International Journal of CAD/CAM Vol. 2, No. 1, pp. 29~37 (2002)

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Virtual Disassembly

Jianzhong Mo[1]*, Qiong Zhang[2] and Rajit Gadh[3]
[1]*eMedia Center, University of Wisconsin - Madison, 1500 Engineering Dr, Madison, WI 53706, USA*
[2]*State Key Lab of CAD&CG, Zhejiang University, China*
[3]*University of California - Los Angeles, Los Angeles, USA*

Abstract – *De-manufacturing* is an entire process of collecting, disassembling, reusing, refurbishing, recycling, and/or disposing products that are obsolete or un-repairable. Designing the products for inexpensive and efficient disassembly enhances the ease of de-manufacturing. *Virtual disassembly* addresses the difficulty and the methods to disassemble a product in design stage rather than really disassemble a product at the end of its life cycle. Based on the virtual disassembly analysis results, design will be improved for better assembling/disassembling. This paper presents a systematic virtual disassembly methodology such as disassembly relation modeling, path/sequence automatic generation and evaluation. This paper also presents a new virtual disassembly interface paradigm via virtual reality technology for disassembly simulation in virtual environment.

***Keywords*: Virtual Reality, Virtual Disassembly**

1. Introduction

Currently, products are acquired with no clear plan of disposal of the product. Often, consumers are left with no other choice at the end of a product's life but to dispose it off by throwing it in to the trash. The de-manufacturing addresses the question of systematically designing for easier product dismantling and disposal. The two stages of de-manufacturing are (1) disassembling the assembly into its individual components (requiring disassembly analysis), and (2) recycling the individual materials that constitute each component (requiring component material recycling assessment). We developed a virtual-reality-based software tool - *Motive3D* - that supports collaborative de-manufacturing (disassembly, service, recycling and disposal) between manufacturer/de-manufacturer/disposer and designer. This paper will address the methodology of *virtual disassembly*, a core part of Motive3D. *Virtual disassembly* addresses the difficulty and the methods to disassemble a product in design stage rather than really disassemble a product at the end of its life cycle. Our motivation of research on *virtual disassembly* is to complement traditional tools to bring de-manufacturing trade-off analysis into the design process, allowing engineers to adopt a "predict and prevent" approach.

The rest of this paper is organized as following: Section 2 reviews the related research work; Section 3 presents virtual disassembly software environment - Motive3D; Section 4 presents virtual disassembly hardware environment; Section 5 is a conclusion.

*Corresponding author:
E-mail : jzmo@ieee.org

2. Related Research

The disassembly evaluation addresses the problem of estimating the cost, time, or design effectiveness of disassembly plans. This evaluation can be served to determine the product design for disassembly and demanufacturing. Boothroyd and Alting [1], Jovane *et al.* [2], Gupta and McLean [3], Penev and Ron [4] have reviewed Design For Assembly (DFA) methods and discussed the research trends in Design For Disassembly (DFD). Subramani and Dewhurst [5] introduced time standard charts to make disassembly evaluation. Hrinyak *et al.* [6] examined the existing disassembly software. Bras and Emblemsvag [7] evaluated the cost incurrence of different design based on Activity-Based-Costing (ABC). Kroll [8] developed a rating method based on the difficulty index of every disassembly task. Suga *et al.* [9] introduced energy and entropy for disassembly. Meanwhile, several applications specific to recycling/maintenance approaches have been developed. Kirby and Wadehra [10] suggested the Design For Environment (DFE) factor should be considered into design. Johnson and Wang [11] evaluated the recyclability and material recovery of a product. Zussamnn *et al.* [12], Ishii [13], Geiger and Zussmann [14] addressed end-of-life approach for DFD/demanufacturing. Mo *et al.* [15] investigated DFA-oriented assembly relation modeling.

Virtual Reality (VR) is a new technology that creates a real-time visual/audio/haptic experience with computer systems including hardware and software. It provides a potential way for disassembly simulation. Siddique and

Rosen [16] investigated disassembling in a virtual environment. Jayaram *et al.* [17] developed a prototype system called Virtual Assembly Design Environment (VADE). Gupta *et al.* [18] developed system called Virtual Environment for Design for Assembly (VEDA) for 2D models. Srinivasan *et al.* [19] investigated selective disassembly based on Conceptual Virtual Design System (COVIRDS). COVIRDS is a VR system coupled with 3D hand tracking, voice command, and stereoscopic visualization which provides a high fidelity visualization and easy-to-use interface, presented by Chu *et al.* [20]. Mo *et al.* [21] investigated Internet-based virtual assembly /disassembly for e-manufacturing.

Automatic path/sequence generation and manual path/ sequence generation with VR device are complementary to each other. However current researches rarely address both problems together. This paper presents a methodology to fill in this gap by virtual disassembly based on 3D CAD models, integrating VR device for disassembly simulation, automatic generation of path and sequence, visualizing disassembly sequence and path, and disassembly cost evaluation.

3. Virtual Disassembly Software Environment - Motive3D

The three steps to achieving the goals of disassembly-oriented design are (1) investigate algorithms of assembly /disassembly path/sequence generation and evaluation, and implement the software tool of Virtual Disassembly Analyzer (VDA); (2) investigate effective virtual environment to simulate disassembly process using virtual reality technology (VE); (3) investigate fast approaches to deploy assembly/disassembly information on to the network and implement the software tool of Data Deployment Tool (DDT); (4) Design data representation, investigate interactive 3D visualization, and implement a 3D visualizer, Data Poster, and Data Collector. Motive3D is developed according to above three steps. Step (1), (2) and (3) are implemented as Motive3D server. Step (4) is implemented as Motive3D client. This paper will focus on Motive3D server, esp. VDA and VE. DDT and Motive3D client will be discussed very briefly. The architecture of Motive3D sever is shown in Fig. 1. Its Virtual Environment (VE) will be discussed in Section 4.

In this section, we will focus on VDA. A very brief introduction of DDT and Motive3D Client will also be presented at the end of this section for system integrity.

3.1. Virtual Disassembly Analyzer (VDA)

VDA takes CAD models as inputs. VDA will generate assembly relation information based on the CAD models. Then VDA will generate disassembly sequence and path, optimize them, and/or users can edit them. VDA will also complete disassembly evaluation.

3.1.1. Product Preprocess

Motive3D accepts CAD models in B-rep, that it, a component can be represented by BOOD, LUMP, SHELL, FACE, LOOP, COEDGE, EDGE, and VERTEX as its topologic data structure, and SURFACE, CURVE, APOINT *et al.* as its geometric definitions. The simplified B-rep of a component is shown in Fig. 2 (ACIS^{TM} type).

What the module of Product Preprocess will do is rebuilding assembly relation, calculating clearance

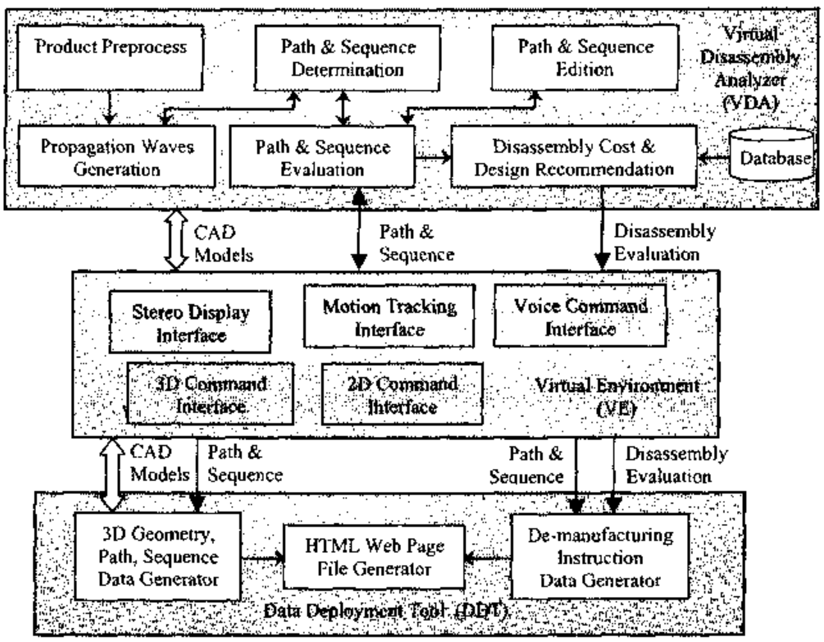

Fig. 1. Motive3D server: VDA, VE and DDT.

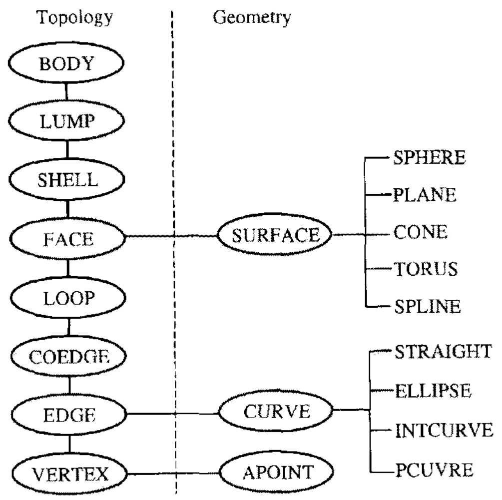

Fig. 2. Component's CAD model in B-rep.

between components and performing interference checking between components. All of those tasks are finished based on B-rep CAD models.

(1) Assembly relation establishment

There is one assembly relation, contacting constraints, should be established for automatic sequence generation. Contacting constraints can be classified into two categories: mating constraints and fitting constraints. Based on contacting constraints, the disassembly directionality of a component can be calculated. The paper represents the product to be disassembled with symbol *A*.

Mating constraint*:** $\forall$ PLANE type geometry sf_1 of $f_1 \in \text{FACE} \subset b_1 \in \text{BODY}$, and $\forall$ PLANE type geometry sf_2 of $f_2 \in \text{FACE} \subset b_2 \in \text{BODY}$, nv_1 is the normal vector of sf_1, nv_2 is the normal vector of sf_2. If nv_1 parallel to $-nv_2$, $f_1 \cap f_2 \neq \phi$, then b_1 and b_2 has mating constraints ***mating(f_1, f_2). $\cap$ represents intersection. See Fig. 3(a).

Fitting constraint*:** $\forall$ CONE type geometry sf_1 of $f_1 \in \text{FACE} \subset b_1 \in \text{BODY}$, and $\forall$ CONE type geometry sf_2 of $f_2 \in \text{FACE} \subset b_2 \in \text{BODY}$, the centerline of sf_1 is $c_1 \in$ STRAIGHT, the centerline of sf_2 is $c_2 \in$ STRAIGHT. If c_1 parallel to c_2, $f_1 \cap f_2 \neq \phi$, then b_1 and b_2 has fitting constraints ***fitting(f_1, f_2). $\cap$ represents intersection. See Fig. 3(b).

With respect to other types of surfaces, special checking method should be used to establish contacting constraints. For an example, the mating definition of two B-spline surfaces will be: $\forall$ SPLINE type geometry $sf_1(u, v)$ of $f_1 \in \text{FACE} \subset b_1 \in \text{BODY}$, and $\forall$ SPLINE type geometry $sf_2(u, v)$ of $f_2 \in \text{FACE} \subset b_2 \in \text{BODY}$, $nv_{1(ui, vj)}$ is the normal vector of sf_1 at knot (u_i, v_j), $nv_{2(ui, vj)}$ is the normal vector of sf_2 at knot (u_i, v_j), $i \in [0, m]$, $j \in [0, n]$, m, n are the dimension of rectangular arrays of control points of sf_1 and sf_2. If for every $i \in [0, m]$, $j \in [0, n]$, $nv_{1(ui, vj)}$ parallel to $-nv_{2(ui, vj)}$, and $f_1 \cap f_2 \neq \phi$, then b_1 and b_2 has mating constraints ***mating***(f_1, f_2). $\cap$ represents intersection.

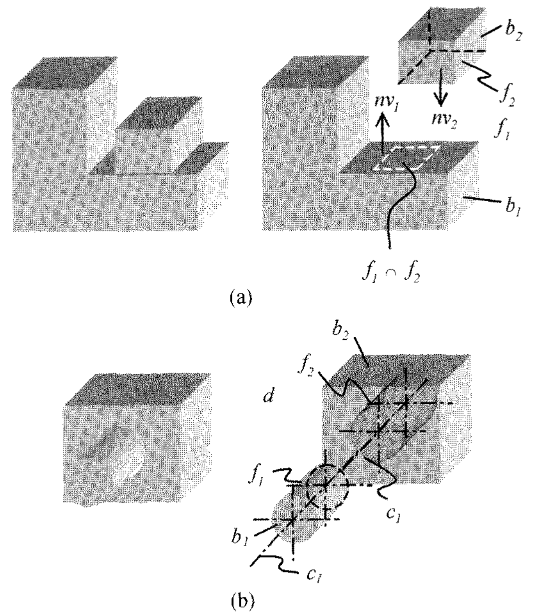

Fig. 3. Contacting constraints (a) mating (b) fitting.

(2) Disassembly directionality

Two other major tasks of Product Preprocess of VDA are disassembly directionality determination and interference checking among mating parts. The assembly relation and contacting constraints should be established for automatic sequence generation. Based on contacting constraints, the disassembly directionality of a component can be calculated.

Disassembly Directionality (*DD*) of a component *C* is a geometry entity containing direction vectors in which the component *C* can be disassembled from components $\{C_a\}$, $C_j \in \{C_a\}$ contacting with *C*. If DD_i denotes the *DD* of component C_i, and DD_{ij} denotes DD of C_i with respect to C_j, then $DD_i = DD_{i1} \cap DD_{i2} \cap \ldots \cap DD_{ik}$, *k* is the element number of $\{C_a\}$. $DD_{ij} = DD_{m(i,j)} \cap DD_{f(i,j)}$. $DD_{m(i,j)}$ is the *DD* based on mating constraints, $DD_{f(i,j)}$ is the *DD* based on fitting constraints,

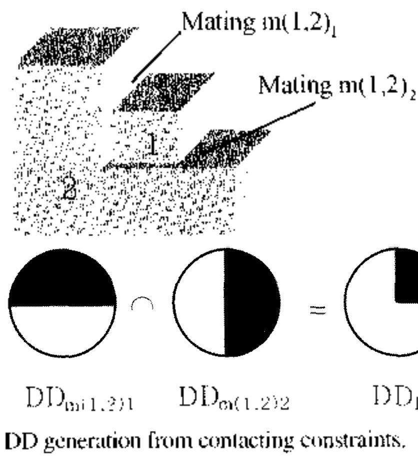

Fig. 4. DD generation from contacting constraints.

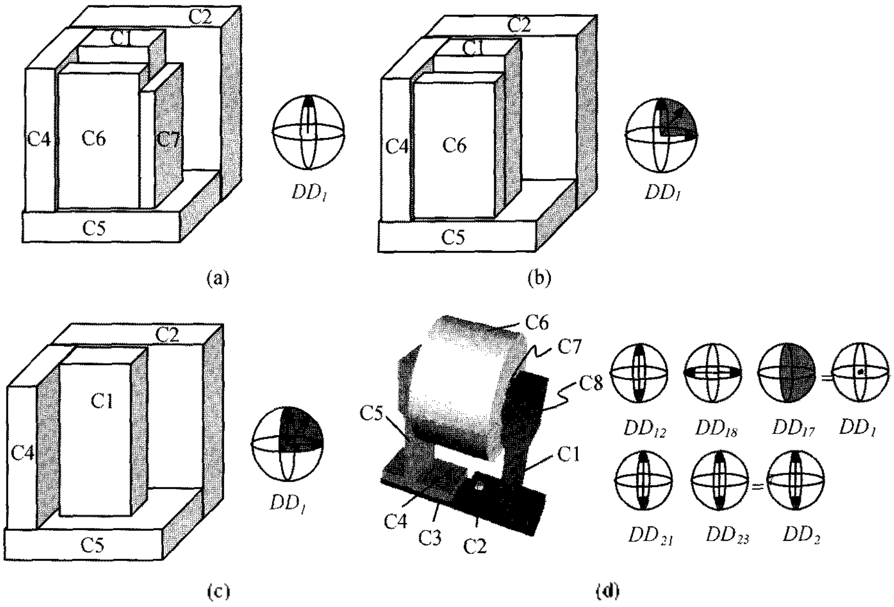

Fig. 5. Disassembly directionality.

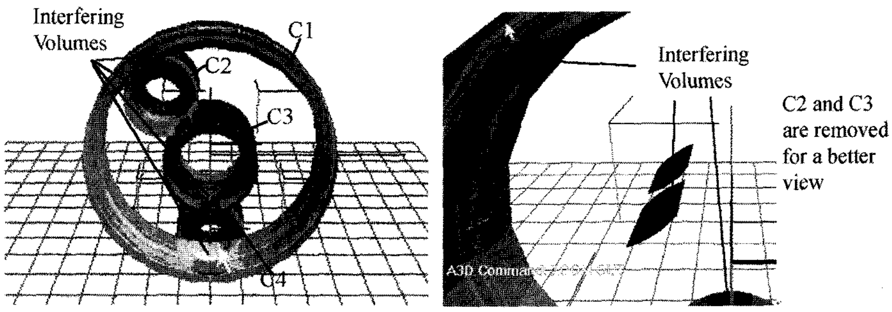

Fig. 6. Interference checking.

$\cap$ represents intersection, C_i, $C_j \in A$.

According to the definition, the generation of DD_{ij} based on contacting constraints is shown in Fig. 4.

DD_i can be any of geometric entity on a unite Gaussian sphere:

a) APOINT [a center point, DD_1 in Fig. 5(d)]
b) STRAIGHT [a radium, DD_1 in Fig. 5(a)]
c) PLANE [a fan, DD_1 in Fig. 5(b)]
d) LUMP [a sphere cone, DD_1 in Fig. 5(c)]

The disassembly direction vectors dv is determined by following equation:

$$dv=p2-p1.$$

$p1$ is the center of the unite Gaussian sphere, $p2$:

If the DD is APOINT, $p2$ is equal to $p1$, the component can not be disassembled;

If the DD is STRAIGHT, $p2$ is the other end of this straight line;

If the DD is PLANE, $p2$ is any point on the arc of this plane;

If the DD is LUMP, $p2$ is any point on the sphere surface of this LUMP.

(3) Interference checking

Interference checking is another important task executed in the module of Product Preprocess. Two steps are conducted to improve performance: (1) interference checking of parts' bounding boxes; (2) if parts' bounding boxes collide each other, do interference checking based on native geometry model (ACIS or Parasolid). Step (2) has two sub-steps: (2.1) firstly check the intersection graph between two BODYs (components); (2.2) if the intersection graph is not *null*, take intersection Boolean operation to determine the interfering volumes. An example of collision checking is shown in Fig. 6.

3.1.2. Propagation Waves and Automatically Sequence Generation

***Accessibility Graph* (*AG*)** is a directed graph representing product A's assembly relation in which nodes correspond to the components of A, and an arc, $C_i \rightarrow C_j$, indicates that component C_i is adjacent to

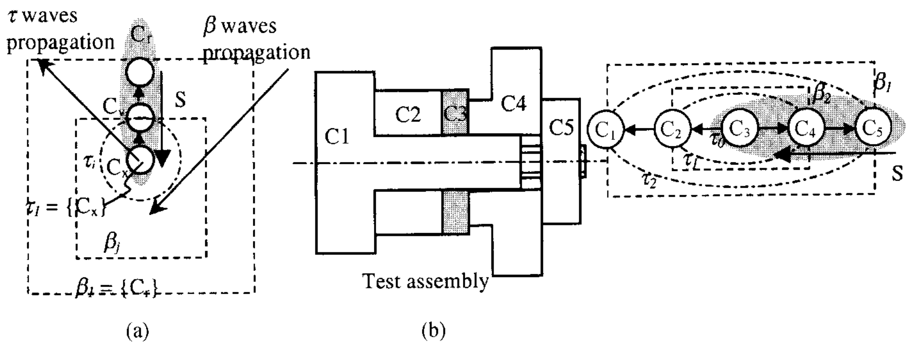

Fig. 7. Disassembly waves (a) conception (b) sequence determination.

component C_j and stores an attribute DD_{ij}, C_i, $C_j \in A$.

Removalability (R_{ij}) of a component $C_i \in A$ is a Boolean value indicating if it can be removed from its adjacent components C_j. R_{ij} is determined by DD_{ij}. If DD_{ij} is not APOINT, R_{ij}=TRUE (removable), else R_{ij}= FALSE (un-removable), C_i, $C_j \in A$.

Removalability Graph **(*RG*)** is a directed graph where nodes correspond to the components of A, and an arc, $C_i \rightarrow C_j$, indicates that R_{ij}=TRUE, C_i, $C_j \in A$.

Path **(*P*)** is an ordered list of disassembly states: P= $<s_1, s_2, \ldots, s_n>$, s_i ! s_j, i, $j<=n$, where s_i is the disassembly state of component $C_i \in A$ and n is the number of states. Disassembly state $s=[p, q]$, p is the position element of s, and q is the orientation element of s represented in quaternion. In automatic path generation, one path contains only two states and only translation movement is used for each state, that is, q remains same for those two path states. In VE, paths are generated through hand motions: translation, rotation, twist, screw motion, and free motion.

Sequence **(*S*)** is a list of paths: $S=\{P_1, P_2, \ldots, P_i, \ldots, P_j, \ldots, P_n\}$, where P_i is the disassembly path of $C_i \in A$, P_j is the disassembly path of $C_j \in A$, $i \neq j$, $i, j<=n$, n is the number of paths, and P_i and P_{i+1} have the relation of $P_i \propto P_{i+1}$ or $P_i \sim P_{i+1}$, where $\propto$ represents the relation of precedence, and $\sim$ represents the relation of indifference. If $n<m$, m is the component number of A, then S is called *selective sequence*, otherwise S is called *complete sequence*. If S has at least one relation of indifference ($\sim$), then S is called *parallel sequence*; otherwise S is called *sequential sequence*. Disassembly directionality, accessibility graph, removalability, and removalability graph are the information on which propagation wave will be generated.

A ***disassembly wave*** is defined to topologically arrange components in A to denote the disassembly order. There are two types of disassembly waves:

(1) τ *waves* propagating outwards from the component set C^s

(2) *b waves* propagating inwards from the boundary of A.

Based on the intersection event between and β *waves*, a disassembly sequence with minimal components removal to remove selected component set C^s is determined. Let τ_i denotes the *i*th wavefront of a τ *waves*, β_j *j*th wavefront of a β *waves*. τ_i and β_j are both represented by component set. $\{C_r\}$ denotes removable component set of A, and P_k denotes the shortest disassembly path for C_k. Fig. 7(a) shows the concept of disassembly waves, where C^s $=\{C_x\}$. In this figure, an intersections event occurs at C_y, where τ_i intersect β_j ($j \leq i$), the arcs are corresponding to ***RG***. This intersection determines a sequence $S_x=\{P_x, P_y, P_r\}$, $P_x \propto P_y \propto P_r$ with locally minimum component removals for $C^s=\{C_x\}$. Fig. 7(b) is an example. For details about propagation waves, refer to our previous paper [19].

3.1.3. Path and Sequence Evaluation

Path evaluation involves the interference checking of one part with others along its disassembly path. There are two steps: (1) extended bounding boxes interference checking. BB_{ei} is the extended bounding box of component C_i disassembled along path $P=<s_1, s_2, \ldots, s_n>$, n is the number of path states. BB_{ei} is defined by two vectors: vector $(x_{max}, y_{max}, z_{max})_i$ representing BB_{ei}'s maximum coordinates, and vector $(x_{min}, y_{min}, z_{min})_i$ representing B_{ei}'s minimum coordinates. BB_{sij} is the bounding box of component C_i at path state s_j. BB_{sij} is also defined by two vectors: $(x_{max}, y_{max}, z_{max})_{sj}$ and $(x_{min}, y_{min}, z_{min})_{sj}$. Then $(x_{max}, y_{max}, z_{max})_i=\max[(x_{max}, y_{max}, z_{max})_{s1}, (x_{max}, y_{max}, z_{max})_{s2}, \ldots (x_{max}, y_{max}, z_{max})_{sn}]$, and $(x_{min}, y_{min}, z_{min})_i$ $=\min[(x_{min}, y_{min}, z_{min})_{s1}, (x_{min}, y_{min}, z_{min})_{s2}, \ldots (x_{min}, y_{min}, z_{min})_{sn}]$. The interference checking result Rp for path P of C_i is calculated by $R_p=BB_{ei} \cap^{k=1,2,\ldots,m,k \neq i} BB_k$, BB_k is the bounding box of component C_k, m is the component number of A. If $R_p=\phi$, the path P is valid. Otherwise do step (2) sweep volume interference checking. This step involves low level interference checking based on native geometry in ACIS or Parasolid. In ACIS, interference checking can be accomplished by collision checking of BODYs of component models. Let B_{si} is the BODY of C_i sweeping along $P=<s_1, s_2, \ldots, s_n>$, n is the number of path states. B_k is the BODY of C_k. Then interference checking result Rp for path P of C_i can be re-calculated by $R_p=B_{si} \cap^{k=1,2,\ldots,m,k \neq i} B_k$,

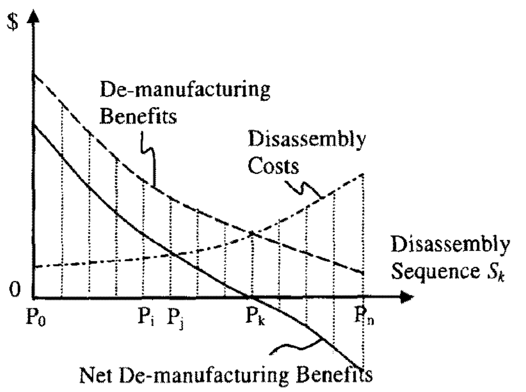

Fig. 8. Disassembly costs *vs.* de-manufacturing benefits.

m is the component number of A. If $R_p=\phi$, the path P is valid. Otherwise P is not valid.

Disassembly Cost (*DC*) of one component can be determined based on attributes of the component (such as its weight, volume, position, orientation, *et al*), its disassembling path (such as distance, rotated angles, state numbers) and standard variables (such as labor rate, fixture cost, standard time for accomplishing a particular task). *De-manufacturing Benefit* (*DB*) of one component can be determined based on its reuse value (shape reuse and part reuse) and recycling value (material reuse). *The Net De-manufacturing Benefit* (*NDB*) of the component is the difference of its de-manufacturing benefit and disassembly cost. Let $S=\{P_0, P_1, ..., P_n\}$ denotes the disassembly sequence of an assembly, DC_i denotes the disassembly cost of component $C_i \in A$, DB_i denotes the de-manufacturing benefit of C_i, and NDB_i denotes the net de-manufacturing benefit of C_i, then $NDB_i = DB_i - DC_i$. De-manufacturing requires that maximum *NDB* will be gained at the earliest stage of disassembling process, that is, $NDB_i \geq NDB_j$, $i<j<n$. This requirement will be satisfied if $DB_i > DB_j$ and $DC_i \leq DC_j$, $i<j<n$, as shown in Fig. 8. If $DB_i<DB_j$ or $DC_i>DC_j$, a design recommendation will be thrown out to CAD. Fig. 8 also illustrates that total *NDB* will reach the maximum at the disassembly stage of P_k for C_k. After that, total *NDB* turns into minus.

The first process of sequence evaluation is finding out a sequence from a sequence set which has maximum NDB. The second process is the evaluation of paths of this sequence. If there is no applicable sequence available, a design recommendation is thrown out. If there is invalid path in a path, a design recommendation is thrown out. Currently, disassembly tools are not considered as a constraint. It's our future research work.

3.2. Data Deployment Tool(DDT)

DDT takes disassembly sequence, path and evaluation as inputs. Then DDT organizes the sequence and path data into a special file together with meshed CAD geometric information. DDT will generate an HTML page file, which will launch 3D Visualizer and Data Poster when the page is viewed by users via web browser. All the output of DDT will be saved at HTTP server side.

3.3. 3D Visualizer, Data Poster, and Data Collector

They are the client side applications. 3D Visualzier will visually display assembly/disassembly path and sequence interactively. Data Poster will post disassembly instructions and de-manufacturing cost/benefits. Data Collector is used to collect the feedback from the users.

3.4. Technology Used in Implementation

VDA and DDT make up Motive3D's server side. VDA and DDT are supported by Virtual Environment (VE). VDA, DDT and VE together are also called ***A3D***, (Assembly/disassembly in 3D). 3D Visualizer, Data Poster, and Data Collector make up Motive3D's client side. Server side is implemented in C/C++, ACIS/Parasolid, OpenGL, and WorldToolKit. CAD models in ACIS/Parasolid format are triangulated and displayed within WroldToolKit rendering environment. IGES to ACIS translator and UG to Parasolid translator are also embedded into *A3D*. Product assembly structure is translated into *scene graph* for 3D operations. Multi-sensory input device are coupled with special nodes of scene graph. Client side is implemented in Java, Java3D, and HTML. In an A3D system: (1) user inputs C^S via a menu-interface, (2) *Disassemblability* and *Removal Influence* are determined from ***AG***, and (3) sequences S are determined using the *WP* abstraction. The generated *SD* sequences are then simulated in *A3D* system. Fig. 9 shows an aero engine assembly without 3D menus and toolbars.

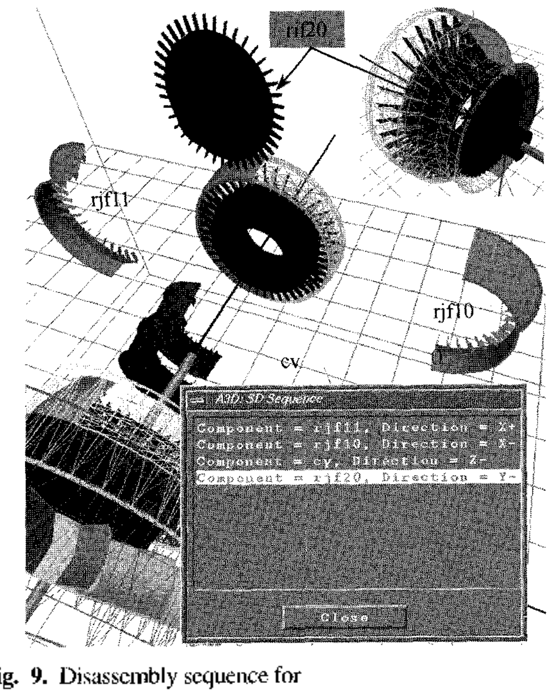

Fig. 9. Disassembly sequence for

A3D also supports Virtual-Reality (VR) devices, such as devices for 3D Hand tracking, voice command and stereoscopic visual display. VR enabled A3D provides high fidelity visualization and an easy-to-use interface for sequence/path generation and visualization. The following section will discuss this issue.

4. Virtual Disassembly Environment

4.1. Computing and Display Hardware

eMedia Center Visualization Laboratory is a unique, theater-style computing center. Three HP UNIX J5600 computers with dual PA-RISC 8600 CPUs and fx10 graphics card per display makes up the 3D rendering system. Those three computers connected by gigabit switch to support Distributed Single Logical Screen (SLS/D), which enables users to configure their workstations to utilize multiple add-on graphics cards, a one large, virtual desktop. The rendering system is running on hp-ux 11.0 and supports OpenGL application transparently. The integrated system is called HP Visualize Center.

The display system is called grv-120 provided by Panoram Technologies. The GVR-120 is a curved screen immersive, project visualization system for Group VR applications. It fits into a 16' 6" X 16' 6" X 10' space. The bright, high-resolution image can be displayed with lights on for work group sessions, information sharing and note taking. The system supports a 6 channels surround sound audio system. Integrator version 3 is used for active matrix Stereographic glasses and emitters control. Three projectors display 3D models from Visualize Center on to the curved screen. The layout of Visualization Laboratory is shown in Fig. 10.

4.2. Virtual Reality Devices

Virtual Environment is a platform that provides 2D, 3D and voice interface. The input interfaces are for 3D mouse, tracking device, grabbing device, and voice device. The outputs are 3D sound, stereo display. As an example of using Virtual Reality in Virtual Disassembly, this section will present the method to record paths by hand gestures.

To track the position and orientation of hand gesture $\{HG: (x, y, z, \theta, \phi, \varphi)\}$, three Ascension FOB 6 D.O.F. motion trackers and a 5th Data glove are used (left hand), shown in Fig. 11(a). To select the target component for path recording, an Anir Mouse is used (right hand), as shown in Fig. 11(b). Both hands are needed to record paths. Left hand wears 5th Data glove, and its gesture is coupled to the selected target. So the target will move as the left hand moves. When left hand grasps, it is ready to record *HG*. When left hand grasps, it means standby, which enables large-scale motion. If left hand translates, element of (x, y, z) of *HG* will be changed, and if left hand rotates, element of (θ, ϕ, φ) of *HG* will

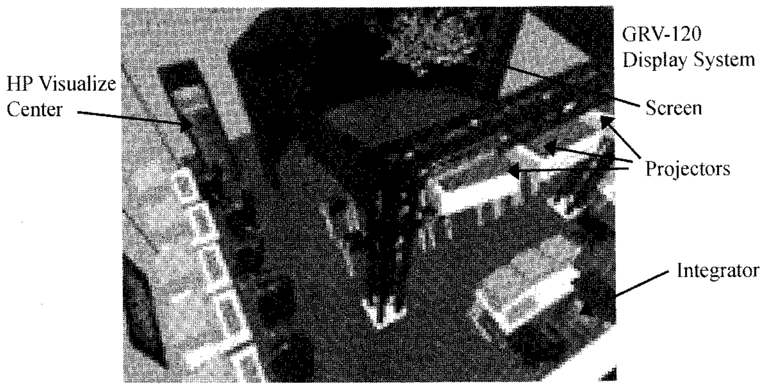

Fig. 10. Rendering and display system.

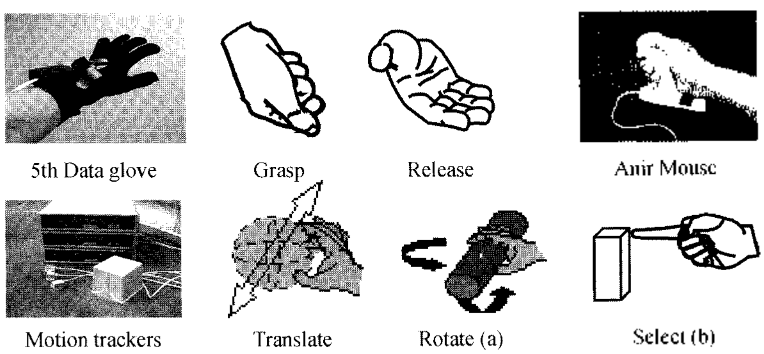

Fig. 11. Gestures and locator (a) left hand (b) right hand.

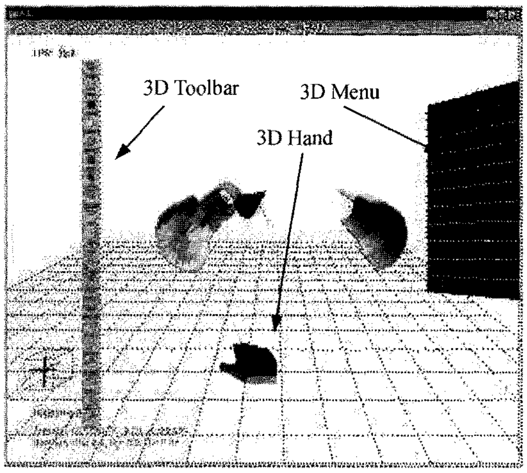

Fig. 12. Disassembling an engine (stereo interface).

be changed, as shown in Fig. 11(a). One recording method is manually clicking a mouse button by right hand to record current *HG*. Another way is recording gestures by definite frame, such recording gestures every one frame, two frames, *et al.*

To prevent generating interfering paths, real time collision detection is introduced, which will prevent moving the target further at particular direction. Usually real time collision detection is time-consuming. An alternative way is checking interference after recording a path.

Fig. 12 shows Motive3D's 3D interface in Virtual Environment. Corresponding to regular window applications' 2D toolbars and 2D menus, there are 3D toolbar and 3D menu in Motive3D's Virtual Environment. These 3D toolbars and menus ease the command selection and activation in stereo mode. To visualize 3D toolbar and menu selection, as well as part selection, a virtual hand is modeled. The virtual hand is attached to 3D mouse and moves as user's hand moves.

VR enabled A3D provides high fidelity visualization and an easy-to-use interface for sequence/path generation, editing and visualization. In *VR* environment of stereo, voice and pinch gloves, users can manipulate the parts, improve ergonomics and carry out disassembling the products virtually. It allows engineers, earlier in the development cycle, to visualize their full-system design ideas on the computer, with realistic 3D models, and make these models disassembled, just as they would in the real world.

5. Conclusion

Virtual disassembly provides a good solution to simulate the disassembly process bottlenecks and evaluate operational sequences, thus to improve product design for easy disassembly. This paper presents a systematic methodology for disassembly relation modeling, path/sequence automatic generation and evaluation independent to any commercial CAD systems. This paper also presents a new interface paradigm with multi-sensory input/output providing an effective, realistic and efficient interface interaction that enables disassembly simulation in 3D, that is, generate/edit/simulate disassembly path/sequence manually within a virtual environment. The future work will integrate disassembly tools into system as constraints for path/sequence generation/evaluation, and design feedback.

References

[1] G. Boothroyd and L. Alting (1992), "Design for assembly and disassembly", *CIRP Annals*, **42**(2), 625-636.

[2] F. Jovane, L. Alting, A. Armillotta, W. Eversheim, K. Feldmann, G. Seliger and N. Roth (1993), "A key issue in product life cycle: Disassembly", *CIRP Annals*, **42**(2), 651-658.

[3] S. M. Gupta, and C. R. McLean (1996), "Disassembly of products", *Computers and Industrial Engineering*, **31**(1-2), 225-228.

[4] K. D. Penev and A. J. Ron (1996), "Determination of a disassembly strategy", *International Journal of Production Research*, **34**(2), 495-506.

[5] A. Subramani and P. Dewhurst (1994), "Repair time estimation for early stages of product development", *Journal of Design and Manufacturing*, **4**(2), 129-137.

[6] M. J. Hrinyak, B. Bras, and W. F. Hoffmann (1996), "Enhancing design for disassembly: a benchmark of DFD software tools", *Proceedings of the ASME DET and CIE Conference*, **96**-DETC/DFM-1271.

[7] B. Bras and J. Emblemsvag (1995), "The use of activity based costing, uncertainty and disassembly action charts in demanufacture cost assemblies", *ASME Design Engineering Technical Conference*, DE-vol 82, **1**, 285-292.

[8] E. Kroll (1996), "Development of a disassembly evaluation tool", *Proceedings of the ASME DET and CIE Conference*, **96**-DETC/DTM-1509.

[9] T. Suga, K. Saneshige, and J. Fujimoto (1996), "Quantitative disassembly evaluation", *IEEE International Symposium on Electronics and the Environment*, 19-24.

[10] J. R. Kirby and I. Wadehra (1993), "Designing business machines for disassembly and recycling", *1993 IEEE International Symposium on Electronics and the Environment*, 32-36.

[11] M. R. Johnson and M. H. Wang (1995), "Planning product disassembly for material recovery", *International Journal of Production Research*, **33**(11), 3119-3142.

[12] E. Zussmann, A. Kriwet, and G. Seliger (1994), "Disassembly Oriented Assessment Methodology to Support Design for Recycling". *CIRP Annals*, **43**(1), 9-14.

[13] K. Ishii (1995), "Life-Cycle engineering design", *ASME Journal of Mechanical Design*, **117**, 42-47.

[14] D. Geiger and E. Zussmann (1996), "Probablistic reactive disassembly planning", *CIRP Annals*, **45**(1), 49-52.

[15] J. Mo, J. Cai, Z. Zhang, and Z. Lu (2000), "DFA-oriented assembly relation modeling", *Int. J. Computer Integrated Manufacturing*, **12**(3), 238-250.

[16] Z. Siddique and D. W. Rosen (1997), "Automated Product Disassembly Reasoning", *Computer-Aided Design*, **29**(12), 847-860.

[17] S. Jayaram, I. C. Connacher, and W. Lyons (1997),

"Virtual assembly using virtual reality techniques", *Computer Aided Design*, **29**(8), 575-584.

[18] S. M. Gupta and C. R. McLean (1996), "Disassembly of products", *Computers and Industrial Engineering*, **31**(1-2), 225-228.

[19] H. Srinivasan, R. Figueroa, and R. Gadh (1999), "Selective disassembly for virtual prototyping as applied to demanufacturing", *Robotics and Computer Integrated Manufacturing*, **15**, 231-245.

[20] C.-C. Chu, T. H. Dani, and R. Gadh (1998), "Multisensory Interface for a Virtual Reality Based CAD System", *Computer-Aided Design*, **29**(10), 709-725.

[21] J. Mo, Q. Zhang, and R. Gadh, "Internet-based collaborative virtual assembly/disassembly for e-manufacturing", CAD/Graphics'**2001**, pp. 493-499, August 22-24, 2001, Kunming, China.

Jianzhong Mo has been a Research Associate at eMedia Center of University of Wisconsin-Madison, USA. His research interests include CAD/CAM, Virtual Reality in Engineering, and Data-Centered Visualization. He received his PhD from Shanghai Jiao Tong University in 1998. He is a member of ASME and IEEE.

Rajit Gadh is a Professor at the Henry Samueli School of Engineering and Applied Science at University of California-Los Angeles, USA. He received his PhD from Carnegie Mellon University in 1991. His research interests include Web-based Product Design, CAD, Virtual Reality for Design, Mobile Field Service, and Wireless Internet for Collaboration.

Qiong Zhang received his BS, MS and PhD all in Computer Science from Zhejiang University, China, in 1994, 1996 and 1999 respectively. He has been a Research Associate at University of Wisconsin-Madison, USA and a Researcher at Philips Research East Asia.