Intelligent Automation and Soft Computing, Vol. 15, No. X, pp. 1-15, 2009

A CONSTRAINT-BASED METHODOLOGY FOR PRODUCT DESIGN WITH VIRTUAL REALITY

YONGMIN ZHONG[1], XIAOBU YUAN[2], WEIYIN MA[3], BIJAN SHIRINZADEH[1]

[1]Department of Mechanical Engineering
Monash University
Australia

[2]School of Computer Science
University of Windsor
Canada

[3]Department of Manufacturing Engineering and Engineering Management
City University of Hong Kong
Hong Kong, China

ABSTRACT—This paper presents a constraint-based methodology for product design with advanced virtual reality technologies. A hierarchically structured and constraint-based data model is developed to support product design from features to parts and further to assemblies in a VR environment. Product design in the VR environment is performed in an intuitive manner through precise constraint-based manipulations. Constraint-based manipulations are accompanied with automatic constraint recognition and precise constraint satisfaction to establish constraints between objects, and are further realized by allowable motions for precise 3D interactions in the VR environment. The allowable motions are represented as a mathematical matrix and derived from constraints between objects by constraint solving. A procedure-based degrees-of-freedom combination approach is presented for 3D constraint solving. A rule-based constraint recognition engine is developed for both constraint-based manipulations and implicitly incorporating constraints into the VR environment. An intuitive method is presented for recognizing pairs of mating features between assembly components. Examples are presented to demonstrate the efficacy of the proposed methodology.

Key Words: virtual reality, product design, model representation, constraint-based manipulation, allowable motions, constraint recognition and constraint solving

1. INTRODUCTION

Current product development mainly relies on computer-aided design (CAD) for fixturing, assembly and automation [1-3]. Virtual reality (VR) technology brings a new paradigm to the community of CAD by allowing design engineers to visualize, analyze, optimize, and implement products and the associated processes without the need of physical objects. Its standard features of intuitive three-dimensional (3D) interaction, direct manipulation, and 3D immersion enable users to fully explore their creativity in concept development. In addition, working in a virtual environment has the advantage of finding and resolving feasibility problems of assembly, manufacturing, and maintenance in the early stages. Therefore, VR provides an efficient way of producing intelligent, flexible, cost-effective, and high quality products.

However, it is not easy to combine VR with product design. Currently, most of the current VR systems only offer very limited tools for object creation [4, 5]. In these systems, complex virtual object models are first created by using the CAD software (AutoCAD, UGII and ProE, etc.), and then are imported into the VR environment for visualization and analysis. The limitations with these systems are: (a) The models are first created in CAD systems by specifying detailed dimensions while these dimensions are not precisely defined in the concept stage; (b) Topological relationships and constraints between entities and parametric information are lost when transferring the models from CAD systems to VR systems; (c) To modify the models, one must return to CAD systems to make desired changes and re-import the revised models into VR systems for verification.

To overcome the above problems, an obvious solution is to directly create and modify object models in a VR environment. All design activities are carried out in the VR environment. Users can intuitively create and modify 3D objects through 3D direct manipulations, and visualize and analyze the design in the same system without any data transfer. However, developing such a new VR-based product design system is not easy task. Although the existing VR design systems [6-8] provides some functions for direct object creation, they lack sophisticated design and modification tools for creating complex product models in VR environments. One of the most important reasons is that the finite resolution of virtual objects without information on topologies and constraints is not suited to represent product models for design purposes. Another important reason is that the limited accuracy and reliability of 3D input and output devices also prevent users from precise design activities.

This paper presents a constraint-based methodology for product design with advanced VR technologies. The goal of this research is to develop an intuitive 3D virtual environment for product design. Compared with other existing solutions, this paper establishes a hierarchical structured and constraint-based data model to support product design from features, parts to assemblies in the VR environment. Constraint-based manipulations are constructed to perform precise 3D design activities in the VR environment. A mathematical matrix is presented for precise representation of allowable motions of an object. A procedure-based degrees-of-freedom (DOF) combination method for solving 3D constraints is presented for deriving the allowable motions. This method combines DOF analysis with 3D direct manipulations in the VR environment for intuitive constraint solving. A rule-based constraint recognition engine is developed for both constraint-based manipulations and implicitly incorporating constraints into the VR environment. An intuitive method is presented for recognizing pairs of mating feature between assembly components. A prototype system has been implemented for product design through intuitive and precise constraint-based manipulations in the VR environment.

2. PREVIOUS WORK

Product design with VR technologies is not new. One of the earliest systems was 3DM [9]. 3DM allows users to interactively create simple geometric objects, such as cylinders and spheres, in the VR environment. It includes several grid and snap functions. However, it lacks many other aids and constraints that are necessary to accomplish precise work. JDCAD also tackled many issues for interactive 3D objects modelling [10]. Users could directly interact in 3D space using a 6-DOF input device. However, only simple solids can be created in JDCAD. Dani and Gadh [11] presented a COVIRDS system for concept design in the VR environment. This system is based on design features and a geometric modelling kernel ACIS is used for their development. The precise interactions mainly rely on voice commands. Stork and Maidhof also reported a method for interactive and precise solid modelling using a 3D input device [12]. In this method, precise modelling is realized using 3D grids, grid snapping and discretized dimensions. Constraint-based interactions are based on pre-defined rules for feature-based modifications. Although precise modelling can be ensured, the constraint-based interactions are too rigid for extensive use. Kiyokawa et al reported a two-handed modelling environment for efficient manipulations in the VR environment [13]. The result was extended to a collaborative VR environment for object creation [8]. However, only simple objects can be created in this system and complex models are constructed by placing some simple objects together. Nishino et al reported some results on gesture-based 3D shape creation [14]. A 3D modeler is developed to create complex shapes by combining the defined hand actions, while precise interactions are not included. Gao et al reported a method on constraint-based solid modelling in the semi-immersive VR environment [7]. In this method, the manipulations to a primitive depend on shape control points (SCP) on a primitive instead of the primitive itself and a 3D mouse must be set to the SCP for manipulating the SCP. In addition, the SCP cannot sufficiently reflect the natural behaviors of the geometric elements of the primitive. Therefore, the interactions in the virtual environment are unintuitive and inconvenient. Ji developed a virtual design and assembly system [6]. In this system, a variational design approach is used for part creation and a knowledge-based library is used to support part assembly. However, only ideas and concepts are discussed excluding any results. Furthermore, many technical issues such as real-time model update and real-time interactions during the product design process in the VR environment are not discussed. Recently, haptics-based object design has been actively investigated, since haptics or force feedback provides a new interaction paradigm between designers and computers. However, most of these studies are mainly focused on modelling of free-form surfaces [15] and deformable objects [16-18], while the investigations on real-time design and modification of mechanical parts and assemblies with haptic devices are still limited.

There also are a few researches focused on the integration of VR, constraint solving and 3D direct manipulations into product design. Fa et al reported a method for 3D object placement in a VR environment [19]. Fernando et al further extended the method to a shared virtual environment [20] and presented a software architecture of a constraint-based virtual environment [21]. Marcelino further extended Fernando’s results and presented a constraint manager to support maintenance operations in a virtual environment [22]. The most important contribution of their method is the concept in constraining 3D direct manipulations through the allowable motions of the object being manipulated for precise locations and operations. Since the allowable motions are derived from predefined rules that are related to constraint types and geometric element types, they are difficult to be used extensively. Furthermore, only simple constraints such as against, collinear and concentric are treated and object models are still created from CAD systems and then are imported into the VR environment.

3. MODEL REPRESENTATION

A fundamental problem for product design in a virtual environment is model representation. Currently, there has been little research on accommodating precise CAD models in a VR environment. If CAD formats were directly used in a VR environment, the online processing time for visualizing a typical CAD model would make it impossible to interact in real time. The polygon model used in most VR systems provides the illusion of being immersed, but it may not be able to precisely define the object geometry. The use of a high-resolution model in a VR environment can increase model precision. However, the system may not be able to respond in real time either. On the other hand, it is difficult to carry out design activities due to the lack of topological relationships and constraint information in the polygon model. Therefore, it is necessary to develop a suitable model representation to support product design in a VR environment. The model representation not only needs to support real-time visualization and interaction in a VR environment, but it also needs to support design activities as well as to reflect the design process.

A hierarchically structured and constraint-based data model is presented to support product design in the VR environment (Figure 1). This data model includes six-levels of information, i.e. assemblies, parts, features, feature elements, geometric and topological relationships, and polygons. The data model integrates a high-level constraint-based model for precise object definition, a mid-level CSG/B-rep hybrid solid model to support the hierarchical geometry abstractions and object creation, and a low-level polygon model for real-time visualization and interaction in the VR environment. The information in the high-level model can be divided into two types, i.e. the object information on the different levels and the constraint information on the different levels. The object can be an assembly, a part, a feature or a feature element. The constraints on each object level that summarize the associativities between the individual objects on the same level not only provide precise object definition, but also provide a convenient way to realize precise 3D interactions. Every object level is represented as a graph-based structure [23, 24]. The advantage of the graph-based structure is that it provides the convenience to establish and edit a graph through interactions, and adding or deleting a node can be easily done in the graph representation through graphical interactions.

The mid-level solid model is the geometric and topological description of an object and is represented as a CSG/B-rep hybrid structure. It not only provides the geometric and topological information of an object to support the hierarchical geometry abstractions and object creation, but it also provides the convenience for interactive feature-based product design. Each face in the B-rep model can be identified by a 2-tuple (Fid, Eid), where Fid and Eid are the identity numbers of a feature and a feature element respectively, which makes it concise and convenient to index between features, feature elements and individual faces in the B-rep model.

The low-level polygon model is a triangle-based representation that corresponds to the mid-level B-rep solid model. It describes each face in the B-rep solid model as a common array of vertices together with a connection list that defines each facet as a set of indices into the array of vertices.

Constraints are embedded in the solid model and are organized at different levels to reflect the entire design process from features to parts and further to assemblies.

Level 1 is the assembly model representation. An assembly consists of parts and the constraints between these parts. The constraints on this level reflect the spatial position relationships between the different parts, i.e. the assembly relationships, which are called external part constraints. An external part constraint has one direction and this direction is dependent on those of the external feature constraints included in the external part constraint. Typical external part constraints include key-slot fitting and shaft-hole fitting.

Level 2 is the part model representation. A part consists of features and the constraints between these features. The constraints on this level represent the spatial position relationships between the different features, and they are called external feature constraints. An external feature constraint has one direction and this direction is dependent on those of the external element constraints included in the external feature constraint. Since a part is subdivided into a set of features, an external part constraint on the first level is decomposed into a set of the external feature constraints between the features that individually belong to different parts on this level.

The constraints on Level 3 are those between feature elements and are called feature element constraints. They can be further divided into feature element geometric constraint and feature element topological constraints according to their attributes. The feature element geometric

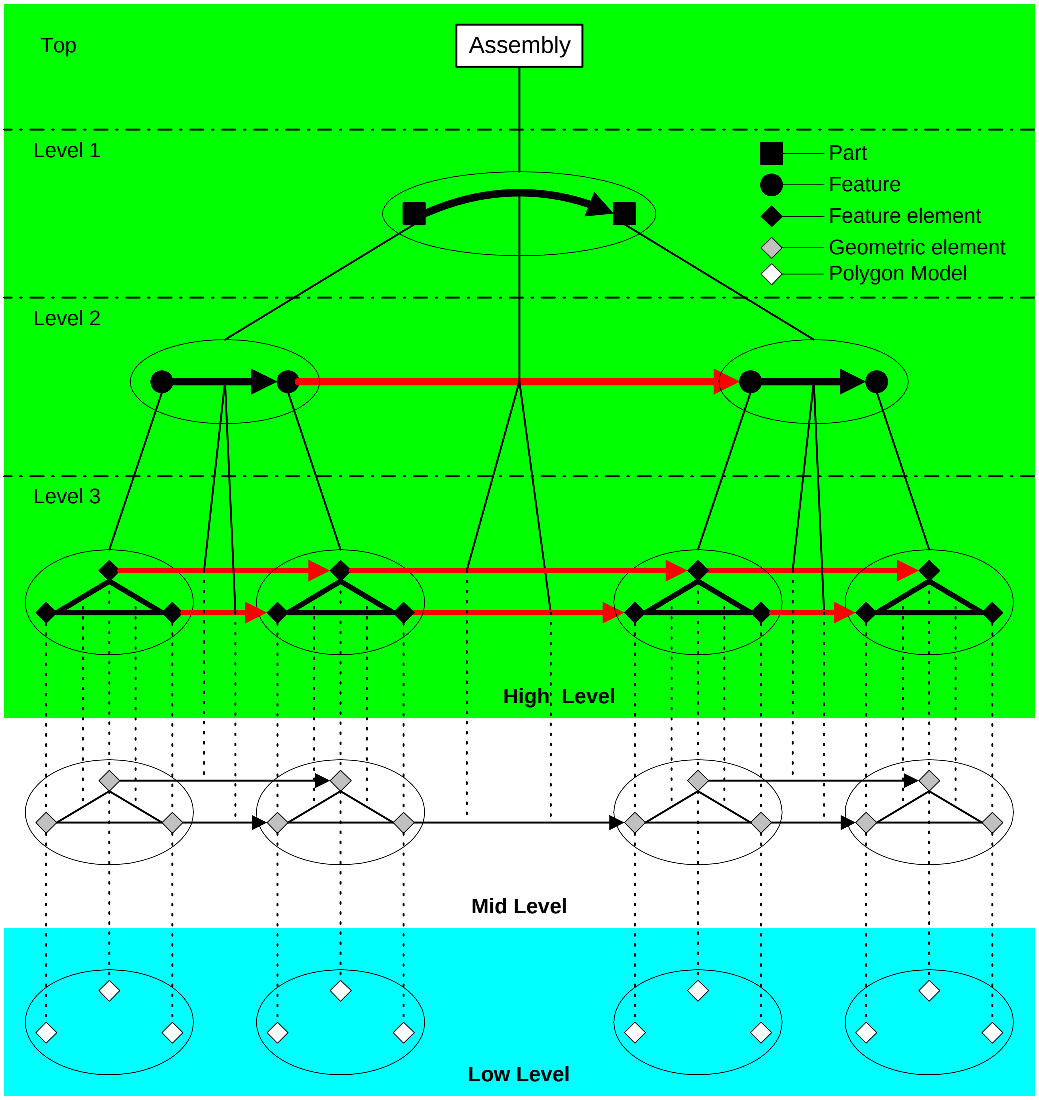


Figure 1. Hierarchically structured and constraint-based data model.

constraints represent the spatial position relationships between feature elements and are described as a face-based representation, such as coincident faces, against faces, alignment faces, parallel faces, perpendicular faces, distance faces and angular faces. The feature element topological constraints represent the topological relationships between feature elements and are described as an edge-based representation, such as co-edge and co-circle. Furthermore, feature element constraints can also be further divided into internal feature element constraints and external feature element constraints according to the feature elements involved in them are belonged to the same feature or different features.

Constraints indicated by the white edges on Level 3 represent the spatial position relationships between the feature elements that individually belong to different features, and they are called external feature element constraints. The external feature element constraints consist of feature element geometric constraints. Since external feature constraints are difficult to represent, we subdivide a feature into a set of feature elements. Correspondingly, an external feature constraint on Level 2 is subdivided into a set of external feature element constraints on this level. An external feature element constraint has one direction and this direction points to the feature element constrained.

A feature at Level 2 is represented as feature elements and the constraints between these feature elements, which are indicated by the black edges at Level 3. The definition of feature elements is equivalent to feature entities discussed in [25]. Feature elements can be further divided into real feature elements that correspond to the elements of solid models such as the six faces of a block and the bottom face of a blind-hole, and virtual feature elements that are not included in solid models such as the axis of a cylinder and the top face of a cylindrical blind-face. The constraints indicated by black edges represent the spatial position relationships between the feature elements that belong to a common feature and they are called the internal feature element constraints. The internal feature element constraints define the shape of a feature and are non-directional. The internal feature element constraints consist of feature element topological constraints and feature element geometric constraints.

The mid level provides the geometric representation of a feature element and the topological or geometric representation of a feature element constraint. A feature element can be mapped to its corresponding geometric element on this level. Correspondingly, a feature element constraint can be mapped to a constraint between geometric elements, i.e. a topological or geometric constraint. The shape of a feature element is represented by its corresponding geometric element with algebraic equations. Feature element constraints are represented by the constraints between geometric elements, i.e. topological and geometric constraints.

The low-level provides the polygon representation of a geometric element in the B-rep solid model.

It is not difficult to see, feature elements and feature element constraints are the most basic components in the entire constraint-based model. Both an external feature constraint on the feature level and an external part constraint on the part level are an abstraction of external feature element constraints, and all the constraints in the constraint-based model can be finally mapped to topological and goemtric constraints. The constraint relationships at different levels can be described as:

$$\boldsymbol{\Gamma f}(\text{internal element topological constraints}) \rightarrow \text{topological constraints}$$

$$\textbf{Љ}\boldsymbol{f}(\text{internal element geometric constraints}) \rightarrow \text{geometric constraints}$$

$$\textbf{Њ}\boldsymbol{f}(\text{external feature element constraints}) \rightarrow \text{geometric constraints}$$

$$\{\text{external feature element constraints}\} \subseteq \{\text{external feature constraints}\}$$

$$\{\text{external feature constraints}\} \subseteq \{\text{external part constraints}\}$$

where, the function Γf represents the mapping from an internal feature element topological constraint to a topological constraint, the function Љf represents the mapping from an internal feature element geometric constraint to a geometric constraint, the function Њf represents the mapping from an external feature element constraint to a geometric constraint, and {A} represents a set of A.

4. CONSTRAINT-BASED MANIPULATIONS

For every object in the virtual environment, such as a feature element, a feature, and a part, it takes an event list as its attribute. Each event in the event list has a link to an action list, and the action list determines the actions to take place when the event occurs. In the event of object manipulation, such as grasping, moving, or dropping off an object, corresponding actions generate the matching constraints to restrict allowable motions of the object. Subsequently, the hand motions acted on the object will be transferred to the allowable translations and/or rotations of the object. The following subsections discuss constraint-based manipulation, including the representation of allowable motions, constraint solving and constraint recognition.

4.1 Representation of allowable motions

A constraint on an object reduces the DOF of the object under manipulation, and imposes a confinement to further operations on the object. For every object in the 3D space, its configuration space has six DOF, including three for translations and three for rotations. The three translational and three rotational DOF are linearly independent from each other. Therefore, allowable motions can be formulated as the following matrix with these basic DOF.

$$\begin{bmatrix} T_x & R_x & T_{x\min} & T_{x\max} & R_{x\min} & R_{x\max} \\ T_y & R_y & T_{y\min} & T_{y\max} & R_{y\min} & R_{y\max} \\ T_z & R_z & T_{z\min} & T_{z\max} & R_{z\min} & R_{z\max} \end{bmatrix} \tag{1}$$

In (1), the elements in the first two columns are the translations along and rotations around the three axes X, Y and Z, respectively. These elements take up a value of either one or zero to indicate if motions are allowed in the corresponding DOF. The pairs of elements in the third and fourth columns determine the allowable ranges of translations and each range is defined by the minimum and maximum values. Those in the last two columns determine the allowable ranges of the three rotations.

4.2 Constraint solving for derivation of allowable motions

Most constraint solving algorithms can only deal with 2D constraints. Comparatively, constraint solving upon 3D constraints is still limited because of the complexity of constraint solving in 3D. Here, we combine DOF analysis and 3D direct manipulations in a VR environment and further present a procedure-based freedom combination method for solving 3D constraints. This method not only overcomes the limitation of the most constraint solving methods, i.e. only superficial and low-level constraints are solved, but it also solves constraints in an intuitive manner.

Constraints on an object limit its geometric displacements with respect to other objects, and thus a constraint applied to the object can be mapped to each of the DOF of this object. This mapping can be also extended to establish the correspondence between a set of constraints and a combined DOF. Therefore, constraints can be obtained by analyzing and reasoning the DOF of an object, and can further be solved by combining the DOF corresponding to each constraint.

Under the restriction of constraints, the allowable motions of an object can be derived from the remaining DOF of the object. Any action of object manipulation, such as grasping an object, is interpreted by the constraint solver as a request to the remaining DOF of the object. Initially, all objects are unconstrained, and free to move and rotate in any directions. When the constraints are simple, a straightforward analysis of the remaining DOF tells the allowable motions. For example, the action of placing a cylinder onto a block with an axis-alignment produces an <face-against>

constraint and a <line-alignment> constraint (Figure 2). The cylinder is still able to translate along the X/Z axis and rotate around the Y axis under the <face-against> constraint, and to translate along and rotate around the Y axis under the <line-alignment> constraint. The corresponding matrices for the allowable motions of the cylinder are given in Eqs. (2) and (3), respectively.

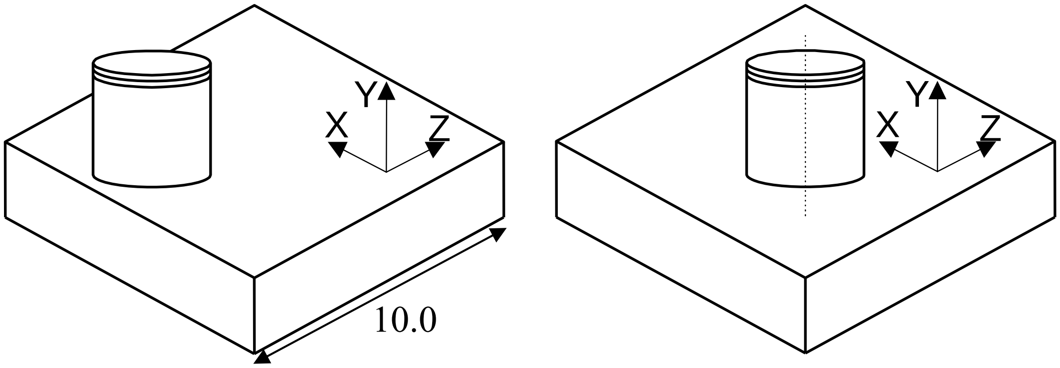


Figure 2. The "against" and "line-alignment" constraints between two cylinders.

In case of a group of multiple constraints on one object, the allowable motions of an object can be derived by combining the DOF corresponding to each constraint. After one constraint introduces a limit to an object on its motions, the object tries to satisfy the remaining constraints one at a time. The new constraint introduces new restrictions to the allowable motion of the object, which are then inserted into the constraint-based data model. An update of the model with the new constraint results in a revision to the remaining DOF of the object. This revision goes on until all the constraints in the group are processed. The refinement of allowable motions of the object is finally applied to object manipulation.

$$\begin{bmatrix} 1 & 0 & -10.0 & 10.0 & 0.0 & 0.0 \\ 0 & 1 & 0.0 & 0.0 & 0.0 & 2\pi \\ 1 & 0 & -10.0 & 10.0 & 0.0 & 0.0 \end{bmatrix} \tag{2}$$

$$\begin{bmatrix} 0 & 0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 1 & 1 & 0.0 & 10.0 & 0.0 & 2\pi \\ 0 & 0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \tag{3}$$

This process of combining multiple constraints is simply realized by combining the matrices of allowable motions for each of the individual constraints. The new matrix is obtained by applying the "AND" Boolean operation to the elements one by one in the first and second columns, and the intersections of the allowable ranges in the next two pairs of columns. For example, by combining the matrices in (2) and (3), a new matrix of allowable motions is obtained in (4) that satisfies both the <face-against> and <line-alignment> constraints in Figure 2. The result shows that the combined DOF only allow the cylinder to rotate around the Y axis. Translations are no longer allowed due to the multiple constraints.

$$\begin{bmatrix} 0 & 0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0 & 1 & 0.0 & 0.0 & 0.0 & 2\pi \\ 0 & 0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix} \quad (4)$$

Since the "AND" Boolean operation of elements is independent of the sequences of elements and the intersection of allowable ranges has nothing to do with the sequences of allowable ranges, the sequences of the constraints involved in the combination process have no effect on the final combination result. In another word, the sequences of allowable matrices involved in the combination process do not affect the final combination result.

4.3 Rule-based constraint recognition

Constraints are implicitly incorporated into the VR environment for constraint-based manipulations by automatic constraint recognition. Constraint recognition refers to verification of the current positions and orientations between two objects to see if they satisfy a particular type of constraint within a given tolerance. While object manipulations are being performed, as soon as a moving event or a release event occurs, an automatic constraint recognition process is triggered to detect all the possible constraints between the related objects. The system recognizes the constraints between objects from the current position and orientation of the being manipulated object according to a rule base. The rule base defines the rules for recognizing specific constraints. The constraints include against, alignment, parallelism, perpendicularity, distance, co-circle and co-edge. If the current positions and orientations between two objects satisfy the conditions of some constraint within the given tolerance, the corresponding constraint is recognized. Once a constraint is recognized within the given tolerance, it is highlighted and awaits the user's confirmation. If the object is further manipulated continuously within the given time, the currently recognized constraint is ignored and the constraint recognition is restarted. Otherwise, the currently recognized constraint is confirmed and the desired constraint is obtained. Furthermore, a dynamic tolerance is adopted into the constraint recognition to improve the efficiency of constraint recognition. If the desired constraint is not recognized within the given tolerance, the tolerance is enlarged according to the given step until the desired constraint is recognized.

5. IMPLEMENTATION AND RESULTS

A prototype system for product design in a VR environment through intuitive and precise 3D constraint-based manipulations has been implemented. Experiments have been conducted to examine the efficacy of the proposed methodology for product design from features, to parts and further to assemblies in a VR environment.

5.1 Part creation

Parts are created through locating feature primitives and Boolean operations between feature primitives or between feature primitives and intermediate objects. The location of a feature primitive relative to other features in a part can be completely determined by a series of constraint-based manipulations. The desired Boolean operation can be selected through 3D menu interaction. Visual feedback is also used to provide further assistance for part design. Shown in Figure 3 is an example of part creation. While the user is trying to place a cylinder for "subtraction", a coordinate frame with a set of arrows appears in Figure 3(b) and (c) to indicate the directions of allowable motions. A forward arrow attached at the end of an axis indicates a

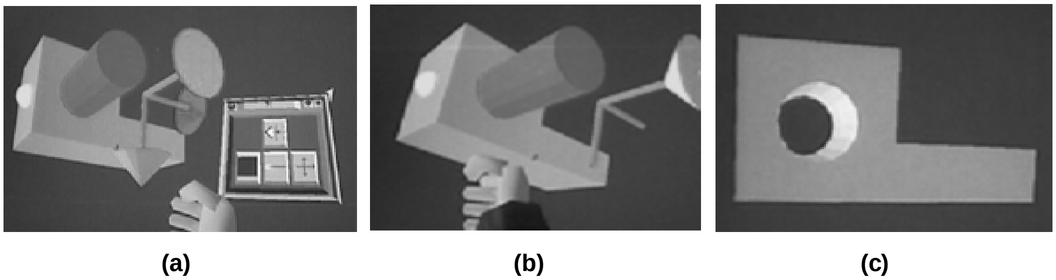

(a) (b) (c)

Figure 3. Part creation by using the “subtraction” Boolean operation.

translation along the axis, and an attached inverse arrow indicates a rotational around the axis. Figure 4 shows some other parts created through precise constraint-based manipulations in the VR environment.

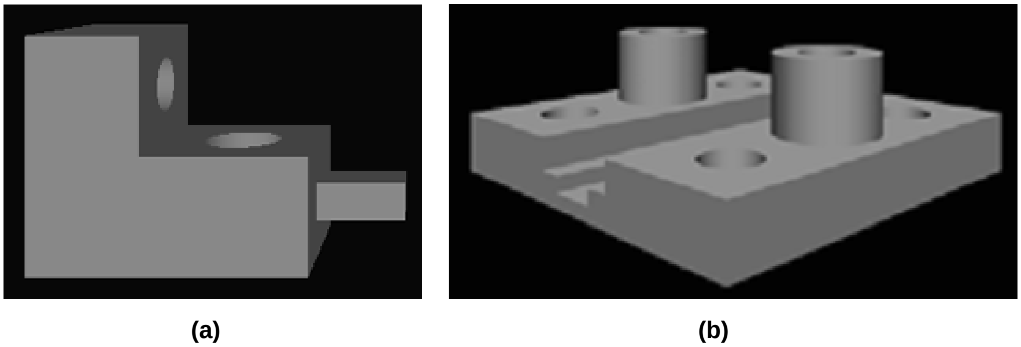

(a) (b)

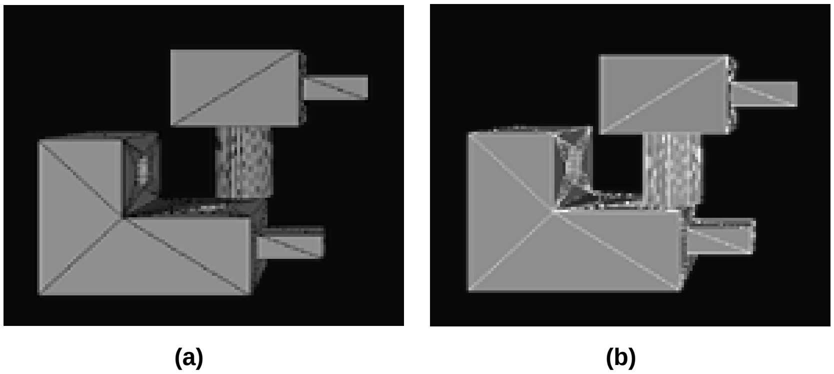
Figure 4. Parts created in the VR environment.

5.2 Assembly modelling

Assembly modelling requires the realization of all mating features to establish face contacts between assembling components and assembled components. This process involves four steps to interactively identify, automatically recognize, and precisely match each pair of mating features.

In the first step, a concept of “the offset solid of a component” is used to determine the assembling and assembled components for reducing the 3D interactive manipulations and enhancing the modelling efficiency. The offset solid of a component is an adjacent domain of the component. It is a new solid generated by moving every surface that constitutes the component a common distance along the normal of the surface. Once the user grasps a component, the component is treated as the assembled component and the offset solid of the assembled component is established correspondingly. The component currently being collided with the offset solid of the assembled component is regarded as the assembling component.

In the second step, the “face-contact” constraint between features of the assembling and assembled components helps to identify the pair of matching features while a user brings the assembling component close to the assembled component (Figure 5(a)). The “face-contact” constraint is the same as the “face-against” constraint except that the virtual feature elements of a negative feature are also included into the process of recognizing a <face-against> constraint. A pair of features can be identified from the involved feature elements by recognizing a “face-



(a) (b)

Figure 5. Face contact between the assembly and assembled components.

contact” constraint between the feature elements respectively belonged to the features, which are included in the assembling and assembled components respectively.

The pair of identified features defines a small region that consists of their child features and themselves for finding the pairs of mating features. In the third step, feature recognition checks and highlights all pairs of feasible mating features in the small region according to the criteria in a knowledge base [26]. This knowledge base defines all the conditions which mating features must satisfy. For example, a cylinder may mate with a straight hole if they have the same radius, but can never mate with a slot. The highlighted features shown in Figure 5(b) are the mating features unless the user moves the assembling component away from the current assembled component.

In the final step, the user completes the assembly of the assembling and assembled components with the assistance of constraints involved in the mating features. If there is one pair of mating features between the assembly and assembled components, the allowable motions of the assembled component are the allowable motions of the mating feature included in the assembled component. The allowable motions of the mating feature included in the assembled component can be obtained from the external element constraints applied to the mating feature through constraint solving. If there are multiple pairs of mating features between the assembly and assembled components, the allowable motions of the assembled component are obtained through the DOF combination of the allowable motions of each mating feature included in the assembled component. The method for the DOF combination is similar to that described in Section 4.2. The difference is that the “union” Boolean operation of the rotational DOF in each allowable matrix is always zero since each allowable motion matrix respectively belongs to the different features, which leads to that the rotational DOF in each allowable matrix do not share a common axis. The final assembly and its components are shown in Figure 6. Two other examples are given in Figure 7.

6. CONCLUSIONS AND FUTURE WORK

Presented in this paper is a new constraint-based methodology for product design with advanced VR technologies. By developing a hierarchically structured and constraint-based data model, the design process from features to parts and further to assemblies can be described and visualized in the VR environment, and constraints are provided for obtaining precise constraint-based manipulations in the VR environment. Constraint-based manipulations are constructed to carry out the precise product design activities in an intuitive manner. A mathematical matrix is presented for precise representation of allowable motions. A procedure-based DOF combination

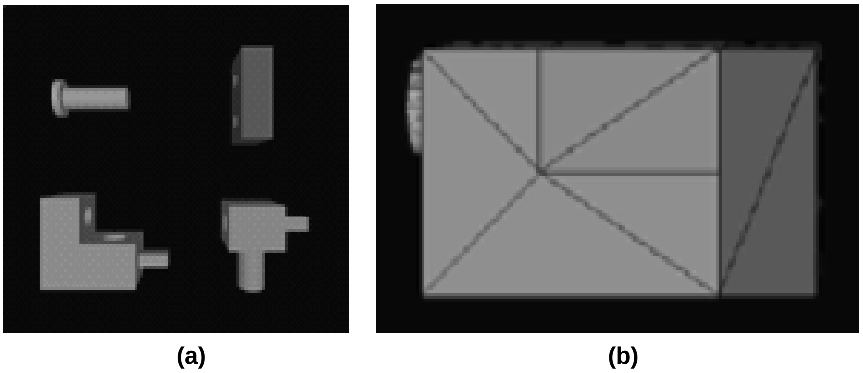


Figure 6. A created assembly from its components in the VR environment.

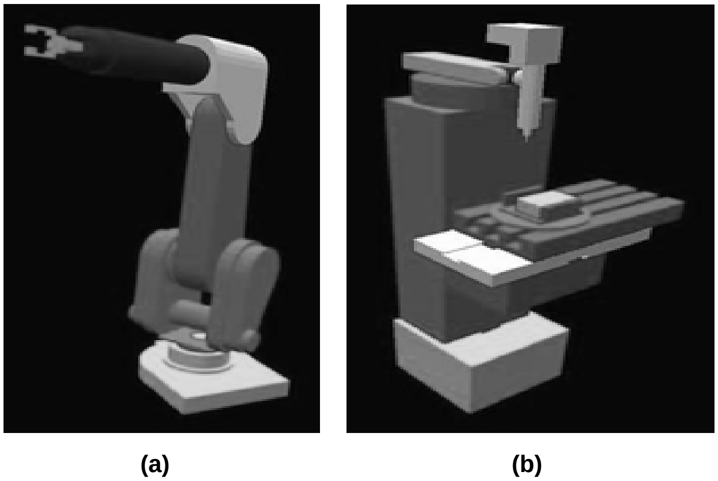


Figure 7. Assemblies created in the VR environment.

constraint solving method is presented for deriving allowable motions. A rule-based constraint recognition engine is developed for both constraint-based manipulations and implicitly incorporating constraints into the VR environment. An intuitive method is presented for recognizing pairs of mating feature between assembly components. A prototype system has been implemented for product design through precise and intuitive 3D constraint-based manipulations in the VR environment.

Future work is mainly focused on model modification. The current system assumes all constraints are valid. If instead a new algorithm is developed for constraint propagation, the modification to a feature or part will be propagated to other related features or parts for automatic validation and revision.

7. ACKNOWLEDGEMENTS

The work described in this paper was supported in part by a grant from the Research Grants Council of the Hong Kong SAR, China, and in part by a grant from Canadian Natural Sciences and Engineering Research Council (NSERC), Canada.

REFERENCES

1. B. Shirinzadeh, CAD-based design and analysis system for reconfigurable fixtures in robotic assembly, Computing and Control Engineering Journal, Vol. 5, No. 1, pp 41-46, 1994.
2. B. Shirinzadeh, Flexible and automated workholding systems, Vol. 22, No. 2, pp 29-34, 1995.
3. B. Shirinzadeh, G. Alici, C. W. Foong, and G. Cassidy, Fabrication Process of Open Surfaces by Robotic Fibre Placement, Journal of Robotics and Computer-Integrated Manufacturing, Vol. 20, No. 1, pp 17-28, 2004.
4. Q. H. Wang, J. R. Li, and H. Q. Gong, A CAD-linked virtual assembly environment, International Journal of Production Research, Vol. 44, No. 3, 467-486, 2006.
5. J. Ye, R. I. Campbell, T. Page, and K.S. Badni, An investigation into the implementation of virtual reality technologies in support of conceptual design, Design Studies, Vol. 27, No. 1, pp 77-97, 2006.
6. P. Ji, A. C.K. Choi, and L. Li, VDAS: a virtual design and assembly system in a virtual reality environment, Assembly Automation, Vol. 22, No. 4, pp 337-342, 2002.
7. S. Gao, H. Wan, and Q. Peng, An approach to solid modeling in a semi-immersive virtual environment, Computer & Graphics, Vol. 24, No 24, pp 191-202, 2000.
8. K. Kiyokawa, Takemura, H. and Yokoya, N., Seamless Design for 3D object creation, IEEE Multimedia, Vol. 7, No.1, pp 22-33, 2000.
9. J. Butterworth, A. Davidson, S. Hench, and T.M. Olano, 3DM: a three-dimensional modeler using a head-mounted display, ACM Computer Graphics, Vol. 25, No. 2, pp 197-208, 1992.
10. J. Liang and M. Green, JDCAD: a highly interactive 3D modeling system, Computer & Graphics, Vol. 18, No. 4, pp 499-506, 1994.
11. T. Dani and R. Gadh, COVIRDS: Shape modelling in a virtual reality environment, ASME 1997 Computers in Engineering Conference, Sacramento, CA, 1997 (CDROM).
12. A. Stork and M. Maidhof, Efficient and precise solid modelling using a 3D input device, Proceedings of Fourth Symposium on Solid Modeling and Applications, Altanta, pp 181-194, 1997.
13. K. Kiyokawa, H. Takemura, Y. Katayama, H. Iwasa, and N. Yokoya, VLEGO: A simple two-handed 3D modeler in a virtual environment, Electronics and Communications in Japan, Part 3, Vol. 8, No. 11, pp 1517-1526, 1998.
14. H. Nishino, M. Fushimi, and K. Utssumiya, A virtual environment for modeling 3D objects through spatial interaction, 1999 IEEE International Conference on Systems, Man, and Cybernetics, Oita University, Japan, pp 81-86, 1999.
15. M. Bordegoni, G. Colombo and L. Formentini, Haptic technologies for the conceptual and validation phases of product design, Computers & Graphics, Vol. 30, No. 3, pp 377-390, 2006.
16. Y. Zhong, B. Shirinzadeh, G. Alici, and J. Smith, A Neural Network Methodology for Deformable Object Simulation, *IEEE Transactions on Information Technology in Biomedicine*, Vol. 10, No. 4, pp 749-762, 2006.
17. Y. Zhong, B. Shirinzadeh, G. Alici, and J. Smith, An Autowave Based Methodology for Deformable Object Simulation, *Computer-Aided Design*, Vol. 38, No. 7, pp 740-754, 2006.
18. Y. Zhong, B. Shirinzadeh, G. Alici, and J. Smith, Soft Tissue Modelling Through Autowaves for Surgery Simulation, *Medical & Biological Engineering & Computing*, Vol. 44, No. 9, pp 805-821, 2006.

19. M. Fa, T. Fernando, and P.M. Dew, Interactive constraint-based solid modelling using allowable motion, Proc. of 2nd ACM Symposium on Solid Modelling and Applications, Montreal, Canada, pp 243-252, 1993.
20. T. Fernando, N. Muttay, K. Tan, and P. Wimalaratne, Software architecutre for a constraint-based virtual environment, Proceedings of the ACM Symposium on Virtual reality software and technology, London, UK, pp 147-154, 1999.
21. T. Fernando, P.M. Dew, and M. Fa (etc.), A shared virtual workspace for constraint-based solid modelling, Virtual Environment'95: Selected papers of the Eurographics workshops in Barcelona, Spain, Springer Wien, New York, pp 185-198, 1995.
22. L. Marcelino, N. Murray, and T. Fernando, A constraint manager to support virtual maintainability, Computers & Graphics, Vol. 27, No. 1, pp 19-26, 2003.
23. R. Anatha, G. A Krammer, and R. H Crawford, Assembly modelling by geometric constraint satisfaction, Computer-Aided Design, Vol. 28, No. 9, pp 707-722, 1996.
24. J. K. Gui and M. Mantyla, Functional understanding of assembly modelling, Computer Aided Design, Vol. 26, No. 6, pp 435-451, 1994.
25. G. Brunetti, T. De Martino, B. Falcidieno, and S. Habinger, A relational model for interactive manipulation of form features based on algebraic geometry, Proceedings of 3rd Symposium on Solid Modeling and Applications, Utah, USA, pp 95-104, 1995.
26. C. F. You and C. C. Chiu, An automated assembly environment in feature-based design, The International Journal of Advanced Manufacturing Technology, Vol. 12, No. 4, pp 280-287, 1996.

ABOUT THE AUTHORS

Y. Zhong is a research fellow at Department of Mechanical Engineering, Monash University, Australia. His research interests include virtual reality, geometric modelling and CAD/CAM. Prior to joining Monash University, he worked as a postdoctoral fellow at School of Computer Science, University of Windor, ON, Canada. He also worked as a research fellow at School of Computer Engineering, Nanyang Technological University, Singapore and a lecturer at Department of Aeronautical Manufacturing, Northwestern Polytechnical University, China.

X. Yuan is an Associate Professor with the School of Computer Science, University of Windsor, ON, Canada. His research interests include advanced human/computer interaction, robotics, intelligent systems, and software engineering and testing. He received the B.Sc. degree in computer science from the University of Science and Technology of China, Hefei, in 1982, the M.S. degree in computer science from the Institute of Computing Technology, Academia Sinica, Beijing, China in 1984, and the Ph.D. degree also in computer science from University of Alberta, Edmonton, AB, Canada, in 1993. Dr. Yuan has been a reviewer of international journals and a member in program committees of international conferences, including IEEE ICRA'2000, ICRA'2002, and ICRA'2004.

W. Ma is an Associate Professor in manufacturing engineering at the City University of Hong Kong (CityU). He lectures in the areas of geometric modeling, CAD/CAM and rapid prototyping. His present research interests include computer aided geometric design, CAD/CAM, virtual design and manufacturing, rapid prototyping and reverse engineering. Prior to joining CityU, he worked as a research fellow at Materialise N.V., a rapid prototyping firm in Belgium. He also worked in the Department of Mechanical Engineering of Katholieke Universiteit Leuven (K.U.Leuven), Belgium, and the Department of Space Vehicle Engineering of East China Institute of Technology (ECIT), Nanjing, China. He obtained a BSc in 1982 and an MSc in 1985 from ECIT, an MEng in 1989 and a Ph.D. in 1994 from K.U.Leuven.

B. Shirinzadeh received engineering qualifications: BE (Mechanical), BE (Aerospace), MSE (Mechanical), and MSE (Aerospace) from the University of Michigan, and PhD in Mechanical Engineering from University of Western Australia (UWA). He has held various positions in academia and industry including research fellow at the University of Western Australia, senior research scientist at Commonwealth Scientific Industrial Research Organization (CSIRO), Australia. Dr Bijan Shirinzadeh is currently an Associate Professor, and the Director of Robotics & Mechatronics Research Laboratory (RMRL) which he established in 1994, in the Department of Mechanical Engineering at Monash University, Australia. His current research interests include modelling and planning/simulation in virtual reality, haptics, medical robotics, laser-based measurements and sensory-based control, micro-nano manipulation systems, systems kinematics and dynamics, and automated fabrication and manufacturing.