Content-based multi-criteria similarity assessment of CAD assembly models

Katia Lupinetti[a,b,*], Franca Giannini[a], Marina Monti[a], Jean-Philippe Pernot[b]

[a] Istituto di Matematica Applicata e Tecnologie Informatiche "Enrico Magenes", CNR Via De Marini 6, 16149 Genova, Italy
[b] Arts et Métiers, LISPEN EA 7515, HeSam, Aix-en-Provence, France

Keywords:
Assembly similarity evaluation
Multiple similarity criteria
3D assembly model retrieval
Partial and local similarity

A B S T R A C T

The use of Digital Mock-Up (DMU) has become mainstream to support the engineering activities all along the Product Development Process. Over the years, companies generate large databases containing digital models and documents related to their products. Considering complex products, the DMU can be composed of several hundred thousand parts assembled together in assembly trees containing tens of sub-assemblies, and representing several terabytes of data. The ability to retrieve existing models is crucial for the competitiveness of companies, as it can help to leverage existing solutions, results and knowledge associated with previous products. To speed up the access to this large amount of reusable information, CAD models search approaches have been proposed, including the so-called content-based search techniques which do not rely on metadata and data organization but exploit the implicit knowledge embedded in the models. As part of a system for the retrieval of CAD assembly models, this paper introduces a set of four measures to evaluate assembly similarities according to multiple criteria. These measures are combined to assess three different levels of similarity (local, partial and global). The local measure only considers the contribution of the parts that are similar in the compared assemblies, while partial and global measures take also into account the number of similar parts compared to the total number of parts in the query and in the target model. Moreover, an ad-hoc visualization interface has been designed to clearly highlight the different similarities and to allow a fast identification of the target models. The validation of the proposed method is discussed, the dataset used to this aim is provided with the specification of the adopted ground truth and some examples of the obtained results are shown.

1. Introduction

Today, being able to efficiently retrieve CAD assembly models from large databases has become an important issue. Indeed, over the years, companies accumulate a huge amount of Digital Mock-Up (DMU) of their products. For complex products, the size of the DMU can reach several hundred thousand parts assembled together in assembly trees containing tens of subassembly levels, and requiring several terabytes for the storing.

This large amount of 3D digitalized data represents a paramount source of knowledge and data which can be exploited to be more competitive on the market. This is particularly important also in the context of *Industry 4.0*, which aims to improve the digital value chain (i.e. how products or solutions are brought from concept to delivery) but also to reduce stopping conditions of products/systems and to better design new ones through the concept of the digital twin combining digital data with those acquired from real-world working products [1]. Thus, analyzing the behavior of specific products and systems can provide good hints to foresee the behavior of similar ones allowing to take the correct countermeasures to guarantee longer and more efficient working conditions. To track and control data related to a product during the Product Development Process (PDP), modern CAD systems are integrated with Product Data Management (PDM) and Product Lifecycle Management (PLM) systems. These systems manage search based on textual efficiently, while they offer limited capabilities for the search according to some specific geometric characteristics [2]. To overcome these limitations, content-based algorithms represent a solution that can be integrated in these systems to filter and retrieve relevant models according to criteria that usually are not stored in text files. Thus, new tools have to be developed, such as tools for the comparison and retrieval of CAD models. Indeed, the retrieval of similar models from company and

* Corresponding author at: Istituto di Matematica Applicata e Tecnologie Informatiche "Enrico Magenes", CNR Via De Marini 6, 16149 Genova, Italy.
E-mail address: katia.lupinetti@ge.imati.cnr.it (K. Lupinetti).

supplier databases provides a way to access to knowledge gained from previous designed solutions [3]. This is interesting not only when considering the design of new products, but in all the cases when it is useful to consider previous advances and results instead of starting from scratch. For example, to speed up the definition of the assembly process planning of a certain product, it would be valuable to have access to existing assembly instructions of similar models. The similarity evaluation provides benefits in other engineering activities, such as for the identification of interchangeable parts among different projects to reduce management and manufacturing costs [4] or in the standardization of industrial components. The ability to evaluate the similarity between two assembly models plays a central role in the development of a retrieval approach. However, it is not an easy task as it involves several issues for which traditional text-based and shape-based object retrieval methods are not sufficient.

A first issue concerns the plurality of the *similarity criteria* according to which two models can be considered similar. This strongly relies on user's purpose. Deshmukh et al. [5] describe different application scenarios and present the most suitable similarity criteria to be satisfied for each application. For instance, if a designer wants to reuse a digital model, he/she can start with a rough query (i.e. a simple CAD model with few details) to retrieve a more detailed similar one, which can then be adapted and modified according to the new requirements. In this scenario, the overall shape could be one of the salient characteristics to evaluate the similarity. Another scenario requires the retrieval of similar models to get access to existing product information (e.g. simulation results, assembling strategies or failure reports). In this case, only considering the shape to evaluate the similarity of the products could be limiting, while evaluating the similarity according to the part arrangement (e.g. positions and joint types) point of view could be more meaningful. For instance, such a consideration might be discriminatory when considering the selection of bearings components. Indeed, according to the load they must support, bearings are made of several occurrences of balls or rollers (of cylindrical or conical shape) arranged in different ways. Here, not only it is important to consider the shape of the constitutive elements, but also the way they are organized and linked to each other to support specific load conditions. In addition, other useful characteristics exist and can be used to evaluate the similarity (e.g. products functions, dimensions, required manufacturing processes or materials, design intent). Thus, a good similarity assessment technique should be able to consider multiple similarity criteria.

A second issue concerns the *availability of the information* necessary to compare models according to these criteria. Actually, the requested information is not always directly available as it is not coded into the CAD models, and not even in associated systems, as in PDM or in PLM systems. In this case, the similarity assessment technique should embed specific reasoning mechanisms to extract information implicitly encoded in the CAD models. This is not an easy task considering that the CAD models may integrate simplified representations (e.g. balls of a bearing not represented or simplified by a torus shape) and contain inconsistencies (e.g. non physically possible configurations where solids are intersecting) possibly resulting in misleading interpretations.

A third issue concerns the plurality of the *similarity levels* according to which the similarity can be assessed. Indeed, retrieval methods have to deal with *global* and *partial* similarities, where the partial similarity can be achieved as part-in-whole (i.e. the query model is contained in the retrieved object) and whole-to-whole partial matching (i.e. a subpart of the input object is similar to a subpart of a retrieved object) [6,7]. This is particularly important considering that a product, and consequently its associated CAD models, can be decomposed in a set of elementary functions (e.g. speed reduction, specific movement transformation mechanisms, sealing solutions), realized by a subset of CAD models, whose identification and exploitation can be of high interest when considering scenarios involving products with similar functions. An example of the different similarity levels is depicted in Fig. 1. The CAD assembly model M_1 is globally similar to the model M_2, since they have analogous parts. The model M_1 (resp. M_2) is considered partially similar (i.e. part-in-whole) to M_3 (resp. M_4), since it is completely included in the second one. Finally, models M_3 and M_4 are locally similar (i.e. whole-to-whole by partial similarity), since they share similar parts. Ideally, content-based retrieval methods should be able to tackle those three similarity levels to cover most of the similarity assessment scenarios.

A fourth issue originates from the two previous ones. Due to the multiple criteria and levels characterizing the similarity between two CAD assembly models, it is not straightforward *to properly present to the user the ranked results* of a given query according to specific criteria. Thus, ad-hoc visualization interfaces must be devised to clearly highlight the different similarities and to allow a fast identification of the target models, i.e. models which are the

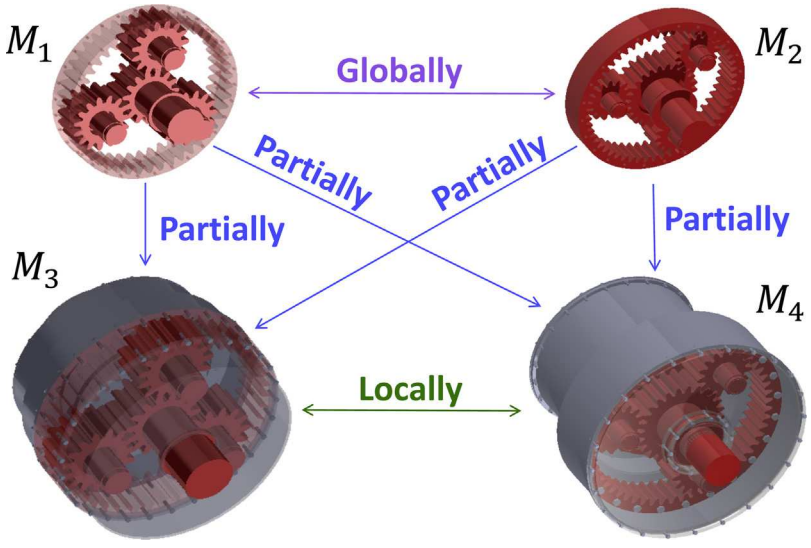


Fig. 1. Example of possible similarity levels for the retrieval of CAD assembly models (local, partial and global).

most similar to a query model with respect to similarity criteria and levels.

From this analysis, it follows that the current search methodologies are not sufficient to support designers in the different stages of the PDP. Actually, existing techniques need to be enhanced and integrated with systems that are able to evaluate digital models according to their content and associated implicit information. The contents to be evaluated must be adjusted according to the search scenarios adopted by the designers, and according to multiple criteria able to cover the search needs all along the PDP. Finally, a multi-criteria retrieval system should be able to detect also partial similarities and to extract the required information automatically to fully support users and avoid manual integration of annotations.

To address most of these issues, a system for the retrieval of globally or partially similar assembly models has been developed to analyze the similarities according to different search criteria that can be convenient for designers. The work described in this paper completes and extends the research presented in [8] which introduces the overall architecture of the retrieval system. The novel contribution of this paper is manifold: (i) illustration of how the criteria, according to which two assemblies may be considered similar, are handled to drive the mapping process; (ii) management of refined measures to quantify at which extent two assembly models are similar according to different similarity levels (local, partial and global) with respect to the query similarity criteria; (iii) set up and publication of a dataset of CAD assembly models for the validation and comparison of retrieval methods; (iv) a sound validation of the proposed system using an ad-hoc visualization interface and the computation of its precision and recall. The rest of the paper is organized as follows. Section 2 reviews the most pertinent related works. Section 3 reports some details on the adopted framework for the comparison of assembly models, while Section 4 focuses on the creation of the adopted assembly descriptor. Sections 5 and 6 detail the similarity criteria and the proposed measures for their evaluation. Finally, Section 7 presents the used dataset and discusses some results obtained using our prototype software. Section 8 ends the paper providing conclusions and future steps. Some additional details and definitions are provided in the appendixes.

2. Related works

The similarity assessment of 3D objects has seen a copious definition and development of content-based 3D model retrieval methods [6,9–11].

To evaluate the similarity among the models and thus to allow the retrieval of similar ones, these methods usually make use of pre-computed shape descriptors (e.g. shape-based methods [11–17]) or signatures that characterize distinctive elements of the models (e.g. feature-based methods [18,19] and topology-based methods [20–23]). Some of them can also detect partial similarities, i.e. models that are similar only for a subset of their shape [24–28].

Considering CAD assembly models, there exists retrieval methods that describe assemblies just through the shape descriptors of their parts. One of the most popular and adopted shape descriptor is the shape distribution proposed by Osada et al. [13], which has been used by Renu and Mocko [29], Wang et al. [30] and Zhang et al. [31]. These approaches strongly rely on the shape information and do not use any assembly relationships (e.g. geometric constraints, kinematic links or parts arrangement). Thus, these methods generally do not fit assembly retrieval purposes completely, such as when a user tries to recall assemblies from the manufacturing point of view. In this case, the mating relationships among the parts of the assembly are crucial for the definition of the similarity and shape-based as well as part-based methods neglect this aspect.

The relationships between the parts of an assembly model are usually represented by using graph-based descriptors. For instance, Miura and Kanai [32] represent models by attributed graphs, which encode structural information and other data (e.g. contacts, interferences and geometric constraints). Tao and Huang [33] use component attributed relational graphs, where arcs represent the adjacency relationships between two components encoding the types of the involved surfaces (e.g. planar or cylindrical surfaces) and the connection relations (e.g. screw connection, pin joint, key joint, rivet joint). Deshmukh et al. [5] propose a flexible retrieval system exploiting the data present in CAD models represented by mating graphs. Despite the fact that these methods describe assembly models at a local level, their matching approaches allow recognizing just global similarities among assembly models. Moreover, if some pieces of information are missing, they must be made explicit by the user.

A more complete system able to detect also partial similarities has been proposed by Chen et al. [34]. Their assembly descriptor considers different information levels including the topological structure, the relationships between the components, as well as the geometric information. Anyhow, it assesses partial similarities by exploiting the organization of the product in sub-assemblies. As observed by the authors, this practice impacts the ability of their method to detect identical models presenting differences in the organization of their assembly structure.

Recently, Han et al. [35] proposed a retrieval system using a semantic representation of assembly models that allows capturing both the shape and the design intent of an assembly model. In this method, the assembly model is described semantically using an ontology created while analyzing the shape of the parts, the assembly constraints and the function. Despite the fact that this method presents an interesting novelty on the proposed representation, the adopted matching procedure allows evaluating only the global similarity.

Beside traditional analysis of object similarities, there is room for deep learning techniques [28,36–38], which are capable to detect similarities despite small geometry variation in the shape of the models or in their positioning. Anyhow, working on CAD assembly models, the main limitation of these methods regards the assembly data considered during the training. Indeed, current approaches focus on characteristics that can be extracted from mesh representations with the results that they are suitable in the evaluation of the shape similarity but weak in other design characteristics. For instance, Qi et al. [28] segment 3D digital objects obtaining a decomposition that can be very far from the one designed by the user in the CAD models. Then, such decomposition can be used to detect local shape similarity but it is not suitable to evaluate the models according to the design intent point of view. As a consequence, this type of approach is not yet ready to satisfy the needs arising from the previously introduced search scenarios.

Summarizing, even if content-based 3D model retrieval is an active topic, most of the existing works focus on single parts, and the methods that consider assembly models can hardly manage a variety of similarity criteria together with the possibility to assess the similarity at different levels, i.e. partial, local and global similarities. To overcome these limits, this paper aims at introducing measures to evaluate the various assembly similarity levels according to multiple criteria.

3. Assembly retrieval framework

The proposed retrieval process is a two-step procedure: a proper CAD assembly descriptor, called Enriched Assembly Model

(EAM) [8], is first created and then used for the models comparison and similarity assessment. The general framework of our system is illustrated in Fig. 2. To be independent of the information stored in property CAD formats, CAD models are assumed to be represented in the standard format STEP AP 203/214. Theoretically, STEP AP 214 standard supports the representation and exchange of the kinematic relationships between components of an assembly model and its constraints. However, most of CAD systems do not contain the latter ones and generate files that do not incorporate kinematic relationships and constraints. For this reason, the developed framework allows as input both the formats but rely just on the information of the STEP AP 203. The EAM descriptor is computed both for the query model, i.e. the model according to which the similarity has to be assessed, and for all the CAD assembly models of the inquired dataset. After the user selects a folder including the EAMs of the target models he/she desires to compare, the EAM of the query is compared with all the selected EAMs. The results are stored and provided to the user who can rank the compared CAD models according to the level of similarity he/she is interested in.

The framework includes both real-time and batch processes. The batch process computes the complete EAM descriptors for all the CAD assembly models in the dataset. Once the data have been computed, the EAMs are stored in JavaScript Object Notation (JSON) format. This format is completely language independent and is based on two universal data structures: a collection of name/value pairs and an ordered list of values. These properties make JSON an ideal data-interchange language. Thus, the EAM is translated in a list of nodes with several attributes specified by a key-value and a list of arcs defined by their source and target nodes with their attributes. The real-time processes compute partial EAM descriptors for the query including only the data that are pertinent to the user-specified similarity criteria, and then perform the comparison.

The creation of the EAM relies on both explicit information directly available in the CAD models, and implicit information extracted through dedicated procedures. The extracted data provide four types of information: structure, statistic, interface and shape. Differently from [8], in this work the EAM has been enriched to manage also configurations affected by errors on the interfaces between components, as it will be described in Section 3.3. The EAM is implemented as an attributed multi-graph structure, i.e. a graph structure that allows multiple arcs between a pair of nodes and attributes for nodes and arcs. Using this structure, a sub-graph isomorphism is applied to detect the local similarity among the models. Exact approaches to solve this problem are well known to be NP-hard, then a heuristic procedure is adopted to solve it.

The entire framework has been developed as plug-in for the commercial CAD system SolidWorks®. The procedures are developed using Microsoft Visual C# 2015 and the set of Application Programming Interface (API) of SolidWorks® is used to manage the information present in the CAD assembly models. The framework can be integrated in any CAD system providing the functionalities for accessing the B-rep information, for the computation of the parts volume and bounding box, face area, and of the non-regularized intersection among parts [39].

Sections 3.1–3.4 provide details on the stored information, which is used to drive the similarity assessment process according to the multiple criteria and similarity measures as described in Sections 5 and 6. Formalization of the domain range of the attributes related to nodes and arcs is provided in Appendix A. Section 3.5 explains how that information is encoded in an attributed multi-graph representation, and Appendix B details the attribute functions. Section 3.6 formalizes the matching procedure.

3.1. Structure information

The structure information characterizes the hierarchical structure of an assembly, i.e. how the parts are gathered into sub-assemblies. This information is available in CAD models and preserved also when the models are stored in STEP format. It is represented in the EAM by nodes and directed arcs. The root node corresponds to the entire assembly model, intermediate nodes indicate sub-assemblies and the leaves represent the parts constituting the assembly. Directed arcs between nodes represent the relation “made-of” between the components of the assembly. Two attributes characterize nodes representing parts defining the type of component (*CompType*), e.g. nut, screw or bearing, and the arrangement in the 3D space (*PatternType*) of a set of repeated parts. Lastly, an attribute indicating all the patterns in the model can be associated with the root node (*PatternList*). The complete attributes’ domain values are listed in Appendix A.

3.2. Statistic information

The statistic information has been designed to ease the filtering of large datasets and reduce the number of models to be compared. Statistics for single parts include: (1) percentage of the overall model area covered by surfaces of a specific type (i.e. planar, cylindrical, spherical, toroidal and free-form); (2) number of maximal faces (i.e. adjacent faces sharing the same underlying surface correspond to a maximal face) of a specific type of surface. The use of such information allows discarding directly some parts, thus reducing the number of candidates for the matching process. Thus, the generic statistic attribute for parts has values in the set *PartStat*, which gathers together five pairs of values in the range $([0,1] \times \mathbb{N})$, one pair for each type of surface (i.e. planar, cylindrical, spherical, toroidal and free-form). Statistics for assemblies are

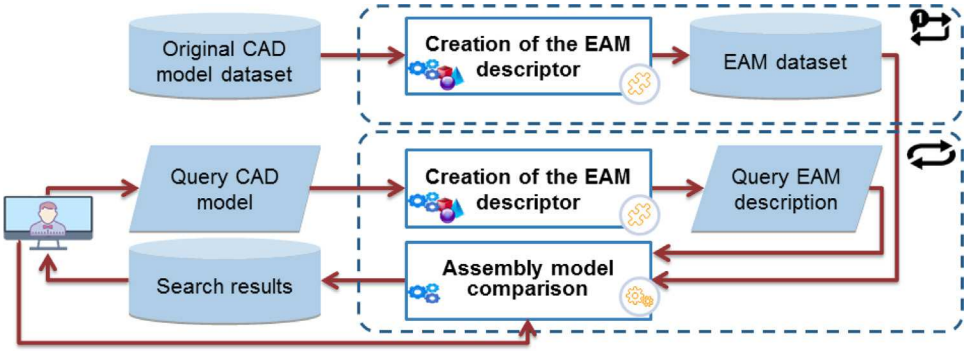


Fig. 2. Assembly retrieval framework.

represented by an attribute taking value in *AssemblyStat*, which includes the number of patterns of each of the four pattern types.

3.3. Interface information

The interface information describes the relationships between parts of an assembly model regardless its structure. These relationships are expressed by contact information and joint information. In most configurations, contacts between two parts can be of type Surface, Curve or Point (Fig. 3a). However, sometimes, unrealistic configurations may be present where two parts share a volume, i.e. they intersect each other (Fig. 3b–d). These unrealistic arrangements are generally due to the inaccurate positioning of assembly components, or to numerical issues in importing files, or made on purpose by designers (as the intersection among thread screws and nuts, or flexible parts, as springs, seals and insulating parts or designing parts that will be assembled by shrink-fitting). Thus, some of these configurations can be interpreted as an imprecise design that will be solved during the analysis of the tolerances, while others as a deliberate artifact to reflect some conventional meaning.

In the EAM, contact information is represented by an attributed arc that specifies the type of contact (*ContactType*) and the degree of freedom (*DOF*). Details on the reasoning process to get this information are reported in Section 4.2.1. The DOF is computed identifying the set of allowed translations (*Tra*) and rotations (*Rot*) between the parts in contact. Here, both sets are expressed by normalized vectors characterizing either a translation direction or a rotation axis according to the global reference frame. In the current version of the system, DOF is indicated only for contacts of surface type and it is computed by considering the surface normal information of the faces in contact. Details on this reasoning are reported in Section 4.2.1.

Joint information describes the motion resulting from all the contacts between two parts. Its characterization involves attributes that define the type of joint (*JointType*), i.e. the type of contacts involved in the joint, and the degree of freedom (*DOF*) resulting from all the constraints imposed by the contacts. Differently from the type of contact, the type of joint can be also mixed. The so-called Mixed type is used to characterize a joint derived from contacts of different types.

The type of contact, the type of joint and the degree of freedom are not explicitly encoded in the CAD models and are computed analyzing the results provided by the interference evaluation functionality provided by the adopted B-Rep modeling kernel, as described in [40]. Additional details are reported in Section 4.2.2.

3.4. Shape information

The shape information gathers two attributes used to characterize the shape and size of the parts. The first attribute characterizes the shape (*Shape*) using 3D spherical harmonics defined by a histogram of 544 bins [41]. While the second attribute characterizes the size (*Size*) and it gathers the volume and the surface area of a part.

3.5. Graph representation

Nodes, arcs and attributes described in the previous subsections, whose values are reported in A, contribute to the creation of an attributed multi-graph that represents an EAM descriptor. Let $G(N, A, \Phi_N, \Phi_A)$ be an attributed multi-graph representation of an EAM descriptor, where N is the set of nodes, A is the set of arcs, and Φ_N and Φ_A are respectively the node and arc attribute functions. Different types of nodes and arcs are defined according to the different types of information they represent. In particular, $N = N_P \cup N_A$ and $A = A_S \cup A_C \cup A_J$, where N_P and N_A are the sets of nodes associated respectively with parts and sub-assemblies, while A_S, A_C and A_J are the sets of arcs representing respectively the assembly hierarchical structure, the contacts between parts and the joints between parts. The node attribute function is defined as:

$$\Phi_N : \begin{vmatrix} N_P \cup N_A & \rightarrow & T_{N_P} \cup T_{N_A} \\ n & \mapsto & \Phi_N(n) = \begin{cases} \Phi_{N_P}(n) \text{ if } n \in N_P \\ \Phi_{N_A}(n) \text{ if } n \in N_A \end{cases} \end{vmatrix} \quad (1)$$

where the parts and sub-assemblies functions (i.e. $\Phi_{N_P}(n)$ and $\Phi_{N_A}(n)$) are shown in Appendix B.

The structure arcs in A_S have no attribute. As a consequence, and similarly to what has been defined for nodes, the arc attribute function sets up the attributes of the elements of A_C and A_J as follows:

$$\Phi_A : \begin{vmatrix} A_C \cup A_J & \rightarrow & T_{A_C} \cup T_{A_J} \\ a & \mapsto & \Phi_A(n) = \begin{cases} \Phi_{A_C}(n) & \text{if } a \in A_C \\ \Phi_{A_J}(n) & \text{if } a \in A_J \end{cases} \end{vmatrix} \quad (2)$$

where the definition of contact and joint arc functions (i.e. $\Phi_{A_C}(n)$ and $\Phi_{A_J}(n)$) are shown in Appendix B.

Fig. 4 illustrates an example of the EAM graph structure created from a CAD model and enriched with semantic information extracted by the geometric reasoning algorithms. For readability purposes, only a subset of the attributes is shown. The double-line nodes correspond to parts belonging to regular patterns, while single line-circled nodes are associated with the other parts. The labels of the nodes represent the type of component. The straight arcs connect two parts that are in contact, and the associated label indicates the DOF. If the label is not present, then the contact is of type UnSolved. The wavy arcs indicate a line contact, thus there is no label specifying the corresponding degree of freedom.

3.6. The matching procedure

Using attributed graphs as assembly descriptors, the problem of searching the local similarity between two models is equivalent to finding the Maximum Common Sub-graph (MCS) between the two graphs, namely G_q and G_k, representing respectively the *query* and

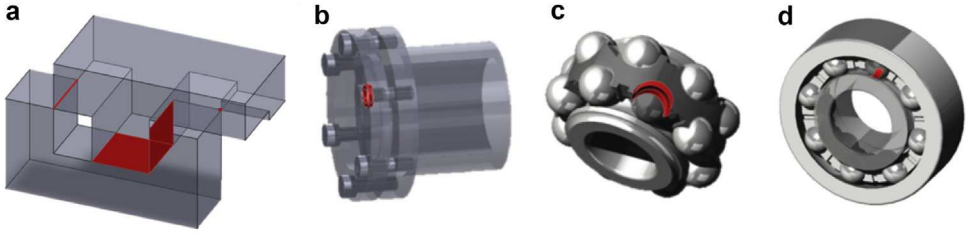


Fig. 3. Possible contacts between parts of a CAD assembly model: (a) contacts of type Surface, Curve and Point, (b) UnSolved contact type, (c) volumetric interference solved as Surface contact type, (d) volumetric interference solved as Curve contact type.

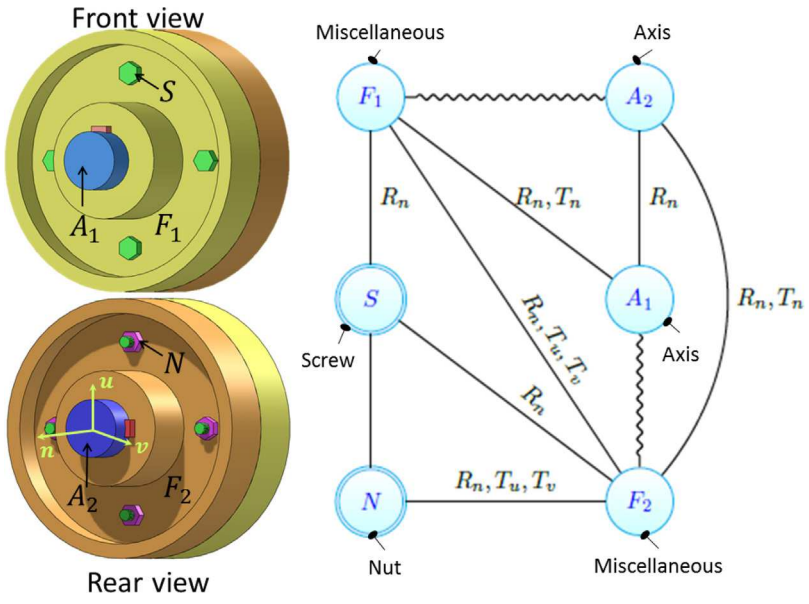


Fig. 4. Example of a partial EAM descriptor, associated with the CAD model of a flange [8].

target models. Among the different techniques to solve the MCS problem [42], in this framework, the MCS problem is transformed into a Maximum Clique (MC) problem [43]. Therefore, a clique (i.e. a complete subgraph where each pair of nodes is connected) has to be detected in a suitable *association graph* derived from G_q and G_k.

In an association graph, nodes represent pairs of *similar nodes* between G_q and G_k, whereas arcs identify *similar relationships*. Here, c_N and c_A respectively refer to the node and arc similarity criteria and are described in Section 5. The association graph is not unique and its definition depends on the similarity criteria. Thus, once c_N and c_A are specified, there exist a unique association graph G_{q,k,c_N,c_A}. The maximum cliques in G_{q,k,c_N,c_A} represent the common sub-graphs between G_q and G_k according to the criteria c_N and c_A. The generic r^{th} clique in the graph G_{q,k,c_N,c_A} is expressed as $(C_{q,k,c_N,c_A})_r \subseteq G_{q,k,c_N,c_A}$; the set of all the cliques for the association graph G_{q,k,c_N,c_A} is denoted as $D_{q,k} = \{(C_{q,k,c_N,c_A})_r | 1 \leq r \leq N_{qk}\}$, where N_{qk} is the number of maximum cliques in the association graph G_{q,k,c_N,c_A}.

Since the MC problem is NP-hard, the proposed framework exploits a heuristic method based on the simulated annealing process to detect the maximum cliques in an association graph [27]. The method, exploits a property of the cliques that link the number of their nodes (k) to the number of their arcs (h) by the relation $h = \frac{k(k-1)}{2}$. Then, a subgraph S represents a clique if and only if $h - \frac{k(k-1)}{2} = 0$. In this way, the MC problem is equivalent to solve an optimization problem, where $f(S) = h - \frac{k(k-1)}{2}$ is the function to be minimized. The use of this technique requires high attention on tuning the simulated annealing parameters, which affect the efficacy as well as the efficiency of the system.

Then, the simulated annealing process represents the core of the proposed matching procedure (shown in Algorithm 1). It creates the attributed multi-graph (G_q) of the query model (described in Algorithm 2). Then, for each JSON file representing an EAM in the folder chosen by the user, its attributed graph (G_k) is read and the corresponding association graph G_{q,k,c_N,c_A} is generated according to the selected similarity criteria c_N and c_A. A lower (lBound) and an upper bound (uBound) are specified representing the minimum and the maximum number of nodes of the desired clique. Since the nodes in the clique represent the similar parts among the two graphs G_q and G_k, the upper bound cannot be greater than N_q and N_k, where N_q, N_k represent the number of nodes of G_q and G_k respectively. A subgraph S of G_{q,k,c_N,c_A} is initialized by taking the first lBound nodes in G_{q,k,c_N,c_A} ordered by their degree. Then a clique is determined by the simulated annealing process and its function f is computed. Finally, the similarity measures (see Section 6) are computed.

Algorithm 1. Matching procedure based on simulated annealing

```
1:  procedure MATCHING(QueryModel, c_N, c_A, Folder)
2:    G_q = CreateAttributetedGraph(QueryModel)
3:
4:    for each file in Folder
5:      G_k = GetAttributedGraph(file)
6:      G_{q,k,c_N,c_A} = CreateAssociationGraph (G_q, G_k, c_N, c_A)
7:      lBound = 3
8:      uBound = min(N_q, N_k)
9:      S = Initialize(G_{q,k,c_N,c_A}, lBound)
10:     cliqueLenght = NumberOfNodesOf(S)
11:     Clique = null
12:     F = 0
13:     while cliqueLenght ≤ uBound &F == 0do
14:       Clique = SimulatedAnnealing(G_{q,k,c_N,c_A}, S)
15:       F = f(S)
16:       if F==0 then
17:         Clique = S
18:       end if
19:       S = AddNodeToClique(G_{q,k,c_N,c_A}, S)
20:       cliqueLenght = NumberOfNodesOf(S)
21:     end while
22:     Measures ← ComputeMeasures(Clique)
23:   end for
24:   Return Measures
25: end procedure
```

4. Enriched Assembly Model creation

The EAM creation starts reading a CAD model in STEP file format by exploiting the SolidWorks® kernel. First, the nodes with some of their attributes and the arcs of the EAM structural layer are created as detailed in Section 4.1. The *ContactsComputation* and *JointsComputation* procedures analyze the interactions between the parts to identify the contacts and the joints corresponding to the EAM interface layer (see Section 4.2). The repeated components and their regular patterns are detected by using the

PatternArrangementComputation procedure (described in Section 4.3). Then, statistics' numerical values linked with the entire assembly are stored as attributes by the *AssemblyStatisticsComputation* procedure (described in Section 4.4). Finally, to overcome issues risen from simplified shapes of parts, components are classified by the *ComponentClassification* and the *ShapeAndContextBasedPartClassification* procedures. These modules are described in Section 4.5.

Algorithm 2. The EAM creation algorithm

```
1:  procedure CreateAttributedGraph(StepModel)
2:    Root = GetRoot(StepModel)
3:    n_Root = New SubAssembly(Root)
4:    N_A ← n_Root
5:    Q ← RootA.getChildren
6:
7:    //Nodes and structure creation (Section 4.1)
8:    while Q ≠ ∅ do
9:      component = GetFirstElementIn(Q)
10:     Remove component from Q
11:     n_f = GetFatherNodeOf(component)
12:     if component.getChildrenNumber ≠ 0 then
13:       Q ← component.getChildren
14:       n_c = New SubAssembly(component)
15:       N_A ← n_c
16:     else
17:       n_c = New Part(component)
18:       N_P ← n_c
19:       Φ^PS_Np(n_c) = PartStatisticsComputation(component)
20:       Φ^Si_Np(n_c) = SizeValuesComputation(component)
21:       Φ^Sh_Np(n_c) = 3DSphericalHarmonicsComputation(component)
22:       Φ^CT_Np(n_c) = ShapeBasedPartClassification(component)
23:     end if
24:     A_S ← (n_f, n_c)
25:   end while
26:
27:   //Relation creation (Section 4.2)
28:   [A_C, Φ_Ac] = ContactsComputation(N_P)
29:   [A_J, Φ_Aj] = JointsComputation(N_P)
30:
31:   //Pattern creation (Section 4.3)
32:   Pattern_List = PatternArrangementComputation(N_P)
33:   Φ^PL_NA(n) ← Pattern_List
34:
35:   for each i in {(patType_i, RP_i), ∀ i ∈ {1, ..., N_p}}
36:     for each part in RP_i
37:       n = GetNodeOf(part)
38:       Φ^PT_Np(n) = patType_i
39:     end for
40:   end for
41:
42:   //Assembly statistics computation (Section 4.4)
43:   Φ^AS_NA(n_Root) = AssemblyStatisticsComputation(Pattern_List)
44:
45:   //Part and component classification module (Section 4.5)
46:   for each n in N_A
47:     Φ^PL_NA(n) = ComponentClassification(n)
48:   end for
49:
50:   for each n in N_P
51:     Φ^CT_Np(n) = ShapeAndContextBasedPartClassification(n)
52:   end for
53:
54:   Return G(N, A, Φ_N, Φ_A)
55: end procedure
```

4.1. Nodes and structure creation

The creation of the nodes associated with parts (N_P) and subassemblies (N_A) and the set of arcs related to the structure (A_S) starts off traversing the assembly model. The assembly is traversed by an iterative procedure that for each sub-assembly in the model reads its sub-components, i.e. its "children". As shown in Algorithm 2, the procedure takes as input the root of the assembly *RootA*, creates its corresponding assembly node and initializes a queue with the children of the root of the assembly. Then, for each element of the queue without children a part node is created, otherwise a sub-assembly node is created together with structure type arcs between the node and the part nodes corresponding to the children. During the creation of part nodes, some node attributes are computed and stored into the EAM. First, the values of the *PartStat* attribute ($\Phi^{PS}_{N_p}$) is computed by the *PartStatistics-Computation* procedure that, by using the SolidWorks® API functionalities for reading the B-rep data (e.g. type of surface of the faces of the parts) and for computing the related areas, returns the percentage over the overall area of the area of the faces of type planar, cylindrical, conical, spherical, toroidal and free form; and the number of maximal faces of each surface type. The $\Phi^{Si}_{N_p}$ representing the value of the *Size* attribute is computed still using the SolidWorks® functionalities. Then, the value of the *Shape* attribute ($\Phi^{Sh}_{N_p}$) representing the 3D spherical harmonics associated to a part is computed by the software tool defined in [44] (with its default parameters).

Finally, the *ShapeBasedPartClassification* procedure evaluates the attribute function $\Phi^{CT}_{N_p}$ by classifying parts according to their shape. The classification is based on a learning process applied on a collection of descriptors of the parts as defined by Rucco et al. [45]: 3D spherical harmonics; shape distribution (with D2 measure); inner distance; size values, i.e. surface area and volume; and proportions among the minimum bounding box dimensions. These descriptors have been chosen because, from the analysis of Jayanti [46], they are considered the most suitable to discriminate mechanical components. In the proposed framework, the D2 shape distribution and the inner distance are computed respectively by the procedure of [41] and [47] on a mesh representation of the part computed through SolidWorks® API; while the surface area, volume and proportions among the minimum bounding box dimensions are obtained using SolidWorks® API. This classification is then improved by the *ShapeAndContextBasedPartClassification* procedure (described in Section 4.5) which analyses the surrounding context where parts are placed within the assembly.

4.2. Relation creation

The procedures to compute the relations among the parts of a model generate the information belonging to the interface layer of the EAM. More precisely, they originate the arcs representing the contacts and the joints between parts (A_C and A_J) and their set of attributes (Φ_{A_C} and Φ_{A_J}). The process takes as input the B-Rep model and, as described in [40], it relies on the following hypotheses: (i) models include only rigid parts; (ii) contacts do not change over time; (iii) faces involved in the contact are mainly associated to analytic surfaces. This process is based on two main procedures: the detection of the contacts and the computation of their DOFs (Section 4.2.1) and the computation of the resulting motions (Section 4.2.2).

4.2.1. Contacts computation

This procedure detects and classifies the type of contacts between each pair of parts. It is described in the Algorithm 3 where the generic value of the attribute $DOF \times ContactType$ associated with the arc $a_{i,j} = (n_i, n_j)$ is indicated as $t_{i,j}$, while A^{ij}_C and T^{ij} represent

respectively the set of contact arcs between the nodes n_i and n_j and the set of their attributes.

Algorithm 3. The component relationship detection algorithm

```
1:  procedure ContactsComputation N_P
2:    for each n_i, n_j ∈ N_P
3:      I = GetIntersection(n_i, n_j)
4:      if I ≠ ∅ then
5:        if Volume(I) == 0 then
6:          [A_C^ij, T^ij] = GetContactArcsAndAttributes(I, n_i, n_j)
7:          A_C ← A_C^ij
8:          T_Ac ← T^ij
9:        else
10:         A_C ← a_i,j = (n_i, n_j)
11:         T_Ac ← t_i,j = TryToSolveVolumetricIntersection(I, n_i, n_j)
12:       end if
13:     end if
14:   end for
15:   Return A_C and T_Ac
16: end procedure
```

First, the contacts are detected by the procedure *GetIntersection* that exploits the SolidWorks® API, which returns the elements (i.e. vertices, edges and faces) corresponding to the overlapping surface or to the intersecting volumetric portion between two solids. If the procedure *GetIntersection* returns a volumetric intersection not empty and with null volume, then the two parts are in contact and their contact arcs and the related attributes are computed by the *GetContactArcsAndAttributes* procedure detailed by Algorithm 4. Otherwise, if the intersection is not empty and its volume is not null, an interference between parts exists. In this case a further analysis is required to possibly deduce the original configuration and compute the DOF. This investigation is fulfilled in the *TryToSolveVolumetricIntersection* procedure described in Algorithm 5.

The *GetContactArcsAndAttributes* procedure (Algorithm 4) takes as input two nodes n_i and n_j and their intersection I and returns the set of contact arcs for the pair of nodes n_i and n_j (A_C^{ij}) and their set of attributes (T^{ij}).

Algorithm 4. The component relationship detection algorithm

```
1:  procedure GetContactArcsAndAttributes I, n_i, n_j
2:    A_C^ij set of arcs
3:    T^ij set of attributes
4:
5:    if ∃f ∈ I then
6:      for each f ∈ I
7:        A_C^ij ← a_i,j = (n_i, n_j)
8:        T^ij ← t_i,j =(ParametersOf(f), "Surface")
9:      end for
10:   end if
11:
12:   else if ∃e ∈ I then
13:     A_C^ij ← a_i,j = (n_i, n_j)
14:     T^ij ← t_i,j =(∅, "Curve")
15:   end else if
16:
17:   else if ∃v ∈ I then A_C^ij ← a_i,j = (n_i, n_j)
18:     T^ij ← t_i,j =(∅, "Point")
19:   end else if
20:
21:   Return [A_C^ij, T^ij]
22: end procedure
```

Let P_i and P_j be the parts in the assembly model corresponding to the pair nodes (n_i, n_j). For each face, if exists, belonging to the interface, its corresponding DOF is assigned according to Table 1 by using the parameters which define the underlying surface of the face, where R indicates a rotation, T a translation, the subscripts u, v and n the vector along which the rotations/translations are allowed. Here, R_{u+O} corresponds to a rotation about the vector formed by the directional versor u applied in the point O. These parameters are set by the function *ParametersOf* in Algorithm 4.

The DOF assignment for curve and point contacts requires more computations. Indeed, in this case, the intersection defines just a line or a point (with their parameters) in the 3D space of the assembly model. This information is not sufficient to determinate the blocked and free motions. For instance, considering the edge contact in Fig. 5, the two parts can translate along x and y axes and rotate about y and z axes. To deduce this information, it is necessary to know the parameters of the surfaces that generate the contacts, in this example the normal of the planar face. This detection requires to project the curve obtained from the non-regular intersection on each faces of the parts involved in the contact to identify which faces produce the contact. The complexity of this operation is rather high, and considering the frequency of only a curve and a vertex contact in mechanical objects, this framework considers the assignment of DOF just in case of face contact.

In case P_i and P_j define a volumetric intersection, the *TryToSolveVolumetricIntersection* procedure (Algorithm 5) is invoked returning $t_{i,j}$. As in the case of contact interface, the DOF is assigned for configurations which originate a surface contact, while for curve and point contacts no DOF is assigned. So far, only the intersection of a spherical part S with a part P has been faced, while in the other cases, the *ContactType* attribute assumes the value "UnSolved".

Algorithm 5. The component relationships detection algorithm

```
1:  procedure TryToSolveVolumetricIntersection I, n_i, n_j
2:    P_i = GetPartOf(n_i) and P_j = GetPartOf(n_j)
3:    if P_i is sphere or P_j is sphere then
4:
5:      if ∃f_i, f_j ∈ I such that their surfaces are two spheres
6:          with the same radius of S then
7:          Return t_i,j = (ParametersOf(entity), "Surface")
8:      end if
9:
10:     if ∃f_i ∈ I whose surface is a torus or a cylinder with
11:         the same radius of S then
12:         Return t_i,j = (∅, "Curve")
13:     end if
14:
15:     if ∃f_i ∈ I whose surface is a torus or a cylinder
16:         with radius different from the one of S then
17:         Return t_i,j = (∅, "Point")
18:     end if
19:
20:   end if
21:
22:   Return t_i,j = (∅, "UnSolved")
23: end procedure
```

Fig. 6 reports examples of intersections with a spherical part, where in Fig. 6a the contact is "solved" in a vertex, an edge in Fig. 6b and a face contact in Fig. 6c.

These assumptions aim at solving some volumetric intersections to improve the ability of the retrieval system to manage some errors automatically without human intervention, but the problem of handling generic volumetric intersections still remains an open issue.

4.2.2. Joints computation

According to the mechanical analysis [48], the contacts between two parts form a parallel kinematic chain and the DOF resulting from all the contacts can be computed composing the

Table 1
DOF values according to the face contact type.

Face contact type	Parameters	DOF
Planar	n normal	R_n, T_u and T_v, where u and v are orthogonal to n
Cylindrical	u axis O origin	R_{u+O} and T_u
Conical	u axis O origin	R_{u+O}
Spherical	O origin	R_{u+O}, R_{v+O} and R_{n+O}
Toroidal	u axis O origin	R_{u+O}

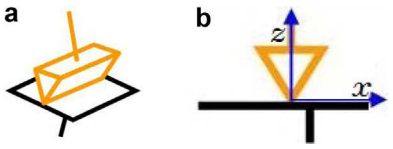


Fig. 5. Two view of a linear contact.

kinematic tensors of all the contacts. The procedure to compute the motion derived from all the contacts between two nodes n_i and n_j associated with the parts P_i and P_j in the assembly model is illustrated in Algorithm 6. This procedure returns the set of joint arcs A_J and the set of its attributes T_{A_j}, where $t_{i,j}$ represents the attribute for the arc $a_{i,j} = (n_i, n_j)$.

Algorithm 6. The component relationship detection algorithm

```
1:  procedure JointsComputation(N_P)
2:    for each n_i ∈ N_P
3:      for each n_j ∈ N_P
4:        contactList = GetContactsBetween(n_i, n_j)
5:        t_{i,j} = RelativeMotionComputation(contactList)
6:        A_J ← a_{i,j} = (n_i, n_j)
7:        T_{A_j} ← t_{i,j}
8:      end for
9:    end for
        Return A_J and T_{A_j}
10: end procedure
```

For each pair of nodes (n_i, n_j) the arcs corresponding to their contacts are read by the procedure *GetContactsBetween*, then the motion that they allow and the type of joint (i.e. the values of the DOF and *JointType* attributes) are computed by the *RelativeMotionComputation* procedure. This procedure is based on the type and number of contacts, in particular, the following cases are distinguished:

(i) **There is only one contact arc between the nodes n_i and n_j** (Fig. 7a).
In this case, the DOF and the *JointType* attributes of the joint arc are the same of the contact arc.

(ii) **The contact arcs between the nodes n_i and n_j are associated only with planar surfaces** (Fig. 7b).
In this case, each planar face prevents a translation along its normal, thus between two planar faces non-parallel or coplanar with normal n_i and n_j respectively, only one translation is allowed, that is along the resulting vector $n_i \times n_j$. Since the joint arc arises by only planar contacts, the *JointType* attribute will be *Planar*.

(iii) **The contact arcs between the nodes n_i and n_j are associated with different types of surfaces** (Fig. 7c).
In this case, only the DOFs allowed by **all** the contacts are permitted in the final motion. In this case, if the joint arc arises by contacts of the same type, its *JointType* attribute is inherited, otherwise it is set up to *Mixed*.

4.3. Pattern of repeated parts detection

The procedure for the detection of regular patterns of repeated components is based on the work of Chiang et al. [49] and its generalization to assembly models [50].

This module defines the attribute *PatternList* for the root node of an EAM. First of all, the set of repeated instances of the same object is identified. Thus, for their detection, the first check is performed on the component names. However, since sometimes, identical parts are instantiated as different objects then, to overcome this issue, parts having identical values of volume and surface area are considered as repetition of the same object.

Then, for each list of repeated parts, the centroid of each part is computed. The set of centroids represents the input for the procedure defined by Chiang et al. [49], which identifies all the possible patterns formed by equidistant centroids satisfying a

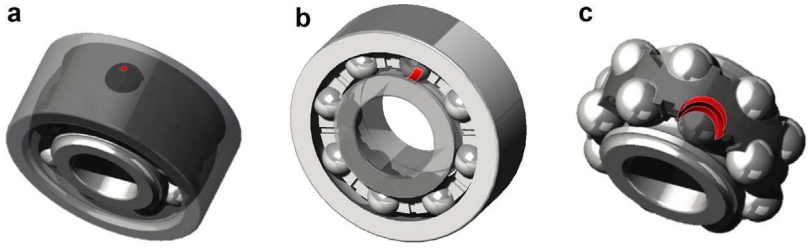


Fig. 6. Examples of volumetric intersections of spherical parts.

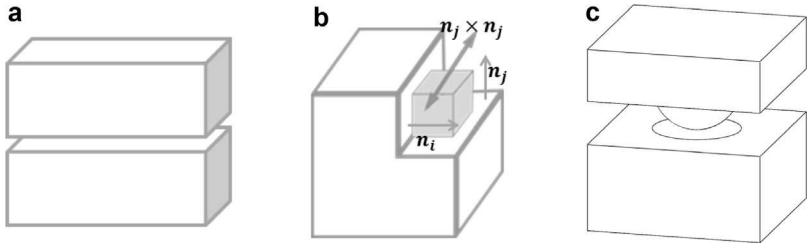


Fig. 7. Example of combinations of contacts.

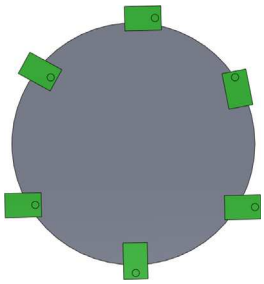


Fig. 8. Repeated parts whose centers of gravity are aligned.

certain transformation (i.e. linear translation, circular translation, circular rotation or reflection). These potential patterns (with their type and parameters) are confirmed if the entities of the corresponding repeated parts satisfy the same transformation, i.e. if they are arranged as specified by the detected possible pattern. This check verifies the correct orientation of the corresponding repeated parts and avoids to recognize configurations where the centroids are arranged in a pattern, but not their corresponding parts, as depicted in Fig. 8.

4.4. Assembly statistics computation

Once most of the data necessary for the EAM creation are extracted, assembly statistics can be computed and associated as attributes to the root node in n_{Root}. The data present in the assembly statistics regard the number of pattern for each type, i.e. linear translation, circular translation, circular rotation and reflection. Then, the procedure to compute the assembly statistics takes as input the list of patterns *PatternList* and counts the number of patterns for each type.

4.5. Part and component classification

In order to allow more semantic-oriented retrieval, parts are classified and some functional sets are identified. In case the retrieval is integrated in a PDM system, and if models are well annotated, this step can be avoided and the value of the *ComponentType* node attribute can be set directly inquiring the PDM data. The proposed part and component classification is a multi-step process where parts are primarily classified simply based on their shape through the *ShapeBasedPartClassification* procedure detailed in Section 4.1. Then, by exploiting this initial classification and part relations, the ones characterizing functional sets are identified by the *ComponentClassification* procedure (see Section 4.5.1). Finally, the part classification is confirmed or revised by using engineering considerations on the surrounding elements in the assembly through the *ShapeAndContextBasedPartClassification* procedure (see Section 4.5.2).

4.5.1. Component classification

This module identifies assembly components that represent some functional sets characterized by the presence of specific types of components generally positioned according to some rules. The output of this module represents the attribute function $\Phi^{CT}_{N_A}$. This step exploits the *ShapeBasedPartClassification* (Section 4.1) and the relationships between parts stored in the EAM (Section 4.2). Since the EAM is represented by a graph structure, the identification algorithm is performed by graph matching, where subgraphs representing assembly components are compared with graphs representing predefined templates characterizing the mechanical functional sets to be identified.

So far, the classification focuses on bearing components. Fig. 9 shows the two graphs used for the subgraph matching for the rolling ball bearings. In Fig. 9a the yellow node identify all the balls that form the circular pattern. In this case, the arcs identify a vertex or linear contact. While in Fig. 9b, the yellow node is associated with a single part of the CAD model and the arcs represent a surface contact. The nodes of the bearing template graphs represent the main elements characterizing a bearing, which include the repeated elements (balls or rollers) arranged in a circular pattern or idealized with a toroidal/cylindrical shape, the inner and outer rings. Arcs represent the interactions among these elements. For instance, balls are in contact by a point or a curve with the inner and outer rings, while rollers are always in contact by a line and eventually two planar faces. To focus on the most promising candidates, the matching consider parts classified as *part of bearing* for inner and outer rings, *sphere-like* for balls, *cylinder-like* for rollers, *torus-like* and *spacer* in case of simplification of rolling elements.

4.5.2. Shape and context-based part classification

This classification assesses the shape-based part classification by analyzing the context of use of the part in the assembly model. Then, this module confirms or changes the value of the attributes $\Phi^{CT}_{N_P}$ previously set by the *ShapeBasedPartClassification* procedure. The analysis of the context of use of a part in an assembly relies on some a priori engineering knowledge related to generally present interactions between components in specific mechanisms. The rationale behind is that a specific mechanical component may perform its function within a functional set if it is positioned according to specific conditions with respect to certain classes of components. Hence, to determine if a component effectively belongs to the assigned class it is necessary to check with which types of components/parts the part interacts. At this stage, the proposed framework focuses on speed reducer mechanisms, which are included in many products. For further details, refer to [51].

5. Multi-level similarity criteria

Considering nodes, the similarity criteria that can be set up concern shape, size, component type and pattern type information.

Let $c_N = (\alpha_{Sh}, \alpha_{Si}, \alpha_{CT}, \alpha_{PT})$ be a vector representing the criteria that can be selected, where α_{Sh}, α_{Si}, α_{CT} and α_{PT} represent respectively the activation of the shape, size, component type and pattern criterion. The generic criterion $\alpha_* \in \{0, 1\}$ and $\alpha_* = 1$

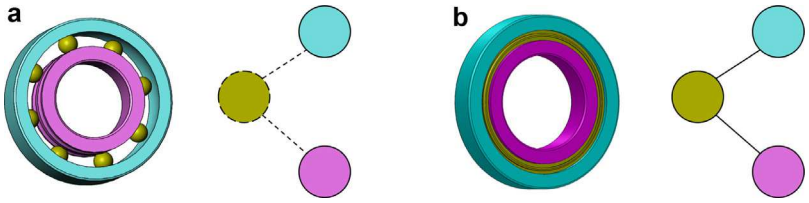


Fig. 9. Abstract bearing definition (a) Bearing with ball pattern; (b) bearing with idealized pattern. (For interpretation of the references to color in the text, the reader is referred to the web version of this article.)

means that the criterion * is activated. With this notation, the generic association node (n_q^i, n_k^j) is generated if n_q^i and n_k^j are compatible for all the $\alpha_* = 1$. The definitions of the possible node compatibility are described in Appendix C.

Concerning arcs, let $c_A = (\alpha_C, \alpha_{C_{num}}, \alpha_{J_{num}})$ be a vector representing the arc criteria that can be selected, where α_C, $\alpha_{C_{num}}$ and $\alpha_{J_{num}}$ represent respectively the contact, allowed DOF for contacts and allowed DOF for joints criteria. Also for arcs, the generic filter $\alpha_* \in \{0, 1\}$ and $\alpha_* = 1$ means that the filter * is activated. With this notation, the generic pair of association nodes (n_q^i, n_k^j) and (n_q^h, n_k^l) are linked if the arcs a_q^{ij} and a_k^{hl} are compatible for the selected $\alpha_* = 1$. The definitions of the possible arc compatibility are described in Appendix C. Differently from the node criteria, when a criterion of compatibility on the arcs is selected the other two are set to 0. This is due to the fact that the contact similarity entails the allowed DOF for contacts similarity, which requires the similarity of the allowed DOF for joints.

The right part of Fig. 10 shows an assembly model representing a flange with three screws and a portion of its attributed multi-graph. For sake of readability, the root node corresponding to the entire assembly model is omitted and only joint arcs A_J are depicted. The left part of Fig. 10 shows a similar model together with a portion of its attributed multi-graph. Here, the assembly model only contains two screws. The nodes of the two attributed multi-graphs represent the parts of the CAD models, same type of line indicates same value of the spherical harmonic shape descriptor (i.e. parts with similar shapes) and parts with the same color belong to patterns of a specific type (i.e. green for circular translation and red for linear translation). The arcs represent the joint contacts where the labels indicate the DOF allowed between two linked parts.

In this example, the hypothesis is that the user seeks assemblies in which parts with similar shape are connected by the same joint relationships. This means that two nodes create an association node if their corresponding parts have similar shape, while association arcs are added if the joint arcs (between the related pairs of nodes in the attributed multi-graphs) have the same number of rotations and same number of translations. The association graph, resulting from these criteria, is illustrated in Fig. 11. Here, there are six possible maximum cliques:

$$
\begin{aligned}
(C_{1,2,c_N,c_A})_1 &= \{AA', BB', CC'\},\\
(C_{1,2,c_N,c_A})_2 &= \{AA', BC', DB'\},\\
(C_{1,2,c_N,c_A})_3 &= \{AA', CB', DC'\},\\
(C_{1,2,c_N,c_A})_4 &= \{AA', CC', DB'\},\\
(C_{1,2,c_N,c_A})_5 &= \{AA', BB', DC'\},\\
(C_{1,2,c_N,c_A})_6 &= \{AA', BC', CB'\}.
\end{aligned}
$$

Each clique represents a possible sub-graph matching between the two attributed multi-graphs G_1 and G_2 (Fig. 10). The detected cliques are used to evaluate the similarity, as explained in the following section.

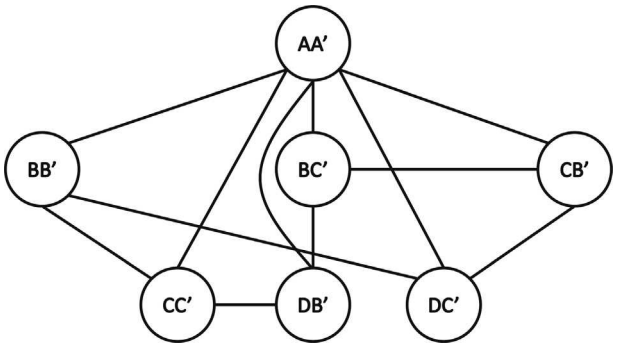


Fig. 11. Association graph for the objects in Fig. 10 built with shape and joint criteria of similarity.

6. Similarity assessment

Kim et al. [52] point out that similarity (or dissimilarity) among complex models such as assemblies has to be measured taking into consideration different factors that are able to emphasize a specific aspect of the model. Indeed, in this way, the final measurement can be tuned according to the user's needs.

Depending on the user-specified similarity criteria c_N and c_A, the similarity between two CAD assembly models can be assessed accordingly. To evaluate the similarity, a bundle of 4 measures $S = [\mu^{shape}, \mu^{joint}, \mu^{position}, \mu^{structure}]$ is proposed, where each of them characterizes a single aspect of the similarity between two assemblies. As previously said, the similar elements between the query and the compared assemblies correspond to the common sub-graphs between the two corresponding attributed multi-graphs. Thus, the similarity is computed on the detected cliques. To simplify the writing, in the following, the generic rth clique $(C_{q,k,c_N,c_A})_r$ will be denoted as C, the set of its nodes as N_C, and the number of nodes in N_C as $|N_C|$.

6.1. Shape similarity measure: μ^{shape}

The $\mu^{shape}(C)$ shape similarity measure is based on the shape descriptor of each node involved in the clique C, i.e. the 3D spherical harmonics and the size values described by the attribute function $\Phi^{Sh}_{N_P}(n)$ (see Appendix B). Since two objects can have exactly the same shape but different dimensions, the assessment of the shape similarity is based uniquely on the 3D spherical harmonics, while the size values are used to refine the retrieval results. The L_2-norm is an appropriate norm to compare 3D spherical harmonics [41]. Thus, the shape similarity of a clique C is defined as the average of the shape similarity of each node in the clique:

$$\mu^{shape}(C) = \frac{1}{|N_C|} \sum_{(n_q^i, n_k^j) \in C} 1 - \frac{\Phi^{Sh}_{N_P}(n_q^i)}{\|\Phi^{Sh}_{N_P}(n_q^i)\|_2} - \frac{\Phi^{Sh}_{N_P}(n_k^j)}{\|\Phi^{Sh}_{N_P}(n_k^j)\|_{2_2}} \tag{3}$$

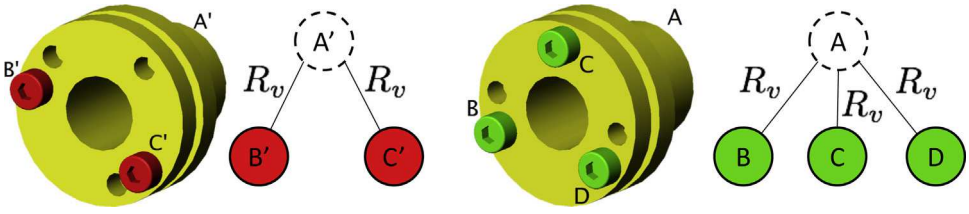


Fig. 10. Example of two assembly models with a different number of screws, and their EAM attributed multi-graphs G_1 and G_2. (For interpretation of the references to color in the text, the reader is referred to the web version of this article).

6.2. Joint similarity measure: μ^{joint}

The $\mu^{joint}(C)$ joint similarity measure is defined to assess how much two assemblies are similar in terms of the relative DOF among their parts. A joint can arise from contacts of different types (Surface, Curve, Point or UnSolved). In case of joints derived from contacts of type Surface, the allowed DOF of the two linked parts is available, otherwise only the information on the type of joint is accessible. Considering the differences in the types of attributes, this measure is defined as a combination of two other measures $\mu^{joint}_{surf}(C)$ and $\mu^{joint}_{curve,pt}(C)$ whose computation is not straightforward as explained in the next paragraphs.

Actually, for any couple of assembly models identifying the same objects, their joint similarity measure should be the same independently of their position and orientation in their coordinate reference frames. Therefore, a simple comparison between the DOF of the corresponding elements is not appropriate. This is illustrated in Fig. 12. An alternative approach is to consider the deviation among the axes characterizing the DOF between the parts. It must be noted that just considering the deviations between the axes characterizing the DOF between two parts at a time is not appropriate. For instance, the deviation obtained considering only parts P_1 and P_4 of Fig. 12a, and the axes characterizing the DOF of the same parts in Fig. 12b, corresponds to the angle defined by the axis u and n that is equal to 90 degrees even if the objects are the same.

On the contrary, the variations of each pair of axes defined by the DOF between a given part with all those in contact, e.g. (P_1, P_4), (P_2 P_4) and (P_3 P_4), are the same in both configurations. Thus, the variation of the rotation and translation axes defined by the DOF of a part with respect to all its parts in contact is considered. Therefore, μ^{joint}_{surf} is computed on the nodes of the clique using the attributes of all its incident arcs (see Definition (B.4) in Appendix B). To manage configurations in which several angles defined by the axes of the DOF exist, matrices are used to capture all the variations of the rotation/translation angles related to a part. The elements of the matrices are computed using the dot product of a pair of axes, as specified in the following definitions:

Definition 1. $Var_{Tra}(n)$ is the matrix characterizing the variations of translations related to the node n. Its generic element is defined as $(\text{Var}_{Tra}(n))_{i,j} = t_i.t_j$, with t_i, $t_j \in Tra(n)$, and $Tra(n)$ the set of all the joint translations between the node n and its generic adjacent node n^*:

$$Tra(n) = \{\bigcup_a \Phi^{Tra}_{A_j}(a), \quad \forall a : a = (n, n^*) \in A_J\}$$

Definition 2. $Var_{Rot}(n)$ is the matrix characterizing the variations of rotations related to the node n. Its generic element is defined as $(\text{Var}_{Rot}(n))_{i,j} = r_i.r_j$, with r_i, $r_j \in Rot(n)$, and $Rot(n)$ the set of all the joint rotations between the node n and its generic adjacent node n^*:

$$Rot(n) = \{\bigcup_a \Phi^{Rot}_{A_j}(a), \quad \forall a : a = (n, n^*) \in A_J\}$$

Since the axes are normalized, the dot product corresponds to the cosine of the angle between the considered axes. The final variations of a node in the clique are obtained by computing the averages of those matrices. Here, the average $\sigma(M)$ of a matrix M of size $N \times N$ is meant as the arithmetic mean of the elements in the matrix divided by the number of elements:

$$\sigma(M) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} m_{i,j} \quad \forall m_{i,j} \in M \tag{4}$$

Following this method, when a single arc is incident to a node, its variation is equal to 1. Moreover, the joint measure has to take into account also contacts and joints of type UnSolved, for which the DOF information is not available. In such a configuration, the only possible difference between two models is the number of joints concurring to the definition of the variation matrix, i.e. joints of surface type. Thus, this information is used to distinguish these cases by dividing the average of each variation matrix by the number of translations/rotations involved in their definition, i.e. $|Tra(n)|$ and $|Rot(n)|$.

Finally, as the DOF are not computed for joints arisen from curve or point contacts, the defined joint measure $\mu^{joint}_{curve,pt}$ is based on the type of contacts and has maximum value 1 if the joints are of the same type (both Curve or Point) and a lower value if the joints are of different types. The lower value is set to 0.8 and it was chosen heuristically to slightly decrease the measure.

As a result of this analysis, the overall joint similarity measure is defined as follows:

$$\mu^{joint}(C) = [\mu^{joint}_{surf}(C) + \mu^{joint}_{curve,pt}(C)]/2 \tag{5}$$

where the measures related to Surface and Curve/Point are computed as follows:

$$\mu^{joint}_{surf}(C) = \frac{1}{|N_C|}\sum_{(n^i_q, n^j_k)\in C}\left[1 - \frac{d_{Tra}((n^i_q, n^j_k)) + d_{Rot}((n^i_q, n^j_k))}{2}\right] \tag{6}$$

$$\mu^{joint}_{curve,pt}(C) = \frac{1}{|N_C|}\sum_{((n^i_q, n^j_k),(n^l_q, n^h_k))\in C^2}[1 - d_{edge}((a^{ij}_q, a^{lh}_k))] \tag{7}$$

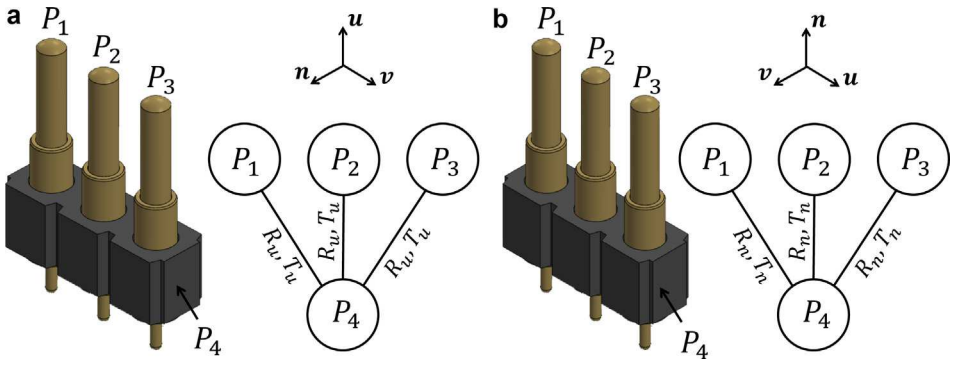


Fig. 12. Two instances of the same object embedded in two different reference frames: (a) query model, (b) target model.

with:

- $d_{Tra}((n_q^i, n_k^j)) = \text{abs}\left(\frac{\sigma(\text{Var}_{Tra}(n_q^i))}{|Tra(n_q^i)|} - \frac{\sigma(\text{Var}_{Tra}(n_k^j))}{|Tra(n_k^j)|}\right)$
- $d_{Rot}((n_q^i, n_k^j)) = \text{abs}\left(\frac{\sigma(\text{Var}_{Rot}(n_q^i))}{|Rot(n_q^i)|} - \frac{\sigma(\text{Var}_{Rot}(n_k^j))}{|Rot(n_k^j)|}\right)$
- $d_{edge}((a_q^{ij}, a_k^{lh})) = \begin{cases} 0 & \text{if } \Phi_{A_j}^{JT}(a_q^{ij}) = \Phi_{A_j}^{JT}(a_k^{lh}), \\ 0.2 & \text{otherwise.} \end{cases}$

So far, in Eq. (5), the combination of the two individual measures on Surface and Curve/Point has been weighted equally, but a different weighting could also be imagined to give more or less importance to the specific type of joints.

6.3. Position similarity measure: μ^{position}

Another salient characteristic affecting the level of similarity between two assemblies is the relative arrangement of the assembly components. For instance, the assembly models in Fig. 13 have parts with similar shapes and relationships (i.e. the colored parts are not in contact among them) but with a different arrangement. The objective of the position similarity measure is to distinguish such configurations while characterizing the position of the similar parts.

To this aim, this measure considers the directional versors between the center of gravity of each couple of parts in the clique not in contact each other. The use of versors makes the measure size independent. Since the versors are dependent on the reference frame, to overcome this problem, the relative position similarity is computed following the same approach adopted for the computation of the joint similarity in case of surface type joint:

$$\mu^{position}(C) = \frac{1}{|N_C|} \sum_{(n_q^i, n_k^j) \in C} [1 - d_{Dir}((n_q^i, n_k^j))] \tag{8}$$

with:

- $d_{Dir}((n_q^i, n_k^j)) = \text{abs}(\sigma(\text{Var}_{Dir}(n_q^i)) - \sigma(\text{Var}_{Dir}(n_k^j)))$
- $\text{Var}_{Dir}(n_q^i)$ is the variation matrix of the directional versors between the gravity centers of the parts corresponding to the node n_q^i and the nodes n_q^l such that $(n_q^i, n_q^l) \notin A_{C_q}$,
- $\text{Var}_{Dir}(n_k^j)$ is the variation matrix of the directional versors between the gravity centers of the parts corresponding to the node n_k^j and the nodes n_k^h such that $(n_k^j, n_k^h) \notin A_{C_k}$.

6.4. Structure similarity measure: $\mu^{\text{structure}}$

Similar products and CAD models can be organized in sub-assemblies in different ways according to the designer objectives; therefore, it is important to capture the differences at the level of the structure. The structure similarity measure characterizes the way parts are assembled in the assembly tree of the CAD model. The EAM descriptor encodes the hierarchical structure of an assembly model by a set of arcs A_S.

Thus, the proposed measure is based on the comparison of the structural relations of the pairs of nodes (n_q^i, n_q^l) and (n_k^j, n_k^h), where (n_q^i, n_k^j) and (n_q^l, n_k^h) are nodes of the clique. Its evaluation requires verifying if the nodes n_q^i and n_q^l belong (or not) to the same sub-assembly in G_q, and similarly if the nodes n_k^j and n_k^h belong (or not) to the same sub-assembly in G_k. Using a distance function equal to 0 if the pair of nodes (n_q^i, n_q^l) has the same relation as the pair (n_k^j, n_k^h), or 1 otherwise, the structure similarity measure of a clique C is defined as follow:

$$\mu^{structure}(C) = \frac{1}{|N_C|^2} \sum_{((n_q^i, n_k^j), (n_q^l, n_k^h)) \in C^2} [1 - d_{Str}((n_q^i, n_k^j), (n_q^l, n_k^h))] \tag{9}$$

with:

$$d_{Str}((n_q^i, n_k^j), (n_q^l, n_k^h)) = \begin{cases} 0 & \text{if } [\exists n_q^* \in N_q \text{ and } \exists n_k^* \in N_k] \text{ s.t.} \\ & \left[((n_q^i, n_q^*), (n_q^l, n_q^*)) \in A_{S_q}\right. \\ & \left.\text{and } ((n_k^j, n_k^*), (n_k^h, n_k^*)) \in A_{S_k}\right] \\ 1 & \text{otherwise.} \end{cases}$$

6.5. Combination of similarity measures

The definition of measures able to rank the retrieved models requires combining the above measures. Here, three measures are defined to characterize the local, partial and global similarities between two CAD assembly models. Through the combinations, it should also be possible to weight differently each similarity measure using a factor either specified in the query or chosen during the browsing of the results. Currently, the weights are specified by the user. However, in future, the idea is to study the possibility to assign default weight values according to particular usage scenarios.

Definition 3. The *local similarity* measure between two assembly models is the weighted average of the four individual similarity measures and it is defined by the function:

$$\gamma_\ell : \begin{vmatrix} D_{q,k} \times W & \to & [0,1] \\ (C, w) & \mapsto & \dfrac{\left[\begin{array}{l} w^{sh}\mu^{shape}(C) + w^{jo}\mu^{joint}(C) \\ +w^{po}\mu^{position}(C) + w^{st}\mu^{structure}(C) \end{array}\right]}{w^{sh} + w^{jo} + w^{po} + w^{st}} \end{vmatrix}$$

with $w = \{w^{sh}, w^{jo}, w^{po}, w^{st}\} \in W = [0,1]^4$.

To provide information on how many parts are similar among all of the query and the target models, two so-called coverage

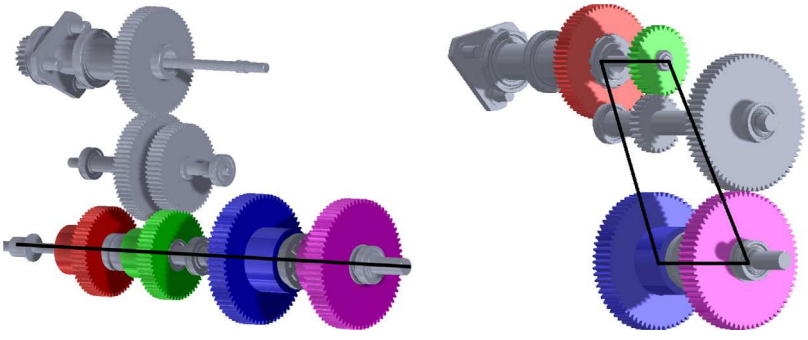


Fig. 13. Example of assembly models with similar parts arranged according to different configurations of a gear box.

factors have been identified. They are used to define the partial and global similarity measures by opportunely weighting the local similarity one. The term *coverage factor* refers to the percentage of elements of the query and target models that are considered similar with respect to all the elements in the two models.

Definition 4. The partial and global coverage factors (*PCF* and *GCF*) of a clique C are defined as:

$$PCF(C) = \frac{|N_C|}{|N_q|} \quad \text{and} \quad GCF(C) = \frac{2|N_C|}{|N_q| + |N_k|}$$

where N_C, N_q and N_k represent respectively the set of nodes in the clique C, the nodes in the query model and the nodes in the target model.

The global and partial similarity measures are defined weighting the local similarity with these coverage factors. Right now, those measures do not consider the "relevance" of the parts yet. For instance, the contribution of a small part (e.g. rivet, c-clip) could be weighed differently from a larger part (e.g. gear, shaft, bearing). In some cases, it could also be considered as negligible.

Definition 5. The assembly *partial similarity* measure between two models is evaluated by the function:

$$\gamma_p : \begin{vmatrix} D_{q,k} \times W & \rightarrow & [0,1] \\ (C,w) & \mapsto & \gamma_p(C,w) = PCF(C) \times \gamma_\ell(C,w) \end{vmatrix}$$

Definition 6. The assembly *global similarity* measure between two models is evaluated by the function:

$$\gamma_g : \begin{vmatrix} D_{q,k} \times W & \rightarrow & [0,1] \\ (C,w) & \mapsto & \gamma_g(C,w) = GCF(C) \times \gamma_\ell(C,w) \end{vmatrix}$$

Note that global similarity implies partial and local similarities. However, while partial similarity implies local similarity, the vice versa does not hold. In the end, the following rules stand:

- Two models are 100% *globally similar* if and only if they are 100% *locally similar* and *GCF* = 1.
- Two models are 100% *partially similar* if and only if they are 100% *locally similar* and *PCF* = 1.

7. Experimentation and results

To test the effectiveness of the proposed framework and to evaluate the ability of the proposed measures to characterize the similarities of assembly models, a proper dataset of CAD assembly models is required. Unfortunately, the available mechanical shape benchmarks in literature, as the Princeton Shape Benchmark (PSB) [53], the National Design Repository (NDR) [54] and the Engineering Shape Benchmark (ESB) [46], are not proper for our purpose. Indeed, they include just parts and do not consider assembly models. The Princeton ModelNet repository [55] (http://modelnet.cs.princeton.edu/) provides a selection of CAD models in Object File Format (OFF), i.e. the models are represented by a collections of regular polygons (e.g. triangles) that describe the model surface. This type of representation allows a fast visualization of the model but does not include the whole geometric and topological information provided by the boundary representation (B-Rep) that, conversely, is the reference representation created during the detailed product design phase. Therefore, existing benchmarks are useful to test retrieval methods that address similarity evaluation solely based on the shape but not for systems directly working on B-Rep models and including specific assembly information. Thus, so far, no public CAD assembly database exists to evaluate and compare assembly retrieval systems. Moreover, the authors of assembly retrieval methods have developed their own datasets, which are not public [56,57]. Therefore, a proper comparison with existing methods is directly not possible.

Consequently, the proposed assembly retrieval framework has been benchmarked using our own dataset that contains 137 assembly models arranged in 11 classes (Table 2). There are 12,783 parts in total, out of which 4871 parts are unique. To ease future comparisons, the complete dataset is available at http://3dassemblyrepository.ge.imati.cnr.it/#. Downloading the models, also the number of parts and number of unique parts of each model are available.

To display and browse the results in an intuitive and user-friendly manner, multi-view dynamic web pages have been developed. For any retrieved results, they provide a 3D model view in which the matched components are highlighted and a three-bar histogram that indicates the values of the computed

Table 2
Classification of the CAD assemblies forming our testing set.

Category	Model	Part	Unique Part
Coupling flange	5	70	23
Differential	5	1520	278
Double rotor turbine	17	1080	554
Hydraulic reduction	9	1473	439
Hydraulic rotor	8	1240	508
Landing gear	6	81	57
Linear actuator	6	77	30
Mill max	8	103	30
Propeller mixer	20	2599	1253
Rotor wind turbine	24	2969	946
Other	29	1571	753
Total	**137**	**12,783**	**4871**

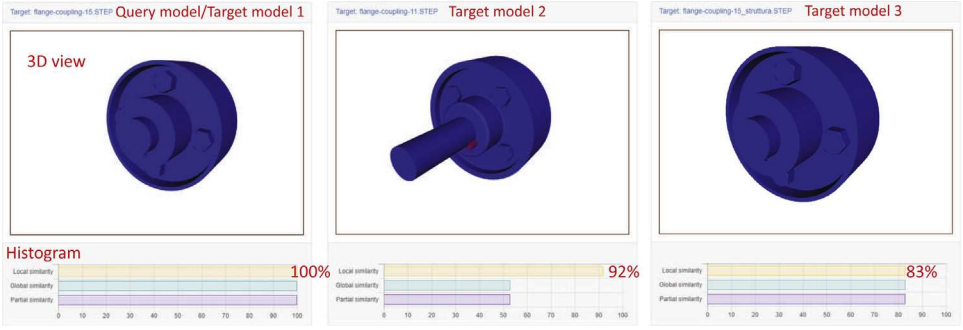


Fig. 14. Example of two target models (middle, right) retrieved from a given query model (left) and characterized by their similarity levels (local, global and partial). (For interpretation of the references to color in the text, the reader is referred to the web version of this article).

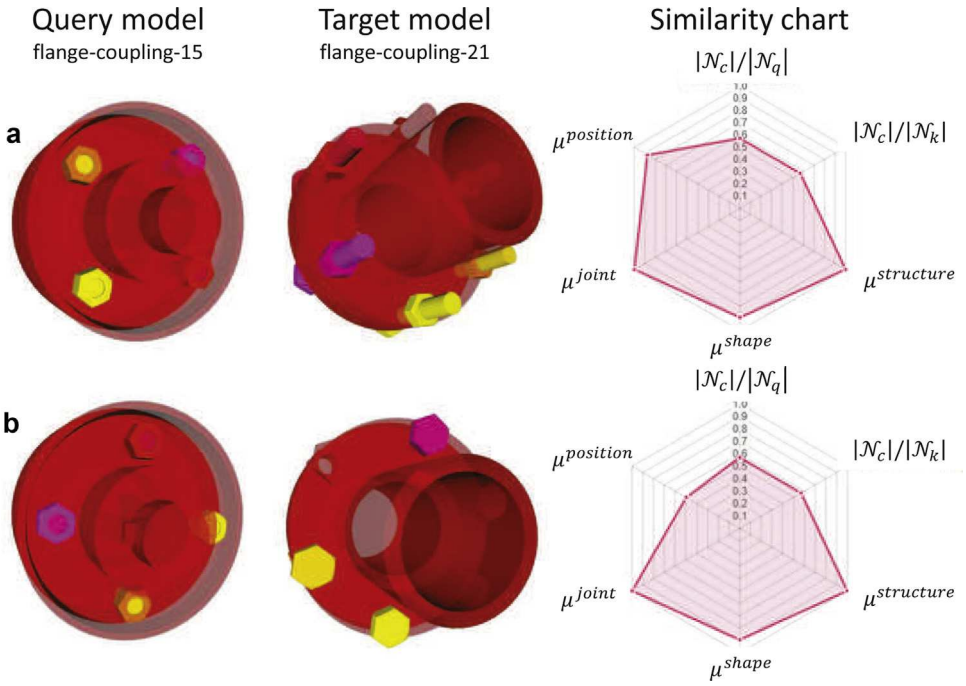


Fig. 15. Two different cliques (a, b) obtained comparing the target model (#4) to the query model (#1) in Table 3. The different arrangement of the parts is reflected by different values of $\mu^{position}$ in the radar charts. (For interpretation of the references to color in the text, the reader is referred to the web version of this article).

similarity levels (local, global and partial). This is illustrated in Fig. 14. In the 3D view, the matched components are colored in blue while the unmatched ones are in red. The bars of the histograms indicate the values of the local (yellow bar), global (green) and partial (purple) overall similarity measures.

Each model can be further analyzed in another view, where the query and the target models are displayed. This is illustrated in Fig. 15. In the 3D view, each pair of matched parts is highlighted by a different color, and the values of the single measures are reported in a radar chart. This example is further discussed in the next section.

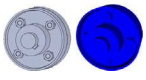
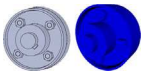
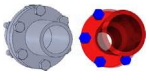
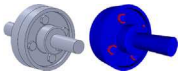
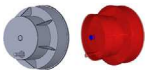
7.1. Results: coupling flange query model case

The developed retrieval system has been tested and validated with several query models and criteria combinations. At http://3dassemblyrepository.ge.imati.cnr.it/# the ground truth for eight of them is available. Here, due to space limitation, only two results are described.

The first experiment aims at retrieving mechanical coupling flanges similar to a given one. A mechanical coupling flange is a set of components linking two parts of a product. In this example, the query model corresponds to the first model (#1) illustrated in Table 3. It contains four screws and four nuts arranged in a circular translation pattern, two main flanges, two shafts and two keys. All the parts are organized in a flat structure, i.e. without any sub-assembly. The CAD model does not present any volumetric intersection and each screw is in contact with the corresponding nut through an idealized cylindrical face (i.e. the screw thread is not modeled).

The user-specified similarity criteria $c_N = (1, 0, 1, 1)$ include the shape similarity among components, same component type and same pattern type. In this case, the threshold ε_{shape} used for the shape criterion is set up to 0.20, thus two components should have shapes similar at 80% according to the values of their 3D spherical harmonics. The user-specified similarity criteria $c_A = (0, 0, 1)$ require that two pairs of compatible nodes should have the same number of allowed rotations and translations. Based on these

Table 3
Subset of the models retrieved using model #1 as query with indication of the related similarities' measures with a weighting vector $w_{flange} = \{1, 1, 1, 1\}$, and similarity criteria $c_N = (1, 0, 1, 1)$ and $c_A = (0, 0, 1)$.

CAD models &matched parts	#1	#2	#3	#4	#5	#6
						
μ^{shape}	1.00	1.00	0.77	0.89	0.74	0.77
μ^{joint}	1.00	1.00	0.68	1.00	1.00	0.00
$\mu^{position}$	1.00	1.00	0.99	0.87	0.95	0.82
$\mu^{structure}$	1.00	0.35	1.00	1.00	1.00	1.00
γ_ℓ	1.00	0.83	0.86	0.94	0.92	0.65
γ_p	1.00	0.83	0.74	0.54	0.52	0.19
γ_g	1.00	0.83	0.74	0.54	0.52	0.05

criteria, the association graph can be built putting in relation the query model with each one in the dataset on which cliques are detected and the similarity measures μ^{shape}, μ^{joint}, $\mu^{position}$ and $\mu^{structure}$ can be computed. Then, the overall similarity measures γ_ℓ, γ_p and γ_g can be evaluated. Here, the weights are all equal and $w_{flange} = \{1, 1, 1, 1\}$. This means that the single similarity measures have the same importance in evaluating the local, partial and global similarity measures. The numerical results are shown in Table 3.

The result target model #1 corresponds to the query model, as expected all the similarity measures have maximum values as it fully matches itself.

The second result (i.e. model #2) has the same components as the query model, i.e. same number of parts, same shape and same contacts but organized in a different way. Indeed, the structure of the query model is flat, while in this model, the set of screws and the set of nuts are gathered together forming two sub-assemblies. Thus, its $\mu^{structure}$ is less than 1 and this factor decreases the final value of the local similarity measure. Since all the components of the query and of the target model are matched, the values of the partial and global similarities correspond to the local one, i.e. $PCF = 1$ and $GCF = 1$ according to Definitions 5 and 6.

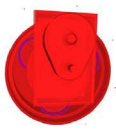
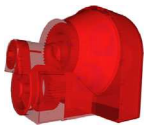
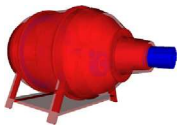
In the third model (#3) screws and nuts have some volumetric intersections and are matched thanks to the use of the UnSolved attribute. Differently from the arcs in the query model, in the target model #3 the arcs of type UnSolved do not have a proper number of allowed rotations and translations. Thus, according to Eq. (6), this difference affects their similarity at the level of the joint. In this example, the number of matched components is twelve while the number of components in the query and in the target models is fourteen, then the overall partial and global similarity measures are lower than the local one according to the same factor, i.e. $GCF = \frac{2 \cdot 12}{14+14} = 0.86$ and $PCF = \frac{12}{14} = 0.86$.

The fourth (#4) and fifth (#5) models have very similar measures. This is due to the fact that, the coverage of the models is computed from the number of matched elements and the two models have the same number of matched elements: four screws and four nuts for the fourth model and four screws, two main flanges, a shaft and a key in the fifth one. A similarity evaluation based also on the volume may be nearer to the visual perception of similarity, but in general, it is more meaningful to consider the relevance of the matched parts, i.e. fastener elements should be less important than a shaft. Of course, such a consideration requires a study on the relevance of each component category depending on the query objectives.

For the target model (#4), Fig. 15 reports two different cliques, which correspond to two different sets of similar parts. Both the query and the target models have a circular pattern of screws and nuts, but with a different number of repeated elements (i.e. four in the query model and six in the target model). Thus, in the target model, it is not possible to find four equidistant screws and nuts that cover an entire circumference. This affects the position similarity measure as depicted on the radar charts which give a global overview of the similarity.

The fifth model (#5) is very similar to the query one and the shape of its parts differs only for the shafts and the main flanges which have a border thicker than the query one. However, in this model the screws and the nuts present clearances, which means that these components are not in contact. The fact that not all the components are matched is reflected by the partial and global measures that are lower than the local one. This difference indicates that the matched parts are very similar, but do not cover all the query model and neither the target model. This is confirmed analyzing the values of the single similarity measures. As expected, the values of joint, position and structure are very low, while the value of the shape similarity highlights small differences in the matched components. The most significant variation is in the number of matched components. In this example, both the query and the target models have fourteen parts and eight of them are matched, then $PCF = GCF = 0.57$ and these values affect negatively partial and global measures. In addition, what hinders a full match is the type of contact. In particular, the fours nuts in the target model are not in contact with the screws, and the key and the shaft present different contacts. Indeed, the key and the shaft in the query model are in contact by three planar faces, thus a translation is allowed, while in the target model the two parts are in contact by four planar faces and no motion is possible.

Table 4
Subset of the models retrieved using the gearbox model #1 as query with indication of the related similarities' measures with a weighting vector $w_{gear} = \{1, 1, 1, 0\}$, and similarity criteria $c_N = (0,\ 0,\ 1,\ 1)$ and $c_A = (0,\ 0,\ 1)$.

CAD models	#1	#2	#3	#4	#5	#6
						
Matched parts						
μ^{shape}	1.00	1.00	0.78	0.66	0.76	0.74
μ^{joint}	1.00	1.00	0.00	0.00	0.00	0.00
$\mu^{position}$	1.00	1.00	1.00	1.00	0.80	0.61
γ_ℓ	1.00	1.00	0.59	0.55	0.52	0.45
γ_p	1.00	1.00	0.44	0.42	0.39	0.34
γ_g	1.00	0.18	0.06	0.04	0.06	0.10

Finally, from the values of the different levels of similarity measures, the user can easily understand that the sixth model (#6) is not suited for design reuse. Indeed, the partial similarity measure is very low indicating that many elements of the query model do not match. Actually, this model has been incorporated in the reported results to demonstrate how the measures can help discarding models not suited to the user purposes.

7.2. Results: planetary gearbox query model case

The second experiment here reported aims at retrieving objects similar to planetary gearboxes. This functional set has several sun-planet and ring-planet gear pairs as the first model (#1) in Table 4, which has been used as query model. In this example, the similarity criteria for the nodes $c_N = (0, 0, 1, 1)$ are relaxed considering the component type and the pattern type. The similarity criteria for the arcs $c_A = (0, 0, 1)$ require the same number of allowed rotations and translations. Here, the weights used to compute γ_ℓ, γ_p and γ_g are set up to $w_{gear} = \{1, 1, 1, 0\}$. Thus, the weight $w^{st} = 0$ which means that the $\mu^{structure}$ similarity measure is not taken into account in the computation of the various similarity values (global, partial and local). Note that discarding a measure in the assessment of the overall similarity does not imply that the correspondent criteria are not considered during the matching; indeed, the similarity criteria are used to build the association graph, while the evaluation of a similarity measure is used to rank the results of the matching process.

In general, for all the retrieved models in Table 4, one can observe that the global similarity measure is much lower than the others. This suggests that the query model (#1) is included in the target models. Of course, for the first model being the query, all its measures have maximum values since it perfectly matches itself. The second model (#2) has high values of local and partial similarities, this indicates that the single similarity values are high and that the entire query is included in the target model. Indeed, in this case all the measures have almost maximum values and only the global measure is low. This is due to the fact that the query model is entirely included in a bigger target model.

For the third retrieved model (#3), the matched components are the three gears and three axes. The measure μ^{joint} is null since no contacts are present between the matched components. Thus, there is no variation of incident rotation/translation to compare. This affects negatively the final value of the local similarity measure when the weight w^{joint} is not null. Anyhow, even if it would be technically possible, assigning $\mu^{joint} = 1$ when no contact is present could be misleading for the user in his/her model analysis.

For the fourth model (#4), the matched components are also the three gears and three axes, but differently from model #3, gears are modeled in a simplified form and they are recognized thanks to the attribute *CompType* of the EAM, which identifies three simple rings as gears by exploiting the surrounding context of the components. Note that including shape compatibility, and depending on the chosen shape similarity threshold, this configuration would be probably not retrieved since the shapes of the planar gears are quite different from the ones proposed in the query model.

For the last two models (#5) and (#6), the same considerations as for the third model hold, i.e. three gears and three shafts are retrieved. The measure μ^{joint} is also null since there is no contact between the retrieved parts.

7.3. Effectiveness evaluation

Several techniques can be used to evaluate the efficacy of a method. In this work, a qualitative and quantitative study is reported by the evaluation of the precision and the recall achieved. As mentioned before, the absence of public CAD assembly databases, including a proper classification of models as relevant or not according to specific queries and/or user needs (i.e. ground truth), makes arduous a proper evaluation and precludes comparisons with respect to the existing approaches. To overcome this limitation, not only the entire dataset used in this evaluation is provided (Table 2), but also the adopted ground-truth. Hence, for 8 queries, the query model is included together with the adopted similarity criteria and the expected retrieved models. Those models have been used to compute the precision and recall in case of global and partial similarity respectively. Thus, all the assemblies have been labeled as *relevant* or *irrelevant* according to 8 different queries. Four of the documented queries aim to retrieve models globally similar to the query model; the others four search for models that include the query model and thus that are partially similar. An example of the employed queries is represented by the models in Tables 3 and 4. In the example of Table 3 the relevant models are those from #1 to #5, while model #6 is considered not relevant. Concerning the partial similarity, models are tagged as relevant if they contain similar parts arranged in similar configurations independently by their model organization in sub-assemblies.

Actually, in the evaluation of the precision and recall, the position in the ranked list of the retrieved models is much more important than the value of the similarity measure. For each query, our system produces a vector with the ranked models and a binary vector that indicates if the corresponding model was set to relevant

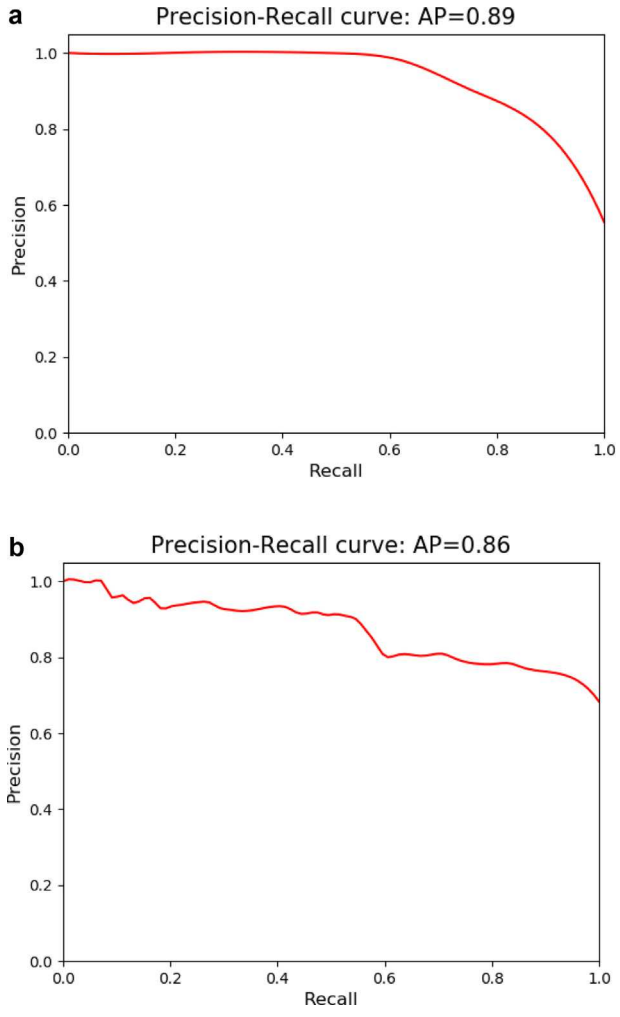


Fig. 16. Precision and recall plots of assembly retrieval results.

or not for that particular query. Then, the precision-recall curve is computed by the python procedure *precision _ r ecall _ c urve* [58] using the generated vectors. The obtained curves are then interpolated and their average is computed and illustrated in Fig. 16.

The average of the precision (*AP*) of the proposed method is very good ($AP > 0.86$) in both cases (global and partial similarity), where *AP* is almost 1 when evaluating the global similarity and close to one ($AP > 0.90$) for the evaluation of the partial similarity. In general, it can be observed that *AP* starts to decrease for high level of recall ($AP > 0.80$).

Analyzing the relevant models retrieved with low measures, which are responsible for the decrease of the precision, one can notice that they were designed in an unusual way. For instance, in the evaluation of global similarity, a query requires to retrieve the landing gears similar to a given one according to their shape, type of pattern and contacts. In the proposed dataset, there are landing gears where the wheels, tires and shaft are represented as five individual parts, while in other models this set of parts is represented as a single part. These two types of representation of the same object affect negatively the capability of the method to retrieve relevant models. This condition may arise when models are downloaded from public repositories or bought from third parts for which the modeling strategies and the adopted representations cannot be fully controlled.

8. Conclusions and perspectives

This paper introduces a system to assess the similarity of CAD assembly models according to multiple criteria and levels of similarity. This objective is achieved using both explicit information encoded in CAD models in STEP format and implicit information extracted automatically. The collected data are represented in an attributed multi-graph structure and the similarity is recognized by detecting the maximum common sub-graph between two graphs representing the compared assembly models.

To allow such a content-based similarity assessment, four measures have been introduced. In particular, the shape, joint, position and structure similarity measures have been defined. They are combined to evaluate the three possible levels of similarity between assemblies: global partial and local. Their combination allows to weight the various aspects depending on the user interests and purposes. The experiments and results confirm the usefulness of the information extracted from CAD models and stored in the EAM files for their similarity assessment. An ad-hoc visualization interface has been devised and used to enhance the user experience when analyzing the results.

The proposed method is also able to retrieve locally similar assemblies whose matched components are disconnected. This is possible since, in the definition of the association graph, two association nodes are connected if they have the same relationship, where “same relationship” includes also the “not in contact” case. Moreover, differently from most of the assembly retrieval systems present in literature, the query model can be made by more parts than the target ones.

Nevertheless, some improvements can be foreseen in the involved processes. During the creation of the association graph, improvements are possible to optimize the matching process. The complexity in the detection of the maximum clique is strictly related to the number of nodes in the association graph, and then reducing their number can enhance the efficiency of the entire system. To this purpose, future work will focus on the possibility to gather in a unique association node the attributed nodes representing different instances of the same component. This study involves several challenges. First, the criteria to group together different parts must be investigated. For instance, it is possible to gather parts simply belonging to the same pattern or also being in contact with the same components. Similarly, depending on the objectives, some parts (e.g. screw, nuts) could be neglected reducing the number of components to be considered. In all these cases, how the relationships in the association graph are updated and how the procedure handles the compatibility to build the association graph have to be investigated. The impact of this strategy on the similarity measures has to be evaluated to provide the appropriate adaptations.

In the future, other measures could be defined. For instance, a size measure could assess the similarity of two models according to their dimensions. In the current implementation, the size is only involved in the creation of the association graph. So far, the weights used to combine the set of measures to compute the similarity are set by the user and are applied to all the parts equally. Weights related to the type or size of parts could also be investigated to stress the similarity on the most important characteristics. To discover which weights best fulfill the user requirements according to different use scenarios, it is necessary to investigate how weight combinations affect the final score and, above all, it is essential to include the user feedback to validate which results are considered pertinent for the specific query. Using the reciprocal comparison of all the models in the dataset can facilitate the browsing by visualizing similar models according to all the defined measures.

This retrieval framework has be designed and developed to be a stand-alone system; in the future, it could be integrated in PDP or PLM systems. This will allow using it either to refine the results of a retrieval function included in the PDM system and/or to exploit other useful data available in the PDM system to improve the similarity assessment. For instance, the information on the functionality of the different component may easy the classification process and the functional set identification.

Conflicts of interest

None declared.

Appendix A. Data associated to an EAM

The *structure information* of the EAM is encoded by the attributes *CompType* that defines the type of the component, *PatternType* that characterizes the arrangement in the 3D space of a set of repeated parts (indicated by *RP*), and *PatternList* that indicates all the patterns in the model. These attributes can assume the following values.

$$\begin{aligned} CompType \;=\; \{&\text{bearing},\ \text{c}-\text{clip},\ \text{gear},\ \text{shaft},\ \text{spacer},\ \text{key},\ \text{nut},\\ &\text{linkage arm},\ \text{parts of bearing},\ \text{screw and bolt},\\ &\text{cylinder}-\text{like},\ \text{cube}-\text{like},\ \text{sphere}-\text{like},\\ &\text{torus}-\text{like},\ \text{miscellaneous}\} \end{aligned} \tag{A.1}$$

$$\begin{aligned} PatternType \;=\; \{&\text{linear translation},\ \text{circular translation},\\ &\text{circular rotation},\ \text{reflection}\} \end{aligned} \tag{A.2}$$

$$PatternList = \{(\text{patType}_i, \text{RP}_i),\ \forall i \in \{1, \ldots, N_p\}\} \tag{A.3}$$

where N_p is the number of patterns in the model, patType_i corresponds to the type of the ith pattern and RP_i to its constituting repeated parts.

The *statistic information* of the EAM is encoded by the attributes *PartStat* and *AssemblyStat*. They can assume the following values.

$$PartStat = ([0,1] \times \mathbb{N})^5$$

$$AssemblyStat = \mathbb{N}^4$$

The *interface information* of the EAM is encoded by the attributes *ContactType* that indicates the type of entities shared by two parts in contact, *JointType* that indicates the joint resulting from a set of contacts between two parts, and *DOF* that indicates the degrees of freedom, it can be associated both to a contact and a joint. These attributes can assume the following values.

$$ContactType = \{\text{Surface}, \text{Curve}, \text{Point}, \text{UnSolved}\}$$

$$JointType = \{\text{Surface}, \text{Curve}, \text{Point}, \text{UnSolved}, \text{Mixed}\}$$

$$DOF = Tra \times Rot$$

The *shape information* of the EAM is encoded by the attributes *Shape* that indicates the histogram of 544 bins representing the 3D spherical harmonic associated to a part, and *Size* that indicated the volume and the surface area of a part. These attributes can assume the following values.

$$Shape = \mathbb{R}^{544}$$

$$Size = \mathbb{R}^+ \times \mathbb{R}^+$$

Appendix B. Definition of the attributed functions

Nodes and arcs of an EAM representing a CAD model are associated with the attributes defined in Appendix A by a parts attribute function Φ_{N_P}, a sub-assemblies attribute function Φ_{N_A}, a contact arc attribute function Φ_{A_C} and a joint arc attribute function Φ_{A_J}. These attribute functions are defined as follow.

$$\Phi_{N_P} : \begin{vmatrix} N_P & \rightarrow & T_{N_P} = Shape \times Size \times CompType \\ & & \times PatternType \times PartStat \\ n & \mapsto & \left(\Phi^{Sh}_{N_P}(n), \Phi^{Si}_{N_P}(n), \Phi^{CT}_{N_P}(n), \Phi^{PT}_{N_P}(n), \Phi^{PS}_{N_P}(n)\right) \end{vmatrix} \quad (B.1)$$

$$\Phi_{N_A} : \begin{vmatrix} N_A & \rightarrow & T_{N_A} = PatternList \times CompType \\ & & \times AssemblyStat \\ n & \mapsto & \left(\Phi^{PL}_{N_A}(n), \Phi^{CT}_{N_A}(n), \Phi^{AS}_{N_A}(n)\right) \end{vmatrix} \quad (B.2)$$

$$\Phi_{A_C} : \begin{vmatrix} A_C & \rightarrow & T_{A_C} = DOF \times ContactType \\ a & \mapsto & \left(\left(\Phi^{Tra}_{A_C}(a), \Phi^{Rot}_{A_C}(a)\right), \Phi^{CT}_{A_C}(a)\right) \end{vmatrix} \quad (B.3)$$

$$\Phi_{A_J} : \begin{vmatrix} A_J & \rightarrow & T_{A_J} = DOF \times JointType \\ a & \mapsto & \left(\left(\Phi^{Tra}_{A_J}(a), \Phi^{Rot}_{A_J}(a)\right), \Phi^{JT}_{A_J}(a)\right) \end{vmatrix} \quad (B.4)$$

Appendix C. Definition of the compatibilities

Let us indicate the generic arc in G_q (resp. G_k) between the node pair (n^i_q, n^j_q) (resp. (n^l_k, n^h_k)) as a^{ij}_q (resp. a^{lh}_k). Here, a^{ij}_q and a^{lh}_k are respectively part of the sets A_q and A_k. Since two nodes can be linked by multiple contact arcs, the set of contact arcs between the node pair (n^i_q, n^j_q) (resp. (n^l_k, n^h_k)) is indicated as $A^{ij}_{C_q}$ (resp. $A^{lh}_{C_k}$). The number of elements in a set * is indicated by $|*|$.

Definition 7. Two nodes n^i_q and n^j_k are considered similar according to the *shape* criterion if and only if:

$$\Phi^{Sh}_{N_P}(n^i_q) - \Phi^{Sh}_{N_P}(n^j_k)_2 < \varepsilon_{shape}$$

where ε_{shape} represents the threshold set in the query.

Definition 8. Two nodes n^i_q and n^j_k are considered similar according to the *size* criterion if and only if:

$$\text{abs}\left(\Phi^{Si}_{N_P}(n^i_q) - \Phi^{Si}_{N_P}(n^j_k)\right) < \varepsilon_{size}$$

where ε_{size} represents the threshold set in the query.

Definition 9. Two nodes n^i_q and n^j_k are considered similar according to the *component type* criterion if and only if:

$$\Phi^{CT}_{N_P}(n^i_q) = \Phi^{CT}_{N_P}(n^j_k)$$

Definition 10. Two nodes n^i_q and n^j_k are considered similar according to the *pattern type* criterion if and only if:

$$\Phi^{PT}_{N_P}(n^i_q) = \Phi^{PT}_{N_P}(n^j_k)$$

Definition 11. Two arcs a^{ij}_q and a^{lh}_k are considered compatible according to the *contact* type criterion if and only if:

$$(a^{ij}_q \in A_{C_q}) \quad \text{and} \quad (a^{lh}_k \in A_{C_k})$$

Definition 12. The set of arcs $A^{ij}_{C_q}$ and $A^{lh}_{C_k}$ are considered compatible according to the *allowed DOF for contacts* type criterion if and only if $\forall a_s \in A^{ij}_{C_q}$, $\exists\, a_t \in A^{lh}_{C_k}$ such that:

$$\left[\left|\Phi^{Tra}_{A_C}(a_s)\right| = \left|\Phi^{Tra}_{A_C}(a_t)\right| \quad \text{and} \quad \left|\Phi^{Rot}_{A_C}(a_s)\right| = \left|\Phi^{Rot}_{A_C}(a_t)\right|\right]$$
$$\text{or} \quad [\Phi^{CT}_{A_C}(a_s) = \text{UnSolved}] \quad \text{or} \quad [\Phi^{CT}_{A_C}(a_t) = \text{UnSolved}]$$

Definition 13. Two arcs a^{ij}_q and a^{lh}_k are considered compatible according to the *allowed DOF for joints* type criterion if and only if:

$$\left[\left|\Phi^{Tra}_{A_J}(a^{ij}_q)\right| = \left|\Phi^{Tra}_{A_J}(a^{lh}_k)\right| \quad \text{and} \quad \left|\Phi^{Rot}_{A_J}(a^{ij}_q)\right| = \left|\Phi^{Rot}_{A_J}(a^{lh}_k)\right|\right]$$
$$\text{or} \quad [\Phi^{JT}_{A_J}(a^{ij}_q) = \text{UnSolved}] \quad \text{or} \quad [\Phi^{JT}_{A_J}(a^{lh}_k) = \text{UnSolved}]$$

References

Oesterreich, T.D., Teuteberg, F., [1]. Understanding the implications of digitisation and automation in the context of industry 4.0: a triangulation approach and elements of a research agenda for the construction industry. Comput. Ind. 83, 121–139.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K., [2]. Shape-based searching for product lifecycle applications. Comput.-Aided Des. 37 (13), 1435–1446.

Bai, J., Gao, S., Tang, W., Liu, Y., Guo, S., [3]. Design reuse oriented partial retrieval of CAD models. Comput.-Aided Des. 42 (12), 1069–1084.

Brière-Côté, A., Rivest, L., Maranzana, R., [4]. Comparing 3D CAD models: uses, methods, tools and perspectives. Comput.-Aided Des. Appl. 9 (6), 771–794.

Deshmukh, A.S., Banerjee, A.G., Gupta, S.K., Sriram, R.D., [5]. Content-based assembly search: a step towards assembly reuse. Comput.-Aided Des. 40 (2), 244–261.

Zhen-Ba, L., Shu-Hui, B., Kun, Z., Shu-Ming, G., Jun-Wei, H., Jun, W., [6]. A survey on partial retrieval of 3D shapes. J. Comput. Sci. Technol. 28 (5), 836–851.

Ferreira, A., Marini, S., Attene, M., Fonseca, M.J., Spagnuolo, M., Jorge, J.A., et al., [7]. Thesaurus-based 3d object retrieval with part-in-whole matching. Int. J. Comput. Vis. 89 (2–3), 327–347.

Lupinetti, K., Giannini, F., Monti, M., Pernot, J.P., [8]. Multi-criteria retrieval of CAD assembly models. J. Comput. Des. Eng. 5 (1), 41–53.

Tangelder, J.W., Veltkamp, R.C., [9]. A survey of content based 3D shape retrieval methods. Multimed. Tools Appl. 39 (3), 441–471.

Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K., [10]. Three-dimensional shape searching: state-of-the-art review and future trends. Comput.-Aided Des. 37 (5), 509–530.

Cardone, A., Gupta, S.K., Karnik, M., [11]. A survey of shape similarity assessment algorithms for product design and manufacturing applications. J. Comput. Inf. Sci. Eng. 3 (2), 109–118.

Hilaga, M., Shinagawa, Y., Kohmura, T., Kunii, T.L., [12]. Topology matching for fully automatic similarity estimation of 3d shapes. Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques SIGGRAPH'01. ACM, New York, NY, USA, pp. 203–212. doi:http://dx.doi.org/10.1145/383259.383282 ISBN 1-58113-374-X.

Osada, R., Funkhouser, T., Chazelle, B., Dobkin, D., [13]. Shape distributions. ACM Trans. Graph. (TOG) 21 (4), 807–832.

Corney, J., Rea, H., Clark, D., Pritchard, J., Breaks, M., MacLeod, R., [14]. Coarse filters for shape matching. IEEE Comput. Graph. Appl. 22 (3), 65–74.

Novotni, M., Klein, R., [15]. Shape retrieval using 3d zernike descriptors. Comput.-Aided Des. 36 (11), 1047–1062.

Hong, T., Lee, K., Kim, S., [16]. Similarity comparison of mechanical parts to reuse existing designs. Comput.-Aided Des. 38 (9), 973–984.

Biasotti, S., Cerri, A., Bronstein, A.M., Bronstein, M.M., [17]. Quantifying 3D shape similarity using maps: recent trends, applications and perspectives. Eurographics (State of the Art Reports), , pp. 135–159.

Cardone, A., Gupta, S.K., Deshmukh, A., Karnik, M., [18]. Machining feature-based similarity assessment algorithms for prismatic machined parts. Comput.-Aided Des. 38 (9), 954–972.

Li, M., Zhang, Y.F., Fuh, J.Y.H., Qiu, Z.M., [19]. Design reusability assessment for effective CAD model retrieval and reuse. Int. J. Comput. Appl. Technol. 40 (1/2), 3–12. doi:http://dx.doi.org/10.1504/IJCAT.2011.038546.

El-Mehalawi, M., Miller, R.A., [20]. A database system of mechanical components based on geometric and topological similarity. Part I: Representation. Comput.-Aided Des. 35 (1), 83–94.

El-Mehalawi, M., Miller, R.A., [21]. A database system of mechanical components based on geometric and topological similarity. part ii: indexing, retrieval, matching, and similarity assessment. Comput.-Aided Des. 35 (1), 95–105.

Chu, C.H., Hsu, Y.C., [22]. Similarity assessment of 3D mechanical components for design reuse. Robot. Comput.-Integr. Manuf. 22 (4), 332–341.

Çiçek, A., [23]. Similarity and scaling assessments of mechanical parts using adjacency relation matrices. J. Mater. Process. Technol. 206 (1–3), 106–119.

Biasotti, S., Marini, S., Spagnuolo, M., Falcidieno, B., [24]. Sub-part correspondence by structural descriptors of 3D shapes. Comput.-Aided Des. 38 (9), 1002–1019.

Tao, S., Huang, Z., Ma, L., Guo, S., Wang, S., Xie, Y., [25]. Partial retrieval of cad models based on local surface region decomposition. Comput.-Aided Des. 45 (11), 1239–1252.

Zehtaban, L., Elazhary, O., Roller, D., [26]. A framework for similarity recognition of CAD models. J. Comput. Des. Eng. 3 (3), 274–285.

Giannini, F., Lupinetti, K., Monti, M., [27]. Identification of similar and complementary subparts in B-rep mechanical models. J. Comput. Inf. Sci. Eng. 17 (4) 041004-041004-11.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., [28]. Pointnet: deep learning on point sets for 3d classification and segmentation. Proc Computer Vision and Pattern Recognition (CVPR), vol. 1(2), IEEE, pp. 4.

Renu, R., Mocko, G., [29]. Retrieval of solid models based on assembly similarity. Comput.-Aided Des. Appl. 13 (5), 628–636.

Wang, P., Li, Y., Zhang, J., Yu, J., [30]. An assembly retrieval approach based on shape distributions and Earth Mover's distance. Int. J. Adv. Manuf. Technol. 86 (9–12), 2635–2651.

Zhang, J., Pang, J., Yu, J., Wang, P., [31]. An efficient assembly retrieval method based on hausdorff distance. Robot. Comput.-Integr. Manuf. 51, 103–111.

Miura, T., Kanai, S., [32]. 3D shape retrieval considering assembly structure. Proceeding of Asian Symposium for Precision Engineering and Nanotechnology 2009 (ASPEN 2009) 11–13.

Tao, S., Huang, Z., [33]. Assembly model retrieval based on optimal matching. Software Engineering and Knowledge Engineering: Theory and Practice. Springer, pp. 327–336.

Chen, X., Gao, S., Guo, S., Bai, J., [34]. A flexible assembly retrieval approach for model reuse. Comput.-Aided Des. 44 (6), 554–574.

Han, Z., Mo, R., Yang, H., Hao, L., [35]. CAD assembly model retrieval based on multi-source semantics information and weighted bipartite graph. Comput. Ind. 96, 54–65.

Zheng, Y., Cohen-Or, D., Averkiou, M., Mitra, N.J., [36]. Recurring part arrangements in shape collections. Computer Graphics Forum, vol. 33. Wiley Online Library, pp. 115–124.

Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., [37]. Multi-view convolutional neural networks for 3d shape recognition. Proceedings of the IEEE international conference on computer vision 945–953.

Yi, L., Guibas, L., Hertzmann, A., Kim, V.G., Su, H., Yumer, E., [38]. Learning hierarchical shape segmentation and labeling from online repositories. ACM Trans. Graph. (TOG) 36 (4), 70.

Requicha, A.A., [39]. Representations for rigid solids: theory, methods, and systems. ACM Comput. Surv. 12 (4), 554–562.

Lupinetti, K., Giannini, F., Monti, M., Pernot, J.P., [40]. Automatic extraction of assembly component relationships for assembly model retrieval. Proc. CIRP 50, 472–477.

Kazhdan, M., Funkhouser, T., Rusinkiewicz, S., [41]. Rotation invariant spherical harmonic representation of 3D shape descriptors. Symposium on Geometry Processing, vol. 6 156–164.

Conte, D., Foggia, P., Sansone, C., Vento, M., [42]. Thirty years of graph matching in pattern recognition. Int. J. Pattern Recognit. Artif. Intell. 18 (03), 265–298.

Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M., [43]. The maximum clique problem. Handbook of Combinatorial Optimization. Springer, pp. 1–74.

Rotation Invariant Shape Descriptors. . http://htmlpreview.github.io/?https://github.com/mkazhdan/ShapeSPH/blob/master/descriptors.html.

Rucco, M., Giannini, F., Lupinetti, K., Monti, M., [45]. A methodology for part classification with supervised machine learning. AI EDAM 33 (1), 100–113.

Jayanti, S., Kalyanaraman, Y., Iyer, N., Ramani, K., [46]. Developing an engineering shape benchmark for CAD models. Comput.-Aided Des. 38 (9), 939–953.

Liu, Y.S., Fang, Y., Ramani, K., [47]. Idss: deformation invariant signatures for molecular shape comparison. BMC Bioinform. 10 (1), 157.

Pommier, S., Berthaud, Y., [48]. Mécanique générale. Dunod.

Chiang, L., Giannini, F., Monti, M., [49]. Identification of Regularities in CAD Part and Assembly Models. Springer International Publishing, Cham, pp. 355–365. doi:http://dx.doi.org/10.1007/978-3-319-33111-9_33 ISBN 978-3-319-33111-9.

Lupinetti, K., Chiang, L., Giannini, F., Monti, M., Pernot, J.P., [50]. Regular patterns of repeated elements in CAD assembly model retrieval. Comput.-Aided Des. Appl. 14 (4), 516–525.

Lupinetti, K., Giannini, F., Monti, M., Pernot, J.P., [51]. Identification of functional sets in mechanical assembly models. ICIDM2017 conference, Milan, Italy, 2017 July 17–19.

Kim, H., Cha, M., Mun, D., [52]. Shape distribution-based approach to comparing 3d cad assembly models. J. Mech. Sci. Technol. 31 (12), 5627–5638.

Shilane, P., Min, P., Kazhdan, M., Funkhouser, T., [53]. The princeton shape benchmark. Shape Modeling Applications, 2004. Proceedings, IEEE, pp. 167–178.

Regli, W.C., Foster, C., Hayes, E., Ip, C.Y., McWherter, D., Peabody, M., et al., [54]. National design repository project: a status report. International Joint Conferences on Artificial Intelligence (IJCAI), Seattle, WA, August, pp. 4–10.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., et al., [55]. 3d shapenets: a deep representation for volumetric shapes. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1912–1920.

Hu, K.M., Wang, B., Yong, J.H., Paul, J.C., [56]. Relaxed lightweight assembly retrieval using vector space model. Comput.-Aided Des. 45 (3), 739–750.

Qin, F., Gao, S., Yang, X., Li, M., Bai, J., [57]. An ontology-based semantic retrieval approach for heterogeneous 3D CAD models. Adv. Eng. Inform. 30 (4), 751–768.

Scikit-Learn Library. . http://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_curve.html.