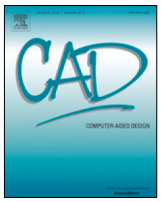
Computer-Aided Design 154 (2023) 103417

Contents lists available at ScienceDirect

Computer-Aided Design

journal homepage: www.elsevier.com/locate/cad



A Novel Shape Retrieval Method for 3D Mechanical Components Based on Object Projection, Pre-Trained Deep Learning Models and Autoencoder

S. Bickel *, B. Schleich, S. Wartzack

Engineering Design, Friedrich-Alexander-Universität Erlangen-Nürnberg, Martensstraße 9, Erlangen, 91058, Germany

ARTICLE INFO

Article history:
Received 26 May 2022
Received in revised form 19 August 2022
Accepted 9 September 2022

Keywords:
3D object retrieval
Shape retrieval
Projection method
Part alignment
Deep Learning
Autoencoder

ABSTRACT

The reuse of existing design models offers great potential in saving resources and generating an efficient workflow. In order to fully benefit from these advantages, it is necessary to develop methods that are able to retrieve mechanical engineering geometry from a query input. This paper aims to address this problem by presenting a method that focuses on the needs of product development to retrieve similar components by comparing the geometrical similarity of existing parts. Therefore, a method is described, which first converts surface meshes into point clouds, rotates them, and then transforms the results into matrices. These are subsequently passed to a pre-trained Deep Learning network to extract the feature vector. A similarity between different geometries is calculated and evaluated based on this vector. The procedure employs a new type of part alignment, especially developed for mechanical engineering geometries. The method is presented in detail and several parameters affecting the accuracy of the retrieval are discussed. This is followed by a critical comparison with other shape retrieval approaches through a mechanical engineering benchmark data set.

1. Introduction

Product customization is a global megatrend, with companies seeking to meet customers' needs through mass customization. This trend continues to be fuelled by the development of new manufacturing processes such as 3D printing, which results in more products than ever to be generated, while also increasingly more design models are available. In addition to these developments, case studies for specific product development domains [1,2] show the increasing need for reusing existing designs to achieve a faster and more efficient process [3]. The multiple benefits of using existing product models according to [4] include improving product quality by increasing product variability and eliminating unnecessary components, resulting in fewer trials overall.

In order to utilize existing geometry models, a powerful method for finding similar components is necessary, since a simple text-based search using keywords is not practical for the geometry search [5]. Consequently, various methods have been proposed, which rely on the labelling of existing data, followed by supervised learning to determine the shape descriptor. In product development, this approach is not suitable, because the geometry is often very domain-specific and the effort to manually label the data is not economically viable. For this reason, a method is presented in this paper, which allows finding similar components from the field of mechanical engineering, without a previous labelling of the data. This allows to benefit from the results of a similarity search, without the preliminary effort of tagging existing geometry.

Specific properties of the mechanical engineering parts are utilized by a uniform alignment in combination with the new method to improve the retrieval results. The new method is then compared with established algorithms to demonstrate the benefits of the approach.

2. State of the art

The general term similarity is subjective and can be defined in multiple ways by each individual, depending on acquired knowledge, decisions or the environment. The meaning also depends on the respective domain, such as movies, images, texts or music [6]. However, all sectors agree that similarity is always related to comparable characteristics in several cases [7]. As described in the introduction, one of the most important results of virtual product development is the 3D geometry of the product, since almost all product properties depend on geometric features. Therefore, the geometric similarity is primarily considered in this paper [8]. In the context of geometric similarity, the terms *shape description*

* Corresponding author.
E-mail address: bickel@mfk.fau.de (S. Bickel).

https://doi.org/10.1016/j.cad.2022.103417

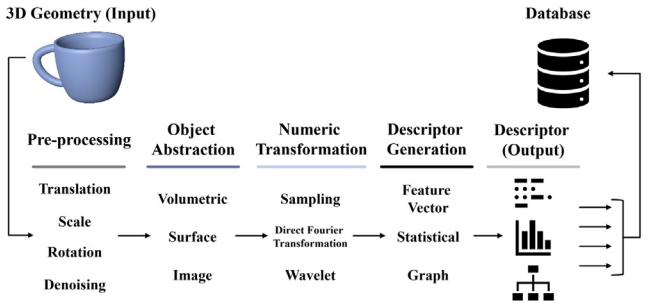


Fig. 1. General procedure of the shape descriptor generation based on [15].

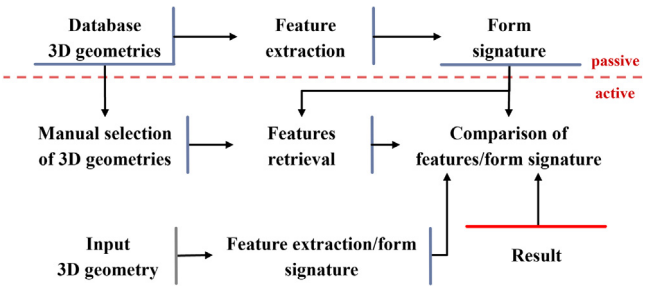


Fig. 2. Example framework for a similarity search method according to [16].

and *shape representation* are commonly referred to as a geometrical representation for a similarity comparison. There are different definitions in the literature for both expressions. In this work, the definition of [9] is applied, which describes shape representation as a non-numeric representation of the original shape and shape description as a numeric description of the shape, also called shape descriptor.

This shape descriptor provides the foundation for a geometric similarity search since it transforms the geometry into a uniform and comparable structure. The procedures for generating this shape descriptor are manifold and differ significantly in their function. Therefore, the following section introduces the general explanation of the process for a similarity search and explains the various ways of generating the shape descriptor.

2.1. Shape similarity search

The overall purpose of geometric similarity search is to provide the best matching geometry as an answer for a given input. The search input is distinguished by [10] between solid and shape models. Solid methods apply engineering features from solid geometry models for the component search, for example manufacturing information. Well-known methods are described in [3,11,12].

The focus of this publication is on shape-based methods, in which the polygon mesh of a geometry is normally the input for the algorithm that transforms the shape into a descriptor through different techniques. Shape-based methods have been categorized by [13] depending on the input type, which differs between a 2D sketch and a 3D model as a search query. A good overview of sketch-based methods can be found in [14]. In this paper only search by query methods are considered, since it is one of the most common shape retrieval category [13] and geometry models are an essential result in the product development process. Furthermore, a classification is made between rigid and non-rigid retrieval methods. Non-rigid models are more relevant for medical or animated film applications; in the case of product development, the focus is on rigid models as mentioned by [13]. After the positioning of the paper in the research field, the typical process for the geometry transfer is described, with a special focus on the transformation of geometry into a shape descriptor.

2.1.1. General procedure

The general procedure for a similarity search is described by [15], which addresses the problem of heterogeneous geometry representations. First, the geometry is processed, then abstracted and transformed. Finally, a shape descriptor is generated as the result, which is stored in a database linked to the original geometry. This forms the basis for the subsequent similarity search, which is illustrated in Fig. 1.

Various procedures and methods can be employed for each partial step, for example in the pre-processing stage translation, scaling, rotation or denoising of the geometry can be performed according to [15] or other procedures such as remeshing. The remaining methods of the sub steps are listed in Fig. 1. The different shape descriptors are then compared through calculating the distance between them. This comparison enables to find the most similar vector and therefore the matching geometry.

Several frameworks have been developed to implement a similarity search. Examples can be found in [16,17]. The passive part of the framework is usually carried out in advance and contains the existing database including the components, converted into their shape descriptors. The active part in the process architecture represents the search query through the input geometry and the result determination. A comparison between the input geometry and the parts from the database is carried out and the results are then displayed in ranked form (see Fig. 2). [17] extends this framework with a web-based search and the possibility to allow multiple inputs for the search; besides geometry also keywords or images. For the shape comparison, various methods have been developed to calculate the shape descriptors, which are presented in the following chapter. Ready-to-use solutions for part search are also offered as stand-alone software or as plug-ins for CAD systems. 3DPartFinder [18] and iSeek [19] can be deployed universally, whereas shape-search [20] is only useable in Siemens NX.

2.1.2. Overview of techniques for calculating the shape descriptor

Over the past two decades, different categorizations for the creation of shape descriptors have been presented. Well-known examples are [5,15,17,21]. The distinction in this paper is based on a more recent classification by [13]. There are four categories: geometry-based, graph-based, view-based and hybrid techniques. In the context of product development feature-based and grouping technologies (GT) methods are also applicable [22]. However, these require BRep data as an input and are therefore not considered further in this paper. Although only parts are examined and not assemblies, more information to assembly retrieval is presented in [23].

Geometry-based methods use the distribution of geometric elements on a component for identification. Features can be generated globally or locally for each geometry. Examples for global methods are shape histograms [24] and for local methods extend gaussian image [25].

The graph-based methods transform the geometry into different graph structures. This technique is often chosen for BRep geometry, for example [26] uses this principle for step-files. According to [13], the application to surface meshes can be divided into skeletal and topological graphs. The methods for skeletal graphs abstract the geometry into a graph depending on the attributes, such as the popular method from [27]. Topological graphs take advantage of the differences in the topological structure of geometry for comparison. A well-known example is the Reeb graph from [28] which is based on Morse theory.

Another form of conversion relies on rendered images from the geometry as the basis for generating the shape descriptor. Afterwards, the similarity between the generated images is calculated. Several methods have been developed for this purpose, some extended through alignment or multi-view approaches. A very popular representative of this type is the Light-Field Descriptor [29]. The bag feature approach [30] is frequently deployed and has been advanced several times, for example by KAZE from [31] or SIFT [32]. A more detailed summary of view-based methods can be found in [33].

The last category includes the combination of two approaches from the three previously mentioned types. This category is called hybrid technologies and is correspondingly diversified. An example of a hybrid method is the PANORAMA method of [34]. This approach uses panorama images and derived parameters for the geometry search. Another method from this field is DBNAA DERE [35], which combines several methods, including shape distribution, bounding boxes and depth buffer.

In addition, new developments in the field of Machine Learning (ML) have led to the appearance of innovative approaches for geometry search. Especially the evolution of Convolutional Neural Networks (CNN), as an important variant of Deep Learning, has resulted in many new methodologies. This type of network is inspired by human vision and has been given a boost by the increasingly powerful computers of the early 2010s. Important representatives are the LeNet [36], AlexNet [37] and ResNet [38]. For a recent overview of the available methods and functions in the context of geometry, see [39].

When applying Deep Learning methods to geometry, classification and retrieval are often examined together. As the name "Deep Learning" suggests, all methods require a large data set to learn the respective model. For the training of a classification model labelled data is necessary, which means that all components have to be assigned to classes before the process starts. This way of training is called supervised learning.

Unsupervised learning, in contrast, needs no labels for training a model on the existing data. Developed methods in this segment often use shape descriptors as an input and combine them with Autoencoders, for example [40]. In general, it is possible to categorize Deep Learning methods the same way as presented above, but now the category specifies the input for the Deep Learning model. For example, the Multi-View CNN (MVCNN) by [41] is considered a view-based method, as several images are transferred to a Machine Learning model. In contrast, the PointNet by [42] is characterized as a geometrical method, as the point cloud directly serves as the input for the neural network. Based on these two criteria, the distinction between the input of a descriptor and the type of training (label required, unsupervised, no training necessary) was examined and categorized for several published approaches.

2.1.3. Results of the literature review

All analysed procedures and their respective categories can be seen in the appendix in Table B.6. The listing includes a total of 200 publications, the author, the title of the method, the mode of function category and the type of training. A summary of the literature is provided in a diagram in Fig. 3.

In this diagram, the timeline is displayed on the X-axis and is divided into four time periods, starting before 2006 and ending in 2022. The number of published procedures is plotted on the Y-axis. The four categories for the operating principle are shown as stacked bars, with the graph in yellow, hybrid in light blue, view-based in blue and geometric in dark blue. The number of approaches separated by the learning procedures are plotted as lines with the colour differentiation: "Label required" black, "Unsupervised Learning" dark blue and "no Training" grey. The term "Label required" includes all methods that need more input than just the data set of geometries for the part retrieval, which includes user feedback, supervised, semi-supervised and weakly supervised methods.

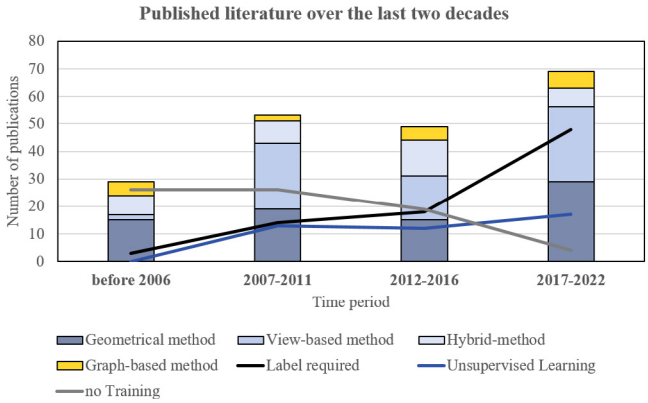


Fig. 3. Overview of the number of publications over time, sorted into categories.

The diagram illustrates how the general number of publications in this field is increasing, indicating a continuing and growing interest in the topic. Two trends are clearly visible in the line graphs of the different learning methods: At the beginning of the 2000s only methods without training were established and increased over time until they decreased with the appearance of Machine Learning. Procedures based on learning methods begin to emerge at the beginning of the 2010s and then increase considerably. This is generally explained due to the higher performance capabilities of newer computer generations and the better accessibility of Machine Learning algorithms. The ability to integrate shape classification and retrieval into a single approach explains the rise of supervised learning procedures.

In the distribution of the different functional principles, it is obvious that geometric based methods had the largest share. The view-based methods have started to grow alongside with the Machine Learning methods through the rising popularity of CNNs and exceed the geometric based methods in the latest time period. The share of hybrid methods declined slightly, which could be because Machine Learning methods often only use one input format. Finally, the graph-based methods play a subordinate role in the distribution.

Summary and research gap. This excessive literature study demonstrates that view-based approaches in combination with supervised Deep Learning networks are able to achieve good retrieval results, which justifies their recent popularity. But the supervised approach does not fulfil the requirement of product development for a component search: the search for similar parts without labelling of existing data. For this purpose, the supervised methods cannot be applied. However, to incorporate the good retrieval results of supervised image-based methods, pre-trained models from the field of image recognition were chosen, which are freely available for download. They support the creation of the shape descriptor through feature extraction and should improve the overall retrieval results, without the additional need of part labels. In the following, the new procedure is presented in detail and the individual steps are explained.

3. Presentation of the shape retrieval method

The method relies on surface meshes as an input and adopts the idea from [43], which transfers geometry into matrices by applying a projection algorithm and then ranks the results according

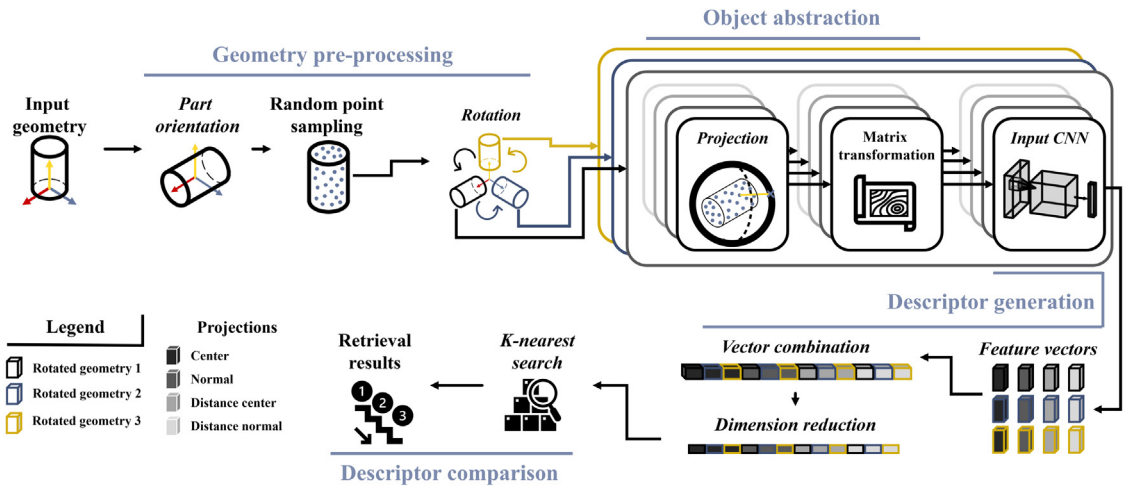


Fig. 4. Overview of the new process for finding similar parts.

to their similarity through different comparison metrics. This process was enhanced extensively, especially through the addition of different rotations of the geometry, extra projection variants and a new way of generating the shape descriptor. An overview of the method is given in Fig. 4, the new steps are highlighted in italic type. The rotation principle is based on multi-view methods, which rotates the geometry by 90 degrees and then transforms each rotated geometry into a matrix. In addition, matching for similarity is achieved with the help of pre-trained Deep Learning networks. These compute the feature vector, which is passed to a k-nearest-neighbour (knn) search algorithm to find the most similar feature vector. The process is described in its entirety in the next chapters, starting with an overview of the steps.

3.1. Overview

The pre-processing of the input polygon mesh is the first step and is followed by a conversion of the mesh into an evenly distributed point cloud with a defined number of points. The next stage, the object abstraction, performs the transformation into a matrix. For this purpose, the point cloud is projected onto a detector sphere in four different ways and then converted into matrices. This conversion is conducted for each rotation, so in total 96 (24×4) matrices are generated in this step. In the numeric transformation phase, each matrix is fed into a pre-trained Deep Learning network to extract the feature vector. Subsequently, the different rotations and projection methods are merged into one large feature vector that is condensed through dimension reduction and then passed to a knn-search algorithm. This examines the similarity of the input vector using different distance metrics and determines the most similar vector. Accordingly, the matching component is retrieved from the database.

3.2. Geometry pre-processing

To start the pre-processing, the geometry must be imported as a surface mesh. Then a significant stage in the procedure is executed: the component alignment. This is necessary since CAD products are never aligned in an identical way in the application. Because the entire process is view-based and consequently part rotation-dependent, part alignment is essential prior to the transformation. In industrial practice, part alignment is dependent on many variables, such as assembly position, boundary conditions or the design environment. Various methods for component alignment have been developed, but in the context of this paper two of them will be investigated. One is the commonly used principal component analysis (PCA), and the other is a specially developed orientation method which is based on the design process of components in a CAD program.

Afterwards, the aligned surface meshes are converted into a point cloud, whereas the number of points is adjustable. The amount of 50.000 points per part model has proven itself to be a good compromise between accuracy and processing time. This is shown in the appendix in Figs. A.16 and A.17, where the computation times for a whole data set for various point cloud numbers and an example matrix for different amounts of points are displayed. The point cloud is then rotated and submitted to the next step, the object abstraction.

Data: Faces and Vertices of imported Part
Result: Get largest combined Faces in Mesh
[*faces*, *vertices*] = `importMesh(`*filename*`)`;
[*VNew*] = `RotatePartToLargestFace(`*faces, vertices*`)`;
[*BBVert*,*BBFaces*] = `BoundingBox(`*faces*,*Vnew*`)`;
IdxNormalBB = `normals(`*BBVert*,*BBFaces*`)` == *NormalTo*;
AreaBB = sum(`AreaFaces(`*BBVert, BBFaces(IdxNormalBB)*`)`);
if *AreaBB/faceMaxArea* $>$ 0.12 **then**
  VAligned = `alignPCA(`*vertices*,*faces*`)`;
else
  DirTo = `getLargestEdge(`*Input*`)` ;
  VAligned = `Rotated3D(`*VNew*,*DirTo*,*[1,0,0]*`)`;
end
[*Ixx*, *Iyy*] = `polySecAreaMoments(`*VAligned*`)`;
[*DistPolar*] = `polySignature(`*VAligned*,*10*`)`;
if *Ixx/Iyy* == 1 & `max(`*DistPolar*`)`−`min(`*DistPolar*`)` $<$ 0.3 **then**
  VAligned = `alignPCA(`*faces*,*[VAligned(1:2)*, 0]`)`;
end
`checkXAlignment(`*VAligned*`)`;

Algorithm 1: General overview of the algorithm

3.2.1. New alignment method

The application of PCA is particularly difficult with mechanical engineering parts, as these can have multiple holes or cutouts leading to inconsistent alignment results, caused through an uneven point distribution on the part. Also, mechanical engineering components are usually designed according to a well-defined pattern not represented in the PCA. Additionally, many components are not symmetrical, which means that the alignment of the components is hardly ever flat on a plane of the global coordinate system. These problems are considered by the new procedure for

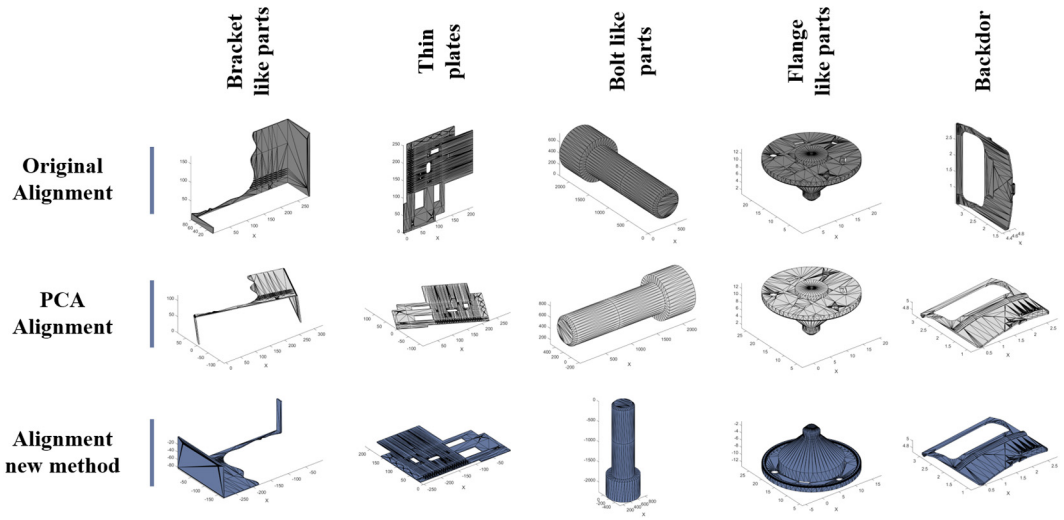


Fig. 5. Examples of different alignment methods.

component alignment. This approach is inspired by the design principle of mechanical components, in which the following rules should be obeyed: Build in Z direction, start the part design in the X–Y plane and point the longest edge in X direction. The new process is supposed to lead already existing components back into this structure as well as possible. The general procedure is explained with Algorithms 1 and 2. For the new alignment method, it is first necessary to identify the largest adjacent faces of the geometry with the same normal direction, which is then aligned with the Z-axis. A rotation around the Z-axis is carried out with the goal of the longest edge of the surface running in parallel to the X-axis. The shorter edge is consequently aligned in parallel to the Y-axis. If the largest face is round or symmetrical, a PCA is conducted only on the X and Y vertices. Finally, the geometry can only be rotated by 180°, whereby the side with the larger Z-value should be closer to the origin. If the largest surface in the first check is too small in comparison to the largest surface of the bounding box of the part, the mesh is aligned through the PCA and all possible further steps are skipped.

3.2.2. Example of different alignment results

After the theoretical description of the procedures, the different results for the alignment are presented with several components. For this purpose, three alignments of five components are shown in Fig. 5: the original alignment in the data set, aligned by PCA and by the new method. The components are taken from the Engineering Shape Benchmark (ESB) data set, which is presented in more detail in chapter 4.1.

The figure illustrates that for non-uniform parts, such as the first and second mesh, the new method is more consistent compared to the PCA. Looking at the rotational components the difference is less obvious, the distinguishing feature mainly being the alignment to a specific axis (PCA: X-axis; new method: Z-axis). The last example demonstrates the case where the area of the largest face is too small. In this case the alignment is carried out by PCA alone, leading to identical alignment results. These exemplary parts are intended to show the advantages of the new method in comparison to PCA for mechanical engineering parts. Due to the uniform alignment of the meshes, components should theoretically be better recognizable.

3.3. Object abstraction

Each rotated point cloud is then converted into a matrix. The procedure is based on our preliminary work [43] and shares the basic idea with [44], which is the projection of points onto a detector sphere. The sphere is divided into sections called pixels, comparable to division through longitude and latitude degrees of a globe. Then the points are projected onto the sphere and the number of projections per pixel are counted. The detector sphere is unfolded like a map creating the matrix for the subsequent investigations. The exact configuration of the detector sphere is explained in [45]. This method differs from other view-based approaches like [41,46,47] through the matrix generation. In the previous named methods, the input image for the CNN is rendered from a mesh or a point cloud. In contrast, our approach uses the projection in different directions, followed by the sum or distance of the projected points per pixel.

An important parameter for this projection type is the number of pixels in accordance to the matrix size. This parameter can be theoretically varied as desired, whereby an evaluation must be conducted between the accuracy of the projection and the computational effort. Another way to influence the transformation is the type of projection for which four different ways have been developed: centre, normal, distance-centre and distance normal.

For the centre projection, the geometry's centre point and the point cloud vertices are used as the projection direction onto the detector sphere. In contrast, the normal projection takes the normal direction of the point as the projection direction. Both projection types are described in Eq. (1).

$$\begin{aligned} centre &= \vec{C} + \vec{PC}; \\ normal &= \vec{PC} + \vec{t1} * \vec{n}; \end{aligned}$$

with

$$\begin{aligned} \vec{C} &= [0, 0, 0] \\ \vec{PC} &= PointCloudVertices \\ \vec{n} &= NormalsVertices \end{aligned}$$

```
Function RotatePartToLargestFace(faces, vertices):
    normals = GetFaceNormals(faces);
    (normalsAll,~,IC) = unique(normals);
    for n ∈ normalsAll do
        facesNormal = faces(n == IC);
        adjMatrix = AdjacencyMatrix(facesNormal);
        connGraph = ConnComponents(Graph(adjMatrix));
        forall connGraph with size(connGraph) > 2 do
            fSameN = GetFaces(connGraph);
            areaConn = sum(AreaFaces(vertices,fSameN));
        end
        maxAreaNormals = max(areaConn);
    end
    [faceMaxArea,locMaxArea] = max(maxAreaNormals);
    facesLargArea = fSameN(locMaxArea);
    VNew =
     Rotated3D(vertices,[0, 0, −1],normalsAll(locMaxArea));
     return VNew
  Function getLargestEdge(facesLargArea):
    adjEdgeMatrix = AdjaEdgeCostMat(vertices,
     outline(facesLargArea));
    connGraphEdge = ConnComponents(Graph(adjEdgeMatrix));
    forall connGraphEdge with size(connGraphEdge) > 2 do
        cycles = cyclebasis(connGraphEdge);
        polyin = polyshapeAdd(VerticesCycle);
    end
    Vpoly = sortboundaries(polyin);
    for i ∈ polyVertices do
        distance = dist(Vpoly,Vpoly(i + 1));
        direction = normalize(Vpoly(i + 1) − Vpoly(i));
    end
    [DistMax,locDistMax] = max(distance);
    DirEdgeMax = [direction(locDistMax, 1:2), 0];
    return DirEdgeMax
Function CheckXAlignment(vertices,faces):
    XFar = vertices(vertices(:, 1) ==max(vertices(:, 1)),3);
    XNear = vertices(vertices(:, 1) ==min(vertices(:, 1)),3);
    if dist(max(XFar) - min(XFar)) > dist(max(XNear) -
     min(XNear)) then
        Valgined=vertices∗rotZ(180);
    end
    IdxComp = vertices(:, 1) == min(vertices(:, 1));
    if max(vertices(:, 3))>max(vertices(IdxComp)) then
        IdxXZMax = vertices(:, 3), 2) ==max(vertices(:, 3));
        VXMaxZ = vertices(IdxXZmax, 1);
        DistMin = dist(min(vertices(:,1)),VXMaxZ);
        DistMax = dist(max(vertices(:,1)),VXMaxZ);
        if DistMin > DistMax then
            Valgined=vertices∗rotZ(180);
        end
    end
    if max(XFar) == max(XNear) then
        AreasMax = AreaFaces(faces(IdxMax));
        AreasMin = AreaFaces(faces(IdxMin));
        if sum(AreasMax) > sum(AreasMin) then
            Valgined=vertices∗rotZ(180);
        end
    end
    return Valgined
```

Algorithm 2: Support algorithms for the part alignment

$$
\begin{aligned}
\vec{t1} &= \frac{-b+\sqrt{b^2-4*a*c}}{2*a} \\
a &= \vec{n}\cdot\vec{n} \\
b &= 2*(PC_X*n_X - C_X*n_X + PC_Y*n_Y - \\
&\quad C_Y*n_Y + PC_Z*n_Z - C_Z*n_Z);
\end{aligned}
$$

$$
\begin{aligned}
c &= PC_X^2 + PC_Y^2 + PC_Z^2 - r^2 + C_X^2 + \\
&\quad C_Y^2 + C_Z^2 - 2*(PC_X*C_X + \\
&\quad PC_Y*C_Y + PC_Z*C_Z) \\
r &= \sqrt{2*max(|\vec{PC}|)^2 + 0.05}
\end{aligned} \tag{1}
$$

The shown projection vectors are then assigned to the specific pixels, according to their azimuth and inclination angle. With the first two methods, the points per pixel are counted. The last variant, the distance projection, uses the distance from the centre point to the point onto the detector sphere. Subsequently, the arithmetic mean per pixel is calculated. The different projection types (normal and centre) are shown in Fig. 6.

The converted matrices for each view are passed to a pre-trained Deep Learning network in the next step, to extract the feature vectors.

3.4. Descriptor generation and comparison

In this stage of the process, the generated matrices must be transferred into vectors through image feature extraction by pre-trained CNNs. The data set for the pre-trained CNN is ImageNet from [49], which consists of almost 1.3 million images in 1000 classes. The input matrix is processed normally through the network, using the calculated vector of the last convolutional layer, respectively the last layer before the fully connected layer. Consequently, for example, a pre-trained resnet50 has a feature vector with the dimension $1 \times 1 \times 2048$. In this way the trained feature extraction properties of the CNN are taken advantage of, without having to train a network with labelled data.

For each geometry and rotation, a vector is generated in the same process. Afterwards, the vector is trimmed down by dimension reduction to ensure that only the relevant features are included in the comparison. The methods for dimension reduction are manifold, for example the principal component analysis. Methods from the unsupervised learning domain are also feasible, for example Autoencoder. The prepared and transformed vectors are then analysed for similarity by finding the most similar vector using a k-nearest neighbour search. For this approach, a total of five different distance measures were examined, the euclidean, cityblock, Minkowski, correlation and cosine distance. More information about the calculation of the distance metrics can be found in [50–52].

4. Application

After the introduction of the new method, the process is now examined in depth. In order to understand the approach, the data set for testing the similarity search is presented at first, followed by the evaluation metrics for ranking the search results. On this basis, the different impact parameters for the process are shown and their effect on the result is analysed.

4.1. Benchmark data set

Because the presented method is applied in the product development domain, the benchmark data set is selected accordingly. The Engineering Shape Benchmark (ESB) data set from [48] is chosen as it consists primarily of mechanical engineering components. The data set includes a total of 866 mainly technical or standard components. The parts are sorted into three main categories depending on their general appearance: flat-thin wall components, rectangular-cubic prisms, or solids of revolution. Furthermore, the components are arranged into more specific categories, like brackets, L blocks or nuts. There are 45 classes in total, with a minimum of 4 components and a maximum of 58

Table 1
Collection of all classes of the ESB data set according to [48].

Flat-thin components	Nr.	Rectangular-cubic prism	Nr.	Solids of revolution	Nr.
Backdoors	7	Bearing blocks	7	90 degree elbows	41
Bracket like parts	18	Contoured surfaces	5	Bearing like parts	20
Clips	4	Handles	18	Bolt like parts	53
Contact switches	8	L Blocks	7	Container like parts	10
Curved housings	9	Long machine elements	15	Cylindrical parts	43
Miscellaneous	12	Machined blocks	9	Discs	51
Rectangular housings	14	Machined plates	49	Flange like parts	15
Slender thin plates	12	Miscellaneous	21	Gear like parts	36
Thin plates	23	Motor bodies	7	Intersecting pipes	9
–	–	Prismatic stock	36	Long pins	58
–	–	Rocker arms	10	Miscellaneous	33
–	–	Slender links	13	Non-90 degree elbows	8
–	–	Small machined blocks	12	Nuts	19
–	–	T shaped parts	15	Oil Pans	8
–	–	Thick plates	12	Posts	11
–	–	Thick slotted plates	20	Pulley like parts	12
–	–	U shaped parts	25	Round change at end	21
–	–	–	–	Simple pipes	16
–	–	–	–	Spoked wheels	15

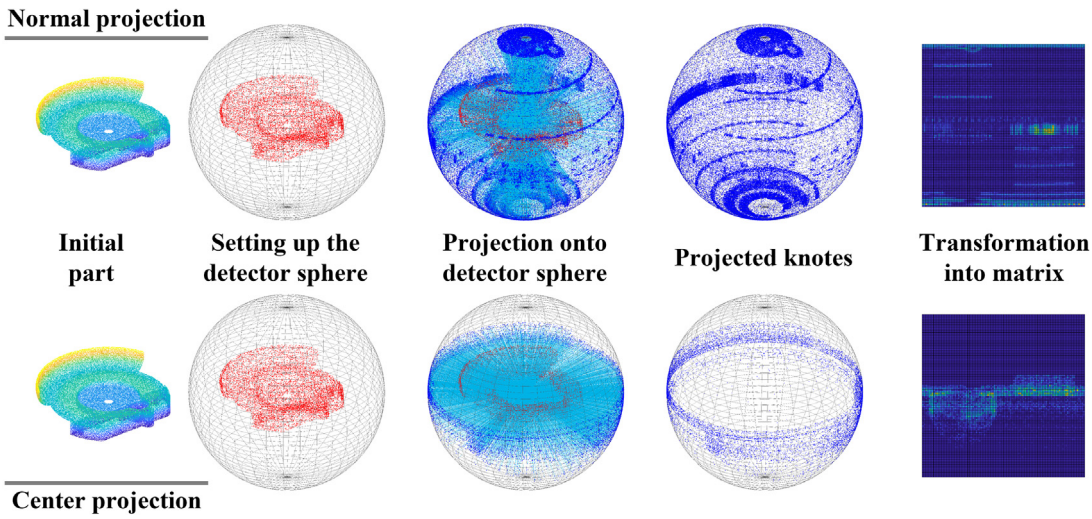


Fig. 6. Comparison of different projection approaches.

per class. Table 1 lists all classes including the main category and their number of components.

The parts are stored as meshes in the data set and saved in the obj and stl format. The components are not pre-aligned, instead they are collected in an unordered form. Fig. 7 displays randomly selected components from the ESB data set. This data collection was selected not only because of its specialization in mechanical components, but also because it has been available since 2005. As a result, many others methods have been examined using the ESB data set, which increases the significance of the evaluation of the obtained results in this new approach.

Because "similarity" can be determined differently from altered viewpoints, the definition of part classes for a data set can cause problems. In this regard, a study was conducted by [53] using crowdsourcing to investigate class separation for flat-thin components in the ESB data set. In this survey, the participants were anonymously asked to classify all components into groups on their own. The results revealed that the classes of the used data set correspond well with the division of the participants. While some components were assigned differently, the overall conclusion was quite positive. Therefore, this data set was selected to evaluate the new approach.

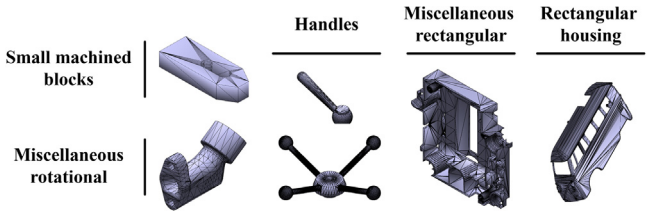


Fig. 7. Random selected components of the ESB data set.

4.2. Evaluation metrics

In addition to the data set, the analysis of the search results is a vital part of the study. A lot of different metrics have been defined for the evaluation of similarity results over the years, each with different aims. The frequently employed and relevant metrics for geometry search, according to [13,54], are presented in this section.

The first type of evaluation includes a total of three scores, which are generally applied at the same time: Nearest Neighbour (NN), First Tier (FT) and Second Tier (ST). The first value, Nearest Neighbour, describes the precision of the first and therefore most similar retrieval result. FT contains the recall results of the

best *C-1* results, where C is the number of relevant models for the retrieval task. The Second Tier result is calculated with the same principle, but for twice the number of results. In [55] the equations for precision and recall are stated and are adapted for the evaluation metrics.

$$Precision = \frac{|retrieved@R \cap relevant|}{|retrieved@R|} \quad (2)$$

for NN: $R = 1$

$$Recall = \frac{|retrieved@R \cap relevant|}{|relevant|} \quad (3)$$

for FT: $R = C - 1$

for ST: $R = 2 * (C - 1)$

In addition, the Discounted Cumulated Gain (DCG) is likewise used as a performance metric. This value weighs relevant results according to their position in the retrieval list. DCG therefore rates earlier results higher by using the normalized summed weighted value for the position of the relevant models. According to [13,54], DCG is calculated using the result vector G of the retrieval result R. $G = 1$ if R_i is a relevant result, $G = 0$ if it is not.

$$DCG_i \begin{cases} G_1 & \text{i=1} \\ DCG_{i-1+\frac{G_i}{lg_2 i}} & \text{otherwise} \end{cases} \quad (4)$$

Subsequently, the result is divided by the maximum possible DCG value and therefore normalized.

$$DCG = \frac{DCG_n}{1 + \sum_{j=2}^{C} \frac{1}{lg_2 j}} \quad (5)$$

Other evaluation metrics are R-Precision, Precision–Recall Curve (PR), or the Average Precision (AP), more information can be found in [13,54,56]. For all metrics listed in this section applies, the higher the score value, the better the retrieval result. For the verification of the calculated results, the code provided by [57] from the SHREC'14 data set was adapted.

4.3. Examination of influencing factors on the retrieval accuracy

After presenting the evaluation metrics, the method itself is examined and several influences on the detection accuracy identified. For convenience, these are examined in the chronological order of their application, starting with the part orientation, followed by the feature extraction, the projection type and the distance measurement and the dimension reduction.

4.3.1. Part alignment

The component alignment method presented in Section 3.2.1 is first investigated with a defined feature extraction network and otherwise unchanged parameters. The points per component are set to 50.000, the projection resolution is 256 × 256, the sum of all four projections is entered as the input for the network, the pre-trained network is a densenet201 [58], the Euclidean distance is used as distance metric and there is no dimension reduction of the results. The comparison is made with an unaligned and with a PCA-aligned data set. The four different evaluation metrics explained in the previous section are considered for the first comparison. Fig. 8 presents the comparisons in a graphical way. The bar chart on the left displays the four evaluation metrics in comparison; the outcomes for the own alignment are plotted in dark blue, for the PCA in grey and for the unaligned components in light blue.

The comparison of these three types of alignment clearly demonstrates that the own type of component alignment is an

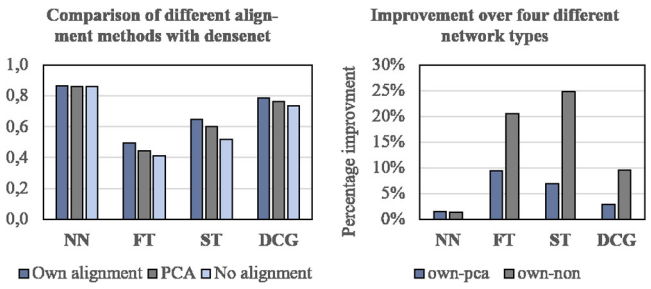


Fig. 8. Comparison of different part alignment methods.

improvement in the retrieval results. For the Nearest Neighbour the enhancement is minor, however for First and Second Tier the advancements are significant. Also, the comparison of the DCG values reveals that an improvement of the retrieval results was achieved. To avoid the possibility that this enhancement is a random result in combination with the specific network, the next step is to analyse the percentage improvement for four networks. All other parameters remain unchanged and the networks for this analysis are expanded with densenet201, resnet18, resnet50 [38] and shufflenet [59]. For the evaluation, the main value of the percentage deviation between the results of the own alignment and the PCA are calculated, as well as the value between the own alignment and without the alignment. The results are also depicted with bar charts on the right in Fig. 8, where the first comparison is represented in dark blue and the second in dark grey. This comparison reinforces the results from the first investigation and reveals that both, the own alignment and the PCA, achieve an increase compared to no alignment. This is quite reasonable, due to the rotation dependence of view-based methods. However, the comparison also shows an improvement of almost 10% over the PCA method.

Like the first study, the difference in the NN value is very small, but considering the remaining retrieval list, the difference becomes evident. In conclusion, this comparison states that an improvement in retrieval results is achieved by the new way of aligning the components. A visual comparison of the different matrices also confirms the effectiveness of the new part alignment process. For this purpose, two different components in the three alignment configurations and the corresponding matrices have been plotted in Fig. 9. Here, the uniform alignment significantly improves the structure of the matrices visually by showing the features of the geometry uniformly in the projection matrix.

4.3.2. Feature extraction

Following the examination of the component alignment, the next phase is to evaluate the feature extraction through pre-trained Deep Learning networks. The parameters of the first study were applied for the comparison, including the new alignment of the components. All tested networks are pre-trained on the ImageNet data set and the feature vector of the last layer before the first fully connected layer is extracted. The input size for all matrices are adapted to the specific input of the network which is normally around the size 256 × 256. A total of 12 different pre-trained networks were employed: densenet201, squeezenet [60], inceptionv3 [61], googlenet [62], resnet18, resnet50, resnet101, shufflenet, dark-net53 [63], nasnet mobile and nasnet large (both [64]). For a detailed evaluation, all four evaluation metrics are summarized in Table 2, with the best values marked in bold text.

This listing demonstrates that especially the shallower resnet networks perform very well for the Nearest Neighbour value, as the highest values for this evaluation criterion are achieved by

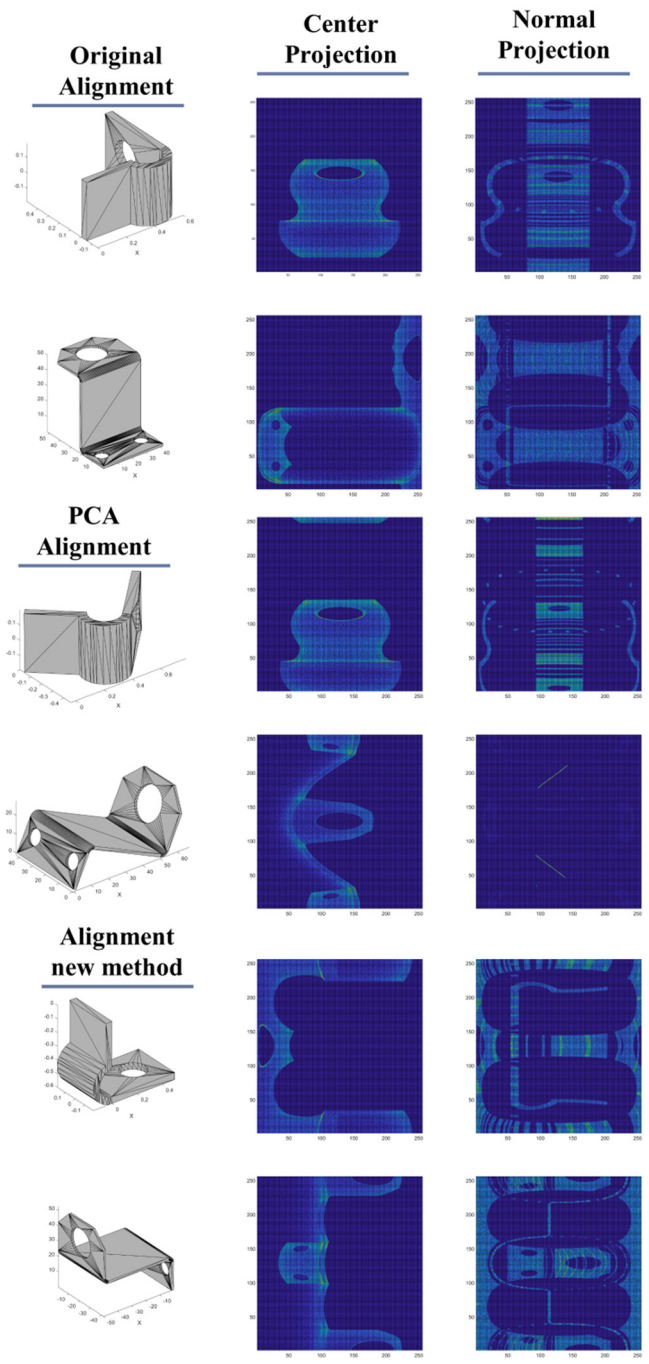


Fig. 9. Visual comparison of different part alignment methods and the corresponding projection matrices.

resnet18 and resnet50. For the combination of First/Second Tier and DCG scores, densenet provides the best results. Therefore, this network was selected for feature extraction and is applied for all following evaluations.

4.3.3. Projection method and distance metric

After examining the network architecture, the effects of the projection types and the distance metrics are analysed subsequently. Again, a fixed point cloud number of 50.000 and the new alignment method are selected for the examination. The results are plotted in a heatmap in Fig. 10.

The various combinations of the feature vector are listed in Table 3, with a total of five permutations examined: the sum of all matrices, centre projection only, normal projection only, all distance matrices, and all non-distance matrices.

Table 2
All results for different pre-trained networks with euclidean distance.

Network	NN	FT	ST	DCG
densenet	**0,865**	**0,493**	**0,646**	**0,786**
resnet50	0,868	0,492	0,643	0,787
resnet101	0,864	0,487	0,638	0,787
googlenet	0,861	0,483	0,641	0,783
resnet18	0,871	0,481	0,637	0,783
shufflenet	0,863	0,479	0,632	0,782
squezzenet	0,853	0,475	0,618	0,773
nasnet large	0,829	0,410	0,539	0,734
inceptionv3	0,820	0,402	0,516	0,721
nasnet mobile	0,816	0,358	0,458	0,691
darknet53	0,794	0,349	0,451	0,693

Table 3
Equations for all different feature vector combinations.

Sum	Centre all	Normal all	Distance all	Non distance all
$= c + n + dc + dn$	$= c + dc$	$= n + dn$	$= dc + dn$	$= c + n$

with c = centre matrix; n = normal matrix;
dc = distance centre matrix; dn = distance normal matrix.

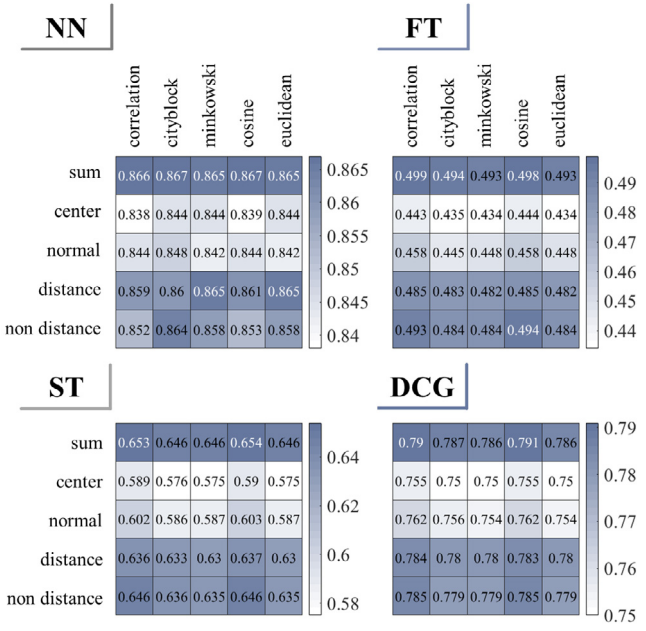


Fig. 10. Result-heat-map for the comparison of distance metrics and projection method.

In addition to the arrangement of the feature vector, the type of distance calculation is also investigated. Here five ways of calculation are consulted as stated in Section 3.4: correlation, cityblock, Minkowski, cosine and euclidean. The results are again evaluated with all four comparison metrics.

Nevertheless, the four different heat-maps illustrate how strong the arrangement of the vectors influences the retrieval results. It is also noticeable that across all four evaluation metrics a combination of centre and normal projection brings a significant improvement in retrieval results, regardless of whether it is the total sum, distance only, or non-distance combination. A further head-to-head comparison of the three configurations shows that the total sum performs better than the other two configurations, although the difference is less distinct, as it is with only centre or normal. When analysing the distance measures, it becomes

Table 4
Best results of the dimension reduction.

	NN	FT	ST	DCG
Baseline	0,867	0,499	0,654	0,791
Autoencoder (65)	0,874	0,527	0,684	0,807
PCA (70)	0,870	0,509	0,670	0,799

apparent that there is no major influence for the Nearest Neighbour but for the remaining other three values. In this regard, it is interesting to observe that the distance metrics correlation and cosine provide an improvement of the result values over all five feature vector combinations. All of these observations lead to the conclusion that the sum of all generated matrices in combination with the cosine distance metric produces the best results for the given input. This leads to the following result values:
NN: 0.867; FT: 0.499; ST: 0.654; DCG: 0.791.

4.3.4. Dimension reduction

In the last step of the analysis of variables affecting the retrieval accuracy, the dimension reduction is considered. The comparison is made between two methods, principal component analysis and an Autoencoder. Regardless of the improvement in search results, dimension reduction also enables a faster search process due to the reduced vector size for matching similarity.

Using the dimensions reduction methods, the feature vector of each transformed matrix is concentrated, resulting in a total of 24 reduced feature vectors per projection type. These vectors are then combined into one large vector. Generally, when decreasing dimensions, it is necessary to specify the compressed size. For this purpose, several experiments were carried out for each method and the best results per method are compared; the dimension value is listed in brackets behind the method name in Table 4. Different hidden layer sizes have been examined for the Autoencoder as well, with all other parameters remaining the same. The following parameters have been applied: Transfer Function: logsig; Sparsity Proportion: 0.05; L2 Weight Regularization: 0.0001 and Max Epochs: 700. For the PCA variant only the dimension value was adjusted.

The best retrieval results are collected in Table 4 and demonstrate that significant improvement across all evaluation metrics could be achieved through the application of dimensionality reduction methods. When applying the principal component analysis, enhancements of up to 2.5% could be achieved for the different metrics. Due to the simple and fast application of the PCA, this is a very suitable variant for an industrial application, because it increases the accuracy while also reducing the search time.

However, by applying unsupervised learning via an Autoencoder, the search results could be enhanced even further. The percentage of improvement is plotted in Fig. 11. Particularly First and Second Tier results achieved significant gains, with increases of up to 5.5%.

Overall, this evaluation reveals that an optimization of the results is achievable and, in some cases, even an acceleration of the process. Subsequently, the results are compared with the previous concept and current state of the art.

5. Comparison with other approaches

A first concept for searching similar components was presented by [43]. The approach uses centre projection with a subsequent comparison of the matrices for their similarity, applying different algorithms. For better comparability, the same parameters were employed for the matrix generation (50.000 points; resolution 256×256; same alignment) and the matrix comparison

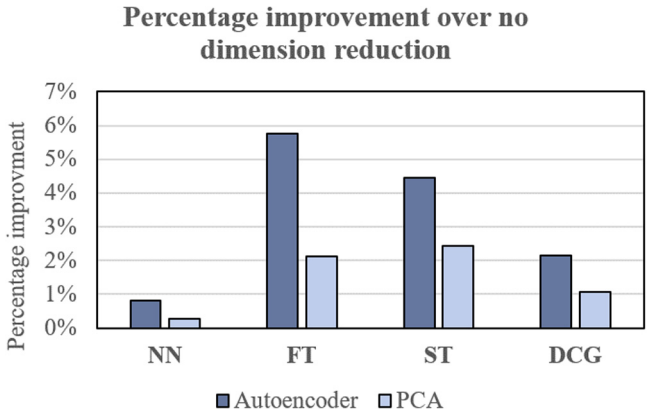


Fig. 11. Percentage improvement compared to no dimension reduction.

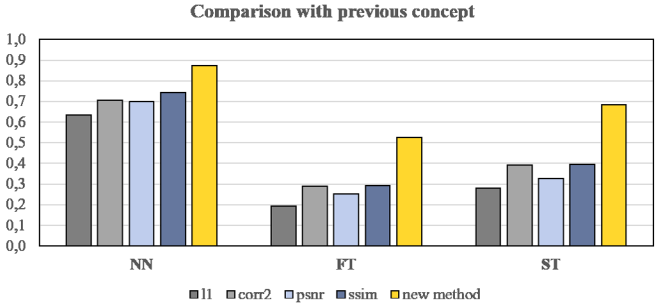


Fig. 12. Comparison of the new method with previous methods.

is extended by three additional algorithms. In the original version only the correlation coefficient (corr2) is considered, but in this study the structural similarity index for measuring image quality (SSIM), peak signal-to-noise ratio (PSNR) and the euclidean distance (l1) are also examined. Fig. 12 displays the results of the comparison.

The diagram shows that among the five methods, especially the correlation coefficient and the SSIM achieve good results, whereas the euclidean distance provides the worst results. Nevertheless, the experiment shows conclusive confirmation that the new method delivers significantly better outcomes. The evaluation of the Nearest Neighbour results is not quite obvious but especially in the First and Second Tier scores the advance is apparent. In some cases, an increase in the evaluation value of almost half is achieved.

This positive comparison provides the basis for the following analysis and the evaluation with other state-of-the-art methods.

5.1. Comparison with established methods

For the comparison with currently available methods, a literature search was performed, with the aim to collect all retrieval results for the ESB data set. The outcome of this investigation is shown in Table 5. The methods found are listed according to their function category, training type and the evaluation metrics. For the function and training type, the same categories have been used as in Section 2.1.3. In the table, the abbreviation NT stands for “no Training”, USL for “Unsupervised Learning” and LR for “Label required/Supervised Learning”. The input categories are indicated by the acronym G for “Geometric”, VB for “View-Based” and H for “Hybrid”. The list is also sorted in descending order by the value of the First Tier results. Nearest Neighbour (NN), First Tier (FT) and Second Tier (ST) were selected as evaluation

Table 5
List of all retrieval results using the ESB data set [29,34,65–89].

Acronyms	NN [%]	FT [%]	ST [%]	Input-category	Semantic	Source
Local and global features	88,50	64,02	49,21	G	NT	[65]
Spectral clustering	95,12	63,58	47,63	G	USL	[66]
TLC + I-Pair	88,20	56,80	72,00	VB	NT	[67]
PAN + HC	90,38	55,09	66,48	H	USL	[68]
HLO-SVM-OSS	88,70	53,70	71,20	G	LR	[69]
HSR-DE	90,20	52,80	69,70	H	NT	[70]
densenet AE	87,40	52,77	68,40	H	USL	–
PANORAMA SymPan+	88,00	52,60	67,80	H	NT	[71]
VSC WCO	84,00	51,00	58,00	H	LR	[72]
densenet PCA	87,00	50,90	67,00	H	NT	–
ROSy+	87,40	50,80	65,70	H	NT	[73]
DBNAA PANO	87,10	50,00	65,40	H	NT	[74]
PANORAMA + LRF	87,00	49,90	65,80	H	USL	[34]
densenet	86,72	49,85	65,44	H	NT	–
Dynamic Multi-Descriptor	–	49,49	63,25	H	USL	[75]
MFSD	87,50	49,40	65,77	G	USL	[76]
PANORAMA	86,50	49,40	64,10	H	NT	[34]
ShapeVocabulary	84,60	48,70	64,00	H	LR	[77]/[69]
Autoencoder	85,70	47,90	63,10	VB	USL	[78]
ZFDR	84,10	46,80	60,90	H	LR	[79]
3D Hybrid	82,90	46,50	60,50	H	NT	[80]
PH	83,30	46,40	61,80	G	NT	[81]/[69]
CSD	82,20	46,10	61,90	G	NT	[82]/[69]
DESIRE	82,30	41,70	55,00	H	NT	[83]/[34]
Single Depth Image	79,70	41,40	54,30	H	NT	[84]/[69]
D-IA	52,00	41,00	30,00	G	NT	[85]
LF	82,00	40,40	53,90	VB	NT	[29]/[34]
SH-GEDT	80,30	40,10	53,60	G	NT	[86]/[34]
PPA-DP	72,30	32,20	47,30	G	NT	[87]/[69]
D2	40,00	29,00	18,00	G	NT	[88]
AAD	68,70	26,50	44,60	G	NT	[89]/[69]

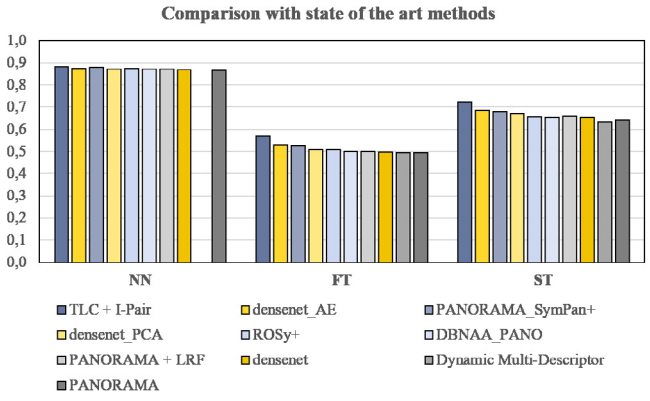


Fig. 13. Comparison with state of the art methods.

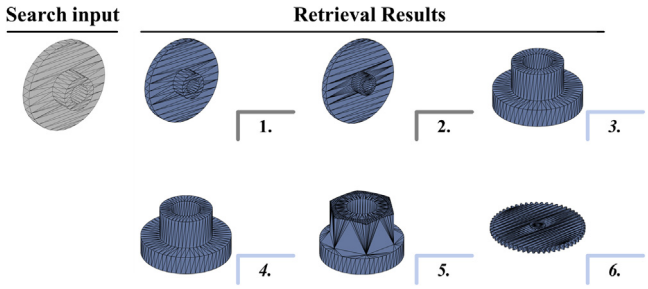


Fig. 14. Example of the retrieval results for one part in the class "Flange like Parts".

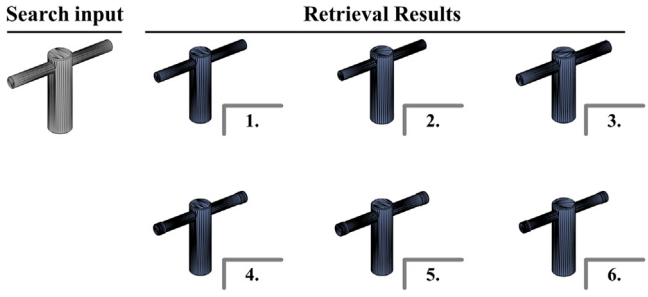


Fig. 15. Example of good retrieval results.

metrics for the retrieval results, since these were used in most of the papers. In some cases, methods with other evaluation metrics were presented but are not included in this comparison. Furthermore, results were partially determined with a reduced ESB data set. These results are marked in dark grey in the table and are not considered in further analysis. In addition, procedures that require a labelling of the data are also not taken into account, but are still listed in the table and highlighted in light grey. The three variants of the newly developed method are marked in bold letters.

In some cases, the result values were not determined directly in the original publication with the ESB data set but in a subsequent publication. This is marked in the table in the column "Source" by stating another reference after the first one, which represents the original source of the result value.

For a better comparison of the values, an excerpt from the table has been converted into a bar chart. Here, only the comparative values of the relevant methods are picked and the top ten procedures are displayed for clarity reasons, the diagram is shown in Fig. 13.

This chart illustrates that for the Nearest Neighbour value all algorithms have a relatively high level and their score varies within a small range across all methods. For the result values relevant to an application in mechanical engineering, which are FT and ST, the presented method performs favourably. Only one

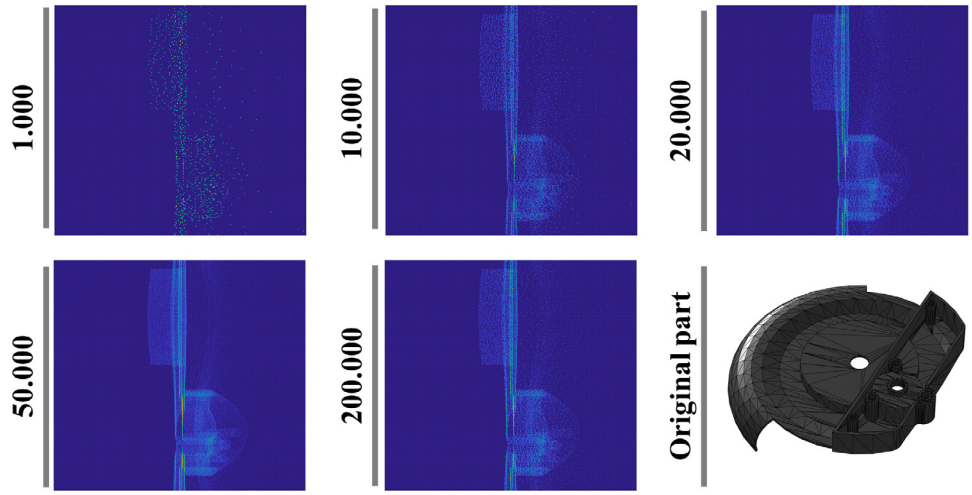


Fig. A.16. Examples of projected matrices for the same component, which are transformed through centre projection and various point cloud numbers.

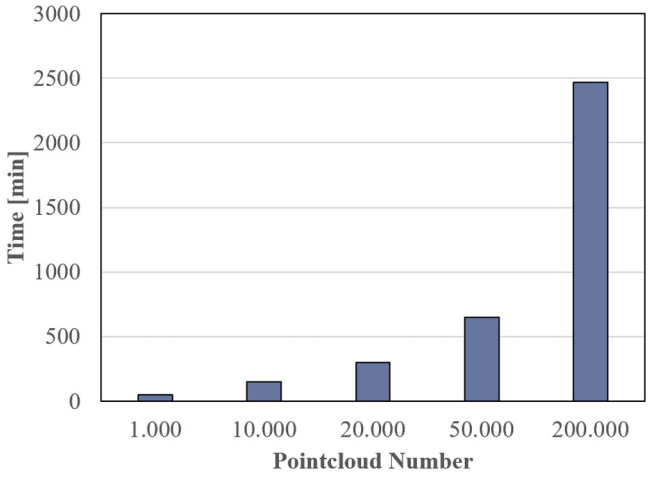


Fig. A.17. Computational time comparison for different point cloud numbers on the whole ESB data set.

of the 18 methods was able to achieve better values for these two evaluation metrics. It is also noticeable that the application of dimensionality reduction enabled another significant improvement compared to the existing methods.

In summary, the new technique achieves very good results, compared to previously developed methods. It is definitely competitive to the other approaches while still offering possibilities to further improve retrieval results.

6. Discussion

The presented results show that the new method performs competitively against the current state of the art algorithms, since only one method was able to achieve better retrieval results. The requirements of the product development on part similarity search were met, and the results were further improved by the use of dimensional reduction techniques. The comparison with the previously developed concept through the ESB benchmark data set is also definite, with the new method achieving significantly better scores.

However, the evaluation also reveals that the method poses challenges that need to be solved. One aspect is the dependence on the component alignment. The results demonstrate that an improvement of the similarity search is possible through a uniform alignment, which is inspired by the CAD design process. Hence, it was found that domain-specific knowledge has a positive impact on the overall search process. But the component alignment can also lead to potential errors or poor search results if the procedure does not function properly.

Furthermore, the analysis of performance scores indicates that some classes deliver poor results, as shown in Fig. 14 for the class "Flange Like Parts". The numbers in italics depict results of a false class, whereas numbers written normally represent the correct class.

In contrast, Fig. 15 represents a positive example. Here the first six results for a component from the class "T shaped parts" are listed, where all retrieved parts are from the correct category. This presentation of the specific results illustrates that not all components are found perfectly, but the comparatively high score demonstrates that the procedure can keep up with competing methods very successfully.

7. Conclusion and outlook

This paper presented and investigated a new method for searching similar components. In the first chapters, an extended literature review on similarity search of components was conducted. The new procedure was then described in detail and relevant parameters and their effect on the search result were investigated in depth. A special novelty is the alignment of the parts in the context of the mechanical engineering environment and the objective to get results without additional labelling of data.

Afterwards, the new method was evaluated and discussed using an established benchmark data set. The positive results provide the foundation for further work on this method since the possibilities for improvement are still large, due to the wide parameter space of the method. First, the alignment of the components could be improved, especially through the implementation of fitted bounding boxes to enhance the results. Also, other pre-trained models for feature vector extraction should be investigated and models that were not pre-trained on the image data set could also prove to be useful. In addition, other data sets should be tried on the method, for example the Mechanical Benchmark data set from [90]. This data set was not suitable for a first evaluation because here the available comparative results are fewer. Furthermore, it is also necessary to enhance the methods of dimension reduction continuously and to experiment with new approaches. Especially in the context of unsupervised learning, there are still many unexplored possibilities that need to be investigated in the future.

Table B.6
List of all classified methods for shape retrieval.

Year	Author	Title	Input category	Learning type	DOI
1989	Wong	Recognition and shape synthesis of 3D objects based on attributed hypergraphs	Graph	no Training	/10.1109/34.21797
1999	Johnson	Using spin images for efficient object recognition in cluttered 3d scenes	Geometric	no Training	/10.1109/34.765655
2000	Paquet	Description of shape information for 2-D and 3-D objects	Geometric	no Training	/10.1016/S0923-5965(00)00020-5
2000	Paquet	Nefertiti: A tool for 3-d shape databases management	Hybrid	no Training	/10.4271/1999-01-1891
2001	Vranic	Tools for 3d-object retrieval: Karhunen–Loeve transform and spherical harmonics	Geometric	no Training	/10.1109/MMSP.2001.962749
2001	Hilaga	Topology matching for fully automatic similarity estimation of 3D shapes	Graph	no Training	/10.1145/383259.383282
2002	Osada	Shape distributions	Geometric	no Training	/10.1145/571647.571648
2002	Ankerst	3D shape histograms for similarity search and classification in spatial databases	Geometric	no Training	/10.1007/3-540-48482-5_14
2002	Horn	Extended gaussian images	Geometric	no Training	/10.1109/PROC.1984.13073
2002	Edelsbrunner	Topological persistence and simplification	Graph	no Training	/10.1007/s00454-002-2885-2
2002	Frosini	Measuring shapes by size functions	Graph	no Training	/10.1117/12.57059
2002	Ip	Using shape distributions to compare solid models	Geometric	no Training	/10.1145/566282.566322
2002	Mahmoudi	3d models retrieval by using characteristic views	View-Based	no Training	/10.1109/ICPR.2002.1048337
2003	Obuchi	Shape-Similarity Search of 3D Models by using Enhanced Shape Functions	Geometric	no Training	/10.1109/tpcg.2003.1206936
2003	Kazhdan	Rotation invariant spherical harmonic representation of 3D shape descriptors	Geometric	no Training	/10.2312/SGP/SGP03/156-165
2003	Chen	On visual similarity based 3D model retrieval	View-Based	no Training	/10.1111/1467-8659.00669
2003	Kortgen	3D shape matching with 3D shape contexts	Geometric	no Training	–
2003	Sundar	Skeleton based shape matching and retrieval	Graph	no Training	/10.1109/SMI.2003.1199609
2003	Leibe	Analysing appearance and contour based methods for object categorization	Hybrid	no Training	/10.1109/CVPR.2003.1211497
2003	Vranic	An improvement of rotation invariant 3D-shape descriptor based on functions on concentric spheres	Geometric	no Training	/10.1109/ICIP.2003.1247355
2003	Tangelder	Polyhedral model retrieval using weighted point sets	Geometric	no Training	/10.1142/S021946780300097X
2003	Funkhouser	A search engine for 3-D models	Geometric	no Training	/10.1145/588272.588279
2004	Frome	Recognizing objects in range data using regional point descriptors	Geometric	no Training	/10.1007/978-3-540-24672-5_18
2004	Vranic	Model Retrieval	Hybrid	no Training	–
2004	Bustos	Using entropy impurity for improved 3D object similarity search	Hybrid	Label required	/10.1109/ICME.2004.1394465
2005	Vranic	Desire: a composite 3D-shape descriptor	Hybrid	no Training	/10.1109/ICME.2005.1521584
2005	Huang	3D shape context based gesture analysis integrated with tracking using omni video array	Geometric	no Training	/10.1109/CVPR.2005.382
2005	Leifman	Semantic-oriented 3D shape retrieval using relevance feedback	Hybrid	Label required	/10.1007/s00371-005-0341-z
2005	Hou	SVM-based semantic clustering and retrieval of a 3D model database	Hybrid	Label required	/10.1080/16864360.2005.10738363
2007	Kuo	3D model retrieval using principal plane analysis and dynamic programming	Geometric	no Training	/10.1016/j.patcog.2006.06.006
2007	Hou	Using Enhanced Shape Distributions to Compare CAD Models	Geometric	no Training	/10.1007/978-3-540-77255-2_41
2007	Rustamov	Laplace–Beltrami eigenfunctions for deformation invariant shape representation	Geometric	no Training	/10.2312/SGP/SGP07/225-233
2007	Jain	A spectral approach to shape-based retrieval of articulated 3D model	Geometric	no Training	/10.1016/j.cad.2007.02.009
2007	Chaouch	A new descriptor for 2D depth image indexing and 3D model retrieval	View-Based	no Training	/10.1109/ICIP.2007.4379599

(*continued on next page*)

Table B.6 (*continued*).

Year	Author	Title	Input category	Learning type	DOI
2007	Xu	3D shape retrieval integrated with classification information	Geometric	Label required	/10.1109/ICIG.2007.13
2007	Ohbuchi	Learning semantic categories for 3D model retrieval	Geometric	Label required	/10.1145/1290082.1290090
2007	Laga	A boosting approach to content-based 3D model retrieval	View-Based	Label required	/10.1145/1321261.1321301
2007	Biasotti	3D classification via structural prototypes	Graph	Label required	/10.1007/978-3-540-77051-0_16
2007	Shih	A new 3d model retrieval approach based on the elevation descriptor	View-Based	no Training	/10.1016/j.patcog.2006.04.034
2007	Ansary	A Bayesian 3-D search engine using adaptive views clustering	View-Based	Unsupervised Learning	/10.1109/TMM.2006.886359
2007	Papadakis	Efficient 3D shape matching and retrieval using a concrete radialized spherical projection representation	Geometric	no Training	/10.1016/j.patcog.2006.12.026
2008	Papadakis	3D Object Retrieval using an Efficient and Compact Hybrid Shape Descriptor	Hybrid	no Training	/10.2312/3DOR/3DOR08/009-016
2008	Ben-Chen	Characterizing shape using conformal factors	Geometric	no Training	/10.2312/3DOR/3DOR08/001-008
2008	Ohbuchi	Salient local visual features for shape-based 3D model retrieval	View-Based	no Training	/10.1109/SMI.2008.4547955
2008	Laga	Supervised learning of salient 2D views of 3D models	View-Based	Label required	/10.3756/artsci.7.124
2008	Laga	Supervised learning of similarity measures for content-based 3D model retrieval	Geometric	Label required	/10.1007/978-3-540-78159-2_20
2008	Yamamoto	SHREC'08 entry: semi-supervised learning for semantic 3D model retrieval	Geometric	Label required	/10.1109/SMI.2008.4547987
2008	Leng	A powerful relevance feedback mechanism for content-based 3-D model retrieval	Geometric	Label required	/10.1007/s11042-007-0188-6
2008	Ohbuchi	Accelerating Bag-of-Features SIFT Algorithm for 3D Model Retrieval	View-Based	Unsupervised Learning	–
2008	Ohbuchi	Ranking on Semantic Manifold for Semantic 3D Model Retrieval	View-Based	Label required	/10.1145/1460096.1460163
2008	Osada	SHREC'08 Entry: Local 2D Visual Features for CAD Model Retrieval	View-Based	Unsupervised Learning	/10.1109/SMI.2008.4547985
2008	Papadakis	Relevance Feedback in Content-based 3D Object Retrieval A Comparative Study	Geometric	Label required	/10.3722/cadaps.2008.753-763
2009	Tatsuma	Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval	Geometric	Unsupervised Learning	/10.1007/s00371-008-0304-2
2009	Pan	3D shape retrieval by Poisson histogram	Geometric	no Training	/10.1016/j.patrec.2011.01.003
2009	Jing	3D Mechanical Models Retrieval Based on Combined Histograms for Rapid Product Design	Geometric	no Training	/10.4028/www.scientific.net/AMM.16-19.65
2009	Furuya	Dense sampling and fast encoding for 3D model retrieval using bag-of-visual features	View-Based	Unsupervised Learning	/10.1145/1646396.1646430
2009	Akgül	3D Model Retrieval Using Probability Density-Based Shape Descriptors	Geometric	no Training	/10.1109/TPAMI.2009.25
2009	Sun	A concise and provably informative multi-scale signature based on heat diffusion	Hybrid	no Training	/10.1111/j.1467-8659.2009.01515.x
2009	Carlsson	Topology and data	Graph	no Training	/10.1090/S0273-0979-09-01249-X
2009	Mademlis	3d object retrieval using the 3d shape impact descriptor	Geometric	no Training	/10.1016/j.patcog.2009.04.024
2009	Li	Weighted subspace distance and its applications to object recognition and retrieval with image sets	View-Based	no Training	/10.1109/LSP.2008.2010819
2009	Ohbuchi	Scale-Weighted Dense Bag of Visual Features for 3D Model Retrieval from a Partial View 3D Model	View-Based	Unsupervised Learning	/10.1109/ICCVW.2009.5457716
2010	Daras	A 3D shape retrieval framework supporting multimodal queries	View-Based	no Training	/10.1007/s11263-009-0277-2
2010	Lian	Visual similarity based 3D shape retrieval using Bag-of-Features	View-Based	Unsupervised Learning	/10.1109/SMI.2010.20
2010	Jégou	Aggregating local descriptors into a compact image representation	View-Based	no Training	/10.1109/CVPR.2010.5540039

(*continued on next page*)

Table B.6 (*continued*).

Year	Author	Title	Input category	Learning type	DOI
2010	Papadakis	PANORAMA: a 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval	Hybrid	Unsupervised Learning	/10.1007/s11263-009-0281-6
2010	Laga	Semantics-driven approach for automatic selection of best views of 3D shapes	View-Based	Label required	/10.2312/3DOR/3DOR10/015-022
2010	Wessel	Learning the compositional structure of man-made objects for 3D shape retrieval	Hybrid	Label required	/10.2312/3DOR/3DOR10/039-046
2010	Ohbuchi	Squeezing bag-of-features for scalable and semantic 3D model retrieval	View-Based	Label required	/10.1109/CBMI.2010.5529890
2010	Makadia	Spherical correlation of visual representations for 3d model retrieval	View-Based	no Training	/10.1007/s11263-009-0280-7
2010	Gao	Representative Views Re-Ranking for 3D Model Retrieval with Multi-Bipartite Graph Reinforcement Model	View-Based	Unsupervised Learning	/10.1145/1873951.1874120
2010	Gao	3D Model Comparison using Spatial Structure Circular Descriptor	Hybrid	no Training	/10.1016/j.patcog.2009.07.012
2010	Ohbuchi	Distance Metric Learning and Feature Combination for Shape-Based 3D Model Retrieval	View-Based	Unsupervised Learning	/10.1145/1877808.1877822
2010	Gao	3D Object Retrieval with Bag-of-Region-Words	View-Based	Unsupervised Learning	/10.1145/1873951.1874122
2011	Sfikas	Rosy+: 3d object pose normalization based on pca and reflective object symmetry with application in 3d object retrieval	Hybrid	no Training	/10.1007/s11263-010-0395-x
2011	Aubry	The wave kernel signature: A quantum mechanical approach to shape analysis.	Hybrid	no Training	/10.1109/ICCVW.2011.6130444
2011	Pan	3D shape retrieval by Poisson histogram	Geometric	no Training	/10.1016/j.patrec.2011.01.003
2011	Li	View context based 2D sketch-3D model alignment	View-Based	no Training	/10.1007/s11263-009-0277-2
2011	Axenopoulos	3D model retrieval using accurate pose estimation and view-based similarity	View-Based	no Training	/10.1145/1991996.1992037
2011	Chen	A new framework for composing vectorial semantic labels in 3D model retrieval	Hybrid	Unsupervised Learning	/10.1117/12.910564
2011	Gao	Efficient 3D Object Retrieval with Query View Selection	View-Based	Label required	/10.1109/TMM.2011.2160619
2011	Bronstein	Shape google: Geometric words and expressions for invariant shape retrieval	Geometric	Unsupervised Learning	/10.1145/1899404.1899405
2012	Alcantarilla	KAZE Features	View-Based	no Training	/10.1007/978-3-642-33783-3_16
2012	Gao	K-Partite Graph Reinforcement and Its Application in Multimedia Information Retrieval	View-Based	Unsupervised Learning	/10.1016/j.ins.2012.01.003
2012	Wen	View-based 3D Object Retrieval by Bipartite Graph Matching	View-Based	Unsupervised Learning	/10.1145/2393347.2396341
2012	Gao	Camera Constraint-Free View-Based 3D Object Retrieval	View-Based	Label required	/10.1109/TIP.2011.2170081
2012	Gao	3D Object Retrieval and Recognition with Hypergraph Analysis	View-Based	Label required	/10.1109/TIP.2012.2199502
2012	Endoh	Efficient manifold learning for 3D model retrieval by using clustering-based training sample reduction	Geometric	Unsupervised Learning	/10.1109/ICASSP.2012.6288385
2012	Kawamura	Local Geometrical Feature with Positional Context for Shape-based 3D Model Retrieval	Geometric	Unsupervised Learning	–
2013	Li	Retrieving 3D Model Using Compound-Eye Visual Representation	Hybrid	Unsupervised Learning	/10.1109/CADGraphics.2013.30
2013	Li	3D model retrieval using hybrid features and class information	Hybrid	Label required	/10.1007/s11042-011-0873-3
2013	Chowdhuryt	A New Shape Descriptor based on Local and Global Feature for 3D ShapeRetrieval	Geometric	no Training	/10.11485/itetr.37.56.0_19
2013	Bae	Content-based 3D model retrieval using a single depth image from a low-cost 3D camera	Hybrid	no Training	/10.1007/s00371-013-0819-z
2013	Aono	3D shape retrieval focused on holes and surface roughness	Hybrid	no Training	/10.1109/APSIPA.2013.6694132
2013	Sipiran	Data-aware 3D partitioning for generic shape retrieval	Geometric	no Training	/10.1016/j.cag.2013.04.002

(*continued on next page*)

Table B.6 (*continued*).

Year	Author	Title	Input category	Learning type	DOI
2013	Li	Symmetry discovery and retrieval of nonrigid 3D shapes using geodesic skeleton paths	Graph	no Training	/10.1007/s11042-013-1417-9
2013	Barra	3D shape retrieval using kernels on extended Reeb graphs	Graph	no Training	/10.1016/j.patcog.2013.03.019
2013	Biasotti	PHOG: photometric and geometric functions for textured shape retrieval	Graph	no Training	/10.1111/cgf.12168
2013	Bonaventura	3D shape retrieval using viewpoint information-theoretic measures	View-Based	no Training	/10.1002/cav.1566
2013	Zhang	Retrieving 3D model using compound-eye visual representation	View-Based	no Training	/10.1109/CADGraphics.2013.30
2013	Sfikas	3D object retrieval via range image queries in a bag-of-visual-words context	View-Based	Unsupervised Learning	/10.1007/s00371-013-0876-3
2013	Li	Combining topological and view-based features for 3D model retrieval	Hybrid	no Training	/10.1007/s11042-012-1000-9
2014	Zou	A novel 3D model retrieval approach using combined shape distribution	Geometric	no Training	/10.1007/s11042-012-1130-0
2014	Zhu	Deep Learning Representation using Autoencoder for 3D Shape Retrieval	View-Based	Unsupervised Learning	/10.1109/spac.2014.6982699
2014	Sfikas	Pose normalization of 3D models via reflective symmetry on panoramic views	Hybrid	no Training	/10.1007/s00371-014-0935-4
2014	Furuya	Fusing Multiple Features for Shape-based 3D Model Retrieval	View-Based	Unsupervised Learning	/10.5244/C.28.16
2014	Bai	Shape vocabulary: A robust and efficient shape representation for shape matching	Hybrid	Label required	/10.1109/TIP.2014.2336542
2014	Litman	Supervised learning of bag-of-features shape descriptors using sparse coding	Hybrid	Label required	/10.1111/cgf.12438
2014	Li	Persistence-based structural recognition	Graph	no Training	/10.1109/CVPR.2014.257
2014	Ding	Sphere image for 3-D model retrieval	View-Based	no Training	/10.1109/TMM.2014.2314073
2014	Liu	Sparse patch coding for 3D model retrieval	Hybrid	no Training	/10.1007/978-3-319-04117-9_11
2014	Tabia	Covariance descriptors for 3D shape matching and retrieval	Hybrid	no Training	/10.1109/CVPR.2014.533
2015	Mehrdad	3D object retrieval based on histogram of local orientation using one-shot score support vector machine	Geometric	Label required	/10.1007/s11704-015-4291-y
2015	Chen	3D CAD model retrieval based on the combination of features	Hybrid	no Training	/10.1007/s11042-013-1850-9
2015	Bai	3D Shape Matching via Two Layer Coding	View-Based	no Training	/10.1109/TPAMI.2015.2424863
2015	Wu	3D ShapeNets: A Deep Representation for Volumetric Shapes	Geometric	Label required	/10.1109/CVPR.2015.7298801
2015	Maturana	VoxNet: A 3D convolutional neural network for real-time object recognition	Geometric	Label required	/10.1109/IROS.2015.7353481
2015	Su	Multi-view convolutional neural networks for 3D shape recognition (MVCNN)	View-Based	Label required	/10.48550/arXiv.1505.00880
2015	Fang	3D deep shape descriptor	Hybrid	Unsupervised Learning	/10.1109/CVPR.2015.7298845
2015	Masci	Geodesic convolutional neural networks on Riemannian manifolds	Geometric	Label required	/10.48550/arXiv.1501.06297
2015	Boscaini	Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks	Geometric	Label required	/10.1111/cgf.12693
2015	Shi	DeepPano: Deep panoramic representation for 3-D shape recognition	Geometric	Label required	/10.1109/LSP.2015.2480802
2015	Furuya	Diffusion-on-Manifold Aggregation of Local Features for Shape-based 3D Model Retrieval	Graph	Unsupervised Learning	/10.1145/2671188.2749380
2016	Furuya	Accurate aggregation of local features by using K-sparse autoencoder for 3D model retrieval	View-Based	Unsupervised Learning	/10.1145/2911996.2912054
2016	Han	Unsupervised 3D local feature learning by circle convolutional restricted Boltzmann machine	Geometric	Unsupervised Learning	10.1109/TIP.2016.2605920
2016	Qi	Volumetric and multi-view CNNs for object classification on 3D data	Hybrid	Label required	/10.48550/arXiv.1604.03265
2016	Garcia-Garcia	PointNet: A 3D convolutional neural network for real-time object class recognition	Geometric	Label required	/10.1109/IJCNN.2016.7727386

(*continued on next page*)

Table B.6 (*continued*).

Year	Author	Title	Input category	Learning type	DOI
2016	Bai	GIFT: A real-time and scalable 3D shape search engine	View-Based	Label required	/10.48550/arXiv.1604.01879
2016	Boscaini	Anisotropic diffusion descriptors	Geometric	Label required	/10.1111/cgf.12844
2016	Brock	Generative and Discriminative Voxel Modelling with Convolutional Neural Networks	Geometric	Label required	/10.48550/arXiv.1608.04236
2016	Guo	Multi-View 3D Object Retrieval With Deep Embedding Network	View-Based	Label required	10.1109/TIP.2016.2609814
2017	Zhuang	A Novel 3D CAD Model Retrieval Method Based on Vertices Classification and Weights Combination Optimization	Hybrid	Label required	/10.1155/2017/6049750
2017	Shih	Three-Dimensional Model Retrieval Using Dynamic Multi-Descriptor Fusion	Hybrid	Unsupervised Learning	/10.11989/JEST.1674-862X.6071527
2017	Getto	Unsupervised 3D object retrieval with parameter-free hierarchical clustering	Hybrid	Unsupervised Learning	/10.1145/3095140.3095147
2017	Sedaghat	Orientation-boosted Voxel Nets for 3D Object Recognition	Geometric	Label required	/10.48550/arXiv.1604.03351
2017	Klokov	Escape From Cells: Deep Kd-Networks for the Recognition of 3D Point Cloud Models	Geometric	Label required	/10.1109/ICCV.2017.99
2017	Xie	Progressive shape-distribution-encoder for learning 3D shape representation	Geometric	Unsupervised Learning	10.1109/TIP.2017.2651408
2017	Xie	DeepShape: Deep-learned shape descriptor for 3D shape retrieval	Geometric	Unsupervised Learning	/10.1109/TPAMI.2016.2596722
2017	Han	Mesh convolutional restricted Boltzmann machines for unsupervised learning of features with structure preservation on 3-D meshes	Geometric	Unsupervised Learning	/10.1109/TNNLS.2016.2582532
2017	Han	Unsupervised learning of 3-D local features from raw voxels based on a novel permutation voxelization strategy	Geometric	Unsupervised Learning	/10.1109/TCYB.2017.2778764
2018	You	PVNet: A Joint Convolutional Network of Point Cloud and Multi-View for 3D Shape Recognition	View-Based	Label required	/10.1145/3240508.3240702
2018	Su	Hierarchical Graph Structure Learning for Multi-View 3D Model Retrieval.	Graph	Unsupervised Learning	/10.24963/ijcai.2018/127
2018	Ma	Multi-Dimensional Network Embedding with Hierarchical Structure	Graph	Unsupervised Learning	/10.1145/3159652.3159680
2018	Charles	PointNet: Deep learning on point sets for 3D classification and segmentation	Geometric	Label required	/10.1109/CVPR.2017.16
2018	Dominguez	General-purpose deep point cloud feature extractor	Geometric	Label required	/10.1109/WACV.2018.00218
2018	Feng	MeshNet: Mesh neural network for 3D shape representation	Geometric	Label required	/10.1609/aaai.v33i01.33018279
2018	Li	SO-Net: Self-organizing network for point cloud analysis	Geometric	Label required	/10.48550/arXiv.1803.04249
2018	Yang	FoldingNet: Point cloud auto-encoder via deep grid deformation	Geometric	Label required	/10.1109/CVPR.2018.00029
2018	Xu	SpiderCNN: Deep Learning on Point Sets with Parameterized Convolutional Filters	Geometric	Label required	/10.48550/arXiv.1803.11527
2018	Li	PointCNN: Convolution On X-Transformed Points	Geometric	Label required	/10.48550/arXiv.1801.07791
2018	He	Triplet-Centre Loss for Multi-View 3D Object Retrieval	View-Based	Label required	/10.48550/arXiv.1803.06189
2018	Lee	Cross-domain image-based 3d shape retrieval by view sequence learning	View-Based	Unsupervised Learning	/10.1109/3DV.2018.00038
2018	Xu	Emphasizing 3d properties in recurrent multi-view aggregation for 3d shape retrieval	View-Based	Label required	–
2018	Lin	Learning a disentangled embedding for monocular 3d shape retrieval and pose estimation	Hybrid	Label required	/10.48550/arXiv.1812.09899
2018	Li	Scale-invariant wave kernel signature for non-rigid 3D shape retrieval	Geometric	Unsupervised Learning	/10.1109/BigComp.2018.00072
2018	Luciano	Geodesic-based 3D Shape Retrieval Using Sparse Autoencoders	Geometric	Unsupervised Learning	/10.2312/3dor.20181049
2018	Benjelloun	3D shape retrieval basing on representatives of classes	View-Based	Label required	/10.48550/arXiv.1810.09008

(*continued on next page*)

Table B.6 (*continued*).

Year	Author	Title	Input category	Learning type	DOI
2018	Benjelloun	Content-based 3D shape retrieval using deep learning approach	View-Based	Label required	/10.1145/3230905.3230957
2018	Shao	3D Shape Retrieval using Volumetric and Image CNNs: A Meta-Algorithmic Approach	View-Based	Label required	/10.2352/ISSN.2470-1173.2018.10.IMAWM-419
2018	Rucco	A methodology for part classification with supervised machine learning	Hybrid	Label required	/10.1017/S0890060418000197
2019	Liu	Relation-Shape Convolutional Neural Network for Point Cloud Analysis	Geometric	Label required	/10.1109/CVPR.2019.00910
2019	Gao	Cognitive-inspired class-statistic matching with triple-constrain for camera free 3D object retrieval	View-Based	Label required	/10.1016/j.future.2018.12.039
2019	Jiang	MLVCNN: Multi-Loop-View Convolutional Neural Network for 3D Shape Retrieval	View-Based	Label required	/10.1609/aaai.v33i01.33018513
2019	Ouyang	Learning Cross-Domain Representation with Multi-Graph Neural Network	Graph	Unsupervised Learning	/10.48550/arXiv.1905.10095
2019	Hanocka	MeshCNN: A network with an edge	Geometric	Label required	/10.1145/3306346.3322959
2019	Liu	Point-voxel CNN for efficient 3D deep learning	Geometric	Label required	/10.48550/arXiv.1907.03739
2019	Cheraghian	3DCapsule: Extending the capsule architecture to classify 3D point clouds	Geometric	Label required	/10.1109/WACV.2019.00132
2019	Bold	3D point cloud retrieval with bidirectional feature match	Geometric	no Training	/10.1109/ACCESS.2019.2952157
2019	Han	SeqViews2SeqLabels: Learning 3D Global Features via Aggregating Sequential Views by RNN With Attention	View-Based	Label required	/10.1109/tip.2018.2868426
2019	You	Pvrnet: Point-view relation neural network for 3D shape recognition	View-Based	Label required	/10.1609/aaai.v33i01.33019119
2019	Han	3D2SeqViews: Aggregating Sequential Views for 3D Global Feature Learning by CNN With Hierarchical Attention Aggregation	Geometric	Label required	/10.1109/tip.2019.2904460
2019	He	View n-gram network for 3D object retrieval	View-Based	no Training	/10.48550/arXiv.1908.01958
2019	Xu	Enhancing 2D representation via adjacent views for 3D shape retrieval	View-Based	Label required	/10.1109/ICCV.2019.00383
2019	Zhou	Web3D learning framework for 3D shape retrieval based on hybrid convolutional neural networks	View-Based	Label required	/10.26599/TST.2018.9010113
2019	Zhu	Learning domain-invariant feature for robust depth-image-based 3D shape retrieval	View-Based	Label required	/10.1016/j.patrec.2017.09.041
2019	Li	Rethinking loss design for large-scale 3D shape retrieval	View-Based	Label required	/10.48550/arXiv.1906.00546
2019	Wang	A dimensional reduction guiding deep learning architecture for 3D shape retrieval	Geometric	Label required	/10.1016/j.cag.2019.04.002
2019	Wang	Multi-feature distance metric learning for non-rigid 3D shape retrieval	Geometric	Label required	/10.1007/s11042-019-7670-9
2019	Zhu	Training convolutional neural network from multi-domain contour images for 3D shape retrieval	View-Based	Label required	/10.1016/j.patrec.2017.08.028
2019	Xu	Learning discriminative and generative shape embeddings for three-dimensional shape retrieval	View-Based	Label required	/10.1109/TMM.2019.2957933
2019	Li	Non-rigid 3D shape retrieval based on multi-view metric learning	Geometric	Label required	/10.48550/arXiv.1904.00765
2019	Lu	3D Shape Retrieval through Multilayer RBF Neural Network	Geometric	Unsupervised Learning	/10.1109/ICIP.2019.8803384
2019	Zhu	Skeleton Tree based Non-rigid 3D Shape Retrieval	Graph	no Training	/10.1145/3356422.3356436
2019	Zhang	Anisotropic Laplace–Beltrami Operators for Non-Rigid 3D Shape Retrieval	Geometric	Label required	/10.1109/ITME.2019.00141
2020	Gao	Multiple Discrimination and Pairwise CNN for view-based 3D object retrieval	View-Based	Label required	/10.1016/j.neunet.2020.02.017
2020	Nie	Multi-graph convolutional network for unsupervised 3D shape retrieval	View-Based	Unsupervised Learning	/10.1145/3394171.3413987
2020	Li	MPAN: Multi-Part Attention Network for Point Cloud Based 3D Shape Retrieval	Geometric	Label required	/10.1109/ACCESS.2020.3018696

(*continued on next page*)

Table B.6 (*continued*).

Year	Author	Title	Input category	Learning type	DOI
2020	Fu	RISA-Net: Rotation-invariant structure-aware network for fine-grained 3D shape retrieval	Geometric	Unsupervised Learning	/10.48550/arXiv.2010.00973
2020	Afolabi	Extending DeepSDF for automatic 3D shape retrieval and similarity transform estimation	View-Based	Label required	/10.48550/arXiv.2004.09048
2020	Li	Gram Regularization for Multi-view 3D Shape Retrieval	View-Based	Label required	/10.48550/arXiv.2011.07733
2021	Wang	Centre-push loss for joint view-based 3D model classification and retrieval feature learning	View-Based	Label required	/10.1007/s11760-021-01923-4
2021	Bai	Multi-Scale Representation Learning on Hypergraph for 3D Shape Retrieval and Recognition	Graph	Label required	10.1109/TIP.2021.3082765
2021	Uy	Joint learning of 3d shape retrieval and deformation	Hybrid	Label required	/10.48550/arXiv.2101.07889
2021	Lin	Single Image 3D Shape Retrieval via Cross-Modal Instance and Category Contrastive Learning	View-Based	Unsupervised Learning	/10.1109/ICCV48922.2021.01121
2021	Luo	Multi-Modality Learning for Non-Rigid 3D Shape Retrieval via Structured Sparsity Regularizations	Geometric	Label required	/10.1109/JSEN.2021.3094122
2021	Nie	CLN: Cross-domain Learning Network for 2D Image-based 3D Shape Retrieval	View-Based	Label required	/10.1109/TCSVT.2021.3070969
2021	Jin	PREMA: Part-based REcurrent Multi-view Aggregation Network for 3D Shape Retrieval	View-Based	Label required	/10.48550/arXiv.2111.04945
2021	Kim	A method of generating depth images for view-based shape retrieval of 3D CAD models from partial point clouds	View-Based	Unsupervised Learning	/10.1007/s11042-020-10283-z
2021	Shen	Graph-Based Shape Analysis for Heterogeneous Geometric Data sets: Similarity, Retrieval and Substructure Matching	Graph	no Training	/10.1016/j.cad.2021.103125
2022	Agapaki	Geometric Digital Twinning of Industrial Facilities: Retrieval of Industrial Shapes	Hybrid	Label required	/10.48550/arXiv.2202.04834

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

The authors would like to thank the NVIDIA Corporation and the academic GPU Grant Program for the donation of a Tesla GPU.

The authors thank the German Research Foundation for funding this research under grant number WA 2913/47-1.

Appendix A. Comparison of point cloud number

See Figs. A.16 and A.17.

Appendix B. Literature review

See Table B.6.

References

[1] Pakkanen J, Huhtala P, Juuti T, Lehtonen T. Achieving benefits with design reuse in manufacturing industry. Proc CIRP 2016;50:8–13. http://dx.doi.org/10.1016/j.procir.2016.04.173, URL https://www.sciencedirect.com/science/article/pii/S2212827116304139, 26th CIRP Design Conference.

[2] Stenholm D, Corin Stig D, Ivansen L, Bergsjö D. A framework of practices supporting the reuse of technological knowledge. Environ Syst Decis 2019;39(2):128–45.

[3] Cardone A, Gupta SK, Karnik M. A survey of shape similarity assessment algorithms for product design and manufacturing applications. J Comput Inf Sci Eng 2003;3(2):109–18.

[4] Iyer N, Jayanti S, Lou K, Kalyanaraman Y, Ramani K. Shape-based searching for product lifecycle applications. Comput Aided Des 2005;37(13):1435–46.

[5] Iyer N, Jayanti S, Lou K, Kalyanaraman Y, Ramani K. Three-dimensional shape searching: state-of-the-art review and future trends. Comput Aided Des 2005;37(5):509–30.

[6] Chaudhari AM, Bilionis I, Panchal JH. Similarity in engineering design: A knowledge-based approach. In: International design engineering technical conferences and computers and information in engineering conference, Vol. 59278. American Society of Mechanical Engineers; 2019, V007T06A045.

[7] Dudenredaktion (Hrsg.). (o.j.), aehnlich. Duden online. 2022, https://www.duden.de/rechtschreibung/aehnlich/ Accessed: 2022-03-01.

[8] Vajna S, Weber C, Zeman K, Hehenberger P, Gerhard D, Wartzack S. CAx für ingenieure: Eine praxisbezogene einführung. Springer-Verlag; 2018.

[9] Loncaric S. A survey of shape analysis techniques. Pattern Recognit 1998;31(8):983–1001.

[10] Bespalov D, Ip CY, Regli WC, Shaffer J. Benchmarking CAD search techniques. In: Proceedings of the 2005 ACM symposium on solid and physical modeling, 2005, p. 275–86.

[11] McWherter D, Peabody M, Regli WC, Shokoufandeh A. Transformation invariant shape similarity comparison of solid models. In: International design engineering technical conferences and computers and information in engineering conference, Vol. 80241. American Society of Mechanical Engineers; 2001, p. 303–12.

[12] Cicirello VA, Regli WC. Resolving non-uniqueness in design feature histories. In: Proceedings of the fifth ACM symposium on solid modeling and applications, 1999, p. 76–84.

[13] Li B, Lu Y, Li C, Godil A, Schreck T, Aono M, Burtscher M, Chen Q, Chowdhury NK, Fang B, et al. A comparison of 3D shape retrieval methods based on a large-scale benchmark supporting multimodal queries. Comput Vis Image Underst 2015;131:1–27.

[14] Li B, Lu Y, Godil A, Schreck T, Bustos B, Ferreira A, Furuya T, Fonseca MJ, Johan H, Matsuda T, et al. A comparison of methods for sketch-based 3D shape retrieval. Comput Vis Image Underst 2014;119:57–80.

[15] Bustos B, Keim DA, Saupe D, Schreck T, Vranić DV. Feature-based similarity search in 3D object databases. ACM Comput Surv 2005;37(4):345–87.

[16] Madelmis A, Daras P, Tzovaras D, Strintzis MG. Three dimensional content-based search and retrieval of CAD objects. In: Encyclopedia of multimedia. Boston, MA: Springer US; 2008, p. 853–9.

[17] Qin Z, Jia J, Qin J. Content based 3D model retrieval: A survey. In: 2008 international workshop on content-based multimedia indexing. IEEE; 2008, p. 249–56.
[18] 3Dsémantix. 2022, https://www.3dpartfinder.com/ Accessed: 2022-03-30.
[19] iSEEK corporation. 2022, https://www.iseekcorp.com/products Accessed: 2022-03-30.
[20] Shape search. 2022, https://www.plm.automation.siemens.com/global/de/products/collaboration/shape-search.html Accessed: 2022-03-30.
[21] Tangelder JW, Veltkamp RC. A survey of content based 3D shape retrieval methods. Multimedia Tools Appl 2008;39(3):441–71.
[22] Zehtaban L, Roller D. Beyond similarity comparison: Intelligent data retrieval for CAD/CAM designs. Comput Aided Des Appl 2013;10(5):789–802.
[23] Lupinetti K, Pernot J-P, Monti M, Giannini F. Content-based CAD assembly model retrieval: Survey and future challenges. Comput Aided Des 2019;113:62–81.
[24] Ankerst M, Kastenmüller G, Kriegel H-P, Seidl T. 3D shape histograms for similarity search and classification in spatial databases. In: Advances in spatial databases. Lecture notes in computer science, Berlin, Heidelberg: Springer Berlin Heidelberg; 1999, p. 207–26.
[25] Horn BKP. Extended gaussian images. Proc IEEE 1984;72(12):1671–86.
[26] El-Mehalawi M, Miller RA. A database system of mechanical components based on geometric and topological similarity. Part II: indexing, retrieval, matching, and similarity assessment. Comput Aided Des 2003;35(1):95–105.
[27] Sundar H, Silver D, Gagvani N, Dickinson S. Skeleton based shape matching and retrieval. In: 2003 shape modeling international. IEEE; 2003, p. 130–9.
[28] Hilaga M, Shinagawa Y, Kohmura T, Kunii TL. Topology matching for fully automatic similarity estimation of 3D shapes. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques - SIGGRAPH '01. New York, New York, USA: ACM Press; 2001.
[29] Chen D-Y, Tian X-P, Shen Y-T, Ouhyoung M. On visual similarity based 3D model retrieval. In: Computer graphics forum, Vol. 22. Wiley Online Library; 2003, p. 223–32.
[30] Lian Z, Godil A, Sun X. Visual similarity based 3D shape retrieval using bag-of-features. In: 2010 shape modeling international conference. IEEE; 2010.
[31] Alcantarilla PF, Bartoli A, Davison AJ. KAZE features. In: European conference on computer vision. Springer; 2012, p. 214–27.
[32] Li L, Zhang S, Bai X, Shao L. Retrieving 3D model using compound-eye vis-ual representation. In: Computer-aided design and computer graphics (CAD/ Graphics) 2013 international conference on. 2013, p. 172–9.
[33] Liu Q. A survey of recent view-based 3D model retrieval methods. 2012, arXiv:1208.3670.
[34] Papadakis P, Pratikakis I, Theoharis T, Perantonis S. PANORAMA: A 3D shape descriptor based on panoramic views for unsupervised 3D object retrieval. Int J Comput Vis 2010;89(2–3):177–92.
[35] Vranic DV, Saupe D. 3D model retrieval (Ph.D. thesis), Citeseer; 2004.
[36] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE Inst Electr Electron Eng 1998;86(11):2278–324.
[37] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Commun. ACM 2017;60(6):84–90.
[38] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2016.
[39] Xiao Y-P, Lai Y-K, Zhang F-L, Li C, Gao L. A survey on deep geometry learning: From a representation perspective. Comput Vis Media 2020;6(2):113–33.
[40] Fang Y, Xie J, Dai G, Wang M, Zhu F, Xu T, Wong E. 3d deep shape descriptor. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, p. 2319–28.
[41] Su H, Maji S, Kalogerakis E, Learned-Miller E. Multi-view convo-lutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision, 2015, p. 945–53.
[42] Qi CR, Su H, Mo K, Guibas LJ. PointNet: Deep learning on point sets for 3D classification and segmentation. 2016, arXiv:1612.00593.
[43] Bickel S, Sauer C, Schleich B, Wartzack S. Comparing CAD part models for geometrical similarity: A concept using machine learning algo-rithms. Proc CIRP 2021;96:133–8.
[44] Yavartanoo M, Kim EY, Lee KM. Spnet: Deep 3d ob-ject classification and retrieval using stereographic projection. In: Asian confer-ence on computer vision. Cham: Springer; 2018, p. 691–706.
[45] Spruegel TC, Bickel S, Schleich B, Wartzack S. Approach and application to transfer heterogeneous simulation data from finite element anal-ysis to neural networks. J Comput Des Eng 2021;8(1):298–315.
[46] Kim H, Yeo C, Cha M, Mun D. A method of generating depth images for view-based shape retrieval of 3D CAD models from partial point clouds. Multimedia Tools Appl 2021;80(7):10859–80.
[47] Kim H, Yeo C, Lee ID, Mun D. Deep-learning-based retrieval of piping component catalogs for plant 3D CAD model reconstruction. Comput Ind 2020;123:103320.
[48] Kalyanaraman S, Iyer Y, Ramani N. Developing an engineering shape benchmark for CAD models. Comput Aided Des 2006;38(9):939–53.
[49] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. ImageNet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE; 2009.
[50] Singh A, Yadav A, Rana A. K-means with three different distance metrics. Int J Comput Appl 2013;67(10):13–7.
[51] Cha SH. Comprehensive survey on distance/similarity measures be-tween probability density functions. City 2007;1(2).
[52] Székely GJ, Rizzo ML, Bakirov NK. Measuring and testing dependence by jayanti. 2007.
[53] Jagadeesan P, Wenzel J, Corney JR, Yan XT, Sherlock A, Torres-Sanchez C, Regli W. Validation of purdue engineer-ing shape benchmark clusters by crowdsourcing. In: Proceedings of the International Conference on Product Lifecycle Management, Bath, UK, 2009.
[54] Shilane P, Min P, Kazhdan M, Funkhouser T. The princeton shape benchmark. In: Proceedings shape modeling applications, 2004. IEEE; 2004.
[55] Godil A, Lian Z, Dutagaci H, Fang R, Vanamali T, Cheung CP. Benchmarks, performance evaluation and contests for 3D shape retrieval. In: Proceedings of the 10th performance metrics for intelligent systems workshop, 2010, p. 42–7.
[56] Leifman G, Katz S, Tal A, Meir R. Signatures of 3D models for retrieval. In: Proceedings of the 4th Israel-Korea bi-national conference on geometric modeling and computer graphics, 2003, p. 159–63.
[57] Godil AA. SHREC'14 track: Large scale comprehensive 3D shape retrieval. 2020.
[58] Huang G, Liu Z, Van Der Maaten L, Weinberger KQ. 2017. Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, p. 4700–8.
[59] Zhang X, Zhou X, Lin M, Sun J. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, p. 6848–56.
[60] Iandola FN, Han S, Moskewicz MW, Ashraf K, Dally WJ, Keutzer K. Squeezenet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. 2016, arXiv preprint arXiv:1602.07360.
[61] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, p. 2818–2826.
[62] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, p. 1–9.
[63] Redmon J, Farhadi A. Yolov3: An incremental improvement. 2018, arXiv preprint arXiv:1804.02767.
[64] Zoph B, Vasudevan V, Shlens J, Le QV. Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, p. 8697–8710.
[65] Chowdhury NK, Tatsuma A, Aono M. A New Shape Descriptor based on Local and Global Feature for 3D Shape Retrieval. ITE Technical Report 37.56, The Institute of Image Information and Television Engineers; 2013, p. 19–22.
[66] Tatsuma A, Aono M. Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. Vis Comput 2009;25(8):785–804.
[67] Bai X, Bai S, Zhu Z, Latecki LJ. 3D shape matching via two layer coding. IEEE Trans Pattern Anal Mach Intell 2015;37(12):2361–73.
[68] Getto R, Kuijper A, Fellner DW. Unsupervised 3D object retrieval with parameter-free hierarchical clustering. In: Proceedings of the computer graphics international conference, 2017, p. 1–6.
[69] Mehrdad V, Ebrahimnezhad H. 3D object retrieval based on histogram of local orientation using one-shot score support vector machine. Front Comput Sci 2015;9(6):990–1005.
[70] Aono M, Koyanagi H, Tatsuma A. 3D shape retrieval focused on holes and surface roughness. In: 2013 Asia-Pacific signal and information processing association annual summit and conference. IEEE; 2013.
[71] Sfikas K, Theoharis T, Pratikakis I. Pose normalization of 3D models via reflective symmetry on panoramic views. Vis Comput 2014;30(11):1261–74.
[72] Zhuang T, Zhang X, Hou Z, Zuo W, Liu Y. A novel 3D CAD model retrieval method based on vertices classification and weights combination optimization. Math Probl Eng 2017;2017:1–12.
[73] Sfikas K, Theoharis T, Pratikakis I. ROSy+: 3D object pose normalization based on PCA and reflective object symmetry with application in 3D object retrieval. Int J Comput Vis 2011;91(3):262–79.
[74] Chen Q, Fang B, Yu Y-M, Tang Y. 3D CAD model retrieval based on the combination of features. Multimedia Tools Appl 2015;74(13):4907–25.
[75] Shih JL, Lee CH, Hou YW, Yeh PT. Three-dimensional model retrieval using dynamic multi-descriptor fusion. J Electron Sci Technol 2017;15(2):169–77.

[76] Tatsuma A, Aono M. Multi-Fourier spectra descriptor and augmentation with spectral clustering for 3D shape retrieval. Vis Comput 2009;25(8):785–804.
[77] Bai X, Rao C, Wang X. Shape vocabulary: a robust and efficient shape representation for shape matching. IEEE Trans Image Process 2014;23(9):3935–49.
[78] Zhu Z, Wang X, Bai S, Yao C, Bai X. Deep learning representation using autoencoder for 3D shape retrieval. Neurocomputing 2016;204:41–50.
[79] Li B, Johan H. 3D model retrieval using hybrid features and class information. Multimedia Tools Appl 2013;62(3):821–46.
[80] Papadakis P, Pratikakis I, Theoharis T, Passalis G, Perantonis S. 3D object retrieval using an efficient and compact hybrid shape descriptor. In: Eurographics workshop on 3d object retrieval. 2008.
[81] Pan X, You Q, Liu Z, Chen QH. 3D shape retrieval by Poisson histogram. Pattern Recognit Lett 2011;32(6):787–94.
[82] Zou K-S, Ip W-H, Wu C-H, Chen Z-Q, Yung K-L, Chan C-Y. A novel 3D model retrieval approach using combined shape distribution. Multimedia Tools Appl 2014;69(3):799–818.
[83] Vranic DV. Desire: A composite 3D-shape descriptor. In: 2005 IEEE international conference on multimedia and expo. IEEE; 2005.
[84] Bae MS, Park IK. Content-based 3D model retrieval using a single depth image from a low-cost 3D camera. Vis Comput 2013;29(6–8):555–64.
[85] Hou X, Zhang X, Liu W. Using enhanced shape distributions to compare CAD models. In: Advances in multimedia information processing – PCM 2007. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007, p. 385–8.
[86] Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3 d shape descriptors. In: Symposium on geometry processing, Vol. 6. 2003, p. 156–64.
[87] Kuo C-T, Cheng S-C. 3D model retrieval using principal plane analysis and dynamic programming. Pattern Recognit 2007;40(2):742–55.
[88] Osada R, Funkhouser T, Chazelle B, Dobkin D. Shape distributions. ACM Trans Graph 2002;21(4):807–32.
[89] Ohbuchi R, Minamitani T, Takei T. Shape-similarity search of 3D models by using enhanced shape functions. Int J Comput Appl Technol 2005;23(2/3/4):70.
[90] Kim S, Chi H-G, Hu X, Huang Q, Ramani K. A large-scale annotated mechanical components benchmark for classification and retrieval tasks with deep neural networks. In: Computer vision – ECCV 2020. Lecture notes in computer science, Cham: Springer International Publishing; 2020, p. 175–91.