Interactive Assembly Modelling within a CAVE Environment

3.5

Terrence Fernando Luis Marcelino Prasad Wimalaratne Kevin Tan
The Centre for Virtual Environments, University of Salford, Salford M5 4WT, United Kingdom.
Email : (t.fernando, l.marcelino)@salford.ac.uk, (g.d.s.wimalaratne, k.t.w.tan)@iti.salford.ac.uk
Tel : +44 161 295 2914 Fax + 44 161 295 2925

Abstract

This paper presents the design and implementation of a constraint-based virtual environment for supporting interactive assembly and maintenance tasks within an immersive virtual reality environment. The system architecture of the constraint-based virtual environment is based on the integration of components such as OpenGL Optimizer, Parasolid geometric kernel, a Constraint Engine and an Assembly Relationship Graph (ARG). The approach presented in this paper is based on pure geometric constraints. Techniques such as automatic constraint recognition, constraint satisfaction, constraint management and constrained motion are employed to support interactive assembly operations and realistic behaviour of assembly parts. The rendering and the interaction capabilities of the OpenGL Optimizer graphical API has been extended to run on a CAVE environment. Interactive assembly and disassembly operations are performed through a glove-based interface. The current system has been evaluated using two industrial case studies: digger mechanism and a helicopter rotor blade mechanism. This work is being carried out as a part of a research programme referred to as IPSEAM (Interactive Product Simulation Environment for Assessing Assembly and Maintenance), at the University of Salford.

KEYWORDS

Assembly Simulation, Maintenance Simulation, Constraint-based Modeling

---

1. INTRODUCTION

At present, very little research has been carried out to develop software environments for assessing maintainability issues during design of mechanical products. The consideration of maintainability issues requires support for assessing physical access, part handling, mobility around the product, difficulty in removing and replacing part etc. Such issues are typically assessed using physical prototypes. However, the building of physical prototypes is extremely expensive and time consuming, increasing the time to market the product. Furthermore, once a physical prototype has been built, there is usually so much inertia in the design that major changes are very difficult to incorporate. Therefore, there is a need for better software environments to assess the maintenance process, early in the design phase.

A research programme referred to as IPSEAM (Interactive Product Simulation Environment for Assessing Assembly and Maintainability) has been established at the Centre for Virtual Environment at Salford to investigate the development of such a software environment. The aim of this research is to investigate the design, implementation and evaluation of an *Interactive Product Simulation Environment*, which can support the assessment of ease of assembly and maintenance of a product during design. This research is addressing a major question concerning the viability and utility of virtual environments to perform design analysis on a virtual prototype as realistically as with a physical prototype. The specific objectives of this research are to:

1. Design and implement a constraint manager that supports constraint-based interaction between parts to assemble and disassemble parts as intuitively as performed on the physical prototype.

   Real-time simulation of physical constraints within the virtual environment is essential for supporting realistic interaction between virtual assembly components. Simulation of an assembly relationship can be considered as a constraint specification and satisfaction problem. For example, alignment of a shaft with a cylindrical hole involves satisfying an axis-alignment constraint. Furthermore external forces such as gravity, friction and forces resulting from collisions play a role in assembly operations. Disassembly operations involve breaking the previously defined constraints by applying an external force.

2. Design and implement an intuitive interface, which supports natural manipulation of constraint-based assembly parts.

   Maintainability assessment on virtual prototypes requires a sophisticated VR interface for supporting 3D direct manipulation of assembly parts. This research is investigating the type of operation required on the virtual prototype and develop an intuitive VR interface. The use of a CAVE like environment together with 3D input devices (e.g. gloves) is being use to provide an advance interface.

3. Conduct user-centred evaluation of the constraint-based virtual environment using realistic industrial case studies.

This work builds upon the first author's previous research work on interactive assembly modelling (Fernando et al, 1995; Fernando and Dew., 1995; Fa, et al 1993a; Fa, et al 1993b) and developments in graphic environments such as Optimizer from Silicon Graphics. This research work is being carried out in collaboration with Rolls Royce Aero-engine Group, British Aerospace, EDS Parasolid and D-Cubed Ltd.

This paper will discuss the following issues:

1. Current constraint-based modelling approaches employed within the IPSEAM system.
2. IPSEAM system architecture for supporting the CAVE environment and the glove-based interface.
3. The performance issues related to rendering and constraint management with respect to the current case studies.

2. RELATED WORK

The design and implementation of the IPSEAM requires bringing together various technologies such as virtual environments, constraint-based modelling, assembly modelling, CAD data representations and interfaces. This section summarises the state-of-the-art in these technologies.

2.1. Virtual Environments

A number of commercial and non-commercial virtual environments are currently available to support the development of virtual environments for industrial applications. However, a common weakness of the current commercial virtual environments is the lack of efficient geometric constraint management facilities such as run-time constraint detection and the maintenance of constraint consistencies for supporting accurate part positioning and constrained 3D manipulations. Hence 3D assembly modelling operations using direct manipulation are not feasible in the current virtual environments.

Furthermore, most of the current virtual environments are based on polygonal data. When importing CAD data, these virtual environments throw away important geometric information of the CAD models and convert all the geometric surfaces into a set of polygons. This lose of information reduce the gain obtained through the use of highly interactive systems in the engineering process because reintegration of manipulated data in the engineering tools is difficult and error prone. As a result, current virtual environments lack the semantic information necessary for supporting engineering operations such as assembly modelling.

The introduction of OpenGL Optimizer from Silicon Graphics has overcome this limitation by implementing a rich scenegraph which is capable of maintaining both surface representations and polygonal data of CAD models. Furthermore, the OpenGL Optimizer provides important features such as efficient occlusion and frustum culling, built-in algorithms for simplifying level of detail (LOD) of models, efficient surface tessellators without surface cracks and multi-threaded scenegraph operations. This graphics engine has been designed by SGI to support large CAD applications. Due to the improved functionality, OpenGL Optimizer has been chosen as the baseline virtual environment for developing the IPSEAM system.

2.2. Constraint-Based Modeling

Two main approaches are being pursued by researchers to support constraint-based interaction between assembly parts: Physically-based Modeling and Constraint-based Geometric Modeling. In **physically based modeling** (Baraff, 1995; Mirtich, and Canny, 1995; Bouma, and Vanecek, 1991), physical forces acting upon objects and their motion equations are integrated and solved at each time step, using standard numerical methods. During the simulation, collisions are detected (Bouma, and Vanecek, 1991; Ponamgi, et al 1997) at each time step and forces arising from such impacts (Baraff, 1995; Mirtich and Canny 1995) are calculated and new initial conditions are passed to the dynamic integrator to continue the simulation. Unfortunately, this approach is time consuming and therefore real-time simulation and interaction are only possible for a small number of components. Furthermore numerical instability can be a problem in this approach. In **constraint-based geometric modeling**, objects are accurately positioned in terms of geometric constraints. In this domain, much research has been conducted to develop efficient geometric constraint solvers by exploiting the geometric domain knowledge together with degrees of freedom of objects.

The first phase of this research program is focused on exploring the constraint-based geometric **modeling** approach. Future research will explore the simulation of physical forces within this framework. The following section summarises the state-of-the-art in constraint-based geometric modeling.

2.2.1. *A Characterisation of Constraint-Based Approaches.*

The current constraint-based approaches can be divided into two main categories: Equation-based and Geometric Constructive.

In the equation-based approach, the constraints are described as a set of simultaneous equations and solved using either numeric, symbolic or graph-based techniques. Numeric technique (Lin, et al 1981; Light, and Gossard 1982) solves equations using iterative methods such as Newton-Raphson. Symbolic technique (Kondo, 1992), solves the equations through symbolic algebraic methods, such as Grobner bases (Becher, 1993). Although the numeric and symbolic techniques are quite general, they can have convergence problems and are also computationally expensive, making them unsuitable for supporting interactive constraint-based operations in virtual environments. Graph-based technique first maintains constraints (equations) and variables in an undirected bipartite graph. This graph is then directed to give a sequence of constraint satisfaction. Examples of this technique include (Serrano, and Gossard 1992; Sannella, 1993).

In the Geometric Constructive approach, constraints are not translated into a unique system of equations as in the equation-based approach. Instead, a set of constructive steps is provided which place geometric elements relative to each other through rigid body transformations, according to the degrees of freedom (DOF) of the geometric entities. In this approach, DOF of geometric objects are considered as resources which are consumed by moving an object to satisfy a given constraint relative to a fixed geometry. Each constraint, upon being satisfied, reduces the DOF of an object and hence reduces the allowable rigid body motion of the object. Examples of this approach can be found in (1992; Kramer, 1992; Owen, 1991; Bouma 1995; Fa, et al 1993; Fernando, et al 1995).

The use of geometric knowledge, DOF of objects and graph-based techniques result in efficient constraint satisfaction algorithms in the Geometric Constructive approach. Therefore a constraint solver based on Geometric Constructive approach has been employed in this research to support constraint management within our Interactive Product Simulation Environment.

2.3. Representation of Assembly Relationships as Allowable Rigid-Body Motion

The assembly process consists of a succession of tasks, each of which consists of joining assembly parts (Components) to form the final assembly. Parts are considered joined when the necessary contacts and alignments between parts are established. These contacts and alignments are referred to as assembly relationships. These assembly relationships can be described in terms of geometric constraints and solved using the approaches described in Section 2.2.1.

Several researchers have proposed techniques (Morris, and Haynes, 1987; Kim, and Lee,1989; Turner, et al 1992; Mullins, and Anderson 1993; Fa, et al 1993a; Fernando, et al 1995) for representing assembly relationships in terms of the relative motion (DOF) permitted in each of the mating pairs. This approach is more efficient for simulating the interactive constrained motion of assemblies and hence is being used within the IPSEAM system. One important feature in this line of research is that assembly constraints are described as a kinematic problem.

2.3.1. *Representation of Multiple Constraints.*

When several constraints are associated with an assembly part, the resulting rigid body motion can be found by intersecting the rigid body motion of each constraint. In (Fa, et al 1993a; Fernando, et al 1995), technique called allowable motion intersection is presented to represent multiple constraints. Turner (1992) proposes similar concepts through constraint reduction techniques. Effectively, the resultant rigid body motion of two constraints is the intersection of the two sets of rigid-body motions of the original constraints. Intersections are performed for translation and rotational freedom independently. Refer to (Fa, et al 1993a; Fernando, et al 1995; Turner, et al 1992) for more information.

2.4. Interactive Assembly Constraint Specification

Two dimensional auto-constraint techniques are increasingly being used by CAD systems to build 2D constraint-based models. Bier (1990) proposed a 3D snapping technique for building 3D models. These concepts were further extended by Fa and Fernando (Fa, et al 1993a; Fernando et al 1995) to support interactive assembly **modeling**. In their approach constraints were recognised between geometric elements when the assembly parts were coming together. Such auto-constraint recognition techniques are being explored within our IPSEAM system to support complex assembly disassembly operations.

3. SYSTEM ARCHITECTURE

The baseline virtual environment is developed around the OpenGL Optimizer. The OpenGL Optimizer has been chosen as the graphics engine for the virtual environment due to its powerful CAD capabilities. However, the OpenGL Optimizer lacks of CAD interfaces to import data into the scenegraph. Therefore a CAD interface was developed as a part of the IPSEAM project for importing CAD data into the Optimizer scenegraph. This CAD interface is now capable of importing Parasolid models into the scenegraph while preserving the integrity of the CAD data.

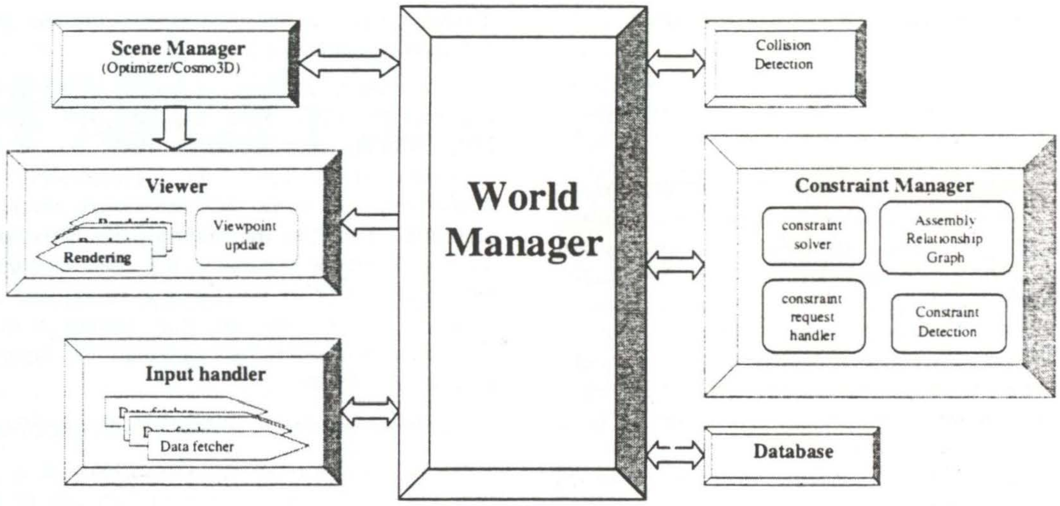

Figure 1: System architecture

The architecture of the IPSEAM (Fernando et al., 1999) has now been redesigned to utilise advanced VR display technologies (i.e. CAVE, Responsive Workbench), 3D input devices and multi-processor graphics architectures. The current IPSEAM system now supports multi-pipe rendering and multi thread processing to achieve the performance required by high-end VR systems. Multi-thread processing are used for supporting tracking (input devices and head), rendering (e.g. each wall on the CAVE is rendered by a separate thread) and computational modules (i.e. collision detection, constraint management).

Figure 1 shows the new architecture of the IPSEAM system. A brief description of this architecture is presented in this section.

3.1. The world manager

The main task of the world manager is to manage the interaction between the different components of the system. It also has the responsibility to synchronise various threads.

3.2. The Viewer

Extension to the Optimizer has been made to view the scene within different display technology (CAVE, Responsive Workbench). The rendering of images on different walls is performed employing parallel threads to provide real time response.

3.3. The Input Handler

Two different types of input are available: the control input and *user* input.

- The purpose of the ***control input*** is to configure and supervise the system (e.g. the configuration of the *user input* and output devices, inter-module communication). This input is achieved mainly though the keyboard (and the mouse).
- The *user interaction* is handled by the ***input handler***. It supports devices such as tracking system, gloves etc.) This input describes the *user* action/command within the virtual environment. Each input handler has its own thread to process data from the input device. These threads run in parallel with the rendering threads to achieve low latency.

Once the assembly parts are loaded into the scenegraph via the CAD interface, the input handler allows the user to grab and manipulate objects in the 3D space.

3.4. The Constraint Manager

The constraint manager comprises of four main modules: constraint detection, constraint solver, assembly relationship graph manager and constraint request handler. The constraint request handler is the interface between the constraint manager and the world manager. It processes the requests and directs the action to the constraint solver or the assembly graph.

The user manipulations are monitored by the collision detection module. While an object is being manipulated, the position of the moving object is sampled to identify collisions between the manipulated object and the surrounding objects. These collisions are passed on to the constraint detection module to identify potential assembly constraints between the collided surfaces.

The task of the constraint solver is to satisfy the specified constraints specified by the system in response to user interaction. The constraint solver satisfies a given set of constraints and produces relative rigid body motion for assembly relationships.

The Assembly Relationship Graph (ARG) maintains assembly relationships between the mating surfaces of assembly parts. The ARG is an undirected graph where each node represents either a geometric entity (mating surface) or a constraint. The nodes representing geometric entities are connected to constraint nodes using arcs to represent their assembly relationships. The ARG is not a solid representation scheme but is concerned with maintaining the relationship between assembly parts.

However, the geometric information for each mating surface is maintained within a corresponding entity node. This is done by providing a pointer to the surface node in the scenegraph. This geometric information is used by the constraint solver when evaluating and solving constraints.

The constrained rigid body motions of the assembly parts are used to support realistic manipulations of assemblies without breaking the existing assembly constraints. This is done by converting the 3D manipulation data received from the 3D input device into allowable rigid body motions. A particular manipulation of an assembly model is not allowed if it is not supported by its allowable rigid body motion.

3.5. The database

The database provides a repository for maintaining assembly components and the assembly relationships. This allows the user to save and retrieve assemblies during testing.

4. PERFORMANCE EVALUATION

The evaluation of the case studies has raised many issues, such as the performance issues and the user interface issues. This section presents the issues related to the real-time performance.

The initial system presented somewhat unsatisfactory results even for moderately complex scenes. To identify the performance bottleneck of the system, some measurements were made.

The system specification of the graphics machines used for the evaluation is presented in Table 1.

	Onyx2 InfiniteReality (Rack system)
CPU:	14x250MHz MIPS R10000
Main Memory:	4.5 GB
Secondary Instruction-Data Cache:	4 MB
Operating System:	IRIX64 Release 6.5
Graphics:	InfiniteReality2E (5x)

Table 1: System Specification of the Hardware

One graphic pipe and one process (single thread) was used for this evaluation. This means that the process that handles constraints between objects is also responsible for rendering the scene. If stereo rendering is used, then the same process renders the scene twice. The measurements were done counting every frame during a fixed interval. The number of frames was then divided by the time elapsed. In every test, care was taken for the entire scene to remain within the frustum. Motion of the entire scene was archived through translation and rotation within the frustum. The values in Table 2 represents the frame rate in a steady state, i.e. peaks were not considered.

The case study used in evaluating the system performance is the model of a digger (Figure 2) with 120939 triangles (225391 vertices).

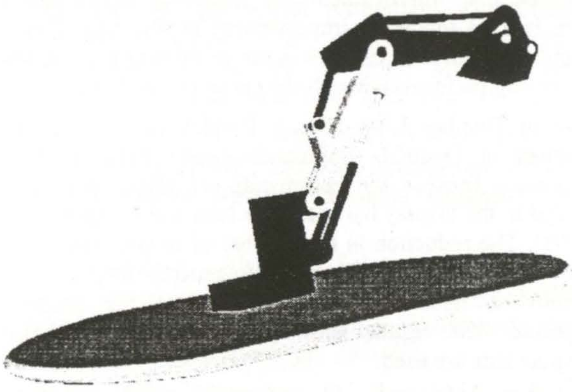

Figure 2: Assembled Digger

Action Description	Frames/sec
Whole scene moving	8.9
Move one constrained object	9.0

Table 2: IPSEAM performance in frames per second

By examining the data in Table 2 it is clear that the constraint management during interaction is not computer intensive and hence does not affect the performance.

It was clear from these tests that the performance bottleneck is due to rendering rather than the constraint management. As a result, the following steps were taken to increase the rendering performance.

The scene was tuned in different stages of the pipeline adapted from the openGL Optimizer Programming Guide (available on-line). The graphic pipeline as used by these scenes has no tessellation, reducing one phase in the host

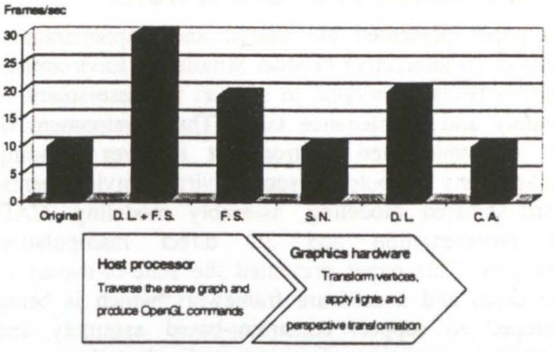

Figure 3: Graphic pipeline tuning

stage. A graphic representation of the conducted tests is shown in Figure 3. A description of the tests follows.

Collapse all the Appearances in the Scene (C.A.): This save many state changes in the OpenGL hardware. This feature organises all the objects with similar appearance, a situation usually not desirable for most virtual environments. However, this is capable of providing considerable improvement on rendering performance. The lack of rendering improvement in this case clearly highlights that state changes is not contributing to the low rendering performance with the chosen case studies.

Use of Display Lists (D.L.): Display lists reduce the number of OpenGL commands issued from the host processor because the commands are precompiled and stored in the display list cache for later use (Neider et al., 1993). The reduction in the number of issued commands will both save the time required to generate them and the bandwidth necessary to pass them to the graphics pipeline. The results show great improvement when display lists are used.

Use of Short (Integer) normals (S.N.): This converts the normal vectors in the scene graph from floating-point vectors to short-integer vectors. This shortening of the memory segments holding normals reduces the amount of data that must be sent from the host to the graphics pipeline. When normals are represented as integers, it improves the performance in situations where host-to-graphics-pipeline bandwidth is the limiting factor. The reduced data volume also enhances the performance by allowing more of the scene to reside in the display-list cache. The measured frame rate gives evidence that the system do not suffer from bandwidth restriction.

Use of Fans and Strips (F.S.): Converting the objects representation into fans and strips of triangles simplifies greatly the command generation and number of vertices processed. The result highlights the effect of using fans and strips on rendering performance.

By using these techniques (display lists, fans and strips) the rendering performance of the IPSEAM was increased by 300%.

5. CONCLUSIONS AND FUTURE WORK

This paper presented the design and implementation details of an Interactive Product Simulation Environment which is being developed to support the assessment of assembly and maintenance tasks. The development of such a sophisticated environment involves bringing together many technologies such as virtual environments, constraint-based modeling, assembly modeling, CAD data representation and 3D direct manipulation techniques. This paper presented the state-of-the-art in these areas and a software framework which is being developed to support constraint-based assembly and maintenance operations. Two case studies have been used to evaluate and demonstrate the constraint-based manipulation of assembly parts. Although the current implementation only has an interface to import Parasolid models, it is possible to develop interfaces to bring different CAD formats into the IPSEAM system.

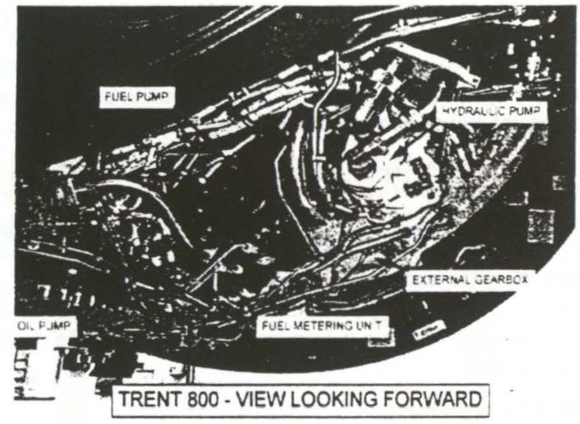

Figure 4: TRENT 800 case study

The IPSEAM environment presented in this paper is now being further developed to perform maintainability assessment of the Trent 800 aero-engine (Fig. 4). During this research, further attention will be given to the constraint management capabilities necessary for supporting maintenance.

6. ACKNOWLEDGMENTS

Thanks go to Rolls-Royce PLC, British Aerospace, EDS Parasolid and D-Cubed for providing case study material and support for this project.

7. REFERENCES

Baraff, D. 1995, "Interactive Simulation of Solid Rigid Bodies", IEEE Computer Graphics and Applications, pp 63-74.

Becher, T and Weispfenning, V, 1993, "Grobner Bases – A Computational Approach to Commutative Algebra", Graduate Texts in Mathematics, Springer-Verlag, New York

Bouma, W.J. and Vanecek, G., 1991,"Collision Detection and Analysis in a Physically-based Simulation", Proceedings, Eurographics Workshop on Animation and Simulation, pp. 191-203.

Bouma, W., Fudos I., Hoffmann C., Cai J. and Paige R., 1995,"Geometric Constrain Solver", Computer-Aided Design, Vol. 27, No. 6, pp. 487-501.

Bier, A., 1990, "Snap-Dragging in Three Dimensions", Symposium on Interactive 3D Graphics, pp. 193-204.

Fa, M., Fernando, T., and Dew, P.M., 1993a "Interactive Constraint-based Solid Modeling using Allowable Motion", ACM/SIGGRAPH Symposium on Solid Modeling and Applications, pp. 243-252.

Fa, M., Fernando, T., and Dew, P.M., 1993b "Direct 3D Manipulation Techniques for Interactive Constraint-based Solid Modeling", Computer Graphics Forum: Conference Issue, Vol.12, No. 3, pp. 237-248

Fernando, T., Fa, M., Dew, P.M. and Munlin M., 1995, "Constraint-based 3D Manipulation Techniques for Virtual Environments", Proceedings, International State of the Art Conference (BCS) on Virtual Reality

Applications, Leeds, 1994. This was also published in Virtual Reality Applications (et al. Earnshaw), Academic Press, 1995, pp. 71-89.

Fernando, T., and Dew, P.M., 1995, "Constraint-based Interaction Techniques for Supporting A Distributed Collaborative Engineering Environment", Proceedings, First Workshop on Simulation and Interaction in Virtual Environments (SIVE'95), Iowa City, pp. 265-270.

Fernando, T., Wimalaratne, P and Tan, K.; 1999, " Constraint-based Virtual Environment dof supporting assembly and maintainability tasks ", Proceedings of DETC99, ASME Computers in Engineering Conference, September, 1999, Las Vegas, Nevada

Jayaram, S., Kreitzer, R., Jayaram, U, 1998, "Preserving Design Intent Between Virtual Prototyping and CAD Systems", Proceedings of the ASME DETC/CIE

Jayaram, S., Wang, Y., Jayaram, U., Lyons, K., Hart, P., 1999, "A Virtual Assembly Design Environment", IEEE VRAIS Conference.

Kim, S.H. and Lee, K., 1989, "Assembly Modeling System for Dynamic and Kinematic Analysis", Computer-Aided Design, Vol.21, pp. 2-12.

Kondo, K., 1992, "Algebraic Method for Manipulation of Dimensional Relationships in Geometric Models", Computer-Aided Design, Vol. 24, pp.141-147.

Kramer, G. A., 1992, "A Geometric Constraint Engine", Artificial Intelligence, Vol. 58, pp. 327-360.

Lin, V.C., Gossard, D.C. and Light, R.A., 1981, "Variational Geometry in Computer Aided Design", ACM Computer Graphics (SIGGRAPH'81), Vol 15, pp. 171-175.

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

Mirtich, B. and Canny, J., 1995, "Impulse-based Simulation of Rigid Bodies", Procedings, Simposium on Interactive 3D Graphics.

Morris, G.H. and Haynes, L.S., 1987, "Robotic Assembly by Constraints", Procedings, IEEE Conf. Robotics Automation, pp. 1507-1515.

Mullins, S.H. & Anderson, D.C., 1993, "A Positioning Algorithm for Mechanical Assemblies with Closed Kinematic Chains in Three Dimensions", Proceedings, Symposium on Solid Modeling and Applications, ACM Press, New York, pp. 271-282.

Neider, J.; Davis, T.; Woo, M.; 1993, "OpenGL Programming Guide", OpenGL Architecture Review Board, Addison-Wesley Publishing Company, pp. 117-138.

Owen, J.C., 1991, "Algebraic Solution for Geometry from Dimensional Constraints", ACM/SIGGRAPH Symposium on Solid Modeling Foundations and CAD/CAM Applications, pp. 397-407.

Ponamgi, M.K., Monocha, D., Lin, M.C., 1997, "Incremental Algorithms for Collision Detection Between Polygonal Models", IEEE Transaction on Visualization and Computer Graphics, Vol.3, No.1, pp. 51-64.

Sannella, M., 1993, "The SkyBlue Constraint Solver and its Applications", First Principles and Practice of Constraint Programming Workshop (PPCP'93), Newport, RI.

Serrano, D. and Gossard, D,. 1992,. "Tools and Techniques for Conceptual Design", Artificial Intelligence in Engineering Design, Vol. I, C. Tong and D. Sriram (Eds), pp. 71-116.

Turner, J., Subramaniam, S. and Gupta, S., 1992, "Constraint Representation and Reduction in Assembly Modeling and Analysis", IEEE Transaction on Robotics and Automation, Vol.8, No.6.