Selective Disassembly of Virtual Prototypes

N. Shyamsundar, Rajit Gadh
Department of Mechanical Engineering,
University of Wisconsin, Madison,
Madison, WI - 53705, USA
E-mail: gadh@engr.wisc.edu

Abstract

Maintenance and reuse of products often require the disassembly of particular components of the products. In other words, selective disassembly is necessary. This paper presents a method to find the disassembly sequence required to selectively disassemble an identified component of a virtual prototype. Use of the virtual prototype facilitates the disassembly analysis to be performed during the design process. The disassembly sequence and the directions of disassembly of the components are generated, by using the components’ mating faces. In the current work a recursive method is used to identify the disassembly sequence required to disassemble the identified/target components. The final output is the disassembly sequence and the range of directions along which the components are to be disassembled to remove the target components from an assembly.

1. INTRODUCTION

The assembly analysis of a product is usually done during the product’s design. This results in an assembly plan that gives the sequence and the path used to place the components in the assembly. Hence, during disassembly the disassembly plan could obtained by reversing the assembly plan. However, this is only possible if no irreversible operation is done during assembly. In practice this is not possible when the several irreversible operations like riveting and welding are used. This is one of the factors that makes disassembly planning different from assembly planning. Another key difference between assembly and disassembly occurs during the removal of only one or more components from the assembly. This we term selective disassembly. Using the reverse of the assembly plan for selective disassembly might be possible. But it may not be the best possible order in which to remove the components. In this paper we present a method for generating the plan for selective disassembly of components. Selective disassembly is motivated by the need for recycling, reuse and maintenance. In these application domains the need for disassembling a portion of the assembly arises very often. For example, during maintenance of a product it is usually necessary to replace only a few components of the assembly. Hence, a disassembly plan for removing the components and replacing the new one is required.

If the product is to be analyzed for maintenance and reuse during the design phase, a selective disassembly tool would be very useful. In the detailed design phase the material of the components and their life expectancy would be known. This information could be used to identify
- the components that are prone to failure and would require replacement.
- the components that could be reused at the end of the life of the product
- the components that are to be separated in case of recycling of the material of the components.

In most of the above mentioned situations selective disassembly would be used rather than complete disassembly of the components. Thus methods for automatically generating the disassembly plan for selective disassembly of the virtual prototype will help in suggesting design modifications. In this paper the focus is on generating automatically the selective disassembly plan for virtual prototypes. When a product is not designed for disassembly, the disassembly strategy for the product has to be found at the end of its life. Finding the disassembly strategy at the end of the life of a product is expensive and time consuming. Our approach emphasizes the development of the disassembly strategy for the product at the design stage when the virtual prototype is present. A virtual prototype is an abstract model (like a computer model) that is created after the detailed design phase before the manufacture of the physical prototype. Performing the disassembly analysis on the virtual prototype of the object well before it is produced results in the reduction of the disassembly cost for the product to be recycled or reused at the end of its life.

2. RELATED RESEARCH

Disassembly planning techniques are related to assembly planning ones. Reversing the assembly sequence would yield the disassembly plan. The assembly planning for components of an assembly and the sequence of these operations has been analyzed by Wilson and Latombe [11]. The concept of Non-directional blocking graphs has been developed by Wilson to analyze assemblies. Methods to determine the constraints on the transitional and rotational motion of planar and 3-D objects from their contact geometry has also been investigated [8]. Determination of the geometric path required to remove a portion of the assembly contained in a cavity within the parent assembly has been addressed by Xu et al [14]. In this work, all possible geometric paths are identified by generating the partial medial axis of the free space within the assembly. A graph search technique is used to find the optimal path.
Partitioning of an assembly into several sub-assemblies has been studied by Halperin [3]. The Non-directional Blocking graph has been extended to handle compound removal paths for assembly partitioning. The assembly planning scheme developed by Liu and Popplestone [7] uses the matching of the models with a library of standard component features. However, there exist several differences between assembly and disassembly of a product at the end of its life. The assembly process is reversible only as long as no irreversible operation is done, but in practice there might be several irreversible operations done on an assembly. This leads to the observation that the most economical assembly sequence need not necessarily be the most economical disassembly sequence. Hence a study of disassemblability of products is essential.

There exist several approaches for finding the disassemblability of products. Woo and Dutta [13] have developed an algorithm to determine if the components of an assembly are disassemblable. In this algorithm, the geometric constraints of the component with respect to the rest of the assembly are considered. Only sequential assemblies with 1-disassemblable components have been considered. They have proved that in order to determine the disassemblability of a component, it is necessary and sufficient to consider only the faces of that component which mate with the rest of the assembly (i.e. the mating faces). In a later study of disassembly, Dutta and Woo [2] take parallel assemblies into account. They describe a way of identifying a subassembly to initiate the disassembly process for a 2.5D parallel assembly (an assembly with a uniform cross-section in one of the three dimensions). Beasley and Martin [1] consider the generation of disassembly motion for voxelized models of objects. A voxelized model is one in which the object is represented as a collection of cubes. The infinitesimal motion and the finite motions required to disassemble a component can be generated by the method proposed by them. Shin and Cho [9] present a method in which knowledge based rules are used to determine the disassemblability of a part. To analyze the part a topological product model is required. Though these approaches determine the disassemblability of the components, the problem of selective disassembly has not been addressed.

Kroll et al [4] present a rating scheme. This scheme is useful for the designer to identify weaknesses in design and compare alternatives. In particular, the focus is on robotic disassembly of small electrical appliances. Lowe and Niku [6] present a framework of concepts used to analyze disassembly during the design phase. A scoring method has been proposed in which the cost effectiveness of disassembly of the entire product can be found [6]. Hanft and Kroll [12] have proposed a chart and rating scheme for quantifying disassembly difficulty. The method is used to identify the sources of difficulty in performing each task. Manual disassembly of business equipment is the primary focus of the method. Automatic evaluation of the ease of disassembly of virtual prototypes in the context of robotic disassembly has also been addressed [10]. The approaches discussed provide methods to quantify the ease of disassembly.

In this paper the relevance of selective disassembly to application domains of maintenance, recycling and reuse has been discussed. The method proposed in this paper uses a recursive algorithm and utilizes the contacting faces of the component to find the disassembly plan for selectively removing the desired component from the assembly.

3. MAINTENANCE

During maintenance of a product it is necessary that certain components be disassembled to be replaced or serviced. In this application domain, the assembly needs to be dismantled in such a manner as to facilitate the easy removal of the target components. During the design of the product there are tools to estimate the frequency of maintenance required for the components. Once the information is available a disassembly analysis for selectively removing those components can be done. Such an analysis if done, on the virtual prototype, will help the designer to suggest possible modifications in the design. This might include changes in the shape or material of the component. For example, the break linings of an automobile would be subjected to a lot of wear. So the break lining needs to be replaced. A selective disassembly analysis of the breaking system with the lining as the target component could be done. This would give the designer an insight into the possible design modifications necessary for easy replacement of the lining during maintenance.

4. REUSE

Reuse of the components of a product can be planned at the design stage itself if the appropriate design tools are provided. Alternatively, with the available techniques the suitability of a component for reuse can be determined and the disassembly plan for removing certain selected components from the product before disposing the product can be done. As an example we could consider the engine of an automobile. Using non-destructive testing it can be determined if some of the components are suitable for reuse. Then the disassembly plan for removing the identified components can be generated by analyzing the virtual prototype.

5. RECYCLING

When products comprising of component(s) manufactured from a homogeneous material are recycled, the operation of recycling is straightforward since disassembly is not essential. But it turns out that the more bulky products like automobiles, computer equipment and other home appliances have a large number of parts that are made from different materials. To make a product with parts composed of a single material, like an all-aluminum engine, results in an inefficient design. Optimizing each component for its function results in each component being made of different materials. For recycling of the materials chemical separated of the materials in necessary.
Separation of the materials in a product requires the
disassembly of the components. Selective disassembly is
invariably very useful for analyzing the recyclability of
products. A typical situation is one which all but a few
components of the product are made of materials that are
suitable for melting. In order to melt the whole product for
chemical separation of the materials, the removal of the
incompatible components is required. For example during the
recycling of cars, the car seats need to be removed before
further processing is done. This is because the seats are
composed of materials which make the chemical separation of
the rest of the materials difficult. The material of the
components of the a product would be available at the detailed
design phase. The selective disassembly analysis method
presented in this paper can be used as a design tool to evaluate
the recyclability of the product at the design stage.

**Separation Directions and Disassembly Directions**

The disassembly plan generated by the method discussed in
this paper is for robotic disassembly of the components. A
detailed explanation of the algorithm requires the use of
certain terms which are explained in the rest of this section. In
an assembly of \( n \) components the \( i^{th} \) component is \( C_i \), \( i = 1, \ldots, n \). Components that are identified for removal from the
assembly are the target components and are denoted as \( C_i^t \).
Similarly the component currently under consideration is
called an active component and is denoted by \( C_i^a \). The
identification of the components in contact with the active
component and the choosing of the active component(s) for the
next recursive call is done during a recursive call. The set of
components in contact with the active component is denoted by \( N^t \). The set of components which are chosen from \( N^t \) as the
active components is denoted by \( R \). For simplicity in the rest of
the paper the cardinal number of the set \( R \) is assumed to be
one.

For each pair of components in contact, the range of
directions along which the components can be separated from
each other gives the separation direction. The **separation
directions** of component \( i \) from \( j \) is denoted by \( S_{ij} \).

![Figure 1. Separation Directions](image)

A disassemblable component can be removed along a direction
or a set of directions. These directions are called the **range of
disassembly directions** and are denoted by \( DD \). \( DD \) of a
component \( C_i \) is denoted by \( DD_i \).

![Figure 2. Range of Disassembly Directions](image)

For example the separation directions of \( C_1 \) and \( C_2 \) in Figure
1 (a) is \( S_{12} \) and is a single direction. In this research the range of
disassembly directions and the separation directions are
represented by mapping them onto a Gaussian Sphere (of unit
radius). The separation directions may be any one of the
following:

a) A single direction which maps as a **point** on the
Gaussian sphere (e.g. Figure 1 (a) or (b)).

b) A set of directions which map as an **arc** on the
Gaussian sphere (e.g. \( S_{12} \) in Figure 1 (c) is the
bounding semi-circle of the shaded area).

c) A set of directions which map as a on the Gaussian
sphere (e.g. \( S_{12} \) in Figure 1 (d) is the shaded area).

Each of these entity types is of a different dimensionality and
therefore cannot be compared to one another. Hence, it is
necessary to represent them in a consistent manner. To
achieve this, the surface of the Gaussian sphere is discretized
into quadrilateral patches of equal area to unifying points, arcs
and surfaces. The area of the quadrilateral patch is chosen to be
\( 1/u^2 \) (where \( u > 1 \) and \( 1/u \) is in radians) and the choice of "\( u \"
depends on the resolution of the robotic manipulator. \( H \) for a
component is defined as the number of quadrilaterals required
to span the range of disassembly directions on the gaussian
sphere. This approximation works because in practical
situations the robotic manipulator used for disassembly will
have very little resolution so two directions a few degrees apart
will not produce any significant change in the motion of the
manipulator. Thus, a proper value of "\( u \" is chosen based on
the resolution of the robotic manipulator being used for the
disassembly operation. In this paper \( u \) is assigned a numeric
value of ten.

\( H \) for different \( S_{ij} \) is termed as \( H_{ij} \) and its evaluation is
illustrated in Figure 1. \( H \) is a dimensionless number. In Figure
1 (a), the separation direction for $C_i$ is along one direction only. Hence the value of $H_i$ is 1 since only one quadrilateral is required to span the disassembly direction. $H_i$ for assembly in Figures 1 (b) is 2 as there are two directions along which $C_i$ can be disassembled (two quadrilaterals are needed to span $S_i$).

In Figure 1 (c) $S_{c1}$ is composed of several directions whose end points trace a semi-circle on the Gaussian sphere. In order to span the range of disassembly directions a strip composed of quadrilaterals of width $1/2$ and the total length adding to $\pi$ is required. Hence, the corresponding $H$ value is $\pi u$. Thus $\pi u$ quadrilaterals are required to span the range of disassembly directions. In Figure 1 (d), $H$ is the ratio of the area of the hemisphere and the area of each quadrilateral patch which is $2\pi u^2$.

The disassembly directions of several components are shown in Figure 2. The mapping of the range of disassembly directions onto a gaussian sphere is also illustrated. The range of disassembly directions could be points (Figure 2(a),(b)), arcs (Figure 2(c)), or a surface (Figure 2(d)). The component can be removed from the rest of the assembly along any one of the directions within the range of disassembly directions.

The order in which the components of an assembly are removed is called a disassembly sequence. For example, to remove $C_i$ from the assembly shown in Figure 2(a), components $C_o$, $C_p$, $C_s$, $C_n$, $C_c$ could be removed in that order. This is denoted by

$$ S = \{ C_o, C_p, C_s, C_n, C_c \} $$

The approach and the metric discussed in the paper are used for 3-D assemblies. For easy illustration, the working of the approach is discussed using 2-D assemblies.

In the current research, assemblies which satisfy the following requirements are considered.

a) Sequential assemblies. A sequential assembly is one in which the disassembly of no component requires the prior disassembly of $k$ (>1) adjacent components, in parallel [13]. The assembly shown in Figure 2(a) is a sequential assembly.

b) Components are 1-disassemblable. A 1-disassemblable component is one in which only one motion is required to disassemble the component [13]. All the components in the assembly shown in Figure 1 are 1-disassemblable.

c) Components have only translational degree of freedom and are non-deformable. Also, the assemblies are assumed to be frictionless.

**Determination of Disassembly Directions and Separation Directions for a Component**

To find the disassembly directions for a 1-disassemblable component the knowledge of the contacting faces of the component is utilized. These faces are called the mating faces. In Figure 3, the $C_i$ and $C_j$ are in contact along two surfaces $M_i$ and $M_j$. These are also termed as the mating faces of the components. The separation direction of a component $C_i$ from the mating face $M_k$ of component $C_j$ is denoted by $S_{ik}^k$. In Figure 3, $S_{21}^1$ and $S_{21}^2$ are the separation directions of $C_2$ from the mating faces $M_1$ and $M_2$ of component $C_1$. In general, the separation directions of a component $C_i$ from $C_j$ is calculated as follows.

$$ S_{ij} = S_{i1} \cap S_{i2} \cap \ldots \cap S_{in} \cap S_{jn} $$

where $\cap$ represents intersection, and there are $r$ mating faces between $C_i$ and $C_j$. $S_{21}$ gives the separation direction of $C_2$ from $C_1$ (Figure 3).

![Figure 3. Finding the separation directions](image)

Since there are only two components in the assembly shown in Figure 3, the disassembly direction of $C_2$ is same as the separation direction of $C_2$ from $C_1$. In the example shown in Figure 4 there are three components. The disassembly directions of component $C_i$ in contact with $m$ other components is given by

$$ DD_i = S_{i1} \cap S_{i2} \cap \ldots \cap S_{im} \cap S_{jm} $$

where $\cap$ represents intersection and components $C_{i1}, C_{im}$ have at least one mating face with $C_i$. In Figure 4 the disassembly direction of $C_2$ is a semi-circle. Also if $DD_i = \text{NULL}$, $C_i$ is not disassemblable.

![Figure 4. Finding the range of disassembly directions](image)

**Disassembly Sequence Generation**

The current work uses a recursive algorithm to identify the disassembly sequence required to disassemble the identified/target component. The components which mate with the target component are identified. These are the neighbors of the target component which is also the active component for this recursive call. The separation directions for the active component with each of its neighbors is found. Then the range of disassembly directions for the component is found. If it is not null then the recursion stops. Otherwise, one of the neighbors is chosen as the target component for the next iteration. Several active components can be chosen from the neighboring components. For simplicity we choose only one component. This has the largest range of separation directions and if removed should result in a non empty range of disassembly directions for the active component. As stated earlier, the recursion proceeds until the target component is disassemblable from the rest of the assembly. The inverse of the order in which the recursion proceeds gives the disassembly
sequence for disassembling the target component from the assembly.

The algorithm is as follows:

**Step 1:** For all components in the assembly find the separation directions and the disassembly directions and store it in a suitable data structure (DT).

**Step 2:** Make one of the target components ($C_i^*$) the active component ($C_i^*$). Call the **Procedure Sequence** ($C_i^*$).

**Step 3:** Repeat step 2 till all target components are disassembled.

**Procedure Sequence ($C_i^*$)**

```java

Step 1: If the disassembly direction of $C_i^*$ is not NULL, update DT and return.
Step 2: Find the set of contacting components $N'$ for $C_i^*$.
Step 3: Choose one component $C_j$ that belongs to $N'$ and has the greatest separation direction ($\text{Max}(H_{ij})$). Also $\text{DD}_j$ after removal of $C_j$ should be non empty. Call the **Procedure Sequence** ($C_j^*$). This becomes the active component for the next recursive call.
Step 4: Output $C_j$
```

$C_j$ is chosen to be the next active component since the greater the range of disassembly directions the easier it is for a robotic manipulator to access the component. In case removal of $C_j$ does not yield a non empty $\text{DD}_j$, the component with the next higher range of separation directions is chosen. This process is repeated till the component chosen if removed produces a non null range of disassembly directions for the active component.

For the assembly shown in Figure 5, $C_3$ is the target component. So during the first recursive call $C_3$ is the active component. The neighbor with the greatest separation direction is $C_2$. Also removal of $C_2$ would yield a non null $\text{DD}_2$. In the next recursive call $C_2$ is the active component. The disassembly direction of $C_2$ is not NULL. Hence the recursion stops.

![Figure 5. Assembly with $C_3$ as the target component.](image)

**First recursive call:**

$C_3^* = C_j^* = C_3$, $N' = \{ C_1, C_2, C_3, C_4 \}$, $\text{DD}_j = \text{NULL}$.

$R = \{ C_j \}$.

![Figure 6. Separation directions of $C_j$.](image)

**Second Recursive call:**

$C_3^* = C_j^*$, $N' = \{ C_j, C_3, C_4 \}$, $\text{DD}_j$ is not NULL.

![Figure 7. Separation directions of $C_j$.](image)

The disassembly sequence for removing component $C_3$ from the assembly is

$S = \{ C_3, C_3 \}$.

Once the sequence is generated the disassembly directions for each component is re-computed. This is necessary since the removal $C_3$ changes the disassembly directions of $C_3$. Thus except for the last component ($C_2$ in Figure 5) the disassembly directions for the rest of the components need to be re-computed after the removal of each component. Figure 8 shows the disassembly directions of the components.

![Figure 8. Disassembly Directions for disassembling $C_3$.](image)

**DISCUSSION**

In the algorithm presented, during a particular recursive call there will be an active component (e.g. $C_j^*$). The active component for next recursive call then is chosen from the set of neighboring components, $N'$ of the current active component. If more than one neighbor is present only one of them is chosen. This choice is based on the separation direction. As stated earlier the component with the greatest separation direction and whose removal results in a non empty range of disassembly direction for the active component is chosen (say $C_j$). It is possible that $C_j$ if removed would provide greater range of disassembly direction for the removal of $C_j$. This results in easier accessibility to the component $C_j$.

- Designing fixtures for such components is economical because there is more flexibility in terms of directions along which a component can be fixtured.
Generating the path plan for robotic disassembly is also less complicated. A less complex path ensures quicker removal of the components with minimal effort.

In case there are more than one target components. The algorithm first chooses one of the target components and identifies the disassembly sequence required for removing that component. The remaining portion of the assembly is examined to determine if any more target components are still present in the assembly. If present, one of them is chosen and the procedure for finding the disassembly sequence is repeated.

The algorithm does not consider the destructive disassembly of parts. Destructive disassembly involves the cutting of one or more components to facilitate the removal of other components [5]. This option needs to be investigated when disassembly cost is of importance during the evaluation of the suitability of products for recycling. More often when the materials of the components are to be recycled destructive disassembly is used. During maintenance destructive disassembly is usually not practiced. Also when a large number of products are to be reused destructive disassembly might not be preferred. To incorporate destructive disassembly into the selective disassembly approach, the cost and nature of the destructive approaches are to be considered.

There exist methods for quantifying the ease of disassemblability in case of complete disassembly [10]. The scope of the selective disassembly algorithm presented in this paper can be increased by forming metrics which quantify the ease of disassembly and other important characteristics of the assembly.

**SUMMARY**

The paper presents an algorithm for generating the disassembly plan for selective disassembly. The relevance of selective disassembly to the application domain of maintenance, recycling and reuse has been outlined. Factors like irreversible assembly operations and breakage of components make disassembly planning different from assembly planning. Examples have been used to illustrate the algorithm for selective disassembly. The disassembly plan is optimized for robotic disassembly. An automatic method for analyzing the virtual prototype for selective disassembly will help the designer to effectively design the product for disassembly.

Future work would include the generalizing the selective disassembly method to generate the disassembly plans for components requiring complex motions for disassembly. When there are several target components present. Having more than one active component might yield a better disassembly sequence. Another area of research is optimizing the selective disassembly method to handle cases which require destructive disassembly.

**Bibliography**


