Disassembly movements for geometrical objects through heuristic methods

Jorge Pomares^{*}, Fernando Torres, Santiago T. Puente

Department of Physics, Systems Engineering and Signal Theory.

University of Alicante, Alicante (Spain)

This paper presents a non-destructive disassembly method. It tries to make up for limitations of other methods based on contact surfaces. It uses a contact surfaces graph and shows an algorithm to obtain the movement sequence needed for the disassembly of two objects, using a composition of transfer movement. It performs a second filter to obtain the exact disassembly direction from the group of directions obtained from the contact surfaces. The filter employs a mobile robotic heuristic to feedback the contact surface graph throughout the disassembly process. The heuristic generates an environment map to infer the region where the probability of collision with the other objects is lower. To achieve the disassembly the paper presents a method for modelling the elements implied in the disassembly process. This virtual model of the environment allows us to quickly process the elements and the simulation of the movements for disassembly. This model can be used to either schedule the disassembly process or to test, during the design stage, whether the products can be easily disassembled.

^{*}jpomares@disc.ua.es; phone 34 965903682; fax 34 965903682; <http://www.disc.ua.es/gava>;
Dep. Of Physics, Systems Engineering and Signal Theory. University of Alicante, Alicante
(Spain).

1. Introduction

At present, recycling is being introduced more and more in the process of design of new products. Once the life cycle of the product has finished, its recycling allows us to reduce the costs, thanks to the reusability of its components. Another interesting aspect of the recycling, which is increasing in importance today, is the reduction of the ecological impact of the product (this is what green engineering refers to).

The optimisation of resources with their reusability is the motive of recent research due to the interest that it supposes. Recycling allows, on the one hand, a reduction of expenses and, on the other hand, a reduction of the ecological impact of the product once its life-cycle has finished.

The first phase of the recycling process is the disassembly of components. This disassembly process allows us to break the product down into its different components, taking different criteria into account (functional, according to materials...). In (Güngör et al. 2001) are shown some of the desired objectives in the disassembly of products. Among such objectives we should mention the recovery of valuable components, the elimination of defective parts, the reusability of certain parts, reduction of the amount of left-overs... In (Eckardt et al. 1998), once again, it shows the convenience of reusing the components of an assembly, once its life-cycle has finished. In such a case, a study is made from the economic and industrial point of view describing the different elements that require an automatic disassembly system. The need to have a preconception of the requirements for the disassembly of products from the initial phases of their design, has given rise to the term “design for disassembly”.

A product designed with the application of the principles of “design for disassembly” could easily be disassembled and, thus, their components could be recycled with less difficulty and within a shorter period. When the criterion of “design

for disassembly” is applied, we introduce in the designing of a product, the possibility of the disassembly and the recycling of its components.

In (Dowie et al, 1995) the main guide-lines that the designers must follow in “designing for disassembly” are described. An evaluation of the facility of disassembly and reusability can be made through either the creation of a hardware assembly or using a software simulation. In this article, we propose the second option which allows us to obtain a simulation of the disassembly process, from a CAD model of the product to be disassembled.

There are two main varieties of recycling, one of which is “destructive recycling”. In this type of recycling we eliminate some part of the product previously assembled by destroying or damaging some other component of the assembly. On the other hand, there is “non-destructive recycling, in which we can disassemble each one of the components without affecting any of the others. In this article we shall present a method for the modelling of assemblies that, later on, allows us to determine the sequence of movements necessary to eliminate any component of the assembly. The method presented is based on non-destructive disassembly.

The recycling of electronic components, as in recycling in general, requires several different steps. After the life cycle of the electronic components of a product has finished, its disassembly is the first step in recycling. This disassembly implies the non-destructive separation of the components. As is shown in (Hesselbach et al. 1994) the disassembly is an essential part of the process of the recycling of electronic components, such as a PC. It allows us to eliminate toxic substances and recycling operative components for their reusability.

In this article we propose an algorithm for the generation of the movements for disassembly. We apply this algorithm to the disassembly of components of a PC. Within

this field there are several recent publications like (Kuo et al. 2000, Lee and Martin 2000, Zhang and Kuo 1997) in which different techniques are applied for the disassembly of computers. Nowadays, the disassembly and recycling of PCs is gaining more and more importance (Kopacek and Kopacek 1998).

To be more specific, in this research we examine the case of a PC that must be disassembled automatically. In order to carry out this process, a model of the PC is made in such a way that allows us to determine the characteristics of the elements implied in the assembly. The characteristics are those that offer sufficient information to determine the sequence of disassembly movements required for a given component. Nevertheless, to be able to carry out automatic disassembly, using an robot arm, other characteristics are necessary to allow the grasping of objects and the determination of trajectories to be followed by the arm. We use the model mainly for the determination of movements for disassembly. It must, therefore, include the necessary geometric information. We also include considerations for its future implementation in automated systems based on robot arms.

In the proposed disassembly system we have a module of simulation that allows us to make a pre-visualization of the steps that are required for the elimination of a given component which is determined by the scheduling module.

The rest of the article is organized as follows: In section 2 we show the steps needed for the disassembly process. In the sections 3, 4, 5 and 6 we describe these steps. In the section 7 we show a simulation of the results obtained. Finally, some of the conclusions that can be extracted of this research are presented, along with suggestions for possible future studies.

2. Phases of the disassembly process

The main tasks considered in the disassembly process are the following: the modelling of the assembly, disassembly scheduling, the determining of the sequence of movements for the disassembly of a given component and the simulation of the process.

In Figure 1 we observe the different phases of the disassembly process:

[Insert Figure 1 about here]

We have a database of components that is a source of information for the generating of the assembly model. The generated model is the base for the determination of the movements of disassembly of a component from the rest of the assembly. At every moment, the scheduling module indicates what component is to be separated.

Another important aspect of a disassembly application is the virtual simulation of the disassembly process. This phase is represented in figure 1 and it shows the user a simulation that allows him to evaluate the automatic disassembly process. There are many recent publications (Bullinger et al. 1998, Gutierrez et al. 1998, Srinivasan et al. 1999) on this matter. The simulation system implemented uses the characteristics saved in the model generated and it has been made using VRML as observed in the following sections in which we describe each of the phases.

3. Geometric model

In this section we show a proposal for a geometric model of an assembly that allows us to make the automatic disassembly of each of its components. We propose an automatic approach in which the disassembly model is generated automatically from

the CAD design of the assembly. The proposed model, apart from affording the determining of the directions of disassembly of each component, allows us to capture the necessary information for the simulation of the disassembly process. The characteristics of its future implementation in a robotic system will also be added.

The components of the assembly are modelled as a polyhedron. Each polyhedron is formed by a set of flat surfaces. Each surface is associated with a vector $d(A, B, C)$ which is perpendicular to it and is directed outwards from the polyhedron. Given that vector $d(A, B, C)$, a point is required to obtain the complete equation of the plane corresponding to the surface. Then, we also need to have stored the coordinates of one of its vertices. Thus, for each surface a vector and a point will be needed (for a surface i):

$$(\vec{v}_{i1}, p_{i1}) \quad (1)$$

Where the subindex, 1, indicates that it is the initial position of the surface of the polyhedron. Throughout the disassembly, successive transformations will be carried out on the equation (1). After j transformations in the polyhedron, the value of the equation (1) will be:

$$\begin{bmatrix} \vec{v}_{ij} \\ p_{ij} \end{bmatrix} = T \begin{bmatrix} \vec{v}_{i1} \\ p_{i1} \end{bmatrix} \quad (2)$$

Where T represents the transformations that the surface has undergone. The problem to be solved now will be to determine the set of transfers, T , that allows the disassembly of a component.

The entry point for the automatic generation of the geometric model will be a CAD description of the object to be disassembled. This description is transformed to VRML. The VRML description will be the information needed for the automatic generation of the geometric model.

Therefore, the process followed to obtain the geometric model of the assembly is the following:

[Insert Figure 2 about here]

We have developed a model of the assembly for the inclusion of all the characteristics (including the geometric ones) that are implied in the problem of disassembly. The information on each object is represented hierarchically using an Object Oriented annotation (Class, object, method). The attributes of a component may be geometric, physical, relational... Within the geometric attributes, information like location is stored. The location defines the position and direction of the object with respect to the global system of coordinates. Furthermore, the characteristics of the geometric model, as for example, the pair $(\vec{v}_{i1}, p_{i1})$ for each surface, are stored. We store physical attributes, such as the centre of gravity of the objects. The relational attribute refers to the type of contact made between two objects. We also store other information like the surfaces that allow the manipulation of the object, as well as attributes of reachability that provide information to the manipulator.

4. Graph of contacts

Each component of an assembly S is modelled in a polyhedron form. Each polyhedron will also be stored in the form of a graph with a node for each surface, so

that there will be an edge when one surface of the polyhedron is adjacent to another. In addition to the previous graph, we have a graph of contact surfaces between the components of the assembly. This graph has an edge between each pair of surfaces of each component that are in contact in the original assembly. Each edge has an attribute that indicates the type of contact (flat, screw, cylindrical...).

We have joined these two graphs into just one, which represents both bits of information, generating the graph labelled “graph of contacts”. This graph is generated automatically from the CAD description of the assembly. In the Figure 3, we show an example of the graph of contacts.

In the graph in Figure 3, we show the contacts that appear between the different components from the assembly. In the graph of contacts an attribute for each edge corresponding to a contact surface appears. This attribute indicates the type of contact, which in the case of Figure 3 is a flat contact. For the sake of clarity, in Figure 3, the adjacent surfaces between each of the surfaces have not been indicated and we have shown a subgroup of the surfaces of each component of the assembly.

The information represented in the graph of contacts is dynamic. That is to say, the graph will be modified during the course of the disassembly process.

5 Simple movements

In this section, we shall determine the direction of separation of a component with respect to the rest of the assembly. We consider simple directions so that the disassembly is made with just one transfer movement.

5.1 Set of directions for disassembly

Before determining the precise direction for disassembly, we shall present the method for determining the set of directions (SDD) which will allow us to separate the object to be disassembled from the rest of the assembly.

[Insert Figure 3 about here]

If we want to disassemble a component from the rest of the assembly (for example C3 in figure 4), the contact surfaces are the first things that we have to determine. These contact surfaces restrict the directions throughout which the component is to be disassembled. We represent the set of all the possible directions of separation by a Gaussian sphere (Gadth et al. 1991). Each one of the contacts (which can be determined from the graph of contacts) will suppose a cut in the Gaussian sphere, thus limiting the set of separation directions. This way, once all the contact surfaces have been considered, we will have a sphere section that will represent the set of disassembly directions for that component with respect to the rest of the assembly.

In Figure 4 we show an example in 2D to determine the SDD for the component C3.

5.2 Determination of the disassembly direction

In this section we shall determine the disassembly direction (which we shall call DD) between the set of directions represented by SDD.

[Insert Figure 4 about here]

We have a “discrete world”, represented in the form of a cube divided into squares. Each square corresponds to a discrete global coordinate and contains a numerical value with the probability that there is an obstacle in that position. The information contained in the “discrete world” is obtained through the detection of obstacles during the development of the disassembly process. This detection is made for a set of rays coming from the centre of gravity of the object that is being disassembled (similar to a ring of sonars in robotics). A sampling of the trajectory followed by the disassembled object is made so that at given intervals of time the information in the “discrete world” is updated.

The following step consists of the creation of a histogram that shows, for each possible direction of disassembly, the probability that a collision takes place. This histogram is made from the information in the “discrete world” that is nearer the object that is being disassembly. From the cartesian global coordinates of the discrete world, we convert them into spherical local coordinates, so that the value of the squares nearest the object that is being disassembled, accumulates in a histogram. Observing the information in this histogram, we obtain information about the directions that are occupied and with what probability.

In order to determine the direction of the SDD that is selected for disassembly, to each value of the histogram we associate a repulsive force against the object to be disassembled, which is directly proportional to the stored value in the histogram. This force will be inversely proportional to the distance between the obstacle and the object. The repulsion force has the following value:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = -\frac{hist}{\sqrt{x^2 + y^2 + z^2}} \cdot \begin{bmatrix} x/|x| + |y| + |z| \\ y/|x| + |y| + |z| \\ z/|x| + |y| + |z| \end{bmatrix} \quad (3)$$

Where *hist* is the value of the histogram for the cartesian local coordinate (x, y, z).

From the histogram we will also know the free directions of the obstacles. These directions will exert a force of attraction that will be combined with the previous ones of repulsion. Once the direction of resulting force is calculated and verified, the nearest direction within the SDD will be the direction chosen for the disassembly, DD.

6 Determination of the sequence of movements

Up to now we have considered the disassembly process based on simple movements, that is to say, with only one transfer movement. Next, we shall describe the technique used for the disassembly of objects that require more than one transfer movement (n-disassembly). In order to determine these movements the following steps are made:

1. Heuristic determination of the direction for disassembly, DD.
2. Detection of possible collisions. In this section we determine what collisions would be obtained if the object is moved in direction DD. If no collision had taken place we would have already found the correct direction for the disassembly. Otherwise, we would have a distance, *d*, in the direction DD, after which we would have a collision.

3. We determine a distance $d' < d$, so that, once the object has been transferred this distance d' for its disassembly, an update of the “discrete world” is made and we return to point 1. This transfer causes the data stored in the object model to be updated (location, centre of gravity...). It also causes the updating of the graph of contacts.

The succession of the three previous steps will generate the movements of disassembly. We have developed additional techniques for the control of the trajectory. These techniques consist of guaranteeing smooth and non-repetitive movements so that the disassembly is obtained in the smallest possible number of movements.

In the second step for determining the sequence of movements of the disassembly, we have shown the need to have a detection system for collisions. In order to develop this detector the information previously modelled has been used.

In the algorithmic description of the disassembly system that we are going to show, we shall make the following considerations: DD is the direction of movement of the object C obtained from SDD. The algorithm that determines where the object can move to, without any collision taking place, is the following:

For all the surfaces (Pu) of the object to be disassembled C, (where j is the rank of numeration for the surfaces of the object C), whose vector $\vec{v}_{ji}$ forms an angle with DD that is in the rank $]-\pi/2, +\pi/2[$ do:

For all the surfaces (Pv) of the rest of objects that have a director vector who forms an angle with the vector of movement that is in the rank $]\pi/2, 3\pi/2[$ do:

Solve the problem of linear programming by minimizing P_u . The restrictions associated with this problem of linear programming will be the adjacent surfaces to P_u (with a direction that is parallel to DD), P_u , P_v and the surfaces adjacent to P_v (with a direction that is parallel to DD). The restrictions will be obtained from the equations of the previous planes making all of them < 0 (the points included within the surface limited by the plane). The resolution of this problem gives the maximum coordinate to which the P_u surface can be moved without a collision taking place.

The object to be disassembled can be moved up to the minimum of the distances determined in the previous loop without a collision taking place.

7. Simulation

In this section we do an evaluation of the proposed disassembly system. To do so, we show several examples obtained with the proposed system of simulation.

In Figure 5, we show the PC modelled for the tests. If we indicate to the system that it should make the disassembly of the power supply, the system indicates that the previous disassembly of the connection elements is necessary, as is shown in Figure 5.

[Insert Figure 5 about here]

Once the connection elements of the system have been eliminated, it shows the disassembly movements required to eliminate the power supply. We show these movements in Figure 6.

[Insert Figure 6 about here]

Another disassembly example using the proposed system is shown in Figure 7. In this case they are the movements obtained for the disassembly of the CD:

[Insert Figure 7 about here]

8. Conclusions

The main objective of this article has been the determination of the necessary movements for the disassembly of a product. To do so, a method based on virtual prototypes has been developed, which incorporates the principles of “Design for Disassembly”. A system has been developed that allows us to evaluate the ease of disassembly of a product. The information required by the system to generate the virtual prototype is obtained from the CAD model of the disassembled product.

The method designed allows us, apart from obtaining the disassembly sequence, to verify whether this sequence exists and, if not, to indicate to the user or to the scheduling module why it is impossible. Information about the connection elements has been added to the assembly model.

We have implemented techniques for the detection of collisions that allow us to correctly simulate the movement of the objects in a virtual environment. We have also done a study on the structuring and storage of the required information for each component. We have decided on an object-oriented representation of the information required for the determination of the movements for disassembly. Within this latter

information we include the tools and places that allows a manipulator to grasp and do the disassembly.

At the present time we are working on implementing the disassembly using a robot arm that carries out the inferred movements by the system. As such, after simulating and verifying that the disassembly of a component can be done correctly, the robotic arm carries out these movements on the real environment.

9. Acknowledgements

This work was funded in part by the following Spanish CICYT project “Sistema Robotizado de Desensamblado Automático basado en Modelos y Visión Artificial” (TAP1999-0436).

References

BULLINGER, H. and MENRAD, W., SEIDEL, K. 1998. Advanced Assembly and Disassembly Planning Using Virtual Reality Techniques. Proceeding volume from IFAC Workshop pp 7-10..

DOWIE, T., SIMON, M., FOGG, B. 1995. Product disassembly costing in a life-cycle context. Conference Publication 415 IEE pp 202-207.

ECKRTH, G., BÖNKER, T., KREIS, W. 1998. Intelligent Disassembly Systems For Industrial Reusing and Closed Loop Economy. Proceeding volume from IFAC Workshop pp 105 - 110.

GADTH, R., HJALMARSSON, H., PRINZ, F.B., 1991. On the recognition and classification of shape features using MANU-Facture. In Issues in Design/Manufacture Integration. 39, ASME, pp 53 - 58.

GÜNGÖR, A., GUPTA, S. M., 2001. Disassembly sequence plan generation using branch-and-bound algorithm. *International Journal of Production Research*. 39, No. 3, 481 – 509.

GUTIÉRREZ, T., BARBERO, J.I., EGUIDAZU, A., 1998. Virtual Assembly and Disassembly Simulation. Proceeding volume from IFAC Workshop pp 35-40.

HESELBACH, J., FRIEDRICH, R., SCHÜTTE, G. 1994. Automatization in Dismantling of Printed Circuit Boards. Proc of the 2nd Int. Seminar on Life Cycle Engineering, pp. 254 – 266.

KOPACEK, B., KOPACEK, P., 1998. Intelligent disassembly of electronic equipment. Proceeding volume from IFAC Workshop pp 87-92.

KUO., T.C., ZHANG, H. C., HUANG, S. H., 2000. Disassembly analysis for electromechanical products: a graph-based heuristic approach. *International Journal of Production Research*. 38, No 5. 993 – 1007.

LEE, K., MARTIN, M., 2000. Modeling and Supervisory Control of a Disassembly Automation Workcell Based on Blocking Topology. *IEEE Transactions on Robotics and Automation*. 16 No 1. 67 – 76.

SRINIVASAN, H., MO, J., FIGUEROA, R. J., GADTH, R., 1999. Virtual Assembly and Disassembly *Silicon Graphics World Magazine*.

ZHANG, H. C., KUO, T. C., 1997. A Graph-Based Disassembly Sequence Planning for EOL Product Recycling. *IEEE/CPMT Electronics Manufacturing Technology Symposium* pp 140 – 151.

Figure 1: Steps for the disassembly process.

Figure 2: Steps of the assembly model.

Figure 3: Graph of contacts for the interconexion of a card.

Figure 4: Set of directions for disassembly

Figure 5: 5.A (left) PC for the disassembly. 5.B. (centre) Power supply and a connection element, screw. 5.C. (right) Disassembly of a screw.

Figure 6: Disassembly process for the power supply. 6.A (Upper-left) Original position of the power supply. 6.B (Upper-right) Position of the power supply once the first movement is made. 6.C (Lower-left) Position of the power supply once the second movement is made. 6.D (Lower-right) The last movement for the disassembly of the power supply.

Figure 7: Disassembly of the CD. 7.A (Upper-left) First direction for the disassembly. 7.B (Upper-Right) Second movement for the disassembly obtained for the CD. 7.C (Lower-left) Third movement of disassembly. 7.D (Lower-right) The last movement of the disassembly.