Design and implementation of a haptic-based virtual assembly system

Pinjun Xia, António Lopes and Maria Restivo
IDMEC-Polo FEUP, Faculty of Engineering, Porto University, Porto, Portugal

Abstract

Purpose – Haptics can significantly enhance the user's sense of immersion and interactivity. Especially in an assembly task, haptic feedback can help designers to have a better understanding of virtual objects and to increase task efficiency. The purpose of this paper is to investigate the design and implementation of a haptic-based virtual assembly system (HVAS).

Design/methodology/approach – A multi-thread system structure was designed, an automatic data integration interface was developed to transfer geometry, topology, assembly and physics information from a computer-aided design system to virtual reality application, and a hierarchical constraint-based data model and scene graph structure was designed to construct the virtual assembly environment. Unlike traditional virtual assembly systems based on collision detection or geometry constraint only, a physics-based modeling approach combining with haptic feedback and geometry constraint was undertaken to realize and guide the realistic assembly process. When two parts collide into each other, the force and torque can be computed and provide feedback, and a spring-mass model is used to prevent penetration and simulate dynamic behaviour. When two parts are close enough to each other and the assembly simulation state is activated, a geometry constraint can be captured, an attractive force can be generated to guide the user to assemble the part along the correct position, and the repulsive force can also be generated to realize the mating process as natural and realistic as in real life.

Findings – The implementation details and application examples demonstrate that haptic-based virtual assembly is a valuable tool for assembly design and process planning.

Originality/value – The paper presents an HVAS.

Keywords Virtual reality, Virtual assembly, Haptics, Data integration, Physics-based modeling, Design

Paper type Research paper

Introduction

As innovative and promising technology emerged in recent years, virtual reality (VR) has now pervaded a large spectrum of applications in industrial, medical and educational environments. Virtual assembly, one of the most challenging applications of VR in engineering, can be defined as the use of computer tools to make or "assist with" assembly-related engineering decisions and designs through analysis and predictive models, visualization and presentation of data, simulating realistic environment behaviour and part interaction without physical realization of the product or supporting processes (Jayaram *et al.*, 1997). The first objective of virtual assembly is to test the feasibility of the assembly operations at the design stage of the product. The manipulation of the digital prototype allows for the improvement of the product design and its assemblability. The second objective is to generate optimal assembly plans for the product using virtual assembly approach. Proactive production planning can be carried out and resource allocation, assembly time and cost estimation and the assembly operators' training can be performed before making the real production. Maintenance operations and planning can be also made using the virtual assembly/disassembly approach.

Realistic virtual environment and natural human-machine interaction are the two most important aspects of a virtual assembly system. Today's virtual assembly environments are capable of simulating visual realism to a very high level, and present successful applications for the early evaluation of various aspects of products, such as functional, ergonomics, usability, etc. The next big challenge for the virtual assembly community is simulating realistic interaction. Haptics is an evolving technology that offers a revolutionary approach for realistic interaction in virtual environment, which can be defined to feel and manipulate virtual objects by using special input/output devices to get tactile and force feedback. Researchers have found that a haptic interface is desirable for performing assembly tasks in virtual environments, because it can help the designers obtain a better understanding of the geometry of virtual objects, to feel more secure and be more confident in regards to the real-world process, and therefore increase the user's sense of immersion and interactivity.

The aim of the work presented in this paper is to investigate the design and implementation of a haptic-based virtual assembly system (HVAS). Related works are reviewed in the second section. In the third section, a multi-thread system

The current issue and full text archive of this journal is available at
www.emeraldinsight.com/0144-5154.htm

Assembly Automation
31/4 (2011) 369–384

[DOI 10.1108/01445151111172961]

structure was designed, and the construction of virtual environment is discussed in the fourth section, including computer-aided design (CAD) and VR data integration, data representation and reorganization and scene graph design. Haptic-based assembly modeling and dynamic simulation are discussed in the fifth section, including collision detection and dynamic simulation, and haptic feedback and assembly simulation. System implementation details and application examples are discussed in the sixth section, some conclusions are given in the seventh section.

Related work

Virtual assembly with geometry constraint modeling

Since the emergence of VR technology, many academic and industrial research groups have been interested in the application of this new technology in assembly planning and training, and many prototype systems have been developed. Early virtual assembly systems, which do not have physics and haptics, are mainly based on geometry constraint modeling. A virtual assembly design environment (VADE) (Jayaram *et al.*, 1999) was developed by Washington State University who were the first to provide a definition of virtual assembly, and it was a well-known application of VR technology in manufacture industry. VADE modeled part behaviour by importing geometry constraints and data from CAD systems, one- or two-handed assembly operations were supported by data glove, and stereo vision was provided by an HMD or Immersadesk device. An unbelievable vehicle for assembling virtual unit (Ritchie *et al.*, 1999) developed by Heriot-Watt University was another system to assemble the product from CAD models in virtual environment. The operators' actions could be monitored and assembly sequence plans were automatically generated. Proximity snapping and collision snapping were developed as a positioning tool to mate parts together. A virtual environment for general assembly (VEGAS) (Johnson and Vance, 2001) was developed by Iowa State University. The system used data glove from 5DT for human grab motion, VRJuggler software for virtual environment management and Voxmap PointShell from Boeing Corporation for collision detection; however, no haptics was implemented in this early system. V-REALISM (Li *et al.*, 2003), a desktop VR-enabled system for assembly and maintenance, was developed by Nanyang Technological University in Singapore. The geometric models of virtual environment were constructed using CAD systems, and then transferred into the prototype through a conversion interface. A hierarchical structure was proposed to partition and organize these imported models in virtual environment. Based on this structure, a visibility culling approach was developed for fast rendering and user interaction. Marcelino *et al.* (2003) developed a geometric constraint manager for simulating interactive assembly or disassembly tasks in virtual environment, which could support features like multi-platform operation, scene graph independence, multiple constraint recognition and automatic constraint management. The constraint manager was capable of validating existing constraints, determining broken constraints, enforcing existing constraints, solving constrained motion and recognizing new constraints.

Because most of these systems lack force feedback and haptics, geometry constraint is the main approach to model part behaviour, no reaction forces are calculated and haptic behaviours between parts are not achieved. When the related parts come in close proximity to each other, the geometry constraint can be captured and recognized, the precise assembly position and the reduced degrees-of-freedom of objects can be calculated by the constraint solver, and then constraint-based motion simulation can be visualized. Compared to the physics-based method, geometry constraint modeling method needs less computation and memory requirements, and can compute the correct position for assembling part with very high accuracy. However, although geometry constraint-based modeling method has been proved to be successful for assembling application, the main shortcoming of this method is that it cannot support natural interaction and realistic object behaviour in virtual environment.

Virtual assembly with physics and haptics modeling

Natural interaction has an important effect on human performance during assembly task execution, and physical behaviour of virtual objects and haptics sensation play important roles in human perception. In recent years, many researchers have made a great effort to integrate haptics into virtual assembly environment, and several prototyping systems have been also developed. A system for haptic assembly and realistic prototyping (Seth *et al.*, 2006) was developed by VR applications center of Iowa State University. This system was built based on VEGAS, and used haptics to realize physically based modeling to simulate realistic part-to-part and hand-to-part interactions. A dual-handed haptic interface for a realistic part interaction using PHANTOM devices was presented, and the swept volumes were implemented for addressing assemblability and maintainability issues. A collaborative haptic assembly simulator (Iglesias *et al.*, 2008) developed by Labein Tecnalia in Spain, was a new peer-to-peer collaborative virtual assembly system. The users could simultaneously carry out assembly tasks using haptic devices even though they were not in the same place. Two major challenges had been addressed: virtual scene synchronization (consistency) and the provision of a reliable and effective haptic feedback. A consistency-maintenance scheme was designed to solve the challenge of achieving consistency, and a force-smoothing algorithm was developed to improve the quality of force feedback under adverse network conditions. A haptically enabled interactive and immersive VR system (Bhatti *et al.*, 2009) was developed by Deakin University of Australia. Unlike existing VR systems capable of providing knowledge about assembly sequences only, the presented system tried to imitate real physical training scenarios by providing comprehensive user interaction, and simulating constrained dynamics within the physical limitations of the real world imposed by the haptics devices within the virtual environment. Owing to its high physically interactive nature, the proposed system helped in procedural learning and skill development. Garbaya and Zaldivar-Colado (2009) studied how to model dynamic behaviour of mechanical parts during the execution of virtual assembly operation. The concept of visual dynamic behaviour representing the manipulation of real parts was developed, and the concept of spring-damper model was adopted to preclude the interpenetration of parts during the mating phase. Vo *et al.* (2009) investigated the benefits of haptic-based interaction for performing assembly-related tasks in virtual environment. Compared with visual-only methods, quantitative results showed that haptic-based interaction was beneficial in improving performance by reducing completion time for weight discrimination, generating higher placement accuracy to position virtual objects and guiding steadier hand motions along 3D trajectory. Christiand *et al.* (2009) described an enhanced haptic assembly simulation system, in which an optimal assembly algorithm was used to provide optimal paths for haptic guidance as well as an assembly sequence of the parts to be assembled. Experimental results showed that the haptic-path sequence-guidance (SG) mode gave the best performance improvement in terms of accumulated assembly time (28.56 percent) and travel distance (15.64 percent) compared to the unguided mode, while the SG mode alone increased performance by 16.91 percent for assembly time and 11.66 percent for travel distance.

Although these studies have made evident progress in this area, the development of a realistic virtual assembly environment with haptics is still a challenging problem because of the complexity of the physical processes and the limitation of natural interaction. Dynamic simulation of part physical behaviour, real-time collision detection with high precision parts, guarantee of haptic feedback at a high update rate, all these problems are attracting more and more researchers to study. In this paper, a hybrid approach combining with physics modeling and haptic feedback and geometry constraint is realized to simulate and guide the realistic assembly process.

System architecture design

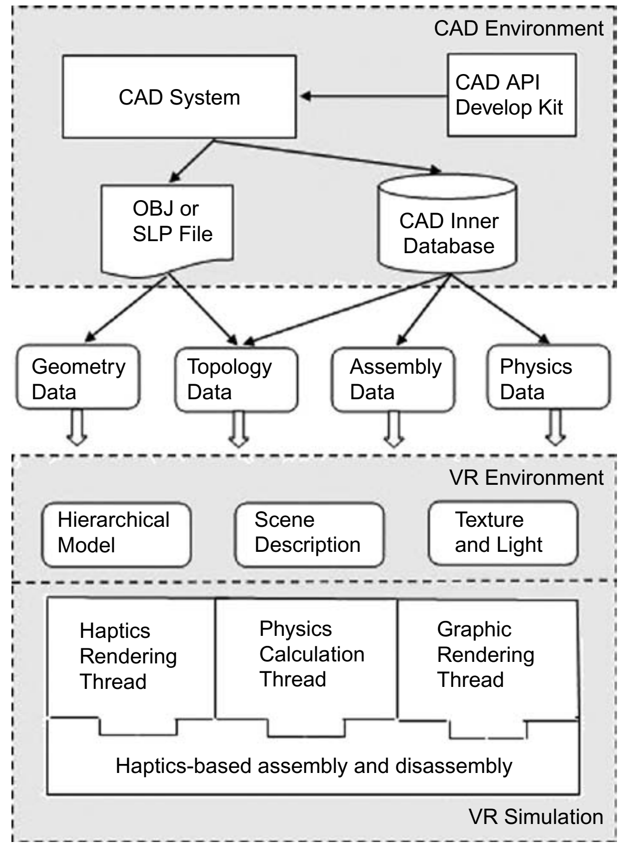
The overall architecture of the HVAS can be represented by the diagram shown in Figure 1. According to the function requirements, the system can be divided into three parts, which includes CAD environment, VR environment and VR simulation.

For the CAD environment, the product models, tools and fixtures models are designed in a commercial CAD system such as Pro/ENGINEER or SolidWorks. Because of the difference of data expression format between CAD system and VR system, for example in CAD system a part can be expressed by precise mathematic method using CSG or B-Rep, however, in VR environment the models are usually expressed only by polygons, which cannot be used for virtual assembly operation, planning and evaluation. An automatic data integration interface should be developed to finish data transformation from CAD to VR. By using CAD API development kit, the related information and data can be extracted from CAD neutral files (OBJ or SLP) and CAD inner database. Four types of data are mainly taken into account, including geometry data, topology data, assembly data and physics data. These models and data are then input into VR environment.

For the VR environment, three components are necessary to construct haptic-based virtual assembly environment. After geometry data, topology data, assembly data and physics data are transformed from CAD to VR, a hierarchical data model was designed to represent and reorganize these models and data in virtual environment, and then an efficient scene graph structure was also designed to display and manage virtual objects. At the same time, virtual factory, texture, light and surrounding environment were also set up to simulate real production scenarios.

For the VR simulation, the main objective is to carry out assembly or disassembly operations by haptic device, to give the user realistic force and tactile feedback as in real life, and

Figure 1 System structure of HVAS



to simulate natural human-computer interaction and dynamic physical-based behaviour. The kernel of VR simulation is a multi-thread method to realize graphic display, collision detection, constraint recognition, physics computation, dynamic simulation and haptic feedback.

There are three separate threads in the system: haptic rendering thread, physical calculation thread and graphical rendering thread. The haptic rendering thread is responsible for communicating with the PHANTOM device, launching at a high priority and high frequency (about 1,000 Hz). There are two components in this thread: repulsive force computation model and guiding force computation model. The repulsive force is a resistance force generated by penetration depth using Hook law, and the guiding force is an attractive force generated by geometry constraint to aid for assembly operation and positioning. At each simulation loop, the position and orientation of the virtual probe attached to the stylus, which is often referred to as the haptic interface point (HIP) are obtained, and this data are used to control the position and orientation of the operated virtual object which is connected to the haptic device. When the operated virtual object collides with the other virtual objects, the penetration depth can be calculated and a feedback force can be generated to the user. When the related two objects are close enough and a geometry constraint is captured, a guiding force can be generated and sent back to the user to help for inserting and mating operation. The physical calculation thread performs all the work including collision detection, physics computation, dynamic simulation of realistic part behaviour, and geometry constraint recognition and solution, etc. which is running at a second priority and frequency (about 100 Hz). When two parts contact each other, a real-time collision-detection algorithm is supported to compute collision position, direction and penetration depth, and a spring-mass model is used to simulate dynamic behaviour of virtual object. When two parts are close to each other, a real-time computation about the distance and angle of related assembly features are carried out, and if they come into a certain range, a geometry constraint can be automatically captured, and the precise assembly position and orientation can be calculated. The graphic rendering thread is mainly responsible for visualizing the entire scene graph and virtual objects, and it runs at a low frequency about 30 Hz.

Because there are several threads in the system, and each of them runs at a different rate, an efficient communication mechanism must be provided. As shown in Figure 2, when the user operates the haptic device to move an object, the graphic rendering thread gets the new move command from the device, and then it sends the new position of HIP for the operated virtual object to the physics calculation thread, at the same time, the position of the other virtual objects can be also obtained from graphic rendering thread. In the physics computation thread, a real-time collision algorithm is provided to detect if the virtual objects contact each other. If they are in contact state, a spring-mass model is used to calculate the new position of virtual object to simulate the dynamic behaviour. If they are in assembly state, a geometry constraint recognition and resolution algorithm is used to calculate the new position of virtual object for assembly operation. After calculation, the validated new position can be sent back to the graphic rendering thread to update the virtual objects. At the same time, the collision position, direction, penetration depth and geometry constraint information can be sent to haptic rendering thread to calculate the feedback force.

Construction of virtual environment

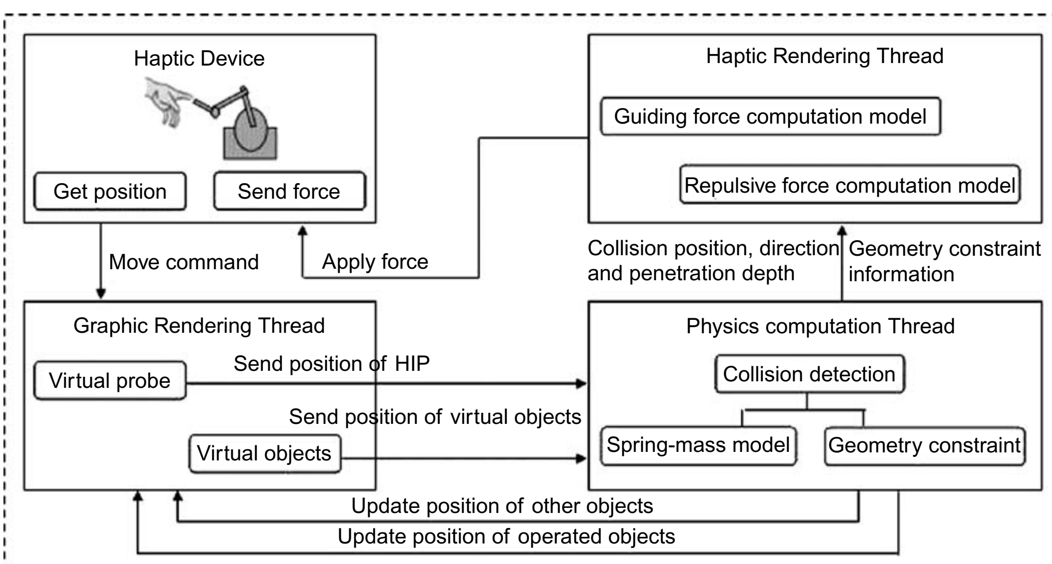
A data integration interface is developed to finish the data transformation from CAD to VR. We divide CAD model data into four types: geometry data, topological data, assembly data and physical data. The geometry data are used for model display and collision detection in virtual environment, and the topology data are used to rebuild the hierarchical mapping relation of product, part, feature, surface and polygon. A part is composed of assembly features, a feature is the aggregation of geometry surfaces and a surface is the aggregation of polygons. The polygons are the mesh unit to visualize in virtual environment and the surfaces are the foundation to define geometry constraints such as against, collinear, concentric, etc. Each part object, feature object, surface object and polygon object has a unique identity number for recognition in virtual environment. The assembly data mainly include assembly constraint, assembly position matrix and assembly tolerance, which is used for constraint recognition and precise positioning. The physical data contain all the physical properties of an object such as mass, material, inertia, surface friction and surface hardness, etc. These four types of data can be extracted from CAD system inner database or neutral files, respectively, by different approach using API redevelop tool kits, and then input into VR environment.

As shown in Figure 3, a hierarchical constraint-based data model is designed to represent and reorganize these data and information in virtual environment, which is composed of product layer, subassembly layer, part layer, feature layer, surface layer and polygon layer. For the elements in different layers, there exist the hierarchical mapping relationships, and for the elements in the same layer, there exist the constraint relationships. This data model can be described by the following method.

The product layer can be described by $Z = (D)$, where D is the product object, including product name, product ID and the other design and management information.

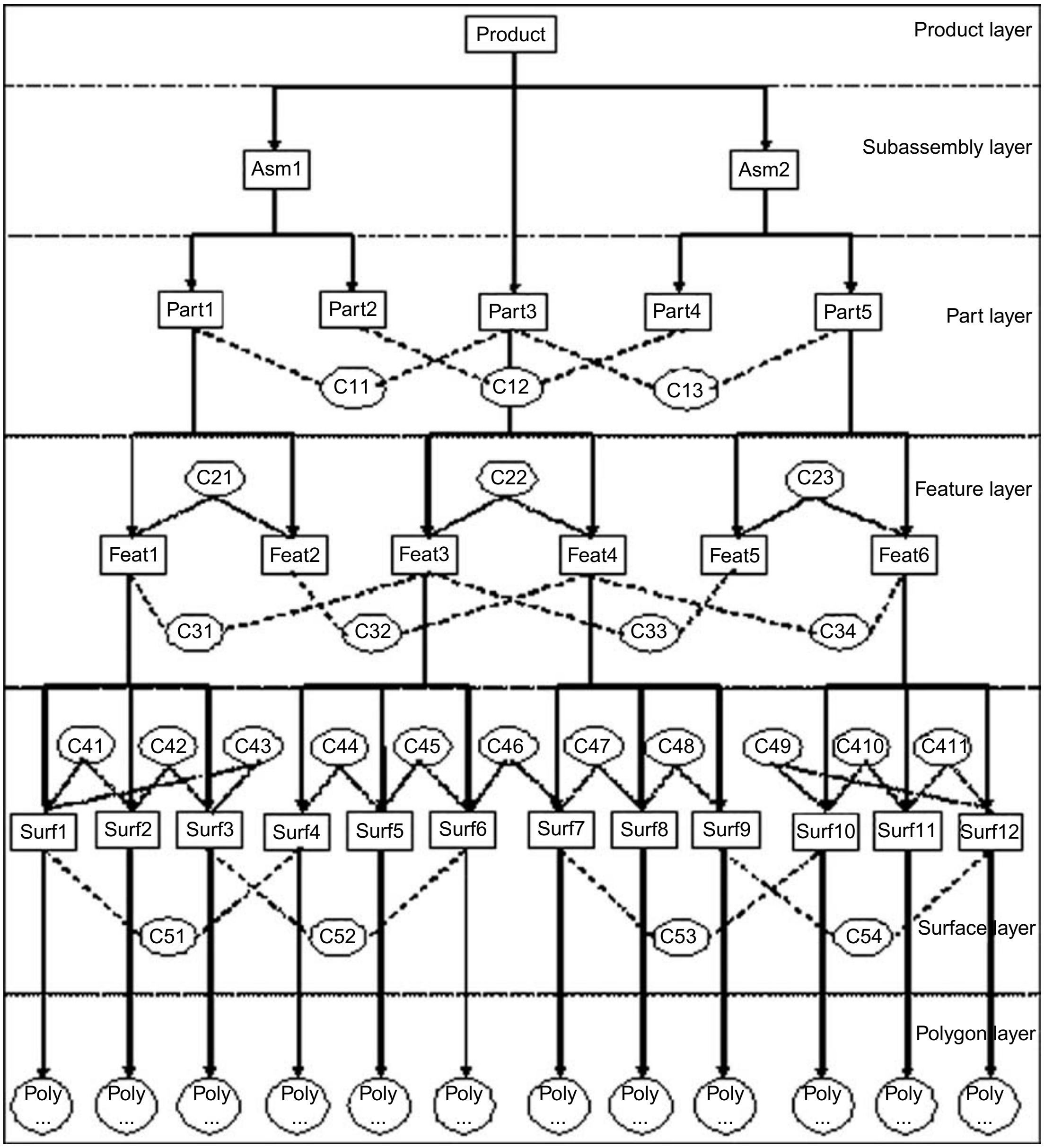
The subassembly layer can be described by $B = (A, H^A, H^Z)$, where A is the subassembly object, H^A is the hierarchical mapping relationships between subassembly objects, because a subassembly object may be composed of other subassembly objects. H^Z is the hierarchical mapping relations between product and subassembly objects.

Figure 2 Framework of multi-thread communication



Assembly Automation
Volume 31 · Number 4 · 2011 · 369–384

Figure 3 The hierarchical constraint-based data model



The part layer can be described by $L = (P, H^B, H^Z, C^P)$, where P is the part object, H^B is the hierarchical mapping relationships between the subassembly and part objects and H^Z is the hierarchical mapping relationships between product and part objects, because a part may belong to a subassembly, or belongs to the product directly. C^P is the constraint relationships between part objects, such as the C_{11}, C_{12}, C_{13}, etc. and they can be understood as the assembly constraints which reflect the spatial relations among individual parts, i.e. axis-hole assembly constraint and face-mating assembly constraint, etc.

The feature layer can be described by $T = (F, H^L, C_1^F, C_2^F, HC^{F-P})$, where F is the feature object, and H^L is the hierarchical mapping relationships between part and feature objects. C_1^F is the inner constraint relationships between the features of the same part, such as C_{21}, C_{22}, C_{23}, etc. which are used to define the part shape and structure. C_2^F is the external constraint relationships between the features of different parts, such as C_{31}, C_{32}, C_{33}, C_{34}, etc. which are mainly pointed to assembly constraint relationships. Because the part object is composed of features, the constraint relationships between parts (C_{11}, C_{12}, C_{13}, etc.) can be hierarchically decomposed as the external constraint relationships between features (C_{31}, C_{32}, C_{33}, C_{34}, etc.).

The surface layer can be described by $J = (S, H^T, C_1^S, C_2^S, HC^{S-F})$, where S is the surface object, and H^T is the hierarchical mapping relationships between feature and surface objects. C_1^S is the inner constraint relationships between the surfaces of the same part, such as C_{41}, C_{42}, C_{43}, etc. which are used to define the feature shape and structure. C_2^S is the external constraint relationships between the surfaces of different parts, such as C_{51}, C_{52}, C_{53}, C_{54}, etc.

Assembly Automation
Volume 31 · Number 4 · 2011 · 369–384

which are mainly pointed to assembly constraint relationships between surfaces, such as parallelism, coincidence, perpendicularity, alignment and co-edge, etc. Because the feature object is composed of feature elements such as geometry surfaces, the external constraint relationships between features can be also decomposed as the external constraint relationships between surfaces.

The polygon layer can be described by $M = (O, H^J)$, where O is the polygon object, and H^J is the hierarchical mapping relationships between surface and polygon objects. For each surface objects, it is composed of many polygons, which are used for real-time graphic display and precise collision detection in virtual environment.

Generally, for the hierarchical constraint-based data model, the inner constraints are only used to maintain object inner structure and shape, and the external constraints are used to define the relative position and assembly relation between different objects, and they are the main geometry constraints which should be taken into account for constraint recognition and assembly operations.

According to the hierarchical constraint-based data model, a hierarchical scene graph can be also designed as shown in Figure 4. The root node of the scene graph includes light node, virtual factory node, product node and tool node, and the product node is organized hierarchically according to subassembly node, part node and surface node. Each separate node has its transform node and geometry node, the transform node is used to control the position and direction of the object, and the geometry node is used to display the object in virtual environment. The scene graph has several advantages. First, it can be integrated with the data model directly, in virtual environment when the product models are loaded, the scene graph structure can be generated automatically. Second, the user can select different layer object to operate, for example, the user can operate a part object to assembly or disassembly, and he can also select a subassembly object as a whole to assembly or disassembly, and each layer object has its own position, orientation and physics information, etc. Finally, geometry surface is separated from the model as an intermediate layer object, which provides convenience for collision detection, geometry constraint recognition and feedback force computation.

Haptics-based assembly modeling and dynamic simulation

The overview of haptic-based assembly process is shown as Figure 5, which can be divided into two stages: contact simulation state and assembly simulation state. Contact state is mainly referred to simulate the collision reaction and dynamic behaviour of virtual part, and assembly state is mainly referred to simulate the mating or insertion phase of virtual part. In virtual environment, when the two parts are close to each other and the distance and orientation of their assembly features reach a specified range, the assembly simulation state can be activated. Otherwise, the contact simulation state is executed. The realization of haptic-based assembly process can be described by the following steps:

- *Step 1*. Operating the virtual object by haptic device. When the user moves the haptic device, the position data can be obtained and mapped from haptic workspace to virtual environment workspace, and then used to operate the virtual object. The user can move or rotate the virtual part in virtual environment and feel its gravity feedback.
- *Step 2*. Judge the state. We divide the whole process into two stages: contact simulation state and assembly simulation stage. When the user operates the virtual part and it does not reach the ready state for assembly, execute the contact simulation, turn to step 3. Otherwise execute the assembly simulation, turn to step 5.
- *Step 3*. If the contact simulation state is activated, a real-time collision-detection algorithm is supported to calculate the collision position, direction and penetration depth, and then the collision force and torque can be computed and sent back to the user to get the sensation of realism.
- *Step 4*. A spring-mass model is used to simulate the dynamic contact behaviour of virtual parts. A virtual coupling method is applied to supply a dual representation for the operated part: the tracked model and the displayed model. The tracked model of the part is invisible for the user and controlled by the PHANTOM device to calculate its position and orientation, and then use this data for collision detection and penetration depth calculation. The displayed model is used for graphic display and visible to the user. A spring-mass model is used to couple the tracked model and the displayed model. The realistic manipulation of parts is obtained by displayed model to visually prevent 3D models passing through each other, and the collision force is computed using the penetration depth of the tracked model of the operated part into the other parts. So when the user moves a part, the displayed model tends to follow the tracked model to simulate the dynamic interaction process by the spring-mass model, and collision detection and the associated response is simulated to prevent the displayed model of the part from penetrating into other objects in virtual environment.
- *Step 5*. If the assembly simulation state is activated, a geometry constraint recognition and resolution algorithm is supplied to calculate the precise assembly position for the operated part. In order to avoid unnecessary computation and improve system efficiency, the collision-detection algorithm is closed. The virtual part is adjusted to the precise position by geometry constraint, not by collision detection. This is very useful for low-clearance parts and complex products, because positioning a part only by high-accuracy collision detection is a time-consuming and low-efficiency process.
- *Step 6*. During assembly simulation process, a real-time guiding force or repulsive force can be generated to make the user feel that he is assembling the virtual part as he would the real products. Although the collision detection is closed, combining haptic feedback with geometry constraint is a good way to realize the interaction process as natural and intuitive as in real life. According to the geometry constraints, a guiding force can be generated to help the user to assemble the part along the free-collision path to the correct position. If the user wants to deviate this position, a repulsive force or torque can be also generated to prevent the user's action just as he feels the collision and obstruction in real life.

Collision detection and dynamic simulation

A hierarchical collision-detection algorithm is developed based on the multi-thread mechanism and hierarchical data

Assembly Automation
Volume 31 · Number 4 · 2011 · 369–384

Figure 4 The structure of scene graph

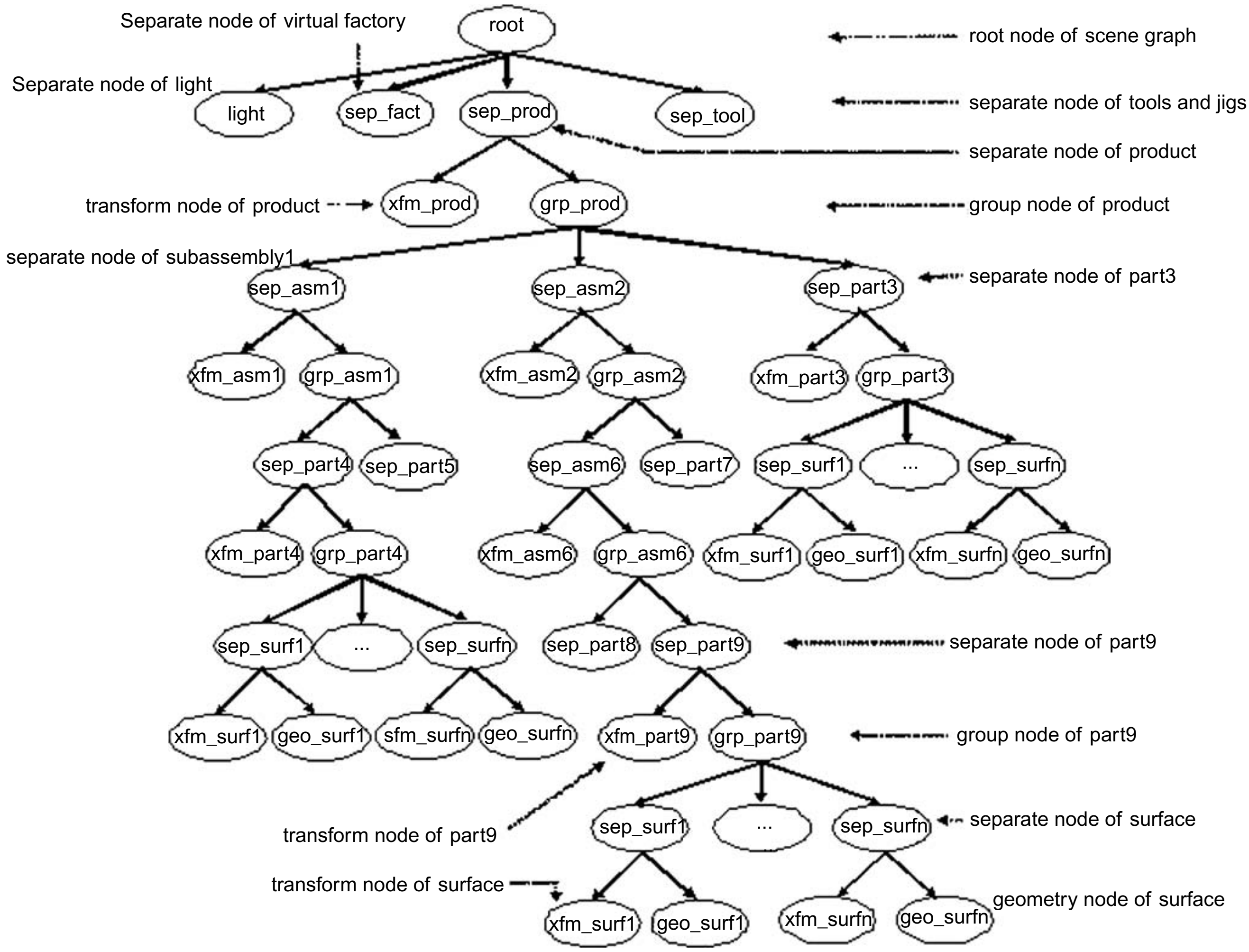
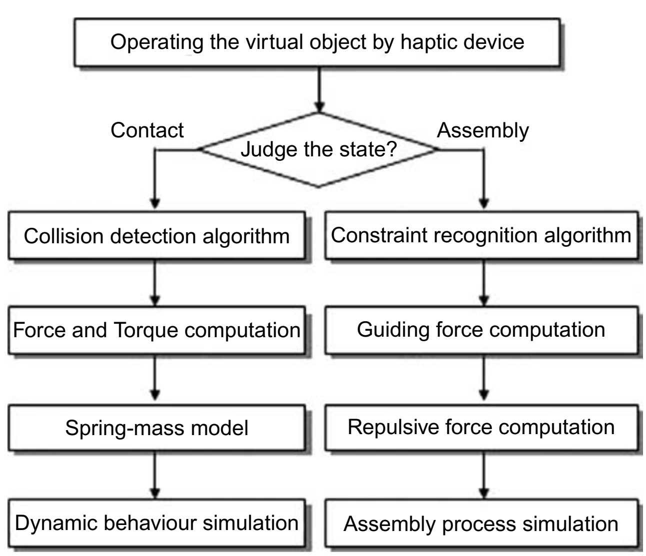


Figure 5 Overview of the haptic-based assembly process



model. Before the collision detection, a space subdivision of part bounding box is prepared. For example, the part bounding box can be divided into several small space cells (e.g. $3 \times 3 \times 3$). A mapping relation between geometric surfaces and space cells can be built. A space cell can include one or several geometric surfaces, and a geometric surface can also belong to one or several space cells. First, collision detection between part bounding boxes is carried out. Second, if the part bounding boxes collide into each other, further collision detection between bounding boxes of space cells is carried out. Third, if two space cells collide into each other, further collision detection between the bounding boxes of geometry surfaces, which belong to these space cells is carried out. Only when two geometry surface bounding boxes collide into each other, the detailed surface data can be sent to the physics thread for precise collision detection between polygons. The multi-thread mechanism can be used to improve the algorithm efficiency and performance. Collision detection between part bounding boxes, space cell bounding boxes and surface bounding boxes is finished in graphic rendering thread, because this detection is a time-consuming process and the graphic rendering thread runs at a lowest rate. After this computation, most of the detection

Assembly Automation
Volume 31 · Number 4 · 2011 · 369–384

objects can be filtered out and only a small amount of surface data are sent to physical computation thread for precise collision detection. In physical computation thread, real-time collision detection between polygons is carried out and the collision position, normal and depth can be calculated, and this data can be sent to the haptic rendering thread to compute the collision force and torque and then fed back to the user for haptic rendering.

A spring-mass model is used to simulate the dynamic behaviour of the colliding part, which was first developed by Borst and Indugula (2006) and then used by Garbaya and Zaldivar-Colado (2007), as shown in Figure 6. When the user moves the haptic device, the tracked model is controlled by the haptic device and then this model can be used for collision detection and force or torque calculation. The displayed model is controlled by a spring-mass model, which includes a linear spring and a torsional spring. The linear spring is used to apply a translational force F to the displayed model in order to follow the tracked model by performing translational movement. The torsional spring is used to apply a torque T to the displayed model in order to keep the orientation of displayed model relative to the tracked model. By applying the linear force F and the torque T to the displayed model, it keeps following the tracked model moved by the operator without the interpenetration of parts, and the displayed part can obtain dynamic behaviour that mimics the real-world interaction of parts.

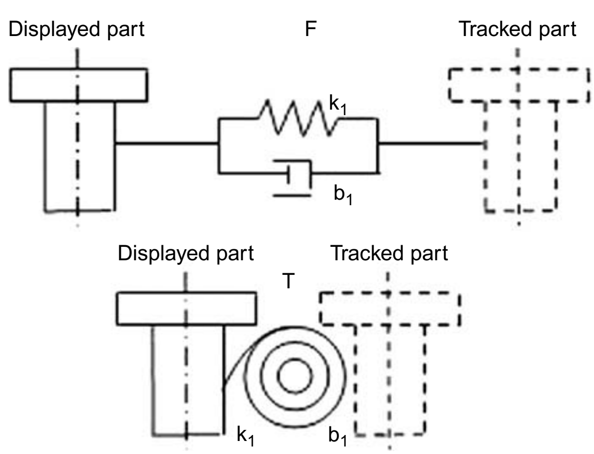
The spring-mass model can be described by the following formulas:

$$F = k_T(p_t - p_d) - b_T(v_t - v_d) \tag{1}$$

$$T = k_R \times \theta_{t-d} - b_R \times \omega_{t-d} \tag{2}$$

k_T is the linear spring stiffness constant, b_T is the linear damping constant. p_t and p_d are the positions of center of mass for tracked model and displayed model in the world coordinate frame. v_t and v_d are the linear velocities of the tracked model and displayed model. k_R is the torsional spring stiffness constant, b_R is the torsional damping constant. θ_{t-d} is the rotation angle of displayed model relative to tracked model in world coordinate frame, and ω_{t-d} is the rotation angle velocity of displayed model relative to tracked model.

Figure 6 Spring-mass model between tracked and displayed part



Haptic feedback and assembly simulation

In virtual assembly environment, a hybrid approach combining haptic feedback with geometry constraint is used to avoid unnecessary collision and computation. When the assembly simulation state is activated, a geometry constraint can be captured between the operated part and the related part. For example, a face-mating geometry constraint can be recognized between two planes when the angle between their normals is less than the angular tolerance and the distance between the planes is less than the linear tolerance. An axis-mating geometry constraint can be recognized between two cylinders when the angle between their axes is less than the angular tolerance and the distance between the axes is less than the linear tolerance.

As shown in Figure 7, a peg-hole mating is used as an example to discuss the realization of geometry constraint and haptic feedback. By using the haptic device, the user operates the axis-part, which is composed of five surfaces, to mate it into the hole of the base part, which is composed of four surfaces. According to the hierarchical constraint-based data model, the surface is designed as an independent object, and in the scene graph it is constructed as the separated node. In virtual environment, each surface can be represented by specific elements that define its parametric equation. For example, a point on the plane and its normal vector defines a planar surface while a point on the cylinder axis, its direction and the radius value define a cylinder surface. Besides surface specific elements, each surface has a bounding box to define its borders. In the scene graph structure, for each separated surface node, it has a transform node (beginning with xfm_) to control its position and orientation, and it has also a geometry node (beginning with geo_) for graphic display and collision detection. This hierarchical data model and scene graph structure provides great convenience for geometry constraint computation and haptic feedback.

For the base part, a point on the hole axis and the vector of the hole axis can be represented by:

$$P_{base} = P'_{base} \times M_{C_{base}-C_{global}} \tag{3}$$

$$V_{base} = V'_{base} \times M_{C_{base}-C_{global}} \tag{4}$$

P'_{base} is the center point of the hole axis and V'_{base} is the direction vector of hole axis in the base part frame of reference, which can be exacted from the CAD system during data transformation. $M_{C_{base}-C_{global}}$ is the 4×4 transform matrix from the base part frame of reference to the global world frame of reference.

Similarly, for the operated part, a point on the axis and the vector of the axis can be represented by:

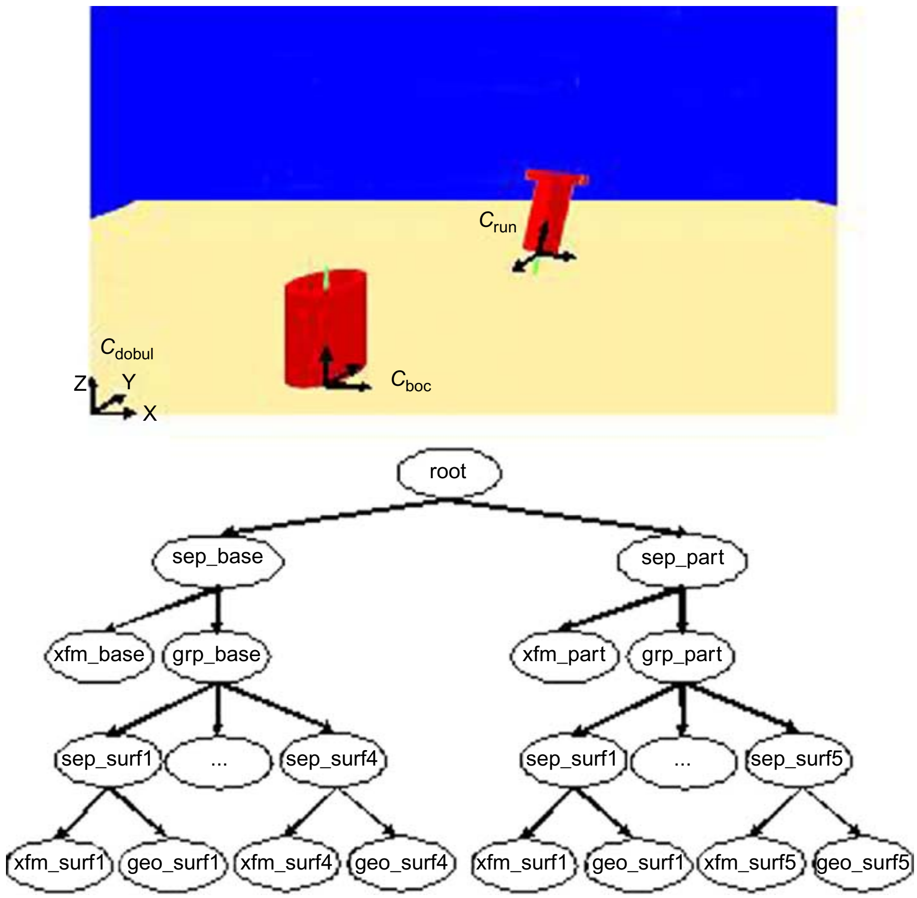
$$P_{part} = P'_{part} \times M_{C_{part}-C_{global}} \tag{5}$$

$$V_{part} = V'_{part} \times M_{C_{part}-C_{global}} \tag{6}$$

P'_{part} is the center point of the axis and V'_{part} is the direction of axis vector in the operated part frame of reference, and they can be also read from the CAD system. $M_{C_{part}-C_{global}}$ is the 4×4 transform matrix from the operated part frame of reference to the global world frame of reference.

For axis and hole mating, the assembly process can be divided into two phases. The first phase is adjusting the operated part to make their mating axis align, as shown in Figure 9(a). The second phase is inserting the operated part

Figure 7 Axis-hole mating and scene graph



into the hole to get to its final position, as shown in Figure 9(b). Then the precise position of the operated part can be calculated by geometry constraint resolution.

The angle of the two parts can be computed by the direction vectors of the mating cylinder axis:

$$\alpha = \arccos(V_{base} \cdot V_{part}) \tag{7}$$

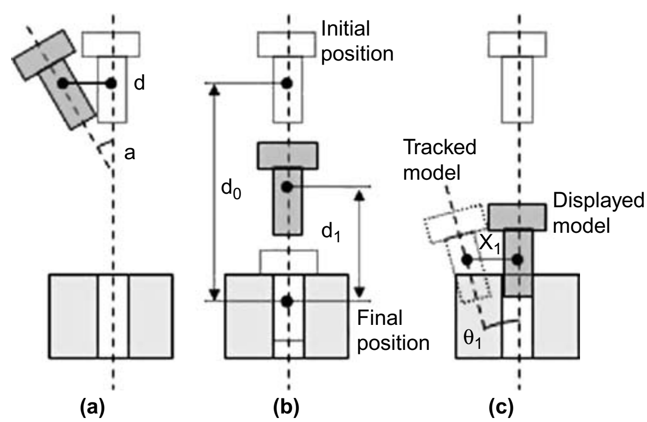
The distance can be computed from the center point of the cylinder axis of the operated part to the beeline of the direction vector of the base part cylinder axis:

$$d = \frac{|V_{base} \times P_{base}P_{part}|}{|V_{base}|} \tag{8}$$

So, for the first phase in Figure 8(a), in order to adjust the operated part to the correct position, a rotation matrix M_R which rotates the operated part around vector $n = (V_{base} \times V_{part})$ via point P_{part} must be calculated and applied at first to make the axis parallel, and then a translation matrix M_T can be calculated and applied to translate the operated part along distance d to make the axis align. This work is finished by the physics calculation thread in real time, and the detailed realization of the rotation matrix M_R and translation matrix M_T can be found in Yao *et al.* (2006).

After axis alignment, the user can operate the haptic device to simulate the assembly process, and haptic feedback is an important cue to assist the operator in finding appropriate position and orientation. According to geometry constraint,

Figure 8 Haptic-based assembly process for axis-hole mating



two types of force feedback can be generated: the attractive force and the repulsive force. Similarly, a dual-model representation mechanism can be also used during this mating phase: the tracked model and the displayed model. First, after axis alignment, the tracked model and the displayed model are put at the same position. But then when the user operates the haptic device, the tracked model is controlled by the haptic device, and the displayed model is controlled by geometry constraint. As shown in Figure 8(b) and (c), the tracked model may be deviated from the mating cylinder axis, but because of geometry constraint, the DOF of the displayed model is restricted and it can be only moved along the mating cylinder axis. In each simulation loop, we use the haptic data to update the tracked model, and we use geometry constraint to calculate and update the corresponding displayed model.

During mating phase, an attractive force can be generated to guide the user to move the operated part along the constrained direction:

$$F^i_{att} = K_1 \times \frac{d_i}{|d_0|} \times F_{\max} \tag{9}$$

K_1 is a stiffness constant, $F_{\max}$ is the maximum force value defined by haptic device, d_0 is the distance from the initial position of displayed model after axis alignment to the final position of displayed model, d_i is the distance from current position of displayed model at the time t_i to the final position of the displayed model:

$$d_0 = P^0_d - P^f_d \tag{10}$$

$$d_i = P^i_d - P^f_d \tag{11}$$

P^0_d and P^f_d are the initial position and final position of center point of the cylinder axis on display model of the operated part, P^i_d is the current position of center point of the cylinder axis on display model at the time t_i.

When the tracked model is deviated from the mating axis, a repulsive force or torque can also be generated to prevent the user's action, this make the user feel the geometry constraint and also make the mating process as natural and realistic as in real life. The repulsive force can be calculated by:

$$F^i_{rep} = K_2 \times x_i \tag{12}$$

K_2 is a stiffness constant, and x_i is the distance from the mating cylinder axis on the tracked model to the mating cylinder axis on the displayed model:

$$x_i = P^i_t - P^i_d \tag{13}$$

P^i_t and P^i_d are the center point of mating cylinder axis on tracked model and displayed model, respectively, at the time t_i. The repulsive torque can be also calculated by:

$$T^i_{rep} = K_3 \times \frac{\theta_i}{\pi} \times T_{\max} \tag{14}$$

K_3 is a stiffness constant, $T_{\max}$ is the maximum torque value defined by haptic device, θ_i is the angle between the mating cylinder axis on tracked model and displayed model:

$$\theta_i = \arccos\left(V^i_t \cdot V^i_d\right) \tag{15}$$

V^i_t and V^i_d are the direction vector of mating cylinder axis on tracked model and displayed model at the time t_i.

Implementation and application

An HVAS has been developed and applied to a pump product on a HP graphic workstation, with PHANTOM premium device providing haptic feedback and stereo glass for 3D display. WorldToolKit, a commercial VR software tool kit is used for the creation of virtual environment. AGEIA PhysX, a famous physics engine is used to calculate and simulate part's dynamic behaviour. OpenHaptics, the SDK tool kit from SensAble Technologies, is used to control and apply haptic rendering.

As shown in Figure 9, the product models are designed in commercial CAD system such as Pro/ENGINEER. After data transformation from CAD to VR, the geometry information, topology information, assembly information and physics information are inputted into virtual environment, and a hierarchical constraint-based data model can be constructed. The creation of hierarchical scene graph can be verified as shown in Figure 10, in virtual environment the user can select a single part to operate, he can also select a geometry surface to define assembly features, as shown in Figure 11. He can operate virtual objects by haptic device PHANTOM to get force and torque feedback, as he would when operating the part in real life. As shown in Figure 12, when the user moves the part in free space, he should feel gravity. When he rotates the part around an axis, he should feel torque feedback. When he moves the part along a face or along an axis, he should feel friction. When the part collides with the other objects, he should feel collision force and torque. As shown in Figure 13, if two virtual objects collide into each other, the spring-mass model is used to prevent part penetration and to simulate dynamic interaction behaviour. At the same time, the system real-time detects the distance and angle of related assembly features. If they meet constraint recognition condition, a geometry constraint can be captured, and a guiding force can be generated to help the user to adjust the part to the correct position. As shown in Figure 14, when the user inserts the part along the mating axis, a guiding force is generated to attract the part to the final position. Meanwhile, if the user wants to move the part to deviate the mating axis, a repulsive force is generated to prevent this action just as in real life. After the part has been assembled to the final position by haptic device, the assembly path can be generated and displayed to analyze the space and process. As shown in Figure 15, the assembly path can be displayed as discrete points or swept volume, and Figure 16 is the swept volume when the part rotates around a fixed axis.

In order to study the performance and usability of the proposed system and method, we perform an evaluation experiment. Ten postgraduate students from our group are chosen as users to do the experiment, eight men and two women, aged from 24 to 33 years. All the students do research work in VR or CAD/CAM, so they have basic understanding about the application. After receiving a brief presentation about the system and haptics, each of them spends about half an hour to practice virtual assembly operations until becoming familiar with the system. Then all the students carry out the same assembly task as shown in Figure 17 to assemble an axis-part to a hole-part, and the completion time can be automatically recorded via a clock function (in C++) in three different cases:

1. *Assembling by collision detection.* In this condition, only collision detection is available to assist the user in assembly. The user picks up the axis-part and aligns the direction with the hole. While inserting the axis into the hole, the axis stops as soon as it collides with the hole-part. In this case, the system does not provide any intuitive help to the user to facilitate assembly, so the user must align the part precisely to complete the assembly, which is extremely difficult with the precision of collision detection and interface hardware.

Figure 9 Product models in CAD system (Pro/ENGINEER)

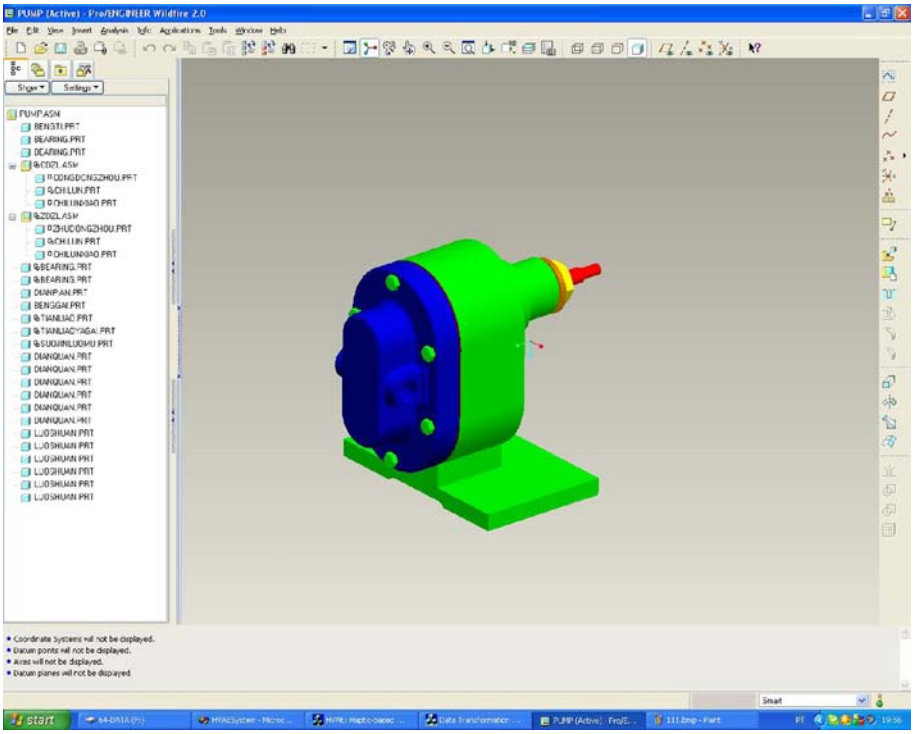
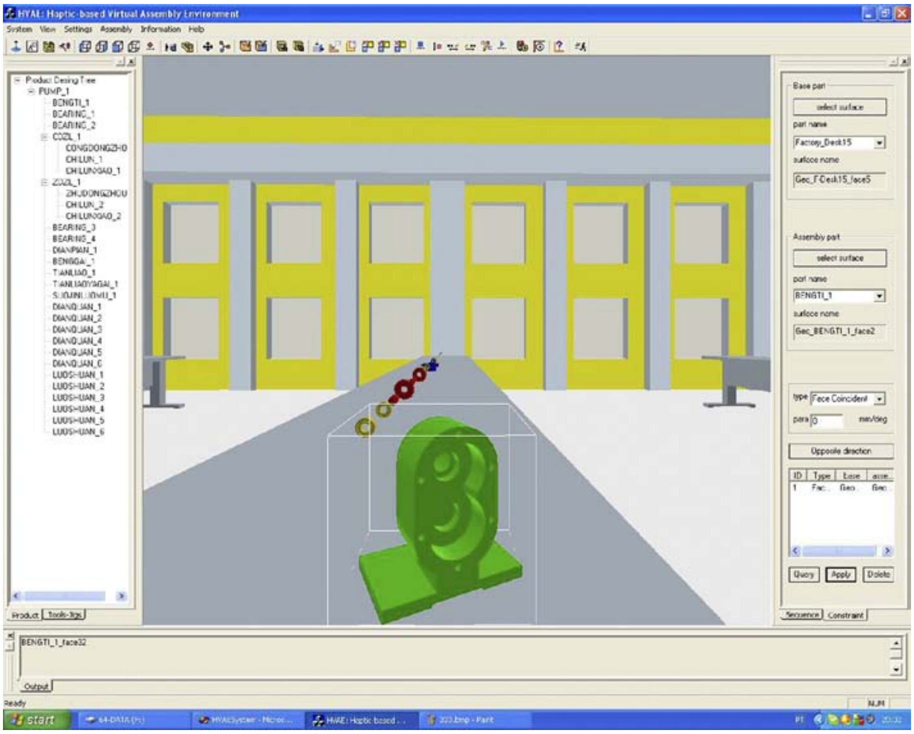


Figure 10 Creation of virtual scene after data transformation (select part)



2 *Assembling by geometry constraint*. In this case, geometry constraint-based method is used for assembling components. The user can manipulate and roughly align the models at first, then define the geometry constraint between cylindrical surfaces, and then apply a concentric constraint between these two surfaces and the part positions are updated such that the axis and hole are properly aligned with each other. The system can reduce the

Assembly Automation
Volume 31 · Number 4 · 2011 · 369–384

Figure 11 Creation of virtual scene after data transformation (select surface)

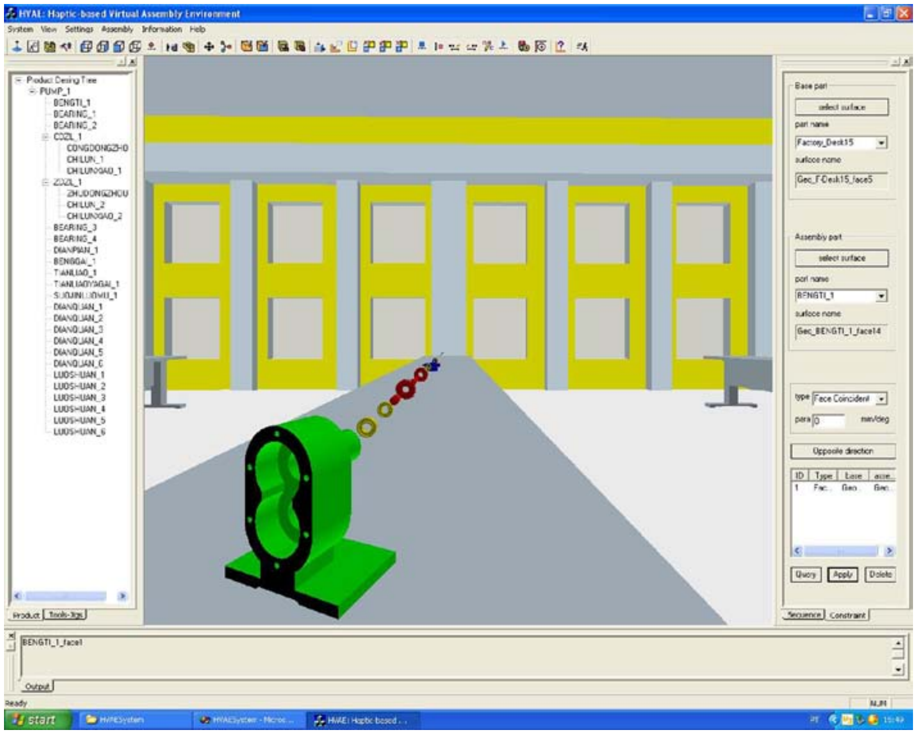
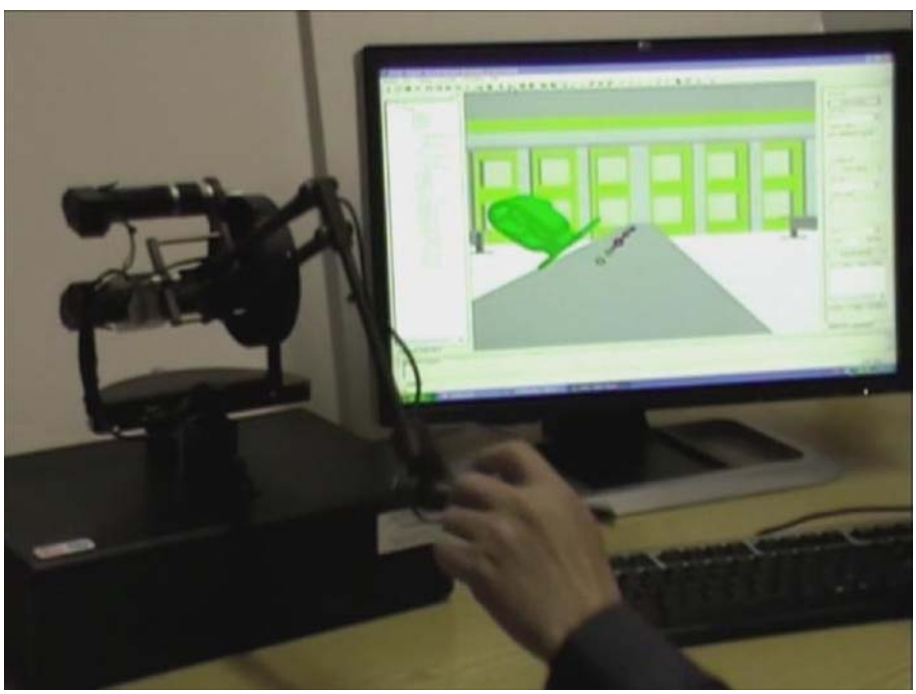


Figure 12 Manipulating virtual objects by haptic device



degrees-of-freedom of the axis-part such that it can only move in and out of the hole and rotate about its axis. Without the presence of collision detection, the parts can interpenetrate each other making the simulation unrealistic.

3 *Assembling by physics modeling and geometry constraint and haptics*. This is the method proposed in this paper, the user is allowed to utilize collisions detection, geometry constraint, physics modeling and haptics feedback together to assemble parts, just as he manipulates the part in real world. Each student repeats ten times for the same task, and the completion time can be shown as Figure 18. From the results, the average time is 9.4 s

Figure 13 Collision detection and dynamic behaviour simulation in virtual environment

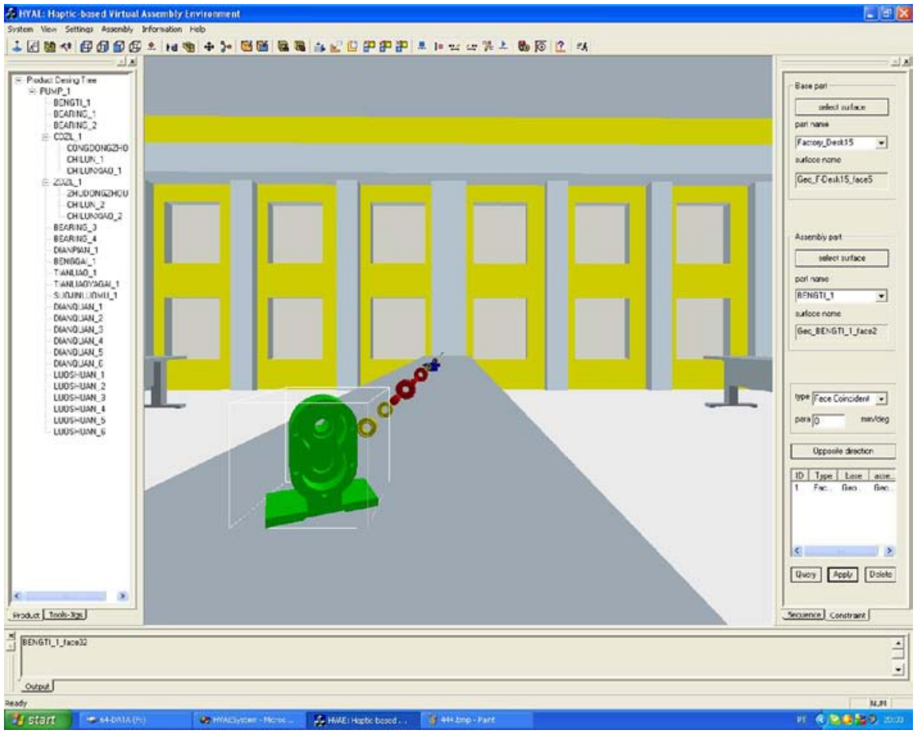
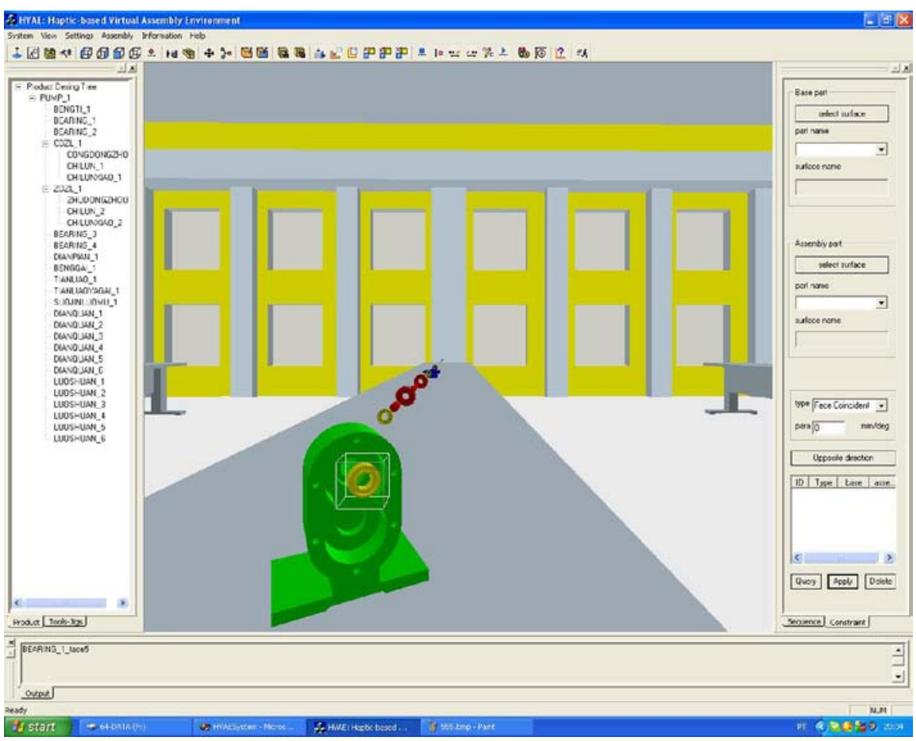


Figure 14 Assembling part with guiding force and repulsive force



in case 3, versus 12.6 s in case 2 and 29.8 s in case 1, the difference is 25.4 percent for case 2 and 68.5 percent for case 1, which shows that the method proposed in this paper provides a real gain in completion time.

In order to verify the stability of haptic rendering, we can also record the force and torque information during the assembly task for the last 3 s. Figure 19 shows the forces arisen by handling the axis-part to insert into the hole-part

Figure 15 Assembly path and swept volume

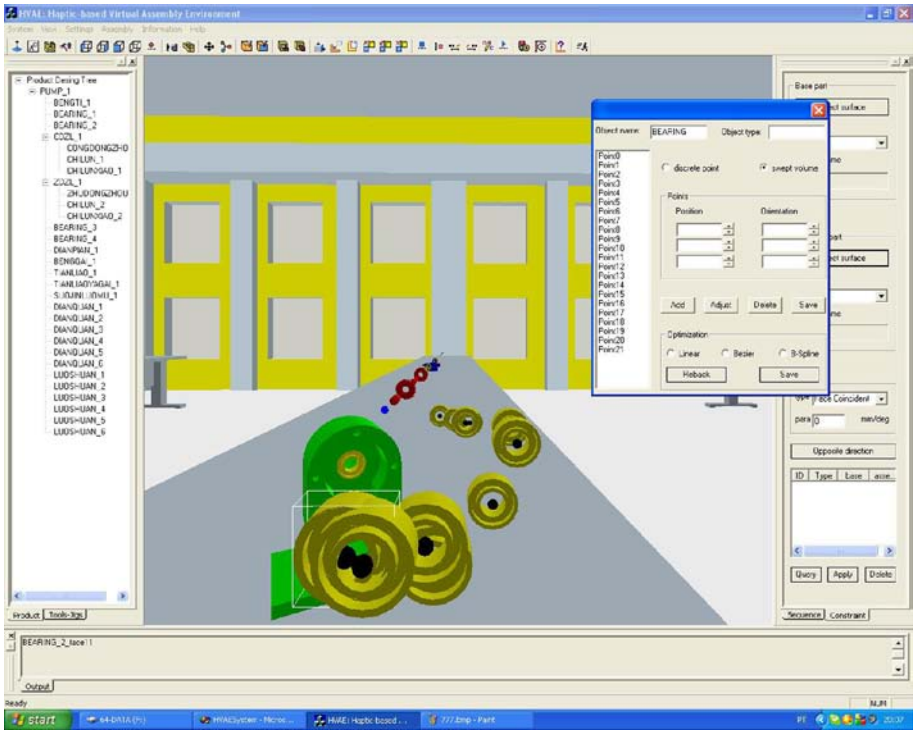
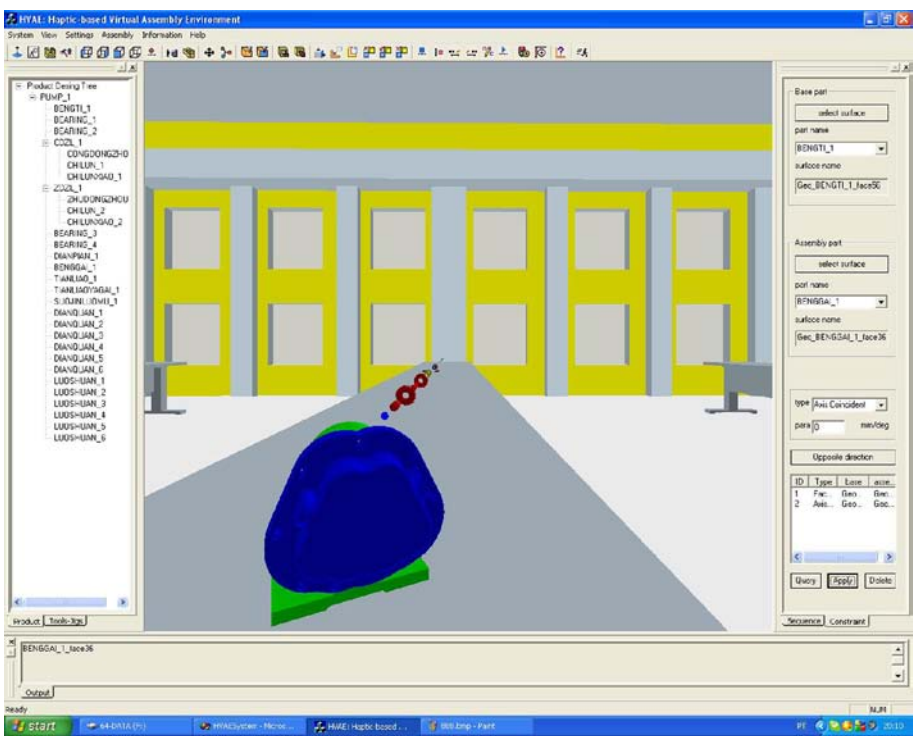


Figure 16 Rotating around a fixed axis and swept volume



after axis-align constraint. When the middle axis of axis-part deviates from the middle axis of hole-part, F_x, F_y, T_x, T_y are produced, and F_z is attracting force to guide the user to assemble the part to the final position. From the results, we can know that the computation algorithm is reasonable and the haptic rendering is stable.

Conclusion

Integrating haptics into virtual assembly is a promising and valuable application for mechanical products, which can result in faster product development process, faster identification of assembly and design issues and an efficient

Figure 17 Assembling an axis-part to a hole-part

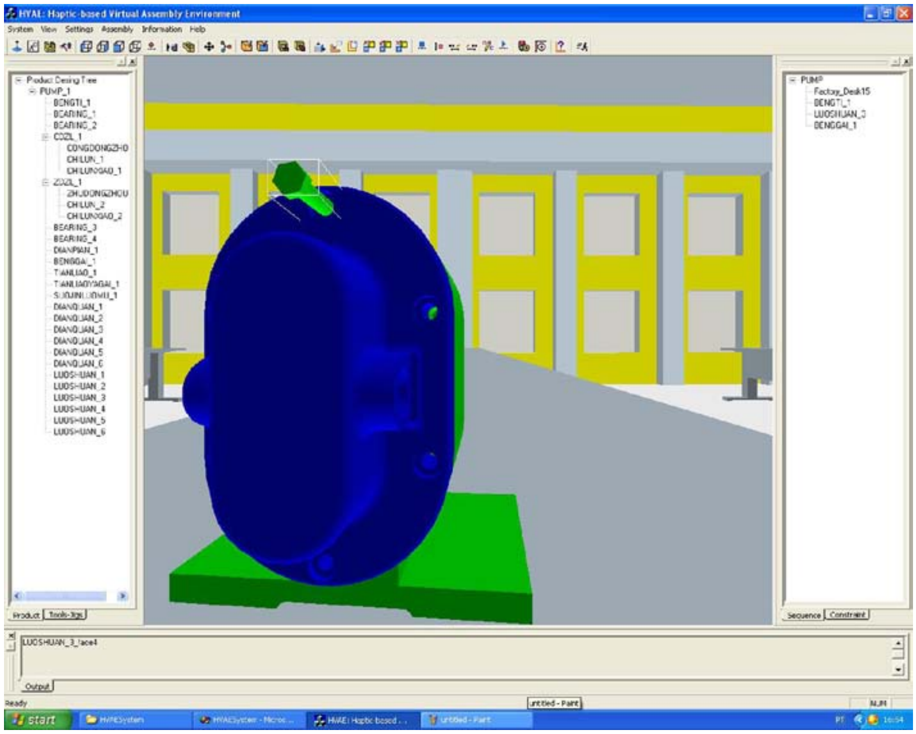


Figure 18 Average time completion through the assembly task

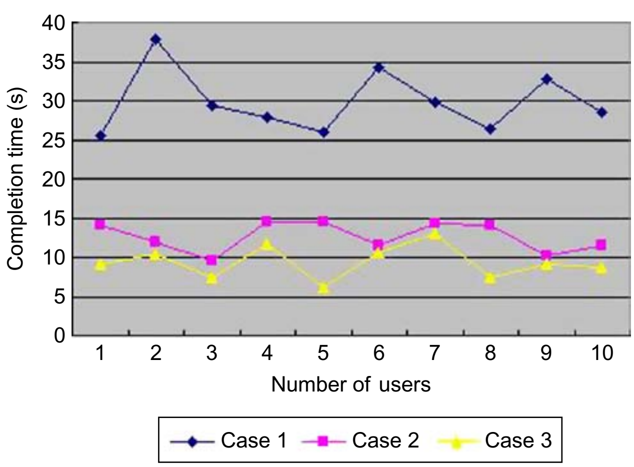
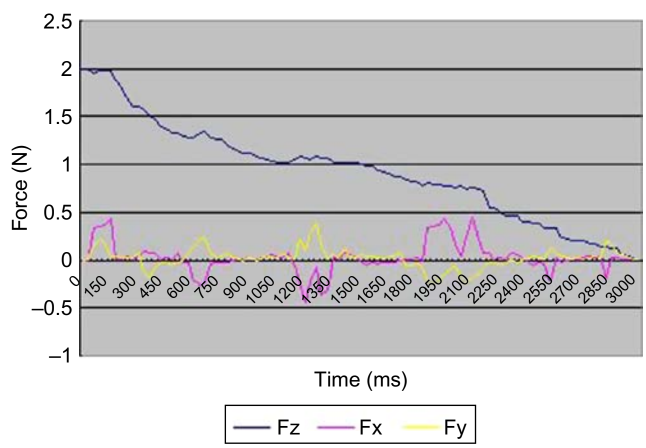


Figure 19 Forces and torques during the assembly task



and low-cost approach for assembly planning and training. In this paper, the design and implementation of an HVAS is investigated. The system can integrate with CAD models by an automatic data interface developed to transfer geometry information, topology information, assembly information and physics information from CAD to VR, a physics-based modeling approach with haptics feedback is undertaken to simulate the realistic assembly process. The innovative work of the system includes the following two aspects:

1 A hierarchical constraint-based data model and scene graph structure. A hierarchical constraint-based data model is designed to construct the virtual assembly environment, which is composed of product layer, subassembly layer, part layer, feature layer, surface layer and polygon layer. For the elements in different layers, there exist the hierarchical mapping relationships, and for the elements in the same layer, there exist the constraint relationships. According to the data model, a hierarchical scene graph can be automatically generated to organize and display the virtual objects. The user can select different layer object to operate, for example, he can operate a part object for assembly or disassembly, and he can also select a subassembly object as a whole for operation, or select a geometry surface to define assembly features.
2 A hybrid approach combining with physics modeling and geometry constraint and haptic feedback to realize and guide realistic assembly process. Unlike traditional virtual assembly systems based on collision detection or geometry

constraint only, a physics-based modeling approach with haptic feedback and geometry constraint is undertaken to simulate the realistic assembly process. When two parts collide into each other, the force and torque can be computed and fed back, and a spring-mass model is used to prevent penetration and simulate dynamic behaviour. When two parts are close enough to each other and the assembly simulation state is activated, a geometry constraint can be captured, an attractive force can be generated to guide the user to assemble the part along the correct position, and the repulsive force or torque can also be generated to realize the mating process as natural and realistic as in real life. In the future, the main objective is to extend the current system to the application of industrial complex products consisting of hundreds of parts. A further aim is to develop more efficient collision detection and physics modeling algorithms in order to realize more natural and intuitive interaction.

References

Bhatti, A., Nahavandi, S., Khoo, Y.B., Creighton, D., Anticev, J. and Zhou, M. (2009), "Haptically enable interactive virtual assembly training system development and evaluation", *SIMTECT 2009: Proceedings of the 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference, Adelaide, June 15-19*, pp. 1-6.

Borst, C.W. and Indugula, A.P. (2006), "A spring model for whole-hand virtual grasping", *Presence Teleoperators Virtual Environment*, Vol. 15 No. 1, pp. 47-61.

Christiand, Yoon, J., Auralius, M. and Yu, W. (2009), "An enhanced haptic assembly simulation system for the efficiency of assembly tasks", paper presented at the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St Louis, MO, October 11-15.

Garbaya, S. and Zaldivar-Colado, U. (2007), "The affect of contact force sensations on user performance in virtual assembly tasks", *Virtual Reality*, Vol. 11, pp. 287-99.

Garbaya, S. and Zaldivar-Colado, U. (2009), "Modeling dynamic behaviour of parts in virtual assembly environment", *Proceedings of the World Conference on Innovative Virtual Reality WIN VR'09, Chalon-sur-Saône, France, February 25-26.*

Iglesias, R., Casado, S. and Gutierrez, T. (2008), "Simultaneous remote haptic collaboration for assembling tasks", *Multimedia Systems*, Vol. 13, pp. 263-74.

Jayaram, S., Connacher, H.I. and Lyons, K.W. (1997), "Virtual assembly using virtual reality techniques", *Computer Aided Design*, Vol. 29 No. 8, pp. 575-84.

Jayaram, S., Jayaram, U., Wang, Y., Tirumali, H., Lyons, K. and Hart, P. (1999), "VADE: a virtual assembly design environment", *IEEE Computer Graphics and Applications*, Vol. 19 No. 6, pp. 44-50.

Johnson, T.C. and Vance, J.M. (2001), "The use of the Voxmap PointShell method of collision detection in virtual assembly methods planning", paper presented at ASME Design Engineering Technical Conferences, Pittsburgh, PA.

Li, J.R., Khoo, L.P. and Tor, S.B. (2003), "Desktop virtual reality for maintenance training: an object oriented prototype system (V-REALISM)", *Computers in Industry*, Vol. 52, pp. 109-25.

Marcelino, L., Murray, N. and Fernando, T. (2003), "A constraint manager to support virtual maintainability", *Computers & Graphics*, Vol. 27, pp. 19-26.

Ritchie, J.M., Dewar, R.G. and Simmons, J.E.L. (1999), "The generation and practical use of plans for manual assembly using immersive virtual reality", *Proc. Inst. Mech. Eng. Part B*, Vol. 213, pp. 461-70.

Seth, A., Su, H.J. and Vance, J.M. (2006), "SHARP: a system for haptic assembly and realistic prototyping", paper presented at ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Philadelphia, PA.

Vo, D.M., Vance, J.M. and Marasinghe, M.G. (2009), "Assessment of haptic-based interaction for assembly tasks in virtual reality", *Third Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, Salt Lake City, UT*, pp. 494-9.

Yao, Y.X., Xia, P.J., Liu, J. and Li, J. (2006), "A pragmatic system to support interactive assembly planning and training in immersive virtual environment (I-VAPTS)", *International Journal of Advanced Manufacturing Technology*, Vol. 30 No. 9, pp. 959-67.

Corresponding author

Pinjun Xia can be contacted at: smallping@fe.up.pt