Title:
Automatic CAD Assemblies Generation by Linkage Graph Overlay for Machine Learning Applications

Authors:
Lucas Vergez, lucas.vergez@gadz.org, Arts et Métiers Institute of Technologies
Arnaud Polette, arnaud.polette@ensam.eu, Arts et Métiers Institute of Technologies
Jean-Phillipe Pernot, jean-philippe.pernot@ensam.eu, Arts et Métiers Institute of Technologies

Keywords:
Shape synthesis, linkage graph, CAD models generation, CAD assembly.

DOI: 10.14733/cadconfP.2021.46-50

Introduction:
Enlarging 3D model databases by shape synthesis is a large field of research. Indeed, the use of machine learning techniques requires a huge amount of labeled CAD models, and it is therefore crucial to rely on large and varied databases. Most of existing works in shape synthesis focus on everyday life objects generation [1]. However, these methods often do not work on assemblies composed of several CAD models, and it is the aim of this paper to develop a new shape synthesis method to enlarge existing CAD assembly databases.

Today, there exist lots of free databases of non-labeled CAD models (e.g. GrabCAD, 3D Warehouse, Turbosquid) often available as STEP or IGES files. Unfortunately, very few of these databases are labeled. Other databases like PartNet [2] and ShapeNet [3] are currently labeled by crowdsourcing, but they do not contain complex mechanical assemblies.

Current works in shape synthesis often use auto-encoders to generate new coherent CAD assemblies [4]. Moreover, there are probabilistic models to create diversity in large 3D Database [5] or to classify 3D assemblies [6]. Those techniques often use linkage graphs [7] to classify and generate new coherent assemblies. Furthermore, information within the linkage graphs differs according to the method, and those graphs are not suitable for complex CAD assemblies like hydraulic pumps. But the main issue of all methods is still that those databases have to be labelled.
The method explained in this paper consists in creating new labeled CAD assemblies from existing ones by linkage graph overlay. Here, the STEP file format has been adopted in order to be the most reproductible and to be adaptable. The linkage graphs are automatically created thanks to the identification of the linkages between the components. Indeed, linkages are not included in the STEP files and they need to be computed. Theses linkage graphs are then analyzed and components with similar linkages are detected. Finally, once the similarities detected, the corresponding components can be exchanged to created new assemblies for which the labels can be directly inherited from the source assemblies.

The contribution is threefold: (i) a method to create linkage graphs from existing non-labelled CAD assemblies; (ii) a method to recognize basic components using linkage graphs; (iii) a smart overlay method to replace some components while keeping the coherence between all the components of the assembly. The algorithm has been implemented in Python on FreeCAD and it has been tested on several test cases. Figure 1 shows the overview of the method, from the graph synthesis to the components overlay, finishing with the replacement of the components. The results are presented and discussed, and a conclusion ends this extended abstract while discussing the next steps.

Main Idea:

Overview

The proposed method reduces user input in order to be able to apply it for any CAD assembly like pumps, motors, or many more complex CAD assemblies. The method starts by extracting a complete linkage graph from an assembly, and it stores a lot of useful information on its nodes and edges. Then, components sharing the same functional surfaces are recognized thanks to the complete linkage graph, and they are stored in a component family database. Finally, similar components are replaced to define new assemblies, thanks to the information stored into the linkage graphs. The components can be modified before placement to fit with the assembly. This process is illustrated on Figure 1 and detailed in the next sections.

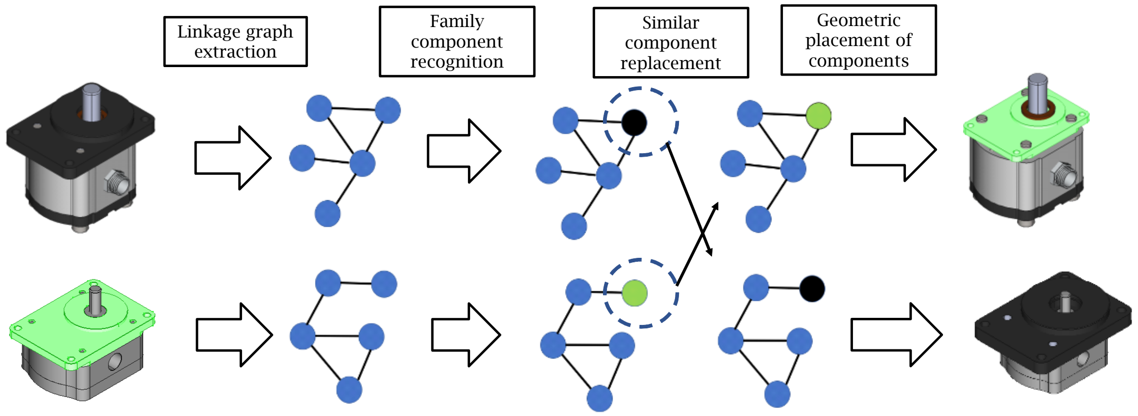

Fig. 1: Overview of the proposed method creating a set of new assemblies (right) from existing ones (left).

Graph Synthesis

The created graphs are composed of nodes and edges. Each node represents a component, and the edge between two components keeps track of the mechanical linkage between them. The edges also support additional information like the mechanical characteristics of the linkage. All this information is described below.

To create a linkage graph of a CAD assembly, it is mandatory to first recognize linkages between components. In this paper, only two linkage families are studied, but this could be extended with other types of linkages. The considered linkages are the pivot linkage and the flat-surface linkage. These two types are sufficient to work initially on partially complete graphs, even if some linkages can thus not be detected. The graph created is a network of components, with vectors associated to each edge, which completely characterizes the linkage between the two components.

In order to recognize these linkages, the main idea is to browse all faces of the assembly and, for each face, to check the other faces and verify if some conditions are satisfied in order to decide if the components related to those faces are connected or not.

Here are the conditions for the two linkages:

Pivot linkage:
-The two surfaces are cylinders
-Axes are coaxial
-The two parts can fit together
-Radius of the two cylinders are the same
-The two components are different
-The two parts are nested

Flat-surface linkage:
-The two surfaces are planar
-The two components are different
-Orientations of the two surfaces are different
-The two axes are colinear
-The distance between the two surfaces is equal to zero.

When all those conditions are met, the type of linkage is added to the edge of the graph linking the two related components of the assembly under analysis. This linkage is characterized by a vector which characterize the linkage. An example of a CAD assembly and his linkage graph is shown Figure 2.

These two types of links are sufficient to work initially on partially complete graphs, even if some links are not detected here. The complexity of the algorithm is $O(n^2)$ with n the number of faces and takes 3 min with an IntelCore i7 at 2,5 GHz and 8 Go of RAM for a 1000 surfaces CAD assembly (which correspond to a small hydraulic pump).

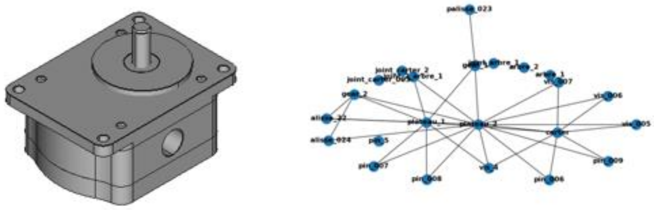

Fig. 2: A hydraulic pump and its computed linkage graph.

Components Overlay

In order to distinguish components that can be replaced from the ones that cannot, the similarities between two components have to be recognized. The hypothesis is that two components are interchangeable if they have the same type of linkage. This notion can differ depending on the type of parts, as parts are linked differently in mechanical assemblies. Thus, in its current version, our algorithm requires a human decision to characterize a family of components by its connections with the others.

As illustrated, the proposed method is particularly interesting as it allows identifying similar components with very few hypotheses. As theses components are now known to be similar, they are replaceable and can be exchanged between the considered assemblies.

Our method allows to recognize potential replaceable components, and it is easy to foresee the number of new assemblies possibly generated. Indeed, if there are 10 assemblies with for each assembly 5 components known to be replaceable, there are 50 replaceable components overall. So, this method can create 50^5 new assemblies, i.e., more than 3 billion assemblies, starting from a database of 10 assemblies only. However, this calculation does not consider configurations wherein components are geometrically similar, like for instance the repetition of a screw in an assembly. Thus, in the current version, the generation has not been fully automated so as to let the user decide which components are to be replaced. Otherwise, this would generate a huge number of assemblies, among which many of them would be exactly the same.

Components Placement

As the proposed method handles CAD models described in STEP files, when imported in the CAD modeler they are considered as dead models without construction trees. Thus, it is difficult to modify them in order to adjust their shapes to their new positioning. Therefore, the modification and placement of parts is an important step of our algorithm.

Indeed, the manipulated assemblies often do not have the same scale and shape geometry. The modifications of the geometry depend on the considered family of parts. For example, screws do not have to undergo major geometric modifications, as they just need to adapt to new holes in which they will be placed. For some parts, the required modifications could however be more complex.

This paper focuses on four types of operation: rescale, cropping, fill and drill holes, which already answer many adjustment requirements.

Figure 3 shows a partial list of possible geometric modifications of a part a), with filled holes b), drilled holes c), rescaling d) or cropping e).

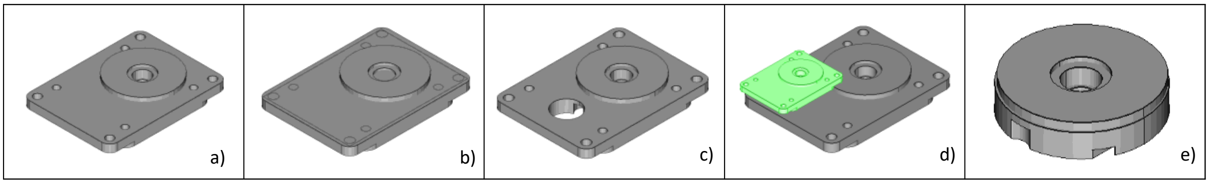

Fig. 3: Examples of geometric modifications of a CAD model required to welcome new parts or to fit in a new component.

In order to place the new part within the assembly, a new local reference frame is defined for all parts of the family. This reference frame allows to make transformations with respect to it. Thus, the placement operation will be the same for all parts of the family.

Then a rotation and a translation are defined to place it at the right place. All information required to perform those transformations is available in the linkage graph.

Once components have been placed, the newly generated assemblies are exported as STEP files. Moreover, this process generates new assemblies from existing ones, if those initial assemblies belong to the same family, then the newly created ones can also be associated to this family.

Results and Discussion

The method has been tested on several example, and the screw replacement is shown Figure 4.

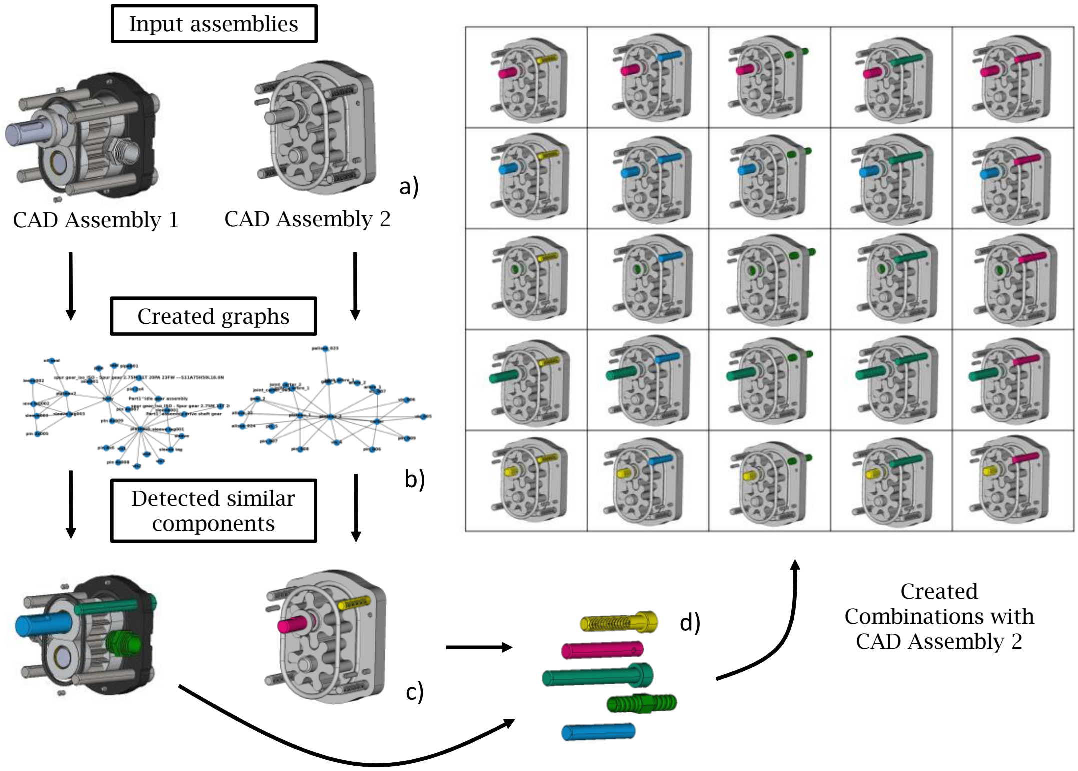

Fig. 4: Creation of 25 new assemblies while automatically replacing screws between two input assemblies.

The Figure 4.a shows two hydraulic pumps, where the body and the drive plate are hidden, and used as inputs of the assembly generation algorithm. The associated linkage graphs are automatically created (Figure 4.b), and the similar components are detected (Figure 4.c). All those components can replace the two components of the second pump (Figure 4.d). Then, $5^2=25$ CAD assemblies can be created with automatically adapted and placed components as shown in Figure 4.e.

Here, the geometrically similar components are cleaned to avoid duplicate assemblies. However, our method does not yet verify collisions within the created assemblies, and some operations could be improved. This is part of our future works.

Conclusion:
This paper introduces a method to create new assemblies from diverse existing assemblies of a same family. It allows to generate linkage graph from a non labelled assembly, and it analyzes these graphs to find similar components. Finally, it replaces similar components in an assembly to create multiple new consistent CAD assemblies. For the moment, this method requires a human interaction during a pre-processing step, in order to define the characteristics of linkages for a part family. Furthermore, the method could be improved by changing the structure of the graph before creating new assemblies. Indeed, a semantic graph could improve the diversity of the assemblies during the creation process. Finally, the proposed approach creates many CAD Assemblies, which can be stored in a database and used for Machine Learning applications.

References:

[1] Li, J., Xu, K., Chaudhuri, S., Yumer, E., Zhang, H., Guibas, L., 2017. GRASS: generative recursive autoencoders for shape structures. ACM Trans. Graph. 36, 1–14. https://doi.org/10.1145/3072959.3073637

[2] Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H., 2019. PartNet: A Large-Scale Benchmark for Fine-Grained and Hierarchical Part-Level 3D Object Understanding, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Presented at the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, Long Beach, CA, USA, pp. 909–918. https://doi.org/10.1109/CVPR.2019.00100

[3] Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F., 2015. ShapeNet: An Information-Rich 3D Model Repository. arXiv:1512.03012 [cs].

[4] Mo, K., Guerrero, P., Yi, L., Su, H., Wonka, P., Mitra, N., Guibas, L.J., 2019. StructureNet: Hierarchical Graph Networks for 3D Shape Generation. arXiv:1908.00575 [cs].

[5] Kalogerakis, E., Chaudhuri, S., Koller, D., Koltun, V., 2012. A probabilistic model for component-based shape synthesis. ACM Trans. Graph. 31, 1–11. https://doi.org/10.1145/2185520.2185551

[6] Fish, N., Averkiou, M., van Kaick, O., Sorkine-Hornung, O., Cohen-Or, D., Mitra, N.J., 2014. Meta-representation of shape families. ACM Trans. Graph. 33, 1–11. https://doi.org/10.1145/2601097.2601185

[7] Jaiswal, P., Huang, J., Rai, R., 2016. Assembly-based conceptual 3D modeling with unlabeled components using probabilistic factor graph. Computer-Aided Design 74, 45–54. https://doi.org/10.1016/j.cad.2015.10.002