HOSTED BY

ELSEVIER

Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Design and Engineering 3 (2016) 274–285

www.elsevier.com/locate/jcde

A framework for similarity recognition of CAD models

Leila Zehtaban*, Omar Elazhary, Dieter Roller

University of Stuttgart, Universitätsstr. 38, 70569 Stuttgart, Germany

Received 18 November 2015; received in revised form 12 April 2016; accepted 14 April 2016
Available online 19 April 2016

Abstract

A designer is mainly supported by two essential factors in design decisions. These two factors are intelligence and experience aiding the designer by predicting the interconnection between the required design parameters. Through classification of product data and similarity recognition between new and existing designs, it is partially possible to replace the required experience for an inexperienced designer. Given this context, the current paper addresses a framework for recognition and flexible retrieval of similar models in product design. The idea is to establish an infrastructure for transferring design as well as the required PLM (Product Lifecycle Management) know-how to the design phase of product development in order to reduce the design time. Furthermore, such a method can be applied as a brainstorming method for a new and creative product development as well. The proposed framework has been tested and benchmarked while showing promising results.

Keywords: Design classification; Opitz coding system; Similarity retrieval

1. Introduction

Market competition and constantly changing customer demands yield to massive production. Consequently a huge amount of information is produced and archived every day. Reusing such information can reduce the product cost and time; along with optimizing the product design. There is no doubt that the dimension of reusing such know-how greatly affects the designing of new products in the conceptual design phase. Different information could be derived and learnt from the already existing design including geometry, material, process planning, manufacturing, price and generally Product Lifecycle Management (PLM) information. It is possible to become skilled at PLM knowledge by knowing the similar cases, therefore making better design decisions. In this regard, an efficient similarity recognition algorithm is a fundamental prerequisite which assists in providing an automated and intelligent decision making.

Alongside all research developed by engineers on decision making and Decision Support Systems (DSS) [1–3] and models [4], psychologists suggest four essential techniques for improving the problem of decision making and choice overload [5–7] listed in following:

(1) *Cut:* get rid of the extraneous alternatives
(2) *Concretize:* Make it real
(3) *Categorize:* more categorization, fewer choice
(4) *Condition:* for complexity, it is easier to make complex decisions by gradually increasing the complexity

The above mentioned techniques have been considered in the current research aiming for modeling a knowledge-based framework. Such a structure guides the designer to an optimized decision making with respect to reuse of the existing similar artifacts data, Fig. 1.

A CAM-based classification system was applied and developed further intended for automatic extraction of the design information from STEP file. In addition, an infrastructure was designed and developed for a comprehensive retrieval, i.e.

*Corresponding author. Tel.: +49 711 685 88 327; fax: +49 711 685 88 320.
E-mail addresses: zehtaban@informatik.uni-stuttgart.de (L. Zehtaban), omazhary@gmail.com (O. Elazhary), roller@informatik.uni-stuttgart.de (D. Roller).
Peer review under responsibility of Society of CAD/CAM Engineers.

http://dx.doi.org/10.1016/j.jcde.2016.04.002

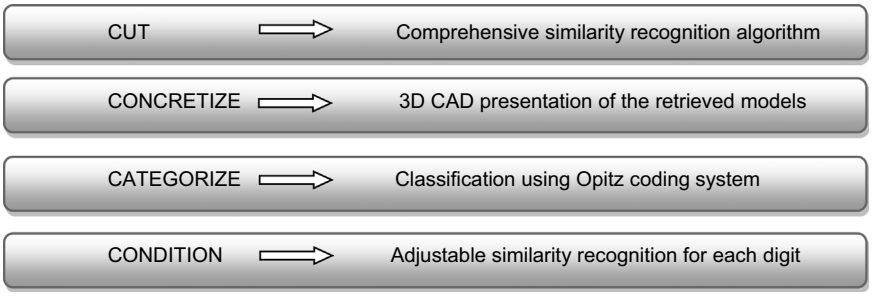

Fig. 1. Comparable techniques of better decision making and the developed system.

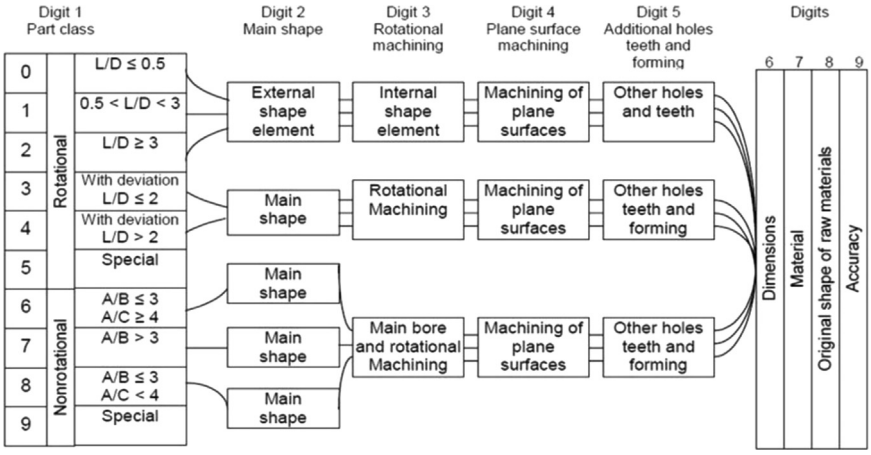

Fig. 2. The Opitz code's main structure [8,9].

selective similarity recognition and part retrieval. The following paragraphs explain these phases briefly.

- *Comprehensive similarity recognition algorithm:* A comprehensive similarity recognition algorithm has been developed to calculate distance function. A distance function is applied to measure the similarity and accordingly to discard the dissimilar cases and retrieve only the similar models. The percentage of total similarity can be tuned by the designer.
- *3D CAD presentation of the retrieved models:* To make the retrieved similar models more tangible, all the related 3D CAD models are also represented to the user in addition to STEP files and the classification codes (Opitz codes).
- *Opitz coding system:* For categorization and classification, Opitz coding system as a successful method of group technology (GT) in manufacturing has been applied.
- *Adjustable similarity recognition for each digit:* The similarity setting for each digit can vary from 0 to 100 percent. 0 indicates dissimilar features and 100 for identical models and the required weight for each digit step-by-step can be adjusted between these two numbers by the user.

Opitz coding system is a method of group technology which is applied in Computer Aided Manufacturing (CAM) for part classification. It is a well-known method for classification of manufacturing features and is named after Herwart Opitz who originally proposed this coding system [8]. Opitz coding system consists of alphanumerical digits, each presenting a feature and its type. In the other words, a digit is the aggregation of all the feature conditions it is composed of according to the code's definition. Created by Herwert Opitz in 1970, Opitz code is a hybrid code consisting of a maximum of 14 digits. The code itself is divided into 3 sections. The first section consists of five digits which are dubbed the “form code” and describe the geometry and topology of the product/part. The second section, also called the “supplementary code” was added later and consists again of four digits that represent the dimensions, material, original shape of raw stock, and accuracy of the product/part. Each digit in these two sections may contain 10 different values ranging from 0 to 9. The third section consists of only four characters; A, B, C, and D. It is called the “secondary code” and allows for organization customization of the code. Here, organizations can include proprietary and organization-specific information regarding the product/part. The Opitz code's structure is given below in Fig. 2.

The first five digits of the code are referred to manufacturing features and highlight the manufacturing features such as bore, step, forming, etc. and their specifications. As an instance if the discussed manufacturing feature is bore, blind bore, through bore, number of the bore/s, position/s and main/axial bore are specifically pointed and highlighted in the code.

The number of digits or the size of Opitz code is fixed for all parts and it is independent of complexity of a part. The current research incorporates feature recognition and the translation of such features into a model's group comparison functions code,

explained in Section 3. The group technology code will then be used to retrieve identical or similar models from a benchmark discussed in Section 4. After constructing such an application, it will be evaluated by some of the most common evaluation techniques. For example, the feature recognition functionality will be evaluated with regard to the accuracy of the produced code and the similarity-retrieval functionality will be evaluated using one of the most famous query evaluation techniques; the precision–recall technique discussed in Section 5.

The objective of this paper is to describe a framework for feature recognition and similarity retrieval of a CAD model. The modules of the framework will be discussed in detail. In addition, the test and benchmark results of shape representation and similarity retrieval of the proposed method will be discussed and compared with other well-known similar methods.

2. Related works

Shape signature is a replacement for 3D geometry as a transferable and understandable model of 3D geometry for the shapes digital comparison. A shape signature is supposed to contain all shape features and characteristics [10,11]. There are different approaches to convert a shape into its signatures such as statistical methods, graph-based methods, group technology (GT), etc., as Fig. 3 presents. Among these methods which convert a shape into different types of presentation such as graph or statistics, group technology is an alphanumerical code which contains more information than the geometry. One of the advantages of applying such a signature is the potential of code extension for a comprehensive product signature. Such a comprehensive product signature can be later used as standard format for exchanging product data with design intent and in downstream applications. Until recently, the exchange of product data has been limited to transferring geometry which has been led to standard format for CAD data including ISO 10303 (STEP). The exchange of CAD models between different CAD systems has led the research to develop standard CAD data format such as STEP. However, such standardizations are mostly limited to geometry and there is a lack of standardization for product data. Typical GT coding systems comprises of Opitz coding system (13 digits), MICLASS (12 digits), KK3 system (21 digits) and DCLASS (8 digits) [12,13]. Between different methods of part classification systems in GT, Opitz coding system [8] is one of the most well-known and widely applied approaches. Group technology codes and classification methods are all based on manufacturing classification of part families.

In the approach done by Henderson and Musti [14], Prolog rules are used to determine the DCLASS code of a part/product. An application, called CODER, was constructed to perform this operation. It consists of three modules; a solid model converter, a geometry interpreter, and a part coder. The solid model converter prepares the given CAD file for processing (i.e. preprocessing) and then passes it along to the geometry interpreter. A significant part of preprocessing consists of transforming the model into predicates that are interpretable by Prolog. The geometry interpreter then takes over and uses this predicate calculus representation of the model to find and identify all mid- and low-level features that exist within the model. These may consist of axis sets, protrusions, depressions, and edge types. All the feature information is then stored into a "description list". Finally, the part coder uses the description lists in order to generate the DCLASS code. With regard to the employment of the DCLASS code vs. the Opitz code, it is found that the Opitz code provides much more detail. By simply comparing the length of the codes, where the DCLASS has 8 digits as opposed to a 9-digit Opitz code with 4 additional customizable digits, it is calculated that the DCLASS can have about 676000000 possible enumerations, as opposed to the 4.56976×10^{14} possible enumerations given by the Opitz code. In this research, the more details are available, the better the similarity-retrieval functionality will work. In addition, the DCLASS code does not place enough emphasis on form features, which are only assigned one digit. The Opitz code allocates four digits to the description of form features, which is crucial in filtering search results.

Kyprianou proposed and defined a so-called feature grammar describing all possible shape features similar to phrase-structure of Chomsky grammar. It consists of four finite sets for structural primitives, terminal primitives, production rules and initial primitives/starting points [15]. Different approaches for shape grammar and automatic feature recognition of a shape using boundary representation was continued with Braid (1979), Stroud (1980) and later by Cary (1980) [16]. All these approaches are used for downstream application programs such as finite element mesh generation and process planning.

Another popular method is that presented by Liao and Lee [17] as well as Kaparthi and Suresh [18] which uses artificial

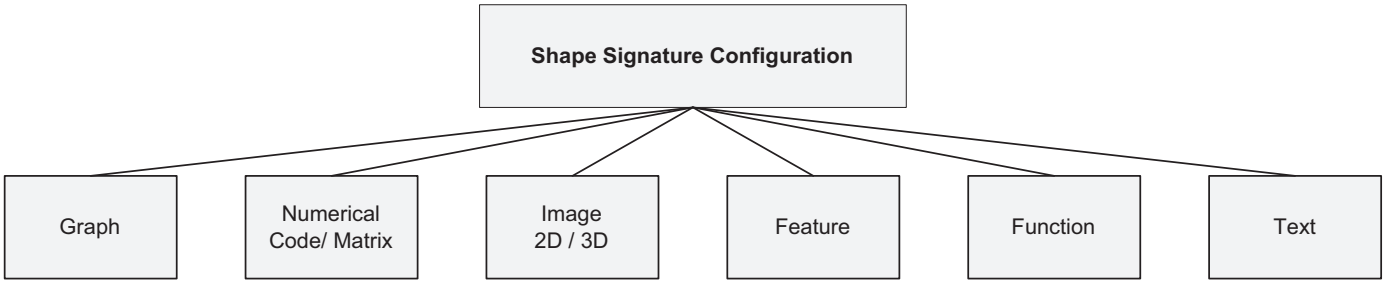

Fig. 3. Various types of shape signature representation.

neural networks (ANNs) to classify part models into groups based on common features. For the purpose of the approach, the authors created their own group technology code which would be used to classify different parts. The network is first initialized and a new pattern (the new model's features) is input into it. Then, the model is compared to the various existing groups in the system, and is assigned to the most similar one in terms of features.

Artificial neural networks are flexible when it comes to classifying models and provide good results for feature recognition. However, they rely on the existence of a large set of training data. The training data are used to teach the ANN how to classify and assign part models to different groups. The drawback of using a machine-learning method in the context of this project is the large number of models that have to be available in order to perform both the training and testing activities. In addition, the approach proposed by Liao and Lee [17] only considers geometrical and topological features. It does not take into account material or accuracy. The use of ANNs also limits the possibilities for expansions. For example, if it were decided that additional information would be needed to describe a model, the ANN would have to be discarded, re-designed and rebuilt, then it would have to be re-trained and re-tested. On the other hand, the Opitz code can simply be expanded by adding several more digits. And given the encapsulation of the functions constructed, only minor changes will be made in order to incorporate the new criteria.

Another method to obtain model signatures is slicing. The signature consists basically of several sections (cross-sections or longitudinal sections). The sections are then used to produce a rudimentary set of measures which can be used as a shape signature. While this approach reduces the effort required to acquire model information, it does not provide enough information regarding relevant features. Especially those pertaining to manufacturing, for example material. Due to the sampling that occurs within this approach, it is very likely that an indentation or a curvature be overlooked, and thus the relevant model is ignored in the retrieval process. However, this method does have advantages; mainly the simplicity of the distance function, as opposed to the cosine similarity and its implementation, as well as the entire feature recognition module. The proposed approach by Jiantao et al. [19] uses 2D slice similarity measurements to establish a model signature, and then compares them using a very basic distance function. Wei et al. [20] proposed a retrieval method to develop the CAD model retrieval system which can be also applied in the partial structure retrieval of CAD model. However this method applies a boundary matching.

In the research form Li et al. [21] a reuse-oriented retrieval method to bridge the gap between CAD model retrieval and reuse however by using a graph-based representation for capturing modeling knowledge embedded in CAD models. The proposed retrieval method by Bai et al. [22] can retrieve reusable subparts queries and support design reuse. However, the method suffers from several issues such as dependency on the design representation and the computational efficiency problem. Another example of graph signature for components has been proposed by Marefat and Pitta [23] which captures the spatial relationships between features by feature interaction graphs as signature.

One of the approaches for graph-based shape signature is proposed by El-Mehalawi and Miller [24,25] to create a topological graph from a STEP, where the nodes correspond to surfaces and the edges correspond to the connecting edge curves between surfaces. Each node may have multiple properties. Another example of the graph-based shape signature has been applied by Ansaldi et al. [26] for building a relational graph structure based on a B-Rep has introduced the dual graph of the object and feature recognition using Face Adjacency Graph (FAG). However, these approaches limit themselves only to geometrical properties. This makes it very hard to establish any manufacturing properties or features. The largest sub-graph between models is searched for and is considered indicative of the similarity between two models (i.e. they share the most common topological and geometrical features). As with previous approaches, this approach is computationally expensive. This is true especially for larger models which consist of a large number of surfaces and connections. This is avoided in the proposed application by simply comparing the Opitz code vectors.

3. Proposed framework: similarity recognition of CAD models

The proposed system architecture is depicted in Fig. 4. The application consists of five basic modules, each of which interacts with other specific modules. There are also two storage modules, the CAD model index (which is internal to the system), and the CAD model repository (which is external to the system). The five basic modules include a GUI module, a feature recognition module, a similarity-retrieval module, a CAD model reader module, and a repository interface module.

The CAD model reader module is responsible for reading CAD model files, and translating them into a set of objects that the application can deal with. This set of objects is later passed to the feature recognition module.

The feature recognition module then uses these objects to identify and recognize features relevant to the required model signature. The resulting signature is then visible through the application's GUI. The resulting signature can also be used as an input to the similarity retrieval module. The similarity-retrieval module will compare the given signature to the internal CAD model Index, all the while exposing the query parameters to the user via the GUI. Once the parameters are set, the user can initiate the search function and a list of models to which the requested parameters apply will be retrieved from the CAD model index. The repository interface module is used to update the CAD model index so that it conforms to the latest version of the CAD model repository. The separation of storage media (local and external) was chosen since it provides greater portability, flexibility regarding different types of repositories, and more efficiency when it comes to query execution. The feature recognition applies a rule-based system (RBS) as a method of automatic feature recognition (AFR). The feature library was constructed according to an Opitz

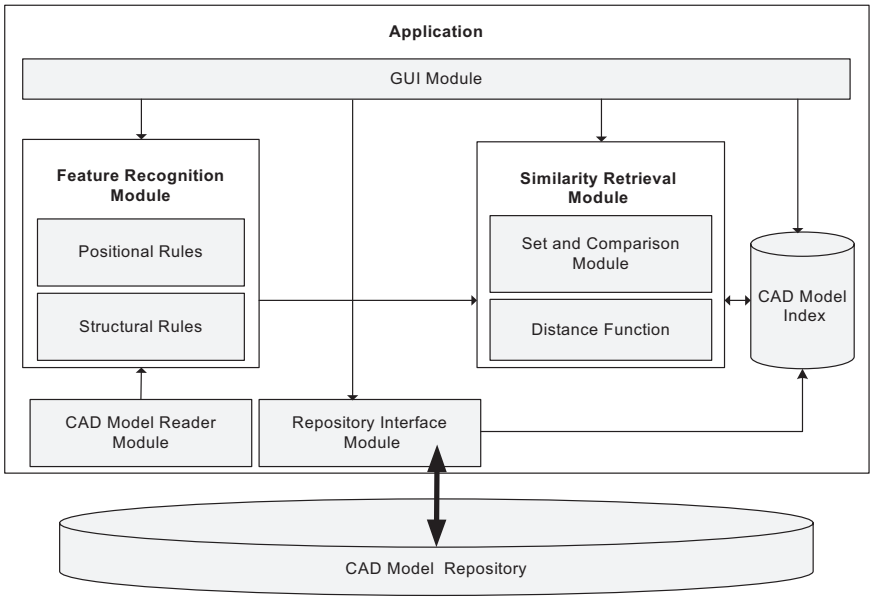

Fig. 4. Application architecture diagram.

Fig. 5. Pin example.

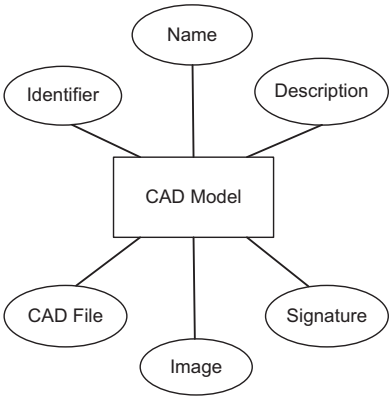

Fig. 6. CAD model repository relationship diagram.

coding system and consequently Opitz features to fulfill the requirements of RBS. The feature recognition module includes two distinct sections, positional rules and structural rules. These two phases are required to drive Opitz features from a CAD file with STEP format as a neutral CAD file format.

This approach is quite similar to the process employed by most graph-based methods. However, the primary difference here is that the end result of the recognition process is not a graph, rather an object that consists only of the required interconnected components that are needed to perform the classification process. Consequently it falls under the two categories of graph-based and heuristics-based approaches. Since the STEP file is constructed in a graph-like fashion, it results in graph-like structured object. This object is then analyzed via predefined rules in order to infer shape and machining features.

The process of generating Opitz code is rule-based in nature. Heuristics are drawn from the Opitz code specification [8], and are converted into predicates. These predicates are then implemented as functional checks within the program's code and are triggered when their conditions are met. For example, the fifth digit of the Opitz code depends largely on whether the part model has teeth (if it is rotational). This is transformed into the following predicate: IF has_teeth THEN fifth_digit < 6.

As an example, in Fig. 5 the shape consists of a cylinder, and two circles forming the top and bottom of the part respectively. The parsing process begins as follows: first, the bottom circle that is parallel to the Y-plane is identified, and used as a reference for the rest of the part's components. Shapes that are connected to the bottom circle are then explored, and their machining (if any) is examined. The process repeats in an iterative manner until there are no more unexplored components. The methodology of these applied methods as a solution for this project has been discussed in detail in Zehtaban and Roller [27].

The predicate is then functionally implemented as a function that analyzes the object mentioned before, and checks for the existence of teeth. The CAD model repository is assumed to be any database on a remote system that contains information about models. The application should be able to interface with it in order to retrieve model data. The retrieved model data would then be used to populate the CAD model index within the application so that the user would always have a local copy which would be accessible at all times. It is also assumed that the basic structure of this database would be as simple as possible. This structure is viewed in the ERD (Entity Relationship Diagram) in Fig. 6.

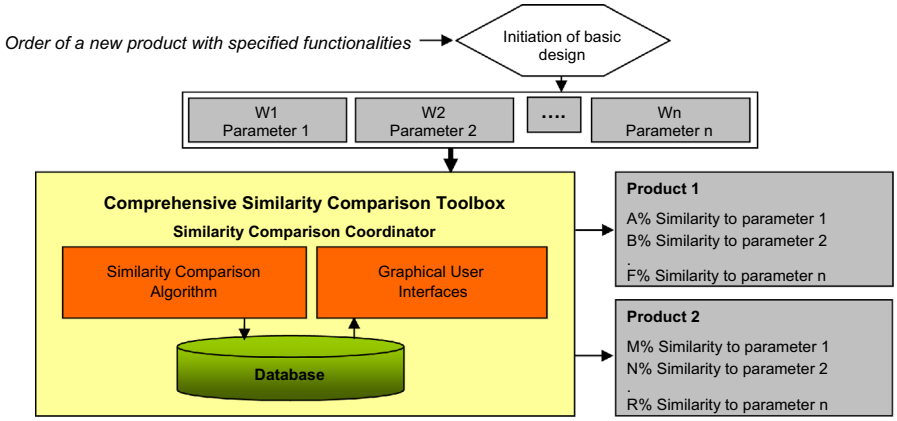

Fig. 7. The conceptual architecture of the similarity recognition [30].

From the ERD above in Fig. 6, and the physical description below, the simplest structure for a CAD model repository is shown. These are the most basic pieces of information that need to be saved in order to retain meaningful information about a model. This information also allows a saved model to be used in the similarity and retrieval functions as a possible search result. The table "CAD Model" consists of six fields in total. The "Identifier" field is a unique surrogate key assigned to each model in order to distinguish it from other existing models. The "Name", "Description", "CAD File", and "Image" fields store the model's name, a short description, the path to its CAD file, and the path to its image file respectively. The "Signature" field is used to store the model's features that have been identified upon insertion. While it may be possible that the remote system contain more fields, these fields are considered mandatory for the application to function. The physical design is given as follows: *CAD_MODEL* (IDENTIFIER, NAME, DESCRIPTION, FILE_PATH, IMG_PATH, SIGNATURE).

4. The similarity retrieval module

The similarity retrieval module includes a distance function. A distance function defines a distance/similarity between each pair of components of a group [28]. If a group consists of shape signatures in the design concept, the distance function would be the unique distance between two shape signatures. Since there are different types of shape signature, Fig. 3, thus the distance function should be properly selected to fit the signature type.

Shape signature applied in this project is based on the features obtained by using the Opitz coding system. Features are presented in an alphanumerical string form. To compute the distance function between two Opitz codes, the cosine coefficient (similarity) method has been applied; see Eq. (1) based on Cha [29]. The cosine coefficient (S_{Cos}) computes the cosine of the angle between two vectors (P and Q). The nominator is the scalar product between the two vectors in question, while the denominator is the product of the norms of the two vectors. Deviating forms the previously introduced functions, this function results in a real number X between -1 and 1 ($-1 \le X \le 1$). The closer the value is to 1 or -1, the more similar the vectors are, and a value moving closer to 0 indicates a growing dissimilarity between both vectors.

$$S_{\mathrm{Cos}} = \frac{\sum_{i=1}^{d} P_i Q_i}{\sqrt{\sum_{i=1}^{d} P_i^2}\sqrt{Q_i^2}} \tag{1}$$

One of the advantages of using the Opitz coding system is its numerical structure. Thus, it is possible to apply one of the distance functions in order to calculate similarity between code vectors. This implicitly means the Opitz code generated by the previous module would be used as a part signature. The cosine similarity is used mainly due to the fact that it always provides a number between -1 and 1. And it can be adjusted to give a number between 0 and 1 indicating a degree of similarity. This is because each number and its additive inverse have the same magnitude, but move in opposite directions. In this context, measuring the direction of the code vectors is not important, their magnitude is. Therefore, the negative solution space of the cosine similarity function can be omitted. By slightly modifying the output of the function, it can yield values between 0% and 100% indicating a level of similarity between two code vectors. This makes it much more user-friendly.

Another benefit that stems from using cosine similarity is that a weight vector may be used in order to emphasize or stress certain digits of the code vector, indicating their importance. This "weight vector" would be multiplied by each code vector, and then they would be submitted to the cosine similarity function, which will then output a similarity value in accordance with the applied weights, Fig. 7.

However, it is not possible to apply the distance function directly to any code vectors. Due to the hierarchical structure of geometrical part of the Opitz code (form code), comparison rules have to be defined. To clarify, an Opitz code with a first digit of 1 cannot be compared with a code that has 8 for a first digit. The reason behind this is that, with the interpretation of the Opitz code taken into account, the following four digits do not mean the same things. For example, when the first digit is 1, the third digit describes the internal shape and when the first digit is 8, the third digit describes principal bores and

rotational surface machining. Hence, the following ranges have been defined:

- first_digit < 3,
- first_digit < 5 AND first_digit > 2,
- first_digit < 9 AND first_digit > 5.

If the first digit of a code falls within one of the previously described ranges, only codes that fall within the same range may be compared to it. Fig. 8 depicts the Opitz decision tree for the fifth digit, when the first digit is less than 9, but greater than 5 (i.e. non-rotational parts).

4.1. Cosine coefficient accuracy

This section is mainly concerned with the effectiveness of the proposed similarity evaluation method, and whether it yields to accurate, comprehensive, and satisfying results. Since the Opitz code is a classification scheme, its numbers do not reflect any numerical relationship. Therefore, it is treated as a string. This makes it a valid target for the application of string comparison functions. The cosine distance is one example of the distance functions used to compare values (not necessarily strings). But it is still applicable to strings [31], if they are converted to their ASCII values and treated as elements of a vector (or a record). So, the principle is sound. Minimum Edit Distance [32] method focuses on string comparison too; however, it was found to be computationally expensive therefore abandoned. Plus, using a numerical distance function allows us to exploit the numbers in the Opitz code with the least computational effort; i.e. no number to string conversion and vice-versa.

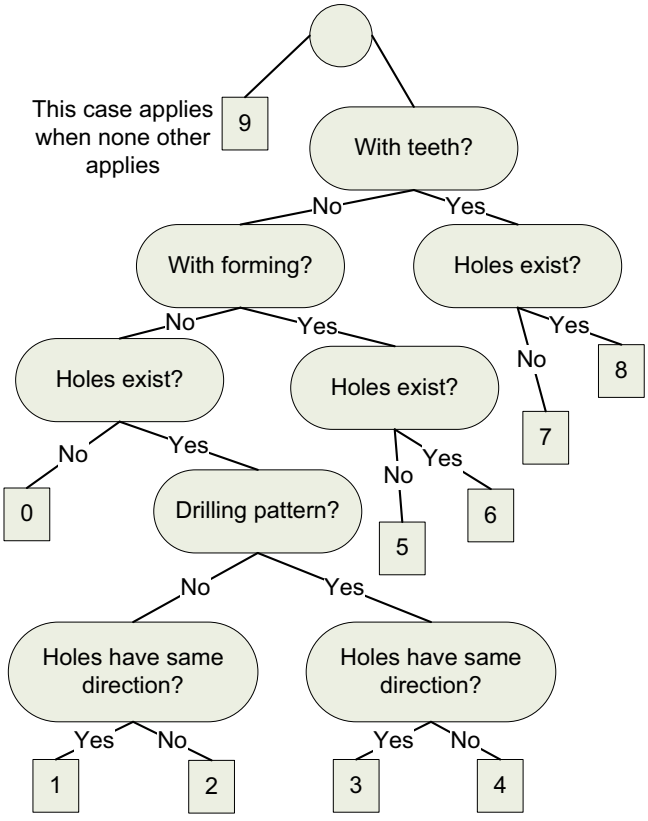

Fig. 8. Opitz code tree for digit 5 (1st < 9).

In addition, it is important to consider that Opitz code is never referred to as a metric. As a matter of fact, in the evaluation, it is not the code itself that is used; rather, the number of instances a digit, or a code was classified correctly. This essentially results in a ratio when the relative frequency is taken into account.

The measure used to evaluate such a function is the *F*-score (Eq. (2)) according to Lu and Callan [33]. The *F*-score is the harmonic mean of the precision and recall, given in Eqs. (3) and (4) respectively. Precision measures the amount of correctly retrieved results out of all results returned, while recall measures the amount of correctly retrieved results from the results relevant to this query.

$$F\text{-score} = 2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \tag{2}$$

$$\text{Precision} = \frac{|r|}{|R|} \tag{3}$$

$$\text{Recall} = \frac{|r|}{|A|} \tag{4}$$

The "*r*" variable in Eqs. (2) and (3) denotes the set of correctly retrieved documents. The "*R*" variable indicates the set of all retrieved documents, while the "*A*" variable indicates the set of all relevant documents. The *F*-score typically results in a value between 0 and 1, 0 being the worst precision and recall, and 1 being the best.

5. Results and evaluation

This section briefly explains the evaluation of the feature recognition module and the similarity-retrieval module. As each module has different evaluation methods, after a detailed description of the evaluation process, the results will be examined and interpreted.

5.1. Feature recognition module

The evaluation method applied by Wester et al. [34] is used for this research. Two basic measures were used to evaluate the performance of the feature recognition approach; accuracy per feature, and accuracy per entire classification. By applying those concepts to this approach, accuracy per feature becomes the accuracy per digit of the Opitz code. The accuracy per entire classification becomes the accuracy per total Opitz code. Thus, an aggregation can be introduced; average accuracy per digit of the Opitz code. The formulas for these measures are shown below, where APD presents accuracy per digit; ND stands for number of instances a digit was classified correctly, TT for total number of test, NC for number of instances the entire code was classified correctly and APC for accuracy per code.

$$\text{Average APD} = \frac{\text{ND}}{\text{TT}} \tag{5}$$

$$\text{Average APD} = \frac{\sum \text{Accuracy per Digit}}{\text{Number of Digits}} \quad (6)$$

$$\text{Average APD} = \frac{\text{NC}}{\text{T T}} \quad (7)$$

It is worth noting that these measures are highly dependent on the test set used to evaluate the system.

5.2. Engineering shape benchmark (ESB)

In regards to different shape representations in mechanical engineering domain for shape based matching and retrieval, there are limited number of standard dataset for the mechanical domain to be applied to benchmark various shape representations. Among the commonly used evaluation methods for shape benchmarking are Princeton Shape Benchmark (PSB) [35] and National Design Repository (NDR) belong to Drexel University [36]. In the current project, one of the most extensive dataset for shape benchmarking, Engineering Shape Benchmark (ESB) developed in Purdue University has been applied based on the advantages discussed by Jayanti et al. [37] PSB includes 867 models in 3D form in three main so called super-class comprising Flat-Thin wall components (107 models), Rectangular-cubic prism (281 models) and Solids of revolution (479 models). Within each super-class, models are additionally classified into groups of similar shapes. Fig. 9(a)–(c) presents some examples of models varieties and classification in three super-classes of ESB including Flat-Thin Wall components, Rectangular-Cubic Prism and Solid of Revolution super-classes.

To evaluate the effectiveness of the algorithm, the conventional precision–recall calculation and plot has been used for the test results. Fig. 10 presents some retrieval results using Opitz codes for ESB. Furthermore, Fig. 11 indicates the efficiency of applying Opitz coding system classification on ESB database by precision–recall curve. In comparison, the precision–recall curve has been presented for some of the most recognized shape representation methods based on [37]. These distinguished methods include: Light Field Descriptor, 2.5D Spherical Harmonics, 2D Shape Histogram, 3D Spherical Harmonics, Convex Hull Histogram, Solid Angle Histogram, 3D Shape Distribution, Surface Area and Volume, Crinkliness and Compactness, Geometric Ratios, Moment Invariants as well as Principal Moments. The tests are run with the default threshold levels, and only take into account the form features recognized from the model file. This means that the supplementary code (i.e. digits 6–9) are not taken into account. This is mainly due to the fact that digit 6 represents the dimensions of the model, which are not hard to interpret. Digits 7 and 8 are supplied by the user. Finally, digit 9 depends on adjusting the threshold levels, which were not adjusted. Furthermore, the correctness of the automatic classified results has been compared with the manual feature classification.

Taking into account the rigidity of rule-based systems, it is found that the resulted values are acceptable for considering the application capable of correct classification with regard to the Opitz code.

Regarding size and dimension, it has to be mentioned that the form code (first 5 digits) does not highlight a part's size. The sixth digit deals with a limited perspective of the part size, where it only takes into account a single dimension. This dimension can be either the diameter (if the part is rotational), or the longest edge (if the part is non-rotational). However, this limitation can be solved by extending the Opitz code beyond its original 9 slots, and make use of the additional 4 slots to provide more descriptive measures. The comparison mechanism is flexible enough to allow the consideration of extra digits.

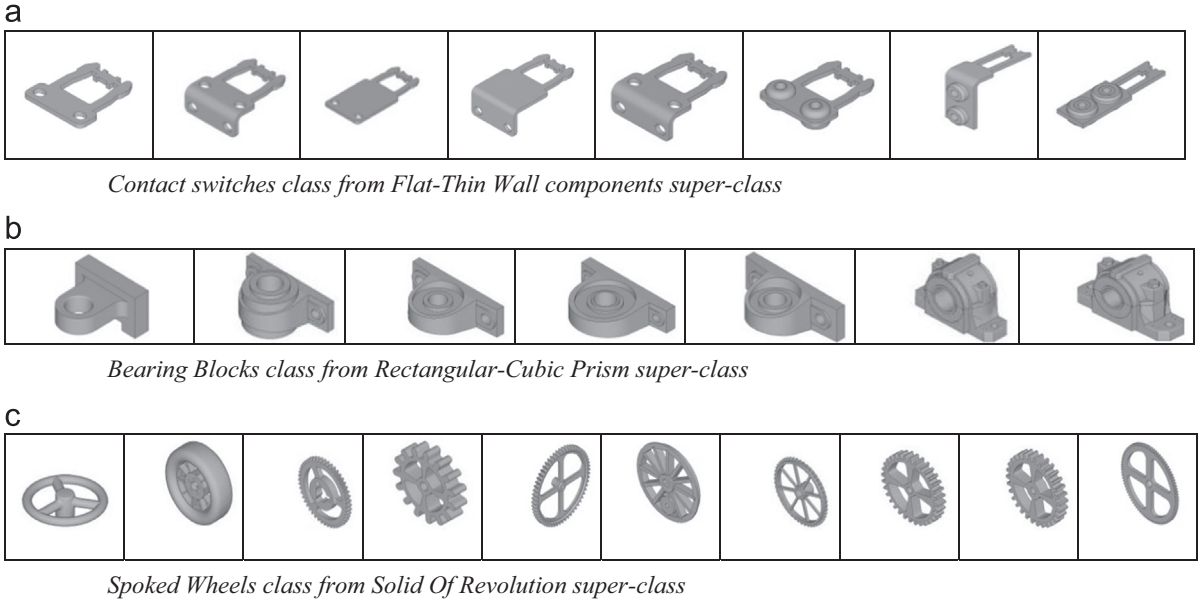

Fig. 9. (a)–(c) Examples of ESB super-class clusters.

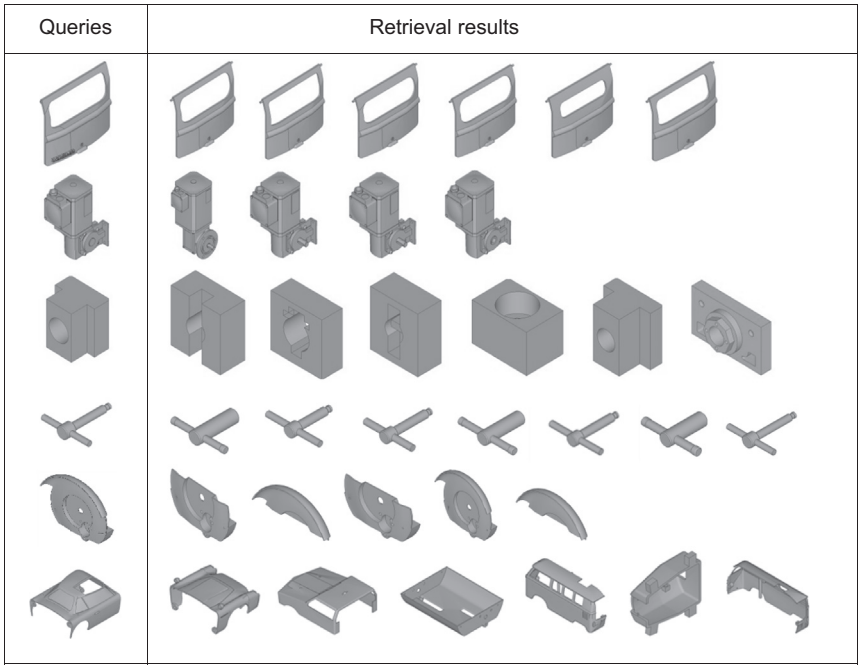

Fig. 10. Retrieval results using Opitz classification system.

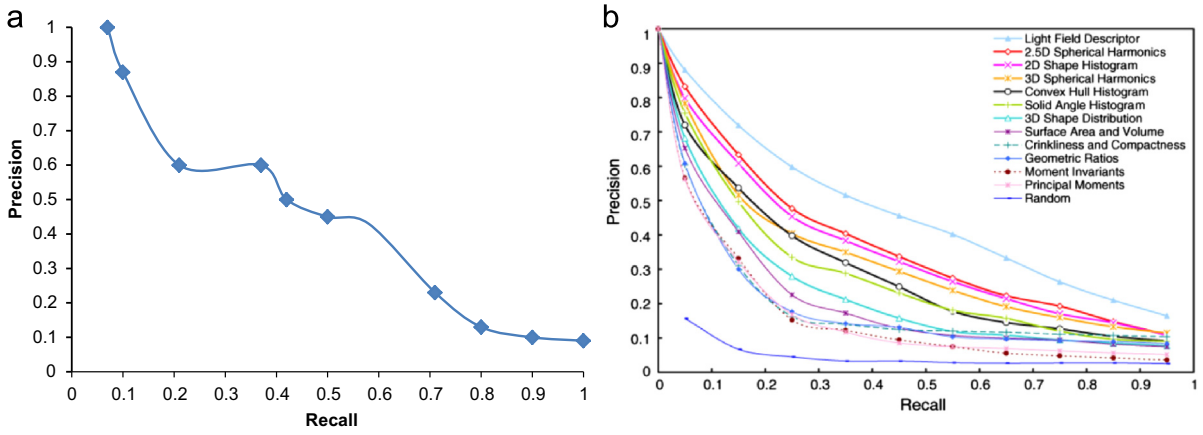

Fig. 11. (a) Precision–recall curve for the proposed method. (b) Precision–recall curve for well-known shape representation methods based on [37].

5.3. Similarity-retrieval module

All the queries were performed using a similarity threshold of 60%. The results are as follows:

- Average precision = 0.450
- Average recall = 0.613
- Average F-score = 0.517

The moderate precision indicates that the application was able to retrieve correct results, albeit not as specifically as one would prefer. That is highly dependent on the query itself, as well as the models currently used as a search space. The higher recall value indicates that among the models retrieved, a high percentage thereof were actually relevant to the query. Finally, an F-score of 0.517 indicates a moderate effectiveness of this approach as a search and retrieval technique.

One of the issues observed with using the Opitz code to classify objects, is the fact that it does not reflect all of the information illustrated in the part-model. For example, when classifying teeth (in the fifth digit), the classification for triangular teeth is the same as that for elliptical teeth so long as both are parallel to the rotation axis. They will both have the same code, but they are different part-models which result in different parts/products and may eventually be used in completely different ways. This is, however, not a failing of the application, rather one of the Opitz code itself, which does not go into such rigorous detail.

The previous results also illustrate the aforementioned issue regarding using the Opitz code. Since it is a group technology code, it inherently groups together models that are not necessarily 100% similar. Since each enumeration of the code signifies a “group” of models, it cannot be treated as a unique identifier, thereby broadening the spectrum of retrieved results.

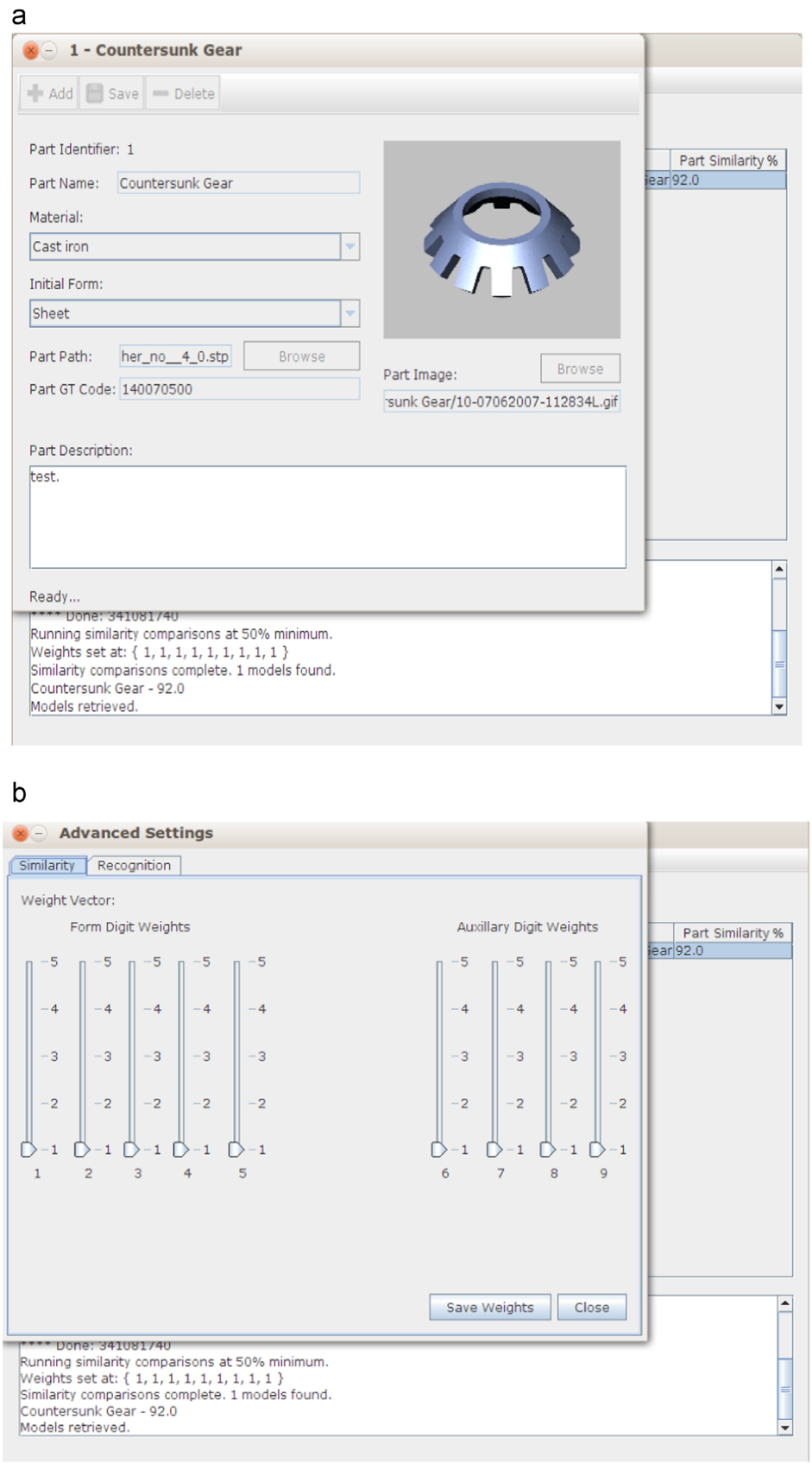

Fig. 12. (a) Resulting CAD model. (b) Advanced settings for weights.

This is the main reason for the low precision value encountered during the tests. However, it is compensated by the high recall value, which indicates that a vast majority of (if not all) relevant models have been retrieved.

5.4. GUI module

When the Opitz code is known, either generated by the system through the wizard, or the designer has already set a design as a basic design, several scenarios were defined to administrate the functions of the GUI. These scenarios encompass all model-related operations and depict the user's interaction with the model repository as well as the search results. At this point, the search results are considered the same as a model repository, but filtered. A menu bar should be available at the top to give the user access to more advanced features, such as calibrating the weight settings, adjusting the flexibility levels for the recognition process, and connecting to a repository. It should also allow access to the local storage medium, so that the user can browse the existing models and view their details.

The user has the possibility to set similarity threshold of his choice in the main window, or predefines weights for the comparison process which can be set in the "Advanced Settings" screen, seen in Fig. 12(b).

And here is an excerpt of the log output shown in Fig. 12(a):

**** Start: Rotational gear with quart teeth/21_25_012_0.stp
Rotational shape
More than 3 external cylinders
External shape… Machined
Holes found: 2
Teeth found, holes found.
Rotational: using diameter as 'measure' at a value of
27.496164321899414
**** Done: 341081740
Running similarity comparisons at 50% minimum. Weights set at: { 1, 1, 1, 1, 1, 1, 1, 1, 1 } Similarity comparisons complete.
1 models found. Countersunk Gear − 92.0
Models retrieved.

Using a STEP file as input is not always an option. For these cases, a user can apply the recognition wizard. This wizard is launched from the main screen and proceeds by asking the user questions regarding the features which will be translated and converted into an Opitz code.

6. Conclusions

The main objective of this paper is to present a framework to support similarity and retrieval system that would assist to search for and retrieve similar models from a given CAD model. It was also required that the resulting system be evaluated as well as validated against other existing endeavors in this area. The system would, in turn, allow inexperienced designers to relate their textual requirements via wizard or rough model to existing models and give them something to start with instead of rebuilding models from scratch (re-inventing the wheel). It was shown that the approach is sound when it comes to using an Opitz code as a shape signature. It was also shown to be consistent method into some of the popular approaches in that field, and its issues have been outlined and discussed. It has also been shown that its issues do not impede its main functionality, whether these are incorrectly classified models, or the fact that using an Opitz code as a signature returns many more results than expected. Rather, these issues can be used to show that the system indeed does perform as expected, and has room for improvement. However, due to the limitations of the Opitz code approach, models that are not completely similar may be considered similar.

Using this method it is possible to realize partial similarity retrieval. By giving a larger weight to a certain property, and relaxing the similarity threshold, the similar models with regards to the certain property are retrieved. Although the other properties will also be compared, but the weights will significantly influence the similarity values.

The work presented in this research is similar in its reliance on rule-based systems to generate the group technology code, but different in how it is applied. Instead of using predicates to generate feature information, the application simply runs through a decision tree. The first applicable path is selected and followed to its end in a forward chaining manner. This reduces the memory utilization usually consumed by storing predicates and conclusions of such predicates. As seen in the evaluation section, this process is slightly rigid. This has been somewhat mediated by the use of thresholds and flexibility levels. Though the overall code accuracy is not high, the accuracy per feature as well as the average accuracy per feature measures are promising.

References

[1] Bucherta T, Neugebauerb S, Schenkerc S, Lindowa K, Starka R. Multi criteria decision making as a tool for sustainable product development benefits and obstacles. *J. Procedia CIRP* 2015;**26**:70–5.
[2] Kristianto Y, Gunasekaran A, Helo P, Sandhu M. A decision support system for integrating manufacturing and product design into the reconfiguration of the supply chain networks. *J. Decis. Support Syst.* 2012;**52**(4)790–801.
[3] Liu E, Hsiao SW, Hsiao SW. A decision support system for product family design. *J. Inf. Sci.* 2014;**281**:113–27.
[4] F. Gao, D. Roller, Modelling of feature-based design process, in: Proceedings of ASME Design Engineering Technical Conferences, 13–16 September, Atlanta, USA, 1998, 331.
[5] Iyengar SS, Lepper MR. When choice is demotivating: can one desire too much of a good thing?. *J. Personal. Soc. Psychol.* 2000;**79**(6)995–1006.
[6] S.S. Iyengar, How to Make Choosing Easier, 2011. Available from: ⟨http://www.ted.com/talks/sheena_iyengar_choosing_what_to_chooset-53993⟩.
[7] Mogilner C, Shiv B, Iyengar SS. Eternal quest for the best: sequential (vs. simultaneous) option presentation undermines choice commitment. *J. Consum. Res.* 2013;**39**(6)1300–12.

[8] Opitz H. *Verschlüsselungsrichtlinien und Definitionen zum werkstück-beschreibenden Klassifizierungssystem*. Essen: Girardet; 1966. [In German].
[9] Opitz H. *A Classification System to Describe Workpieces*. New York: Pergamon Press; 1970. Translated by Acton Taylor.
[10] Hong T, Lee K, Kim S. Similarity comparison of mechanical parts to reuse existing designs. *Comput.-Aided Des.* 2006;**38**:973–84.
[11] Roller D. Design by features: an approach to high level shape manipulations. *J. Comput. Ind.* 1989;**12**(3)185–91.
[12] Shannon S. *Trends in Computer Science*. New York: NOVA Science Publisher, Inc.; 2004.
[13] Salvendy G. *Handbook of Industrial Engineering Technology and Operations Management*, 18. New York: John Wiley & Sons, Inc.; 2001.
[14] Henderson M, Musti S. Automated group technology part coding from a three-dimensional CAD database. *J. Manuf. Sci. Eng.* 1988;**110**:278–87.
[15] Kyprianou LK. *Shape Classification in Computer-Aided Design*. 1980. Doctoral Dissertation.
[16] Pickett MS, Boyse JW. *Solid Modeling by Computers: From Theory to Applications*. New York, USA: Plenum Press; 1984.
[17] Liao T, Lee K. Integration of a feature-based CAD system and an ART1 neural model for GT coding and part-family forming. *J. Comput. Ind. Eng.* 1994;**26**:93–104.
[18] Kaparthi S, Suresh N. A neural network system for shape-based classification and coding of rotational parts. *Int. J. Prod. Res.* 1991;**29**:1771–84.
[19] P. Jiantao, L. Yi, X. Guyu, Z. Hongbin, L. Weibin, Y. Uehara, 3D model retrieval based on 2D slice similarity measurements, in: Proceedings of the 2nd International Symposium on 3D Data Processing, Visualization and Transmission, 6–9 September, Thessalonoki, Greece, 2004, pp. 95–101.
[20] Wei S, Tie-Qiang M, Li G. A new CAD models retrieval method based on shape similarity. *Inf. Technol. J.* 2009;**8**:708–16.
[21] Li M, Zhang YF, Fuh JYH, Qiu ZM. Toward effective mechanical design reuse: CAD model retrieval based on general and partial shapes. *J. Mech. Des.* 2009;**131**(12).
[22] Bai J, Gao S, Tanga W, Liu Y, Guo Y. Design reuse oriented partial retrieval of CAD models. *Comput.-Aided Des.* 2010;**42**:1069–84.
[23] M. Marefat, C. Pitta, Similarity-based retrieval of CAD solid models for automated reuse of machining process plans, in: Proceedings of the 3rd Annual IEEE Conference on Automation Science and Engineering Scottsdale, 22–25 September, AZ, USA, 2007.
[24] El-Mehalawi M, Miller R. A database system of mechanical components based on geometric and topological similarity. Part I: representation. *J. Comput.-Aided Des.* 2003;**35**:83–94.
[25] El-Mehalawi M, Miller R. A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching and similarity assessment. *J. Comput.-Aided Des.* 2003;**35**:95–105.
[26] Ansaldi S, De Floriani L, Falcidieno B. Geometric modeling of solid objects by using a face adjacency graph representation. *ACM SIGGRAPH Comput. Gr.* 1985;**19**(3)131–13
[27] L. Zehtaban, D. Roller, Automated rule-based system for Opitz feature recognition and code generation from STEP, Computer-Aided Design and Applications, Taylor & Francis, USA, ISSN 1686-4360, Published online: 16.12.15.
[28] Mešina M, Roller D, Lampasona C. *Mapping of Semantic Distances into Geometrical Coordinates – Visualisation of Semantic Networks*, 1st ed., Achen (DE): Shaker Verlag; 397–408.
[29] Cha SH. Comprehensive survey on distance/similarity measures between probability density functions. *Int. J. Math. Models Methods Appl. Sci.* 2007;**1**(4)300–7.
[30] Zehtaban L, Roller D. Beyond similarity comparison: intelligent data retrieval for CAD/CAM designs. *Comput.-Aided Des. Appl.* 2013;**10**(5)789–802.
[31] M. Bilenko, R. Moone, Adaptive duplicate detection using learnable string similarity measures, in: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003), August, Washington DC, USA, 2003, pp. 39–48.
[32] K. Audhkhasi, A. Verma, Keyword search using modified minimum edit distance measure, in: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP, 15–20 April, 2007, ISSN: 1520-6149, IV-929-IV-932.
[33] J. Lu, J. Callan, Content-based retrieval in hybrid peer-to-peer networks, in: Proceedings of the 12th International Conference on Information and Knowledge Management, 3–8 November, New Orleans, USA, 2003, pp. 199–206.
[34] M. Wester, J. Frankel, S. King, Asynchronous articulatory feature recognition using dynamic Bayesian networks, in: Proceedings of International Conference on Spoken Language Processing (ICSLP), 4–8 October, Jeju Island, Korea, 2004, pp. 1477–1480.
[35] P. Shilane, P. Min, M. Kazhdan, T. Funkhouser, The princeton shape benchmark, in: Proceedings of the Conference on Shape Modeling International, 7–9 June, Genoa, Italy, 2004, pp. 167–178.
[36] D. Bespalov, C. Yiu Ip, W. Regli, J. Shaffer, Benchmarking CAD search techniques, in: Proceedings of the 2005 ACM Symposium on Solid and Physical Modeling (SPM'05), New York, USA, 2005, pp. 275–286.
[37] Jayanti S, Kalyanaraman Y, Iyer N, Ramani K. Developing an engineering shape benchmark for CAD models. *Comput.-Aided Des.* 2006;**38**:939–53.