



**HAL**  
open science

## Automating generation of kinematic keypoints for disassembly process toward virtual reality

Sébastien Pascault, Frédéric Noël, Jérémie Le Garrec, Claude Andriot, Adrien  
Girard

► **To cite this version:**

Sébastien Pascault, Frédéric Noël, Jérémie Le Garrec, Claude Andriot, Adrien Girard. Automating generation of kinematic keypoints for disassembly process toward virtual reality. Lecture Notes in Computer Science, 2021, Virtual Reality and Mixed Reality, 13105, pp.145-151. 10.1007/978-3-030-90739-6\_9. hal-03950713

**HAL Id: hal-03950713**

**<https://hal.science/hal-03950713v1>**

Submitted on 31 May 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Automating Generation of Keypoints for Disassembly Process toward Virtual Reality

Sébastien Pascault<sup>1</sup>[0000-1111-2222-3333], Frédéric Noël<sup>2</sup>[1111-2222-3333-4444], Jérémie Le Garrec and Claude Andriot

<sup>1</sup> Université Grenoble Alpes, 621 Avenue Centrale, 38400 Saint-Martin-d'Hères, France

<sup>2</sup> CEA Paris-Saclay, 91191 Gif-sur-Yvette, France  
sebastien.pascault@outlook.com

## Abstract.

This paper describes a novel approach to generate and model assembly semantic meaning enclosed in product assembly features in order to optimize the process preparation time of Virtual Reality simulations. The proposed approach is based on a set of heuristic rules to generate semantic KeyPoints used to idealized the assembly model. This study will identify though a disassembly process a certain amount of semantic rules in other to extract and translate assembly semantic knowledge from assembly CAD models. The proposed approach is based on two stages: assembly semantic extraction and assembly semantic recognition. In the first stage, the internal boundary representation (B-rep) recognition method is utilized to extract assembly semantic knowledge from assembly CAD models using SolidWorks' API functions. In the second stage, a multi-level semantic rules model is generated. The approach is demonstrated and validated on a use case with a complete disassembly scenario tested in a Virtual Reality.

**Keywords:** CAD Model Processing, Virtuale Reality, Feature Extraction, Key-Point Generation, Disassembly Semantics, Rules Engine.

## 1 Introduction

Assembly process in industrial environments represent complex tasks resulting in significant total manufacturing and labor costs [1]. Changes in the market towards ever customized and sustainabled products requires developing ever more flexible and robust manufacturing processes [2]. The traditional approach of manually customizing production cells and setups to predefined products consumes too much time and cost [3]. To shorten process and to optimize product life cycles, manufacturers are nowadays relying on simulations in Virtual Reality (VR) [4]. The scene's behavior in those VR simulations is mainly leaded by user interaction and dynamics of solids including collision and kinematics, which are based on 3D products usually designed with a Computer-Aided Design (CAD) software. However, CAD modelers and VR environment do not use the same internal models even for rendering. The preparation of VR scenes remains difficult and time consuming because of the translation of models and authoring of extra information or data lost during translation. Even for basic systems, it requires a significant amount of engineering time which can reach to one or several weeks for complex systems. In general, all preparation steps first implement a CAD model processing phase, where the features interesting for the specific process and their relations should be recognized. The problem is that in most cases the data extracted from

CAD systems rely on geometric information, while the intrinsic engineering meaning of the assembly's components are neglected. This because usually details associated with parts functionalities and engineering meanings are implicit. This kind of data may be in fact included as annotations in the CAD model, but these attributes are not rigorous and unique since it depends on the designer choice, thus it may result difficult to explicitly interpret them.

The vision of this research topic is to propose a CAD recognition approach for the semantic interpretation of 3D assembly models in order to solve current problems in terms of data preparation for mixed reality simulations in non-homogeneous software environments. The objective is to generate structured model in order to reduce the overall preparation time and minimize the manual intervention of a human operator.

This paper considers an approach to generate a disassembly semantic based on information extracted from a 3D model to generate automating keypoint and features able to describe the whole disassembly process.

## 2 Virtual Reality disassembly concepts

The automated generation of CAD assembly semantic is a fundamental research subject highly addressed in the last decades [3]. The techniques used in the literature are different, especially according to the format of the analyzed 3D objects, which can be represented by their surfaces (B-Rep) or by their volumes (CSG) [JC Leon]. In general, three main categories of methods can be distinguished, namely: geometry based methods, graph based methods and feature based methods [10, 18]. Our approach will focus on this latter.

### 2.1 Feature based approaches

To generate assembly joint from 3D models, assembly features representation are necessary. Assembly features describe the relation of components in an assembly group. Merging definitions from [7,8], it is defined here as *"a partial form or a product characteristic that is considered as a unit and that has an engineering semantic meaning for assembly design, process planning or manufacture"*. To extract or recognize semantic features in CAD models many approaches exist. Hasan and Wikander [7] differentiate between three approaches: internal approaches, external approaches and ontology-based approaches. In internal methods, the Application Programming Interface (API) of the CAD software is used in order to extract topological, geometrical and assembly information relating to a part or an assembly. While in external methods, a CAD model file is exported in a neutral data format (e.g. STEP-AP 214). C.M.Costat focuses on internal approach to generate assembly sequence planning. A. Neb [] combined external approaches with internal approaches to get the features directly from the 3D assembly model. Other authors use ontology base approaches to exploit geometric and shape information. Some of them are more targeted at the identification of single specific components, possibly exploiting different shape descriptors [8], or at the recognition of a restricted number of entities, to derive structural information about the assemblies [17]. Others aim to classify parts according to kinematic as well as functional properties [15, 19]. Extracting all these features is a first step to create a semantic representation of an assembly model. Those features will be completed in a second step by the use of keypoints to characterize joints states and behaviors.

## 2.2 Keypoints : main specification for physics engines

To provide disassembly process in a virtual world, physics engines are being used. Those engines are program that provide an approximate simulation of certain physical systems such as rigid body dynamics, soft body dynamics or fluid dynamics [] (in order to achieve basic task and reproduce user gesture in a simplified way this study will be limited to rigid bodies). There are many available physics engine on the market such as Bullet, PhysX, ODE or XDE. Knowing our study focuses on VR we decided to use both PhysX and XDE which are direct kinematics engine well suitable for it. XDE Physics on the first hand is an interactive physics engine featuring precise collision detection, multibody and beam dynamics, at rate compatible with haptic rendering. XDE allows to animate complex scene with multiple rigid bodies interconnected with mechanical joints very efficiently [Xav]. PhysX on the other hand is a scalable multi-platform physics engine providing the same features and already integrated into some of the most popular game engines, including Unreal Engine and Unity3D. Both XDE and PhysX offer a simple programming interface where assembly features described in section 2.1 can be used and easily implemented.

To compute dynamics, most of the previously named physics engine have the ability to rely on specific “point features” in the assembly model allowing them to reason on kinematic links rather than collisions. Those “point features” are referred to as “Key-points” (KP) and represents an identification of surface points surrounded by patches amenable to find correspondences [13]. KP are specified by detector algorithms according to certain mathematical base or specific rules (described in section 3.3).

## 2.3 Extra issues for disassembly process

Regarding assembly/disassembly process planning, lot of approaches have been proposed over the years. To generate optimal and close to optimal assembly and disassembly sequences A. Neb [11] proposed an approach to generate automatically an Assembly Graphs by just using information from a 3D Model of a product assembly and Assembly Features generated from a CAD system. His proposition uses a feature detection internal approach by the aid of an API implemented in SolidWorks, but with the method of external approaches to generate assembly features out of low-level features.

With a similar approach C.M.C. Costa [12] focuses and disassembly sequence from SolidWorks Symbolic Geometric Relationships and describes a recursive branch-and-bound algorithm to find the optimal disassembly plan. Costa relies on a geometric reasoning engine analyzing the spatial relationships present in SolidWorks assemblies to automatically compute the precedences and part decoupling directions. The disassembly approach discussed here could exploit the semantic model described in the next section. Last but not least the approaches presented above mainly focus on Robotic while our goal, as already mention, is to use all these semantic features recognition and assembly/disassembly planning methods for VR. Works with a similar objective has been carried out. Z. Liu [] presents a constrained behavior management approach that realizes assembly relationship recognition, constraint solution and constrained motion to facilitate assembly interaction in a virtual environment. P. Bourdot [] goes even further and proposes an approach based on a VR–CAD framework making possible intuitive and direct 3D edition on CAD objects within Virtual Environments (VE). The cornerstone of this approach is to manage implicit editing of CAD model in real-time according to the user’s 3D interactions. This model uses a naming technique of B-Rep components and a set of logical rules to provide straight access to the operators of Construction History Graphs (CHG).

### 3 Feature Recognition and Semantic Rules Approach

#### 3.1 Feature Extraction from CAD models

To solve the problem of data exchange between CAD design and other software two extraction framework based on SW API have been developed, following the approach of internal B-Rep CAD analysis. For this extraction tools SW has been chosen due to its advanced and accessible libraries. A first framework was developed to extract B-Rep topology features using Open CASCADE Technology (OCCT). The goal behind extracting this B-Rep data was to recover as much information as possible at the topology level. The resulting dataset describes the different B-Rep shells of the model and all the associated data namely: transformation matrix, type of patch (plane, cone, cylinder, bspline, etc), orientation, bounding-box, center of mass and inertia matrix. But B-Rep data aren't enough to describe the building complexity of a product. In a SW product, parts are often assembled using symbolic relationships, that specify their spatial arrangement along with the type of movements they were designed for. The relationships describe above are represented by primitive mates that share a common data structure. To extract them a second framework (carried out in C#) was created using a set of function calls from SW libraries. A new dataset was thus created and associated to the B-Rep dataset. This complete set of data (saved in json) was finally imported and parsed in Unity3D to be exploited with a physics engine. We obtained that way a complete dataflow of the CAD information [Fig.1].

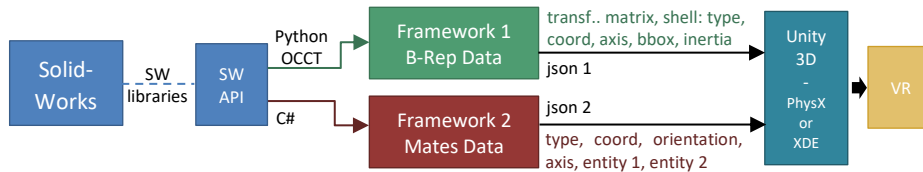


Fig. 1. CAD feature extraction based on SolidWorks API (complete dataflow)

#### 3.2 Semantic Rules Engine

Based on input from CAD dataset recognition stage above an assembly design Semantic Rules Engine (SRE) is proposed [Fig. 2]. This SRE describes a novel approach based on features recognition and detection rules to model assembly semantic meaning. The proposed SRE, is a decomposition into three semantic levels (or layers) inspired by Hasan work [9], namely: Geometric Level, Kinematic Level and Behavior Level.

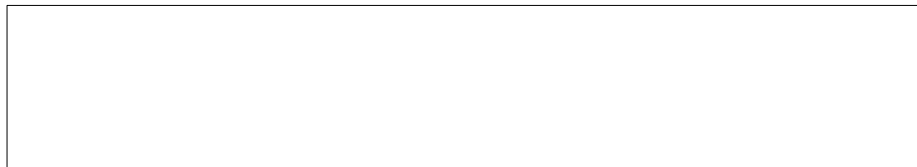


Fig. 2.

The Geometric Level regroups two type of features : Shell Feature and Form Features. Shell Feature are related to B-Rep entities extracted from the geometric/part modeler.

In order to detect specific surface shape for each assembly part, hose shell features are then analyzed according to simple heuristic rules such as nearest neighbors, shell dimension or symmetry axis detection. These rules are logical reasoning based on the analysis of conventional naming or conventional geometric configuration (which differs from ontological methods). That set of rules assesses if the geometry of an assembly part respects a specific class distinguishing characteristics in both shape and sizes. That results in a form classification of assembly objects thus called Form Features (e.g, hole form, cylinder form, thread form, etc). On the Kinematic Level, mating features are further analyzed in order to determine joining features. Each imported mates is linked to two entities that corresponds to part in the assembly. The first entities represents the origin part where the mates is beeing attached and the second entities represents another targeted part giving a first assembly knowledge for potential Joining Feature between them. Lastly, Behaviour Level regroupes all the behavior rules aimed to detect application-specific features whose semantics are carried by KeyPoint. Those KP are classified into three different types. Handling KP firstly represents position on parts's surface where object can be grabbed by user hand or tool. Interest KP then defines sepecific positions inside parts (and herited axis orientation) where a kinematic joint exists. Lastly Transition KP describes specific coord in parts where mates state and related DOF can change according the part position in assembly at a given time in the disassembly scenario. The combination of these KPs allows to add a new semantics to the model. For example, a combinaison of two interest KPs so called "bi-points" can represent an axis. Such "KeyAxis" allows reasoning with angles and parallelism between assembly objects.

#### 4 Use Case

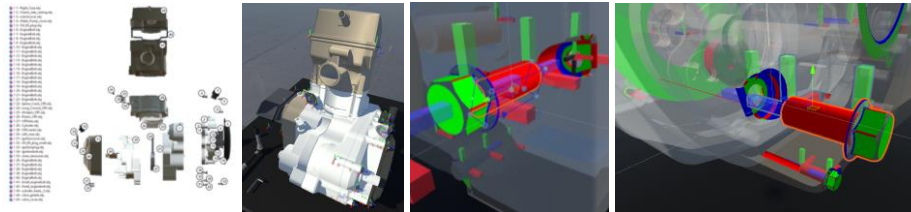


Fig. 3.

For the first experimentations, the use-case of the Honda engine CFR250X realized in SW was chosen, because this assembly was complex enough to be able to test by algorithm the Semantic Rules Engine described above (the same model was used in C.M.C. Costa study []). The objective was to redo the complete disassembly process of the Honda engine in Unity using both PhysX and XDE to compare our automatic (or semi-automatic) approach with manual VR preparation methods. Doing this whole disassembly process was also aimed to validate different aspect of our approach: the nature of the input data, the identification of the missing semantic in our CAD dataflow, and lastly the contribution of our SRE model. The purpose of the experiments carried out on this assembly is to check whether all the information is available as inputs and, if not, to continue to set up new automatic identification rules as data enrichment for our semantic engine (by manually providing real time modification during the Honda Engine preparation process).

In order to facilitate the implementation of the different rules and to identify all possible scenarios, the engine assembly and disassembly graphs have been defined based on the Honda CRF250x engine official documentation. A unique disassembly sequence scenario has thus been defined. Based on this first scenario, a first "control sample" was prepared in a "standard" manual way using XDE in order to obtain a point of comparison point with the SRE to be tested. Each kinematic link and each keypoint have therefore been configured manually through XDE interface.

The first behavioral tests based on this same scenario were then carried out, but limited to logical data generated automatically by the rules engine. Unity's PhysX engine was used in these tests. The results obtained on the helical connections of the screws were in accordance with the expected behavior and similar to the behavior of the XDE engine. On the other hand, more extensive cases of more complex mechanical links have not yet been completed or in a way that is too approximate.

To describe the behaviour of our SRE, we will more particularly focus on the screws and their disassembly sub process. The first SRE Geometric Level starts by analysis each B-Rep data (shells data saved in the ROM memory inside json files) previously extracted and load the Shell Features in the RAM memory. The first level then identify the potential class of the current part by detecting the presence of two shells belonging to cylinder type positioned around a symmetry axis, then detect the presence of plane shells whose center is located on the same axis. The algorithm classified this part has a potential cylinder or hole with shell region limit (based on the part BoundingBox). To confirm it is a screw the SRE firstly make a taxonomic analysis of the part name, in this case the terminology is "EngineBolt". An algorithm concatenates the string, extract the attribute "bolt" then categorized the part as a screw which will help us the identify the Form Features (e.g. identifying the thread on the cylinder). Once the object potential classes have been identified, the Mate dataset is imported and loaded in the RAM the same way as for B-Rep data. Then by analyzing the position of the mates in each part the "shell of interest" are being identified. They are represented shell surfaces where potential contact or kinematic link may exist (symbolized by a set of color in Figure 2, the green belonging to plane shell, red to cylindric ones and blue the torus, b-spline being currently ignored). While done the second level, Kinematic level performs a semantic analysis on Mates Features. He first recognize the mate's origin entities (clutch side crankcase ) and the targeted entities (engine bolt) then creates the parent links between them (which can normally be deduced by only analysing the assembly construction tree, but when the tree is flat which is the case here, those precedence relation cannot be detected without logic rules). A second rules based algorithm analyses the position and generate the potential KeyPoint according to the nature of the Mates, their position in the part and the nature of the surrounding Shells. For the screw we generates that way three KP. A first Transition KP instantiated at the extremity of the thread which will determine the screw DOF according to its position. A second Interest KP is created to give the type of the joint (helical joint in this case because the identification of a hole as threaded determines screwing as a joining process to follow). A third and last Handling KP is generated to specify where the ratchet wrench tool should be position in order to execute the disassembly process.

The preparation time according to this method was estimated at five days, approximately 40 hours of preparation (human time). (missing : % KP detected)

## 5 Conclusion and future prospect

In this paper, an attempt to create a recognition system based on a semantic keypoints and heuristic rules was presented. Differently from existing methods, our heuristic approach provided reasonable results for very complex products with a high number of parts and providing much shorter preparation time. After this initial research work, a question remains unanswered, namely what strategies to adopt in the presence of *non-rigorous* or incomplete 3D assembly in other words how to reconstruct and generate the missing data. To answer this question and to improve our approach the exploration of data enrichment techniques such as ontological methods or model update methods through VR scene could be carried out in the future. Moreover, we could also include the development of new algorithms based on our heuristic approach to find close to optimal disassembly sequences. Other works will be carried out in order to consider the assembly reversed-sequence. Those works could raise other questions concerning the activation or deactivation of the constraints according to the assembly process and the potential identification of common behavioral class for sub-objects (e.g. using the same class of screws in several locations not previously defined in the assembly sequence).

## References

1. Author, F.: Article title. Journal 2(5), 99–110 (2016).
2. Y. Koren, The global manufacturing revolution: Product-process-business integration and reconfigurable systems, Wiley, Hoboken, N.J, 2010.
3. P. Nyhuis (Ed.), Wandlungsfähige Produktionssysteme, GITO-Verl., Berlin, 2010.
4. S. Saniuk, Personalization of Products in the Industry 4.0 Concept and Its Impact on Achieving a Higher Level of Sustainable Consumption
5. Carvalho, L.; von Wangenheim, A.: 3D object recognition and classification: a systematic literature review. Pattern Analysis and Applications, 22(4), 1243–1292, 2019. <http://doi.org/10.1007/s10044-019-00804-4>.
6. El-Mehalawi, M.; Miller, R.A.: A database system of mechanical components based on geometric and topological similarity. part I: representation. Computer-Aided Design, 35(1), 83–94, 2003. [http://doi.org/10.1016/S0010-4485\(01\)00177-4](http://doi.org/10.1016/S0010-4485(01)00177-4).
7. LS. Wierda, "Linking Design, Process Planning and Cost Information by Feature Based Modelling", Journal of Engineering Design, 1991, 2(1): pp. 3-19.
8. W. Van Holland, "Assembly features in modelling and planning," Ph.D. dissertation, Delft: Delft University of Technology, 1997.
9. B. Hasan - Assembly Design Semantic Recognition Using SolidWorks-API – 2016
10. B. Hasan - Features Extraction from CAD as a Basis for Assembly 2017
11. A.Neb - Automatic Generation of Assembly Graphs based on 3D Models and Assembly Features 2019
12. Carlos M. Costa, "Automatic Generation of Disassembly Sequences and Exploded Views from SolidWorks Symbolic Geometric Relationships", 2018
13. R. Spezialetti, "Performance Evaluation of 3D Descriptors Paired with Learned Keypoint Detectors", 2021
14. Miao, H.K., Sridharan, N., Shah, J.J.: CAD-CAM integration using machining features. Int. J. Comput. Integr. Manuf. 15, 296–318 (2002)