Verification of Assemblability of Mechanical Parts and Visualization of Machinery of Assembly in Virtual Space

Norihiro Abe[1], Atsushi Wada[1], Kazuaki Tanaka[1],
Jiangyu Zheng[1], Shoujie He[2], and Hirokazu Taki[3]

[1]Dept of Mechanical Systems Engineering,
[2]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
[3]Mitsubishi Electric Inc. JAPAN
abe@mse.kyutech.ac.jp

Abstract

In addition to the CAD for the design of mechanical parts, computer application has been extended to the machining of the mechanical parts in recent years. However, it still remains challenging to come up with a computer system, which visualizes the process of verifying the assemblability and the functionality of a complete assembly. This paper presents a method for building a computer system, which verifies the assemblability and the functionality by using virtual reality technologies. The steps include the generation of the designed mechanical parts, the assembly of the parts, and the verification of the functionality. The key technologies include the detection of the collision, the representation of the assembly relations among virtual mechanical parts, the inference of the mechanism of motion transmission involved, and the techniques for the visualization of these processes.

Key words: Verification of Assemblability and Functionality, Mechanism Visualization, Virtual Reality

1. Introduction

CAD/CAM have widely been used to improve the efficiency of design and machining of mechanical parts for various applications. The design and machining of the individual mechanical parts, however, are no more than first-stage steps to the production of a functional machine.

One of the most critical steps in the whole process of design and manufacturing, is to check whether or not the mechanical parts under design could be assembled and whether or not the complete assembly will have a functional mechanism as expected. The existing mechanism checking programs require the users to write with a specific format the precise constraints amongst the mechanical parts. As a result, it has never been possible for the designers to make the best use of the CAD data to verify the assemblability and the functionality. Instead of using the mechanism checking programs, the designers have verified the assemblability by implementing the assembly operations in practice. Only if the assemblability test has passed, can the mechanism, i.e., the functionality, be verified. Either the mechanical parts could not be assembled as expected or the complete assembly does not have the functional mechanism will give rise to the modification to the original design specification. Imaginably, this is a time-consuming and cost-intensive trial and error process.

Computer graphics (CG) technologies are often used to simulate and visualize effectively the real-world phenomena. With the advancement of CG technologies and computer hardware technologies, virtual reality as a new enabling technology is drawing more and more attention. The assemblability can actually be tested in a virtual space by assembling the virtual counterparts of the mechanical parts under design. The functionality can also be verified in the virtual space by checking the mechanism of the complete virtual assembly. Since the virtual counterparts of the mechanical parts under design can easily be obtained with the existing 3D CAD tools, it becomes possible for the designers to use the CAD data to straightforwardly verify the assemblability and the functionality. Therefore, a low cost and unified design and verification process can be expected.

The qualitative and dynamic behavior of a complete assembly have been analyzed in the past [1], [2]. Our research, however, for the first time detects inappropriate connections and potential collisions through the implementation of assembly operations in a virtual environment, and successfully clarifies the kinematics function of mechanical parts by analyzing the dynamically changing connections amongst mechanical parts. Language analysis approach to the inference of the terminal state of assembly operations has been proposed [3]. The demerit is that the preparation of dictionary and syntactical rules is quite difficult. Vision based teaching-by-showing approach has also been challenged in the past [4], [5]. Since it is difficult to recognize with sufficient accuracy the shape of mechanical parts and the hand gestures, the assembly operations had been limited to simple ones such as pick-and-place of blocks.

The rest of this paper is organized as follows. Section 2 features the system data structures in detail. Section 3 addresses the problem of how to verify the assemblability in the virtual space. Section 4 describes a method for the verification of functionality, which is done by visualizing the mechanism of the complete assembly. Section 5 shows the experimental results. Section 6 concludes the paper with discussions on the future work.

2. System Data Structure

This system is coded with Inventor [6]. The data structures consist of two parts, for the management of graphics data and for the description of connection relationships, respectively. The graphics data include the following items:

(1) Information necessary for representing a mechanical part as a surface model such as vertices, surfaces, norm vector of the surfaces, colors, transformation parameters for translation, rotation, and scaling,

(2) Information necessary for explicitly showing the function related aspects of the mechanical part, i.e., the coordinates of the primitive components of the mechanical part and the direction vectors for the possible connection with other mechanical parts.

The above graphics data are managed with a tree structure as shown in Fig. 1. The following two rules are predefined to specify the relationships amongst the elements in the tree structure.

(1) Elements in the higher layers are dominant over the elements in the lower layers, i.e., the changes happened to the elements in the higher layers will affect the elements in the lower layers but the reverse is not true.

(2) Elements on the left are dominant over the elements on the right in the same layer, i.e., the changes happened to the elements on the left will affect the elements on the right in the same layer but the reverse is not true.

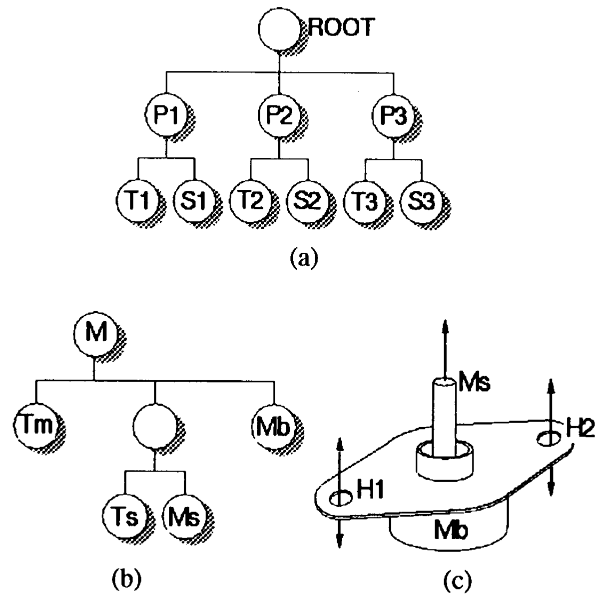

Figure 1: Data structure for the management of graphics data. P, S, and T in (a) stand for part data, shape node, and transformation node, respectively.

Let us use Fig. 1 (a) to further the explanation on the above two rules. In Fig. 1 (a), P1, P2, and P3 denote three mechanical parts. Ti (i = 1, 2, and 3) denotes the transformation such as scaling, translation, or rotation to each of the mechanical parts. Si (i = 1, 2, and 3) denotes the surface model related data. Obviously, P1 are dominant over T1 and S1 because the existence of T1 and S1 is determined by P1. T1 is dominant over S1 because the transformation specified by T1 affects the surface related data such as location, orientation, and size of the surface. Let us see the tree structure shown in Fig. 1 (b). In Fig. 1 (b), motor M is composed of shaft Ms and the motor body Mb. It can be seen that changing Tm will affect the whole motor but changing Ts will only affect the shaft Ms. As being described in Section 3.4.1, in the case that T1 and T2 are given the same amount of transformation, the corresponding two parts will be transformed at the same pace. If the constraints between the transformation amount given to T1 and the amount given to T2 are represented in a formula, we may specify the transformation amount with a single value. For example, the transformation amount corresponding to T1 is explicitly specified. Accordingly, the transformation amount corresponding to T2 will be automatically generated from the formula. In this way, a variety of kinematics propagation can be realized. The convex and concave related information is as shown in Fig. 1 (c). In Fig. 1 (c), the motor shaft Ms is in the convex shape. The potential assembly direction is shown with a directed auxiliary line. The holes H1 and H2, on the other hand, are in the concave shape. The corresponding auxiliary lines are as shown in the Fig. 1 (c). The directed auxiliary lines are not only used for the graphics representation but also for the interference checking necessary for the verification of assemblability and also for the inference of the termination condition of an assembly operation.

The connection relationships, on the other hand, are a bi-directional list of structures (structure: a data type of C programming language). Each of the structures includes the following information:

(1) The ID information of subordinate mechanical parts.

(2) The type of the connection relationships such as insertion, screwing-into, etc.

(3) The precise connection position on the mechanical part being described.

(4) The precise connection position on the subordinate mechanical parts.

Here, the subordinate mechanical parts are those that have connection relationships with and are subordinate to in terms of the motion behavior the mechanical part being described. The bi-directional list is used in two ways, for the inference of the possible transformation based on the constraints that have been satisfied in the process of assembling mechanical parts and for the inference of information related to fixing, constraining, propagation necessary for the operation of the complete assembly.

3. Verification of Assemblability

With the constantly changing demands from the users, the designers need to keep improving and modifying the specification of the mechanism of a machine. The modification of the mechanism, however, may give rise to some unexpected results. For example, the mechanical parts that could have been assembled together prior to the modification may not be assembled anymore. It is, therefore, necessary to verify the assemblability posterior to the modification of the mechanism. As described in Section 1, our system verifies the assemblability in a virtual space by working with the virtual mechanical parts.

If the connection amongst mechanical parts is simply considered a combination problem among the primitive components of the mechanical parts, we may have an overwhelmingly large number of complete assemblies. However, considering the original intention of the designers, we may prune away a large number of meaningless combinations at the beginning. The connection related data of the mechanical parts registered in our system are explicitly specified at the stage of modeling the mechanical parts. Assembly operations are implemented based on the connection-related data.

3.1 Assembly procedure

Mechanical parts displayed in the virtual space are selected with mouse click. The mechanical parts, which have more than one connection relation (i.e., assembly relation) with other mechanical parts, are shown with their auxiliary lines. The auxiliary lines are located close to the primitive components that contribute to the connections. As shown in Fig. 2 (a), part 3 has two holes (the primitive components) featured with two auxiliary lines explicitly. The two holes are designed to connect with other mechanical parts. Suppose an assembly operation of inserting part 1 into the right hole of part 3, is illustrated in Fig. 2 (a). When part 1 is selected with a single mouse click, the auxiliary line belonging to part 1 will be explicitly shown. Similarly, part 3 is also selected with a single mouse click. Further, mouse click is used to specify which auxiliary line on part 3 is the relevant one. In this case, the auxiliary line on the right is the relevant one. The steps for the implementation of the assembly operations are as follows:

(1) Choose a movable part and also its auxiliary line. In Fig. 2 (a), part 1 is the movable one.

(2) Choose a static part and also its relevant auxiliary lines. In Fig. 2 (a), part 3 is the static one.

(3) Move the movable part towards the static part. In Fig. 2 (a), part 1 is moved towards part 3.

(4) Rotate the movable part, if necessary, so that its auxiliary line becomes collinear with the relevant auxiliary line on the static part. In Fig. 2 (a), the auxiliary line on part 1 must be collinear with the auxiliary line at the right hole of part 3.

(5) Compute the movable range of the movable part and keep moving the movable part.

(6) Describe the connection relationship between the movable part and the static part if their auxiliary lines are successfully aligned.

The movable parts are those that are supposed to be attached to the static parts and thus normally small parts. The static parts are relatively large and are spatially fixed.

Let us explain the operations for the assembly in details. The insertion of screws should be accompanied by translation and rotation. In most of technical illustrations, however, the insertion of screws is specified with only translation but no rotation. In our system, the insertion of screw is also considered a kind of translation. The implementation of assembly operations for gears and worms, which requires rotation, is done with a key press.

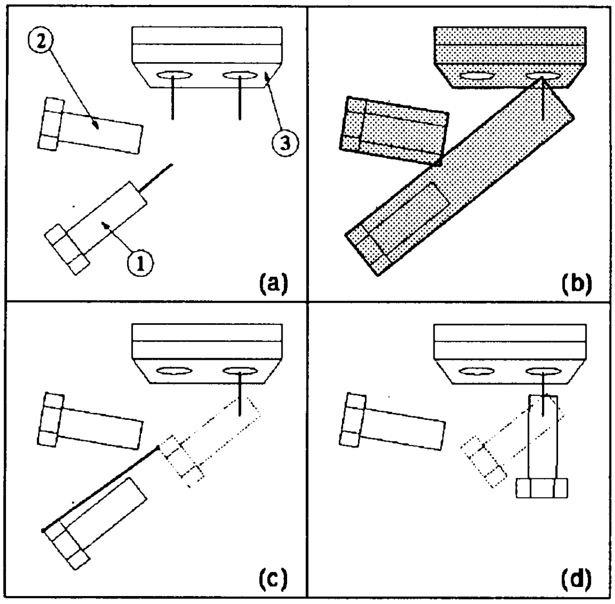

Figure 2: Collision detection on a moving path.

3.2 Collision detection

3.2.1 Handling of movable parts

A movable part such as part 1 in Fig. 2 (a) is usually moved towards a static part such as part 3 in Fig. 2 (a). Fig. 2 (b) shows the first-stage goal state for the movement of part 1. In order to move the movable part to the first-stage goal state, it is necessary to make sure that no obstacle exists on the whole movement path. The potential collision is detected with an efficient method, with which the potential obstacles along the straight-line path are detected with the interference detection.

3.2.2. Bounding box

The interference between two virtual objects is normally detected by checking the crossing between the surfaces of the objects. Since the checking is applied to all the possible combinations, it is very time-consuming when the virtual objects are still far away from each other. Generally, a coarse-to-fine approach to the interference detection is taken. A rough checking is first applied to the bounding boxes of the objects. Only if the bounding boxes are found interfered with each other, is the surface crossing checked [6]. However, since a bounding box is usually generated with their sides parallel to X-, Y-, Z-axes, the bounding box of a rotated object will occupy some extra space. As a result, even two rotated objects are still far away from each other, their bounding boxes might have already been found interfered with each other. For this reason, we propose a method for improving the generation of the bounding boxes. The basic idea is to make the bounding box be the smallest in terms of its volume.

Normally, the smallest bounding box of an object is the one generated from the object prior to the rotation. Our method for the generation of the bounding box is as follows:

(1) Reset the target object to be its initial state, i.e., no rotation is done to the object.

(2) Compute the minimum and maximum values among the vertices of the object.

(3) Use the minimum and maximum values to generate a bounding box.

(4) Rotate the object and its bounding box as required.

The collision detection along the movement path is as shown in Fig. 2 (b).

3.2.3 Precise detection of collision

The precise detection of collision is done by checking the crossing between line segments and surfaces. Fig. 2 (b) shows that the bounding boxes of part 1 and part 2 interfere with each other. The precise detection is done as shown in Fig. 2 (c). Here, line segments are obtained by drawing straight lines from the vertex points of part 1 to the corresponding vertex points posterior to the movement. The precise detection of collision is, therefore, done by checking the crossing between the line segments and the surfaces of part 2. If the crossing is found, it means part 2 is the obstacle on the movement path of part 1. The obstacle part 2 will be colored with red color and part 1 remains no change. As shown in Fig. 2 (c), part 2 is not an obstacle to the movement of part 1. As a result, part 1 can be moved to the first-stage goal state as shown in Fig. 2 (d). As shown in Fig. 2 (d), part 1 is rotated so that its auxiliary line is aligned with the auxiliary line at the right hole of part 3.

3.3 Determination of valid movement range

As shown in Fig. 2 (d), even though there is no other obstacle on the way of inserting part 1 into the hole on part3, the interference between part 1 and part 3 will unavoidably happen. Therefore, it is necessary to determine the movable range for the movable part, part 1. As shown in Fig. 3, the movable range is computed as follows:

(1) Generate a line segment, aa' and let aa' parallel to the norm vector of the hole.

(2) Find the crossing point b and compute the length of line segment ab.

Applying the above two steps to all the vertices of the movable part, we may get a set of line segments like ab. The movable range is determined to be the minimum length of the line segments.

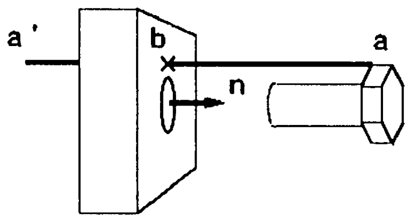

Figure 3: The determination of the movable range.

3.4 Description of connection relationship

In order to assemble the virtual parts and get the complete assembly working correctly, the connection relations amongst the virtual parts are constantly recorded into the bi-directional list described in Section 2. The reason of focusing on the connection relations is that only if the connection constrains are satisfied, can the function of the individual part be realized. Table 1 shows a set of example connection relations. In the mechanism terminology, two parts, which are connected and one of them transmits motion to the other, are referred to as "pair" [8]. In Table 1, "cylindrical-pair" relationships allow the two connected parts rotate and translate together. The "screw-pair" relationship allows the connected parts translate together. The "slider-pair" relationship allows the connected parts slide along a fixed direction.

Table 1. Connection Relationship

Shape A	Shape B	Connection relations
Cylinder	Hole	Cylindrical-pair
Screw	Hole	Cylindrical-pair
Screw	Tapped-hole	Screw-pair
Cylinder	Groove	Slider-pair
Worm	Worm-wheel	Worm-gear

3.4.1 Graphics data updating

As soon as they are connected, two parts must be considered as a whole. Any operation applied to either of them will be applied to the remaining connected counterparts as well. For instance, one of the two parts is selected, the other one is also selected. In order to let the connection work properly, we need not only to update the description of connection relationships but also update the management of graphics data. Figure 4 shows an example of updating the graphics data. Although the connection relationships can be used to infer the common transformation information, rotation is normally applied to the connected parts at the coordinate center of their own coordinate systems. As shown in Fig. 4, a common transformation node Tc is added to the left of the sub-trees of P1 and P2. Tc is used to specify the transformation applied to the connected parts. The old transformation nodes T1 and T2 are still kept, because Tc might be cancelled in the case that the corresponding assembly operation failed.

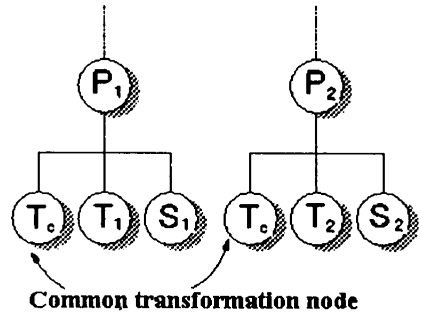

Figure 4: Graphics data updating in accordance with the updating of connection relationships.

3.4.2 Fastening of screws

The fastening of screws is divided into two types, loose and tight. In Fig. 5, the hole on the cylindrical side of part a and part b have a connection relationship. After the connection relationship is realized, i.e., part b is inserted into the hole of part a, keywords "tight" and "loose" will be displayed on the screen. Users are allowed to choose either of the types with mouse click.

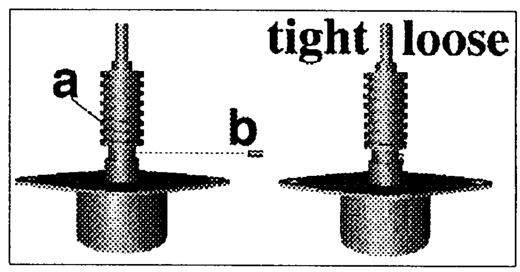

Figure 5: Selection of fastening mode of screw.

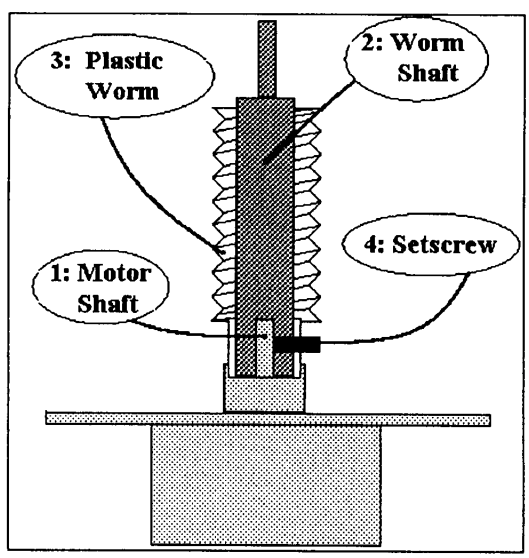

Figure 6: Cross section view of the complete assembly in Figure 5.

3.4.3 Side effects

In this system, one assembly operation is applied to a pair of selected parts at a time. At the beginning stages, one assembly operation only satisfies a single connection relation. At the stages followed, however, the selected parts may not be single parts but sub-assemblies and hence one assembly operation might give rise to the satisfaction of multiple connection relations. Among the connection relations satisfied, some might be invalid. It is necessary to handle all the possible connection relations as side effects.

Figure 6 shows the cross section view of the complete assembly in Fig. 5. In Fig. 6, the connection relation between part 4 (setscrew) and the hole on the cylindrical side of part 3 (plastic worm) has been satisfied. The connection relation is "screw-into", i.e., part 4 is screwed into the hole on the cylindrical side of part 3. The "screw-into" relation has been known at the time that part 3 and part 4 are selected. The satisfaction of the "screw-into" relation gives rise to the following side effects:

(1) The setscrew is tightened on the cylindrical side of part 1 (motor shaft).

(2) The connection relation between the setscrew and the hole on the cylindrical side of part 2 (worm shaft) has been satisfied.

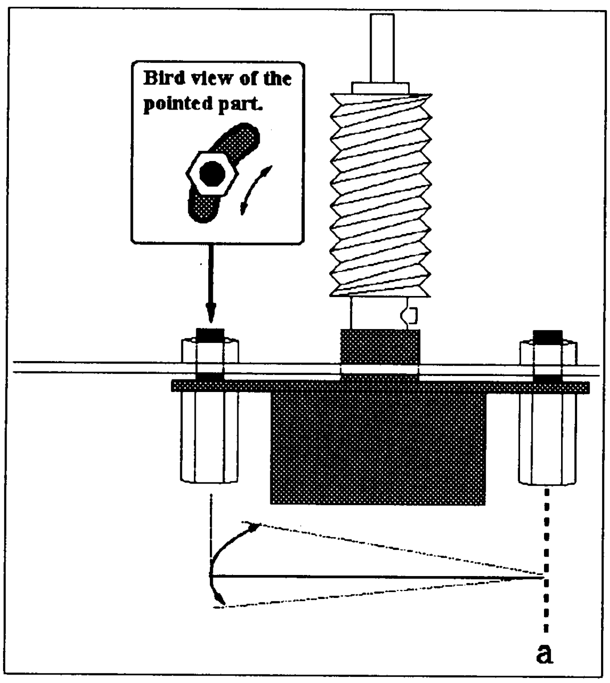

Figure 7: The adjustment of motor assembly for the connection between the worm and the worm wheel.

3.5 Connection constrains for assembly operations

Figure 7 shows the state, at which the complete assembly in Fig. 6 is mounted on the motor housing booth. It can be seen that the bold and nut are temporarily fastened. Since the hole on the housing booth is as shown in the bird view of Fig. 7, motor can be rotated with a limited angle along the a-axis. By rotating the motor assembly, we can adjust the location of the motor assembly so that the worm and the worm wheel are in gear correctly.

The adjustment process is as follows. Let the system know that the motor assembly will be rotated so that the system constrains the rotation freedom properly. Since the right bold and nut are fastened loosely, the rotation of the motor assembly along the a-axis is possible. As soon as the rotation axis is set to be a-axis, the system will handle the rotation moment. The rotation is realized with mouse clicking and dragging.

4. Mechanism Visualization

Even the assemblability test has passed, it is not yet guaranteed that the complete assembly will work as expected. Our system visualizes the mechanism and its behavior through the inference of the constraints for motion transmission based on the data of connection relationships.

4.1 Search for the connection relationships

Let us explain the terminology first. The concept of subassembly is used to address a strongly connected assembly. The main subassembly is used to address a kind of special subassembly, which activates some kind of motion. The subordinate subassemblies, on the other hand, are used to address the subassemblies, which will follow the motion transmitted from the main subassembly. In order to analyze the motion behavior of a complete assembly, we need to know accurately the information such as the composition of the main subassembly and also the composition of the subordinate subassemblies.

Let us consider the case that the motor in Fig. 6 is in operation. As depicted in Fig. 1, motor and motor shaft are modeled separately. If the motor is in operation, the motor shaft will rotate. How the rotation could be transmitted to the other mechanical parts is analyzed through the inference based on the connection relations.

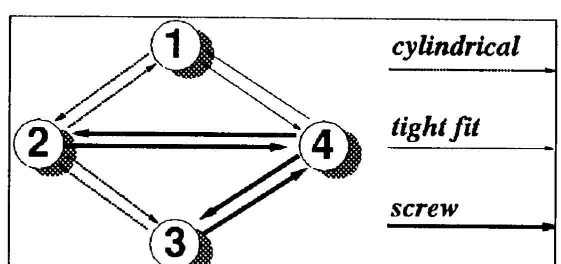

Figure 8: Search for the connection relation.

As shown in Fig. 8, part 1 (motor shaft) and part 4 (setscrew) are connected tightly. Since part 4 (setscrew) is tightly connected with part 2 (worm shaft) and part 3 (plastic worm), all the parts tightened on the motor shaft. Therefore, the rotation of the motor shaft is transmitted to all the other parts.

4.2 Motion transmission

Let us consider the case, in which the worm wheel has been successfully installed through the adjustment of the motor assembly as shown in Fig. 7. In Fig. 9, part 5 denotes the worm wheel. The connection relationship between part 3 and part 5 is an "in-gear" type. Motion is transmitted from left, i.e., part 1 (motor shaft) towards right, i.e., part 5 (worm wheel). Based on the analysis on Fig. 9, the management of the graphics data is updated as shown in Fig. 10. In Fig. 10, parts 1, 4, 2, and 3 are inseparable and hence are managed as a subassembly. Two additional transformation nodes are introduced to describe the motion behavior of the subassembly. We may actually add the transformation node in the way shown in Fig. 4. In order to represent the relationship with the transformation node in the subassembly that part 5 belongs to, the additional transformation nodes are added and linked on the higher level. The two links in Fig. 10, record the information on the rotation of subassembly (parts 1, 4, 2, 3) and the subassembly (part 5). The links update the values in the corresponding transformation node within a fixed time interval so that the transmitted motion is synchronized.

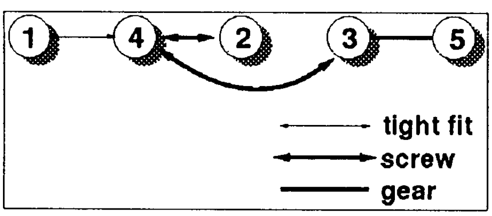

Figure 9: The mechanism for motion transmission.

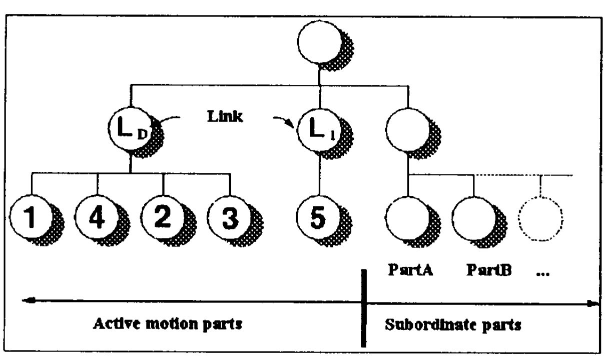

Figure 10: Updating of the graphics data.

4.3 Interference detection

A rough verification of the mechanism can be done by visualizing the motion behavior of the complete assembly. The detailed verification, however, cannot be done with the visualization. The detailed verification is better done through the detection of interference between the main subassemblies.

In Fig. 11, parts d and a have a common rotation axis but the two parts basically rotate independently. In the case that part b is fixed by parts a and c, parts b and d will be constrained together. As a result, parts d and a may rotate together. The interference between parts d and b will be detected when part d is rotated. Based on the interference information, we may conclude that part b will rotate together with part d. Since parts a, b, and c are bound together, parts a, b, c, and d will rotate at the same pace. Accordingly, the graphics data will be updated.

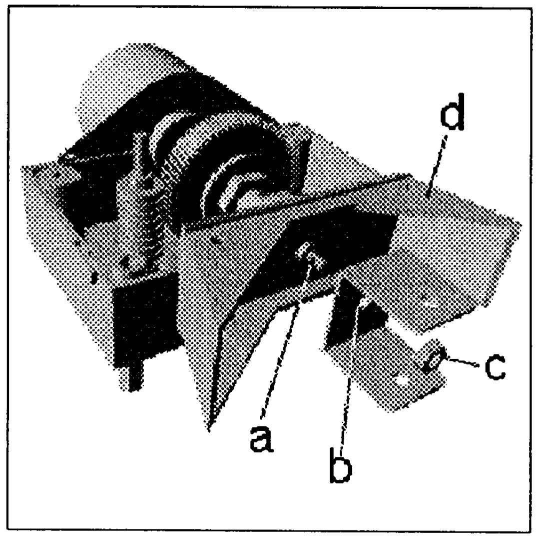

Figure 11: An example for illustrating the case that only connection relationships are not sufficient for analyzing the motion behavior.

5. Experiment: A Case Study

Figures 12, 13, and 14 show the whole process of applying our system to the major components of a robot hand. In this case study, we assume the following are true:

(1) All the mechanical parts used are rigid objects;

(2) Appropriate tolerance exists for all the insertion operations;

(3) Friction accompanied with the insertions is ignored;

(4) The interference between the gears such as worm and worm wheel, which are in complicated shape, is not detected.

6. Concluding Remarks

This paper presented a virtual reality based approach to the verification of the assemblability of mechanical parts and the functionality of the complete assembly. The verification of the assemblability has been done through the detection of the interference between line segments and surfaces. The verification of the functionality, on the other hand, has been done with two steps, a rough check and a detailed check. The rough check has been done through the visualization of the mechanism, while the detailed check has been done through the detection of interference when the complete assembly is in operation. It has also been shown that through the detection of interference and the description of connection relationships, the non-deterministic motion behavior can also be conjectured.

The future enhancement of our system includes, (1) handling mechanical parts modeled with curved surfaces; (2) verifying the motion transmission amongst more than three subassemblies; (3) handling the connection relations with non-rigid objects.

References

1. Joskovies, L. and Sacks, E.P.: Computational Kinematics, Artif. Intell., Vol. 51, pp. 381-416, 1991.
2. Gelsey, A.: Automated Reasoning about Machines, Artif. Intell., Vol. 74, pp. 1-53, 1995.
3. Abe, N., Ishikawa, T., and Tsuji, S.: Generation of Assembly Procedure from Assembly Instruction, JSAI, Vol.3, No.5, pp. 590-598, 1988.
4. Kuniyoshi, K., Inabaand, M., and Enoue, H.: Learning by Showing, IEEE Trans. R & A, Vol. 10, No. 6, pp. 799-822, 1995.
5. Ikeuchi, K and Kang, S. B.: Vision-based Teaching to the Robot Hand, Journal of Robot Society of Japan, Vol. 13, No. 5, pp. 509-602, 1995.
6. Wernecke, J.: The Inventor Mentor, Silicon Graphics, 1994.
7. Toriya, H. and Chiyokura, H.: 3D CAD Fundamentals and Applications, Kyoritsu Publishers, 1991.
8. Yasuda, T.: Machinery, Corona Publishers, 1990.
9. Kitajima et al.: Interactive System for Verifying Assemblability of Machines Based on the Directed Graphs , Precision Instruments, vol. 49, No. 2, p. 208, 1983.
10. Sekiguti, Kojima, Inoue, Honda: Study on Part Development of Rotating Machinery Components, Precision Instruments, vol. 51, no. 2, p. 359, 1985.
11. Tanaka, K., et al.: Verification of Assemblability of Mechanical Parts and Visualization of Machinery of Assembly in Virtual Space, Transaction of Information Processing Society of Japan, Vol. 38, No. 10, pp1976-1985, 1997.

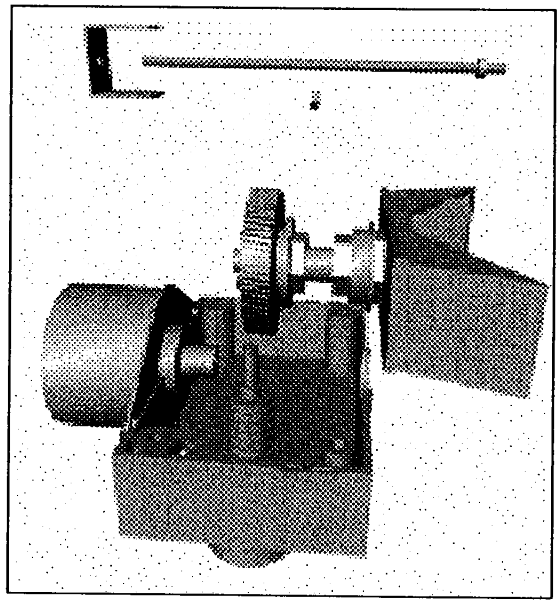

Figure 12: A state of implementing assembly operation.

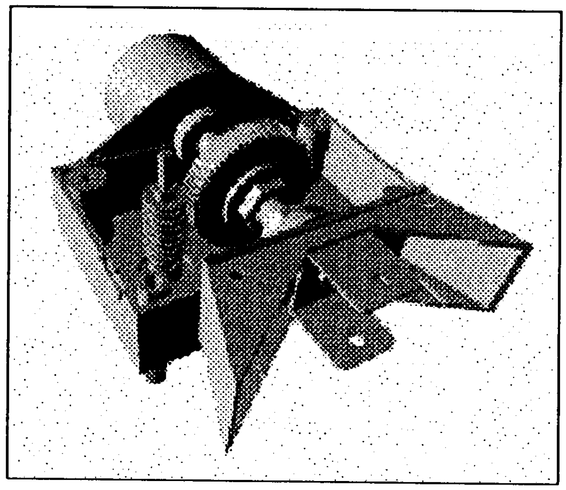

Figure 13: A complete assembly.

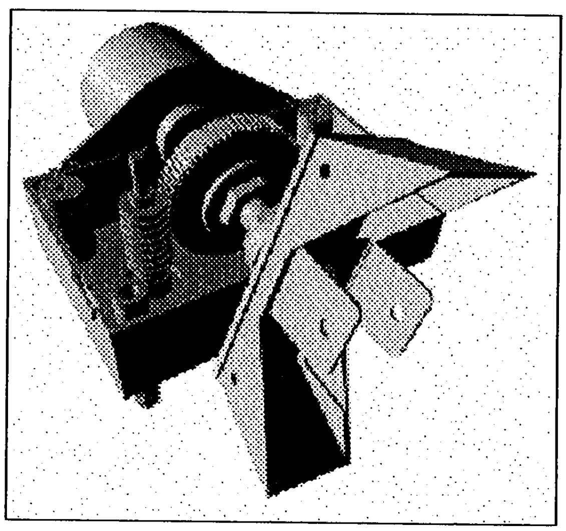

Figure 14: Motor is in operation.