Multi-Perspective Compact Explosion Diagrams ☆

Markus Tatzgern[a,*], Denis Kalkofen[a], Dieter Schmalstieg[a]

[a] *Graz University of Technology, Institute for Computer Graphics and Vision, Inffeldgasse 16, 8010 Graz, Austria*

Abstract

This article presents a system to automatically generate compact explosion diagrams. Inspired by handmade illustrations, our approach reduces the complexity of an explosion diagram by rendering an exploded view only for a suitable subset of the assemblies of an object. However, the exploded views are chosen so that they allow inferring the remaining unexploded assemblies of the entire 3D model. In particular, our approach demonstrates the assembly of a set of identical groups of parts by presenting an exploded view only for a single representative. In order to identify the representatives, our system automatically searches for recurring subassemblies. It selects representatives depending on a quality evaluation of their potential exploded view. Our system takes into account visibility information of both the exploded view of a potential representative as well as visibility information of the remaining unexploded assemblies. This allows rendering a balanced compact explosion diagram, consisting of a clear presentation of the exploded representatives as well as the unexploded remaining assemblies. Since representatives may interfere with one another, our system furthermore optimizes combinations of representatives. The optimization process also generates good views on the explosion diagram. Labels are added to the explosion diagram to increase the visibility of small or occluded parts. Throughout this article, we show a number of examples, which all have been rendered from unmodified 3D CAD data.

Keywords: Spatial layout techniques, illustrative rendering, multi-perspective rendering

1. Introduction

Explosion diagrams provide a powerful technique to enable comprehensible explorations of three dimensional objects. While other illustrative exploration techniques, such as cutaways [11] or ghostings [7] remove parts from the presentation, explosion diagrams present all parts entirely opaque and with full detail, by displacing the elements of an objects. The displacements are carefully designed to encode the assembly of the object. Artists have been trained to intuitively choose comprehensible arrangements of parts. In computer science algorithms have been investigated to compute the layout of an explosion diagram automatically. For example, the current state of the art algorithms define relations between parts, which are subsequently used to control their displacement [1, 9].

However, automatically generated explosion diagrams of complex objects can easily suffer from cluttered layouts. While artists intuitively decide which parts of a complex explosion diagram are really necessary, computer graphics applications had to resort to user interaction to control the complexity at runtime [9]. Such interaction requires a certain effort, which increases with the complexity of the 3D model. In addition, traditional media, such as textbooks, do not allow interaction with the presentation at all. Illustrations targeted for non-interactive media still need to present the entire assembly of an object. Moreover, since interaction allows zooming-in and zooming-out, the current approaches neither consider special handling of small or distant parts nor of those which are hardly visible due to an adverse point of view. These characteristics may lead to an explosion layout which is not suitable for a still image.

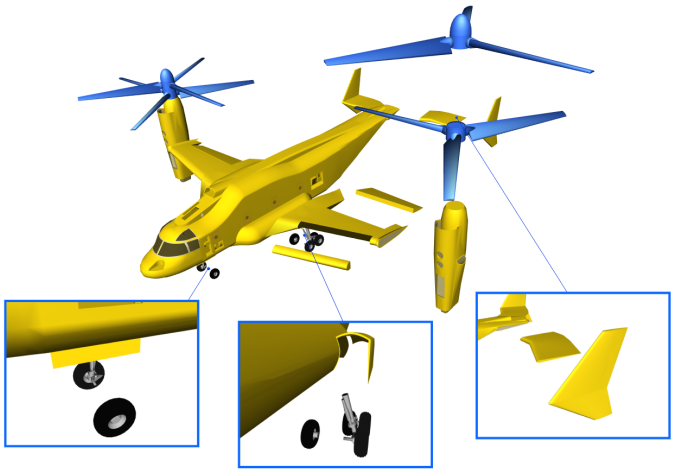
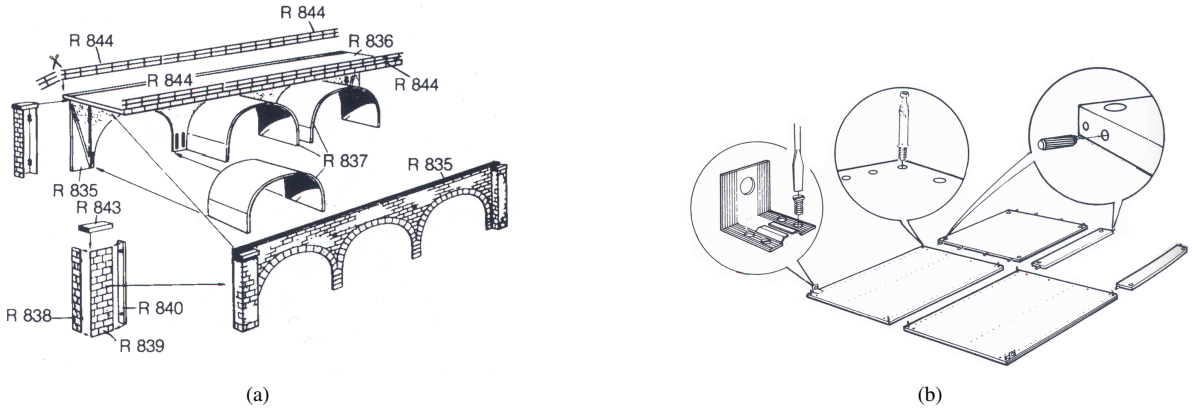


Figure 1: *Multi-Perspective Compact Explosion Diagrams allow to efficiently present the assembly of an object. Explosions of recurring subassemblies are omitted and renderings of hardly visible subassemblies from suitable point of views are added as insets.*

☆ http://dx.doi.org/10.1016/j.cag.2010.11.005

* Corresponding author

Email addresses: tatzgern@icg.tugraz.at (Markus Tatzgern), kalkofen@icg.tugraz.at (Denis Kalkofen), schmalstieg@icg.tugraz.at (Dieter Schmalstieg)



(a) (b)

Figure 2: *Handmade compact explosion diagrams. (a) The assembly of the entire model is presented by a set of representative exploded views only (Adapted from [12]). (b) Small parts do not explode in the main view onto the explosion. Instead, they have been exploded in an additional presentation shown as label (Adapted from [12]. © IKEA. Used with permission.).*

This article presents a system, which is able to automatically reduce the complexity of an explosion diagram. Inspired by handmade illustrations, such as demonstrated in Figure 2a, we reduce the complexity of an explosion diagram by rendering an exploded view only for a subset of the assemblies of an object. The exploded views are chosen so that they allow inferring the remaining unexploded assemblies of the entire 3D model. Note how the illustrator of 2a uses a selective displacement multiple times to render a more compact explosion layout.

Similar to the handmade explosion diagram in Figure 2b, our system is additionally able to increase the visibility of individual explosions of the diagram by rendering those from a more suitable point of view. In order to visually relate the additional renderings, our system presents individual explosions, which were rendered from a secondary point of view, as annotations to the main explosion diagram (Figure 1). In order to avoid clutter, which may appear by introducing additional visual elements as annotations, we reduce the number of labels by sharing labels of similar assemblies (Figure 2b).

Compared to ordinary explosion diagrams, the benefit of our approach can be summarized as follows:

- Due to its compact layout, labeled compact explosion diagrams are able to reduce clutter. Moreover, the integration of multiple points of view within a set of annotations increases the visibility of its individual explosions. Both characteristics enhance the effectiveness of explosion diagrams, especially in none-interactive printed media.
- Compact explosion diagrams provide a space-efficient presentation, which can be used in print media or in applications running on small-screen devices such as mobile phones.

This article presents details on the required components to create a compact explosion diagram. Section 3 gives an overview over the whole system. Section 4 discusses our approach to compute an explosion layout, which ensures that similar parts are exploded in similar ways. Section 5 presents the algorithm to find similar subassemblies, before Section 6 describes the optimization process, which selects appropriate representative groups. Furthermore, different strategies for handling hierarchical subassemblies are presented. Section 7 describes how to detect hardly visible explosions of subassemblies, discusses the computation of more suitable points of view, and their usage as automatically placed annotations.

Section 8 presents the optimization of the main point of view onto the explosion diagram. Section 9 concludes this article with a discussion on the presented techniques and a list of directions for future work.

2. Related Work

Over the past 15 years, computer graphic researchers have investigated a number of different methods to automate the generation of explosion diagrams. These methods create explosion diagrams from many different kinds of data, ranging from 3D-CAD data [15], triangle soups [13] and volumetric data [3] to 2D image data [10]. In addition, a number of different approaches have been presented to automatically compute the explosion's layout. Distortion techniques as presented by Raab [14] scale occluding parts. Force-based techniques as presented by Sonnet [18] and Bruckner [3] use a set of interactively applied repelling and attracting forces, which define directions and distances for offsetting parts. Agrawala et al. [1] and Li et al. [9] use spatial blocking information between parts as well as a size analysis to automatically derive the relations and directions.

To control the visual complexity of an explosion diagram, the existing approaches mainly provide interactive techniques. For example, Sonnet et al. [18] presented an interactive system which moves parts of an object out of the 3D volume of an explosion probe. Bruckner et al. [3] interactively define the amount and the relationships between forces to control the distances, direction and relative movements of parts. Li et al. [9]

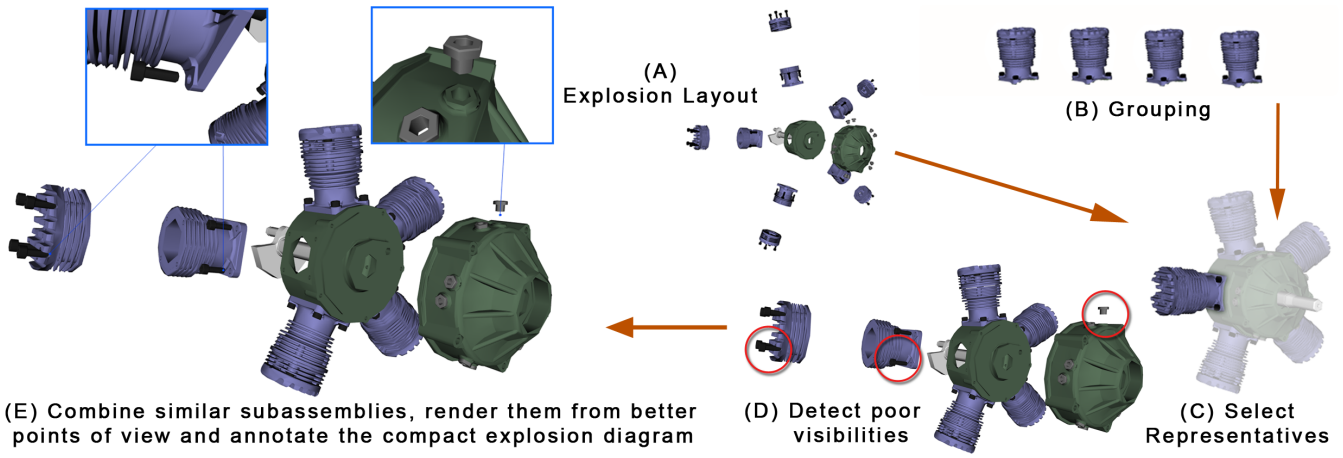


Figure 3: *Abstraction of the modules to compute a multi-perspective compact explosion diagram from a single point of view. (A) By supplying a 3D CAD model, our system automatically computes an initial explosion layout. (B) It is able to detect groups of similar assemblies, and (C) selects an optimized combination of representative explosions of subassemblies. (D) The presented system is furthermore able to detect hardly visible explosions of subassemblies, (E) which it subsequently rendered from a more suitable point of view. To allow for an easy relation of presentations from different points of view of a single subassembly, our system presents the additional renderings as annotations to the compact explosion diagram.*

presented techniques, such as dragging or riffling of parts, to interactively explore a pre-computed explosion diagram, starting from a completely unexploded presentation.

Even though research on rendering of explosion diagrams has often focused on interactive systems, few have investigated an automatic search of groups of parts to simplify the explosion layout. Thus, the works closest to our approach are the systems of Ruiz et al. [16], Kalkofen et al. [8], Agrawala et al. [1] and Niederauer et al. [13]. Niederauer et al. [13] attempt to explode the floors of a building, searching for those triangles which group up to a floor. Since different floors are usually offset at a certain distance and oriented similarly, Niederauer was able to find groups of triangles by applying a statistical analysis of their locations and orientations. Ruiz et al. [16] define the thickness of parallel slabs of a volume, based on a similarity measure between neighboring slabs. The similarity values are computed using mutual information computations. While the former approach only performed well on structures similar to buildings, the latter is optimized for volumetric data.

Explosions of groups of parts of a 3D CAD model have been presented by Kalkofen et al. [8], Agrawala and later Li et al. [1, 9]. While Agrawala et al. and Li et al. manually annotated their models with group information, Kalkofen and his colleagues automatically group elements based on a selected focus element, which they aim to uncover. In a complete AND/OR-Graph data structure, they search for the largest groups of parts which can be displaced from the subassembly containing the object of interest. By recursively applying this search strategy on the AND/OR-Graph data structure, their approach is able to compute a Focus and Context explosion layout with an uncovered object of interest and a minimal number of contextual groups.

Our approach differs from Agrawala et al. [1], Kalkofen et al. [8] and Li et al. [9] in that we do not restrict ourselves to six main explosion directions, but allow all valid removal directions. Furthermore, we employ a sophisticated similarity measure [20] for identifying similar parts, which are then arranged in similar ways in the final explosion layout. Otherwise, it may happen that symmetric structures of assemblies are exploded in different ways (see Figure 5). While Agrawala and Li et al. [1, 9] manually annotate the models with group information, we are able to find similar structures of the assembly automatically and use them to create compact exploded views. Although Kalkofen et al. [8] also group elements automatically, the found groups do not take into account any structural information of the assembly, like for instance similar subassemblies.

This article presents an extended version of a previously published paper [19]. We extended the original version with a more detailed discussion of the impact of the quality parameters used for selecting representative groups. Furthermore, we added a method for creating multi-perspective renderings of poorly represented subassemblies (Section 7) and present an algorithm for automatically selecting good viewpoints on explosion diagrams (Section 8).

3. System Overview

Our system is able to automatically mimic the layout techniques presented in Figure 2a and Figure 2b. Inspired by handmade explosion diagrams, such as illustrated in Figure 2a, our renderings demonstrate the assembly of a set of similar groups of parts by rendering an exploded view only for a single representative. Figure 3 illustrates the main modules of our system which are used to render a multi-perspective compact explosion diagram from a single point of view. In the following, a general

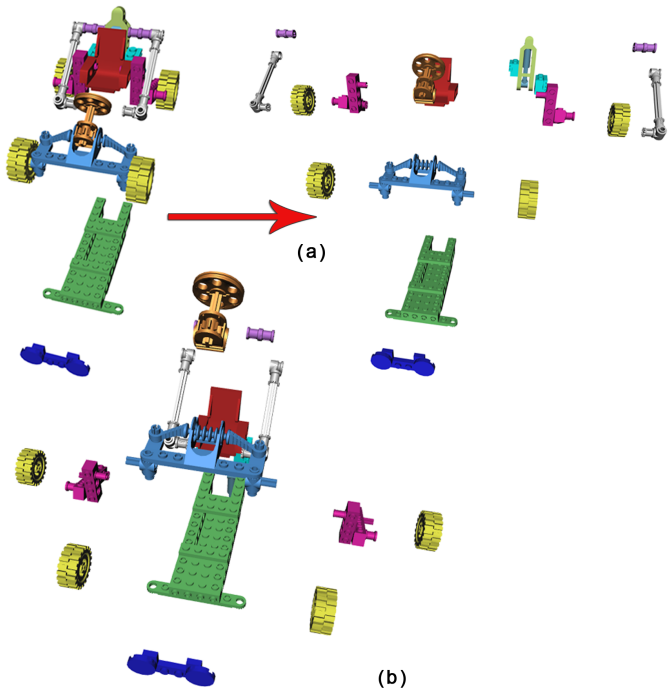


Figure 4: *Different relationships between parts results in different layouts of the explosion diagram. (a) The stacks of parts to the left and the right in the back of the car have been related to the seat. The wheels in the front of the car follow the blue steering gear. (b) The front wheels have been related to the base-plate of the car, as have been both purple elements, which connect the wheels in the back to the car.*

overview over the algorithmic pipeline is given. The subsequent sections of this article contain more detailed descriptions.

Initially, an input assembly is fully exploded (Figure 3(A)) using the method described in Section 4. The presented method identifies similar parts of the assembly by employing the shape descriptor of Vranic [20] and ensures that these similar parts are exploded in similar ways. By performing a frequent subgraph search [22] on a graph representation of the assembly, which incorporates the found similar parts, we are able to extend the similarity measure from single parts to similar subassemblies (Figure 3(B)).

In general, the creation of the explosion layout (Figure 3(A)) and finding similar subassemblies (Figure 3(B)) are independent operations. Therefore, their detailed descriptions are separated into different sections. However, we incorporated the detected subassemblies into the explosion layout generation, to ensure that not only similar parts, but also similar groups of parts are exploded in the similar ways. See Section 5.4 for a detailed discussion on the modification of the initial layout algorithm and on the variations on using and creating similar groups.

From the set of similar groups, we select representatives (Figure 3(C)) depending on a quality evaluation of its potential exploded view, including parameters such as their visibility, their size after 2D projection and the angle between explosion direction and the view vector. Moreover, our system takes into account visibility information of the remaining unexploded assemblies. This allows rendering a balanced compact explosion diagram, consisting of a clear presentation of both the exploded representatives and the unexploded remaining assemblies. Since representatives may interfere with one another, our system furthermore optimizes combinations of representatives using the approach of threshold acceptance [4].

To furthermore increase the effectiveness of the explosion diagrams, our system detects hardly visibly explosions of subassemblies (Figure 3(D)) and renders these from more suitable points of view (Figure 3(E)). Similar to the presentation technique illustrated in Figure 2b, we relate the additional renderings as annotations to their corresponding subassemblies within the main presentation of the explosion diagram. In order to decrease the amount of additional visual elements, we combine similar subassemblies, which are poorly presented, and we use a single annotation for multiple subassemblies. The additional renderings are presented as annotations of the compact explosion diagrams using the technique of Ali and Hartmann [6].

To render the explosion diagram from the most effective point of view, we compute an explosion diagram from viewpoints lying on the bounding sphere of the disassembled object. By comparing the aforementioned quality parameters of each viewpoint, we are able to select the one with the highest score. However, like Blanz et al. [2], we favor points of view which face the object from an elevation below 45° and an azimuth angle which is less than 45°.

4. Initial Explosion Layout

The layout of an explosion diagram depends on the removal direction and distance chosen for each part, which set it apart from its initial position. To reduce the mental load to reassemble an exploded object, explosion directions often follow mounting directions, therefore collisions between displaced parts are avoided. Explosion diagrams implement this feature by introducing relations between the parts of an assembly. For example, each screw in Figure 3 moves relative to the purple cylinder, which it fastens. By displacing one of the purple cylinders, the corresponding black screws will be displaced implicitly.

The relationships between parts of an explosion diagram also allow parts to follow related parts. This enables a part to move relative to its initial location in the assembly, which also reduces the number of mental transformations to reassemble the object. For example, note how the grey bolts follow the green gearbox in the explosion diagram in Figure 3. However, it is often not obvious which part represents the initial location of another part best. For example, while the initial locations of the black screws in Figure 3 are clearly defined by the holes of the engine they fasten, the initial location of the wheels in Figure 4 is surrounded by a number of parts. As demonstrated in Figure 4a, the wheels in front of the car may follow the blue steering gear. This will result in a translation along the up-vector of the car, before the wheels explode along the x-directions of the model's

coordinate system. In contrast, the explosion diagram in Figure 4b uses a relation between the wheels in the front of the car and the green base-plate. This results in a displacement of the wheels without a translation along the up-vector of the coordinate system. The same behavior appears for a stack of parts in the back of the car. Since in Figure 4a the parent of the stack follows the red seat of the car, all the parts between the wheels and the seat have been moved along the up-vector before they have been separated from each other. In contrast, the explosion diagram in Figure 4b uses a relationship between the parent of the stack and the green base-plate of the car, which reduces the number of translations of all the elements in the stack.

4.1. Disassembly Sequence, Part Relations and Explosion Directions

We define relations between parts by computing a disassembly sequence. A relationship is set up between each exploded part and the biggest part in the remaining assembly it has contact with. To avoid collisions between exploding parts, the directions in which a part can be displaced are restricted to only those in which a part is not blocked by any other parts. This implies that the algorithm displaces parts which are unblocked in at least one direction, before it is able to explode parts which are blocked in all directions. Thus, by removing the exploded parts from the assembly, we gradually remove blocking constraints which allows us to explode previously blocked parts in a subsequent iteration of the algorithm. Since the algorithm gradually removes parts from the assembly, the set of directions for which a part is not blocked (and thus the set of potential explosion directions) depends on the set of previously removed parts. Consequently, the disassembly sequence directly influences the set of potential explosion directions.

Disassembly Sequence. Previous approaches [1, 9] compute a sequence depending on how fast a part is able to escape the bounding box of the remaining parts in the assembly. However, since this approach does not comprise any information about the similarity between exploded parts, the resulting explosion layout does not ensure similar exploded views for similar assemblies. Consequently, we encode information about the similarity of the parts in the sequence. We remove similar parts in a row, starting with the smallest. If no similar part can be removed from the assembly, we choose the current smallest part. This strategy enables us to set up relationships which subsequently allow smaller parts to follow bigger ones during explosion. Take note that, by computing a larger amount of similar explosion layouts, our system is able to choose a representative exploded view out of a larger set of similarly exploding assemblies.

Figure 5 demonstrates the difference between previous approaches and our new strategy to find a disassembly sequence. A sequencing based on a bounding box intersection is demonstrated in Figures 5a to 5c. The algorithm first removes part A before part B and part C will be exploded. By using this strategy, relationships between part A and part B and subsequently between part C and part B will be set up. The resulting explosion layout is illustrated in Figure 5c. As can be seen, different

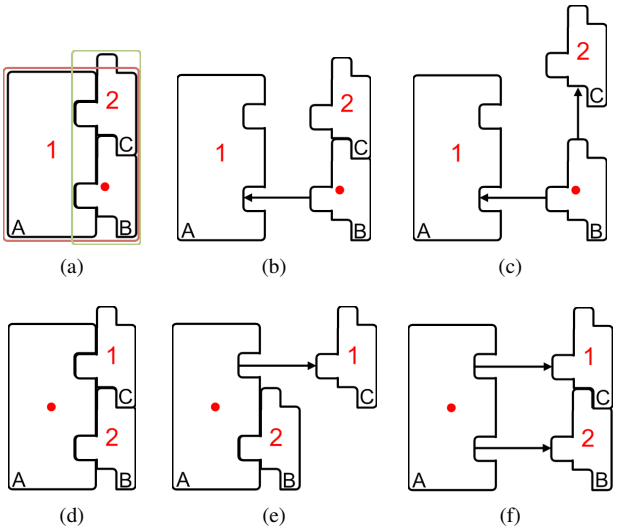


Figure 5: *Different disassembly sequences may result in different layouts. The sequence is labeled in red. The resulting explosion diagram is illustrated in the image on the right. (a,b,c) The computed sequence is based on previous approaches which select parts depending on the distance a part has to be moved to escape the bounding of the remaining assembly. The bounding boxes of the remaining parts have been framed in red and green. (d) We compute the next element in the sequence based on a comparison with the previous one. (e) By removing similar parts in a row we ensure that the remaining assemblies contain the same elements, except for one part which is similar to the next one. (f) This strategy allows us to explode similar parts within similar conditions, which in turn results in more similar exploded views of similar subassemblies.*

explosion directions have been assigned to the similar parts B and C.

In contrast, our algorithm computes a sequence which is based on a comparison of the previously exploded part and all removable part in the remaining assembly. As demonstrated in Figures 5d to 5f, our strategy will result in a sequence which supports similarly exploded views of similar assemblies. Both parts B and C have been displaced in the same direction and both parts have been related to the same part in the remaining assembly (part A).

Relationships. Both strategies in Figure 5 set up relationships between the current part and the bigger one. However, since our sequence removes similar parts one after the other, the remaining assemblies are identical for similar parts, with the exception of the previously removed part (which is similar to the current one). Since almost identical conditions exist for similar parts, our algorithm is able to set up similar relationships for those parts and the parts in the remaining assembly.

In addition to the initial assignment of relationships between parts, we change the relationships for penetrating elements in a stack. For example, the black screws in Figure 3 have contact with the purple cylinder and the green gearbox. Since the green

gearbox is the bigger item, the initial relation is set between a screw and the gearbox. However, this would result in an explosion diagram in which the screws follow the gearbox instead of the purple cylinder.

To handle such cases, we search for stacks of parts by searching for the elements which are located between the exploded part and the one it is related to. If parts exist in-between and if these parts share an explosion direction with the currently removed part, the initial relationships are changed so that the exploded part is related to the closest part in the stack of parts in-between.

Explosion Directions. Previous approaches compute the explosion direction of a part out of a set which contains only the six directions along the three main axes of the model [1, 9, 8]. However, this approach is very limited (e.g. consider the differences in directions in the explosion diagram in Figure 3). Therefore, we compute a non-directional blocking graph, similar to the algorithm proposed by Wilson [21], by computing blocking information between all pairs of parts. For each exploded part, we determine the set of unblocked directions by removing all blocked directions from the set of exiting 3D directions. We represent all directions by a unit sphere and we remove blocked ones by cutting away the half sphere with a cutting plane which is perpendicular to the direction of a blocking part. By iteratively cutting the sphere, using all locking information from parts in contact with it, the remaining patch of the sphere represent all unblocked directions for a part. Thus, we output the center of gravity from the remaining patch of the sphere.

4.2. Explosion Distance

If a subassembly appears multiple times in another subassembly, we introduce a hierarchy of subassemblies from which we choose representatives depending on an explosion style (see Section 6.3 for a discussion on hierarchical subassemblies). For example, the screws in Figure 3 form a cluster of screws which depends on the part they fasten. Each cluster consists of four screws which fasten a single cylinder. If a style is chosen, which explodes all screws in a single cluster, we have to compute a representative out of a higher level group of parts. Therefore, our system has to support an alignment of the distances of similar parts.

Since similar parts appear to be similarly large, we set the distance of displacement from the parent part to be proportional to the size of the exploded part. Nevertheless, since a linear mapping may easily result in overly large displacements, we introduce a non-linear mapping using equation 1.

$$Distance = SizeOfPart \cdot (1 - k \cdot RelativeSize^2) \qquad (1)$$

For parts which cannot be removed at all, we compute a distance for which they can be moved until colliding with other parts. For example, the lower purple cylinder in Figure 3 cannot be removed before the black screws have been removed. However, the black screws will collide with the cylinder they fasten if we explode them into a single direction. Nevertheless, we can explode the screws a certain distance before they collide with the cylinder. Since this distance is sufficient to reveal the screws, we compute the maximal distance they can be exploded. We explode the screws to a distance smaller than this maximal distance and are further able to subsequently explode the cylinder from the assembly.

We compute the maximal distance that a globally locked part can be moved by rendering both parts - the one which is about to be removed and the one which blocks its mounting direction- into a texture. We position the camera at the vector along the explosion direction to point at the exploded part. In a vertex shader, we use the current model-view transformation matrix to transform each vertex into camera space. The corresponding fragment shader finally renders the location of each fragment in camera coordinates into the textures. By calculating the difference between the texture values, we get a map of distances between the fragments of both parts. The maximal distance a part can be removed, before it collides with the blocking part, is finally represented by the smallest difference between the values in the texture.

5. Grouping

We determine sets of similar subassemblies by performing a frequent subgraph (FSG) search on a graph representation of the assembly. The implemented approach is based on the gSpan algorithm of Yan and Han [22], which uses depth-first-search (DFS) codes to differentiate between two graphs. A DFS code describes the order in which parts of a subgraph have been visited. Two graphs are isomorphic if their DFS codes are equal and if their corresponding node labels (which represent the parts) match. By using DFS codes and node labels, the implemented FSG algorithm finds non-overlapping sets $S = \{G_1, \ldots, G_k\}$ of the largest subassemblies G contained in the graph.

5.1. Graph Representation of Assembly

The FSG requires the 3D model to be represented as a graph A_g, which contains all parts $P = \{p_1 \ldots p_n\}$, with n being the amount of parts in the assembly. The parts of the assembly p_i (with $i = 1 \ldots n$) are mapped to an equal number of nodes of the graph. Undirected edges are created between nodes, when their corresponding parts are in contact.

Nodes of parts, which are similar to each other, receive the same label. We detect similar parts by exploiting the DESIRE shape descriptor of Vranic [20]. The descriptor computes a feature vector for each part which we use to compare their shapes with. We consider two parts as being similar, if the l2-distance of their corresponding feature vectors falls below a certain threshold and if the part sizes match. The result of the part comparison is a list of disjoint sets of similar parts $P_s = \{p_g, \ldots, p_h\}$, for $g \neq h$, and $g, h \leq n$, which is used to label the nodes of the graph A_g.

5.2. Frequent Subgraph Mining

Input to the algorithm is the whole graph A_g. Initially, all nodes having a label which occurs only once in the graph are removed. These nodes represent parts, for which no similar parts exist ($|P_s| = 1$). For each remaining set of similar parts P_s one set S_0 is created, containing $|P_s|$ number of groups G_0, each containing a single part $p \in P_s$. The sets S_0 define the nodes at which the FSG search will start execution.

A recursive FSG mining procedure is applied on each of the sets S_0 and iterates through all input groups G_i of an input set S_i, in order to grow the groups G_i to create similar groups of parts. In each iteration, a different group G_i is chosen from S_i to be the reference group G_r. For the current group G_r the set of neighbors N_r is retrieved for the node which was added last to the group G_r. If all neighbors of the node added last have been processed, the neighbors of the previously added nodes are chosen. If all neighbors have been visited, the group G_r cannot be extended further.

For each other $G_i \neq G_r$ the neighbors n_i similar to the ones in N_r are determined. Neighbors n_i are similar to each other if their labels and number of contact parts to the corresponding group G_i are equal to the ones of the neighbor n_r. Furthermore, the DFS codes and labels of the contact nodes contained in the groups must be equal. This similarity measure ensures that the found groups contain nodes, which have been visited in the same order and which have equal relations to their neighbors. After identifying similar neighbors for at least two groups G_i and G_j during the same iteration, a new set S_n is created. The new set contains the groups $G_{n1} = G_i \cup n_i$ and $G_{n2} = G_j \cup n_j$, which are the original input groups extended by the similar neighbors. All groups for which similar neighbors exist are extended in the same way. Note, that for each set of similar neighbors a new set of groups is created and these groups differ only by one part from the groups of S_i. Hence, by recursively calling the mining procedure on the new sets, a DFS is performed, growing these groups further.

All groups G_i which have been extended by a neighbor are removed from the input set S_i, because these groups are then part of larger groups G_n. If $|S_i| \leq 1$ for a set S_i, all groups were extended and the set is deleted. However, the mining algorithm is applied again to any parts left in the set S_i (if $|S_i| = 1$) to eventually extract smaller similar groups.

The FSG mining returns with the sets S_{out} of largest similar groups G_{out}. If the sets S_{out} do not overlap, the algorithm is finished. Otherwise, overlapping output sets must be resolved by keeping only one of the overlapping sets S_{out} and applying the FSG again to the set of $A_g \setminus S_{out}$. This operation is repeated for all results, until the output sets S_{out} do not overlap anymore. The decision on which of the overlapping groups is kept, is based on the following rules. We keep the one overlapping set which contains the groups holding the most number of parts. If this measure is ambiguous, the set having the most groups is preferred. If this is still ambiguous the one containing the largest part is chosen.

5.3. Detecting Hierarchies

After applying the FSG search to the graph A_g of the whole assembly, a list of sets which contain the largest available non-overlapping subassemblies has been discovered. However, the selected subassemblies may even contain other frequent subassemblies. If we also identify these subassemblies we are able to select a representative in multiple levels of the hierarchy, which in turn allows us to further reduce the number of displaced parts in a representative exploded view (see Section 6.3). To find frequent subassemblies within a previously determined subassembly, we apply the FSG algorithm recursively until no subassembly can be determined anymore. When performing the FSG search on a set S of groups G, each group G is considered to be a separate graph to be mined for subassemblies. This means that a subsequent FSG search does not exceed the limits of the groups they are applied to.

By recursively applying the FSG search algorithm to a subassembly we retrieve a hierarchy of frequent subassemblies. The groups of the detected sets and subsets are similar to each other, because their graph representations are isomorphic. However, subgroups of the same set may have different neighborhood relations to the group they are contained in. The reason for this is that the FSG mining algorithm removes all parts from the input graph, which do not have similar counterparts (for which only one label exists in the graph). Basically, this removes the contacts between any subgroups and the group they are contained in. By recovering this information, we are able to refine the hierarchy. This refinement allows us to choose better representatives from a set, because similar groups are then also distinguishable by their neighborhoods. Therefore, we define that similar subgroups G_l not only must be similar in terms of graph isomorphism, but also the neighborhood to the groups G_h they are contained in has to be similar. We implemented the following algorithm, which searches for similar neighbors of groups of a set.

For each neighbor of a group the set of adjacent groups E_n is determined. Sets E_n of similar neighbors in different groups G_h are merged into the set E_s. Then, simple set operations are performed on the sets E_s to retrieve the common neighborhood for similar groups. For a representative E_r from the sets of E_s, the following operations are performed in combination with each other E_s. First, the intersection $E_c = E_r \cap E_s$ is created. If $|E_c| = |E_r|$, all groups share the same neighbor and the algorithm continues. Otherwise, the groups of E_r share different neighbors. These groups are eliminated from E_r ($E_r = E_r \setminus E_c$). The algorithm continues until either all E_s have been considered, or $|E_r| = 0$. Those groups left in E_r have similar neighborhoods. The algorithm finally terminates when all sets of E_s have been considered as representative set E_r.

5.4. Group-Based Layout

Our system calculates similar subassemblies independent from the initial layout of the explosion diagram. However, even though our sequence generator specifically supports similar exploded views of similar subassemblies, if the neighborhood of both differ, the exploded views may be different. For example, the model in Figure 6a consists of one set of four similar

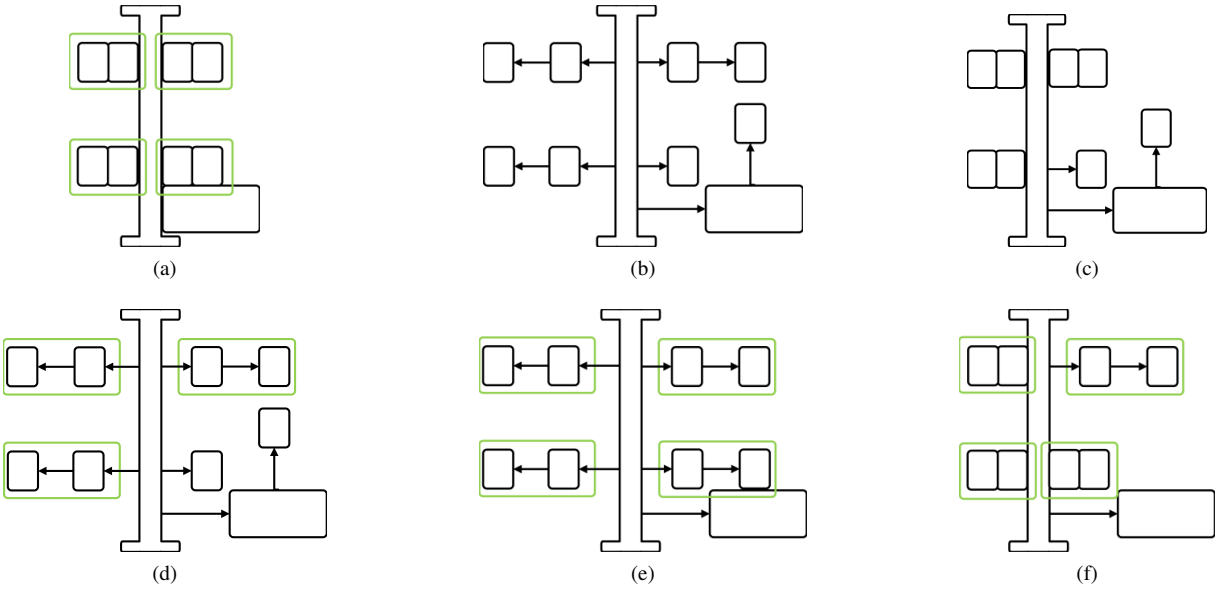


Figure 6: *Explosion Layouts containing group information. (a) Groups have been created independent from the explosion layout. (b) Therefore, the explosion layout does not take information about similar subassemblies into account. This may generates different exploded views of similar subassemblies. (c) If we select a representative out of a set of similar subassemblies which do not explode the same way the explosion does not demonstrate all other subassemblies. (d) By recalculating group information from the layout the number of similar groups is reduced which results in more exploded views. (e) Therefore, we modify the initial layout so that similar subassemblies explode in a similar way (f) This strategy allows us to choose a representative out of a larger set of subassemblies which in turn reduces the amount of required exploded views to demonstrate the assembly.*

subassemblies (marked by the green rectangle). Each of them contains two parts. Figure 6b shows its explosion diagram in which each single part has been displaced. As can be seen from the initial layout, the exploded view of the subassembly in the lower right corner is different from the explosions of the other subassemblies. If we choose this exploded view as the representative of its set of similar subassemblies, the resulting compact explosion diagram lacks a presentation of the other subassemblies of this set (Figure 6c).

To prevent representatives which explode differently to other similar subassemblies, we can adjust the sets of similar subassemblies in a way that only similarly exploding subassemblies will be grouped together. Therefore we use the layout information to modify the identification of similar subassemblies. Only those parts of the assembly are candidates for extending a group which would set up a relationship to another part in the subassembly. Figure 6d shows the result of this restriction. This strategy finds a set of only three instead of the previously identified four similar subassemblies (marked in green). Consequently, less subassemblies will be presented assembled which results in a layout which is not as compact as in the previous case.

In order to create a more compact explosion layout, without risking to choose a representative which does not demonstrate the composition of other similar subassemblies, we modify the layout of the explosion diagram instead of the information about the similarity of subassemblies. As illustrated in Figure 6e we aim to modify the layout to prevent relationships with parts outside the subassembly. We allow only one relationship between a part in the subassembly and the remaining 3D model.

This is similar to the approach of Li et al. [9] who explode a manually defined group of parts as if it was a single element in the assembly. However, we use a different approach to handle interlocking groups. Rather than splitting a subassembly, we ignore blocking parts. This allows us to keep subassemblies connected. Note, this could be at the cost of explosion diagrams which are not completely free from collisions. Nevertheless, we believe that preventing such collisions is less important for the final compact explosion layout than a larger amount of explosions or a representative which does not demonstrate the composition of its associated subassemblies. In the case of a compact explosion diagram, it is more important to select a representative from a rather large set of similar subassemblies, which additionally all explode in a similar way.

Thus, we compute an explosion diagram which ensures similar explosion layouts of similar subassemblies as explained in section 4. However, for each part p we determine if it is a member of a subassembly G which occurs multiple times in the model. If the algorithm is about to explode a part p which is a member of G, we choose a representative part p_r out of G which we explode instead of p . We define p_r as the biggest part in the subassembly G which has at least one face in contact with at least one part of the remaining assembly, not considering other parts of the subassembly. In addition, the representative part p_r

has to be removable in at least one direction without considering blocking constraints of parts of the same subassembly.

Even though p_r influences the explosion direction of the entire subassembly, we may not set the relationship between p_r and a part out of the remaining assembly. Since we are only able to explode each part once and since we want to further continue to explode all frequent subassemblies in the same way, we have to choose the same part in each subassembly to set up the relation to the remaining assembly. Moreover, since we want to explode subassemblies using the guidelines presented in section 4, we want to explode the small parts before the bigger ones. Therefore we choose the biggest part in the assembly as the main part of the assembly and we relate it to the biggest part in the remaining assembly which the subassembly has contact with.

If frequent subassemblies exist in an exploded subassembly we cannot simply search for the bigger part in the main subassembly, because we also want to create a similar exploded view of all frequent subassemblies, even if they appear cascaded. Instead, we first compute a hierarchy of subassemblies (see Section 5.3) before we choose the biggest part from only the highest level of the hierarchy. The highest level ensures that no other part is similar to the chosen one and consequently no conflicting explosion layout can result. Note once again, by removing entire subassemblies in an unblocked direction of a single representative member we ignore collisions between parts during explosion. Even though this may result in physically incorrect sequences to disassemble the object, we are able to explode subassemblies independent of the overall model, which in turn enables to calculate a single explosion layout for all similar subassemblies.

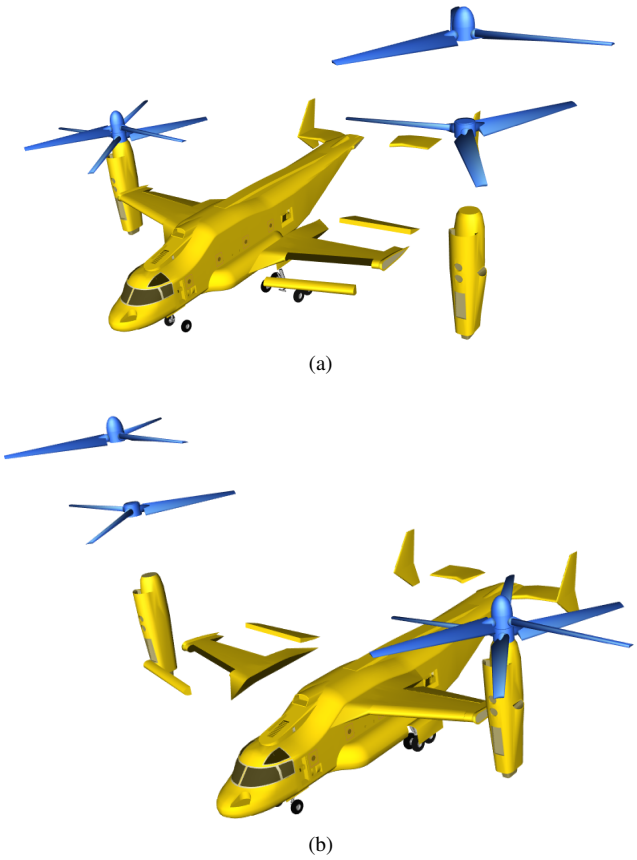
6. Selecting Representatives

After identifying frequent subassemblies and after computing an initial explosion layout, a compact representation is created by displacing only one representative group out of a set of similar groups. We compute the impact of a representative subassembly on the explosion diagram by calculating its quality as the weighted sum of a set of measurements (Section 6.1). Since the combination of representatives may influence the quality of a single subassembly, we optimize the selection based on the idea of threshold accepting [4] (Section 6.2). In the following, we will first describe the parameters for rating the impact of a subassembly, before we present our approach to combine representatives to the final compact explosion diagram.

6.1. Quality Measurements

We define the quality of a group of parts as a combination of several measurements. Therefore, for each group we render its local explosion (which displaces only the parts of the group and parts which block the group) and we compute the following values:

- *Size of footprint of the exploded group.* The size of the footprint f describes the size of the projected area of a part of the object in screen space.



(a)

(b)

Figure 7: *Local Footprint. (a) Emphasis on the footprint of the exploded representatives renders them in the foreground of the presentation (b) In contrast, by putting emphasis on the footprint of the unexploded parts the exploded representatives are shown in the background.*

- *Size of footprint of all other similar groups without any displacements.* The size of the footprint of the parts of other subassemblies f_r describes how big similar, but unexploded subassemblies will be presented.

- *Explosion directions relative to current camera viewpoint.* Assuming that explosions, which are similar to the viewing direction, are more difficult to read than those which explode more perpendicular to the viewing direction, we compute the dot product a between the viewing vector and the explosion direction for each part. The average value of all values a is used as the value for a group of parts within a subassembly

- *Visibility of parts of the exploded representative.* The visibility v is a relative measure. By counting visible pixel of a part and those which are hidden we compute its percentage of visibility from the current point of view.

$$Q_r = f \cdot f_c + v \cdot v_c + (1 - a) \cdot a_c + f_r \cdot f_{rc} \quad (2)$$

The final quality Q_r of an exploded view of a subassembly consists of the weighted sum of these values (see Equation 2).

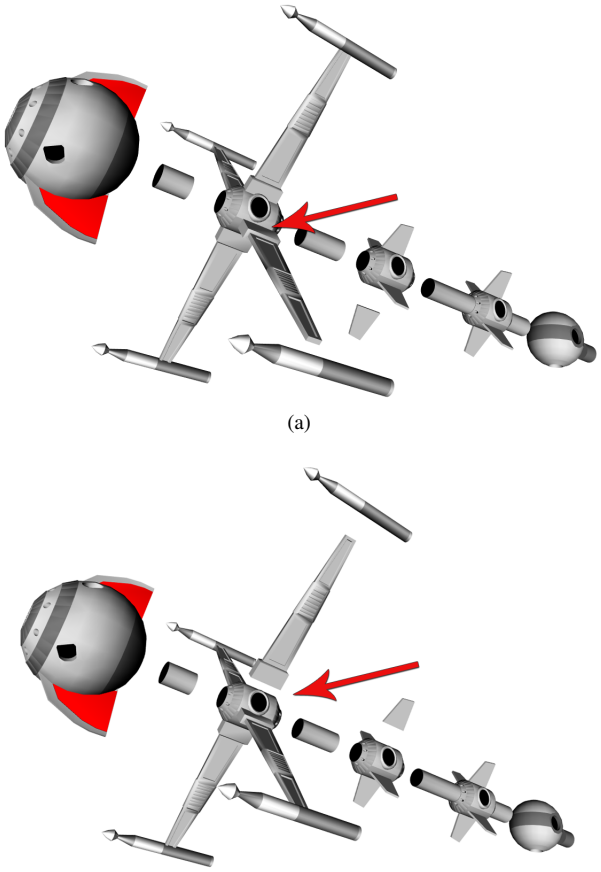


Figure 8: *Direction versus Size of Footprint. (a) The size of the footprint by itself does not ensure a clear presentation of the explosion. (b) By scaling up the importance of the angle between the viewing direction the explosion direction we are able to choose a representative which better demonstrates the explosion of the subassembly.*

The weights (f_c, v_c, a_c, f_{rc}) indicate the importance of each single parameter to describe the quality of the group. By differently scaling these parameter we are able to control the final presentation. For example, the Figure 7 shows two compact explosion diagrams generated with different intensions of the user. The compact explosion diagram in Figure 7a puts emphasis on the representative explosions while it simultaneously shows similar subassemblies in the background as contextual information. In contrast, the image in Figure 7b presents the assembled parts of the compact explosion diagram in the foreground while the exploded representatives are used to fill in contextual area. Both graphics were rendered by scaling up a single weight. While Figure 7a scales up the impact of the size of the footprint of the representatives, Figure 7b was generated by increasing only the values of the impact of the footprint of non-representatives.

Even though the footprints of both, the representatives and the unexploded elements are important parameters to compact explosion diagrams, they may fail to create easily comprehensible presentations. The explosion diagram in Figure 8a was rendered with a high impact of the footprint of representatives. However, such scaling by itself turns out to insufficient from certain points of view. The explosion indicated by the red arrow is almost hidden. A more informative explosion diagram from the same point of view is shown in Figure 8b. The presentation scales up the impact of the angle between the view vector and the average direction of explosion for each representative.

Nevertheless, a high impact of only the explosion directions leads to self occlusions which again may hinder the understanding of the final presentation (Figure 9a). As demonstrated in Figure 9b, by putting emphasis on the visibility of representative parts, the system chooses a different one to explode. However, even if self-occlusions are avoided within a single representative, global occlusion between different representatives cannot be controlled by this parameter (see Figure 10a).

As the examples in Figures 7 to 9 demonstrate, there is no universal rule on which parameter we have to scale up or down in order to ensure comprehensible compact explosion diagrams. However, the weights can still be used to direct the rendering towards the users intention. The quality of the entire compact explosion diagram can only be controlled by taking combinations of explosions of representatives into account. By estimating the quality of an explosion of subassemblies independent from other explosions in the diagram, interdependent explosions and visual overlaps of representatives may change the quality of a representative explosion.

6.2. Combining Representatives

Finding the optimal representative for one set of similar groups, as shown in the previous examples, does not ensure that the representative stays optimal when representatives of other groups are exploded. Exploded groups may interfere with each other and therefore decrease the quality of other representatives. In Figure 10a the representative groups are locally optimal when only these groups are exploded for themselves. However, combining all locally optimal representatives into one explosion significantly decreases the overall layout quality.

To avoid interferences of representatives with each other, we search for an optimal combination of exploded groups using the idea of threshold accepting [4], a heuristic optimization strategy. In each step of the algorithm, the quality of a combination of representative explosions is evaluated by computing the sum of their scores after exploding all of them. The initial layout consists of exploded representatives with the highest local scores. Therefore, if the sum of their local scores is equal to the global score, the local representatives are global representatives too. Consequently, we do not search further for a better combination. However, if the global score is less than the sum of local scores, we change the initial layout by a single representative group and re-compute the global score of the modified layout. If the score of the changed layout is higher than the current best finding, this new one is used as the current best combination of representatives. Therefore, if the new score is equal or less than the current best score, we do not consider the current combination to be displayed. However, even if the current score is less than the best one, we compute the next tested layout based

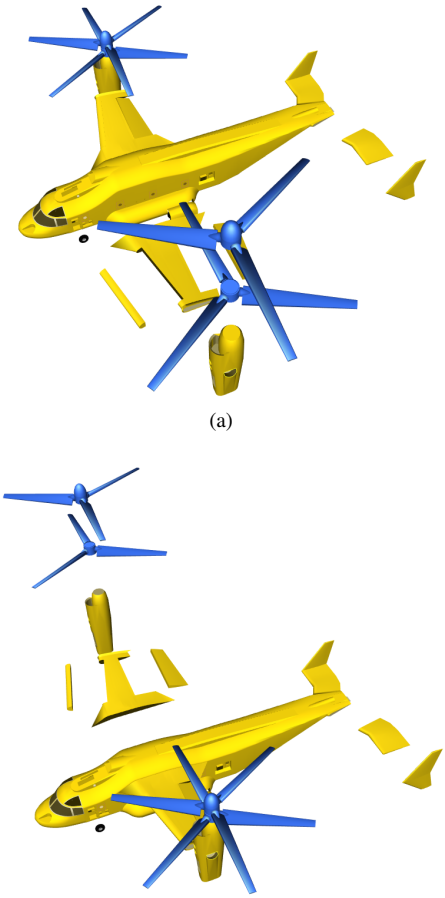


Figure 9: *Visibility (a) If the visibility of the parts of an explosion are not taken into account, parts of a representative may occlude each other (b) The visibility evaluation of representatives is able to resolve self occlusions.*

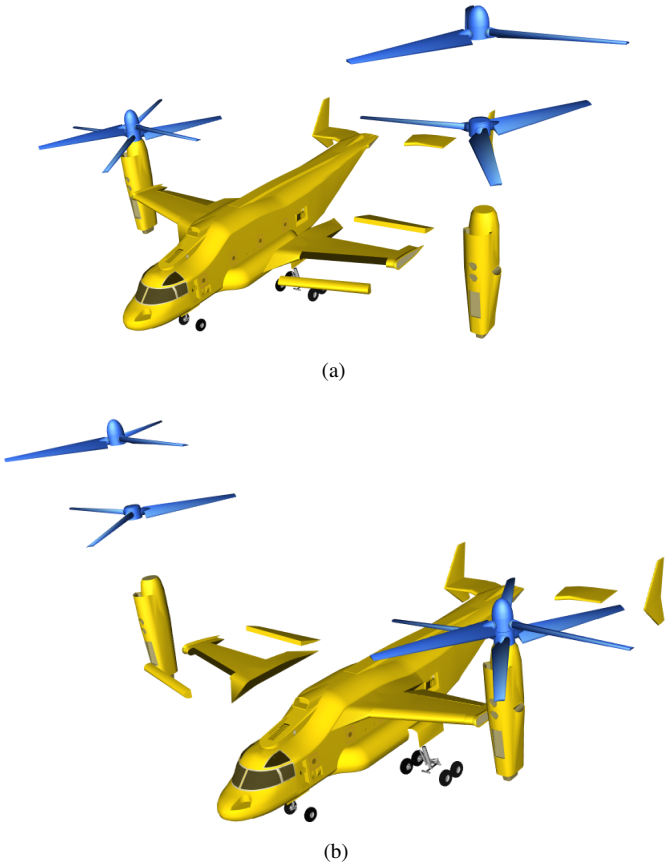


Figure 10: *Global Visibility: (a) Adding weight to the local visibility does not resolve occlusions between different representative groups. (b) Optimizing the layout using threshold accepting ensures that the overall visibility and thus the layout quality is maximized.*

on the current one, if its difference to the best score is less than a threshold value. Otherwise, we modify the layout which the current layout was computed from. While the algorithm progresses, the threshold value decreases, which gradually allows better layouts to be the starting point for further changes.

Figure 10b shows the results of optimizing the locally scored compact explosion diagram presented in Figure 10a. Since representatives in Figure 10a overlap each other, a different subassembly has been selected in the optimized compact explosion diagram in Figure 10b.

6.3. Hierarchical Subassemblies

If a hierarchy of groups exists (see section 5.3), we allow to select representative exploded views using three different strategies. We allow either to choose the representative parts from a single subassembly (Figure 11a, Figure 11b), or to select representative parts independently in different subassemblies of the same set (Figure 11c). If we chose to restrict the explosions to a single hierarchy, we have to decide if we want to explode the entire subassembly (Figure 11a) or only a single representative in each level of the hierarchy (Figure 11b).

Figure 11 shows an example for each given situation. Since it is an open question which strategy results in the perceptually best results, our system allows selecting a strategy at runtime. The strategy shown in Figure 11b seems most reasonable, as it reduces the number of exploded parts compared to Figure 11a, while representative parts are not scattered over the layout as in Figure 11c. We leave a perceptive evaluation of the comprehensibility of each strategy for future work.

7. Multi-Perspective Presentation of Subassemblies

Even though the optimization process selects the best combination of representatives, some of the subassemblies may still be presented in a very small scale or highly occluded. We compensate for these problems by rendering poor explosions of subassemblies from a more suitable point of view. The renderings from secondary points of view allow to zoom small parts as well as to resolve occlusions, which appear from the main point of view.

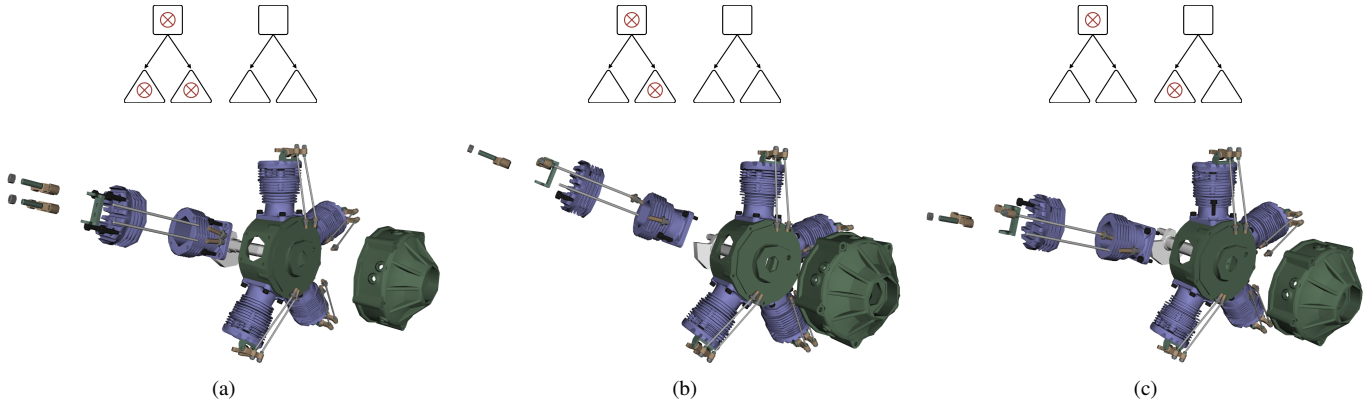


Figure 11: *Selection Strategies in Hierarchical Groups. (a) All parts in a subassembly have been exploded. (b) Representatives have been selected in each level of the hierarchy. (c) Representatives have been selected in different subassemblies.*

7.1. Detecting Poor Explosions of Subassemblies

Our system is able to determine poorly presented parts of the explosion diagram by analyzing the final combination of representatives. The system evaluates each quality parameter of a representative individually and creates renderings from a secondary point of view if one of them falls below an adjustable threshold. Since the footprint of the unexploded elements can be neglected for a rendering from a secondary point of view, we scale down the impact of this parameter by lowering its threshold to the minimum. However, even though the detection of poor explosions on the final rendering allows us to increase the effectiveness of the compact explosion diagram, we select poor elements of the representation independent of representatives. In consequence, we do not generate an optimal presentation with respect to the visibility of representatives.

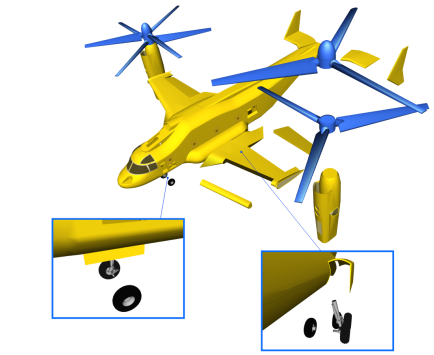
Our system detects poorly presented parts during the selection of representatives and integrates the identification of candidates for a secondary rendering into the overall layout optimization process. In each iteration of the algorithm, which evaluates a new combination of representatives, our system analyzes the visibility and the projected size of the explosion of every single subassembly. If any of the evaluated parameters falls below an adjustable threshold, we exclude it from the quality calculation of the current combination of representatives. This strategy results in a quality value, which represents only the relevant parts of the explosion diagram, but not those which will be presented from a more suitable point of view in a later stage in the rendering pipeline.

By integrating the selection of poorly visibly explosions of subassemblies into the combination of representatives, we exclude poorly presented subassemblies from the layout evaluation. Consequently, the final combination will be better for the representatives which are not presented from a secondary point of view. Another advantage of this approach is that it allows us to control the number of secondary points of view and thus to avoid clutter due to an excessive number of insets. However, the downside of this approach is that the visibility of the already poorly presented subassemblies may become even worse (Figure 12a). Mentally relating secondary points of view for such cases may become very difficult, especially if the subassembly is completely occluded in the compact explosion diagram from the main point of view. Consequently, the system is also able to analyze already optimized layouts for poorly represented parts (Figure 12b). Even though the combination of representative subassemblies may not be perfect, if the visibility of all parts of the assembly is taken into account, the resulting presentation will increase the capability of mentally linking the exploded view and the additional renderings. Therefore, multi-perspective renderings will be supported best if poorly presented parts are detected after the optimization of representatives is finished.

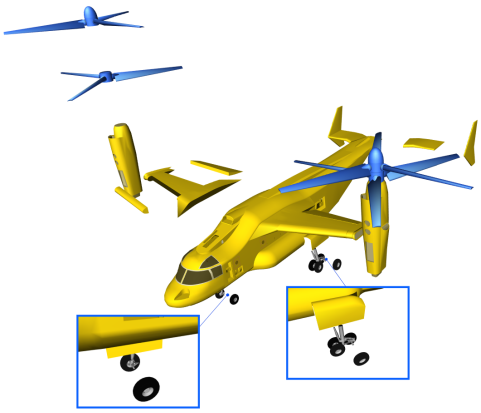
7.2. Viewpoint Restriction and Linking of Multi-Perspectives

In order to present the renderings from secondary viewpoints as close as possible to their location in the compact explosion diagram we place them as annotations into the main explosion diagram. However, by spatially separating the presentations from different points of view, we require the user to put some effort into mentally linking the content of our renderings. To assist the user in this task, the layout of subassemblies shown in additional views is only allowed to change if it is completely occluded in the main view onto the explosion. Otherwise, the layout visible in the secondary view will differ from the one in the main presentation, which makes it difficult to mentally relate these structures to one another.

In addition, we restrict the offset between the secondary viewpoint and the main viewpoint to an adjustable threshold. Otherwise, the difference between the secondary viewpoint and the main viewpoint may lead to presentations, which are difficult to read. In figure 13a the secondary point of view is offset by more than 90 degree to the main point of view. In case of the rear landing gears the difference reaches nearly 180 degrees. The mental linking between the views may become difficult if the points of view have been offset too far. Therefore, we restrict secondary viewpoint to vary only within a certain range to the main viewpoint (Figure 13b).

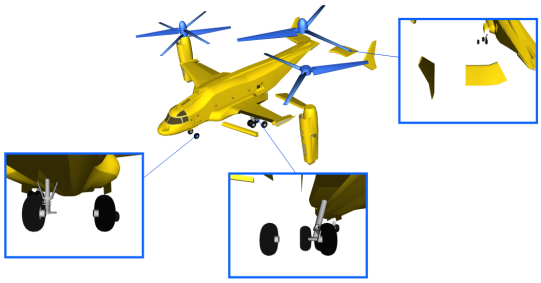


(a)

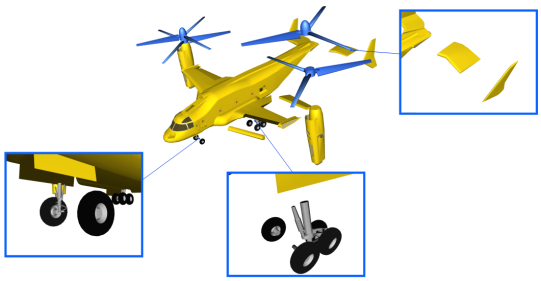


(b)

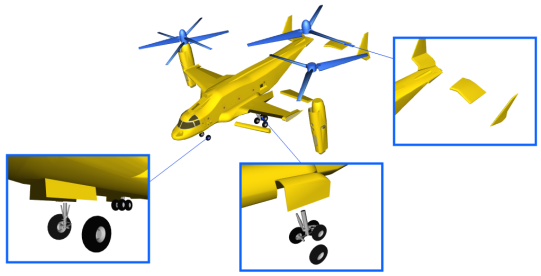
Figure 12: *Multi-perspective rendering as a post-process versus integrated multi-perspective rendering. (a) Poorly presented explosions may become even worse if they have been removed from the layout optimization. Notice the occluded wheels and the resulting lack of context to mentally link the annotation to the main representation (b) By optimizing the combination of all representatives before rendering from secondary points of view, chances are better to provide enough contextual information to mentally link the content in the secondary with the one in the main rendering.*



(a)



(b)



(c)

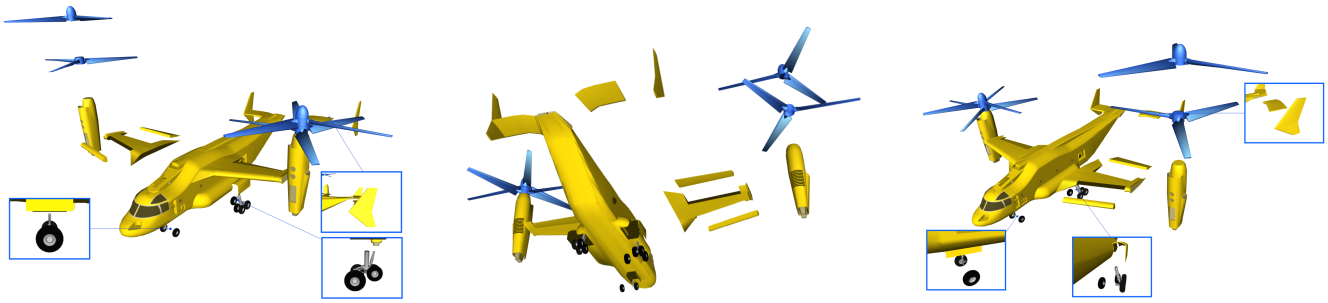
Figure 13: *Linking multi-perspective renderings to their original location. (a) The views depicted in the annotations do not correspond with the view on the layout. This is irritating and mentally linking of annotations and the corresponding subassemblies may be difficult. (b) The secondary view is restricted to stay close to the main point of view. The annotations are less confusing but lack contextual information. (c) Adding contextual information further supports mentally linking the content of annotations to their counterparts in the main view.*

To compute a secondary point of view we do not only include the subassembly itself but also consider its contextual information. Otherwise, our rendering may not show any information besides the subassembly, which may also influence the ability to relate the renderings to one another (Figure 13c). Additional parts can be forced into the view by adding weight to the measure describing the visibility of the rest. However, since these additional parts may increase visual clutter, we introduce a new parameter to the optimization which controls the amount of presented contextual elements. Only those parts are considered to be contextual information, which are in direct contact with the representative subassembly. We measure the amount of contextual information by using the size of its 2D projection, which we force to be within a certain distance to an optimal value.

Equation 3 describes the contextual quality measure which is based on the distance from the optimal amount of contextual information. The difference between the threshold value $contextTh$ and the normalized amount of pixel showing contextual elements ($contextPixel$) ideally must be close to zero. We chose a threshold value of approximately 0.33, which scores points of view highest if a third of the corresponding rendering is covered by contextual information. Such a presentation conforms with the rule of thirds, which is, according to Gooch et al. [5], the best known rule of layout.

$$contextQuality = (1 - |contextTh - contextPixel|) \qquad (3)$$

To ensure an unobstructed view onto the explosion of the subassembly from a secondary point of view, we have to put a higher emphasis on the visibility as well as the direction of



(a) (b) (c)

Figure 14: *Viewpoint optimization. (a) Ineffective representative explosions due to arbitrary viewpoint selection. (b) Even though the quality parameters of all representatives have been taken into account, an inappropriate view on the explosion diagram may be chosen. (c) Front facing viewpoints have been weighted higher to prefer renderings of frontal views.*

the explosion. Otherwise, close objects may occlude parts of the representative or representatives explode into to the viewing direction, making the secondary point of view less valuable.

Compact explosion diagrams which consist of a large number of small subassemblies may lead to a cluttered presentation of an equally large number of annotations. To make efficient use of the available space, we reduce the number of annotations, by combining similar ones into a single annotation (Figure 3(E)). However, even if we combine certain subassemblies within a single secondary presentation, the amount of annotations is still unpredictable. Therefore we furthermore assign importance values based on the visibility of annotated parts. This allows our system to select the most important annotations until the available screen space is filled.

8. Camera Optimization

The system described so far optimizes a manually selected viewpoint. To further automate the generation of compact explosion diagrams we optimize its main viewpoint as well. To render from a proper point of view, we first compute the optimal layout for different viewpoints before we select the one with the highest score. Similar to previous approaches, we select a set of candidate viewpoints by sampling the bounding sphere of the object of interest [5, 17]. The orientations are derived for each candidate viewpoint by pointing the camera to the center of the bounding sphere. An adjustable threshold determines the number of equidistant sample points on the sphere.

8.1. Viewpoint Evaluation

Good viewpoints maximize the quality of combined representative explosions, just as bad viewpoints may result in incomprehensible presentations, even after optimizing the layout according to the set quality parameters. For example, the chosen secondary point of view onto the wheels of the airplane in Figure 14a does not show the explosion direction. Furthermore, all elements except the exploded wheel are occluded. However, the layout is optimal for this viewpoint.

The quality of viewpoints is evaluated using the parameters presented in section 6.1. By performing the optimization of representatives for all viewpoints on the bounding sphere and selecting the viewpoint with the highest score, the system selects the best viewpoint combined with the best representative explosion. Notice the different representations in the annotations of the front wheels in Figure 14a and 14c.

However, even though this algorithm allows us to represent the explosions from an optimal point of view, with respect to the quality parameter of its explosions, the object itself may not be sufficiently represented from this viewpoint. For example, Figure 14b shows the view with the highest score on the exploded representatives. However, the view from the bottom does not provide a good view on the airplane itself. Such defective viewpoints were studied by Blanz et al. [2]. They found out that users select viewpoints which maintain the natural up-orientation of the object, while simultaneously avoiding occlusions. In addition, rather low diagonal views were often preferred, showing objects from familiar positions which contain as much information as possible. Based on this data, we allow users to influence the allowed viewpoint selection by weighting a certain range of viewpoints higher, than others. Using this restriction, we select the viewpoint with the highest quality value, while simultaneously clearly presenting the object of interest (Figure 14c).

9. Conclusion and Future Work

Explosion diagrams provide a powerful technique to visually communicate the composition of 3D objects. However, if complex models have to be presented, the explosion diagram may suffer from clutter caused by the large number of displaced parts. To reduce the complexity illustrators often chose to displace only a subset of parts. In addition, they annotate poorly visible parts of the explosion diagram with renderings from different points of view. In this article, we have presented a system, which is able to automatically mimic both techniques. We presented an algorithm for detecting similar subassemblies in

a 3D assembly and we have presented an algorithm to compute a disassembly sequence, which allows computing an explosion layout of similarly exploding subassemblies. We discussed several strategies to ensure that representative exploded views demonstrate the unexploded remaining subassemblies. In addition, we presented a quality measure for exploded views and discussed each parameter comprising this measure. We demonstrated their impact to the final compact explosion layout. Furthermore, we have presented a strategy to optimize the combination of representative exploded views and we have shown how our system can optimize the point of view.

Our presentation technique is able to reduce the complexity of exploded views. In addition, it increases their effectiveness by offering renderings of exploded subassemblies from multiple perspectives. Both qualities are especially important for renderings meant for books or other print media, which do not have the luxury to interact with the object of interest in order to gain information about its assembly. Even though our work focuses on assemblies which consist of recurring subassemblies, our explosion diagrams do not differ from any traditional ones if no such subassemblies exist.

In a future work, we will conduct a user study to analyze the impact of our presentation technique to interactive explorations. We will investigate its efficiency with and without further interaction techniques by comparing our system to traditional interaction techniques of explosion diagrams such as those presented by Li et al. [9]. We will further investigate additional parameter to support mental linking between exploded views and additional points of view.

10. Acknowledgements

This work was sponsored by the Christian Doppler Laboratory for Handheld Augmented Reality. We would like to thank D.V. Vranic and T. Schreck for providing us with the DESIRE shape descriptor implementation.

References

[1] M. Agrawala, D. Phan, J. Heiser, J. Haymaker, J. Klingner, P. Hanrahan, B. Tversky, Designing effective step-by-step assembly instructions, ACM Transactions on Graphics 22 (2003) 828–837.

[2] V. Blanz, M.J. Tarr, H.H. Bülthoff, What object attributes determine canonical views?, Perception 28 (1999) 575–600.

[3] S. Bruckner, M.E. Gröller, Exploded views for volume data, IEEE Transactions on Visualization and Computer Graphics 12 (2006) 1077–1084.

[4] G. Dueck, T. Scheuer, Threshold accepting: A general purpose optimization algorithm appearing superior to simulated annealing, Journal of Computational Physics 90 (1990) 161–175.

[5] B. Gooch, E. Reinhard, C. Moulding, P. Shirley, Artistic composition for image creation, in: Proceedings of the 12th Eurographics Workshop on Rendering Techniques, Springer-Verlag, 2001, pp. 83–88.

[6] K. Hartmann, K. Ali, T. Strothotte, Floating labels: Applying dynamic potential fields for label layout, in: Proceedings of the 4th International Symposium on Smart Graphics, pp. 101–113.

[7] D. Kalkofen, E. Mendez, D. Schmalstieg, Comprehensible visualization for augmented reality, IEEE Transactions on Visualization and Computer Graphics 15 (2009) 193–204.

[8] D. Kalkofen, M. Tatzgern, D. Schmalstieg, Explosion diagrams in augmented reality, Proceedings of the 2009 IEEE Virtual Reality Conference (2009) 71–78.

[9] W. Li, M. Agrawala, B. Curless, D. Salesin, Automated generation of interactive 3d exploded view diagrams, ACM Transactions on Graphics 27 (2008) 1–7.

[10] W. Li, M. Agrawala, D. Salesin, Interactive image-based exploded view diagrams, in: Proceedings of Graphics Interface, pp. 203–212.

[11] W. Li, L. Ritter, M. Agrawala, B. Curless, D. Salesin, Interactive cutaway illustrations of complex 3d models, in: Proceedings of ACM SIGGRAPH 2007, ACM, New York, NY, USA, 2007, pp. 31–39.

[12] P. Mijksenaar, P. Westendorp, Open Here. The Art of Instructional Design, Thames & Hudson, 1999.

[13] C. Niederauer, M. Houston, M. Agrawala, G. Humphreys, Non-invasive interactive visualization of dynamic architectural environments, in: Proceedings of the Symposium on Interactive 3D Graphics, pp. 55–58.

[14] A. Raab, M. Rüger, 3d-zoom: Interactive visualisation of structures and relations in complex graphics, in: 3D Image Analysis and Synthesis, pp. 87–93.

[15] T. Rist, A. Krüger, G. Schneider, D. Zimmermann, AWI: A workbench for semi-automated illustration design, in: Advanced Visual Interfaces, ACM Press, New York, NY, USA, 1994, pp. 59–68.

[16] M. Ruiz, I. Viola, I. Boada, S. Bruckner, M. Feixas, M. Sbert, Similarity-based exploded views, in: Proceedings of Smart Graphics 2008, pp. 154–165.

[17] D. Sokolov, D. Plemenos, Viewpoint quality and scene understanding, in: 6th International Symposium on Virtual Reality, Archaeology and Cultural Heritage, pp. 67–73.

[18] H. Sonnet, S. Carpendale, T. Strothotte, Integrating expanding annotations with a 3d explosion probe, in: Advanced Visual Interfaces, ACM Press, New York, NY, USA, 2004, pp. 63–70.

[19] M. Tatzgern, D. Kalkofen, D. Schmalstieg, Compact explosion diagrams, in: NPAR '10: Proceedings of the 8th International Symposium on Non-Photorealistic Animation and Rendering, ACM, New York, NY, USA, 2010, pp. 17–26.

[20] D.V. Vranic, Desire: A composite 3d-shape descriptor, in: Proceedings of the IEEE International Conference on Multimedia and Expo, Amsterdam, The Netherlands, pp. 962–965.

[21] R.H. Wilson, On Geometric Assembly Planning, Ph.D. thesis, Stanford University, Stanford, California, 1992.

[22] X. Yan, J. Han, gSpan: Graph-based substructure pattern mining, in: Proceedings of the IEEE International Conference on Data Mining, IEEE Computer Society, Washington, DC, USA, 2002, pp. 721–724.