World Academy of Science, Engineering and Technology
International Journal of Industrial and Manufacturing Engineering
Vol:7, No:2, 2013

Virtual Assembly in a Semi-Immersive Environment

Emad S. Abouel Nasr, Abdulaziz M. El-Tamimi, Mustufa H. Abidi, and Abdulrahman M. Al-Ahmari

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

***Abstract*—**Virtual Assembly (VA) is one of the key technologies in advanced manufacturing field. It is a promising application of virtual reality in design and manufacturing field. It has drawn much interest from industries and research institutes in the last two decades. This paper describes a process for integrating an interactive Virtual Reality-based assembly simulation of a digital mockup with the CAD/CAM infrastructure. The necessary hardware and software preconditions for the process are explained so that it can easily be adopted by non VR experts. The article outlines how assembly simulation can improve the CAD/CAM procedures and structures; how CAD model preparations have to be carried out and which virtual environment requirements have to be fulfilled. The issue of data transfer is also explained in the paper. The other challenges and requirements like anti-aliasing and collision detection have also been explained. Finally, a VA simulation has been carried out for a ball valve assembly and a car door assembly with the help of Vizard virtual reality toolkit in a semi-immersive environment and their performance analysis has been done on different workstations to evaluate the importance of graphical processing unit (GPU) in the field of VA.

***Keywords*—**Collision Detection, Graphical Processing Unit (GPU), Virtual Reality (VR), Virtual Assembly (VA).

I. Introduction

IN the last decade, Virtual Reality (VR) has been considered as a young technology but now it is mature enough to give some serious engineering applications. Although, its evolution can be traced back to the 1960s it was not until the mid-1980s that VR started to draw the attention of research centers and industry. However, since then and until very recently, most of the research efforts are paying an attention on the advances in hardware interfaces, devices, and simple applications, which in the early days were mostly created for demonstration purposes. VR is a technology which is a combination of powerful digital computers with special hardware and software to simulate an alternate world or environment. The objective of VR to simulate the reality which is convincing as real or true to the user and he can interact with it [1]. It is a technology with applications in a wide range of fields. These include education and training, product development, manufacturing, entertainment, architecture and landscaping, urban planning, rapid prototyping, space exploration, medicine, automotive and aerospace industries, defense, oil exploration, etc. The primary concept behind VR is that of illusion. It focuses on the demonstration of the fantasy world of the mind in computer graphics. VR technology has matured enough to warrant serious engineering applications. The integration of this new technology with software systems for engineering, design and manufacturing will provide a new boost to the field of computer-aided engineering [2].

VR has become extensively used over the last decades throughout the product design process. Virtual assembly is one of the applications using VR technology in the field of manufacturing. It permits a designer to perform mechanical assembly tasks in a virtual environment. Therefore, the user can evaluate the prospects of assembly before those mechanical parts are actually manufactured. The user can move around the virtual environment, use his hand, represented by a virtual hand in a virtual environment, to grip parts and assemble them. This brings the user a great feeling of immersion. This feeling will be enhanced if the interactive dynamic simulation is implemented in the virtual environment [3]. The objective of this paper is to develop a VR supported interactive and immersive assembly system. Moreover, a comparison is made for the graphical and processing capabilities of various workstations for two different case studies.

Including this section, the paper is organized into seven sections. Section 2 describes literature review of the previous research efforts in the area of virtual assembly. Section 3 explains the system description which includes hardware and software details. Section 4 illustrates the system architecture. Section 5 presents the case study of ball valve assembly and car door assembly in a semi-immersive virtual environment. Section 6 explains the performance analysis of the virtual assemblies. Finally, section 7 highlights the conclusions.

II. Literature Review

An enormous amount of research has been carried out in the area of virtual assembly. In this section, some of these works will be presented.

Jayaram et al., have developed a virtual reality based application to facilitate the planning, evaluation and verification of a mechanical assembly system [4]. Wang and Li proposed a dynamic data model for visualization of industrial assemblies. A solution was provided to users for practical real-time visualization of complex dynamic virtual assembly environments with affordable graphics hardware [5].

Emad S. Abouel Nasr is an associate professor at Department of Industrial Engineering, College of Engineering, King Saud University, Riyadh-11421, Saudi Arabia and Mechanical Engineering Department, Faculty of Engineering, Helwan University, Egypt (e-mail: eabdelghany@ksu.edu.sa).

Abdulaziz M. El-Tamimi is with Department of Industrial Engineering, and also in Advance Manufacturing Institute, King Saud University, Riyadh-11421, Saudi Arabia (e-mail: atamimi@ksu.edu.sa).

Mustufa H. Abidi is with the Advanced Manufacturing Institute, King Saud University, Riyadh-11421, Saudi Arabia (e-mail: mabidi@ksu.edu.sa).

Abdulrahman M. Al-Ahmari is the Dean of Advanced Manufacturing Institute and also he is in Department of Industrial Engineering, King Saud University, Riyadh-11421, Saudi Arabia. (e-mail: alahmari@ksu.edu.sa).

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

Jayaram et al., presented two industrial based case studies to determine if immersive virtual assembly capabilities allow industry assembly situations to be modeled and studied realistically, and to demonstrate the downstream value of the virtual assembly capabilities in areas such as ergonomics, assembly installation, process planning, installation, and serviceability [6].

Virtual assembly applications have been widely used for verifying and planning in the mechanical assembly. Banerjee and Banerjee described a Virtual Manufacturing Lattice (VML) structure to address the identified limitations of the scenegraph structure for application in virtual assembly sequence planning [7]. A new virtual environment system for assembly application has been presented by Liu et al., [8]. Song and Chung used a Digital Mockup (DMU) system to build a virtual model of designs to prevent the interferences and mismatch during precision design stages. A heterogeneous CAD assembly method to construct the DMU system was proposed through an XML and the lightweight CAD file [9].

Sun and Hujun developed a framework of a two-handed virtual assembly application. They used the system for the interactive task planning of assembly application [10]. Jimeno et al., had presented a comprehensive review about the applications of virtual reality in the field of design and manufacturing processes [11]. Lin et al., presented a case study of virtual assembly prototyping. A torque converter blade assembly procedure was presented [12].

Numerous amount of research have been done to develop virtual assembly applications with force feedback devices [13-15]. Wan et al., developed grasping patterns for virtual assembly tasks. They experimented on assembly models and proved that their algorithms worked well for the virtual assembly application [16]. Seth et al., developed a system known as SHARP (A system for haptic assembly and realistic prototyping). This system uses physical-based modeling and a dual handed force feedback device ability in order to assemble components in the immersive environment [17]. Choi et al., investigated the factors that influence the efficiency of assembly, namely, assembly method, assembly sequence, and assembly time [18].

Unlike, fully immersive VR systems which use specialized interaction devices such as head-mounted displays to create a sense of presence for the user within the 3D world or CAVE™ (Cave Automatic Virtual Environment), desktop systems utilize general-purpose hardware (mouse and keyboard) for interaction and even though they obviously offer a much reduced sense of presence in the VE, they were still useful for many application areas [19].

Gupta et al., have developed a desktop system called the virtual environment for design assembly (VEDA). The goal was to broaden the CAD systems to evaluate and compare alternative designs using Design for Assembly Analysis. A unified physically based model has been developed for modeling dynamic interactions among virtual objects and haptic interactions between the human designer and the virtual objects [20]. Li et al., developed a desktop VR system for maintenance training known as V-REALISM. It provides the users with a desktop virtual environment to practice the disassembly for maintenance of certain components [21].

A desktop-based system ‘‘Vshop’’ was developed by Pere et al., for mechanical assembly training in VE. The research focused on investigating desktop-based systems as a low-cost alternative and utilizing commercial libraries for easy creation of interactive VR software. Disassembly and recycling factors were also taken into account during the initial design stages allowing for an environmentally conscious design. It showed that VA training can provide a platform for offline training of assembly workers which is an important issue when assembly tasks are hazardous or complicated [22, 23].

Different realistic hand grasping patterns involving complex CAD models have been explored by Wan et al. using a multimodal system called MIVAS (A Multi-Modal Immersive Virtual Assembly System) [24]. Xia et al., had developed a haptic-based system for assembly training of complex products in a virtual environment [25]. Yao et al., developed a pragmatic system I-VAPTS, for interactive assembly planning and training in a virtual environment [26].

Collision detection issue is also explored by many researchers. A three layer model (skeletal layer, muscle layer, and skin layer) was adapted to simulate deformation in the virtual hand using simple kinematics models. Hand to part collision detection and force computations were performed using fast but less accurate VPS software, while part to part collision detection was implemented using the RAPID algorithm [27, 28].

Su et al., proposed an efficient and precise collision detection algorithm for constructive solid geometry (CSG) represented objects in a virtual environment. Mainly it took advantages of the CSG ‘divide-and-conquer’ paradigm and efficient distance-aided collision detection for convex bounding volumes [29]. Geometric constraints were utilized in combination with collision detection to calculate allowable part motion and accurate part placement. Users could feel the size and shape of digital CAD models via the CyberGrasp® haptic device from CyberGlove Systems™ [30].

A CAD-linked virtual assembly environment was developed by Wang et al., in order to use a constraint-based modeling for assembly. This PC-based system ran as a separate process and maintained communication with Autodesk Inventor® CAD software. Low level-of-detail (LOD) proxy illustrations of CAD models were used for visualization in the virtual environment. The assembly system required constant communication with the CAD system using proprietary APIs (Application Programming Interface) for acquiring information such as assembly structure, constraints, B-rep geometry, and object properties. The idea of proxy entity was proposed which allowed the system to map related CAD entities (surfaces, edges, etc.) to their corresponding triangle mesh representations present in VR [31]. Jezernik et al., provided a solution to integrate CAD and VR databases in design and manufacturing processes. The developed system for VRML model visualization enabled changes in the configuration file, which is written in XML, and that automatically reviewed the model including the functional

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

behavior. From the evaluated VRML model, the connection to a PDM system was provided with a list of elements and their material properties [32].

Based on literature, it can be concluded that a lot of work has been done in the field of virtual assembly in the last two decades. However, there are some issues left such as the integration of CAD model with the VR software; the graphics in the VR environment should be like the real one and the integration of tracking and haptics devices to give the sensory feedback. These all issues mainly require a solution in a form of VR software that should be capable of handling all these challenges. One of such VR software or toolkit available today which is quite capable of handling these issues is the Vizard virtual reality toolkit (VIZTK), which is a product of Worldviz®. In this paper, a virtual assembly environment has been created with the help of VIZTK for the assembly of a ball valve and car door as illustrative case studies.

III. System Description

A. Hardware

This research is carried out in a semi-immersive virtual environment generated by the 3D immersive visualization studio which is provided by Virtalis ltd. This studio consists of a Christie Mirage S+6K projector, a fabricated screen which is specially designed for the rear projection, a Dell T5500 workstation with a powerful graphics card from Nvidia, Intersense™ wired tracking system consisting of the head and hand tracker, Shutter glasses, 3D surround sound system, and the other supporting devices.

B. Software Platform

To develop the virtual assembly environment, AutoCAD® is selected as a CAD modeling tool for ball valve and CATIA® is used for car door, and Vizard 3.0 toolkit is selected as a platform for constructing the VA system. Since, there is an issue of a common file format between these software systems; therefore, AutoDesk® 3DS Max® is used as a bridge between these two environments. As AutoCAD®, CATIA®, and 3DS Max® are the common packages; therefore a brief insight about VIZTK will be presented in the next sub section.

1. Vizard Virtual Reality Toolkit

Vizard is a high level graphics toolkit for the development of high-performance graphics applications, including virtual reality, scientific visualization, games, and flight simulation. By providing an object-oriented framework encompassing OpenGL, DirectX multimedia, human bipeds, display and peripheral hardware interfaces, and efficient networking, Vizard gives the liberty from low-level programming and instead allows concentrating on creating interaction and content. VIZTK has been developed jointly at Massachusetts Institute of Technology and the University of California in the early 1990's [33].

In the earlier toolkits there, is a lack of user friendliness to interact with the CAD data and virtual reality hardware. The lack of interactive graphical user interface insists the user to go for complex programming practices that have to be developed completely using graphical libraries such as World Toolkit, VR Juggler, IRIS Performer, Java3D, MVL Toolkit, OpenGL, Open Inventor for C++ and Java Developers, Visualization Toolkit and VISAGE, etc. On the other hand, there are a lot of Integrated Development Environments (IDE's) which have been developed to shorten the usage of programming such as PTC Division Mockup which is a commercial software best suits for engineering designs, Dassault Systemes 3DVIA Virtools®, 3D Game Studio®, VR4Max®, etc., for non programming users. The most important feature distinguishing VIZTK from all other products is its rich Python-based development environment. Moreover, VIZTK also avoids the excessive of programming, namely purely visually-based programming, in which the development effort is overburdened by a cumbersome graphical interface that obscures a good and logical programming. Instead, VIZTK offers a clean graphical user interface and visual tools that smartly augment those tasks best handled through scripting. Vizard's graphics engine is all built in C/C++ and based on OpenGL. Therefore, the end-to-end efficiency of its application is nearly identical to one written in pure C/C++. In terms of hardware compatibility, it communicates natively to nearly all common VR peripherals, and when it does not then there is a VRPN plug-in which will take care of the need. The graphical user interface helps to quickly create the script files, and add behaviors and events to the objects instantly without recompiling the code which saves the performance time.

Python is a high level interpreted object oriented scripting language which supports cross platform integrity. VIZTK uses Python scripting language for programming logic and has interactive Graphical User Interface (GUI) for non programmers. VIZTK follows the Open Scene Graph Structure, hence has a parent-child hierarchical representation of nodes. The nodes are called Objects. The reason why to use python as core programming language is presented in comparison with other programming languages [34].

VIZTK has many features which make it a user friendly package for the user such as:

- *Live Characters:* VIZTK has inbuilt support for virtual characters called 'Avatars' which can change their shape and position such as walking, talking, running etc. in virtual worlds. Avatars can be created in Character Animation Library (CAL3D) format i.e., cfg using a third party character design software such as Autodesk® Character Studio which can be imported in VIZTK. CAL3D is Autodesk® 3D MAX plug-in which helps to export the character in virtual world. This cfg file will have all the information of the animated character such as the character's state, character body parts, etc. Second Life and Active Worlds are the best examples of the virtual worlds with characters.
- *Stereoscopic Support:* VIZTK supports stereoscopic viewing through 3D shutter Glasses and Head Mounted Displays. It has inbuilt support to enable the stereoscopic vision through a single line of code, i.e., Viz.MainView.stereo( ) and Viz.ANAGLYPHIC.

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

- *Hardware Support:* VIZTK supports a variety of native hardware devices such as Head Mounted Displays, Stereoscopic Shutter Glasses, 3DOF/6DOF Trackers, Precision Position Trackers, Motion trackers, and Haptics devices which can be operated independently by the OpenGL device drivers [35].
- *Virtual Reality Peripheral Network (VRPN):* The concept of VRPN was first proposed by Taylor et al., [36]. This is a more advance step toward the technology. Applying VR over the internet with interconnected cables and following network protocols between the client and server to perform the mechanical operation using VR hardware. Some examples of such networks are EM, DIVE, and SIMNET [37-39]. Collaboration in virtual networks is a major aspect of networked VR platforms. Now, a lot of work has been developed in the field of collaborative VR. The ability to develop a collaborative VR environment is supported by VIZTK.

IV. System Architecture

The work flow architecture of VIZTK is shown in Fig. 1. The user friendly graphical user interface provides the functionality to perform all the input and output operations such as creating, importing, and managing the python script, C++ files, images, textures and sound files. The most powerful advantage of python script is that it can be embedded in C/C++ based applications. Moreover, customized modules in C/C++ can be created to extend the functionalities of python.

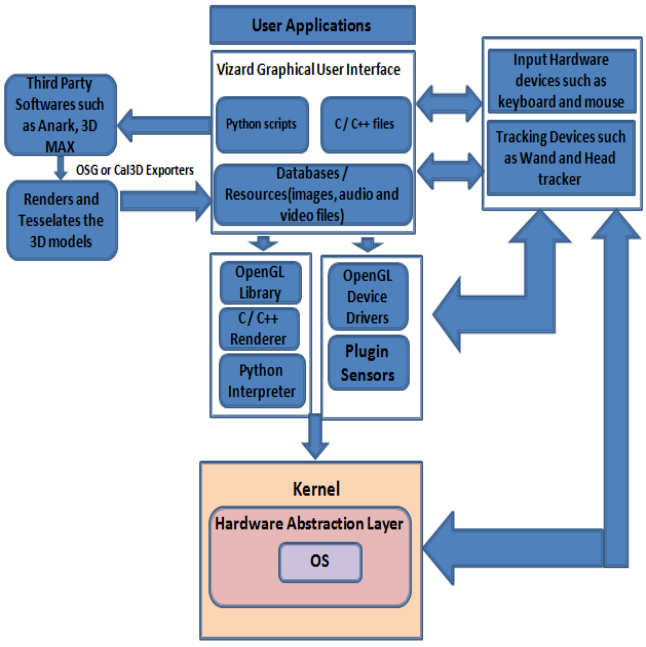


Fig. 1 System Architecture of VIZTK

The functionalities of the main segment of the architecture are discussed briefly below:

A. Open Graphics Library (OpenGL)

OpenGL is an API and universally recognized as a standard for graphical applications developed by Silicon Graphics Inc. in 1992. It has more than 250 functions to render the graphics on display devices [40]. It helps to run cross platform applications and to create the user defined modules.

B. Python Interpreter

Python is a high level interpreted scripting language. The interpreter is responsible for the execution of all the commands that are being passed by the user to perform some operation. Different functionalities such as, executing a script file, invoking a function, invoking a timer event, performing a task, playing an audio or video file, etc. can be used to develop one's own modules.

C. Physics Engine

The physics engine is responsible for several additional tasks such as collision detection, forces, and gravity on 3D objects. Enabling of physics property of objects is done by the following command: viz.phys.enable() and gravity can be enabled using viz.phys.setGravity(x,y,z), where (x,y,z) is a coordinate axis. Moreover, forces can be applied using object.applyForce(dir=[x,y,z], duration=s seconds). Enabling collision detection will be discussed in the later sections.

V. Virtual Assembly

A. Case Study I: Ball Valve

Ball valves are used extensively in industry because they are very versatile and can handle a high pressure and temperature effectively. They are easy to repair and operate manually or by actuators. Ball valves are durable and usually work to achieve perfect shutoff even after years of disuse. Furthermore, it does not offer the fine control that may be necessary in throttling applications but are sometimes used for this purpose. The product used in this paper is a split body, two ways, and reduced port type of ball valve as shown in Fig. 2.The components and their material are listed in Table I.



Fig. 2 Ball Valve used for Virtual Assembly in case study1

1. Data Flow for Ball Valve Assembly

The designing of the valve's CAD model which is required for the assembly process was done in Autodesk® AutoCAD.

TABLE I
COMPONENTS OF THE BALL VALVE

Item No.	Item	Material	Quantity
1	Right side body	Forged brass MS 58	1
2	Left side body	Forged brass MS 58	1
3	Stem	Brass MS 58	1
4	Teflon seal	Teflon	2
5	Ball	Stainless Steel	1
6	Handle	Forged Steel	1
7	Slot washer	Forged Steel	1
8	O ring	Rubber	2
9	Nut	Carbon Steel	1

The CAD model of the ball valve is shown in Fig. 3.



Fig. 3 Different Views of a Ball Valve modeled in AutoCAD

There isn't a standard data exchange format between the CAD system and VA system. Therefore, the information of CAD models can't be transmitted to VE directly. Although, the CAD system can export the CAD graphical models in other formats (e.g., WRL, DXF, 3DS, SLP, etc.) imported into VE and displayed there. However, these formats can only keep partial geometry information. For example, the whole assembly body or a single component can be selected, but its surfaces cannot be selected. So, the models in these formats are only of the concept of body and not for surfaces in VE. However, surface data is such an important concept in assembly and it has to serve such functions as follows: (1) orienting components or assembly bodies, (2) defining constraints, (3) defining joints, (4) defining geometrical and dimensional tolerances, (5) defining precision or roughness, and (6) defining physical information (e.g., colors and materials, etc.). Without the surface data, the assembly concept becomes vague and the VA system is also reduced to a simple simulation application. Sheng et al., have proposed a data decomposition and information translation method to achieve the decomposition and translation [41].

In addition, the VA system needs not only geometry information, but also topology information, assembly information, etc. Therefore, transferring the information of CAD models to VA is not only the first step of constructing a VA system, but also a key step having an influence on the display effect, the assembly effect, and the precision requirements of the VA system. Moreover, for dealing with the above issue, the ball valve model from AutoCAD has been transferred to VIZTK with the help of Autodesk® 3DS Max. The exploded view of the ball valve is shown in Fig. 4.



Fig. 4 Exploded view of Ball valve in 3DS Max

B. Issues in Virtual Assembly of Ball Valve

While importing the 3D models in to VIZTK, the objects will be deformed due to the difference between the sampling rates of the drawings and the display device, which results in jiggering of images. This problem can be overcome by applying anti aliasing, which will be discussed later in this section. The scale factor of objects in the CAD package and VR platform must be taken care of; otherwise the 3D models will be too large or too small when imported into VR software

1. Anti Aliasing

Choosing a suitable *sampling rate* depends on many factors such as data size restraints, need for accuracy, and in some cases the cost per sample. Much research has been dedicated to this topic and one of the most important theories to come from these efforts aids in determining the lowest acceptable sampling rate. The Nyquist Theorem states that the sampling rate must be twice the frequency of the signal or an effect known as *aliasing* occurs. Generally, aliasing refers to the degradation of a sound, image, or the other signals during the sampling process due to low-resolution sampling. The errors caused by aliasing are called artefacts. Common aliasing artefacts include jagged profiles, disappearing, improperly rendered fine detail, and disintegrating textures. Aliasing is among common problems in computer graphics because screen or file resolutions are finite whereas the mathematical models describing an image have an infinite resolution [42].

Anti-Aliasing is a method of fooling the eye that a jagged edge is really smooth. Anti-Aliasing is often referred in games and on graphics cards. Therefore, many techniques have been proposed to overcome the problem in point based models [43, 44]. A Two Point Anti-aliasing scheme is a good candidate technique for line segments, curves, and triangles as constructive primitives [45]. The aim of these techniques is to make the edges of 3D model smooth. To overcome the problem of aliasing a function for anti-aliasing, viz.setMultipleSample(x), is added in python programming. This command will sample the full screen for those many times(x) after each frame is rendered. The general sample rate is 2, 4, or 8. Higher the value of the sampling rate, the more is the smoothness of the curves or edges. The wireframe model of the ball valve in VIZTK is shown in Fig. 5 after anti-aliasing process is done.

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

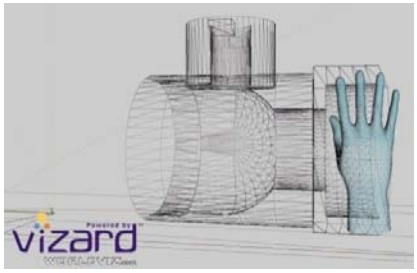


Fig. 5 Wireframe model of valve in VIZTK after anti-aliasing

2. Collision Detection

Collision detection (CD) is one of the major and important issues for the virtual assembly. With the help of collision detection, there would be no interference between the virtual parts of the model. This is the mechanism where one object or user's view is getting into contact or intersecting with the boundary of another object in order to occupy the latter's physical space in virtual environment with or without deformation of shape. There should be a proper system to notify such changes in the forces and gravity of the objects. This is called 'Collision Detection'. The methods of continuous CD have been proposed by Redon et al., [46]. The CD for flying objects with a mesh has been discussed by Held et al., and also a Bounding Box technique is addressed by Zachmann [47, 48]. Tight-Fitting oriented Bounding Box Trees and a methodology for CD for virtual assembly has been proposed by Gottschalk et al., and Su et al., [28, 49].

VIZTK has a physics engine which provides the functionality to detect collisions and forces of gravity. For this, the collision area should be defined. A few lines of python script do the job:

- viz.collision(viz.ON) checks for the collision between the viewer and the world.
- object.collideBox() creates a virtual box surrounding the object and detects the collision.
- object.collideMesh() will create the boundary up to the objects geometry.
- object.collideplane() creates a virtual plane to detect the collision.

C. Scene Graph Structure of Virtual Assembly of Valve

Scene graphs are ubiquitous in VA. Most high-level graphics toolkits provide a scene graph application programming interface (API) to model 3D scenes and to program 3D applications. An ideal scene graph API should simplify programming of 3D applications, optimize rendering performance, and provide the abstraction by encapsulating rendering algorithms. The scene graph APIs differ in the way of the underlying 3D rendering system can be accessed. OpenInventor provides full access to OpenGL whereas VRML and Java3D do not provide the extensibility to that degree.

VIZTK follows the Open Scene Graph structure. Scene graphs are the fundamental data structures for the hierarchical scene modeling. Scene is the root node of the scene graph and there is a content branch which holds the group nodes, transformations along with lights, fog, textures, and audio files. The successive nodes of content branch are its child nodes and at the end its leaf nodes. Final level indicates the core level geometrical primitive structure of the leaf nodes. On the other hand, there is a viewing branch for different camera view projections and their transformations. The scene graph structure of the valve assembly in VIZTK is shown in Fig. 6.

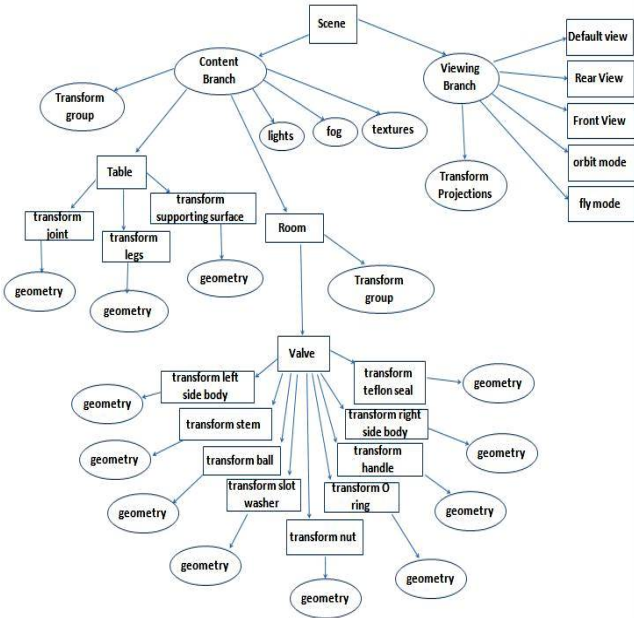


Fig. 6 Scene Graph structure of the Ball valve virtual assembly environment in VIZTK

D. General Algorithm and Flowchart for the VA

The various steps involved in the virtual assembly of the ball valve such as CAD model preparation, data transformation, data import, 3D visualization, etc. are represented in a pictorial form with the help of a flow chart. The flow chart for the assembly process is shown in Fig. 7.

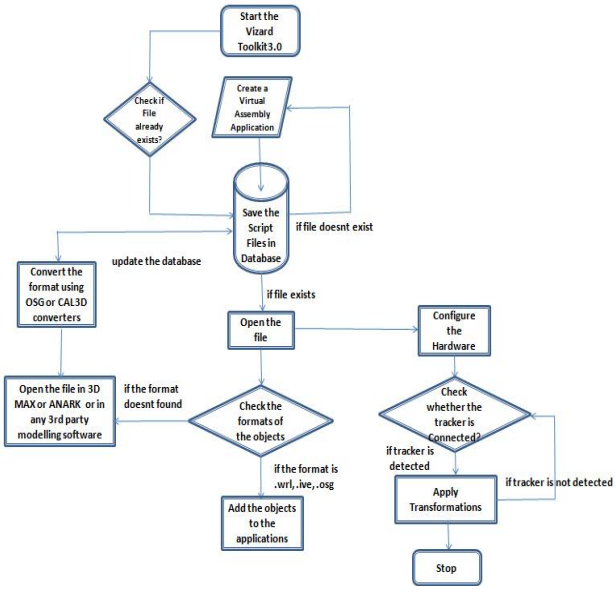


Fig. 7 Flow chart of the Virtual Assembly

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

The program steps in general statements are converted as a generalized workflow algorithm. The data flow from one segment of the proposed system to the other is addressed as follows:

1. Import Vizard 3.0 packages for enabling various hardware and software functions.
2. Launch the application in STEREO or ANAGLYPHIC or FULL SCREEN MODE.
3. Add WAND and HEAD Tracker to the application.
4. Add the objects such as table and valve parts to the application.
5. Turn off the default mouse navigation system.
6. Turn on the collision detection system.
7. Write a function to grab the object:
   {Create a link between the WAND and the OBJECT
   If link already exists, delete the link and reset the link.
   Calculate the distance between the wand tracker pointer and the object; if it is less than the specific amount of distance grab the object}
8. Applying transformations:
   Write a function to keep track of WAND position
   {Note the x and y coordinates positions
   If the object is picked then update WAND key status
   If the object is picked and WAND key is pressed
   Object.rotate(x, y, z, angle)
   Update the current positions of x and y coordinates and call back the function when a WAND button event occurs.}

E. VA in Semi-Immersive Environment

After importing the complete model into the VIZTK and handling all the issues such as collision detection, anti-aliasing, etc., then the virtual assembly is done in a semi-immersive environment generated by the VR lab of Advance Manufacturing Technology Lab of King Saud University-Riyadh. The virtual assembly process in the semi-immersive environment with the help of hand and head tracker is shown in Fig. 8.



Fig. 8 In a semi-immersive virtual environment, assembling the valve using head, wand and ultra sonic trackers

F. Case Study II: Car Door Assembly

A car door is a separation, hinged in front of an opening which is used for entering and exiting a car. It is a combination of various components, which are responsible for different operations, such as locking system, window glass mechanism, etc. The door used in the paper is shown in Fig. 9 and the various components of car door are listed in Table II.

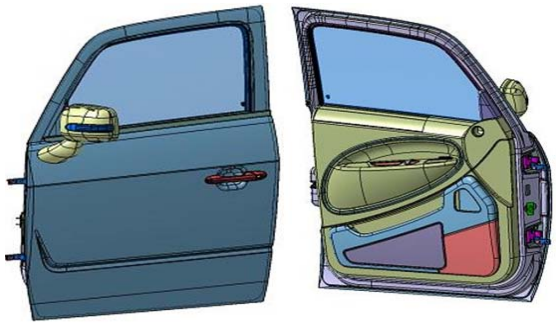
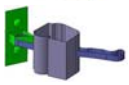


Fig. 9 CAD model of Car Door used for VA in case study II

TABLE II
COMPONENTS OF CAR DOOR

S. No	Part Name	Picture	S.No	Part Name	Picture
1	Complete Door		1.9	Water Seal	
1.1	Door Sheet Metal Panels		1.10	Glass Seal	
1.2	Door Retainer		1.11	External Mirror	
1.3	Dust Seal Upper		1.12	Trim Cover	
1.4	Door Sash		1.13	Handle Outer	
1.5	Glass Regulator		1.14	Grommet	
1.6	Door Latch		1.15	Water Seal Inner	
1.7	Armrest Bracket		1.16	Speaker	
1.8	Door Glass		1.17	Inner Parts	

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

S.No	Part Name	Picture	S.No	Part Name	Picture
1.1	Door Sheet Metal Panel		*1.1.1.4*	Waist Middle	
1.1.1	Inner Panel Assmb.		*1.1.1.5*	Hinge Reinfor-cement	
1.1.2	Outer Panel Assmb.		*1.1.1.6*	Hinge Nut Plate Upper	
1.1.3	Hinge Upper		*1.1.1.7*	Hinge Nut Plate Lower	
1.1.4	Hinge Lower		*1.1.1.8*	Latch Reinfor-cement	
1.1.1	Inner Panel Assmb.		*1.1.1.9*	Handle Plate	
1.1.1.1	Inner Panel		*1.1.1.10*	Intrusion Beam	
1.1.1.2	Glass Surrou-nd		*1.1.1.11*	Regulat-or Plate	
1.1.1.3	Waist Inner		*1.1.1.12*	Glass Bracket	

1. Data Flow in Car Door Assembly

The designing of the car door's CAD model which is required for the assembly process was done in CATIA®. The CAD model of the ball valve is shown in Fig. 9.

The CAD model from CATIA® is exported into *.wrl, which is a format compatible with Vizard Toolkit. Rest of the steps and algorithm for virtual assembly is the same as in case of study 1. The issues related with the virtual assembly such as anti-aliasing, collision detection, are same as discussed in case of ball valve.

After importing the complete model into the VIZTK and handling all the issues such as collision detection, anti-aliasing, etc., then the virtual assembly is done in a semi-immersive environment generated in the VR lab of Advance Manufacturing Technology Lab of King Saud University. The virtual assembly process in the semi-immersive environment with the help of hand and head tracker is shown in Fig. 10.



Fig. 10 Virtual Assembly of a Car Door in a Semi-Immersive Environment

VI. Performance Analysis of the Virtual Assembly

After completing the virtual assembly in a semi-immersive environment, the performance analysis of the VIZTK VA has been carried out on three different workstations. The obtained results for ball valve assembly are listed in Table 3. The criteria of the comparison are based on number of primitives generated, number of frames generated, time taken to load the model, and cull. The results obtained for the car door assembly are listed in Table 4. The criteria of comparison for case study 2 are the same as in case study 1.

TABLE III
Performance Analysis of Virtual Assembly Operation on Three diffeRent Computers for Case Study 1

Computer model	**Dell Optiplex 960**	**HP Elite pc 9561me**	**Dell Precision T5400**
Graphics Card	ATI Radeon HD 3450	NVIDIA Ge Force 9800GT	NVIDIA Quadro FX 5600
Processor	Intel Core2 Quad Q9400, 2.66GHz	Intel Core2 Quad Q9400, 2.66GHz	Intel Xeon E5410, 2.33 GHz
RAM in Giga Bytes	4	4	4
Number of Primitives Generated	23,374	23,442	44,488
Number of Frames Generated	60	60	75
Time Taken to Load in Seconds	0.6	0.55	0.53
Cull	0.16	0.21	0.51

TABLE IV
Performance Analysis of Virtual Assembly Operation on Three Different Computers for Case Study 2

Computer model	**Dell Optiplex 960**	**HP Elite pc 9561me**	**Dell Precision T5400**
Graphics Card	ATI Radeon HD 3450	NVIDIA Ge Force 9800GT	NVIDIA Quadro FX 5600
Processor	Intel Core2 Quad Q9400, 2.66GHz	Intel Core2 Quad Q9400, 2.66GHz	Intel Xeon E5410, 2.33 GHz
RAM in Giga Bytes	4	4	4
Number of Primitives Generated	128,234	317,762	524,956
Number of Frames Generated	25	40	48
Time Taken to Load in Seconds	534	150	123
Cull	45.7	69.5	75.6

The numbers of primitives generated in Dell precision T5400 with NVIDIA Quadro graphics card are more and give a good level of detail than those produced in other computers as shown in Fig. 11 and Fig 12 for case study 1 and 2 respectively. Culling is the technique through which the polygons which don't face the viewer cannot be seen. The culling of the primitives which is not in the viewing range reduces the rendering time and increases the performance.

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

Therefore, from Table III and IV, it can be concluded that Dell Precision T5400 provides more culling resulting in less time for rendering, in spite of the fact that, it has generated more number of primitives. One of the interesting facts that can be concluded from this performance analysis is that the rendering time is less for Dell precision T5400 even it has lesser processing speed than HP elite, but it has more graphical processing speed. Therefore, it can be concluded that graphical processing unit (GPU) speed is more important than the central processing unit (CPU) speed for VR based applications. NVIDIA are the most suitable cards for VIZTK for handling large 3D models with several thousands of primitives.

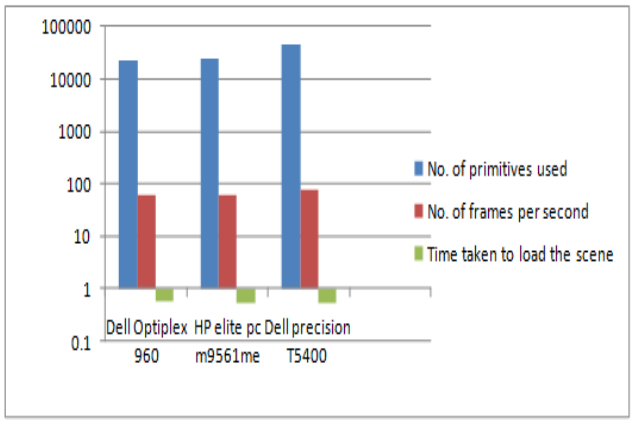


Fig. 11 Graphical comparison of performance features on different computers for case study 1

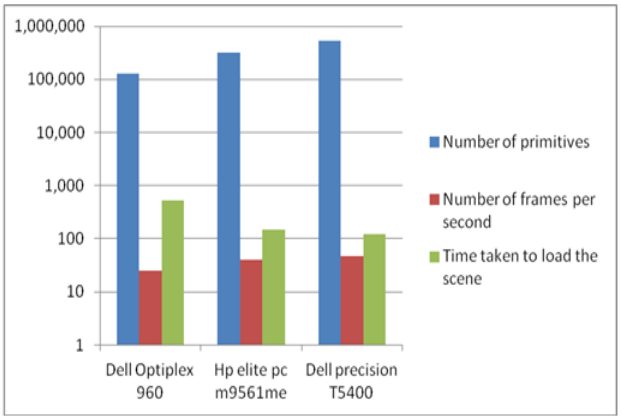


Fig. 12 Graphical comparison of performance features on different computers for case study 2

VII. Conclusions and Discussion

Virtual Assembly technology is developing rapidly. It covers a variety of aspects, and is a key content of virtual manufacture. It shortens product development cycle, and improves product quality. The path from CAD model development to the product assembly shows how VR based assembly simulation can be integrated as an efficient tool into the well-known CAD/CAM environment. This paper presented the concepts behind a VR-based virtual assembly system using VIZTK. VIZTK has many advantages such as, it is compatible with a lot of hardware devices, provides a graphical user interface with python programming API, etc. Still, there are some disadvantages associated with it like VRPN requires an efficient graphics card to render the graphic and high network bandwidth. Anti-aliasing and collision detection techniques results in better VA environment. In Anti-aliasing, higher parametric value for sampling increases the rendering time and decreases the performance.

From the performance analysis, it has been concluded that the consistent results are obtained in drawing the primitives, culling, and producing more frame rates but they depend upon the capability of the graphics card as well. Based on the performance analysis, it can be seen that for the virtual assembly application, GPU speed is more important than CPU speed. With higher GPU speed, the rendering or loading time is less. In case study 1, the difference in various output factors (time to load, cull, etc.) is not very high, but in case study 2 which is more complex relatively, the difference is significant. Still, in both case studies, GPU speed is coming out to be more significant than the CPU speed for VA applications.

Based on semi-immersive VA experience, it can be said that the VA offers an environment which looks realistic to the user. Still, there are some challenges left in this field such as graphics, no matter how impressive, are not lifelike. Audio hardware still could not produce natural and realistic effects. There are a number of important issues related to the software such as graphics programming for VR, 3D – interaction, time for computation, and rendering images, common platform for different systems, etc. need to be explored. Each and every one of these shortcomings is the subject of intense research and work. The object handling can be improved by integrating force feedback systems based on a real-time mechanism for calculating reaction forces, giving the VR operator a more intuitive interface to the assembly simulation.

Acknowledgment

The authors are highly thankful to Advance Manufacturing Institute, King Saud University for providing the facilities to accomplish this research. The authors would also like to thank Mr. Nasruddin Shaik Pasha for his support.

References

[1] Liu X., Yang L. & Ren J. 'Study on Chip Forming and Breaking Process Based on Virtual Reality'. Proc. ICALIP 2008, pp. 1738-1742.
[2] Jayaram S., Connacher H. I. & Lyons K. W. 'Virtual Assembly Using Virtual Reality Techniques'. Computer-Aided Design, 29 (8) (1997), pp. 575-584.
[3] Gaoliang P., Gondong W., Wenjian L. & Haiquan Y. 'A desktop virtual reality-based interactive modular fixture configuration design system'. Computer-Aided Design, 42 (2010), pp. 432-444.
[4] Jayaram S., Wang Y, Jayaram U., Lyons K.W. & Hart P. 'VADE: A virtual assembly design environment'. IEEE Computer Graphics and Applications, 19(6) (1999), pp. 44-50.
[5] Wang Q.H. & Li J.R. 'Interactive visualization of complex dynamic virtual environments for industrial assemblies. Computers in Industry, 57 (2006) pp. 366-377.
[6] Jayaram S., Jayaram U., Kim Y.J., DeChenne C., Lyons K.W., Palmer C. & Mitsui T. 'Industry case studies in the use of immersive virtual assembly'. Virtual Reality, 11 (2007), pp. 217-228.

Open Science Index, Industrial and Manufacturing Engineering Vol:7, No:2, 2013 publications.waset.org/8403/pdf

[7] Banerjee A. and Banerjee P. 'A behavioral scene graph for rule enforcement in interactive virtual assembly sequence planning'. Computers in Industry, 42 (2000), pp. 147-157.
[8] Liu G.H., Yao Y.X. & Zhang H.Z. 'Development of a New Virtual Environment System for Assembly'. Key Engineering Materials, 315-316 (2006), pp. 556-560.
[9] Song I.H. & Chung S.C. 'Synthesis of the digital mock-up system for heterogeneous CAD assembly'. Computers in Industry, 60 (2009), pp. 285-295.
[10] Sun H. & Hujun B. 'Two-handed assembly with immersive task planning in virtual reality'. Virtual Reality, 6(1) (2002), pp. 11-20.
[11] Jimeno A. & Puerta A. 'State of the art of the virtual reality applied to design and manufacturing processes'. Int J Adv Manuf Technol, 33 (2007), pp. 866-874.
[12] Lin Y.J. & Farahati R. 'CAD-Based Virtual Assembly Prototyping – A Case Study'. Int J Adv Manuf Technol, 21 (2003), pp. 263-274.
[13] Seth A., Su H. & Vance J. 'A desktop networked haptic VR interface for mechanical assembly'. Proc. of IMECE '05, (2005), ASME, Orlando, FL.
[14] Jayaram S., Joshi H., Jayaram U., Kim Y., Kate H. & Varoz L. 'Embedding haptics-enabled virtual tool in CAD for training applications'. Proc. of IDETC/CIE (2006), Philadelphia, Pennsylvania.
[15] Adam S.C., Scott D.M. & Bert B. 'A haptic assembly and disassembly simulation environment and associated computational load optimization techniques'. J Comput Inf Sci Eng, 1(2) (2001), pp. 113-122.
[16] Wan H., Shuming G. & Qunsheng P. 'Virtual grasping for virtual assembly tasks'. Proc. of third international conference on Image & Graphics, ICIG '04, (2004).
[17] Seth A., Su H. & Vance J. 'SHARP: A system for haptic assembly and realistic prototyping'. Proc. of DETC '06, ASME, IDETC/CIE (2006), Philadelphia, PA.
[18] Choi A.C.K., Chan D.S.K. & Yuen M.F. 'Applications of virtual assembly tools for improving product design'. Int J Adv Manuf Tecnol, 19(5) (2002), pp. 377-383.
[19] Sayers H. 'Desktop virtual environments: A study of navigation and age'. Interacting with Computers, 16(5) (2004), pp. 939-956.
[20] Gupta R., Whiney D. & Zeltzer D. 'Prototyping and design for assembly analysis using multimodal virtual environments'. Computer-Aided Design, 29 (1997), pp. 585-597.
[21] Li J.R., Khoo L.P. & Tor S.B. 'Desktop virtual reality for maintenance training: An object oriented prototype system (V-REALISM)'. Computers in Industry, 52(2) (2003), pp. 109-125.
[22] Pere E., Langrana N., Gomez D. & Burdea G. 'Virtual mechanical assembly on a PC-based system'. Proc. of ASME design engineering technical conferences and computers and information in engineering conference, DETC '96 (1996), Irvine, CA.
[23] Brough J.E., Schwartz M., Gupta S.K., Anand D.K., Kavetsky R. & Petterson R. 'Towards the development of the virtual environment-based training system for mechanical assembly operations'. Virtual Reality, 11(4) (2007), pp. 189-206.
[24] Wan H., Gao S., Peng Q., Dai G. & Zhang H. 'MIVAS: A multi-modal immersive virtual assembly system'. Proc. of ASME design engineering technical conferences and computers and information in engineering conference, DETC '04, (2004), Salt Lake City, UT.
[25] Xia P.J., Lopes A.M., Restivo M.T. & Yao Y.X. 'A new type haptics-based virtual environment system for assembly training of complex products'. Int J Adv Manuf Technol, (2011), DOI: 10.1007/s00170-011-3381-8.
[26] Yao Y.X., Xia P.J., Liu J.S. & Li J.G. 'A pragmatic system to support interactive assembly planning and training in an immersive virtual environment (I-VAPTS)'. Int J Adv Manuf Technol , 30(9-10) (2006), pp. 959-967.
[27] McNeely W.A., Puterbaugh K.D. & Troy J.J. 'Six degree-of-freedom haptic rendering using voxel sampling'. Proc. of SIGGRAPH '99, annual conference series, (1999), Los Angeles, CA.
[28] Gottschalk S., Lin M.C. & Manocha D. 'OBB Tree: A hierarchal structure for rapid interference detection'. Proc. of ACM SIGGRAPH'96, (1996), pp. 171-180.
[29] Su C.J., Lin F. & Ye L. (1999). 'A new collision detection method for CSG-represented objects in virtual manufacturing'. Computers in Industry, 40 (1999), pp. 1-13.
[30] CyberGlove. (2009). http://www.cyberglovesystems.com/products/cyberglasp/overview.
[31] Wang Q.H., Li J.R. & Gong H.Q. 'A CAD-linked virtual assembly environment'. Int J Prod Res, 44(3) (2006), pp. 467-486.
[32] Jezernik A. & Hren G. 'A solution to integrate computer-aided design (CAD) and virtual reality (VR) databases in design and manufacturing processes'. Int J Adv Manuf Technol, 22 (2003), pp. 768-774.
[33] Worldviz Vizard. (2010). http://www.worldviz.com/products/vizard/features.html
[34] Python. (2010). http://www.python.org/doc/essays/comparisions.html
[35] Vizard Toolkit Hardware. (2010). http://www.worldviz.com/products/peripherals/index.html
[36] Taylor II R.M., Hudson T.C., Seeger A., Weber H., Juliano J. & Helser A.T. 'VRPN: A device-independent, network-transparent VR peripheral system'. Proceedings of the ACM symposium on Virtual reality software and technology (2001), NY, USA.
[37] Wang Q., Green M. & Shaw C. 'EM-an environment manager for building networked virtual environments'. Proceedings of the Virtual Reality Annual International Symposium '95, Washington DC, USA.
[38] DIVE. (2009). http://www.sics.se/dive/manual/dive_file_format/dive_file_format.html
[39] Calvin J., Dickens A., Metzger P., Miller D. & Owen D. 'The SIMNET virtual world architecture'. Proc. of IEEE Virtual Reality Annual International Symposium '93, (1993), Seattle, WA, USA.
[40] Pulkkinen P. 'OpenGL and RenderMan Rendering Architectures'. Seminar on Computer Graphics, HUT, Telecommunications Software and Multimedia Laboratory (2002).
[41] Jiang-sheng L., Ying-xue Y., Pahlovy S.A. & Jian-guang L. 'A novel data decomposition and information translation method from CAD system to virtual assembly application'. Int J Adv Manuf Technol, 28 (2006), pp. 395-402.
[42] Hearn D., Baker M.P. & Carithers W. 'Computer Graphics with OpenGL, 4th ed'. ISBN-10: 0136053580, (2010), Prentice Hall.
[43] Ren L., Pfister H. & Zwicker M. 'Object Space EWA Surface Splatting: A Hardware Accelerated Approach to High Quality Point Rendering'. Computer Graphics Forum, 21 (2002), pp. 461-470.
[44] Grossman J.P. & Dally W.J. 'Point Sample Rendering'. Proc. of Rendering Techniques '98, (1998).
[45] Wu X. 'An efficient antialiasing technique'. ACM SIGGRAPH Computer Graphics, 25(4) (1991), pp. 143-152.
[46] Redon S., Kheddar A. & Conquillart S. 'Fast Continuous Collision Detection between Rigid Bodies'. Proc. of Eurographics '02, vol. 21 (3) (2002).
[47] Held M., Klosowski J.T. & Mitchell J.S.B. 'Evaluation of Collision Detection Methods for Virtual Reality Fly-Throughs'. Proc. of Canadian Conference on Computational Geometry, (1995), pp.205-210.
[48] Zachmann G. 'Real-Time And Exact Collision Detection For Interactive Virtual Prototyping'. Proc. of the 1997 ASME Design Engineering Technical Conferences, (1997), pp. 14-17.
[49] Su C.J., Fu-Hua L. & Zhang X. 'An efficient collision detection methodology for virtual assembly'. Proc. of IEEE conference on Systems, Man and Cybernetics, (1998), pp. 360-365.