Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 86 (2019) 104–109

www.elsevier.com/locate/procedia

7th CIRP Global Web Conference, Towards shifted production value stream patterns inference of data, models and technology

AI-based Computer Aided Engineering for automated product design - A first approach with a Multi-View based classification

Carmen Krahe[a] *, Maximilian Iberl[a], Alexander Jacob[a], Gisela Lanza[a]

[a]Karlsruhe Insitute of Technology, Kaiserstrasse 12, 76131 Karlsruhe, Germany
* Corresponding author. Tel.: +49-721-608-44011 ; fax: +49-721-608-45005. E-mail address: carmen.krahe@kit.edu

Abstract

Today's success of industrial companies is largely determined by engineering competence and the digitization of all corporate processes. The design process and know-how of engineers is strongly individual and a rule-based description of their approach can often not be done at all or only with high effort. Existing knowledge can therefore only be passed on to other engineers with difficulty, which in particular increases the effort required for familiarisation. A further problem is the lack of an overview of existing components within a company, which very often leads to multiple designs and unnecessary waste of time for the engineer. The aim of this approach is to extract the implicit knowledge from existing CAD models with the aid of machine learning methods and thus to make it formalizable. In addition, a suitable classification and similarity analysis should quickly point out existing components. For this purpose, an AI-based assistance system is to be created. Based on the existing database, the assistant first points out to the engineer already existing, but very similar components. For that, the component type currently in construction firstly is identified and then very similar components are searched within the detected scale that are finally suggested to the engineer. The engineer now only has to parameterize the proposed components according to his application. In a further step, the assistant should also be able to suggest useful next design steps, which it has learned on the basis of the CAD data already available and their design history. The implicit experience knowledge that is contained in the existing CAD models thus ensures a design suitable for production and the avoidance of errors in the design.

Keywords: "Computer aided design (CAD); Artificial Intelligence; Machine Learning"

1. Introduction

The success of companies is determined by its grade of innovation [1]. This refers to innovation regarding their respective market offering (e.g. product, service) as well as their internal value stream design. Hence, core competences for industrial companies are their product engineering capabilities and production processes.

Due to digitization huge efforts are undertaken to improve production and logistic systems as well as internal organizational processes. Nonetheless, the field of product development and design is, due to its complex, experience based and unstandardized tasks, harder to improve by means of digital assistance systems [2]. Especially the individual knowledge about product design, the respective link to the product function and the historic knowledge about similar product generations offer huge potential for improving product development and design process [3], if made transparent and available to a complete team of engineers.

An approach to achieve this desired transparency and even suggested automated possible design solutions based on individual knowledge and historic product generations will be presented in this paper. Due to the inconsistent and large data sets as well as the goal to transfer such a support system to

10.1016/j.procir.2020.01.038

different product categories, an AI-based approach is necessary.

2. State of the Art

The design process is a highly individualized process, which is not strictly defined according to fixed rules and therefore is strongly based on experience. Until now, design engineers could orient themselves on a generally defined procedure according to VDI 2221 [4], which divides the design process into several phases. However, the design of the phases and corresponding methods must be implemented by the designer on a case-by-case basis. This often requires experience which is usually implicit and cannot be formalized. Additionally, there are also approaches using non-automated system models for product feature ideation based on product generation data [5].

Current approaches for AI-based processing of CAD models concentrate mainly on classification tasks. These approaches can be differentiated according to the degree of automation. On the one hand, hand-crafted descriptors are used to encode certain features of the model [6] and then process them using machine learning methods. For example in [7], CAD models are firstly clustered by the k-means algorithm. In a second step, those clusters can also be used for classification. On the other hand, there are also approaches with which the features to be considered can be learned automatically from the input data. The latter comprise approaches in which multi-view-images [8,6], voxel models [9], point clouds [10,11] or graphs [12] serve as input for Convolutional Neural Networks (CNNs).

With the multi-view approach, images of the 3D component are generated from different perspectives. The images are then classified using classical CNNs for image processing. For example in [8], very high accuracy values for the ModelNet database introduces by [13], that contains 3D CAD models for objects of the most common categories, are achieved. Their basic idea is to predict the class in conjunction with the perspective from which the image was created.

Conventional CNNs can also be used for the voxel-based approach. This is based on approximating the objects through a cube-shaped grid, as in [9]. In analogy to pixels, the generated voxel structures are classified by CNNs with three-dimensional convolution kernels.

In point-cloud-based approaches, the coordinates of points distributed randomly in the volume or on the surface of the model are used as input for CNNs. The result of the classification must be independent of the input order of the points. To achieve this invariance, [10,11] use symmetric functions.

Graph-based methods are also based on randomly distributed points, where the graph is generated by inserting edges between spatially close points. For the investigation of graphs spectral methods exist, as shown in [14]. With these, however, the input graph structure is fixed. As an alternative, [12] have presented an approach for edge-conditioned convolution in which a certain signal is assigned to the vertices as a function of adjacent vertices and incident edges. The recognition of local features in the convolution layers is done by pooling points or vertices spatially close to each other in both point-cloud and graph-based methods.

To compare the similarity of 3D models, a variety of approaches exists. A widely used approach is based on feature vectors. The 3D models are described using these vectors and their similarity is ultimately determined by the distance between the vectors. In [15], parts are displayed as 13-digit Opitz code, which is interpreted as a vector. Similarity between

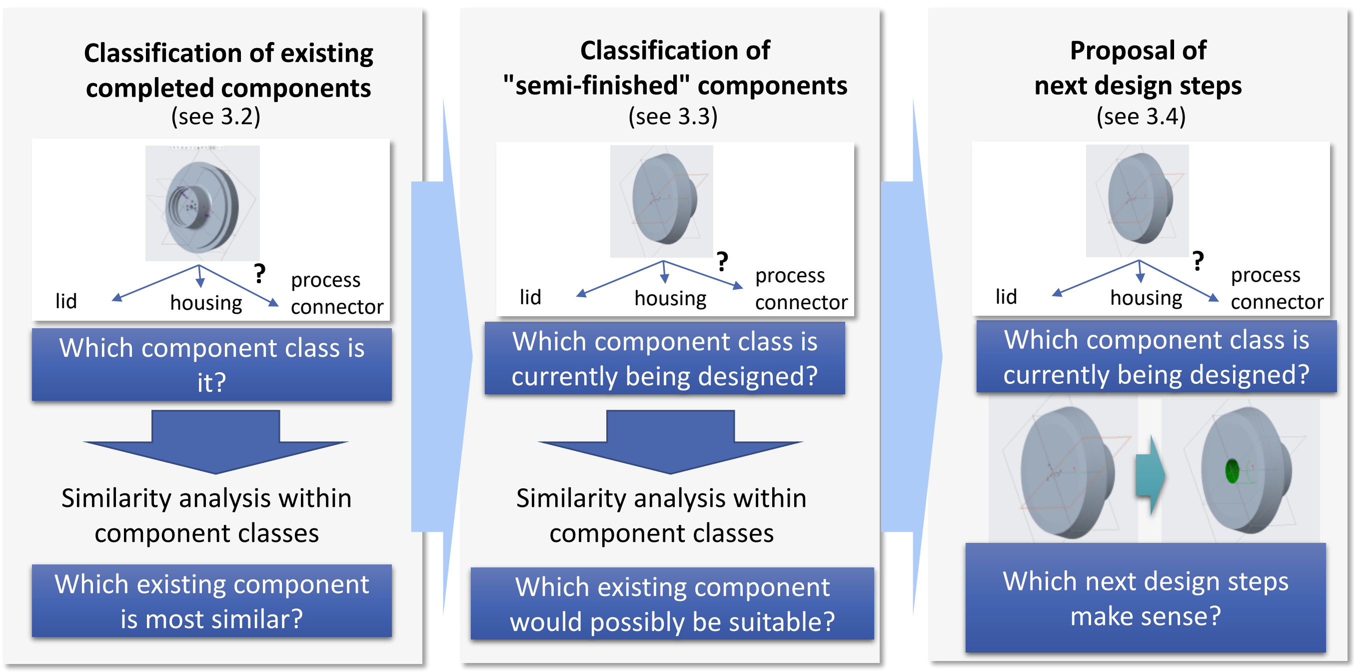

Fig. 1. Development steps of the investigated approach. First, the existing database is categorized to enable the designer to find the most suitable part for his purpose. In a next step, components currently in process are classified in order to propose already existing, very similar components to the designer. In the last step, the explicit next x design steps are suggested, which can then be parameterized at the appropriate point on a case-by-case basis.

two parts is then the cosine similarity of the Opitz vectors. Alternatively, in [16] a shape distribution for each 3D model is determined, i.e. the frequency of the distance between two arbitrary points is plotted over the distances. The components then are classified by comparing this shape distribution. Based on the volume of detailed features a similarity index of two components can be calculated. Further approaches, e.g. [17–19], deal with the similarity of components described by point clouds.

The existing approaches regarding classification and similarity analysis support the designer in gaining an overview of the large amounts of data, avoiding redundancies and therefore shortening reaction times. However, the actual design process and its implementation remains in the hands of the designer and his knowledge. Hence, in this paper an approach is developed on how existing knowledge can be formalized to be passed on to less experienced designers and how it can be used to support repetitive activities, which do not require creativity.

3. Methodology

The investigated design assistance can be divided into three development stages with increasing complexity, see Fig. 1. The approach is first applied to individual components, in the following called components. The aim is to extend the approach to assemblies as well. The individual components are represented by their 3D CAD models, which in the following are the basis for input and output of the developed approach.

In order to identify already existing, very similar CAD models in the data base of a company, in a first step a classification of the different product groups is necessary (see 3.2). In a next step, the classification of semi-finished components must be implemented in order to suggest suitable, already existing CAD models to the designer during the design process on the basis of the design steps already carried out (see 3.4). On the one hand, the right category must be predicted and on the other hand, the probably most similar component(s) within the class must be proposed. In the final stage of development, it is no longer a question of complete parts, but of suitable next design steps (see 3.5). The aim is to present the procedure to the designer as comprehensibly as possible and to disclose the adaptation of case-specific parameters to the corresponding design steps. In the following, 3.1 first describes the database under consideration and its information content in more detail. This is followed by a more detailed description of the three steps mentioned above. In addition, a first implementation approach is presented for step one, the classification task.

3.1. Considered data base and information content

In order to apply machine learning to 3D models, it is essential to understand how 3D data can be stored. According to [20], one can basically distinguish two formats here: Boundary representation (B-Rep), in which the surfaces of the body are stored, and Constructive Solid Geometry (CSG), in which the body is stored as a combination of basic bodies (e.g. cuboids, cylinders). Commercial CAD software normally uses a CSG-based storage format internally. Individual steps that a user carries out to create a basic body, including the parameterized dimensions, are saved. By the parameterization it is possible to subsequently make changes in the basic body. A widely used format is the so-called PART (.prt) format, as used in the approach presented in this paper. In addition to purely geometric information, this file format also contains information about the procedure of the design process, which is represented in the so-called structure tree (also model tree or element tree). It contains the individual design steps, so-called

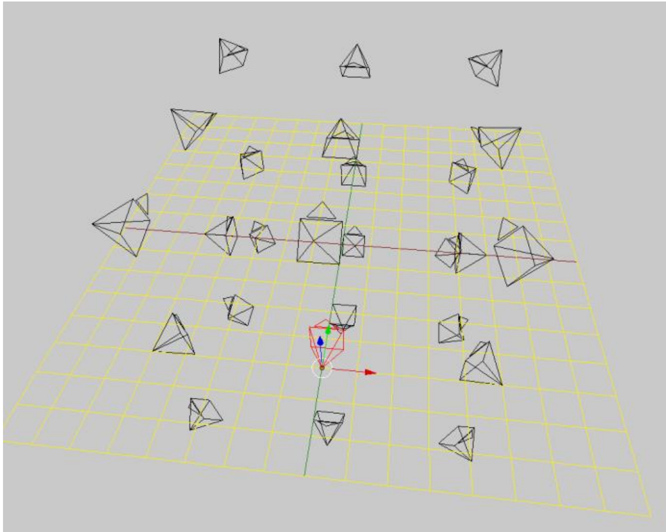

Fig. 2. Alignment of the 26 perspectives regarding the basis perspective (red).

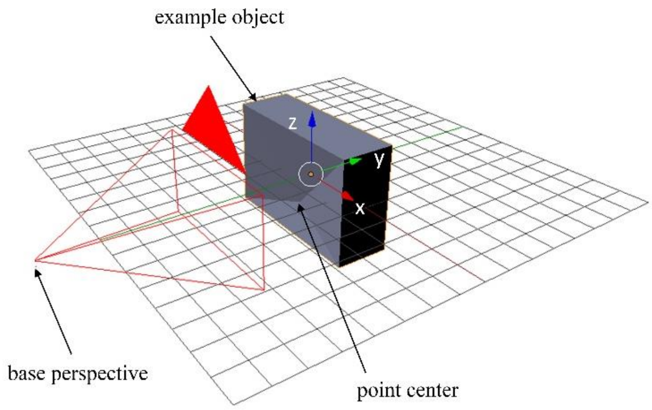

Fig. 3. Alignment of an example object according to the standard orientation.

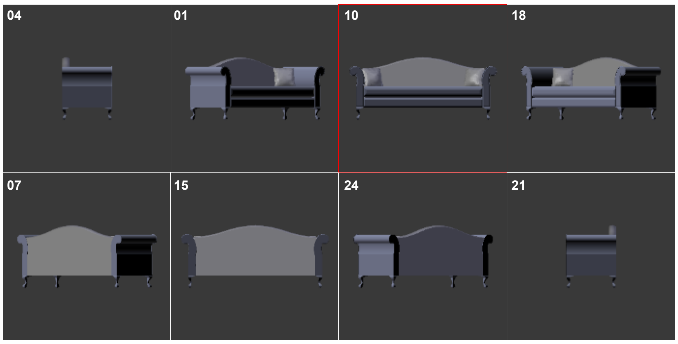

Fig. 4. Views of a sample object from the basis perspective (red) and the 7 other perspectives of the same plane.

features (e.g. drilling) with the corresponding parameterization (e.g. diameter of the drilling). From already existing CAD models, the structure tree can be used to extract and formalize the procedure of experienced designers when designing parts of a certain class. The structure tree can, for example, be read out in the form of an XML file.

In contrast, B-Rep formats focus on file exchange between different CAD programs, but information about the design process is lost during conversion to this exchange format. From this file format, only shape information can be extracted.

With regard to the described development steps of the approach, the pure information about the shape is merely sufficient for the first step, the classification. For the two further steps, the information about the design history is additionally necessary.

3.2. Classification of components

Since the existing CAD models in a company are often stored unstructured, they must be assigned to corresponding component classes in a first step. As mentioned in section 2, there already exist several approaches for classifying 3D-models based on different shape descriptors. Moreover, this classification service is offered by commercial suppliers (e.g. SIMILIA, see [7]).

Within the scope of this work, the classification was implemented based on the Multi-view approach, which is purely based on shape information. In contrast to [6] and [8], a standard orientation of the input data is defined in order to maintain connections between perspectives, since the view of an object from a certain perspective is always the same.

The determination of the standard orientation is based on the minimum bounding box of the 3D objects. The basic idea is that objects of the same class have similar proportions of length, width and height. By rotation and displacement of the minimum bounding box according to certain regulations, a standard orientation can be determined. Subsequently, the corresponding images of the 3D object can be generated via virtual cameras from the defined perspectives. For this purpose 26 cameras were implemented according to Fig. 2 (numbered from 00 to 25). Fig. 3 shows an example object aligned in the standard orientation with respect to the base perspective. The standard orientation is defined by five criteria: (1) The center of the bounding box lies in the origin of the coordinate system, (2) the basic perspective shows the largest side surface of the box, (3) the longest side of the box is parallel to the x-axis, the point center of the polygon mesh representing the 3D object has a (4) y-value and a (5) z-value less than or equal to 0. In Fig. 4, an example of a sofa with pictures of the base perspective (framed in red) and the seven further perspectives located in the same plane is depicted.

Finally, these images serve as input for perspective-specific CNNs. For this purpose, the images of two opposing perspectives are summarized, so there are only 13 CNNs for the 26 perspectives. Each CNN classifies the input data based on the two image perspectives. For the final classification of the object, the probabilities for each perspective-specific CNN are summed up and the object is assigned to the class with the highest probability.

3.3. Classification of semi-finished components

The classification of semi-finished components is essential in order to be able to propose to the design engineer corresponding existing, very similar parts during the design process. A semi-finished part is considered as the part currently in process. To classify these semi-finished components, the approaches presented in section 2 will be adapted accordingly. An essential difference compared to conventional classification is that any missing possible design steps must first be predicted in order to determine the corresponding component class. It is possible that several classes are suitable from the current state of the design process. The more advanced the current design status, the more clearly a component can be classified. In the approach developed, this classification is to be implemented by training the existing approaches with additional semi-finished components. Since normally the intermediate states of CAD models are not stored individually in an enterprise data base, these must be artificially generated. For this purpose, the features in the structure tree are successively removed and the corresponding intermediate statuses of the CAD model are stored, respectively.

By being able to classify semi-finished parts during the designer's work, the design assistant can see what type of part the designer is currently designing and searches in the existing CAD models for parts that are very similar, so that the designer gets a first draft of the finished part and only needs to make case-specific adjustments if necessary.

3.4. Proposal of next design steps

In order to propose the next explicit design steps, it is initially also necessary to classify the semi-finished part in order to determine which component type is present. The methodology from 3.3 is used for this purpose. Once the part class has been defined, proposals for next design steps can be made based on the design procedure learned for this class. In order to learn this component class-specific procedure, the structure trees are first extracted from all existing CAD models of the corresponding class. With the help of these structure trees, appropriate machine learning procedures are trained (e.g. random forest) to find a common design pattern.

The aim of the design assistant is to offer several alternatives to choose from. In order to ensure sufficient traceability, the probability, with which the assistant considers the proposal suitable, is also be stated. The advantage over 3.3 is that the designer is given as much scope for creativity as possible. By suggesting only the next steps, the designer can immediately make any adjustments at the appropriate point. The next proposed steps are finally based on these adjustments. In contrast to 3.3, the designer is therefore not bound to an existing product as a starting point for his adaptations.

4. Case Study

The modified Multi-view approach described in 3.2 for component classification has been validated using the ModelNet10 database, which is a subset of the ModelNet database introduced by [13]. The ModelNet10 database

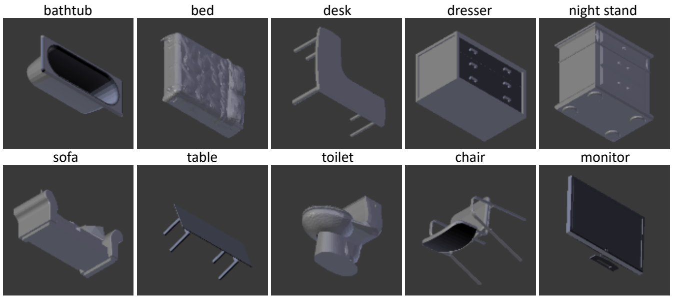

Fig. 5. Exemplary 3D models of the 10 different categories of ModelNet10.

contains 4899 3D models of various furnishings in 10 different categories, see Fig. 5. Consequently, the classification task includes the assignment of models to 10 different categories.

First, three different CNN architectures were tested using the images of only one perspective. One architecture is based on the AlexNet architecture introduced in [21], one on the modified VGG-16 network of [22], and one is an own configuration. Fig. 6 shows the self-developed architecture. The rounded rectangles contain the names and parameters of the layers, e.g. in the first convolutional layer 32 kernels of size 3x3 and the Rectified Linear Unit (ReLU) as activation function are used. The labels of the arrows indicate the input and output formats of the layers, e.g. the input image has a size of 256x256 with three channels.

Despite the large number of parameters, the performance of the depicted architecture is clearly better compared to the other two architectures, which is why the results for this one are reported. With images from a single perspective, an accuracy of 80.0% for the test data is achieved. One reason for the worse performance of the latter two architectures could be that they were designed to distinguish a much larger number of classes.

Due to the results obtained for images from a single perspective, the self-developed architecture was also used for all other perspective-specific CNNs. Each of these CNNs was configured as shown in Fig. 6. The test accuracy for the individual perspectives ranged between 68.4% and 84.0%, depending on the perspective (see also Fig. 7 column *all classes*). By combining the individual forecasts through averaging over all perspective-specific class predictions, a final test accuracy of 88.4% was achieved for the model. The corresponding effect can be seen in Fig. 7. For each object class, the accuracy of the individual perspectives and the overall accuracy after taking all perspectives into consideration (red x) are displayed.

Fig. 8 shows the overall results for all perspectives in form of a confusion matrix. Obviously, in some classes the assignment is often not quite clear. For example, only 58% of desks are correctly classified as such, but a large proportion are falsely classified as sofas. One reason for this is that to some extent objects of different classes look very similar from particular perspectives (e.g. the square base of a sofa or desk). From this it becomes clear that a purely visual view of the objects may not be sufficient, but their structure should also be analyzed. Graph-based approaches would be suitable for this and should be considered in further investigations. The aim is to perform the classification on real CAD data from an industrial measurement technology provider that is available in PART format. Subsequently, the structure tree is extracted

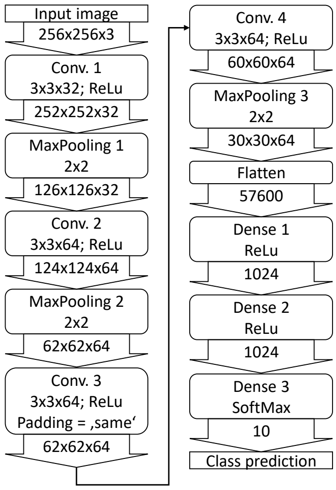

Fig. 6. Architecture of the self-developed CNN.

Fig. 7. Percentage of correct classification by category and perspective. The colors represent the perspective-specific CNNs.

	bathtub	bed	chair	desk	dresser	monitor	night stand	sofa	table	toilet
bathtub	78%	0%	0%	0%	0%	0%	0%	0%	0%	0%
bed	8%	98%	0%	2%	0%	2%	0%	2%	0%	0%
chair	0%	0%	100%	8%	0%	0%	0%	0%	0%	0%
desk	2%	0%	0%	58%	2%	0%	0%	0%	12%	0%
dresser	0%	0%	0%	2%	84%	0%	16%	0%	0%	0%
monitor	2%	2%	0%	0%	0%	98%	0%	0%	0%	0%
night stand	2%	0%	0%	4%	10%	0%	82%	0%	0%	0%
sofa	8%	0%	0%	18%	2%	0%	0%	98%	0%	0%
table	0%	0%	0%	6%	0%	0%	2%	0%	88%	0%
toilet	0%	0%	0%	2%	2%	0%	0%	0%	0%	100%

Fig. 8. Confusion matrix with actual classes in the columns, predicted classes in the rows. The proportion per class that was classified correctly or incorrectly is displayed.

from these CAD models and the further steps of the design assistant are implemented.

5. Conclusion and Outlook

This paper develops a three-step approach for a CAD design assistant based on suitable machine learning methods. In the first development stage, the assistance system can only classify components in the form of 3D CAD models. An own approach is presented for this purpose, which first converts the 3D models into a standard orientation and then generates 2D images from various perspectives. Those perspective-specific images finally serve as input for classification via CNNs, as shown in a case study. In further research, additional CNN architectures should be investigated, especially with a lower number of parameters. In addition, other approaches, e.g. graph-based, should be considered to better map the structural design of the component. In the next development stage of the approach the classification of semi-finished components is considered, so that existing, very similar components can be suggested to the designer during the design process. In the final development step, a general procedure or design pattern for each class is learned from the design history contained in the CAD models using suitable machine learning methods.

Acknowledgments

This paper was also funded by the Federal Ministry of Education and Research (BMBF) project AIAx, Machine Learning-driven Engineering – CAx goes AIAx (01IS18048B).

References

[1] Srinivasan, S., Pauwels, K., Silva-Risso, J., Hanssens, D.M., 2009. Product Innovations, Advertising, and Stock Returns. Journal of Marketing 73 (1), 24–43.

[2] Negri, E., Fumagalli, L., Macchi, M., 2017. A Review of the Roles of Digital Twin in CPS-based Production Systems. Procedia Manufacturing 11, 939–948.

[3] Albers, A., Haug, F., Fahl, J., Hirschter, T., Reinemann, J., Rapp, S., 2018 - 2018. Customer-Oriented Product Development: Supporting the Development of the Complete Vehicle through the Systematic Use of Engineering Generations, in: 2018 IEEE International Systems Engineering Symposium (ISSE). 2018 IEEE International Systems Engineering Symposium (ISSE), Rome. 01.10.2018 - 03.10.2018. IEEE, pp. 1–8.

[4] VDI, 1993. Systematic approach to the development and design of technical systems and products.

[5] A. Albers, A., Bernijazov, R., Kaiser, L., Dumitrescu, R., 2018. Internet of Things Canvas for Ideation in Model-Based Product Generation Planning, in: 2018 13th System of Systems Engineering Conference (SoSE). June 19-22, 2018, Sorbonne université, campus Pierre et Marie Curie, Paris, France. 2018 13th Annual Conference on System of Systems Engineering (SoSE), Paris. 6/19/2018 - 6/22/2018. IEEE, Piscataway, NJ, pp. 327–334.

[6] Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E., 2015. Multi-view Convolutional Neural Networks for 3D Shape Recognition, in: 2015 IEEE International Conference on Computer Vision. 11-18 December 2015, Santiago, Chile : proceedings. 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile. 12/7/2015 - 12/13/2015. IEEE, Piscataway, NJ, pp. 945–953.

[7] SIMUFORUM® Search Solutions GmbH. https://www.simuform.com/. Accessed 12 June 2019.

[8] Kanezaki, A., Matsushita, Y., Nishida, Y., 2016. RotationNet: Joint Object Categorization and Pose Estimation Using Multiviews from Unsupervised Viewpoints, 24 pp. http://arxiv.org/pdf/1603.06208v4.

[9] Sedaghat, N., Zolfaghari, M., Amiri, E., Brox, T. Orientation-boosted Voxel Nets for 3D Object Recognition, in: Procedings of the British Machine Vision Conference 2017. British Machine Vision Conference 2017, London, UK. British Machine Vision Association.

[10] Charles, R.Q., Su, H., Kaichun, M., Guibas, L.J., 2017. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation, in: 30th IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2017 : 21-26 July 2016, Honolulu, Hawaii : proceedings. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI. 7/21/2017 - 7/26/2017. IEEE, Piscataway, NJ, pp. 77–85.

[11] Charles Ruizhongtai Qi, Li Yi, Hao Su, Leonidas J. Guibas, 2017. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space, in: NIPS.

[12] Simonovsky, M., Komodakis, N., 2017. Dynamic Edge-Conditioned Filters in Convolutional Neural Networks on Graphs. http://arxiv.org/pdf/1704.02901v3.

[13] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J., 2015. 3D ShapeNets: A deep representation for volumetric shapes, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). - 12 June 2015, Boston, MA. 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA. 6/7/2015 - 6/12/2015. IEEE, Piscataway, NJ, pp. 1912–1920.

[14] Defferrard, M., Bresson, X., Vandergheynst, P., 2016. Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering.

[15] Zehtaban, L., Elazhary, O., Roller, D., 2016. A framework for similarity recognition of CAD models. Journal of Computational Design and Engineering 3 (3), 274–285.

[16] Hong, T., Lee, K., Kim, S., 2006. Similarity comparison of mechanical parts to reuse existing designs. Computer-Aided Design 38 (9), 973–984.

[17] Huang, J., You, S., 2012. Point cloud matching based on 3D self-similarity, in: IEEE Computer Society Conference on Computer Vision and Pattern Recognition workshops (CVPRW), 2012. 16 - 21 June 2012, Providence, RI, USA. 2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPR Workshops), Providence, RI, USA. 6/16/2012 - 6/21/2012. IEEE, Piscataway, NJ, pp. 41–48.

[18] Lima, J.P.S.d.M., Teichrieb, V., 2016. An Efficient Global Point Cloud Descriptor for Object Recognition and Pose Estimation, in: 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images - SIBGRAPI 2016. São José dos Campos, Sao Paulo, Brazil, 4-7 October 2016 : proceedings. 2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI), Sao Paulo, Brazil. 10/4/2016 - 10/7/2016. IEEE, Piscataway, NJ, pp. 56–63.

[19] Zhong, Y., 2009. Intrinsic shape signatures: A shape descriptor for 3D object recognition, in: IEEE 12th International Conference on Computer Vision workshops (ICCV workshops), 2009. Kyoto, Japan, 27 September - 4 October 2009. 2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops, Kyoto, Japan. 9/27/2009 - 10/4/2009. IEEE, Piscataway, NJ, pp. 689–696.

[20] Roj, R., 2016. Eine Methode für eine automatisierte Informationsextraktion aus großen CAD-Datenbeständen zur Bauteilsuche und Klassifikation. Dissertation, Wuppertal.

[21] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), 84–90.

[22] Simonyan, K., Zisserman, A., 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. http://arxiv.org/pdf/1409.1556v6.