Additive Manufacturing 36 (2020) 101554

Contents lists available at ScienceDirect

Additive Manufacturing

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

Computer aided design (CAD) model search and retrieval using frequency domain file conversion

Wenjin Li [a], Gary Mac [b], Nektarios Georgios Tsoutsos [c], Nikhil Gupta [a,*], Ramesh Karri [a]

[a] Department of Electrical and Computer Engineering, New York University, Tandon School of Engineering, Brooklyn, NY, 11201, USA
[b] Department of Mechanical and Aerospace Engineering, New York University, Tandon School of Engineering, Brooklyn, NY, 11201, USA
[c] Department of Electrical and Computer Engineering, University of Delaware, Newark, DE, 19716, USA

ARTICLE INFO

Keywords:
Additive manufacturing
Computer Aided Design
Shape search
Signal processing

ABSTRACT

The computer aided design (CAD) files are often tagged manually or searched and retrieved from a database using machine learning algorithms. There has been significant interest in developing capabilities of searching CAD files based on shape of the object. The present work demonstrates a method that transforms the CAD files to frequency domain representations and generates a set of fingerprints from the spectrogram to enable a search function. The fingerprints can be adjusted to allow finding either an exact match or a broader range of approximate/partial matches with similar features. A matching index is developed based on the similarity between the designs. The search does not require manual tagging of the files and is purely based on the geometry of the object. The success in comparing two files based on the object geometry is an important capability that can allow authentication of files in addition to conducting the search function.

1. Introduction

Computer aided design (CAD) models are widely used in the manufacturing industry. Digital manufacturing methods such as additive manufacturing (AM) use CAD model files as inputs for 3D printers to manufacture the part [1,2]. CAD models are widely used in architecture, construction, fashion, and many other industries. CAD software have evolved to provide capabilities such as creating complex shapes, developing assemblies of thousands of parts, working collaboratively on design files in the cloud, and conducting analysis (such as stress or heat transfer analyses) directly from the CAD interface. However, it is highly desired to have capabilities built in the CAD programs to search for a specific geometric shape present in a library of design files. This capability is especially missed as the new design tools taking advantage of embedded simulation tools or machine learning capabilities are able to automate the design process and create thousands of design variations, sometimes with only small geometrical differences, without human intervention. Sieving through this large number of design files for the ones containing the features of interest requires manual intervention of opening each file to observe the geometry, which defeats the purpose of automating the process.

An example of such automated design tools is topology optimization methods. These methods can conduct a large number of design iterations based on the pre-set optimization criteria such as minimization of structural weight or obtaining a specific value of structural stiffness [3,4]. Such methods can generate a large number of design files automatically, which the designers have to manually check for the various designs [5]. The design iterations generated by these algorithms look very similar and may have only minute differences from each other in dimensions of certain sections. Search and retrieve capabilities can help in selecting one or a small number of files containing a specific geometry. The search capability has the potential to transform the manufacturing industry into an automated service provider industry [6].

Several shape and solid model geometry retrieval techniques have been developed over the years with different degrees of automation. Manual tagging of files for certain shapes is a widely practiced method. Inclusion of metadata can be used for conducting search at a later stage. These methods, however, require user inputs for defining the shapes present in the file. In addition, tagging requires developing classifications: for example, tags may include manufacturing classification such as cast or machined part, or functional classifications such as brackets, gears, or springs [7]. The Opitz code system is widely used as a shape signature to classify and index CAD models [8–11]. Such algorithms have difficulty in comparing models created by software that may run on

* Corresponding author.
E-mail address: ngupta@nyu.edu (N. Gupta).

https://doi.org/10.1016/j.addma.2020.101554
Received 13 May 2020; Received in revised form 23 July 2020; Accepted 17 August 2020
Available online 25 August 2020

different geometric modeling kernels. Model setup steps also lead to differences in the representation of solid models that cannot be compared with each other. In addition, several design solutions provided by topology optimization algorithms may have the exact same design but only one dimension slightly varied, which will require them to use the same tags and the search would provide all those designs as exact matches. Another approach is based on extracting signatures of an object by taking the eigenvalues of its Laplace–Beltrami operator [12]. Since the spectrum is an isometry invariant, it is independent of factors such as parametrization and spatial position [12,13]. This approach is successful in identifying variants of the same model in the database search. Another approach analyses heat signature of a body by solving the heat equation and computing heat kernel, which is defined as quantity of heat received by a point after a unit of heat is applied at a certain reference point [2,14,15]. The comparison of heat signature provides the basis for comparison of 3D shapes. Such approaches are useful for search but are likely not lossless for recovering the shape from any intermediate step, which can provide additional functionalities in the additive manufacturing application domain, where the design files are used as inputs.

PROBADO3D is a 3D shape retrieval algorithm for automatically indexing 3D models used in architecture [16]. In order for a large database to be searchable using keywords, the models in the database are classified into categories. A supervised learning approach is used to segment 3D objects into local shape descriptors and teach the algorithm to classify each object (content-based indexing) [16]. Likewise, shape-based methods are used to retrieve 3D CAD files from databases [16–18]. VSC_WCO is a shape-based algorithm that converts models into three distance distribution histograms based on vertex classification (VSC) [18]. That algorithm is combined with a Weights Combination Optimization (WCO) scheme to improve the search results [18]. Content-based algorithms apply voxelization on the 3D model [19–21] before splitting the 3D model into a number of subspaces and calculating the entropy for all of them. By comparing those entropies, the algorithm can find the similar models [19]. These algorithms are more successful in finding a particular shape in large databases where a variety of shapes are available. However, if a database has variations of the same design, then the classification and retrieval become very challenging. Defining similarity is a complex issue in 3D designs, where different objectives for search may define similarity differently [22]. Classifications have been proposed to group object shapes together to define similarity. In the present work, either dimensional variations or design variations are used as similarity descriptors, which is elaborated in later sections. It is also pointed out in a previous work that some of these approaches are efficient in searching for manifold or isometric shapes but fail to apply to general 3D geometries of any shapes [23].

Another popular approach to find a matching model is by calculating the similarity between 2D views or sketches of 3D objects [24–29]. Chen et al. [25] proposed a system based on this approach, where the algorithm generates 100 orthogonal projections (excludes symmetry) per model and extracts a set of descriptors from those as the features for comparison. Pu et al. [29] find the difference between 2D views with these steps: (a) samples a large set of points on the edges, (b) calculates the Euclidean distance distribution between the points, and (c) finds the difference between the 2D shape distributions of 3D models [29]. In another sketch-based study, Li et al. [30] integrate adaptive view clustering and semantic information to improve the accuracy and efficiency. Liu et al. [31] use a feature mapping to retrieve 3D models from 2D images: First, they calculate the feature mapping metric based on the different forms of features from 3D models, and then their system uses the mapping metric to map the feature of the 2D image and the feature of the 3D model into one common space [31]. 3D point clouds are also used to represent a 3D shape in many cases. For example, Ip et al. [32] presented an approach that uses 3D point clouds of a physical part and calculates the distances between points cloud and the CAD mesh to retrieve the CAD model in a database.

Multiple groups have developed retrieval methods with design documents [33,34]. Jeon et al. [34] designed a semantic reference model that generates the semantic representations from the design documents (search query) as well as the models in the database. Then, they calculate a similarity measure and find how many components are matched between the representations [34]. Characteristics such as assembly statistics, joint relationships, and mating conditions are commonly used in searching through CAD assemblies [35–37]. Katayama et al. [35] use layouts of the components in the assembly to retrieve sub-assemblies. Their system gets sinograms by applying the Radon transform of the 3D model projections to find differences between shapes and layouts [35]. Min et al. [38] demonstrated a 3D CAD model retrieval system using multiple attributes providing better results than a single attribute-based retrieval system. Moreover, Funkhouser et al. [39] have presented a shape-based search engine that combines text and shape queries. Instead of using either one alone, their system employs text index, 2D index, and 3D index from each 3D model to improve the results [39]. Li et al. [40] proposed a reuse-oriented retrieval method that reduces the gap between 3D CAD retrieval and reuse by constructing a feature dependency directed acyclic graph (FDAG), which exploits the feature dependency of CAD models.

Koch et al. [41] constructed a large CAD Model dataset for geometric deep learning (DL). Using DL for CAD model classification reduces the labor costs and error-prone manual classification [21,42–44]. Likewise, NormalNet is a voxel-based convolutional neural network (CNN) used for 3D object classification and recognition [21]. Qin et al. [42] propose a deep neural network to build automatic classifiers, where 3D shape descriptors (i.e., view-based descriptors) are used to extract and characterize the shape information of 3D models for training purposes.

This work demonstrates the feasibility of a “Search” function by converting the CAD models files into frequency domain files and computing their audio IDStamps that enables search capability. Once a file is successfully converted to the frequency domain, a number of different possible approaches for audio fingerprinting and matching can be applied to find exact or similar matches. Indeed, there are several possible methods for file identification or authentication in the audio domain (e.g., a popular app for song identification is Shazam [45]). Nevertheless, the success of these methods for exact identification of CAD models depends on developing an algorithm that can provide a lossless conversion from CAD to audio domain. The loss in the conversion process would corrupt the audio signature, so that two files having only miniscule differences in shape or size may not be distinguished from each other. Our developed algorithm is demonstrated on complex-shaped 3D industrial component models of a wheel and a robotic gripper assembly. The database of CAD models includes models that have very similar shapes but contain only small differences to test the capability of the search method. The developed search methods are tested on a variety of shapes of different complexities to test the capabilities of the algorithm. The most challenging search case is found to be the design variations of a shape created by topology optimization methods that have only small differences, hence, these results are emphasized in the paper.

2. CAD Model Database Development

Solidworks and Fusion 360 are used to create a variety of shapes used in this work. The shapes include cubes, rectangular prisms, cones, discs, cylinders, donuts and other geometries of various dimensions. All the shapes described in this work are saved in the same database to conduct tests of search function. In addition, topology optimization algorithm is used on two base geometries to create a large number of variations of those designs. These model geometries are selected to have different features such as thin sections, sharp corners and edges, and curvatures of various sizes. Having a variety of geometrical features is useful in identifying potential limitations of the algorithms developed for converting the CAD file to frequency domain, as well as for conducting search. The model development procedure is detailed in the following sections; the exact boundary conditions and loads are not of interest for this work. The intent is to develop a search function for similar looking geometries, which may or may not be a result of the same optimization process. Hence, the details of the design process parameters are omitted here for brevity, but the entire database of designs is made available. The topology optimized shapes are also saved in the same database as the other shapes. In addition, Thingi10 K database is downloaded [46] and over 1600 unique shapes are randomly picked from the database and merged with the in-house developed shapes to create a database of over 2000 shapes to test the search capabilities of the proposed algorithms.

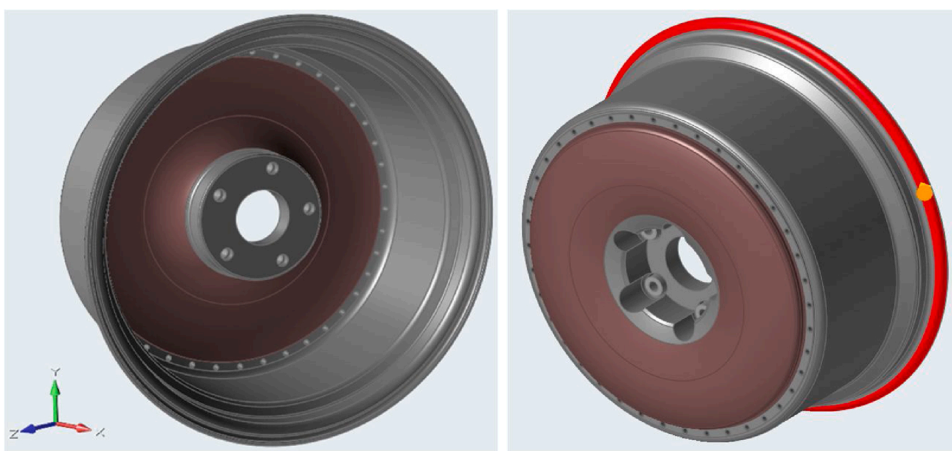
2.1. Model 1: Wheel hub

A wheel assembly model is developed using the *Altair Inspire* software, which allows for topology optimization studies. In the original wheel assembly (shown in Fig. 1a), the brown color zone represents the hub of the wheel and shows where the topology optimization is performed. After creating the geometry, loads and constraints are applied. The red line in Fig. 1b shows the location where the support boundary condition is applied to constrain the displacement of the model in all three dimensions. A force is applied to the wheel in the radial direction from the center of the hub (shown as the Y-direction in Fig. 1a). Cyclic symmetry is applied to the design space to ensure the spoke will result in self-symmetric sectors.

The design optimization goal is to find the optimized shape for the spokes of the wheel assembly with the least amount of mass and maximum stiffness. The design space mass target was set to 30% of the total design space volume for all the optimization studies. Fig. 2 shows the results of the optimization for two different iterations. A different symmetry constraint was applied to the model to achieve the different results. For each topology optimization study, the shape explorer conducts 21 iterations of the design, where the first iteration has a minimal amount of mass of the design and the last iteration shows the original design with no mass removed. In the STL file, the optimized spoke part is saved along with the components that were not affected by the optimization study such as the wheel rim and nuts.

2.2. Model 2: Robotic arm bracket

A robotic arm bracket is used as the second model, with the initial geometry of the model is shown in Fig. 3a. Three different magnitudes of loads are applied to the underside of the bracket, and a planar symmetry



(a) (b)

Fig. 1. (a) Model of a wheel. (b) Displacement boundary condition applied to the model.

(a) (b)

Fig. 2. Optimized model from the first study with (a) 1/4 symmetry and (b) 1/8 symmetry constraint.

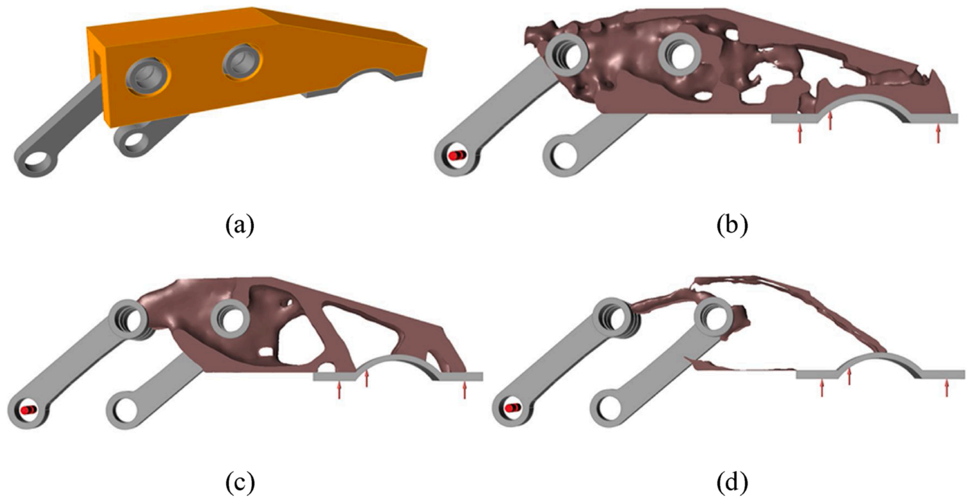


Fig. 3. (a) A robotic arm bracket. Different optimization results of the bracket where the design space kept (b) 50%, (c) 25% and (d) 5% of the total volume.

shape control constraint is applied at the midplane of the bracket's thickness. A total of 19 different topology optimization studies were created with the objective (same objective as for Model 1) of maximizing the stiffness of the bracket. Each of the optimization result groups has about 10 iterations saved for each study, and the dataset has 210 STL files of the optimization study. Each study was run multiple times at different mass target values and some of the results are summarized in Fig. 3(b)-(d).

3. Results and Discussion

3.1. Converting CAD files to frequency domain representation

A lossless algorithm is developed to convert the CAD 3D solid model files to frequency domain representation. The CAD programs offer export functionality, including the ability to store the model files in both STEP and STL formats, two of the widely used file types. Workflow is developed to convert STEP and STL formats to their frequency domain representation, and the conversion algorithm is summarized in the process diagrams of Fig. 4.

Conversion of the CAD files to frequency domain starts with converting a given solid model file into an audio file (WAV in this case). STL files can be saved in the binary format, which can be converted to WAV format using a built in function in Python. The STEP files need to undergo an intermediate conversion from STEP into a binary TXT file and then the Python function can be used for conversion. While using the "Impact Wave" function of Python [47], several parameters need to be defined, which are selected as: Channels: mono, Bitrate: 8, and Frame rate: 9000 or 44100 Hz. The frame rate or sampling rate of 9 kHz is the commonly used AM broadcast frequency and 44.1 kHz is the frequency used in compact discs (CDs). Other frame rates can be selected at will.

STL files do not contain metadata, which ensures that only model geometry is used to generate the frequency domain data in WAV format. Conversely, the metadata embedded in the STEP files needs to be scrubbed to retain only the geometry information. The metadata (such as timestamps) can make the file unique even for the same geometries; therefore, removal of this data can help in comparing the solid geometry. The Fast Fourier Transform (FFT) is used to convert the time-domain data into the desired frequency domain representation. The Python library *scipy.signal.spectrogram* helps to generate an array of sample frequency (f), an array of sample time (t), and the spectrogram of the input audio. Fig. 5a shows the spectrogram output of the wheel model design corresponding to Fig. 1a. This figure contains f and t as x- and y-axes while the colors in the figure represent the amplitude. Likewise, Fig. 5b shows the 3D representation of the spectrogram including amplitude peaks on the vertical axis. When converting the given TXT file into audio file (WAV) using the wave function, the user also needs to define the sampling rate of the output audio file. A possibility exists that the order of vertices in the STL file may be different in the query and the search database file. It is noted that the representation of a geometry does not depend on the order of vertices in the STL file. Hence, such possibilities can be addressed by conducting an intermediate step where the vertices are ordered in the ascending or descending order. However, such step was not conducted in this work.

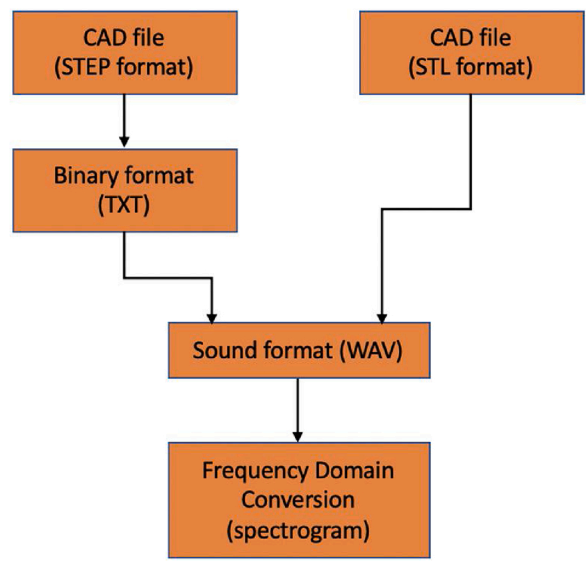


Fig. 4. Workflow to convert the CAD file (in the STEP and STL format files) to the frequency domain.

After analyzing each spectrogram, the coordinates of the amplitude peaks (i.e., pairs of frequency and time) are identified by using a maximum filter. The latter employs a sliding window as input and replaces each amplitude value with the local maximum amplitude within its neighborhood window. For example, Fig. 6 shows the result of applying the maximum filter function on an input image with a size of 5×5. It is noted that the sliding window size is defined by the user, so that when the user inputs a window size of 3 (as in Fig. 6), the maximum filter generates a 3×3 array (sliding window) to filter the local maximum value and returns a new multidimensional array. Then, the

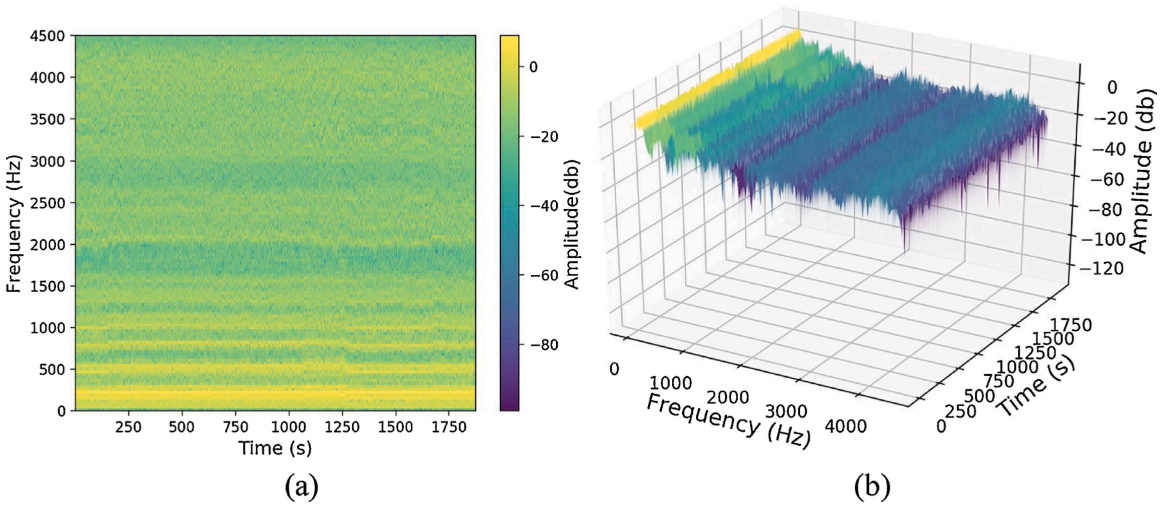


Fig. 5. (a) 2D and (b) 3D spectrograms of the wheel model design without any metadata.

algorithm compares this new multidimensional array with the original spectrogram to get the coordinates of maximum amplitude, which are listed as frequency-time index pairs, which is called "IDStamps." The sampling rate does not affect the IDStamps, which are based on the indices of frequency and time and not on the height of the amplitude peak. Fig. 7 shows the spectrogram of the wheel model design of Fig. 1a, with the identified peak locations marked with red dots. The number of output peaks major depends on the window size.

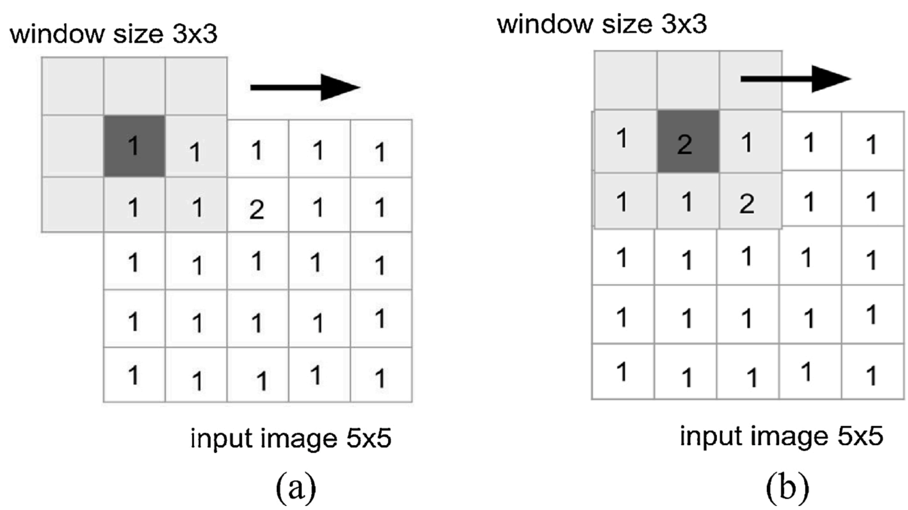


Fig. 6. Example of maximum filter with sliding window size of 3×3 on input image size 5×5 (a) and (b).

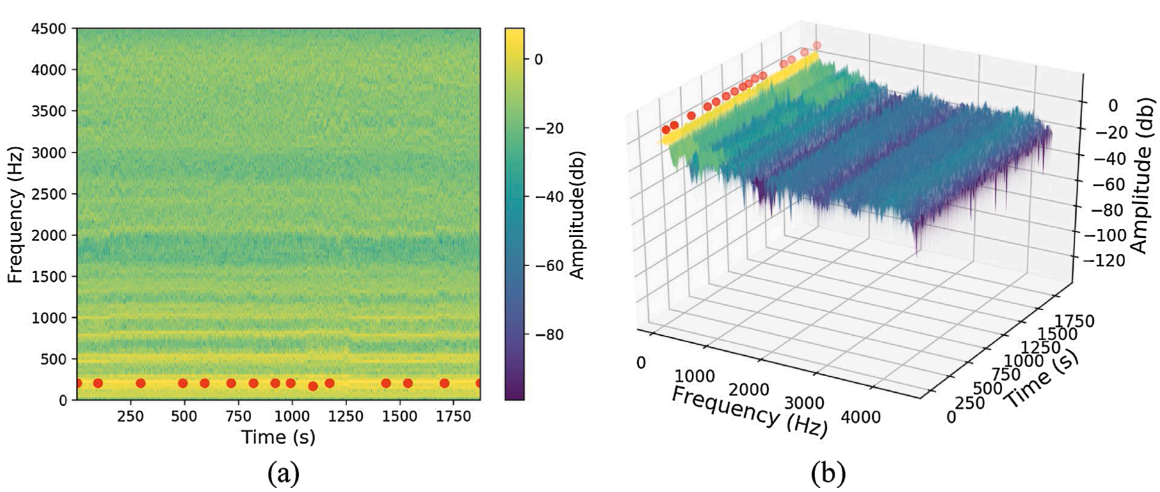


Fig. 7. (a) 2D and (b) 3D spectrograms showing peaks identified to generate IDStamps of the wheel model design.

Using the proposed approach, the same CAD model geometry can be recovered from the generated frequency domain file with a two-step process: First, a wave function library (without loss of generality, this work uses Python functions) is used to read the audio file (WAV) and generate a new TXT file with binary data. Then, the binary TXT file is converted back into a STEP design file from which can ultimately generate a CAD file using Autodesk Fusion 360.[1]

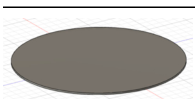
To store the IDStamps and support the search feature, a MySQL database is used. Moreover, to successfully find not only the exact match, but also close matches, the algorithm does not store all the IDStamps along with the corresponding filename into the database as a single entry. Instead, the pairs of frequency and time indices are stored individually along with the filename into the database. Therefore, each filename corresponds to multiple entries of IDStamps in the database. This approach offers freedom to calculate a matching rate based on the number of IDStamp matches between two candidate files. In this study, the implementation uses MySQL Connector Python to build the MySQL database. Although the present database contains all the converted files in WAV format for enabling the search functionality, algorithm can be modified to convert the files on the go from CAD format to frequency domain for capturing the fingerprints before conducting the search function. The files may or may not be saved in the WAV format after conducting the search.

3.2. Searching for the exact geometry match

To employ the search functionality, the query input CAD model is processed to generate its IDStamps. The algorithm executes a search query in the database and finds the result that has 100% match with the IDStamps of the search query. In the first set of tests, a number of different geometric shapes and conversion parameters are tested for determining the capability and accuracy of the algorithm.

All the models are saved in the same database so the search query has a potential to return results of any other shapes stored in the database. In the first test, the model is tested to determine if there is a limit to the smallest feature size that can be detected by the search algorithm. A disc of 5 mm radius and 0.1 mm thickness is taken as the test model. The size of disk in the search query is changed to by a fraction of a mm as per the data shown in Table 1. Change in the radius by 10^{-5} mm is the limit of the software for creating the smallest feature in this design. All cases show that the search query does not show a match with the original model when the dimensions are altered by any extent.

Table 1
A disc shape is tested for search accuracy and dimensional limits.

Disc (.step)			
	Radius (mm) ×0.1 mm Height		Similarity Index
Model ID	Original Model Radius	Search query Radius	
D1	5	5	1
D2	5	*5.1*	0.3967
D3	5	*5.001*	0.2833
D4	5	*5.00001*	0.2143

[1] The lossless recovery is implemented in the binary2step.py script available in the Appendix.

The test is further repeated with a rectangular prism and the results are shown in Table 2. Model P4 represents the limit of the software to make the smallest possible change in the geometry. The algorithm was successful in finding the exact match in the database. The examples of disc and prism are tested in. step format, which usually contains metadata as the part of the. step files. The tests were repeated after removing all the metadata and retaining only the part geometry and found to show the match only for the geometry of the exactly the same dimensions.

One of the parameters in converting the design files to frequency domains is the sampling frequency. The tests results presented in Tables 1 and 2 used 9 kHz sampling frequency for the conversion of files to the frequency domain. The sampling rate dependence of the search function was further tested on a variety of geometries as shown in Table 3. The database files were saved at 9 kHz frequency, while the search query files were saved at 44.1 kHz frequency in this example. The search function was able to find the correct match irrespective of the sampling frequency. Since each fingerprint is a pair of indices of frequency and time but excludes the peak height, the same model with different sampling frequency sound file leads to a match. These tests are also conducted at other frequencies and with and without metadata with success in retrieving the exact match.

Since this test focused on finding only the exact match, in most cases

Table 2
A prismatic shape tested for search accuracy[2].

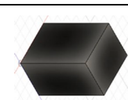
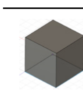
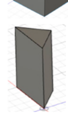
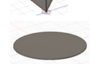
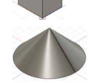
Model ID	Model (.step)		Similarity Index
	 Prism (mm)		
	Original Model Dimensions	Search String Dimensions	
P1	35(L) x *25(H)* x 20(W)	35(L) x *25(H)* x 20(W)	1
P2	35(L) x *25(H)* x 20(W)	35(L) x *25.01(H)* x 20(W)	0.3158
P3	35(L) x *25(H)* x 20(W)	35(L) x *25.0001(H)* x 20 (W)	0.0920
P4	35(L) x *25(H)* x 20(W)	35(L) x *25.00001(H)* x 20 (W)	0.1130

Table 3
Results of matching IDStamps obtained at two different frequencies. In each case, matching index of 1 is shown for the exact match.

Model (.step)		File conversion frequency 1 (Hz)	File conversion Frequency 2 (Hz)
	Cube	9000	44100
	Triangular prism	9000	44100
	Disc	9000	44100
	Prism	9000	44100
	Cone	9000	44100

it was noticed that the number of IDStamps captured using the same window size were different for files having geometric variations of the same shape, for example discs of different thicknesses. A simple comparison of the number of IDStamps can also provide a fast first step elimination, where actual indices captured in the IDStamps are compared only for those files that have exactly the same number of IDStamps. Having several matching IDStamps is an indication of similarity between two shapes, which is explored in the next step in detail.

3.3. Searching and indexing the search results

The search functionality is expanded to include potentially similar matches. To determine whether the file in the database is a match or not, the algorithm calculates a similarity Index (S) for each CAD file in the database using number of IDStamps in the query input file (T), number of IDStamps for each file in the database (D), and number of matches between the two files (N)

$$S = \frac{N}{\frac{(T+D)}{2}} = \frac{2 \times N}{T+D} \tag{1}$$

The index is a statistical parameter that captures the ratio of matching IDStamps with the total number of IDStamps. Since the size of each CAD file is different, the absolute number of IDStamps associated with each CAD file may vary. Thus, while calculating the similarity index, the formula may not consider only the number of IDStamps in query input or only the number of IDStamps in a database file, as there would be two limiting cases when running a search:

$$\text{when } \mathrm{T} > \mathrm{D} \text{ and } \mathrm{D} = \mathrm{N} \tag{i}$$

$$\text{when } \mathrm{T} < \mathrm{D} \text{ and } \mathrm{T} = \mathrm{N} \tag{ii}$$

For case (i), consider the scenario where a query input file A contains 2000 IDStamps, and file B in the database contains 200 IDStamps. Then, if the number of matches is 200 (i.e., the maximum possible matches) and the equation for S simply uses "N divide by D", the result would saturate to 1. Nevertheless, file A may not be an exact match of file B. A similar issues arises in case (ii). Instead, the formula for S computes the sum of the number of IDStamps from both files as total number IDStamps. Then, the number of matches is divided by the average $(T+D)/2$ to compute the similarity index S, as shown as Equation (1).

The matching algorithm ranks all the files based on the IDStamps matched using the aforementioned process and then either all or some of the matches are displayed based on the matching rank; alternatively, the desired number of similar matches can also be pulled from the database. As expected, an exact match of the query file will have $S = 1$. Notably, the major parameter affecting the matching performance is the window size during peak selection: using a larger window size would decrease the number of identified peaks, as it will identify only one peak within the selected window, and eventually decrease the number of IDStamps of a given CAD model. Hence, when the search function runs with the larger window size, the algorithm may output similar (but not exactly identical) CAD files with S reported as equal to 1. Conversely, when the algorithm uses a smaller window size, each CAD file will have a larger number of IDStamps, which offers higher search accuracy.

The tests for searching similar shapes are conducted on the models created by a topology optimization algorithm. These shapes retain geometric features based on applied constraints and result in models that may have similar geometries, sometimes with only subtle differences in dimensions or contours. The file tagging methods cannot be applied to these datasets because the same tags would apply to all the files that belong to the same design family. Here, the present work is expected to demonstrate the advantage. The search is run on the database containing all the shapes described in this work, including the basic geometries shown in the first test. The search will allow analyzing if the models belonging to the same family of designs are identified by the algorithm.

Fig. 8 illustrates the search results in a database containing design files of 90 different wheel models and 224 different gripper models for a particular design iteration to find an exact match. For our evaluation, the window size is set to 5k during search. Then, a CAD file is randomly selected (e.g., 40.stl) and then converted to generate its corresponding IDStamps, which are then used to conduct the search functionality. The results summarized in Table 4 demonstrate 7 out of possible 89 matches from a total of 314 files present in the database. Only one result has $S = 1$, which is the exact model, and the rest of the models are ranked based on their computed similarity score. It is observed that the similar matches are based on the frequency domain parameters and one possible limitation of the current approach is that there is no direct correspondence identified between certain geometric features and the corresponding fingerprints. The latter can be an exciting topic to be investigated in detail in the future work. While some of these models have only subtle differences, the similarity index still deviates from 1 in Table 4. A detailed study was conducted to investigate the level of change in a CAD model that will provide deviation in the similarity index. Even if the change is the smallest allowed dimensional step by the CAD software in one dimension of solid model, the similarity index is not 1 as documented for a disc in results included in Table 1.

The second test was performed using the gripper bracket model and Table 5 illustrates part of the output for an input file "P8.stl" and a window size of 200. The search function returns one exact match and 56 similar matches for this geometry; these partial outputs are illustrated in Table 5. It is observed that the search is able to provide the exact match and also list other gripper models from the database.

The tests of the algorithm developed in this work shows success in identifying exact match and also list the variation of the same model although the same database contains files for both designs. This algorithm can result in the possibility of identifying desired files in a large database of solid models. It is also possible to use the algorithm developed in the present to authenticate the CAD files or test them against any tempering because any change results in a different similarity index value.

```
-------S--------
input file: 40.wav
# of ID stamps: 15
1  exact match
89  similar match
-------E-------
```

Fig. 8. Format of printed search results from the MySQL database.

Table 4
Partial output using window size 5k on input file "40.stl" showing matching models.

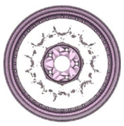
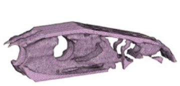
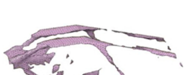
Model	File description	Output	Similarity Index
 40.stl	Study 1 - Iteration #2 Smooth Results - Disabled Total Mass (kg) - 20.109 Total Volume (cm^3) - 2513.658 Optimized Part Mass (kg) - 0.125 Optimized Part Volume (cm^3) - 15.624	**Search string file (query)**	
 40.stl	Same as the search string file.	***Match 1 (Exact match)***	***1***
 21.stl	Study 1 - Iteration #1 Smooth Results - Enabled Total Mass (kg) - 19.984 Total Volume (cm^3) - 2498.0395 Optimized Part Mass (kg) - 4.4004 × 10-5 Optimized Part Volume (cm^3) - 0.0055 contains no volume of material from the design space of the optimization study	***Match 2***	***0.8667***
 20.stl	Study 1 - Iteration #2 Smooth Results - Enabled Total Mass (kg) - 20.109 Total Volume (cm^3) - 2513.658 Optimized Part Mass (kg) - 0.125 Optimized Part Volume (cm^3) - 15.624	***Match 3***	***0.8667***
 Original.stl	Original Wheel Smooth Results - N/A Total Mass (kg) - 45.108 Total Volume (cm^3) - 5638.534 Optimized Part Mass (kg) - 25.124 Optimized Part Volume (cm^3) - 3140.5	***Match 4***	***0.8125***
 39.stl	Study 1 - Iteration #3 Smooth Results - Disabled Total Mass (kg) - 20.418 Total Volume (cm^3) - 2552.253 Optimized Part Mass (kg) - 0.43375 Optimized Part Volume (cm^3) - 54.218	***Match 5***	***0.7879***

Table 5
Partial output using window size 200 using an input query file "P8.stl".

Model	Description	Output	Similarity Index
 P8.stl	Study P - Iteration #8 Smooth Results - Enabled Total Mass (kg) - 0.3726 Total Volume (cm^3) - 46.571 Max Length (cm) - 16.159 Max Height (cm) - 4.612 Max Thickness (cm) - 3.023	**Search string file (query)**	
 P8.stl	Study P - Iteration #8 Smooth Results - Enabled Total Mass (kg) - 0.3726 Total Volume (cm^3) - 46.571 Max Length (cm) - 16.159 Max Height (cm) - 4.612 Max Thickness (cm) - 3.023	***Match 1 (exact match)***	***1***
 R7.stl	Study R - Iteration #7 Smooth Results - Enabled Total Mass (kg) – 0.3934 Total Volume (cm^3) - 49.172 Max Length (cm) - 17.23 Max Height (cm) - 4.613 Max Thickness (cm) - 3.02	***Match 2***	***0.0112***
 A1.stl	Study A - Iteration #2 Smooth Results - Enabled Total Mass (kg) – 0.0320 Total Volume (cm^3) – 4.0002 Max Length (cm) – 15.476 Max Height (cm) - 4.621 Max Thickness (cm) - 3.02	***Match 3***	***0.0051***

4. Conclusions

This work is focuses on developing a new method that enables searching CAD files for 3D solid model geometries. In our approach, the CAD model files are converted to a frequency domain representation using our lossless algorithm that enables fully reversible conversion of CAD files. The desired lossless conversion property is confirmed by transforming the frequency domain file back to the original CAD model without any loss of information in the 3D solid model. The present approach is focused on two widely used file formats, namely STL and STEP, and both formats can be successfully converted to/from the frequency domain. Once the exact frequency domain representation of the CAD file is obtained, search functionality could be implemented by comparing the proposed IDStamps of the corresponding spectrograms. The search function is demonstrated to find the exact match of geometries of various shapes including prism, cylinder, cone, disc, and many other shapes. Moreover, a similarity index is developed that quantifies the similarity between any two files on a scale between 0 to 1, where a perfect match is rated as 1. The method is extensively evaluated using databases of two geometries, a wheel and a robotic gripper generated using topology optimization methods. Our results indicate that the proposed search algorithm can find an exact match for the given query file, as similar matches based on the same geometry.

CRediT authorship contribution statement

Wenjin Li: Methodology, Validation, Investigation, Writing - original draft. **Gary Mac:** Methodology, Validation, Investigation, Writing - original draft. **Nektarios Georgios Tsoutsos:** Conceptualization, Writing - review & editing, Funding acquisition. **Nikhil Gupta:** Conceptualization, Validation, Investigation, Writing - original draft, Funding acquisition. **Ramesh Karri:** Conceptualization, Writing - review & editing, Funding acquisition.

Declaration of Competing Interest

The authors report no declarations of interest.

Acknowledgment

This work is supported by the National Science Foundation grants CMMI-1932264 and CMMI- 1931916. The opinions expressed in the articles are those of authors, not of funding agency.

References

[1] H.E. Burton, S. Peel, D. Eggbeer, Reporting fidelity in the literature for computer aided design and additive manufacture of implants and guides, Additive Manufacturing 23 (2018) 362–373.
[2] Y. Shi, Y. Zhang, S. Baek, W. De Backer, R. Harik, Manufacturability analysis for additive manufacturing using a novel feature recognition technique, Computer-Aided Design and Applications 15 (6) (2018) 941–952.
[3] C. Greer, A. Nycz, M. Noakes, B. Richardson, B. Post, T. Kurfess, L. Love, Introduction to the design rules for Metal Big Area Additive Manufacturing, Additive Manufacturing 27 (2019) 159–166.
[4] F. Mezzadri, V. Bouriakov, X. Qian, Topology optimization of self-supporting support structures for additive manufacturing, Additive Manufacturing 21 (2018) 666–682.
[5] A.T. Gaynor, T.E. Johnson, Eliminating occluded voids in additive manufacturing design via a projection-based topology optimization scheme, Additive Manufacturing 33 (2020), 101149.
[6] A. Angrish, B. Craver, B. Starly, "FabSearch": A 3D CAD Model-Based Search Engine for Sourcing Manufacturing Services, Journal of Computing and Information Science in Engineering 19 (4) (2019).
[7] D. Bespalov, C.Y. Ip, i.C. Regl, J. Shaffer, Benchmarking CAD search techniques, in Proceedings of the 2005 ACM symposium on Solid and physical modeling, Association for Computing Machinery, Cambridge, Massachusetts, 2005, pp. 275–286.
[8] L. Zehtaban, O. Elazhary, D. Roller, A framework for similarity recognition of CAD models, Journal of Computational Design and Engineering 3 (3) (2016) 274–285.
[9] A. Fradi, B. Louhichi, M.A. Mahjoub, B. Eynard, 3D Object Retrieval Based on Similarity Calculation in 3D Computer Aided Design Systems, 2017 IEEE/ACS 14th International Conference on Computer Systems and Applications (AICCSA) (2017).
[10] L. Zehtaban, D. Roller, Automated Rule-based System for Opitz Feature Recognition and Code Generation from STEP, Computer-Aided Design and Applications 13 (3) (2016) 309–319.
[11] L. Zehtaban, D. Roller, Beyond Similarity Comparison: Intelligent Data Retrieval for CAD/CAM Designs, Computer-Aided Design and Applications 10 (5) (2013) 789–802.
[12] M. Reuter, F.-E. Wolter, N. Peinecke, Laplace–Beltrami spectra as 'Shape-DNA' of surfaces and solids, Computer-Aided Design 38 (4) (2006) 342–366.
[13] M. Reuter, F.-E. Wolter, M. Shenton, M. Niethammer, Laplace–Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis, Computer-Aided Design 41 (10) (2009) 739–755.
[14] R. Harik, Y. Shi, S. Baek, Shape Terra: mechanical feature recognition based on a persistent heat signature, Computer-Aided Design and Applications 14 (2) (2017) 206–218.
[15] T.K. Dey, K. Li, C. Luo, P. Ranjan, I. Safa, Y. Wang, Persistent Heat Signature for Pose-oblivious Matching of Incomplete Models, Computer Graphics Forum 29 (5) (2010) 1545–1554.
[16] I. Blümel, R. Berndt, S. Ochmann, R. Vock, R. Wessel, PROBADO3D - Indexing and Searching 3D CAD Databases: Supporting Planning through Content-Based Indexing and 3D Shape Retrieval, in: International Conference on Design and Decision Support Systems in Architecture and Urban Planning, - Eindhoven, Netherlands, 19-22 Jul 2010, 2010, pp. 411–425.
[17] H. Kim, M. Cha, D. Mun, Shape distribution-based retrieval of 3D CAD models at different levels of detail, Multimedia Tools and Applications 76 (14) (2017) 15867–15884.
[18] T. Zhuang, X. Zhang, Z. Hou, W. Zuo, Y. Liu, A Novel 3D CAD Model Retrieval Method Based on Vertices Classification and Weights Combination Optimization, Mathematical Problems in Engineering 2017 (2017), 6049750.
[19] L. Wei, H. Yuanjun, Representation and retrieval of 3D CAD models in parts library, The International Journal of Advanced Manufacturing Technology 36 (9) (2008) 950–958.
[20] T.P. Subba, Ashis, Prerna Rai, "Voxelization" in 3D Searching- A Study, International Journal of Computer Applications 132 (2015) 37–40.
[21] C. Wang, M. Cheng, F. Sohel, M. Bennamoun, J. Li, NormalNet: A voxel-based CNN for 3D object classification and retrieval, Neurocomputing 323 (2019) 139–147.
[22] M. Novotni, R. Klein, Shape retrieval using 3D Zernike descriptors, Computer-Aided Design 36 (11) (2004) 1047–1062.
[23] V. Jain, H. Zhang, A spectral approach to shape-based retrieval of articulated 3D models, Computer-Aided Design 39 (5) (2007) 398–407.
[24] M. Shi, S. Zhang, L. Li, X. Bai, 3D CAD Model Retrieval Using 2D Representative Views, 2012 Spring Congress on Engineering and Technology (2012).
[25] D.-Y. Chen, X.-P. Tian, Y.-T. Shen, M. Ouhyoung, On Visual Similarity Based 3D Model Retrieval, Computer Graphics Forum 22 (3) (2003) 223–232.
[26] L. Li, H. Wang, T. Chin, D. Suter, S. Zhang, Retrieving 3D CAD models using 2D images with optimized weights, 2010 3rd International Congress on Image and Signal Processing (2010).
[27] T. Filaliansary, J.-P. Vandeborre, M. Daoudi, A framework for 3D CAD models retrieval from 2D images, Annales Des Télécommunications 60 (11) (2005) 1337–1359.
[28] F.-W. Qin, S.-M. Gao, X.-L. Yang, J. Bai, Q.-H. Zhao, A sketch-based semantic retrieval approach for 3D CAD models, Applied Mathematics-A Journal of Chinese Universities 32 (1) (2017) 27–52.
[29] J. Pu, K. Lou, K. Ramani, A 2D Sketch-Based User Interface for 3D CAD Model Retrieval, Computer-aided Design and Applications (2005) 717–725.
[30] B. Li, Y. Lu, H. Johan, R. Fares, Sketch-based 3D model retrieval utilizing adaptive view clustering and semantic information, Multimedia Tools and Applications 76 (24) (2017) 26603–26631.
[31] A. Liu, N. Liu, W. Nie, Y. Su, 3D model retrieval via single image based on feature mapping, Multimedia Tools and Applications 77 (17) (2018) 22051–22069.
[32] C.Y. Ip, S.K. Gupta, Retrieving Matching CAD Models by Using Partial 3D Point Cloud, Computer-aided Design and Applications 4 (2007) 629–638.
[33] W.-d. Yu, J.-y. Hsu, Content-based text mining technique for retrieval of CAD documents, Automation in Construction 31 (2013) 65–74.
[34] S.M. Jeon, J.H. Lee, G.J. Hahm, H.W. Suh, Automatic CAD model retrieval based on design documents using semantic processing and rule processing, Computers in Industry 77 (2016) 29–47.
[35] K. Katayama, W. Sato, Subassembly Retrieval of 3D CAD Assembly Models with Different Layout of Components Based on Sinogram, IEICE Transactions on Information and Systems E102.D (4) (2019) 777–787.
[36] K. Lupinetti, F. Giannini, M. Monti, J.-P. Pernot, Multi-criteria retrieval of CAD assembly models, Journal of Computational Design and Engineering 5 (1) (2018) 41–53.
[37] K. Lupinetti, J.-P. Pernot, M. Monti, F. Giannini, Content-based CAD assembly model retrieval: Survey and future challenges, Computer-Aided Design 113 (2019) 62–81.
[38] P. Min, M. Kazhdan, T. Funkhouser, A Comparison of Text and Shape Matching for Retrieval of Online 3D Models, Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.
[39] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, D. Jacobs, A search engine for 3D models, ACM Trans. Graph. 22 (1) (2003) 83–105.
[40] M. Li, Y.F. Zhang, J.Y.H. Fuh, Z.M. Qiu, Toward Effective Mechanical Design Reuse: CAD Model Retrieval Based on General and Partial Shapes, Journal of Mechanical Design 131 (12) (2009).

[41] S. Koch, A. Matveev, Z. Jiang, F. Williams, A. Artemov, E. Burnaev, M. Alexa, D. Zorin, D. Panozzo, ABC: A big CAD model dataset for geometric deep learning, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2019).

[42] F.-W. Qin, L.-Y. Li, S.-M. Gao, X.-L. Yang, X. Chen, A deep learning approach to the classification of 3D CAD models, Journal of Zhejiang University SCIENCE C 15 (2) (2014) 91–106.

[43] C. Zhang, G. Zhou, H. Yang, Z. Xiao, X. Yang, View-Based 3-D CAD Model Retrieval With Deep Residual Networks, IEEE Transactions on Industrial Informatics 16 (4) (2020) 2335–2345.

[44] C. Zhang, G. Zhou, A view-based 3D CAD model reuse framework enabling product lifecycle reuse, Advances in Engineering Software 127 (2019) 82–89.

[45] A. Wang, An Industrial Strength Audio Search Algorithm, in: Proceedings of the 4 th International Conference on Music Information Retrieval, October 26-30, 2003, Baltimore, Maryland, USA, 2003.

[46] Q. Zhou, A. Jacobson, Thingi10K: A Dataset of 10,000 3D-Printing Models, 2016 arXiv:1605.04797, https://arxiv.org.

[47] Python Software Foundation, 2020 [cited 2020, June 14, https://docs.python.org/3/library/wave.html.