A Training System for Detecting Novice's Erroneous Operation in Repairing Virtual Machines

Norihiro Abe*, Tomohiro Amano*, Kazuaki Tanaka*,
Jiang Y. Zheng*, Shoujie He**, Hirokazu Taki***
*Dept of Mechanical Systems Engineering,
**Dept of Computer Science and Electronics,
Faculty of Computer Science and Systems Engineering,
Kyushu Institute of Technology
Kawazu 680-4, Iizuka, Fukuoka 820-8502 JAPAN
***Mitsubishi Electric Inc.
Itami Osaka Japan
abe@mse.kyutech.ac.jp

Abstract

This paper shows a new version of the training system for encouraging novices in repairing mechanical objects broken. The old one forced them to follow a single sequence of operations needed to assemble / disassemble an object from / into a pieces of primitive parts. It is well known that there are many number of procedures assembling / disassembling given object which are described in AND/OR graphs. The new system allows a novice to treat the given 3 dimensional Technical Illustration of the object at will, and examine his own idea he has about repair of it.

The system watches whether his way is correct or not referring to its AND/OR graph representation. He would never be given a warning unless he made an erroneous operation which leads the state of it to fatal one.

1. Introduction

A number of expert systems have been developed so far but most of them failed to know if a user correctly responds to the instruction they gave him. Beginners are especially apt to misunderstand illustration shown to them through GUI when they are required to manipulate objects like mechanical parts which they do not get used to seeing. If an erroneous parts were manipulated, the resulting state of the given mechanical object would be different from that modeled in the system and consultation between them will surely fail sooner or later. Let assume that a human expert instructs a novice and watches what he will do corresponding to his advice. Clearly the expert will be able to correct a large amount of erroneous operation done by the novice. However when the number of novices are increased, the expert is needed a fairly amount of endurance in watching their behavior and it may become quite impossible for him to find their errors and to correct them.

The original study which aims to have computer system watch man's behavior from that done by P.E.Hart [1], which exploited a vision system to have computer observe man's behavior and tried to communicate him with natural language. It was, however, proved that computer vision was not able to recognize novice's behavior. Instead of vision system, we proposed another method using computer graphics which illustrates the states of the machine before and after adding an assembly / disassembly operation to it [2].

We expected him to find his error by comparing the state of the machine in his hand with that of the modeled one. Unfortunately he often reported that his machine was correct although it was really not correct.

We have already reported the first version of our training system [3], but a detailed mechanism of the system was not described and it did not allow a novice to operate the given mechanical object for himself but forced him to follow the instruction from the system. It is well known that there are many procedures for assembling / disassembling a given object from / to a set of primitive parts. An object with complex structures is usually constructed hierarchically, the concept of subassembly should be naturally introduced. However, the concept is difficult for a novice to introduce, an expert has a privilege to decide which parts should constitute a subassembly. Consequently, a group of subassembly is given to the system by an expert through a sequence of assembling operations. The old system forced a novice to disassemble a given object into primitive parts, but we seldom need to take an object to pieces unless we are compel to do so. The purpose of the system is to help a novice take malfunctional parts out and to repair the machine trouble. Consequently, subassemblies not involving the malfunctional parts should not be detached as far as possible.

2. Errors by human and Errors in assembling / disassembling process.

There have been many researches on planning or robotics in artificial intelligence but no system succeeded in attaining a goal in the same way as human being will naturally will take. Almost methods applied in machines are different from those used by us, some of them are not easy for us to acknowledge. Emergence of ability for grasping things hierarchically is one of the most difficult task for computers including hierarchical assembly methods. For a given problem, a set of rules used by us is different from that generated and exploited with a machine, it is quite hard for us to supplement the difference with concept we have. As mentioned early, the concept of subassembly is most appropriate to be introduced by an experts.

Errors are roughly divided into two classes; a skillful driver's error and a stranger's error while driving. The former is due to our inclination that he wants to discount the amount of information processing, and the latter to ignorance that he does not know what must be noted in order to discount the amount of information processing. The objective of the system belongs to the latter case, therefor it is helpful for a novice to follow the procedure an expert gave to the system.

Operations between objects having complex structures like mechanical ones is said to be quite difficult for novices without experience to treat them. It is difficult or impossible for us to select free view points at will in the real world or a projection chart, a virtual world makes it possible to answer such an impossible wish. Even if recognition of the situation succeeds, it does not imply that we can realize the operation needed to change the situation. The approach proposed in this paper is preferable to that using real parts, because there is no risk to break them.

As the present virtual reality system does not realize physical lows such as that of gravity, friction force. Successful operation in the virtual space does not always guarantee that the real one will succeed in the real world. But if novices acquire a perfect procedure to finish their goals through our training system, we believe that they will be able to complete their goals in the real world because the physical lows will not prevent them from attaining the goals but help them achieve complex geometric relation among parts though a sense of force.

3. System Configuration

The hardware configuration is shown in Figure 3.1. Models of mechanical parts and subassemblies are described in OpenInventor [9]. To defined subassemblies, for each parts data on their location and orientation before and after an assembly operation are needed. To give them to the system, an assembly supporting system is used, which help an expert build a subassembly using a 3 dimensional mouse device, and allow him to give it a name. As the result, relative relation among parts mated is made clear and it is available to draw 3 dimensional Technical Illustration (3dTI) of the given machine, but their initial position is useless, then they must be relocated to generate the 3dTI shown in Figure 3.2. The final assembly obtained from this 3dTI is shown Figure 3.3, which is constructed from 3 subassemblies which will be assembled in sequence. For each subassembly, only a portion of 3dTI is sufficient to have a novice recognize the assembly sequence, and it is desirable to conceal the portion irrelevant to the subassembly under operation. The OpenInventor provides us with the primitive mechanism which makes it possible to control the rendering region of 3dTI.

3.1 Definition of a primitive part

Figure 3.4 shows an example of a part called a worm shaft and the corresponding scene graph. It begins at the node MyParts whose data node includes a name, part number, initial-state and goal-state, which are

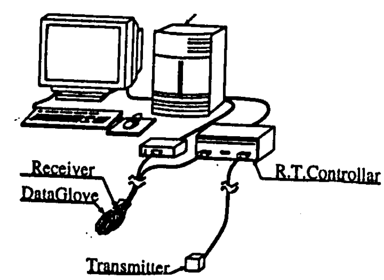


Figure 3.1 System Configuration.

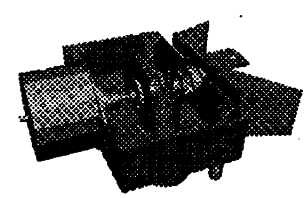


Figure 3.3 A final assembly.

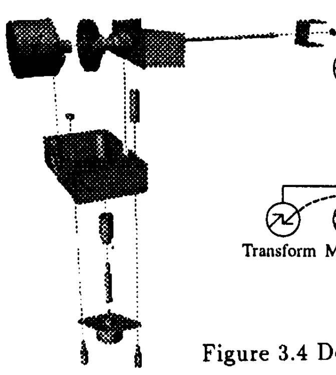


Figure 3.2 A three dimensional TI (3dTI).

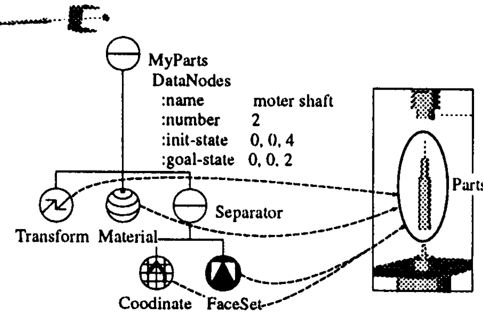


Figure 3.4 Definition of a primitive part MyParts.

exchangeable according to the kind of the current operation added to this part. In this case, as the operation added to the part is just a translation, data concerning rotation is abbreviated. When a user grasps it with a data globe and moves it along the dotted line (here after we call it an auxiliary line), the shape and posture of the globe are computed by referring to the data from magnetic sensor and angles between finger joints, the movement is given to the transform node of the object he is grasping. A user will be able to understand which geometric relation must be hold at the end of the operation through experience.

3.2 Definition of Sub-assembly

Figure 3.5 shows a subassembly MyParts0 which consists of 4 parts MP1, MP2, MP3 and MP4. Each MPi is defined as a primitive part in the same way as 3.1. At first, as there is no assembly relation among them, MyParts0 does not exist as a part. The :operationalp means that MyParts0 is not available as a part and :usedp represents that it can not be mated with others. The ass-seq means a sequence of operations, and there is only a unique assembling sequence in the old system because it forced users to follow this sequence.

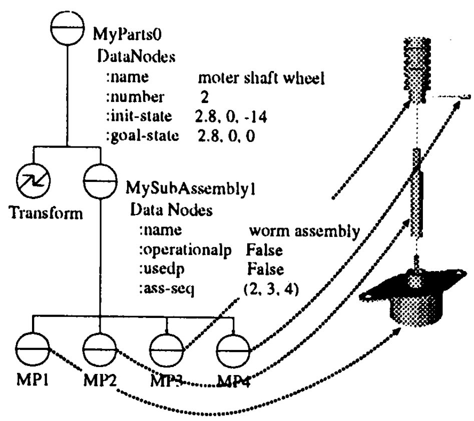


Figure 3.5 Definition of a subassembly MyPart0.

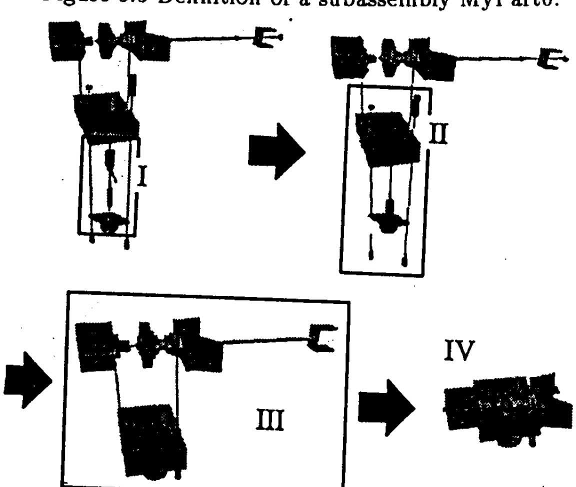


Figure 3.7 A global flow of assembly.

In the new system, it is replaced with AND/OR graph and a user is allowed to choose one of plausible paths unless he never selects a fatal one, in which case the system gives him advice and inhibits his action by neglecting the effect of his manipulation.

3.3 Modeling of Final Assembly

Figure 3.6 shows the mechanism to have a user concentrait a particular subassembly in which he is interested. In the case of our example, there are 3 subassemblies each of which is enclosed with a rectangle as shown in Figure 3.7, when he is engaged in a particular subassembly, it is desirable that any other parts and subassemblies irrelevant to it are invisible to him.

Now assume that he is interested in a subassembly MSAi. Replacing the MySubAssembly node in the above tree with the structure under MSAi makes it possible for the camera system at the left most node of tree to have the relevant portion of the 3dTI visualized. As the result, for an example, the subassembly enclosed with a frame I is visualized as shown in Figure 5.1.

4. Part Selection and Transfer

4.1 selection and grasping

When a user is told to move an object specified by our training system, he will at first grasp it with a dataglove. When the following two conditions hold, it will determine that he has grasped the object.
(1) the area of a triangle whose vertices are on 3 finger tips shown in Figure 4.1 satisfies:

$$\Delta OPQ = \frac{1}{2}|\vec{OP}||\vec{OQ}|\sin\theta = \frac{1}{2}\sqrt{|\vec{OP}|^2|\vec{OQ}|^2 - (\vec{OP}\cdot\vec{OQ})^2} < 10.0$$

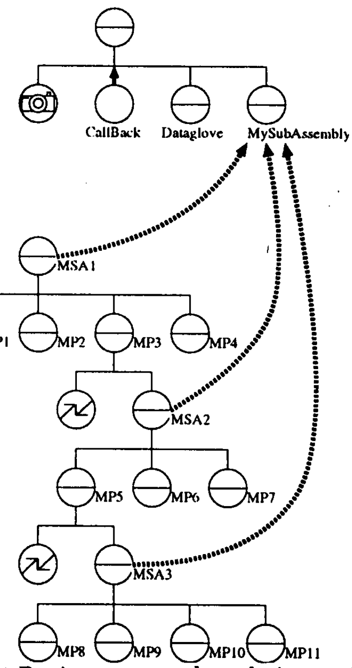


Figure 3.6 Device to control rendering region.

(2) the bounding box of the glove and that of the object have non zero intersection as shown in Figure 4.2.
When he succeeds in grasping operation, the system informs him of that by changing the color of the grasped object. Next he is permitted to move the object. In this case, there are alternative methods concerning translation, the first is to move the object along the auxiliary line attached to the object and the second is unrestrained one. In reality, the latter case is used when a sensible user is permitted to move objects freely. If an object grasped is going to move out of the line in the former case, the system will have it stopped moving. He must regrasp it hoping to continue his operation.

4.2 Starting and end points of transfer

Diagnosis usually starts from a final assembly and proceeds to the diagnosis of subassemblies or primitive parts. Consequently training plan also should progress in the same way as diagnosis, but it is a time consuming task. In new system, we are free to select any subassembly. Let assume that we have to learn how to repair the assembly in Figure 3.3, and that our diagnosis system told us to examine a wrist-rotate-motor. This fact is transmitted to our system, and at first we will be told to assemble the TI in Figure 5.1. The reason why assembling precedes disassembling is to have a user recognize parts shapes and terminal position of assembling operation. Successful completion of a given assembly will lead him to disassembling process. Terminal position of disassembling operation is not so rigorous as that of assembling one. He has only to remove a specified object from its assembly. In our system an object removed is automatically located by the system on an initial position in a given TI.

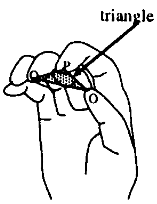


Figure 4.1 Defintion of a triangle.

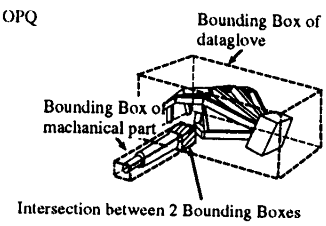


Figure 4.2 Condition of grasp.

Here note that assembling sequence does not correspond to the reverse sequence of disassembling operations. The TI shown in Figure 5.1 is such an example. The current system does not take it into account.

5. How to have subassembly treated at will.

5.1 Selection of a subassembly

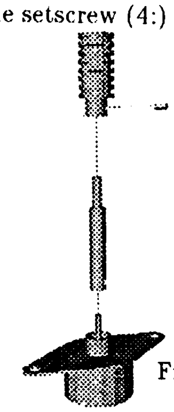
There is just one sequence of assembling / disassembling sequence in the old system, but the new system allows a user try to assemble / disassemble subassemblies whenever they are operational. Now let consider the case he has assembled a 3dTI enclosed in the frame II of Figure 3.7. At this point, he is able to either proceed to the next subassembly III or back to the previous one I. This makes it possible for him to practice the same subassembly repeatedly until he will gain confidence in the operation.

The system examines to see if the subassembly selected is operational by checking the values stored in both :operationalp and :usedp of it. If the value of :operational is T and that of :usedp is F then it has been finished and not been used, that is, it is operational. If not operational then the selection is denied.

Here the method for updating their values is to be explained. When a subassembly is used as part for constructing a new subassembly, then the value of :operationalp is set to F and that of :usedp to T, and when the new one is completed, the :operationalp and :usedp are set to T and F respectively. On the other way, when a subassembly is detached from the parent one, it becomes available as part, then the :operationalp and :usedp are set to T and F respectively.

5.2 Procedures in AND/OR graph

Now consider precisely the procedure for the subassembly in Figure 5.1. The final state of the subassembly is shown in Figure 5.2. There are several assembling ways to transform 4 parts into this state. As the rotating movement of the motor shaft (1:) must be transmitted to plastic worm, each of the fits between the motor shaft (1:) and the worm shaft (2:) and that between the plastic worm (3:) and the worm shaft (2:) must be heavy force fit. Further, the mounting holes of the worm shaft (2:) and the plastic worm (3:) must be aligned to have the setscrew (4:) inserted into the holes. The best



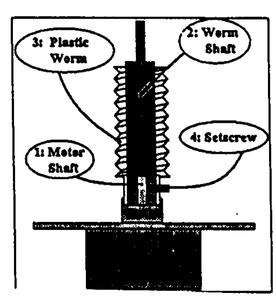


Figure 5.2 The geometric relation among parts.

Figure 5.1 An example of a subassembly.

way to get this state is to use the setscrew as a service bolt so that the alignment between the holes is not lost. This means the following sequence is required: at first two holes are aligned, then the setscrew is inserted into the hole so that it does not obstruct the insertion of the motor shaft. Assuming that the effect of any physical lows is disregarded, and that a number of times of operations added to each part is to be just one, it is not difficult to obtain all possible sequences of operations for the final assembly or each subassemblies. As no physical low is realized in our virtual space, the assumption will hold. Note here that no user is allowed to disassemble any subassembly until it is completely separated from the parent subassembly. In the same way, before a subassembly is completely assembled, it must not be attached to other subassemblies. For examples, no one is permitted to detach the setscrew from the subassembly shown in Figure 5.3. The operation is not permitted until the subassembly is disassembled into the state shown in the frame II of Figure 8.

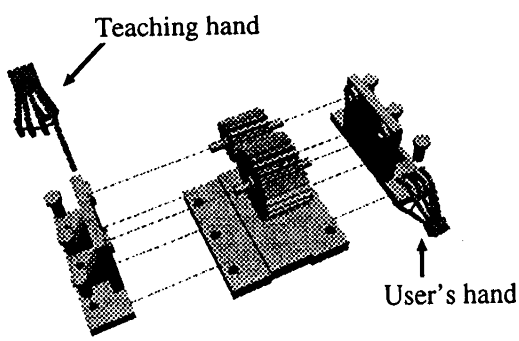


Figure 6.1 Detection of error in restricted mode.

6. Detection of Erroneous Operation

6.1 Training using 3DTI

At first, we intended to give a novice advice in natural language with a virtual hand pointing a right object when he made a mistake or he behaved nothing. It is, however, difficult to have utterance synchronize with rendering of the hand. Consequently, we have the virtual hand appear near the object to which a user's hand must approach, in the following cases.
(1) there is no movement in his hand.
(2) he does not grasp any thing.
(3) he seems to hesitate.
(4) he is going to grasp a wrong object.
Movement of the data globe allows the system to detect and distinguish one from others, then it will be able to generate a proper utterance for each case, but utterance is not implemented really owing to the above reason.

In the mode which have a user follow a unique sequence of operations, the virtual hand appears as shown in Figure 6.1 nevertheless he is never wrong substantially. In another mode, his action is regarded as correct one. On the other hand, if he is going to grasp a setscrew in Figure 5.1, the system never permits such an action based on the assumption that each operation is executable just once and away. Operating the setscrew at this stage means that it must be tightened and consequently the operation makes it impossible for him to insert the worm shaft or motor shaft.

6.2 Repairing malfunctional parts.

When a machine gets out of order, part which suspects to have a defect is must be examined, and the system must guide a user to the part which may be broken. In this case, there are following possible cases:
(1) the part is not wrong.
(2) the part is wrong and it was replaced with good one.
In case(1), there must be at least one wrong part else where. In case(2), when the machine is reassembled it may work but if not works, it imples that there must be at least one wrong part. If he or a diagnosis system suspect more defects in the machine, instead of reassembling it, he will want to partially reassembling it to the point from where disassembling resumes toward the doubtful part. Naturally it must provide him with such a spport. This is easy to implement because all possible path between a doubtful part and the final assembly is retrieved by tracing AND/OR graphs.

6.3 Training without 3DTI

When the system or a user confirms that he has acquired enough experience in assembling / disassembling operations using the mechanical parts shown in Figure 6.2, it allows him to use them at will whatever he want to built.

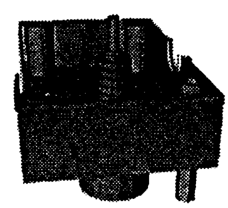


Figure 5.3 The completed subassembly.

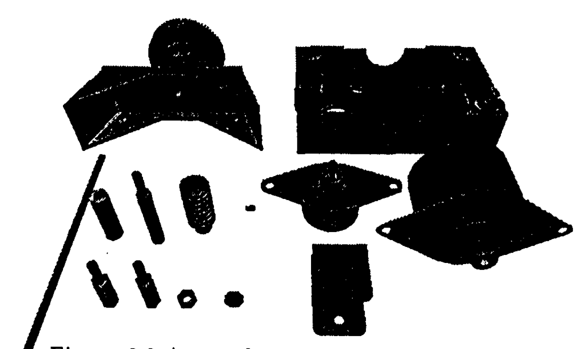


Figure 6.2 A set of mechanical parts provided.

Here note that it is not easy for a user unfamiliar to a virtual environment to grasp or touch virtual objects as well as in the real space. Mechanical assembly in a virtual environment without haptic sense is too difficult to insert a peg into a hole because the wall of the hole does not support the peg during insertion.

A virtual line called an auxiliary line used in 3DTI makes it feasible for a user to insert a peg into a hole. For the purpose, any holes and prominent components of each part are in advance given auxiliary lines. Assume that he wants to mate a part A and B by inserting A into B. In the case, he can choose several auxiliary lines first from those defined at prominent portions on A, and next from them at holes on B, and specify assembly relation to be made by having them cross each other. They are then made collinear as shown in Figure 6.3, which shows how to mate two blocks each of which has two osculating elements. This device makes it easy for a novice to manipulate parts at his will.

This mode allows him to create anything as far as he uses the parts provided with the system, but in the case, as the goal state is unknown to the system it can not give him a good advice when he gets into a trouble. If he intends to assemble an object included in the subassemblies or final assemblies already defined, he is obviously supported with the system.

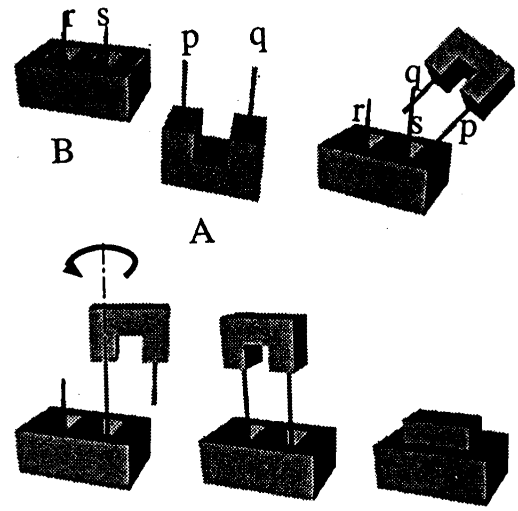


Figure 6.3 How to insert a peg into a hole in free mode.

7. Conclusion

We have shown that our system is able to provide with the training system helping a novice acquire how to assemble / disassemble a given object in three modes:
(1) restricted mode which forces him follow the given sequence of operations expert gave to the system in advance.
(2) intermediate mode which allows him to examine his own way using 3DTI.
(3) free mode which allows him to select and manipulate any parts at his will for constructing / decomposing a final assembly.
The system should augmented so as to give him haptic sensation which will make it more feasible to grasp or manipulate virtual object. man's virtual system guides his hand to the place where he want to manipulate it, but tactile sensation plays important roles afterwards.

References

[1] P. E. Hart: Progress on a Computer Based Consultation. Proc. of 4th IJCAI (1975)

[2] N. Abe, S. Yamada, S. Tsuji: A Consulting System which detects and undoes erroneous operations by novices. Proc. of SPIE (1986)

[3] Norihiro Abe, Jiang Y. Zheng, Kazuaki Tanaka and Hirokazu Taki: A Training System using Virtual Machines for Teaching Assembling/Disassebling Operations to Novices, International Conference on Systems, Man, and Cybernetics, pp.2096–2101 (1996)

[4] N. Abe, T. Shingoh: Generation of Technical Illustration from Description of Machines. 2nd IEEE Workshop on Robot and Human Communication (RO-MAN'93), pp.79-83 (1993)

[5] V. N. Rajan, S. Y. Nof: Minimal Precedence Constraints for Integrated Assembly and Execution Planning, IEEE Trans. Robotics and Automation, Vol.12, No.2, pp.175-186 (1996)

[6] A. Swaminathan, K. S. Barber: An Experience-Based Assembly Sequence Planner for Mechanical Assemblies, ibid, pp.252-267 (1996)

[7] S. Yamada, N. Abe, S. Tsuji: Construction of Consulting System from Structural Description of a Mechanical Object. Proc. of IEEE on Robotics and Automation, pp.1412-1418 (1987)

[8] J. L. Nevins, D. E. Whitney: Assembly Research. In: The factory of the future. In fotech. pp.7/1-7/39 (1979)

[9] J. Wernecke: The Inventor Mentor, addison-Weskey Publishing Company (1994)