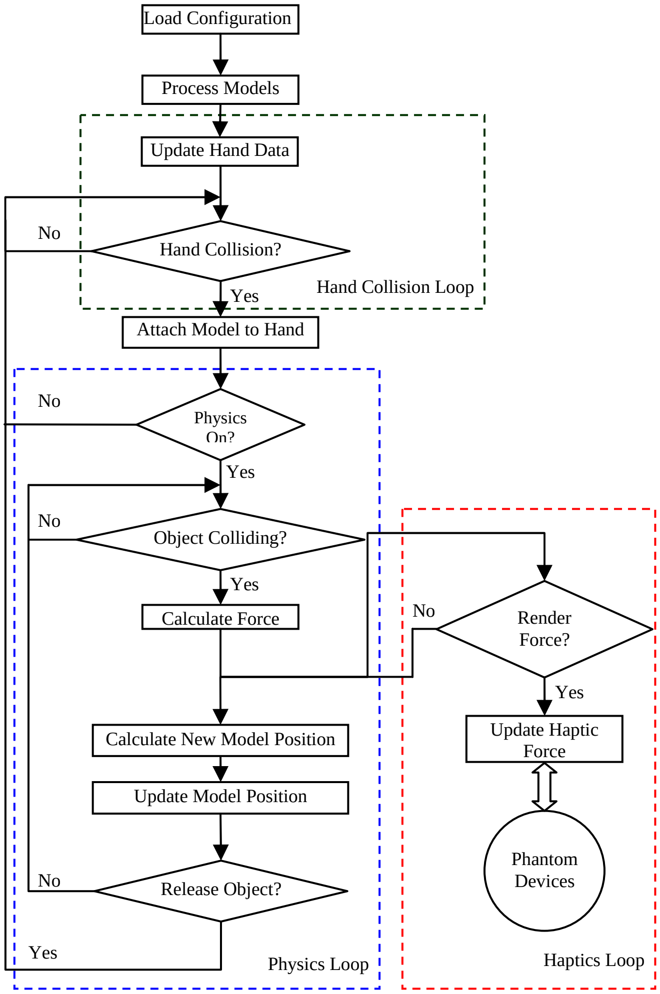


Figure 5: Simulation Loops in SHARP

5.4 Realistic Object Behavior

When developing a virtual environment which supports interactive manipulation and assembly of complex CAD objects, the greatest challenge in 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. A coarsely defined approximate model allows for fast, but inaccurate collision and force calculations. Similarly, a model which closely approximates real geometry may contain unnecessary detail which could prevent the system from maintaining interactive rates.

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 (Figure 6). The distance between the virtual hand and the manipulated object determines the spring force $\vec{\mathbf{F}}_{spring}$ and

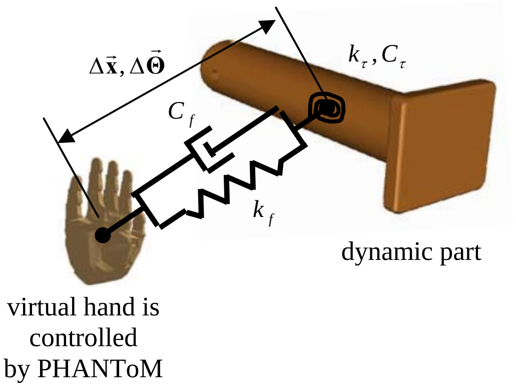


Figure 6: Virtual Spring-damper system

torque $\vec{\boldsymbol{\tau}}_{spring}(t)$ exerted on the object (Figure 7). Note that the spring force and torque also include the viscous force of the damping system. The collision force $\vec{\mathbf{F}}_i$ is proportional to the amount of penetration that one object has into the other object in the environment. The manipulated object is dynamic in nature and its motion is subject to physics laws, more specifically rigid body dynamics. That is, given the dynamic state of a rigid body at time *t*, its motion must satisfy equations (1) and (2).

$$\frac{d\vec{\mathbf{P}}(t)}{dt} = \vec{\mathbf{F}}_{total}(t), \qquad (1)$$

$$\frac{d\vec{\mathbf{L}}(t)}{dt} = \vec{\mathbf{M}}_{total}(t) \qquad (2)$$

where $\vec{\mathbf{P}}(t), \vec{\mathbf{L}}(t)$ are linear and angular momentums of the rigid body and $\vec{\mathbf{F}}_{total}(t) = \vec{\mathbf{F}}_{spring}(t) + \sum \vec{\mathbf{F}}_i + \vec{\mathbf{F}}_{brake}$ 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, the collision force applied by other objects, the damping force and the braking force. The rigid body dynamics equation is solved using the VPS function “VpsPbmEvolve”. See [39] for more details concerning the VPS method. The spring force is sent to the haptic

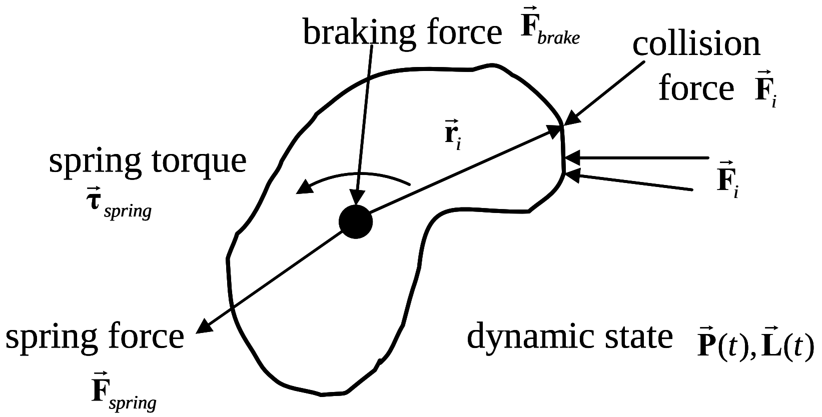


Figure 7: Force Model

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

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 to be manipulated in the environment at interactive rates. Offset layers are layers of voxels which extend beyond the surface geometry of the object to ensure that penetration does not occur between colliding parts. SHARP also allows for individual models to have different voxel sizes and number of surface offset layers for optimizing memory and reducing unnecessary computational loads that arise by voxelizing large parts with a very small voxel size.

5.5 Dual-Handed Haptic Interface

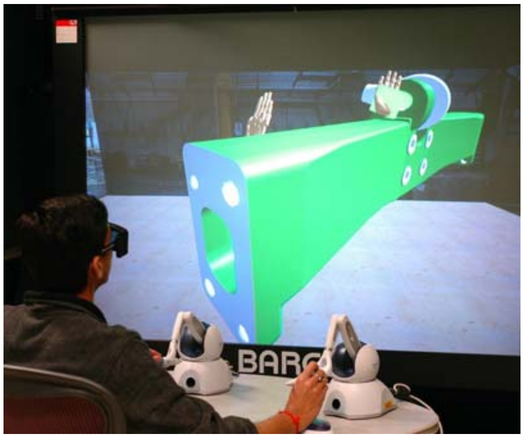


Figure 8: SHARP being used with Barco Baron and dual PHANToMs

Most VR applications require users to perform simple navigational tasks or launch preprogrammed set of events during the simulation. Wands, joysticks, and other advanced wireless controllers have been sucessful in providing us with an effective interface for such applications. Manual assembly simulations on the other hand require users to use both their hands naturally to sucessfully simulate real world tasks.

A single-handed haptic interface was initially created for SHARP which provided users with force feedback whenever collisions occurred during the simulation [49]. All physics computations were performed in a separate high priority thread to get an optimal update rate (~1000Hz) for haptic rendering.

A dual-handed simulation (Figure 8) required expanding this system to support multiple hands in the environment. A new hand model data structure has been created in SHARP which defines properties (haptic data, graphic data, hand position, control source, etc.) and states (colliding, grabbing etc.) of each hand instance present in the scene. This provides the user the capability for simultaneous part manipulation using multiple hand instances. The system can has to compute physical responses for each hand instance present in the scene during every physics frame. Thus, the system's physics update rate is halved every time a new hand instance is added. The graph in figure 9 shows the physics idle update rates for single (~1000Hz) and dual handed (~500Hz) configurations. It is important to note that the physics-update rate is dependent on the CPU speed, however the haptics loop always runs at 1000Hz.

As specified earlier, the spring force provided to the user is directly proportional to the distance between the user's hand and the manipulated object. Thus, for a very small change in distance between consecutive physics frames, the change in transmitted force will be unnoticeable to the user. The system takes advantage of this fact by continuing to render the last calculated force until new forces are calculated. We have found that this approach provides smooth forces with physics update rates as low as ~200Hz (Figure 10).

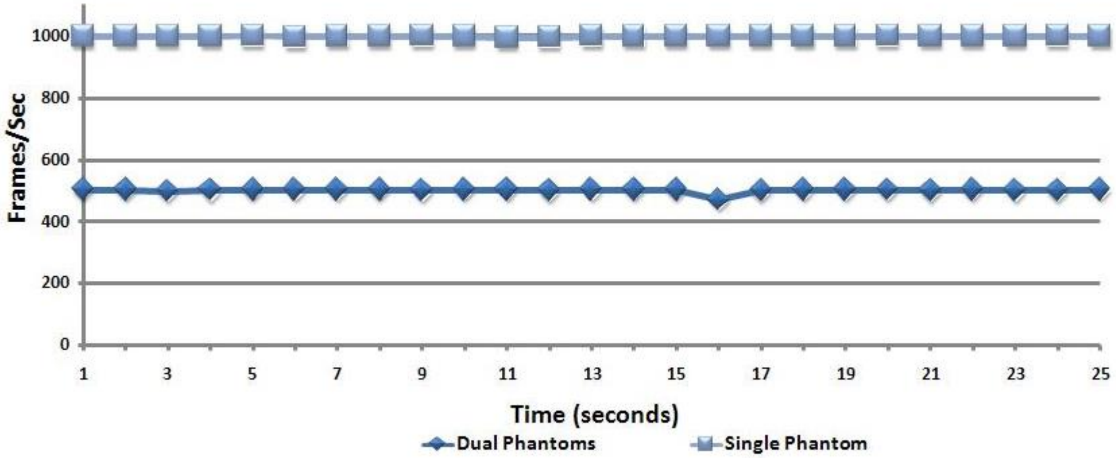


Figure 9: Physics-update rate for single and dual-handed configurations

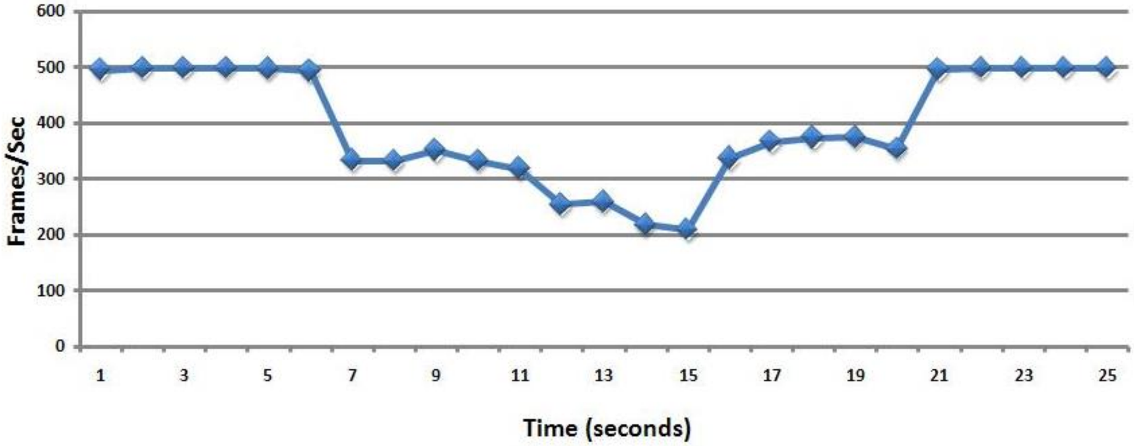


Figure 10: Physics-update rate while performing low-clearance assembly

The dual handed interface with haptic feedback provides a very efficient and intuitive interaction for virtual assembly tasks. Interacting with two hands and receiving force feedback, an operator can more realistically perform assembly tasks with the same dexterity as he/she has in the real world.

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 (Figure 11) will be as follows:

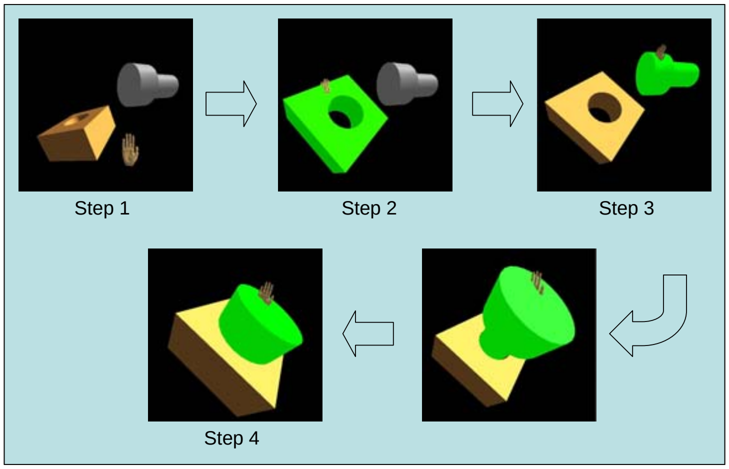


Figure 11: Assembly steps using single haptic hand (CAD models were made using

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, orienting them with respect to each other to complete assembly. Assembly steps using the dual handed haptic interface (Figure 12) will be as follows:

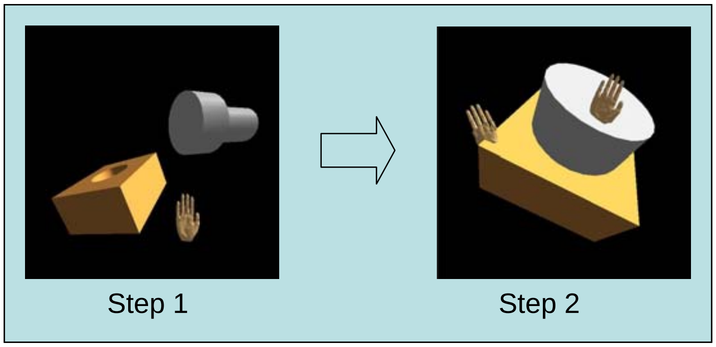


Figure 12: Assembly steps using dual handed assembly

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

Step 2: Orient them simultaneously and assemble together.

Thus a dual handed interface not only reduces the number of assembly steps but also makes the assembly simulation more realistic, by closely replicating real world interactions. SHARP loads pre-voxelized data for hand models during initialization and detect collisions between the hand models and each of the voxelized CAD models present in the environment. The system is capable of simulating scenarios of simultaneous manipulation of parts/subassemblies grabbed in each hand and performing collision detection and physics-based modeling while assembling objects.

5.6 Runtime Voxel Size Variation

Every CAD model present in the SHARP assembly environment consists of a graphic and a haptic representation. Both graphic and haptic data representations for the environment are created during the system initialization process. Haptic model representation for a model consists of voxelized model data (Figure 3) which is necessary for collision detection and physics-based modeling.

SHARP uses a configuration file (.txt) which allows the user to specify environment attributes such as number of parts, part locations, voxel size, number of hands etc. before the application is started. Thus, different voxel sizes could be specified by the user for each model based on the clearance among assembly parts.

In order to minimize memory and computation requirements, parts which do not require assembly or have larger clearances are coarsely voxelized. When assembly clearances are low, the mating parts should be voxelized with smaller voxel sizes which are optimum both for facilitating the assembly task and keeping the memory load minimum. However, in many cases, initial voxel size specified by the user does not allow parts to be assembled. Previously, if such a situation arose in the middle of an assembly process, the user had to restart the system using a reduced voxel sizes for parts.

A run-time voxel size variation module has been developed for SHARP that allows for increase/decrease in voxel size of parts while the application is running. The current haptic (voxelized) model data is deleted from the system and new voxel representation is created. The new model data is then traversed and system recalculates the physical properties for the part. All simulation loops are suspended during this time as only graphic model data is available for the scene. After initialization steps are complete, the application is resumed with the part having a new voxel size. Implementation of this feature has allowed carrying out assembly sequences without the need of shutting down and restarting the application in the middle of an assembly sequence.

6. Modules for Maintenance, Training and Collaboration

6.1 Maintenance: Swept Volumes

Modeling swept volumes is an effective way of resolving issues that may arise while servicing or inspection of complex mechanical assemblies. Questions related to accessibility, room for tooling, etc. for frequently serviced/replaced parts can be effectively answered using swept volumes during early stages of design.

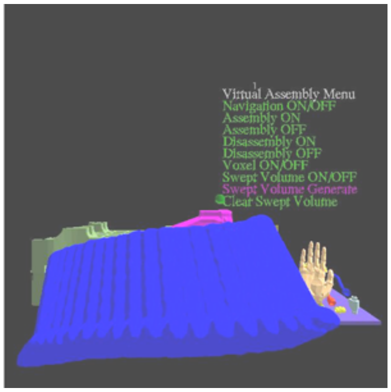


Figure 13: Illustration of Swept Volumes in SHARP

SHARP uses VPS for swept volume generation and OpenGL Performer for swept volume visualization. The positions and orientations of the model during a given time period are recorded and given to VPS for calculating the swept volume. Swept volumes are formed by a boolean union of VPS object models transformed according to each motion frame. To visualize the swept volume generated by VPS, SHARP uses a tessellation function to convert VPS swept volume data into triangle meshes which are then displayed using OpenGL Performer (Figure 13). Note that the swept volume represents the volume of the voxelized models and therefore is an approximation of the model geometry.

6.2 Virtual Training: Record and Play Module

Analyzing and evaluating different assembly sequences is one of the main requirements of a virtual assembly application. They can also be used for training assembly workers. Such virtual training tasks become more critical when the assembly environment or the assembly itself pose a hazard to the worker. Also virtual assembly applications can be used for collaborative assembly tasks where designers from different locations share the same virtual environment. All of these requirements demand a set of assembly steps to be displayed and analyzed several times in a virtual environment. To accomplish this, a record and play module has been developed and integrated into SHARP. This allows the users to record a desired set of assembly steps which can then be played for demonstration or training purposes.

6.3 Virtual Training: 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

SHARP supports creation of subassemblies which can allow training simulations of more comprehensive manual assembly processes. Performing dynamic assembly/disassembly operations in virtual environments requires modification of the underlying scene graph, or object hierarchy tree to maintain consistent object motions. When two or more parts are assembled together, their VPS data and display nodes are 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. . The user then has to inform the application that these parts should be treated as a single object in the virtual world. 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 corresponds to that of the respective physics model in the environment. This also requires storing all properties and current states of models that are assembled together which can later be used for restoring the individual models to their current state if the subassembly is disassembled. However, 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” has therefore been designed to accomplish the subassembly process. (Figure 14) 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 subassembled. 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 physics-based modeling and collision calculations.

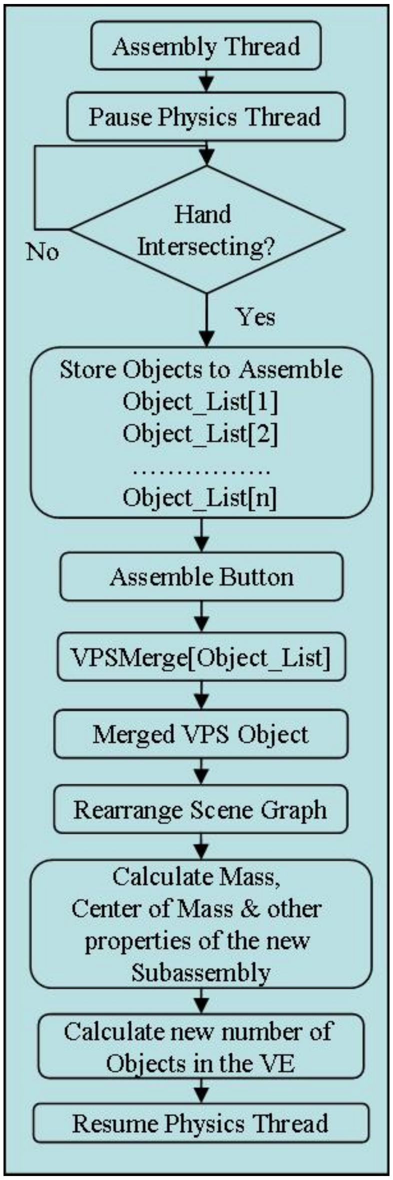


Figure 14: Operations performed by the Assembly thread

The OpenGL Performer scene graph structure is changed and parts to be subassembled are removed from the root node and attached to the part node to which they are subassembled. Figures 15 and 16 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

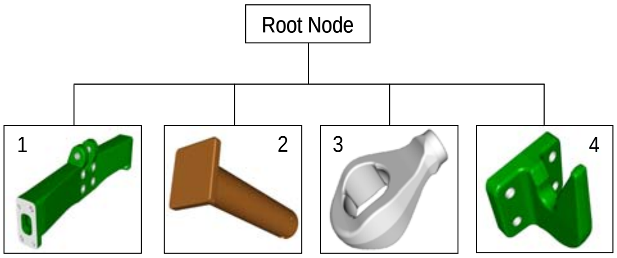


Figure 15: Data structure before assembly

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. This completes the subassembly process. Assemblies can be disassembled using similar techniques.

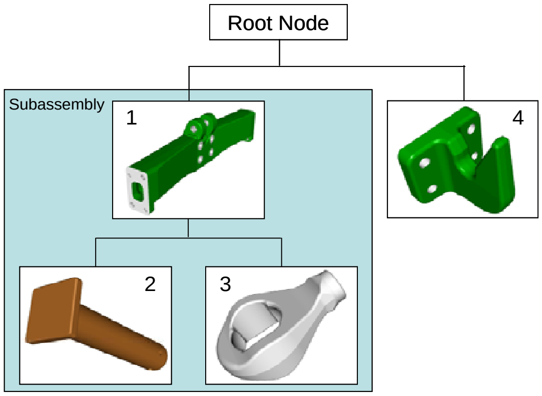


Figure 16: Data structure after assembling Part 1, 2 and 3

6.4 Collaboration: Networked Assembly Demonstration

Consulting with other engineers and taking feedback from people in the organization, like shop floor workers etc. is an important part of an assembly sequence design process. To fulfill such requirements SHARP provides a network module that can be activated selectively. When running in the network configuration, the application (running at the workstation with haptic feedback) acts as a server and communicates with the client application running at a geographically dispersed location through a non dedicated network channel. Figure 17 shows operations performed by the server and client

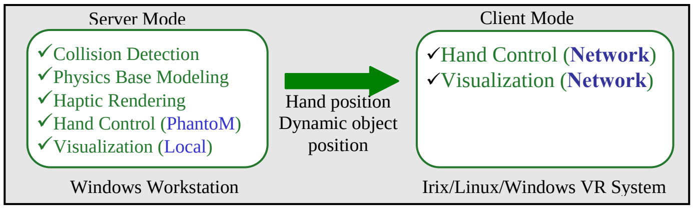


Figure 17: Network Architecture

modules of the application. The server module of the application runs in full mode, i.e. it loads graphic and haptic models and performs collision detection and physics-based modeling, calculates the model's final position and sends the hand and dynamic model's position information to the client.

The client module runs in a reduced capability mode (for demonstration purposes) where the system only loads the graphics world. All haptic computations are performed at the server and their positions matrices are transferred to the client over the network using TCP socket programming. Thus, network module allows an engineer to work on his/her workstation and assemble complex CAD models using haptic and visual feedback while the same assembly sequence is observed and analyzed by the client users in a CAVE, Power Wall or a Desktop system at another location. Figure 18 shows the client module of the application running in the multi-pipe stereo projection environment at VRAC.

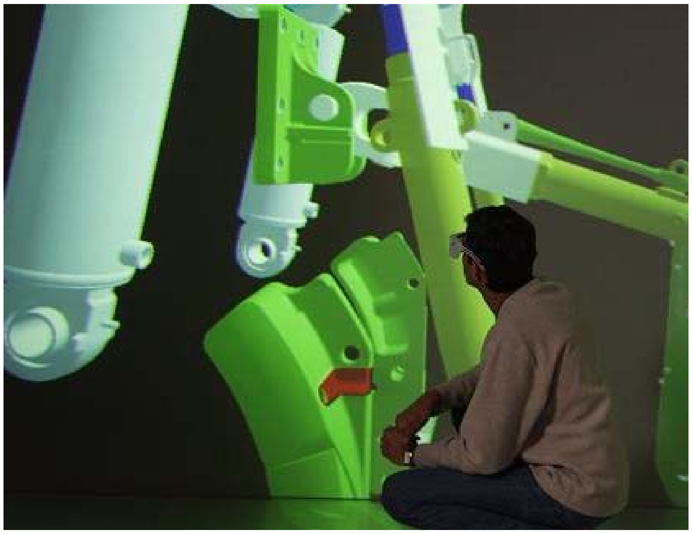


Figure 18: Assembly Demonstration in C6 at VRAC

7. Optimal Voxel Size Test

As mentioned earlier, the performance of application depends on voxel size chosen for each mating part. For mating parts with low clearance, a smaller voxel size is necessary. However, the smaller the voxel size is chosen, the more number of voxels are present in the part. The graph in figure 19 shows that voxel size is exponentially proportional to the number of voxels (hence memory requirement). Since collision detection and physics computation are directly proportional to number of voxels, more computer resources including memory allocation and computation time are needed for smaller voxel size parts. Figure 20 shows two CAD parts, a pin and a block having a hole with nominal diameter of 18.75mm. Here we test our application for assembly of these two parts with three different clearances: 2.5mm, 1.4mm and 1.0mm. For each clearance case, we first fix the peg voxel size and vary the pin voxel size from 0.20mm to 2.5mm. The lower limit chosen is 0.20mm due to the limitation of computer memory. The operator is not limited by trial time, and it typically takes less than 3 minutes to finish the assembly task. The results obtained from the assembly for each trial are recorded and analyzed. If the pin completely goes through the hole, the result is recorded as “yes”. If the pin goes only half way through the hole, the result recorded is “half”. For the remaining case the result recorded is “no”. All the tests are performed by the same operator.

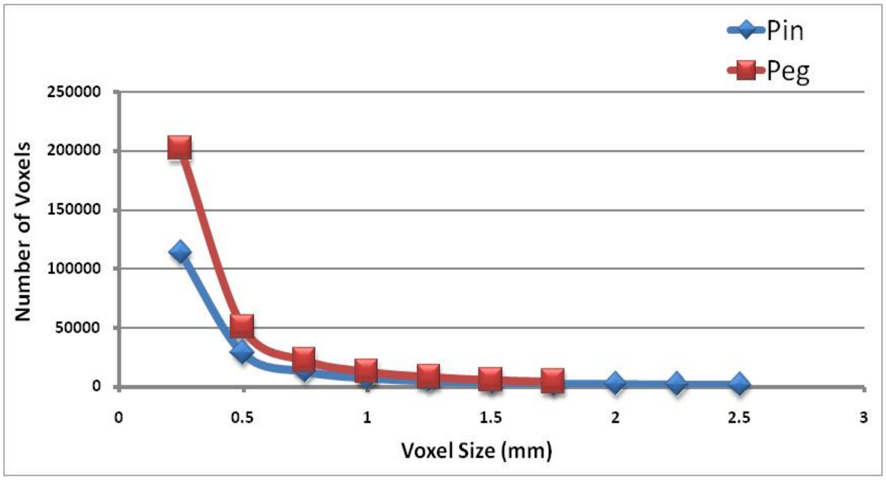


Figure 19: Number of Voxels Vs. Voxel Size

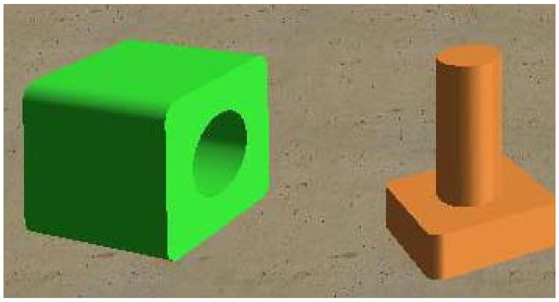


Figure 20: Peg & Hole

Table 2 shows the result of assembly trials with peg voxel size 1.5mm and mating clearance of 2.5mm. The test results indicated that smaller voxel sizes are not always the best choice. Using smaller voxel sizes results in creating a more accurate physics representations of the CAD model. However, this results in more number of pointshell–voxel interactions and an assembly which creates larger interaction forces among models. It also results in parts behaving “sticky” and also adversely affects system robustness. For the cases shown in Table 2, the optimal voxel size of the pin is [0.75, 1.75] mm. A voxel size larger than 1.75mm will block the clearance and a voxel size smaller than 0.75mm will cause the vibration of parts. In either case, the assembly task could not be accomplished.

Table 2: Test Assembly Trials (clearance=2.5mm, peg voxel size=1.5mm)

Pin Voxel Size	# of Voxels of the Pin	Result
0.25	113850	no
0.5	28416	half
0.75	12636	yes
1	7024	yes
1.25	4601	yes
1.5	3190	yes
1.75	2183	yes
2	1820	half

2.25	1360	half
2.5	1172	no

Figures 21, 22, and 23 show the optimal pin voxel sizes for clearance 2.5mm, 1.4mm and 1.0mm respectively. It can be seen here that a higher the clearance, a larger voxel size and a wider range of the voxel size can be chosen. For instance, if the peg voxel size is chosen to be 1mm, the pin voxel size range can be [0.25, 1.8] mm when the clearance is 2.5mm. However this range drops to [0.5, 0.75] mm for a clearance of 1mm. In addition, our test shows that it is not possible to assemble these two parts with clearance of 0.5mm no matter what voxel size is used.

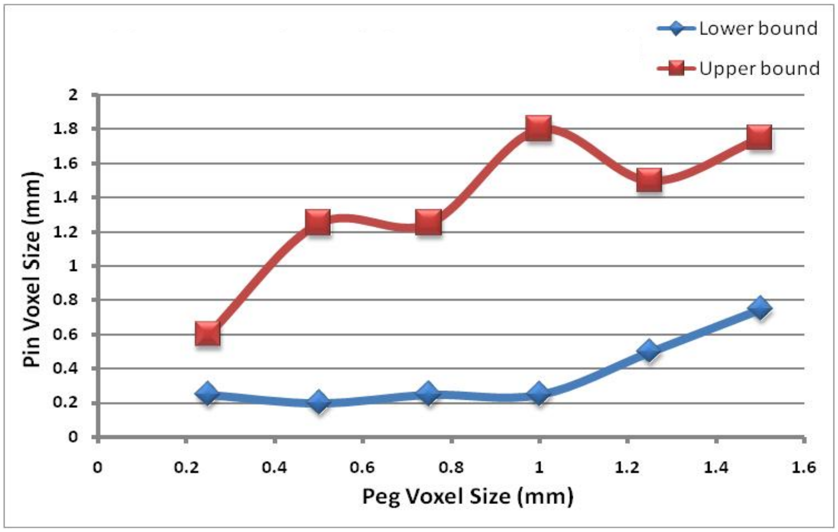


Figure 21: Feasible Pin Voxel Size (Clearance 2.5mm)

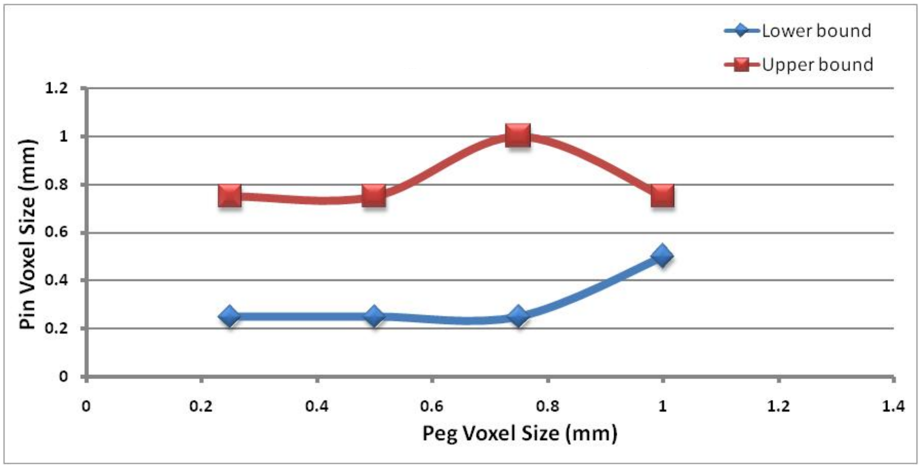


Figure 22: Feasible Pin Voxel Size (Clearance 1.4mm)

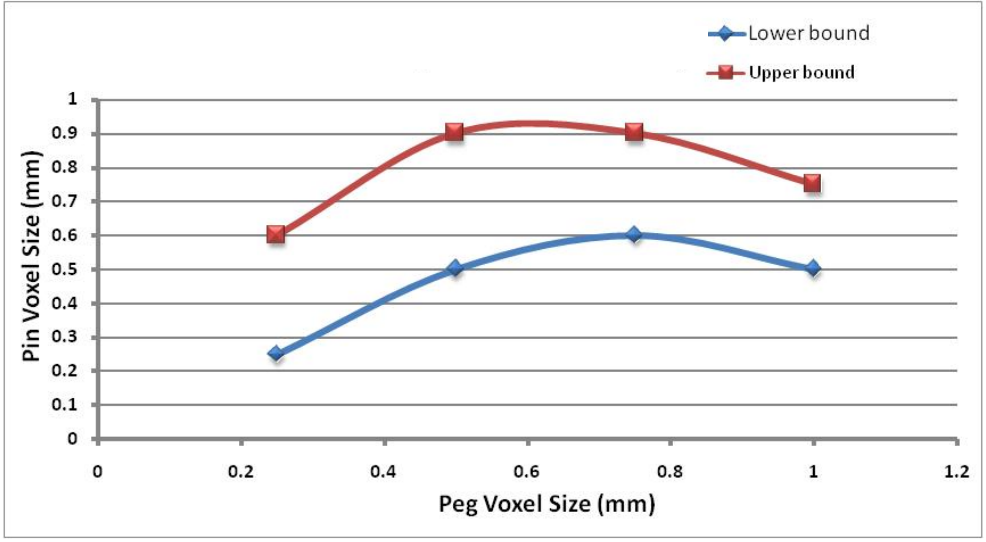


Figure 23: Feasible Pin Voxel Size (Clearance 1.0mm)

8. Conclusions & Future Work

In this paper, a platform independent application, SHARP, has been presented which uses physics-based modeling for simulating realistic part behavior and provides an intuitive dual handed PHANToM® haptic interface for mechanical assembly in an immersive VR 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 simulations can be still be improved. Physics-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, eg. no torque feedback, when objects collide. In many assembly operations, torque feedback is an important factor. Physics-based modeling also depends on the underlying haptic model to detect collisions and generate contact forces. This haptic model represents an approximation of the surface geometry and introduces dimensional error in tight fitting assembly operations. SHARP addresses this issue by providing the ability to have multiple parts with different voxel sizes and the ability to re-voxelize during run time. However, in the future, methods for collision detection and physics modeling using accurate B-Rep surface representations will be examined for more memory efficient and highly accurate collision detection and physics computations. Also, combinations of constraint-based and physics-based methods will be explored to develop an optimum interaction paradigm which can provide solutions to low clearance assembly, realistic part behavior and haptic interactions at the same time.

9. Acknowledgements

We are grateful for the technical assistance of William McNeely of the Boeing Company. This work is funded by Deere & Company. This work was performed at the Virtual Reality Applications Center at Iowa State University.

10. References

[1] G. G. Wang, "Definition and Review of Virtual Prototyping," Journal of Computing and Information Science in Engineering, vol. 2, pp. 232 - 236, 2002.

[2] A. Seth, Vance, J. M., and Oliver, J. H., "Combining Geometric Constraints with Physics Modeling for Virtual Assembly Using SHARP," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2007/CIE-34681), Las Vegas, NV, USA, 2007.

[3] J. Savall, D. Borro, J. J. Gil, and L. Matey, "Description of a Haptic System for Virtual Maintainability in Aeronautics," presented at IEEE International Conference on Intelligent Robots and Systems, Lausanne, Switzerland, 2002.

[4] C. E. Kim, and Vance, J.M., "Using Vps (Voxmap Pointshell) As The Basis For Interaction in a Virtual Assembly Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2003/CIE-48297), Chicago,IL., 2003.

[5] C. E. Kim, 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, vol. 4, pp. 83-90, 2004.

[6] R. Kuehne, and Oliver, J., "A Virtual Environment for Interactive Assembly Planning and Evaluation," presented at Proceedings of ASME Design Automation Conference, Boston, MA., USA, 1995.

[7] X. Yuan, and Sun, H., "Mechanical Assembly with Data Glove Devices," presented at IEEE 1997 Canadian Conference on Electrical and Computer Engineering, St. Johns, Nfld., Canada, 1997.

[8] S. Jayaram, Connacher, H. I., and Lyons K.W., "Virtual Assembly using Virtual Reality Techniques," Computer Aided Design, vol. 29, pp. 575-584, 1997.

[9] S. Jayaram, Jayaram, U., Wang, Y., Tirumali, H., Lyons, K. and, Hart, P., "VADE: A Virtual Assembly Design Environment," Computer Graphics and Applications, vol. 19, pp. 44-50, 1999.

[10] A. Gomes, and Zachmann, G., "Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes," Computers and Graphics, vol. 23, pp. 189-403, 1999.

[11] C. E. Kim, and Vance, J.M., "Development of a Networked Haptic Environment in VR to Facilitate Collaborative Design Using Voxmap Pointshell (VPS) Software," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2004/CIE-57648), Salt Lake City, UT., 2004.

[12] L. Marcelino, Murray, N., and, Fernando, T., "A Constraint Manager to Support Virtual Maintainability," Computers & Graphics, vol. 27, pp. 19 - 26, 2003.

[13] A. S. Coutee, 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, vol. 1, pp. 113-122, 2001.

[14] A. S. Coutee, and, Bras, B., "Collision Detection for Virtual Objects in a Haptic Assembly and Disassembly Simulation Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2002/CIE-34385), Montreal,Canada., 2002.

[15] H. Wan, Gao, S., Peng, Q., Dai, G and Zhang, F., "MIVAS: A Multi-Modal Immersive Virtual Assembly System," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2004/CIE-57660), Salt Lake City, UT., 2004.

[16] S. D. McDermott, and, Bras, B., "Development of a Haptically Enabled Dis/Re-Assembly Simulation Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC1999/CIE-9035), Las Vegas, NV., 1999.

[17] R. Gupta, and Zeltzer, D., "Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments," presented at ASME Computers in Engineering Conference and the Engineering Database Symposium, Boston,MA., 1995.

[18] R. Gupta, Whitney, D., and Zeltzer, D., "Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments," Computer Aided Design (Special issue on VR in CAD), vol. 29, pp. 585-597, 1997.

[19] B. Fröhlich, Tramberend, H., Beers, A., Agarawala, M. and Baraff, D., "Physically-Based Modeling on the Responsive Workbench," presented at IEEE Virtual Reality Conference, 2000.

[20] D. Baraff, "Interactive Simulation of Solid Rigid Bodies," Computer Graphics and Applications, vol. 15, pp. 63-75, 1995.

[21] W. Krüger, and Fröhlich. B., "The Responsive Workbench," Computer Graphics and Applications, vol. 14, pp. 12-15, 1994.

[22] S. Jayaram, Jayaram, U., Wang, Y., and Lyons, K., "CORBA-based Collaboration in a Virtual Assembly Design Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2000/CIE-14585), Baltimore, MD., 2000.

[23] U. Jayaram, Tirumali, H. and, Jayaram, S., "A Tool/Part/Human Interaction Model for Assembly in Virtual Environments," presented at ASME Design Engineering Technical Conferences 2000 (DETC 2000/CIE-14584), Baltimore, MD., 2000.

[24] F. Taylor, Jayaram, S. and, Jayaram, U., "Functionality to Facilitate Assembly of Heavy Machines in a Virtual Environment," presented at ASME Design Engineering Technical Conferences (DETC 2000/CIE-14590), Baltimore, MD., 2000.

[25] Y. Wang, Jayaram, S., Jayaram, U., and Lyons, K., "Physically Based Modeling in Virtual Assembly," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/CIE-21259), Pittsburg, PA, 2001.

[26] H. J. Bullinger, Richer, M. and Seidel, K. A., "Virtual Assembly Planning," Human Factors and Ergonomics in Manufacturing, vol. 10, pp. 331-341, 2000.

[27] T. Fernando, Marcelino, L., Wimalaratne, P. and, Tan, K., "Interactive Assembly Modeling within a CAVE Environment," Eurographics-Portuguese Chapter, pp. 43-49, 2000.

[28] T. C. Johnson, and, Vance, J. M., "The Use of the Voxmap Pointshell Method of Collision Detection in Virtual Assembly Methods Planning," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/DAC-21137), Pittsburgh, PA., 2001.

[29] Z. Liu, and Tan, J., "Virtual Assembly and Tolerance Analysis for Collaborative Design," presented at 9th International Conference on Computer Supported Cooperative Work in Design 2005, Coventry, United Kingdom, 2005.

[30] X. Chen, Xu, N., and Li, Y., "A Virtual Environment for Collaborative Assembly," presented at Second International Conference on Embedded Software and Systems (ICESS'05), 2005.

[31] S. Bryson, "Virtual Reality in Scientific Visualization," Communications of the ACM, vol. 39, pp. 62-71, 1996.

[32] J. Eddy, and Lewis, K. E., "Visualization of Multidimensional Design and Optimization Data using Cloud Visualization," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2002/DAC-34130), Montreal, Canada, 2002.

[33] P. Banerjee, "Data Interface Software for Windows PC-Compatible Virtual Reality Scene Graphs," Journal of Computing and Information Science in Engineering, vol. 2, pp. 72-74, 2002.

[34] M. Lin, and Gottaschalk, S., "Collision Detection between Geometric Models: A survey," presented at Proceedings of IMA Conference on Mathematics of Surfaces, 1998.

[35] P. Jiménez, Thomas,F., and Torras, C., "3D Collision Detection: A Survey," Computers and Graphics, vol. 25, pp. 269-285, 2001.

[36] D. Borro, Hernantes, J., Garcia-Alonso, A., and Matey, L., "Collision Problem: Characteristics for a Taxonomy," presented at Proceedings of the Ninth International Conference on Information Visualisation (IV'05), 2005.

[37] B. Frohlich, Tramberend, H., Beers, A., Agarawala, M. and Baraff, D., "Physically-Based Modeling on the Responsive Workbench," presented at IEEE Virtual Reality Conference, 2000.

[38] K. Erleben, Sporring J., Henriksen, K. and Dohlmann, H., Physics-Based Animation, First ed. Hingham, MA: Charles River Media, Inc., 2005.

[39] W. A. McNeely, Puterbaugh, K. D. and Troy, J. J., "Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling," presented at SIGGRAPH 99 Conference Proceedings, Annual Conference Series, Los Angles,CA., 1999.

[40] J. K. Hahn, "Realistic Animation of Rigid Bodies," Computer Graphics, vol. 22, pp. 299-308, 1988.

[41] B. V. Mirtich, "Impulse-based Dynamic Simulation of Rigid Body Systems," in Computer Science, vol. Ph.D.: University of California at Berkeley, 1996, pp. 246.

[42] E. Guendelman, Bridson, R. and Fedkiw, R.P., "Nonconvex Rigid Bodies with Stacking," ACM Transactions on Computer Graphics, vol. 22, pp. 871-879, 2003.

[43] D. Baraff, "Analytical Methods for Dynamic Simulation of Non-penetrating Rigid Bodies," Computer Graphics, vol. 23, pp. 223-232, 1989.

[44] A. Witkin, Gleicher, M. and Welch, W., "Interactive Dynamics," Computer Graphics, vol. 24, pp. 11-22, 1990.

[45] A. Seth, Su, H. J., and Vance, J. M., "SHARP: A System for Haptic Assembly & Realistic Prototyping," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2006/CIE-99476), Philadelphia, PA, USA, 2006.

[46] A. S. Coutee, and, Bras, B., "A Comparison of Two Collision Detection Libraries in a Haptic Simulated Environment," presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2004-57673), Salt Lake City, Utah, USA., 2004.

[47] S. A. Volkov, Vance, Judy M., "Effiectiveness of Haptic Sensation for the Evaluation of Virtual Prototypes," presented at ASME Design Engineering Technical Conference (DETC2001/DAC-21135), Pittsburgh, PA, 2001.

[48] G. C. Burdea, "Invited Review: The Synergy Between Virtual Reality and Robotics," IEEE Transactions on Robotics and Automation, vol. 15, pp. 400-410, 1999.

[49] A. Seth, Su, H. J., and Vance, J. M., "A Desktop Networked Haptic VR Interface for Mechanical Assembly," presented at ASME International Mechanical Engineering Congress & Exposition (IMECE2005-81873), Orlando, FL, USA., 2005.

[50] C. Just, A. Bierbaum, A. Baker, and, C. Cruz-Neira, "VR Juggler: A Framework for Virtual Reality Development," 2nd Immersive Projection Technology Workshop (IPT98) CD-ROM. Ames, IA., 1998.

[51] G. C. Burdea, "Haptic Feedback for Virtual Reality," presented at Virtual Reality and Prototyping Workshop, Laval, France., 1999.

CHAPTER 4. COMBINING PHYSICAL CONSTRAINTS WITH GEOMETRIC CONSTRAINT MODELING FOR VIRTUAL ASSEMBLY USING SHARP

Abhishek Seth
Department of Mechanical Engineering
2274 Howe Hall, Room 1620,
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
2274 Howe Hall, Room 1620,
Virtual Reality Applications Center
Iowa State University
Ames, IA 50011
oliver@iastate.edu
ASME Fellow

Submitted to: *The ASME Journal of Mechanical Design*

Abstract

This research combines physical constraints with constraint-based modeling 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 physical constraints with 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, Physical Constraint Simulation.

1. 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. Challenges and Related Work

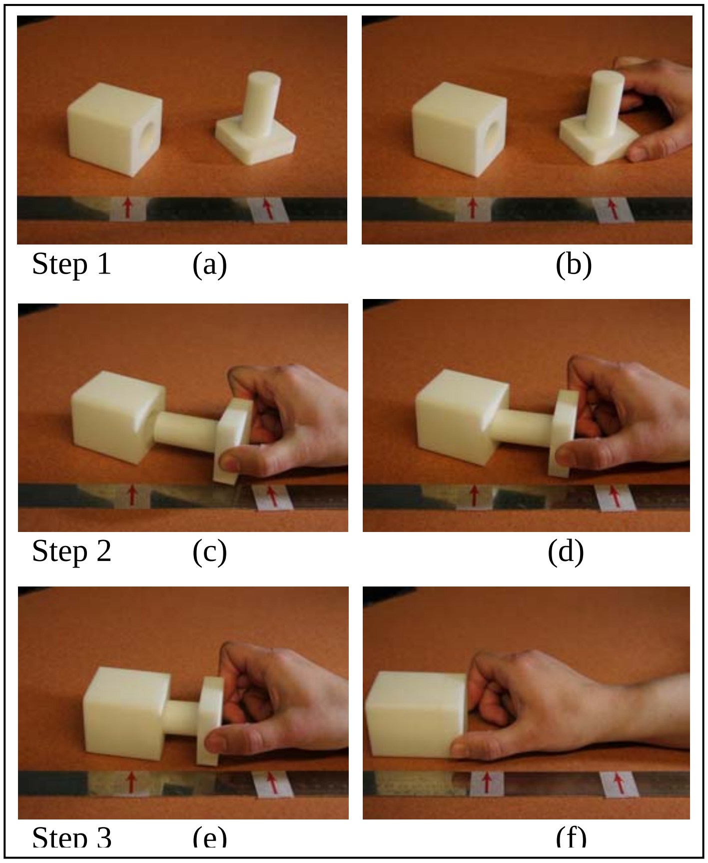
2.1 Mechanical Assembly: Human in the Loop

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.


Step 1 (a) (b)

Step 2 (c) (d)

Step 3 (e) (f)

Figure 1: Assembly sequence of pin and hole

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.

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 a 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 CORIOLISTM [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 imported 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. 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 Haptic Assembly & Realistic Prototyping [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 haptic feedback, voxel-based approximation used by VPS was not accurate enough for performing low clearance assembly.

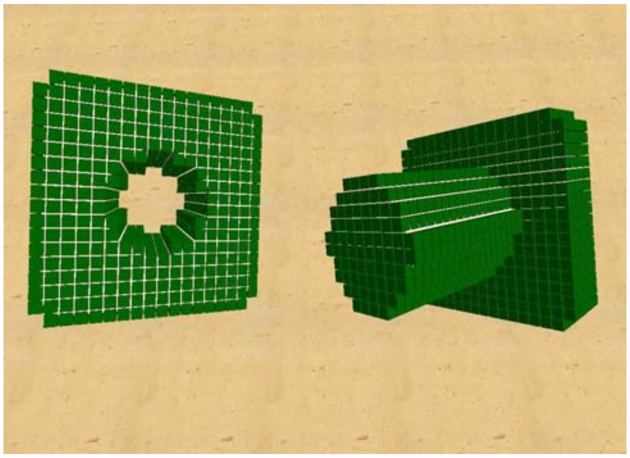


Figure 2: Pin and Hole Voxel Representations

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.

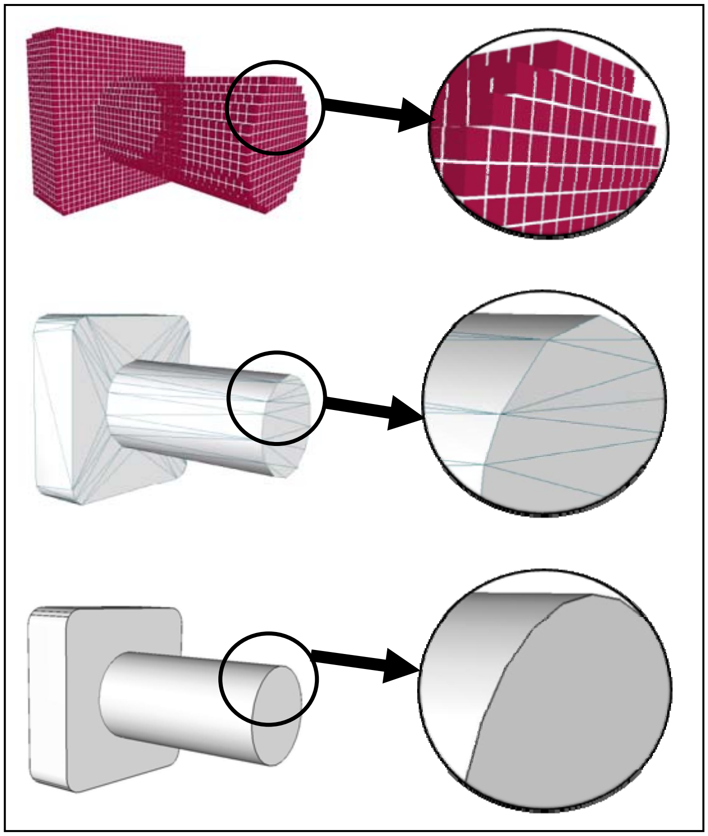


Figure 3: Voxel, Tri-mesh & B-Rep representations of a part

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 physical constraint 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 0.0001mm.

3.2 Runtime Physical/Geometric Constraint Solving in SHARP

Realistic environment behavior in SHARP is obtained by simulated physical constraints among dynamically contacting surfaces, 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) [45] 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 to the geometry only model for performing realistic physical constraint simulation.

Figure 4 shows the applications flowchart. The application first reads a configuration file which contains data about the initial assembly environment setup such as number of parts, initial positions

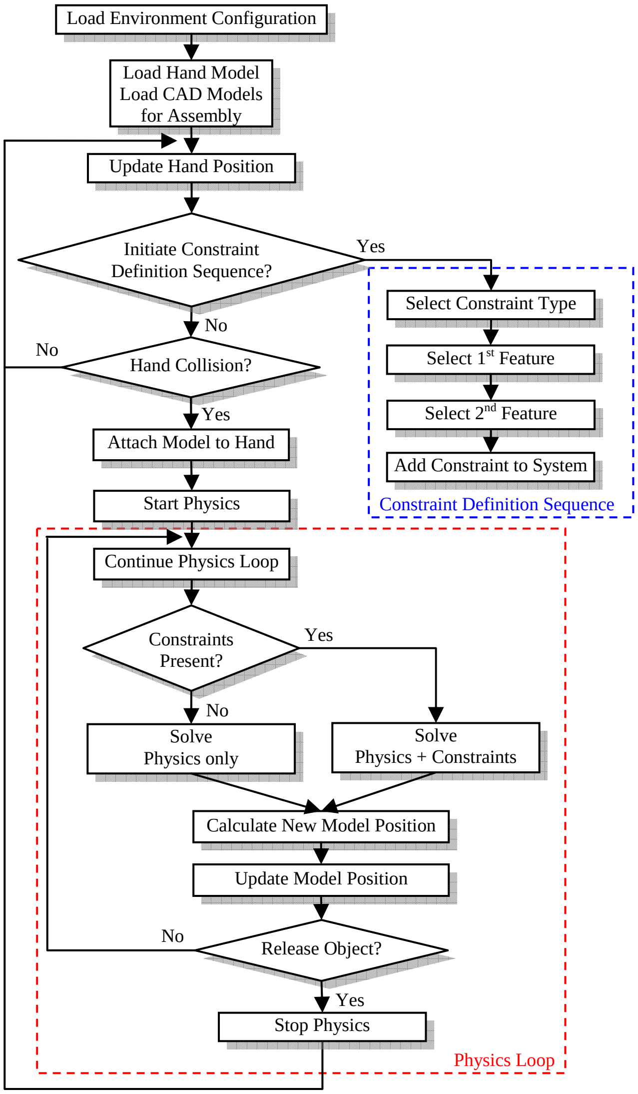


Figure 4: Application Flowchart

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 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 simulation 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 simulated physical constraints 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. Pin and Hole Assembly: Finding the Right Method

As discussed in the literature review section, several techniques (collision detection, physical constraint simulation, geometric 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 physical constraints 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.

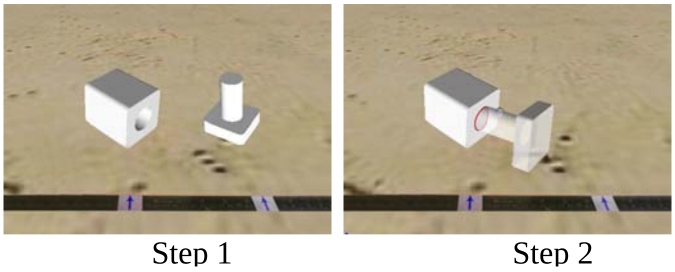


Figure 5: Assembly using collision detection only

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

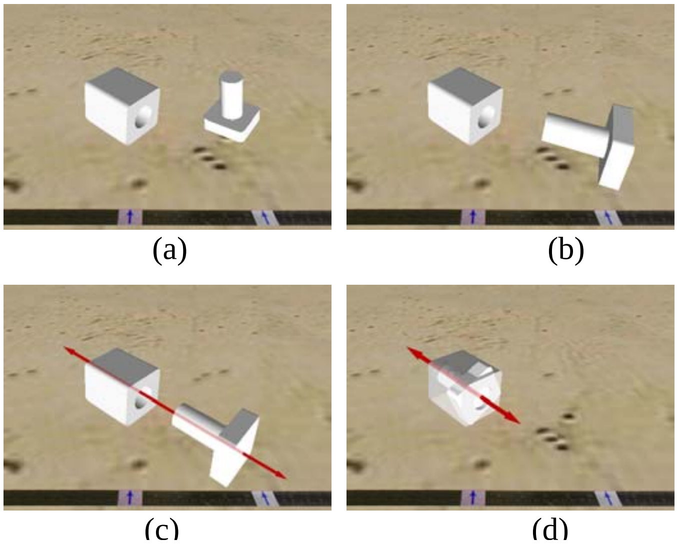


Figure 6: Assembly using Constraint Based

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 + Physical Constraint Simulation

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 the hole, physical constraints (among the colliding surfaces) facilitate in guiding the pin. Physical constraints 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 Detection + Physical Constraints + Geometric 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

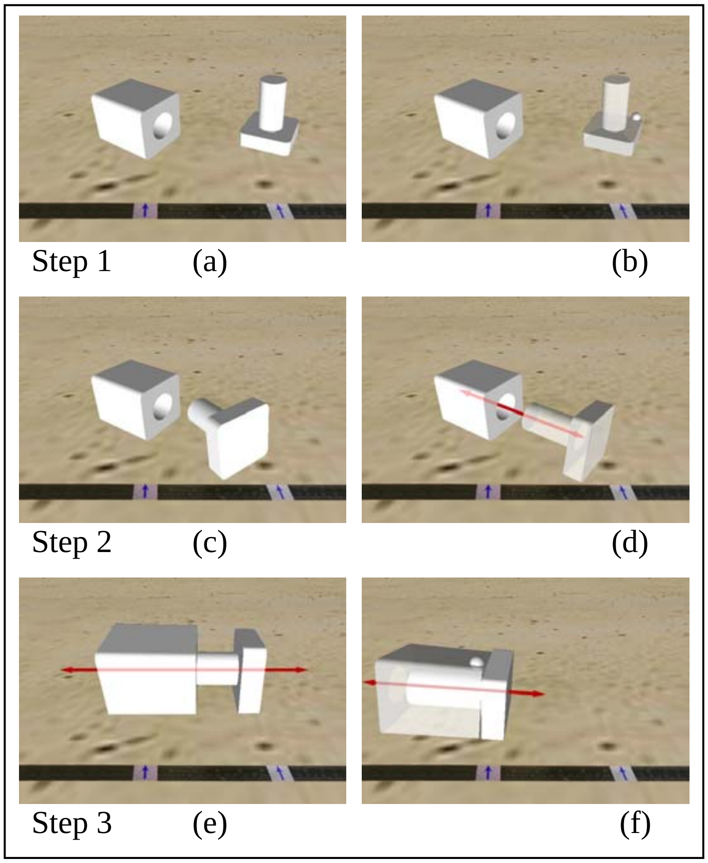


Figure 7: Collision, Physics and Constraint facilitating assembly in SHARP

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.

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 physical constraints 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 physical constraints 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. 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. 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. 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. in 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. in 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. in 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. in 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. A. G., de Sa 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. in 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. Wang, Y., Jayaram, U., Jayaram, S., and Shaikh, I., Methods and Algorithms For Constraint Based Virtual Assembly. Virtual Reality, 2003. **6**: p. 229-243.

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. in 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. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2003/CIE-48297). 2003. Chicago, IL.

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. in SIGGRAPH 99 Conference Proceedings, Annual Conference Series. 1999. Los Angles, CA.

23. Johnson, T.C., and, Vance, J. M. The Use of the Voxmap Pointshell Method of Collision Detection in Virtual Assembly Methods Planning. in 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. in 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. in 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. in 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): pp. 46-49.

31. Wang, Y., Jayaram, S., Jayaram, U., and Lyons, K. Physically Based Modeling in Virtual Assembly. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/CIE-21259). 2001. Pittsburg, PA.

32. Shaikh, I., Jayaram, U., Jayaram, S., and Palmer, C. Participatory Ergonomics Using VR Integrated with Analysis Tools. in 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): pp. 283-296.

34. Marcelino, L., Murray, N., and, Fernando, T., A Constraint Manager to Support Virtual Maintainability. Computers & Graphics, 2003. **27**(1): pp. 19 - 26.

35. Murray, N., and Fernando, T. An Immersive Assembly and Maintenance Simulation Environment. in 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. in 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. in 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. in Second International Conference on Embedded Software and Systems (ICESS'05). 2005.

41. McDermott, S.D., and, Bras, B. Development of a Haptically Enabled Dis/Re-Assembly Simulation Environment. in 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, in in 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. in ASME International Mechanical Engineering Congress & Exposition (IMECE2005-81873). 2005. Orlando, FL, USA.

45. Hoffmann, C.H., D-Cubed's Dimensional Constraint Manager. ASME Journal of Computing and Information Sciences in Engineering, 2001. **1**(1): p. 100-101.

CHAPTER 5. FEATURE-BASED CONSTRAINT RECOGNITION FOR VIRTUAL ASSEMBLY

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

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

For Submission to: *IEEE Computer Graphics & Applications*

Abstract

This paper presents a novel feature-based approach to geometric constraint recognition for virtual assembly in which the proposed algorithm takes advantage of dynamically contacting geometric features to accurately predict the user's assembly intent. A new hybrid method for virtual assembly simulations is presented, which combines physical and geometric constraints to achieve realistic part behavior and allow for precise part movements. In addition to tessellated models for visualization, B-Rep data representations of CAD models are imported and used for highly accurate collision detection, physical constraint simulation and constraint-based modeling. These techniques are demonstrated in the SHARP software (System for Haptic Assembly and Realistic Prototyping). Test results are presented for both simple and complex assembly scenarios which demonstrate a significant improvement in system performance when the automatic constraint recognition algorithm is used. The algorithm allows SHARP to automatically define, activate and delete geometric constraints in such a way that they remain transparent to the user. CAD parts with clearances as low as 0.001% can now be assembled using the system which detects collision responses to an accuracy of 0.0001mm.

Keywords: Virtual Reality, Human-Computer Interaction, Virtual Assembly, Constraint-Based Modeling, Automatic Constraint Recognition.

1. Introduction

Assembly planning is an important component of the product design process in which details about how parts of a new product will be put together are formalized. A well designed assembly process should take into account various factors such as optimum assembly time and sequence, tooling and fixture requirements, ergonomics, operator safety, and accessibility, among others.

Expert assembly planners today still use traditional approaches in which they examine three-dimensional (3D) CAD models of the parts to be assembled on two dimensional (2D) computer screens in order to determine assembly sequences for a new product. CAD systems also allow the user to assemble/disassemble parts using geometric constraints. These systems rely on user defined geometric constraints (align, mate, against, etc.) that snap parts into their final position. Some modern CAD systems have collision detection capabilities to prevent parts from interpenetrating during an assembly. However, it is important to note that geometric constraints always have precedence over collision detection which is only used as a secondary check.

These methods fail to simulate real-world assembly/disassembly mockups because of their inability to allow for intuitive and direct manipulation of computer models or to account for the complex human interactions that are involved in the assembly process. In addition, they do not simulate real-world physical constraints among contacting part surfaces which assembly workers rely upon to complete assembly tasks.

Virtual reality (VR) technology provides a more realistic and intuitive environment for assembly planning by immersing the user in a computer generated scene and allowing him/her to interact using natural human-motions. During the last decade, the virtual prototyping community has expressed considerable interest in harnessing this power for performing assembly/disassembly simulations in virtual environments. Performing assembly/disassembly design and evaluations using virtual mockups provides designers with an advantage over traditional techniques by allowing evaluations of multiple concepts during early stages of the design process, thus reducing the time and cost associated with physical mockups. In addition, such simulations could also be used as a virtual platform for offline training of assembly workers.

However, due to the complex and subtle nature of the human interactions involved, it is challenging to accurately simulate manual assembly tasks using the current virtual reality hardware and state-of-the-art algorithms. The research presented in this paper aims to combine previous methodologies and create a new algorithm to find an optimal solution which will allow the users to intuitively assemble complex CAD models in virtual environments.

2. Background

Several techniques for simulating assembly/disassembly operations in virtual environments have been used in the past. Earlier attempts to create such simulations utilized part snapping approaches where pre-defined final transformation matrices consisting of position and orientation information of each object were imported into the virtual environment. When parts were brought within certain proximity of each other, they were snapped to their final position to complete the assembly task.

Kuehne and Oliver [1] created a virtual assembly system called IVY (Inventor Virtual Assembly) to verify and evaluate the assembly characteristics of components. Assembly hierarchy information was imported and utilized in the virtual environment to select objects. Parts could interpenetrate each other as collision detection was not implemented in the system. Another virtual assembly system was developed by Pere et al. [2] using “World Toolkit”. The PC-based system, VShop, used bounding box collision detection for object selection. Gesture recognition was used for various tasks like switching on and off navigation and selecting parts in the environment.

Ye et al. [3] investigated the potential benefits of VR for assembly planning. An experiment was conducted which compared assembly in a traditional engineering environment, a non-immersive desktop VR environment, and an immersive CAVE (CAVE Computer Aided Virtual Environment) [4, 5] environment. An air cylinder assembly consisting of 34 parts was used in the experiment and it was concluded that subjects performed better in virtual environments than in traditional engineering environments in tasks related to assembly planning. Dewar et al. [6-8] developed two methods for part snapping namely, collision snapping and proximity snapping, for joining parts in the virtual environment at Heriot-Watt University. The system used a head mounted display (HMD) device for immersive visualization and a 3D mouse was used to interact with the system.

A three layer (scene graph layer, scripting layer and application layer) framework for abstraction was developed at BMW [9] which used a Cyber Touch glove device for gesture recognition (for holding parts). The system used proximity detection to trigger part snapping for assembly and provided tactile feedback to the user during assembly. Thus although part snapping approaches aided in precise part placement, they failed to model complex part interaction details such as the trajectories that parts follow when assembled in the real world.

Another approach involved interactively modeling real world physical constraints to simulate realistic part behavior for assembly. These systems relied on collision detection to prevent part interpenetration. Once collisions were detected, physics-based algorithms were utilized for simulating subsequent part trajectories. One of the first systems, developed by Gupta et al., [10, 11] was a desktop based virtual environment called VEDA (Virtual Environment for Design for Assembly). Two PHANToM® haptic devices from Sensable Technologies [12] were used for grasping virtual models between fingertips. This system was limited to handling only two-dimensional models for assembly.

A similar dual-handed desktop configuration was used by Coutee et al. [13, 14] to develop a virtual assembly application called HIDRA (Haptic Integrated Dis/Re-assembly Analysis). HIDRA treated the ‘fingertip’ as a point and thus allowed a user to grab models between two of his/her fingertips. In addition, the system did not allow simulation of realistic part grabbing and had limitations manipulating 3D objects. In addition the system only implemented collision detection between convex parts which necessitated breaking down non-convex parts into convex representations before they could be assembled.

The CORIOLIS^{TM} [15] physics-based simulation package was utilized by Fröhlich et al. [16] to create a virtual assembly system. The system used the Responsive Workbench [17] to simulate bench assembly scenarios. Various spring configurations were examined for simulating realistic interaction with virtual objects. Complex assembly tasks which resulted in several hundred collisions severely affected system update rates. At least five percent clearance was necessary to avoid numerical instabilities.

Kim et al. [18, 19] investigated several collision detection and physics-modeling software applications and found VPS [20] (Voxmap Point Shell) software from The Boeing Company to be most appropriate for assembly operations. The application expanded the capabilities of VEGAS [21] by implementing physics-based modeling for simulating realistic part behavior. Networking capabilities were later added to the application to facilitate collaborative assembly through the internet [22]. Although realistic part behavior was simulated, the volumetric-based approach of VPS, used coarse model representations to maintain interactive update rates and thus did not allow low clearance parts to be assembled.

Garbaya et al. [23] created a physics-based virtual assembly system which used a spring-damper model to provide the user with collision and contact forces during mating phase of the assembly operation. The PhysX® open source toolkit was used for collision detection and physics-based modeling. Grasping force feedback was provided using a CyberGrasp^{TM} haptic device and collision forces were provided using CyberForce^{TM} haptic device from Immersion Corporation. An experimental study was conducted to check system effectiveness and user performance in real and virtual environments. The study concluded that user performance improved when inter-part collision forces were rendered as compared to when only grasping forces are rendered by the system.

Geometric constraint-based modeling has gained a lot of attention in recent years for simulating assembly/disassembly operations in virtual environments. One of the earliest systems to use constraint-based modeling, VADE (Virtual Assembly Design Environment) was developed by Jayaram et al. [24-28] in 1997. The system used Pro/Toolkit to import assembly data (transformation matrices, geometric constraints, assembly hierarchy etc.) which was required for simulating the assembly task. Predefined geometric constraints imported from the CAD system were activated when related parts were in proximity to simulate constrained motion. Snapping methods were used to complete the assembly task. Stereo vision was provided in VADE using a HMD or an Immersadesk [29] system. A physics-based algorithm with limited capabilities was later added to VADE to simulate realistic part behavior [30]. Ergonomic evaluations for assembly tasks [31, 32] could be performed using a commercially available ergonomic software that was integrated into the system.

Marcelino et al. [33] at the University of Salford developed a constraint-based system for simulating virtual maintainability. The system supported multiple constraint recognition, automatic constraint management and multi-platform operation. The system was capable of handling simple planar and cylindrical surfaces for defining and validating constraints, determining broken constraints and solving constrained motion in the system. The D-Cubed constraint engine was used by the system to perform assembly and maintenance operations using complex CAD models [34, 35].

A CAVE-based system for virtual assembly called MIVAS (A Multi-Modal Immersive Virtual Assembly System), was developed by Wan [36] at Zhejiang University. MIVAS used Pro/Toolkit to import CAD geometry and predefined geometric constraints from Pro/Engineer CAD software in a similar fashion to VADE. The application performed hand-to-part collision detection using VPS [20] software, while part to part collision detection was implemented using RAPID [37].

Constraint-based modeling was used by Liu et al. [38] for assembly and tolerance analysis. The concept of “assembly ports” was introduced which imported 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. Various criteria (proximity, orientation, port type and parameter matching) were used for applying constraints among parts.

VECA (Virtual Environment for Collaborative Assembly), developed by Chen et al. [39] 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.

It is evident from the literature review that initial attempts at virtual assembly simulations utilized part snapping to accurately position parts in a virtual environment. These methods did not allow the users to analyze part movements that occur during assembly completion. Later, more advanced methods used later, focused on simulating real world physical constraints or geometric constraints for part positioning during assembly.

Interactive simulation of physical constraints between complex CAD models is a very challenging task. It requires systems to detect collisions among complex surfaces and generate subsequent part trajectories by using physics-based modeling techniques [20]. Low clearance assembly tasks among non-convex CAD models result in several hundreds/thousands of simultaneous collisions, resulting in numerical instabilities in the system, preventing interactive simulations [16]. Volumetric collision detection relies on coarse representation of complex models to maintain high update rates, making such approaches unsuitable for low-clearance assembly scenarios [40, 41].

The geometric constraint-based approach bypasses the complications involved in physics modeling. Rather than removing all degrees-of-freedom from the part as in snapping methods, this technique reduces the degrees-of-freedom of parts depending on the geometric constraint relationship among them. This allows the user the capability to move parts into their final positions with very high precision. However, for every assembly scenario, specific metadata requirements (transformation matrices, geometric constraints, material properties, assembly hierarchy, etc.) result in time consuming and cumbersome model preprocessing requirements whenever a new assembly scenario is imported. In addition most constraint-based applications have relied on Pro/Toolkit or other proprietary application programming interfaces to access the data required for the assembly system. Because most applications imported geometric-constraints from CAD systems, it was not possible to change constraint relationships within the virtual environment.

We see that approaches focusing on interactively simulating physical constraints among part surfaces provide the advantage of building a realistic environment which more accurately simulates manual assembly tasks in a virtual environment. Constraint-based methods on the other hand make it possible to overcome the limitations of VR input devices and the precision with which humans can manipulate virtual objects by applying part relationships which allow precise manipulation for assembly. Thus, both of these approaches provide specific advantages which are critical to achieving a complete solution to the problem. This focus of the research presented here is on creating a virtual assembly system where parts can be assembled naturally by simulating physical constraints among the surfaces of complex CAD models. A hybrid approach, utilizing physical constraints to model realistic part behavior and geometric constraints to obtain precise part manipulation is implemented. A novel feature-based automatic constraint recognition algorithm is presented, which makes geometric constraint recognition and behavior transparent to the user.

3. The SHARP Virtual Assembly System

Over the years, a significant research has been performed 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 examined, each resulting in specific advantages and disadvantages.

Assembly workers rely on physical constraints among mating surfaces for assembling parts in the real world. Thus, real-time simulation of such physical constraints is unquestionably the first step towards accurately simulating the complex interactions and part movements that occur during real world scenarios. Physical constraint simulation requires interactively detecting collisions that occur among complex models using physics-based methods to calculate subsequent part trajectories. Physics-based algorithms simulate forces acting on bodies along with their physical properties to model realistic behavior. In a rigid body dynamics simulation, equations of motion are solved at each time step to calculate movements of objects in the 3D space.

At Iowa State, we created several virtual assembly applications [13, 14, 22, 24, 36, 42] that utilized collision detection and physics-based modeling techniques for simulating physical-constraints during assembly. The newest system, “**SHARP”**, **S**ystem for **H**aptic **A**ssembly & **R**ealistic **P**rototyping [41] took advantage of our previous knowledge and presented a dual handed haptic approach to virtual assembly. SHARP utilizes the VRJuggler [43] software toolkit for controlling the virtual environment. The system is capable of being ported to different VR system configurations including low-cost desktop configurations, Barco Baron [44], 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 CAVE^{TM}) at geographically dispersed locations.

Initially, collision detection and physics modeling were implemented using the VPS [20] software from The Boeing Company. VPS used cubic elements called voxels to generate coarsely approximated volumetric representations of CAD models. These approximate representations were then used for calculating collision and physical responses. Thus, although SHARP could handle arbitrarily complex CAD geometry, 8-10% clearance between parts was required for successful completion of assembly tasks. Model approximations for collision and physics computations resulted in the following problems:

- Not possible to assemble parts with < 8% clearance
- 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 that was used

In addition, parametric CAD data was tessellated and converted into standard triangle-mesh-based formats which were imported into VR for graphic visualization and physics-based modeling. The accurate parametric and topological information (B-Reps) was lost and was not accessible in the VR environment. At the same time, performing low-clearance assembly tasks based solely on physical constraints required the system to be able to perform highly accurate collision detection among part surfaces which was not possible when approximated model representations were being used.

To provide a solution to this problem, parametric model data (B-Rep data) is now imported into SHARP using a standard CAD format and is used as the underlying model upon which collision and physical constraint responses are calculated. Accurate B-Rep models now allow SHARP to detect collisions with an accuracy of 0.0001mm among part surfaces. A B-Rep and graphics representation of a complex CAD model can be seen in figure 1.

3.1 Collision Detection & Physical Constraint Simulation in SHARP

Virtual assembly simulations require realistic interaction with the virtual objects that are present in the scene. This includes reaching out and grabbing parts as well as simulating realistic manipulation of objects in the 3D space. Part selection in SHARP is implemented using collision detection. Before a part is selected by the user, SHARP runs in “collision only” mode where the system detects collisions between the hand model and other objects present in the scene. Once the user selects a part, the system starts computing physical constraints at contacting surfaces. When a collision occurs, the system computes physical-responses and calculates subsequent part trajectories making it possible for the user to push parts realistically, and visualize gravitational and interaction forces.

In order to import a CAD model into SHARP, a graphic model file (*.wrl, *.iv, *.3ds, *.osg) and a Parasolid transmit file format (.x_t) model file containing B-Rep data

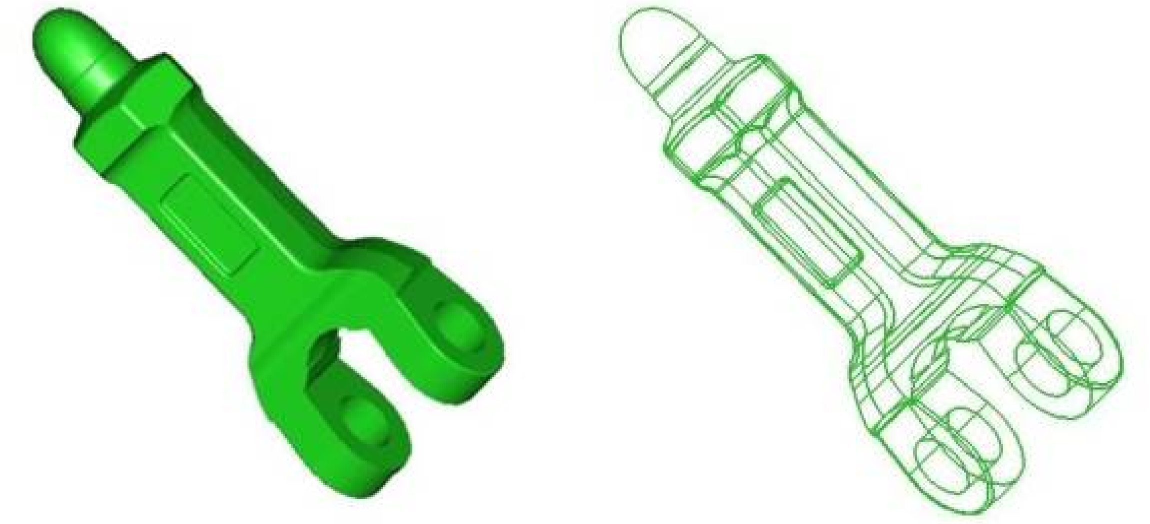


Figure 1: Graphic and B-Rep Representations

are required. OpenSceneGraph, an open-source scene graph library is used for visualization. The D-Cubed family of software components from UGS® is used in SHARP for collision detection and physics modeling. The Collision Detection Manager (CDM) module is used for calculating and querying collision/interference information, and the Assembly Engineering Manager (AEM) module is used for performing realistic physical simulation. When collisions are detected among parametric surfaces of CAD models, the AEM module treats them as contacts and calculates subsequent part trajectories, thus simulating realistic part behavior during assembly. As B-Rep data is used by both CDM and AEM components, SHARP simulates physical constraints with very high accuracy.

During system tests we discovered that despite the high accuracy at which physical constraints were modeled, the following unanticipated problems were encountered. First, when several parts were assembled, any movement in the assembly resulted in multiple collisions that occurred simultaneously causing an unnecessary burden on the physics solver.

Secondly, the noise associated with the input signal from magnetic tracking and other 3D input devices became prominent while assembling parts. This led to two things:

- Users could not assemble low-clearance parts because they could not achieve accurate relative motion required for precise part placement.
- When assembling parts, unnecessary intermittent contacts occurred due to minute hand vibrations being transmitted through the input devices; resulting in an excessive load on the physics solver.

3.2 Hybrid Approach: Combining Physical and Geometric Constraints

Constraint-based approaches for virtual assembly allow precise part manipulation using geometric constraint relationships. Once constraints are defined and applied, the constraint solver computes the new and reduced degrees-of-freedom of the part as well as its relative motion. Table 1 shows a comparison of physical and geometric constraint modeling approaches. To utilize the advantages of both approaches a new hybrid approach which combines physical constraints with geometric constraint-based modeling is implemented in SHARP and demonstrated in [45].

Attributes	Physical Constraint Simulation	Geometric Constraint Simulation	Hybrid Approach
Precise Part Positioning		X	X
Low Computational Load		X	X
Prevent Part Interpenetration	X		X
Realistic Part Behavior	X		X

Table 1: Approaches for Assembly Simulation

SHARP provides the user with an environment where physical constraints are simulated when collisions occur. In addition, a geometric constraint module has been developed and integrated into SHARP which allows the user to define constraint relationships among geometric features of different CAD parts present in the environment. Once defined, the object’s resulting part movements are calculated such that both physical and geometric constraints are satisfied at any given point in time. B-Rep model data, used for collision and physics computations, is also used for constraint definition and the Dimensional Constraint Manager (DCM) [46] from D-Cubed is used as the underlying constraint solver. The next section of the paper describes the automatic constraint recognition algorithm which identifies, defines, activates and deletes geometric constraints automatically, thereby providing an intuitive user experience and achieving optimum system performance when the hybrid approach is used.

4. Automatic Constraint Recognition in SHARP

Let us consider assembling a peg into a hole as shown in figure 2. An assembly worker would approach and grasp the pin. Then he/she would manipulate and align the pin to the hole and finally would try to push the pin to complete the assembly. He/she should be able to complete this assembly task if a positive clearance exists between the two mating parts.

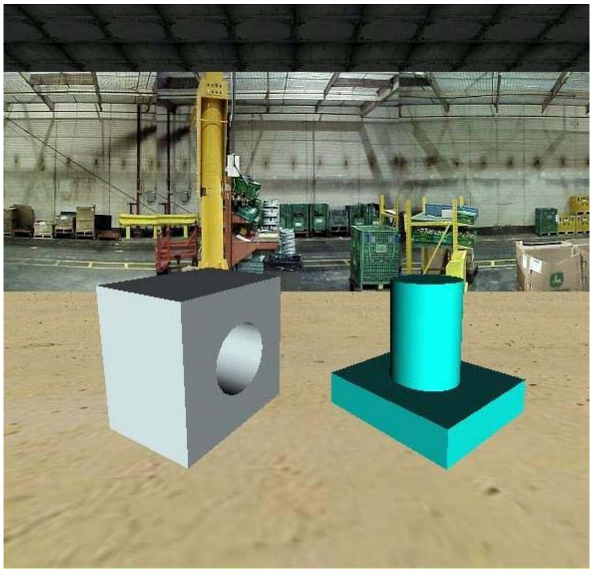


Figure 2: Pin and the Hole part

The user will be able to complete this assembly in SHARP by relying only on the simulated physical constraints when clearances are large. When smaller clearances are encountered, the two parts must be very accurately aligned and manipulated by the user to successfully complete the assembly task. However, in the absence of other modalities (haptic, sound, etc.) the user has to rely solely on visual aids (stereo visualization) to complete the assembly task which requires greater effort on the part of the user. Geometric constraints aid the user in achieving such precise part manipulations by reducing the degrees-of-freedom of the parts, thus facilitating assembly.

Marcelino et al. [33] developed an automatic constraint-recognition algorithm for a constraint-based virtual assembly system. The algorithm was initially slow and was later improved by adding preprocessing steps in which bounding boxes were added to all surfaces of the objects before they were imported. The system used bounding box collision detection to shorten the list of potential colliding surfaces that needed to be analyzed for possible constraints. Another constraint recognizer was implemented in the IVAE constraint-based virtual assembly system developed by Yang [47]. The system imported pre-defined constraints from CAD systems and the algorithm recognized the constraints and highlighted geometry elements based on coarse bounding box collision detection. Constraints were applied at the user’s discretion.

These applications relied on constraint-based modeling for virtual assembly and thus could not simulate realistic physical interactions among parts. In addition they used approximated bounding box collision detection approaches which resulted in large potentially colliding surface lists [33]. Other problems with such algorithms included slow constraint recognition response, intensive model preprocessing requirements and the need for pre-defined geometric constraint metadata from CAD systems [47].

SHARP provides a new perspective to geometric constraint automation by utilizing a feature-based approach where contacting geometric features are analyzed in order to predict the user’s assembly intent. In addition, this approach combines feature-based geometric constraint recognition with physical constraints to facilitate assembly in virtual environments. Seamless integration of geometric constraints and physics behavior, led to the following requirements for the algorithm:

- No constraint metadata from CAD systems
- On the fly constraint definition using B-Rep data
- Fast & robust operation
- No/minimal user intervention
- Automatic constraint deletion

The feature-based automatic constraint recognition algorithm described here consists of five essential steps:

- Predict user’s assembly intent
- Identify possible constraints
- Check constraint validity
- Apply constraint
- Delete constraints when no longer required

SHARP relies on physical constraints for assembling parts in the virtual environment. Geometric constraint definitions are only required when low clearance parts need to be assembled. Before a geometric constraint is defined automatically, the algorithm must first predict the user’s assembly intent. To promote direct data transfer from CAD, the hybrid approach used in SHARP relies solely on B-Rep geometry for simulating physical and geometric constraints. Without access to auxiliary CAD information such as predefined constraints, assembly hierarchy, etc., predicting assembly intent robustly during an interactive assembly simulation is a very challenging problem.

As soon as the user grabs and begins manipulating a part, SHARP's physics sequence is initiated. The constraint recognition algorithm monitors part collisions during the physics sequence and queries surface contacts that occur during each time step. The use of exact B-Rep (surface, edge, and vertex) data for collision detection and physics modeling boosts the algorithm's performance. Contact queries provide the algorithm with a list of the exact geometric entities that are currently contacting; which results in very few contacts (0-3) that need to be analyzed during each time-step. This proves to be a tremendous advantage over previous approaches which used bounding box collision detection resulting in large surface lists to be analyzed for potential constraints. Once the list of contacts is generated, the algorithm analyzes the type of geometric features that are contacting and predicts what the user is trying to achieve.

Considering the above pin and hole assembly example we see that the pin part has 7 planar surfaces and 1 cylindrical surface while the hole part has 6 planar surfaces and 1 cylindrical surface (figure 3). In order to start the assembly task the user grabs the pin and

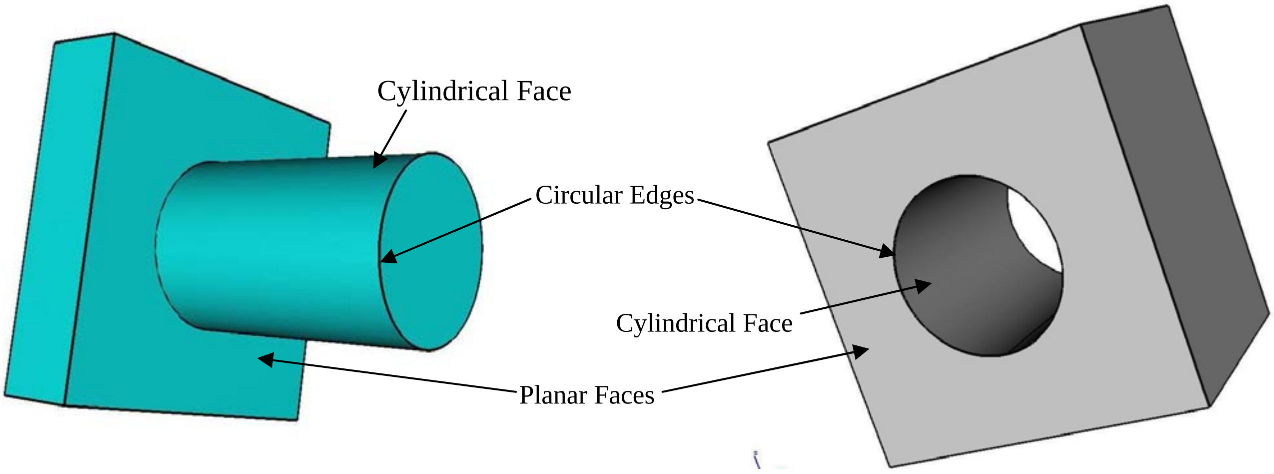


Figure 3: Geometric Entities (faces, edges) in B-Rep Data Models

roughly aligns it with the hole to complete the assembly. As soon as the physics sequence starts, the algorithm analyzes contacts at every time step. Because constraints are only possible between circular elements (faces, edges) and planar faces, the algorithm ignores all other contacts and allows the user to continue the assembly process based solely on simulated physical constraints. Filters are used to make the algorithm more robust and less prone to errors. If constraints are possible, the system queries the face, edge data from the B-Rep geometry database. If two planar faces are colliding and the angle between their surface normals is less than a threshold value, a co-planar constraint is defined. If contacting geometries are circular faces or edges and the difference in diameter of the two entities and the angle between their axes is less than a threshold value, a concentric constraint is defined.

Once constraints are defined, a validity check is performed which determines if a solution is possible when these constraints are activated. If a constraint is valid, it is automatically activated by the algorithm. Thus, once the user tries to insert the pin into the hole, a constraint is defined and activated, which assists the user in proper part alignment and prevents unnecessary computational load on the physics solver. It is important to note that the all automatically defined constraints are temporary and are deleted by the system as soon as the part is released by the user. The algorithm flowchart is shown in figure 4.

5. Test Results

To evaluate the effect of the hybrid approach on system performance, this section presents experimental results of assembly tasks involving complex CAD models. The computer used for these tests was running Microsoft Windows® and had dual Xeon 3.6Ghz processors, 4 GB RAM and an Nvidia Quadro 4400 graphics card. The parts were provided by Deere & Company. In order to take full advantage of the CPU, the physics calculations are performed in a separate high-priority thread. The graphics thread maintains a constant update rate of ~75 frames/sec.

During the first test, we compared the system performance with and without the use of the automatic constraint recognition algorithm. In the first case, the assembly task was to insert the pin part into the two holes of the center-link as shown in figure 5 using only physical constraint simulation. Once the assembly was completed, the user had to move the pin part such that the center-link moves with it. During the

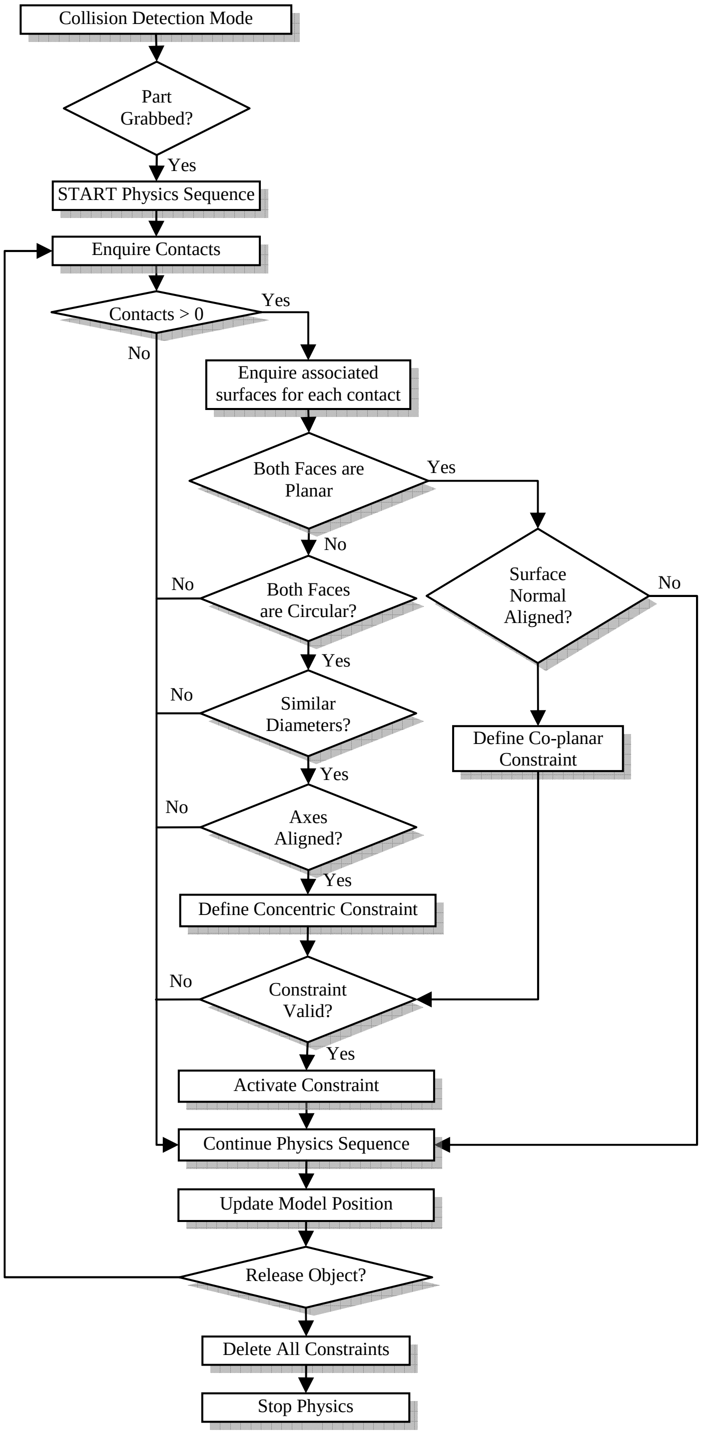


Figure 4: Constraint Recognition

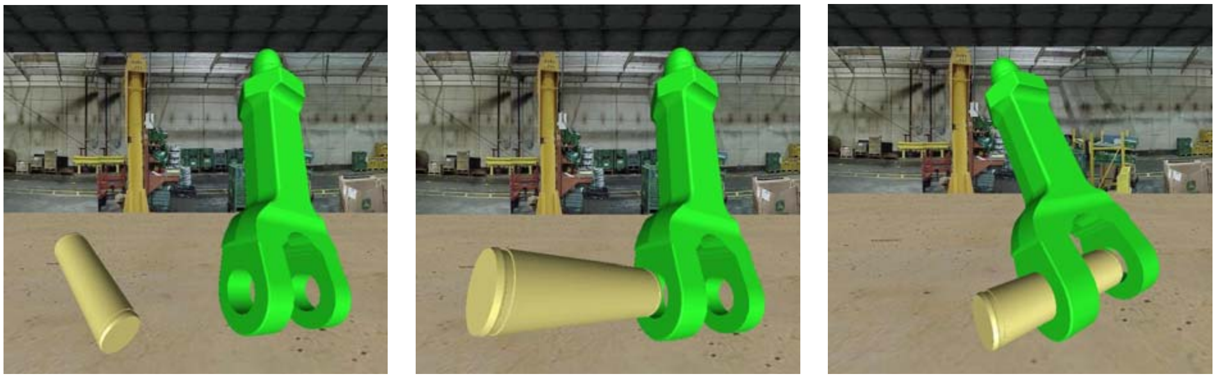


Figure 5: Assembling Pin and Center-link using SHARP

second case, the automatic constraint recognition algorithm was activated. Thus, in this trial as soon as the user starts inserting the pin part, the geometric constraints are identified and temporarily added to the system. The test results are shown in figure 6. Frame rate data is provided when the user first grabs the pin, during part movement and when contacts occur. The graph shows that the in both cases the frame rate remains ~120 frames/sec during stationary non colliding situations and ~45 frames/sec during part movements when no contacts were occurring.

In the physics-only case, the frame rate dropped to ~5 frames/sec during the

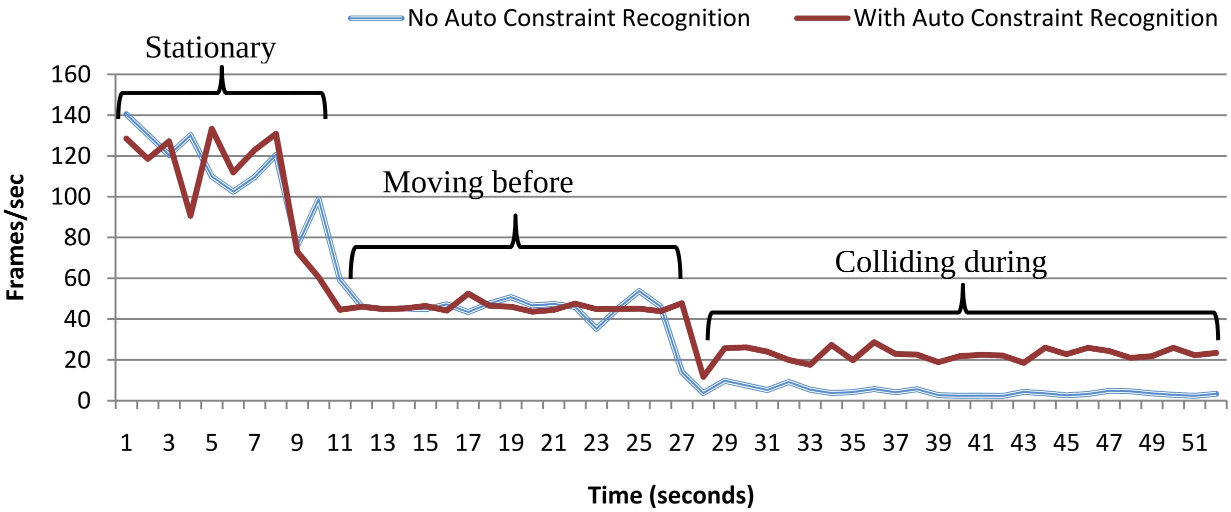


Figure 6: Graph Showing Physics Frame Rates During Various Stages of Assembly

insertion operation. It is evident that when automatic constraint recognition was active, the system used constraints in addition to physics modeling with resulted in avoiding unnecessary collisions as predicted and thus helped maintain the frame rate at ~22 fps. The algorithm took ~15 milliseconds for recognizing the constraint. It is important to note that although the use of highly accurate B-Rep data models for collision/physics computations makes it possible to assemble parts in low clearance situations; it results in low frame rates (~20-140 frames/sec) than our previous approach (~500-1000 frames/sec) which used volumetric approximations for collision and physics computations [41]. The significant increase (approximately 4 times) in frame rate that is seen when using the automatic constraint recognition in conjunction with physical constraints is critical to the interactivity of the simulation.

Previous approaches that used constraint-based modeling for assembly based their constraint recognition/activation on the part that the user is currently manipulating [24, 47, 48]. The feature-based automatic constraint approach implemented in SHARP is designed to identify potential constraints based on any contact that occurs between parts in the virtual environment regardless of the part being manipulated by the user.

Figure 7: Pin and Bracket Assembly

In more complicated assembly scenarios such as the one shown in figure 7, where several pins are already assembled to the bracket, any manipulation task will trigger a sequence of collisions. This would result in intermittently occurring simultaneous physical contacts among the various parts present in the assembly escalating the computation load that is already present. The constraint algorithm in SHARP provides a solution to this problem by identifying potential constraints among already assembled parts in this case the pins and the bracket. As soon as the user manipulates a part, SHARP system automatically identifies and applies a series of concentric constraints which temporarily replace the physical contact between each pin and the bracket resulting in an optimum system performance. All geometric constraints are automatically deleted by the system once the manipulated part is released by the user. The second test involved assembling three pins and the bracket and manipulating the assembly. The results shown in figure 8 indicate that this approach of replacing physical contacts with temporary geometric constraints has a significant positive impact on system performance. In addition to performance improvements, experimental trials proved that using the new feature-based constraint recognition approach users can assemble parts with clearances as low as 0.001% while the system only allows assembly with clearance greater than 0.01% when only physical constraints are used.

The above test results show that the feature-based automatic constraint recognition algorithm led to a distinct improvement in the SHARP system's performance and facilitated assembly in low-clearance scenarios. The use of automatic constraint recognition also had a pronounced impact on the system usability. The algorithm made it possible to transfer the constraint definition, activation and deletion workload away from the user allowing him/her to concentrate better on the task at hand. The demonstrated algorithm proved successful in keeping geometric constraint recognition transparent to the user thus maintaining the realism of the physical simulation.

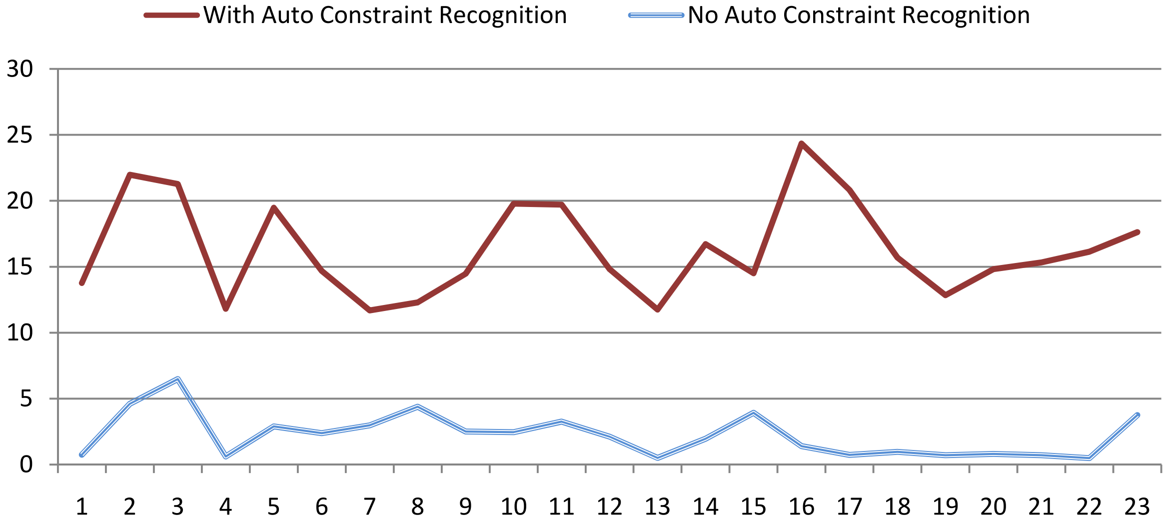


Figure 8: System Performance Comparison in Complex Assembly Scenarios

6. Conclusions and Future Work

This paper presents a novel approach to optimizing system performance by temporarily replacing recurring physical contacts with geometric constraints for virtual assembly simulations. Geometric constraints avoid unnecessary recurring collisions which occur due to the noise associated with the input signal and also facilitate precise part manipulations in virtual environments. A feature-based automatic constraint recognition algorithm is described in this paper. During each time-step, the algorithm analyzes the contacting geometric entities to predict the assembly intent of the user. Once identified, the system automatically chooses the appropriate constraint, defines and activates it. The system relies on auxiliary feature information (such as dimensions, orientation, entity type, etc.) that is available from the model's B-Rep data, to accurately predict the intent as well as a suitable constraint type. All active constraints are automatically deleted by the system as soon as the user releases the part that is being manipulated. SHARP uses accurate B-Rep model data for collision detection and physical modeling which provides an accurate list of contacting surfaces resulting in faster constraint recognition (~15 milliseconds) as compared to previous approaches which used bounding-box collision detection approaches.

The hybrid approach integrated into the SHARP virtual assembly system takes advantage of both physical and geometric constraints. Physical constraint simulation allows the user to visualize real-world scenarios and simulate realistic part behavior when collisions occur. Constraint-based modeling, on the other hand, facilitates assembly by reducing the degrees-of-freedom of the parts allowing precise part manipulation. With the integration of automatic constraint recognition, the hybrid approach demonstrates one of the first attempts in which both physical and geometric constraints are created and deleted automatically at runtime. In addition, this approach allows direct transfer of data from CAD to VR with no processing or proprietary CAD data/toolkit requirements.

The feature-based automatic constraint recognition algorithm allows the user to intuitively assemble parts and optimizes system performance. The test results demonstrate that the feature-based automatic constraint recognition algorithm significantly improves the SHARP system's performance in both simple and complex assembly scenarios. Rather than using constraint detection only for the manipulated object as attempted previously, the algorithm checks all contacts occurring in the environment and applies all appropriate constraints. Automatic constraint recognition allows users to assemble parts with clearances as low as 0.001% without distracting the user with unnatural menu and voice-based interaction.

This research presents a proof of concept for assembly/disassembly simulations in virtual environments. Initial tests show promising results using small assembly sets. Future work will focus on expanding this system to be capable of handling larger assemblies consisting of hundreds of parts. The current system only provides visual and auditory feedback to the user. Although the optimized physics performance is acceptable for visual simulations, integrating haptic feedback for assembly will require physics update rates to be ~500 -1000Hz.

7. Acknowledgements

We are grateful for the technical assistance of Rakesh Aggarwal from 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.

8. References

1. Kuehne, R., and Oliver, J. A Virtual Environment for Interactive Assembly Planning and Evaluation. in Proceedings of ASME Design Automation Conference. 1995. Boston, MA., USA.
2. Pere, E., Langrana, N., Gomez, D., and Burdea, G. Virtual Mechanical Assembly on a PC-Based System. in ASME Design Engineering Technical Conferences and

Computers and Information in Engineering Conference (DETC1996/DFM-1306). 1996. Irvine, CA.

3. 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.
4. 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.
5. 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.
6. Dewar, R.G., Carpenter, I.D., Ritchie, J.M., and Simmons, J.E.L. Assembly Planning in a Virtual Environment. in Proceedings of Portland International Conference on management of Engineering and Technology (PICMET 97). 1997. Portland, OR: IEEE Press.
7. 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.
8. 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. in Proceedings of 12th Conference of the Irish Manufacturing Committee. 1995.

9. A. G., de Sa and Zachmann, G., Virtual Reality as a Tool for Verification of Assembly and Maintenance Processes. Computers and Graphics, 1999. 23: p. 189-403.
10. Gupta, R., and Zeltzer, D. Prototyping and Design for Assembly Analysis using Multimodal Virtual Environments. in Proceedings of ASME Computers in Engineering Conference and the Engineering Database Symposium. 1995. Boston, MA.
11. 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.
12. Wang, Y., Jayaram, U., Jayaram, S., and Shaikh, I., Methods and Algorithms For Constraint Based Virtual Assembly. Virtual Reality, 2003. 6: p. 229-243.
13. 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.
14. Coutee, A.S., and, Bras, B. Collision Detection for Virtual Objects in a Haptic Assembly and Disassembly Simulation Environment. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2002/CIE-34385). 2002. Montreal, Canada.
15. Baraff, D., Interactive Simulation of Solid Rigid Bodies. Computer Graphics and Applications, 1995. 15(3): p. 63-75.

16. Fröhlich, B., Tramberend, H., Beers, A., Agarawala, M. and Baraff, D. Physically-Based Modeling on the Responsive Workbench. in IEEE Virtual Reality Conference. 2000.
17. Krüger, W., and Fröhlich. B., The Responsive Workbench. Computer Graphics and Applications, 1994. 14(3): p. 12-15.
18. Kim, C.E., and Vance, J.M. Using VPS (Voxmap Pointshell) As The Basis For Interaction in a Virtual Assembly Environment. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2003/CIE-48297). 2003. Chicago, IL.
19. 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.
20. McNeely, W.A., Puterbaugh, K. D. and Troy, J. J. Six Degree-of-Freedom Haptic Rendering Using Voxel Sampling. in SIGGRAPH 99 Conference Proceedings, Annual Conference Series. 1999. Los Angles, CA.
21. Johnson, T.C., and, Vance, J. M. The Use of the Voxmap Pointshell Method of Collision Detection in Virtual Assembly Methods Planning. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2001/DAC-21137). 2001. Pittsburgh, PA.
22. Kim, C.E., and Vance, J.M. Development of a Networked Haptic Environment in VR to Facilitate Collaborative Design Using Voxmap Pointshell (VPS) Software. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2004/CIE-57648). 2004. Salt Lake City, UT.

23. Garbaya, S.a.Z.-C., U., The Affect of Contact Force Sensations on User Performance in Virtual Assembly Tasks. Virtual Reality, 2007.
24. 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.
25. Jayaram, S., Connacher, H. I., and Lyons K.W., Virtual Assembly using Virtual Reality Techniques. Computer Aided Design, 1997. 29(8): p. 575-584.
26. Jayaram, U., Tirumali, H. and, Jayaram, S. A Tool/Part/Human Interaction Model for Assembly in Virtual Environments. in ASME Design Engineering Technical Conferences 2000 (DETC 2000/CIE-14584). 2000. Baltimore, MD.
27. Taylor, F., Jayaram, S. and, Jayaram, U. Functionality to Facilitate Assembly of Heavy Machines in a Virtual Environment. in ASME Design Engineering Technical Conferences (DETC 2000/CIE-14590). 2000. Baltimore, MD.
28. Jayaram, S., Jayaram, U., Wang, Y., and Lyons, K. CORBA-based Collaboration in a Virtual Assembly Design Environment. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2000/CIE-14585). 2000. Baltimore, MD.
29. 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.
30. Wang, Y., Jayaram, S., Jayaram, U., and Lyons, K. Physically Based Modeling in Virtual Assembly. in ASME Design Engineering Technical Conferences and

Computers and Information in Engineering Conference (DETC2001/CIE-21259). 2001. Pittsburg, PA.

31. Shaikh, I., Jayaram, U., Jayaram, S., and Palmer, C. Participatory Ergonomics Using VR Integrated with Analysis Tools. in 2004 Winter Simulation Conference. 2004. Washington D.C.
32. 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.
33. Marcelino, L., Murray, N., and, Fernando, T., A Constraint Manager to Support Virtual Maintainability. Computers & Graphics, 2003. 27(1): p. 19 - 26.
34. Murray, N., and Fernando, T. An Immersive Assembly and Maintenance Simulation Environment. in Eighth IEEE International Symposium on Distributed Simulation and Real-TIme Applications. 2004.
35. Murray, N., and Fernando, T. Development of a an Immersive Assembly and Maintenance Simulation Environment. in International Conference on Advanced Research in Virtual and Rapid Prototyping. 2003. Leiria, Portugal.
36. Wan, H., Gao, S., Peng, Q., Dai, G and Zhang, F. MIVAS: A Multi-Modal Immersive Virtual Assembly System. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC 2004/CIE-57660). 2004. Salt Lake City, UT.
37. Gottschalk, S., Lin, M. C. and Manocha, D. OBBTree: A Hierarchical Structure for Rapid Interference Detection. in 23rd Annual Conference on Computer Graphics and Interactive Techniques. 1996.

38. 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.
39. Chen, X., Xu, N., and Li, Y. A Virtual Environment for Collaborative Assembly. in Second International Conference on Embedded Software and Systems (ICESS'05). 2005.
40. Seth, A., Su, H. J., and Vance, J. M. A Desktop Networked Haptic VR Interface for Mechanical Assembly. in ASME International Mechanical Engineering Congress & Exposition (IMECE2005-81873). 2005. Orlando, FL, USA.
41. Seth, A., Su, H. J., and Vance, J. M. SHARP: A System for Haptic Assembly & Realistic Prototyping. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2006/CIE-99476). 2006. Philadelphia, PA, USA.
42. McDermott, S.D., and, Bras, B. Development of a Haptically Enabled Dis/Re-Assembly Simulation Environment. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC1999/CIE-9035). 1999. Las Vegas, NV.
43. Just, C., A. Bierbaum, A. Baker, and, C. Cruz-Neira, VR Juggler: A Framework for Virtual Reality Development, in in 2nd Immersive Projection Technology Workshop (IPT98) CD-ROM. 1998: Ames, IA.
44. Barco Baron (http://www.barco.com/entertainment/en/products/product.asp?element=1192).

45. Seth, A., Vance, J. M., and Oliver, J. H. Combining Geometric Constraints with Physics Modeling for Virtual Assembly Using SHARP. in ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (DETC2007/CIE-34681). 2007. Las Vegas, NV, USA.
46. Hoffmann, C.H., D-Cubed's Dimensional Constraint Manager. ASME Journal of Computing and Information Sciences in Engineering, 2001. 1(1): p. 100-101.
47. Yang, R., Wu, D., Fax, X., and Yan, J., Research on Constraint-Based Virtual Assembly Technologies. Frontiers of Mechanical Engineering in China, 2007. 2(2): p. 243-249.
48. Wang, Q.H., Li, J. R., and Gong, H. Q, A CAD-linked Virtual Assembly Environment. International Journal of Production Research, 2006. 44(3): p. 467-486.

CHAPTER 6. CONCLUSIONS AND FUTURE WORK

This research presented in this dissertation focused on creating and identifying methods to perform manual assembly/disassembly operations in a virtual environment in the most natural and intuitive way possible. The intent is to create a virtual environment in which assembly workers can interact with and assemble virtual prototypes of early design concepts in a manner closely analogous to physical prototypes; without prior VR experience or training.

To create a realistic virtual environment, the research focused on interactively simulating physical constraints among complex parts to facilitate part placement for assembly. A virtual assembly system called SHARP was created during the course of the research. The initial system utilized a volumetric approach for collision detection and physical constraint simulation where model geometries were coarsely discretized using cubic elements called voxels. Natural interaction was provided by implementing a dual-handed haptic interface for the system. The interface allowed the user to simultaneously manipulate and orient parts or subassemblies and feel contacts that occurred during assembly. Although model approximations resulted in very fast computation rates (~1000Hz) they did not allow parts with low clearances to be assembled. In addition, high memory and computation requirements made the system unsuitable for large model datasets.

This problem was addressed by importing accurate model representations (B-Reps) into VR and utilizing them for collision detection and physical constraint simulation. Integration of accurate B-Rep geometry allowed the system to compute collision responses up to an accuracy of 0.0001mm. Additionally, capabilities for simulating geometric constraint-based behavior were also integrated into the SHARP system. The modularized system architecture allowed different methods (collision, physical and geometric constraints) to be used independently or in combination to complete the assembly task at hand. Based on different case studies, it was identified that even with the highly accurate simulation of physical constraints facilitated by B-Rep models; users could not manipulate objects with the required precision when low clearance parts were assembled.

A new hybrid approach which combines physical and geometric constraints to achieve realistic part behavior and to allow for precise part movements is demonstrated. When parts collide during assembly, the system computes their subsequent trajectories such that both physical and geometric constraints are satisfied. The frequent menu and voice-based interactions required for constraint definition are avoided by implementing a feature-based automatic constraint recognition algorithm. The algorithm monitors contacting geometric features to predict the user's assembly intent and handles identification, activation and deletion of geometric constraints such that they are transparent to the user. The algorithm optimizes system performance by replacing intermittently occurring simultaneous physical contacts that occur during assembly, with temporary geometric constraints. With the integration of automatic constraint recognition, the SHARP system demonstrates one of the first attempts at virtual assembly where both physical and geometric constraints are created and deleted automatically at runtime without any preprocessing or proprietary CAD data/toolkit requirements. The research presents a proof-of-concept where physical contacts can be modeled with very high accuracy and an optimum system performance is achieved by combining physical and geometric constraints in a novel way resulting in a more realistic user experience.

In the future, one area worth investigating is expanding the current system to be capable of handling larger assemblies consisting of hundreds of parts. This will result in thousands of faces and edges among which collisions will need to be calculated which will drastically affect system update rates. It is important to note that collision detection accuracy of 0.0001mm is only required for very low clearance assembly scenarios. Part contacts that occur during manipulation tasks or when dynamic parts collide with each other could satisfactorily be modeled by sacrificing this accuracy to achieve high update rates. One possible method might be to have simultaneous triangle mesh and B-Rep representations of part models for collision detection. The system should then be designed such that it can switch between coarse and accurate models for contact queries at runtime. It could be achieved by investigating ways to combine open source dynamics solvers such as PhysX from Ageia® or Open Dynamics Engine (ODE) with the geometric constraint solvers.

In addition, interaction realism can be increased by implementing methods for full body collision detection with the environment. Currently, the system only provides visual and auditory feedback to the user. Methods which will allow the system to provide simultaneous collision (physical constraint among parts) and contact information (grasping) to the user, need to be investigated. This will require modeling realistic hand-part contacts, use of exoskeletons, robotic arms and physics update rates of ~1000Hz.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to my major professor Dr. Judy M. Vance for her excellent guidance and support during the course of my research. Her outstanding insights, unbounded patience and cooperation helped me grow as a researcher and made my graduate studies more enjoyable.

I would also like to thank Dr. James H. Oliver for his expertise and invaluable discussions in overcoming impediments during my research as well as the other members of my committee for their guidance.

In addition, I would like to thank my cousin Akansha and my friends Emily and Dao for their help and support during my graduate studies. At last, I would like to extend my profound sense of respect to my parents for their tremendous patience, encouragement and support during my education.

BIOGRAPHICAL SKETCH

Abhishek Seth received a Bachelor of Technology degree in production engineering from G.B. Pant University of Agriculture and Technology, India in 2002. He graduated with a Masters of Science degree in Industrial Education and Technology from Iowa State in December 2003. His master's research focused on creating a low cost VR application to promote VR in an introductory undergraduate design class. He completed his Ph.D. in Mechanical Engineering and Human Computer Interaction at Iowa State University in August 2007. His research focuses on exploring methods to simulate realistic behavior and interaction for advancing the state-of the-art in virtual prototyping simulations. His research interests are in Virtual Prototyping, Virtual Reality Applications in Mechanical Design, Computer Graphics, Haptic Interfaces, CAD-VR data exchange and Human Computer Interaction. He is currently a member of ASME and IEEE.