Solid Model Similarity for Engineering Applications using Congruence of Triangles

Christopher L. Sousa[1] and Rahul S. Renu[2]

[1]Clemson University, clsousa@g.clemson.edu
[2]Francis Marion University, rrenu@fmarion.edu

Corresponding author: Rahul S. Renu, rrenu@fmarion.edu

Abstract. This research is an investigation of a solid model retrieval method based on congruence of triangles - Congruent Triangle Similarity (CTS). In the CTS method, three side lengths of each triangle of a query mesh solid model are computed, and the frequency of side length triplets are compared with those of a database model. This comparison is computed as a ratio of number of congruent triangles across pairs of solid models to the total number of triangles.
CTS method is evaluated using the Engineering Shape Benchmark with standard information retrieval metrics. Results indicate that (1) retrieval performance shares an inverse-U relationship with resolution of solid models, (2) retrieval performance peaks when number of decimals used is on. It is thus concluded that the CTS method is an effective method of solid model retrieval. The CTS method has an advantage of having good retrieval performance on low fidelity solid model files.

Keywords: Solid Model Similarity, Information Retrieval, CAD Similarity, Design Reuse
DOI: https://doi.org/10.14733/cadaps.2021.1096-1110

1 INTRODUCTION

During product design, three-dimensional solid models are developed to serve as virtual representations of the desired product. The ability to accurately recall existing solid models from a database in order to recycle designs would positively impact the time requirements of the product design phase. Whether it be repurposing existing parts or reverse engineering designs in order to make a new product entirely, design reuse aids in reducing the cost associated with the product development process.

Further, downstream operation data (such as manufacturing quality) concerning the product is collected and coupled with the previously created solid model through the use of Product Data Management systems. This coupling of solid models and relevant manufacturing data allows for an organization to make predictions about product behavior, based on design data [4,30]. For example, if it has been previously learned that a part's geometry is not conducive to the machining capabilities of a company, then a geometrically similar part may also pose issues (assuming machining capabilities remain unchanged). The prediction and mitigation of manufacturing concerns during the product design step would save time and reduce cost in the manufacturing phase of product development. This prediction can be facilitated through retrieval of solid models and their associated information. In order to objectively retrieve models with a similar geometry to what is desired, solid model similarity methods are used.

The household use of solid model similarity stems from the additive manufacturing revolution. With additive manufacturing machines becoming more affordable and web-based databases of solid models expanding their catalogues through crowdsourcing, the need for an engineering solid model retrieval mechanism is increasing. Ultimately, the desire to create a solid model similarity assessment method stems from the need for an objective method of solid model retrieval from a database. The currently popular text-based querying method is associated with high amounts of subjectivity. For example, using text queries to retrieve models from a database relies entirely on the assumption that the language, descriptions, and/or tags used by the user searching for a model will match those used by the creator of the model. Dialect differences between even two English speaking users could be the difference between querying the text phrase "boot" and retrieving either footwear or the rear, storage portion of an automobile. These language differences are only exaggerated by translational disconnects between speakers of different languages. Many multinational companies will find themselves with various solid models created by employees who speak different languages. While the troubles of text querying caused by various dialects of the same language may be solved through detailed part descriptions, language barriers caused by non-exact translations will persist. Additionally, creating and reading hundreds of lengthy descriptions of solid models would only further complicate and lengthen the product development process. Therefore, query-by-solid-model (see Figure 1) is a preferred approach.

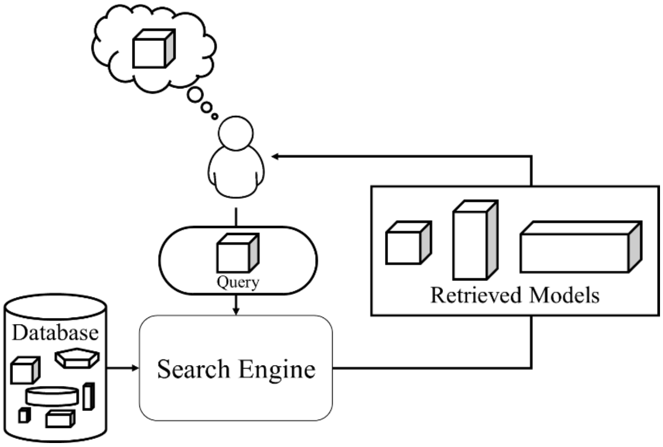


Figure 1: Query-by-solid-model method for solid model retrieval

The work presented in this paper compares triangular mesh solid models based on the congruence of their constituent triangles. The framework and scope of this research is presented in Figure 2. This work requires the existence of three-dimensional solid models (in boundary representation format [28]) within a company's database. These models must then be converted into a triangular mesh format (like STL) using one of many standard methods/software [2,6]. It is critical that the conversion from B-REP to triangular mesh representation occurs with consistent settings for all database models.

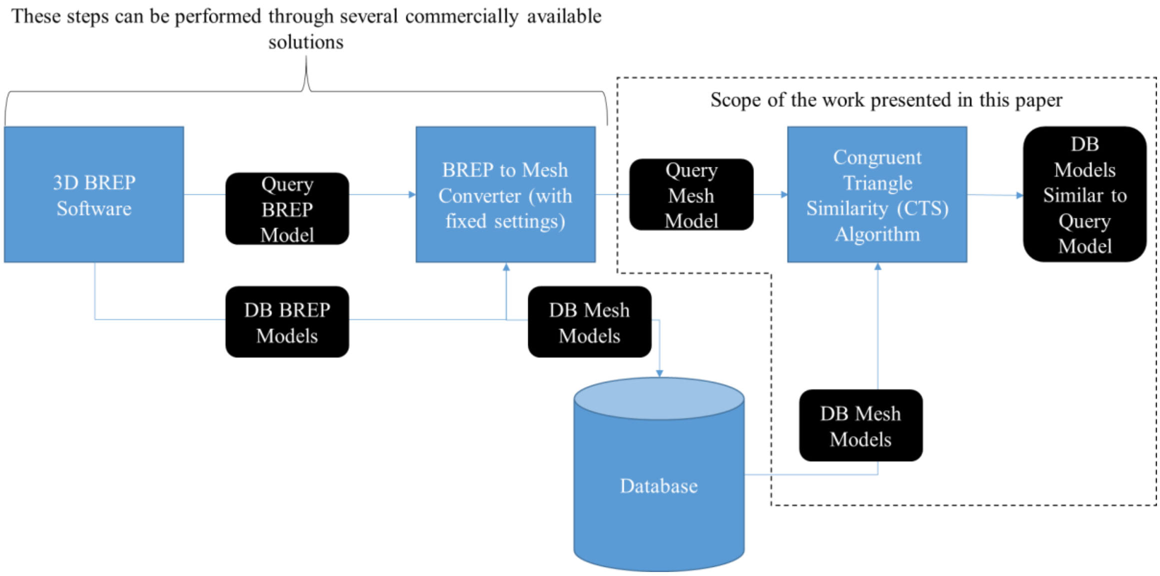


Figure 2: Framework and Scope of this Research.

The premise for this research is that if all triangles from a solid model are congruent to those from another solid model, then these solid models are identical. Furthering this, if some fraction (< 1) of triangles from a solid model are congruent to those from another solid model, then they might be considered similar (yet, not identical). Quantification of the similarity between two such models is investigated, and results presented in subsequent sections.

2 REVIEW OF RELEVANT WORK

This work is motivated by the increasing use of solid model similarity for search and retrieval within engineering databases. While numerous methods to compute solid model similarity exist, few focus on assessing these methods for their ability to retrieve models from the perspective of engineering relevance. In this paper, methods that have been used for engineering purposes are reviewed and their shortcomings are identified. A more extensive review of general solid model similarity methods can be found in [1,10,19].

Solid model similarity methods have been investigated and applied to retrieval of machining process planning [14,23,9], assembly time estimates [13,20,21,27,26], assembly work instructions [24], and other solid models [11,16,22,29,15,17]. One of the oldest forms of solid model comparisons is Group Technology [14]. In this method, an alphanumeric string is assigned manually to a part based on its machining features. This alphanumeric string is then used for search and retrieval of machining process plans. A major drawback of this process is the subjectivity associated with the manual assignment of alphanumeric codes. Researchers have worked to overcome this by automating the assignment process [5] and hence making it repeatable and consistent. However, the granular nature of comparison leads to false positives being identified.

Hong and colleagues [7] have developed a two-stage approach for retrieving solid models of mechanical parts for the purposes of part reuse. In the first stage, a coarse comparison of overall shape is performed. This is performed using the D2 method proposed by Osada and colleagues [22]. Once a subset of coarsely similar solid models have been identified, a finer level of comparison is performed using Boundary Representation (B-rep) [28] information. This allows the comparison of individual features between two solid models. This method is dependent on the presence of B-rep information, which is absent in formats such as STL and AMF. This disallows the method from Hong and colleagues from being used on parts generated for additive manufacturing purposes. The requirement of B-rep information is also found in Li and colleagues' work [15].

Tao and colleagues [29] use partial shapes within solid models to search for other solid models with matching partial shapes. In other words, retrieval is performed on partial shape matches as opposed to global shape descriptors. While there are several high-impact applications of partial shape-based searches, global shape-based searches lend themselves better to be applied in 3D query-by-model search engines.

Another effort to use solid model similarity for engineering applications comes from Iyer and colleagues [11]. They investigated a shape-based search method for applications throughout a product's lifecycle. Their method allows users to Query-By-Example, Query-By-Sketch, or through Feature Vector Choice. Of particular interest is the Query-By-Sketch method, which requires a user to generate an approximate query solid model. This query model is then voxelized using its B-rep data, and subsequently a representative skeletal graph is generated. The graph parameters, voxel parameters, geometric parameters, and moment invariants are used to generate a feature vector for a solid model. Feature vectors of database solid models are compared to that of the query solid model for search and retrieval of the former. The following drawbacks are associated with this method: (1) The multi-step process is computationally intensive; (2) The method relies on the existence of B-rep information for voxelization.

Ramesh and colleagues [23] use solid model similarity for variant process planning in manufacturing. Their method focuses on the extraction of solid model features, mapping these solid model features to machining features, and also focuses on a similarity metric based on solid model features that affect machining. For the last part, the authors use feature existence, feature count, feature direction, feature size, directional distribution, size distribution, and relative orientation to vectorize, and subsequently compare solid models. Similar work comes from Elinson and colleagues [3], who have developed a solid model similarity assessment technique for the purposes of retrieval of manufacturing plans. Their algorithm identifies geometric features relevant to machining and constructs a graph structure from these features. The graph structures from the query model are then compared to those from database models, and manufacturing plans of the latter are retrieved. Unlike the work from Iyer and colleagues [11], these works are narrowly focused on machining applications and do not have extended use in other engineering applications.

A relatively different approach was explored by Manns and colleagues [18], who have investigated methods of predicting assembly process and assembly time information from product geometries. Their method involves the conversion of solid models of an assembly (of multiple parts) to a feature vector. This feature vector includes part weight, centre of gravity and outer dimensions, and is used to cluster similar solid models together. Clusters of similar solid models are then associated with clusters of process descriptions. This allows for a new solid model to be assigned to a cluster (through unsupervised machine learning) and its corresponding process description cluster be retrieved as a prediction. While this research does more than just identify similar solid models, there are several concerns. First, language used in their paper [18] seems to suggest that the feature vectors are composed of factors beyond those that are stated explicitly. Second, rationale for use of the elements of the feature vector are not provided - this disallows one from extending their method to other engineering applications beyond assembly time prediction. Lastly, the features used for clustering include factors beyond the solid model themselves (number of standard parts used, for instance).

Other researchers have implicitly used solid model similarity as a conduit between the early engineering design stages and downstream manufacturing stages. Namouz and colleagues [21] have used the interference between parts in an assembly solid model to predict assembly time estimates. Their approach uses artificial neural networks to relate connectivity metrics to assembly time estimates and therefore, they don't explicitly compute the similarity between two solid models. In addition, the black box approach used doesn't allow researchers to understand the relationship between solid models that allow the prediction of assembly times. Renu and colleagues [24] have used the D1 metric to relate solid models to assembly work instructions. They have also developed a method that generates shape signatures of tessellated solid models based on the area of the triangles. The latter method was found to have poor performance (measured by Precision-Recall) when the query and database models had several curved features. This poor performance was attributed to the larger number of false positives generated when comparing areas of triangles - two triangles can have different shapes, but identical areas. To overcome this weakness, the research presented in this paper investigates a method that computes similarity of tessellated solid models based on the congruence of the constituent triangles – the CTS method. This is evaluated on the Engineering Shape Benchmark (ESB) [12] to assess its suitability to engineering applications.

3 PROPOSED METHODOLOGY

Due to the nature of the similarity assessment, its function hinges on the necessity of all database and query files being of the same tessellation resolution (see Figure 2). This requirement makes the proposed approach more suited to applications within an enterprise where tessellation procedures can be controlled, as opposed to applications on open databases where STL file resolution is not standardized. The solid model similarity assessment method presented in this research uses the STL file format, and can be seamlessly extended to any file format that uses triangular mesh representation. Specifically, the side lengths of each tessellation are calculated in order to build a profile of every triangle that comprises the model. These side lengths are then used to generate a shape signature for each solid model, and subsequently used to compare the number of occurrences of congruent triangles between solid models. The complexity of the algorithm, for a given pair of solid models is O (N^2). Pseudo code for this can be seen below.

1. For the query STL file
   a. Calculate side lengths for every triangle
   b. Round side lengths to one decimal place
   c. Calculate the number of occurrences of each, unique triangle
2. Repeat Step 1 for all database STL files
3. Determine similarity by using Equation (3.1)
4. Normalize similarity scores using Equation (3.2)

$$Similarity = \frac{Number\ of\ congruent\ triangles\ across\ both\ models}{Total\ number\ of\ triangles\ across\ both\ models} \tag{3.1}$$

$$Normalized\ S = \frac{S_d - Min\ S}{Max\ S - Min\ S} \qquad \forall\ d \in D \tag{3.2}$$

$Where:$
$d = database\ model$
$S = similarity\ score$
$D = database\ of\ all\ solid\ models\ being\ compared$

The check for congruence is performed by creating an array of lists. One array is created for the query model and another is created for the database model. Each list has three elements - the three side lengths of a triangle, arranged in ascending order of their magnitude. If a list of ordered side lengths is found in both, the query and the database model, then the difference in their computed.

Normalization of similarity scores occurs following their calculation using Equation (3.2). In this process, each database model is assigned a number from 0 to 1 (inclusive), based on its similarity to the query model, where 1 indicates most similar and 0 indicates least similar. The query model is included in the database set to ensure that 1 indicates an identical match.

The Congruent Triangle Similarity (CTS) method is scale-sensitive by design due to its intended application in engineering settings. For instance, let's consider a case where the CTS method is used to find similar screws to a query screw model for the purpose of assembly instruction retrieval [25]. In this case, scale sensitivity is highly desired because a 1in screw will have significantly different assembly process than a 10in screw.

The CTS method was evaluated using the Engineering Shape Benchmark (ESB) [12]. Jayanti and colleagues have developed this benchmark specifically for engineering applications, and they have also provided Precision-Recall curves showing the performance of solid model similarity techniques from literature. The latter allows the CTS method to be objectively compared with the performance of pre-existing methods.

In terms of testing, the sensitivity of the CTS method to two parameters was evaluated: (1) Resolution of the tessellations for all files; (2) Number of decimal places used to compute side lengths. All 867 models of the Engineering Shape Benchmark (ESB) were modified using a quadric edge collapse tool in Meshlab [2] to adjust the resolution of the files. This process yielded ten sets of ESB files, one at 100% resolution and each subsequent version decremented by a 10% reduction of resolution. The reduction in resolution is reducing the number of tessellations that comprise each solid model. For example, reducing a model which was originally comprised of 100 tessellations by 50% would result in a model with 50 tessellations. A visualization of the reduction of tessellations can be seen in Figure 3. It must be noted that ten sets of ESB files were generated, and testing was performed within (and not across) these sets.

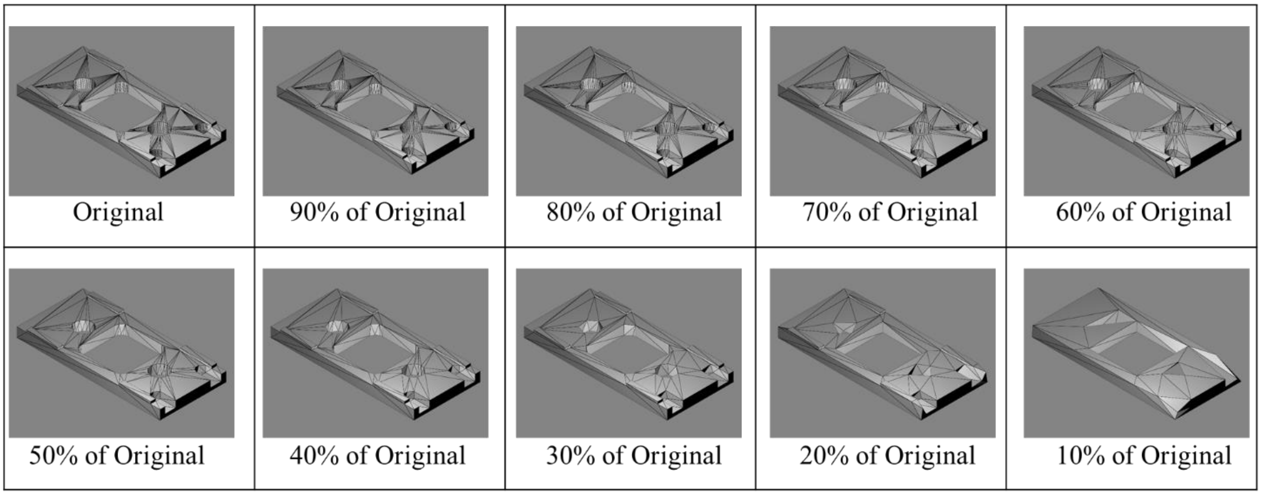


Figure 3: Solid model from the ESB with tessellation overlay alongside reduced resolution versions of the original.

4 TESTING AND EVALUATION

4.1 Engineering Shape Benchmark

The ESB is separated into three main categories "Flat-thin walled parts," "Rectangular-cubic prism parts," and "Solids of revolution." Each main category consists of a number of subcategories within which solid models are placed [12]. As mentioned earlier, ten sets of ESB files were generated, where nine sets are obtained by reducing STL resolutions in steps of ten (90%, 80% … 10% resolution) and one set is the original ESB (100% resolution). Of these ten sets, four were used for sensitivity analysis – 10%, 40%, 70%, and 100%. A sensitivity analysis was also performed on the number of decimal places used to compute triangle side lengths. Performance (defined subsequently) was evaluated when the following number of decimal places were used: 1, 2, 3, 4, 9, 10, 11, and 12. For each resolution set and each decimal rounding place, an M-by-M matrix was generated by treating each ESB file as a query and determining its similarity to all other files from the ESB – resulting in 24,054,048 comparisons. Performance of the CTS was assessed using the following two information retrieval metrics.

4.2 Precision-Recall

Performance was quantified by using Precision-Recall curves (constructed using the NIST TREC standards [8] to stay consistent with Jayanti and colleagues' work [12]). Precision is the fraction of documents retrieved that are relevant to what is queried (Equation (4.1)), and recall is the fraction of total relevant documents that were retrieved (Equation (4.2)). It must be noted that a database solid model is deemed relevant if it is the same subcategory as that of the query model (as defined in [12]). The precision values at various levels of recall (10%, 20%… 100% of relevant documents) are calculated and plotted to result in a Precision vs. Recall (PR) curve. Pre-existing solid model similarity assessment methods from literature have been benchmarked using PR curves on the ESB models. These solid model similarity methods include: Light Field Descriptors, Convex Hull Histogram, 2.5D Spherical Harmonics, and Moment invariants. A comprehensive list and description of all methods can be found in [12]. A further discussion of the comparison between PR curves from the CTS method and PR curves from literature can be seen in the Results and Discussion section.

$$Precision = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ documents\ retrieved} \tag{4.1}$$

$$Recall = \frac{Number\ of\ relevant\ documents\ retrieved}{Total\ number\ of\ relevant\ documents} \tag{4.2}$$

4.3 Precision at Specified Retrieval Sizes

Precision is calculated for a predetermined size of retrieval. This metric is relevant due to the intended application of the CTS method for search and retrieval of solid models from a database. In context of search engines, this metric is used to determine the fraction of the first *n* results that are relevant.

5 RESULTS AND DISCUSSION

The results for the sensitivity analysis are presented first, along with a discussion of a recommendation for number of decimals and resolution to be used. Next, PR curves are constructed using the recommended parameters and superimposed on those from Jayanti and colleagues [12], and therefore provide a qualitative comparison of the CTS method with methods from literature.

Precision at retrieval size of five was used as the metric to determine "performance" as this mimics a search engine that provides five results on its first page. Performance was analyzed for each combination of the four levels of resolutions and eight decimal places. The performance was found to have positive correlation (Pearson's coefficient, r = 0.34) to resolution of STL files used, and negative correlation (Pearson's coefficient, r = -0.40) to number of decimal places used to compute side lengths. A plot of the performance versus decimals and resolutions (Figure 4) shows that performance peaks at low decimal places before it lowers and plateaus at higher decimal places. This behavior is explained as follows. Reducing the number of decimals used to compute the triangle side lengths increases the probability of two triangles being deemed congruent. This, in turn, increases the probability of having a larger numerator in Equation (3.1), and thus having a higher similarity score. This is evidenced by the density of the M-by-M matrices for various decimal places (see Figure 5). The difference between matrix density for one decimal place and matrix density for two decimal places is large. This observation coupled with the high performance of solid models at 70% resolution leads to the following parameters being recommend and used to compute CTS: a medium tessellation resolution (70%) along with the use of one decimal place to compute side lengths.

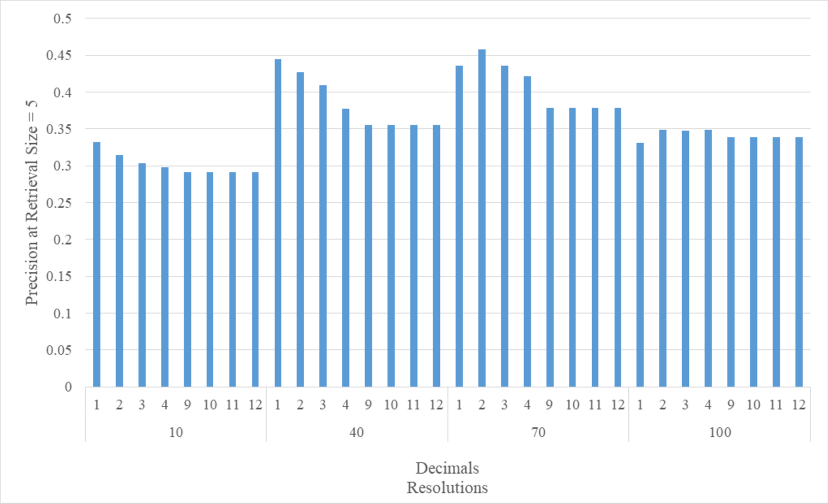


Figure 4: Average Precision for Various Decimals and Resolution Combinations

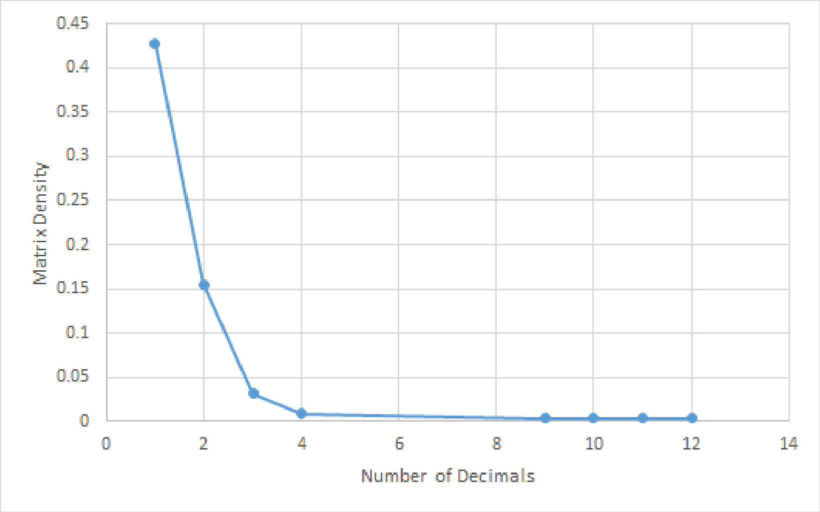


Figure 5: Sparsity of Similarity Score Matrix Significantly Reduces when Number of Decimals Increase.

Using these recommendations, PR curves were constructed for the ESB subcategories (see Figure 6, Figure 7, and Figure 8). Select curves were then superimposed on all available PR curves from [12], and the resulting data is presented in Figure 9. The proposed method's performance was subpar in a few categories. This was further investigated, and a primary contributing factor was found – false negatives. A retrieved item could be deemed 'irrelevant' based on the subcategory it was classified into [12] although it was geometrically similar (and thus, 'relevant') to the query.

The results from the aforementioned evaluation in conjunction with basic requirement of tessellated solid model representations (and no requirement of BREP data) makes the CTS method viable to be used for search and retrieval of engineering solid models. Figure 16 shows examples of query and retrieved solid models using the recommended parameters for CTS. All PR curves, and search and retrieval examples can also be found at the following website: https://github.com/rahulrenu/CongruentTriangleSimilarity

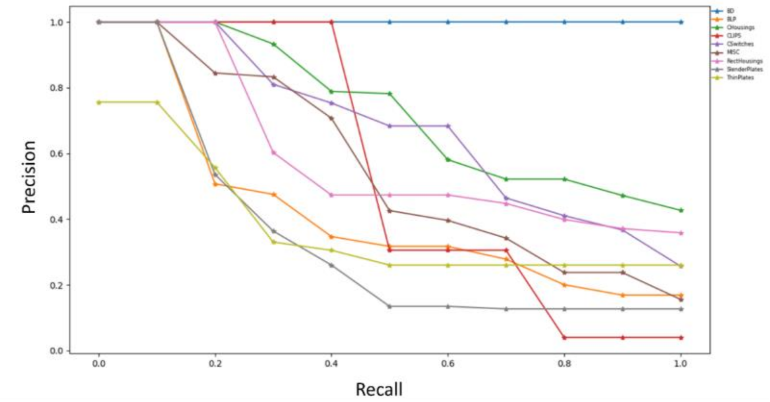


Figure 6: PR Curves for Flat Thin Walled Components Using CTS (70% Resolution, 1 Decimal Place).

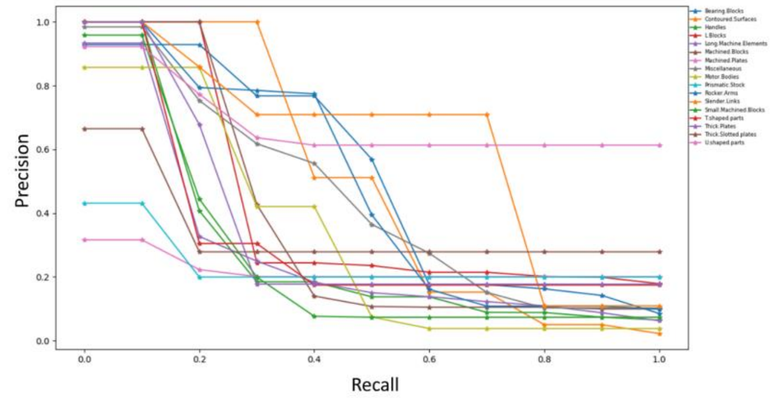


Figure 7: PR Curves for Rectangular Cubic Prisms Using CTS (70% Resolution, 1 Decimal Place).

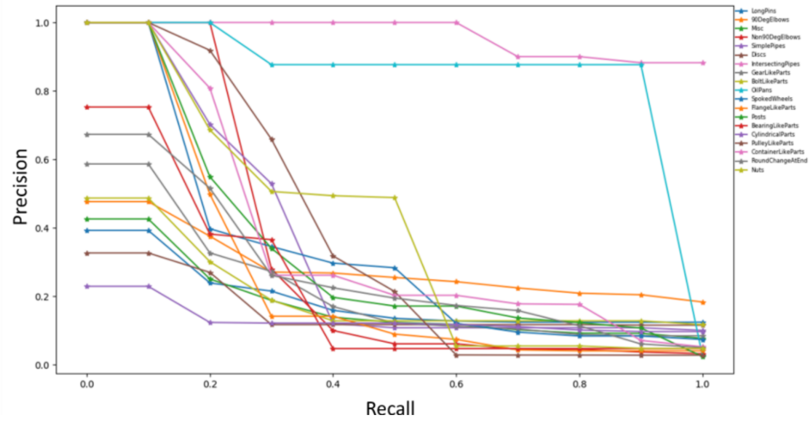
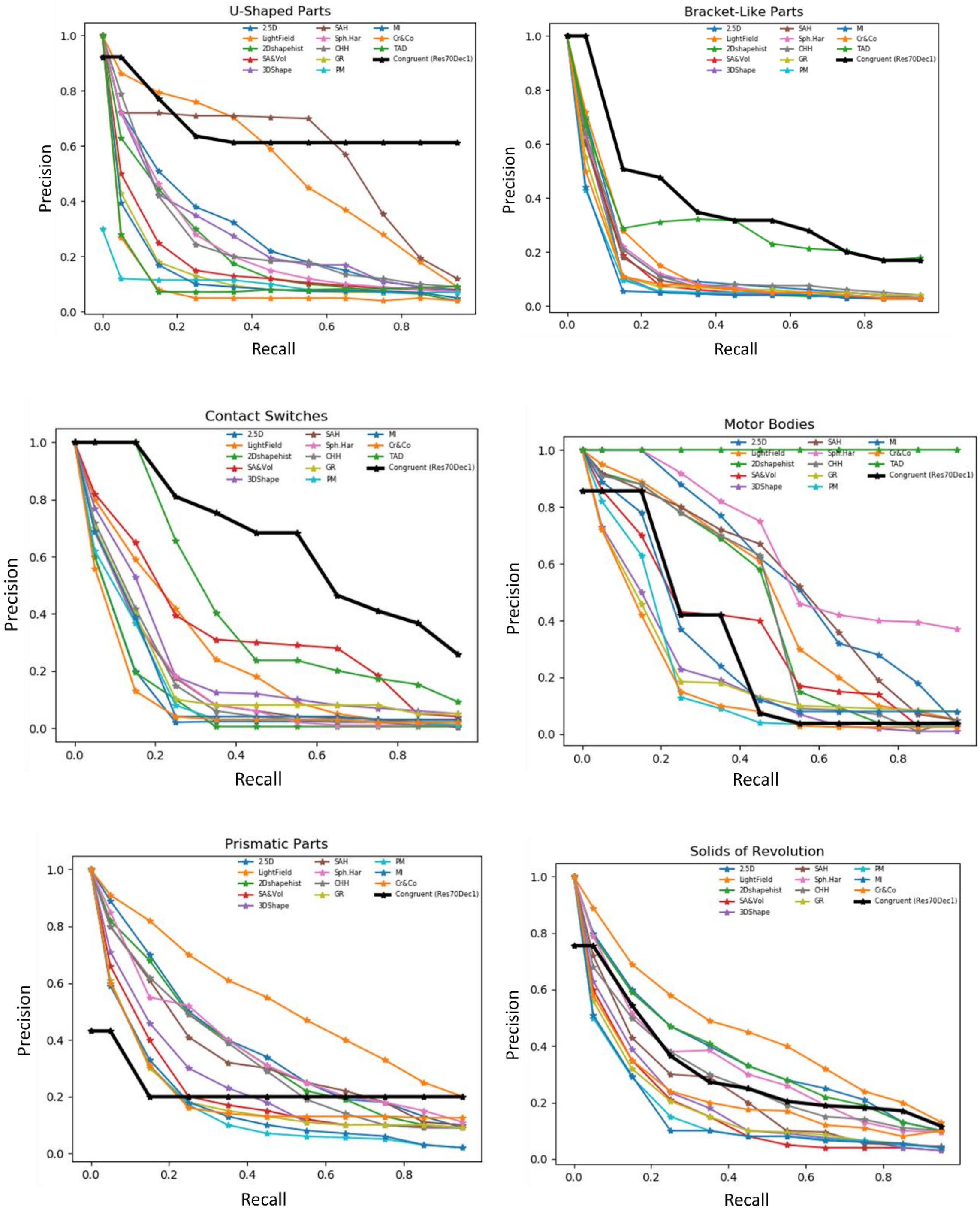


Figure 8: PR Curves for Solids of Revolution Using CTS (70% Resolution, 1 Decimal Place)


U-Shaped Parts
Bracket-Like Parts
Contact Switches
Motor Bodies
Prismatic Parts
Solids of Revolution
2.5D
LightField
2Dshapehist
SA&Vol
3DShape
SAH
Sph.Har
CHH
GR
PM
MI
Cr&Co
TAD
Congruent (Res70Dec1)
Precision
Recall
0.0
0.2
0.4
0.6
0.8
1.0

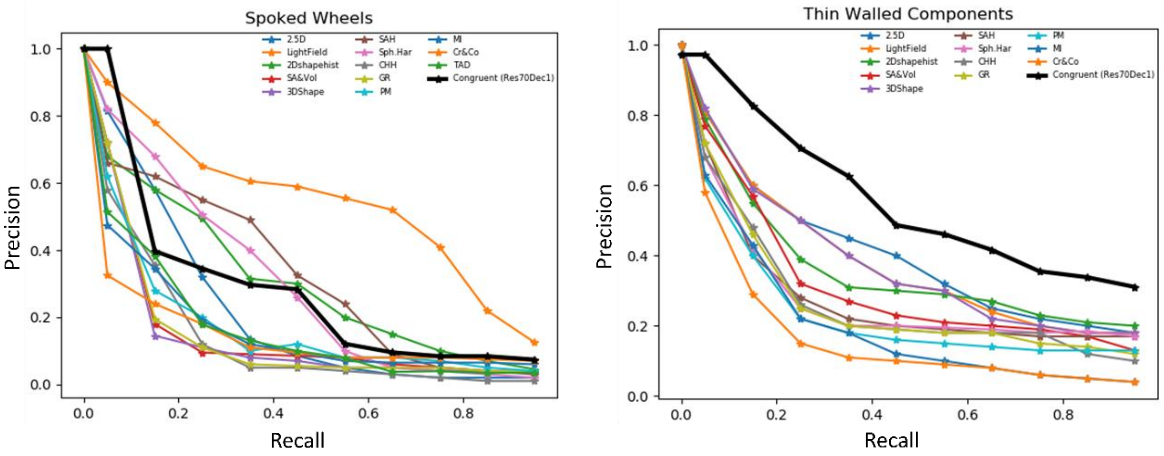


Figure 9: PR Curves Comparing CTS and Methods from Literature.

6 CONCLUSIONS

The ever-increasing use of solid models in the engineering domain (industrial and household) has resulted in large databases of solid models becoming available to engineers and additive manufacturing enthusiasts alike. Research has established that reusing existing designs from these databases has benefits that include: (1) Reduction of design iterations performed, thus reducing product realization lead times; (2) Increased probability of using existing manufacturing capabilities (and not having to invest in new manufacturing machines); (3) Enabled reuse of institutional knowledge. A natural mechanism that allows the reuse of existing designs is solid model similarity. Existing solid model similarity assessment methods either require more information than just mesh information (such as B-Rep data), or have high computational complexity, or use black box approaches. To overcome these deficiencies, in this work, the CTS method of computing solid model similarity is presented. The method is driven by Equation (3.1), is completely transparent, and works on triangularly tessellated files thus supporting solid models developed for additive manufacturing (see Figure 10). Further examples come from testing the method on crowd-sourced STL files [31].

A thorough sensitivity analysis is performed to evaluate the effect of two parameters on the performance of CTS: (1) Number of decimals used to compute side lengths (2) Resolution of STL files used. Through nearly 24 million comparisons, it is found that the CTS methods performs best when one decimal place is used to compute side lengths and even when highest resolution/fidelity STLs are not available. The CTS method is also compared to methods from literature by superimposing the former's PR curves on the latter. From these curves, it is seen that in all but one ESB subcategory (Prismatic Parts), the CTS outperforms most methods from literature.

The work presented in this paper must be carried forward and employed within industry to assess its efficacy as a design reuse mechanism. In addition, this work must be extended to enable the search and retrieval of assembly files (comprising of several individual part files).

Christopher L. Sousa, http://orcid.org/0000-0002-6502-2513
Rahul S. Renu, http://orcid.org/0000-0001-8690-7268

REFERENCES

[1] Cardone, A.; Gupta, S. K.; Karnik, M.: A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications, Journal of Computing and Information Science in Engineering, 3(2), 2003, 109. https://doi.org/10.1115/1.1577356

[2] Cignoni, P.; Callieri, M.; Corsini, M.; Dellepiane, M.; Ganovelli, F.; Ranzuglia, G.: MeshLab: An Open-Source Mesh Processing Tool, 6th Eurographics Italian Chapter Conference 2008 - Proceedings, 2008.

[3] Elinson, A.; Nau, D. S.; Regli, W. C.: Feature-Based Similarity Assessment of Solid Models, Proceedings of the fourth ACM symposium on Solid modeling and applications, ACM, 1997. https://doi.org/10.1145/267734.267806

[4] Hartley, J. R.: Concurrent Engineering: Shortening Lead Times, Raising Quality, and Lowering Costs, 1992.

[5] Henderson, M. R.; Musti, S.: Automated Group Technology Part Coding from a Three-Dimensional CAD Database, Journal of Engineering for Industry, 110(3), 1988, 278–287. https://doi.org/10.1115/1.3187882

[6] Hiller, J. D.; Lipson, H.: STL 2.0: A Proposal for a Universal Multi-Material Additive Manufacturing File Format, Solid Freeform Fabrication Symposium, (1), 2009, pp. 266–278.

[7] Hong, T.; Lee, K.; Kim, S.: Similarity Comparison of Mechanical Parts to Reuse Existing Designs, Computer-Aided Design, 38(9), 2006, pp. 973–984.

[8] http://trec.nist.gov/pubs/trec13/appendices/CE.MEASURES.pdf: *Proceedings of the Thirteenth Text Retrieval Conference*.

[9] Huang, R.; Zhang, S.; Bai, X.; Xu, C.; Huang, B.: An Effective Subpart Retrieval Approach of 3D CAD Models for Manufacturing Process Reuse, Computers in Industry, 67**,** 2015, 38–53. https://doi.org/10.1016/j.compind.2014.12.001

[10] Iyer, N.; Jayanti, S.; Lou, K.; Kalyanaraman, Y.; Ramani, K.: Three-Dimensional Shape Searching: State-of-the-Art Review and Future Trends, Computer-Aided Design, 37(5), 2005, 509–530. https://doi.org/10.1016/j.cad.2004.07.002

[11] Iyer, N.; Jayanti, S.; Lou, K.; Kalyanaraman, Y.; Ramani, K.: Shape-Based Searching for Product Lifecycle Applications, Computer-Aided Design, 37(13), 2005, 1435–1446. https://doi.org/10.1016/j.cad.2005.02.011

[12] Jayanti, S.; Kalyanaraman, Y.; Iyer, N.; Ramani, K.: Developing an Engineering Shape Benchmark for CAD Models, Computer-Aided Design, 38(9), 2006, 939–953. https://doi.org/10.1016/j.cad.2006.06.007

[13] Jin, Y.; Curran, R.; Butterfield, J.; Burke, R.; Welch, B.: Automated Assembly Time Analysis Using a Digital Knowledge Based Approach 1,3, The 26th Congress of International Council of the Aeronautical Sciences, 2008, 14–19. https://doi.org/10.2514/6.2008-8861

[14] Kusiak, A.: The Generalized Group Technology Concept, International Journal of Production Research, 25(4), 1987. https://doi.org/10.1080/00207548708919861

[15] Li, Z.; Zhou, X.; Liu, W.: A Geometric Reasoning Approach to Hierarchical Representation for B-Rep Model Retrieval, CAD Computer Aided Design, 62, 2015, 190–202. https://doi.org/10.1016/j.cad.2014.05.008

[16] Lupinetti, K.; Chiang, L.; Giannini, F.; Monti, M.; Pernot, J.-P.: Regular Patterns of Repeated Elements in CAD Assembly Model Retrieval, Computer-Aided Design and Applications, 14(4), 2017, 516–525. https://doi.org/10.1080/16864360.2016.1257193

[17] Lupinetti, K.; Giannini, F.; Monti, M.; Pernot, J. P.: Content-Based Multi-Criteria Similarity Assessment of CAD Assembly Models, Computers in Industry, 112**,** 2019. https://doi.org/10.1016/j.compind.2019.07.001

[18] Manns, M.; Wallis, R.; Deuse, J.: Automatic Proposal of Assembly Work Plans with a Controlled Natural Language, Procedia CIRP, 33**,** 2015, 346–351. https://doi.org/10.1016/j.procir.2015.06.079

[19] McWherter, D.; Peabody, M.; Regli, W. C.; Shokoufandeh, A.: Solid Model Databases: Techniques and Empirical Results, Journal of Computing and Information Science in Engineering, 1(4), 2001, 300. https://doi.org/10.1115/1.1430233

[20] Miller, M.; Griese, D.; Peterson, M.; Summers, J. D.; Mocko, G. M.: Installation Process Step Instructions as an Automated Assembly Time Estimation Tool, ASME International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, 2012, 1–8. https://doi.org/10.1115/DETC2012-70109

[21] Namouz, E.: Automated Complexity Based Assembly Time Estimation Method, Thesis, 2013.
[22] Osada, R.; Funkhouser, T.; Chazelle, B.; Dobkin, D.: Shape Distributions, ACM Transactions on Graphics, 21(4), 2002, 807–832. https://doi.org/10.1145/571647.571648
[23] Ramesh, M.; Yip-Hoi, D.; Dutta, D.: Feature Based Shape Similarity Measurement for Retrieval of Mechanical Parts, Journal of Computing and Information Science in Engineering, 1(3), 2001, 245. https://doi.org/10.1115/1.1412456
[24] Renu, R.: Product-Process Coupling to Enable Continuous Improvement of Assembly Processes, Dissertation, Clemson University, 2016.
[25] Renu, R. S.; Mocko, G.: Computing Similarity of Text-Based Assembly Processes for Knowledge Retrieval and Reuse, Journal of Manufacturing Systems, 39, 2016, 101–110. https://doi.org/10.1016/j.jmsy.2016.03.004
[26] Rychtyckyj, N.: Intelligent Systems for Manufacturing at Ford Motor Company, IEEE Intelligent Systems, 2007. https://doi.org/10.1109/MIS.2007.13
[27] Rychtyckyj, N.; Klampfl, E.; Rossi, G.: Application of Intelligent Methods to Automotive Assembly Planning, 2007 IEEE International Conference on Systems, Man and Cybernetics, 2007, 2479–2483. https://doi.org/10.1109/ICSMC.2007.4414163
[28] Stroud, I.: Boundary Representation Modelling Techniques, 2006.
[29] Tao, S.; Huang, Z.; Ma, L.; Guo, S.; Wang, S.; Xie, Y.: Partial Retrieval of CAD Models Based on Local Surface Region Decomposition, CAD Computer Aided Design, 45(11), 2013, 1239–1252. https://doi.org/10.1016/j.cad.2013.05.008
[30] Thomke, S.; Fujimoto, T.: The Effect of 'Front-Loading' Problem-Solving on Product Development Performance, Journal of Production Innovation Management, 17(2), 2000, pp. 128–142. https://doi.org/10.1111/1540-5885.1720128
[31] Zhou, Q.; Jacobson, A.: Thingi10K: A Dataset of 10,000 3D-Printing Models, arXiv preprint arXiv:1605.04797, 2016.

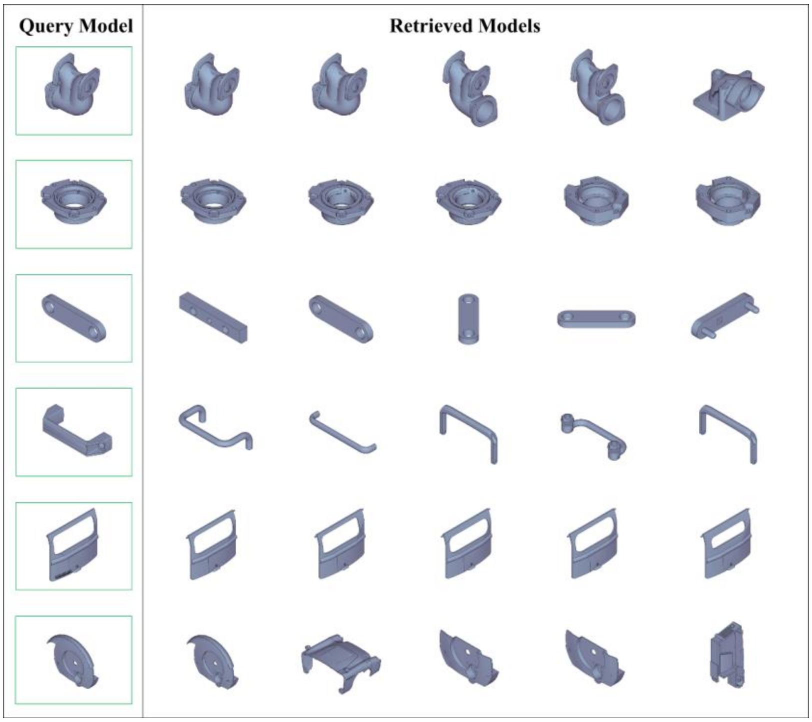


Figure 10: Example Search and Retrieval Performed Using CTS and Query-by-Model.

7 APPENDIX A

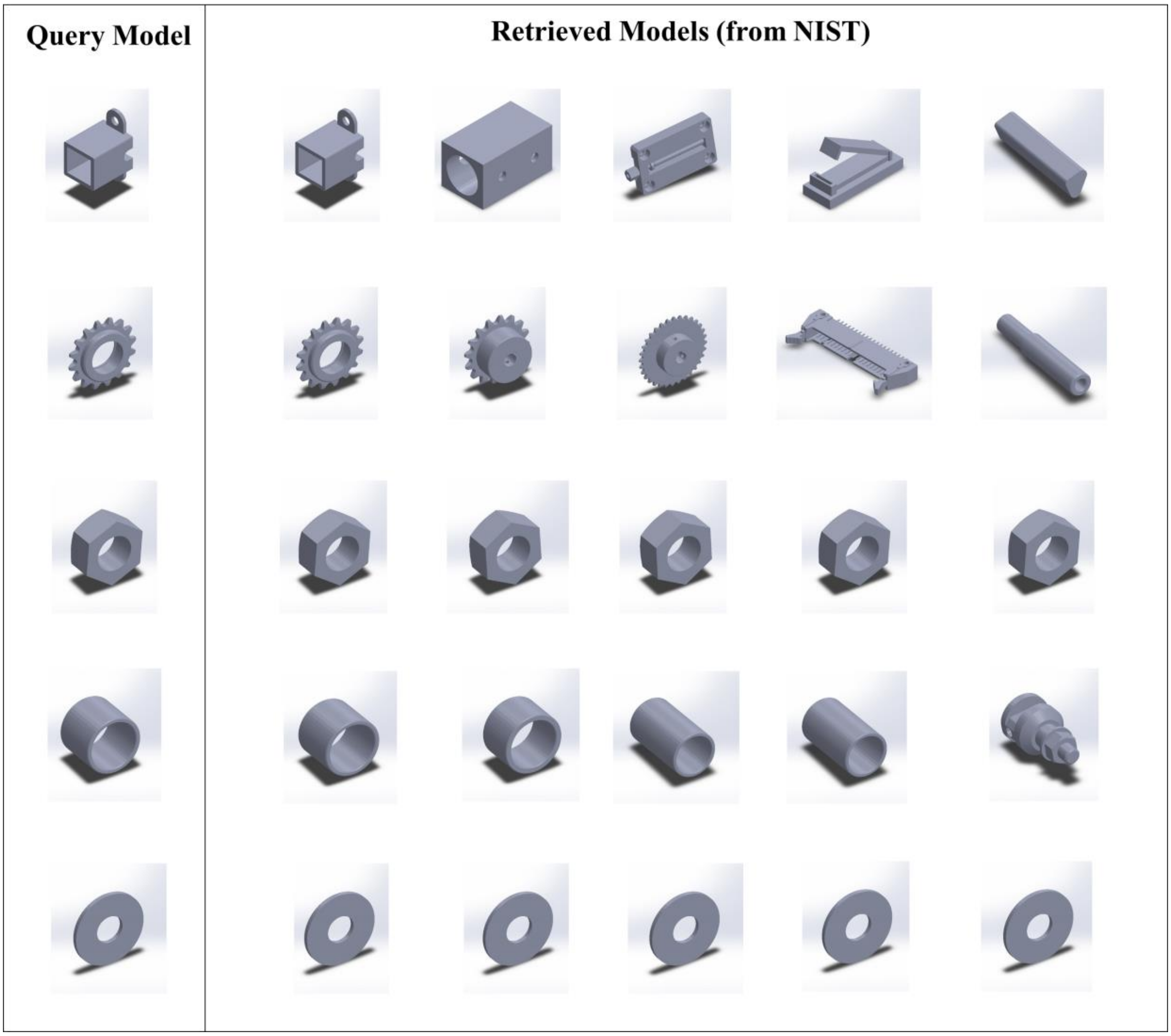


Figure 11: Search and Retrieval Examples from NIST[1,2]

1. https://www.nist.gov/system/files/documents/2019/09/27/belt_dive_stl_files.zip
2. https://www.nist.gov/system/files/documents/2017/08/08/task2_stl_files.zip

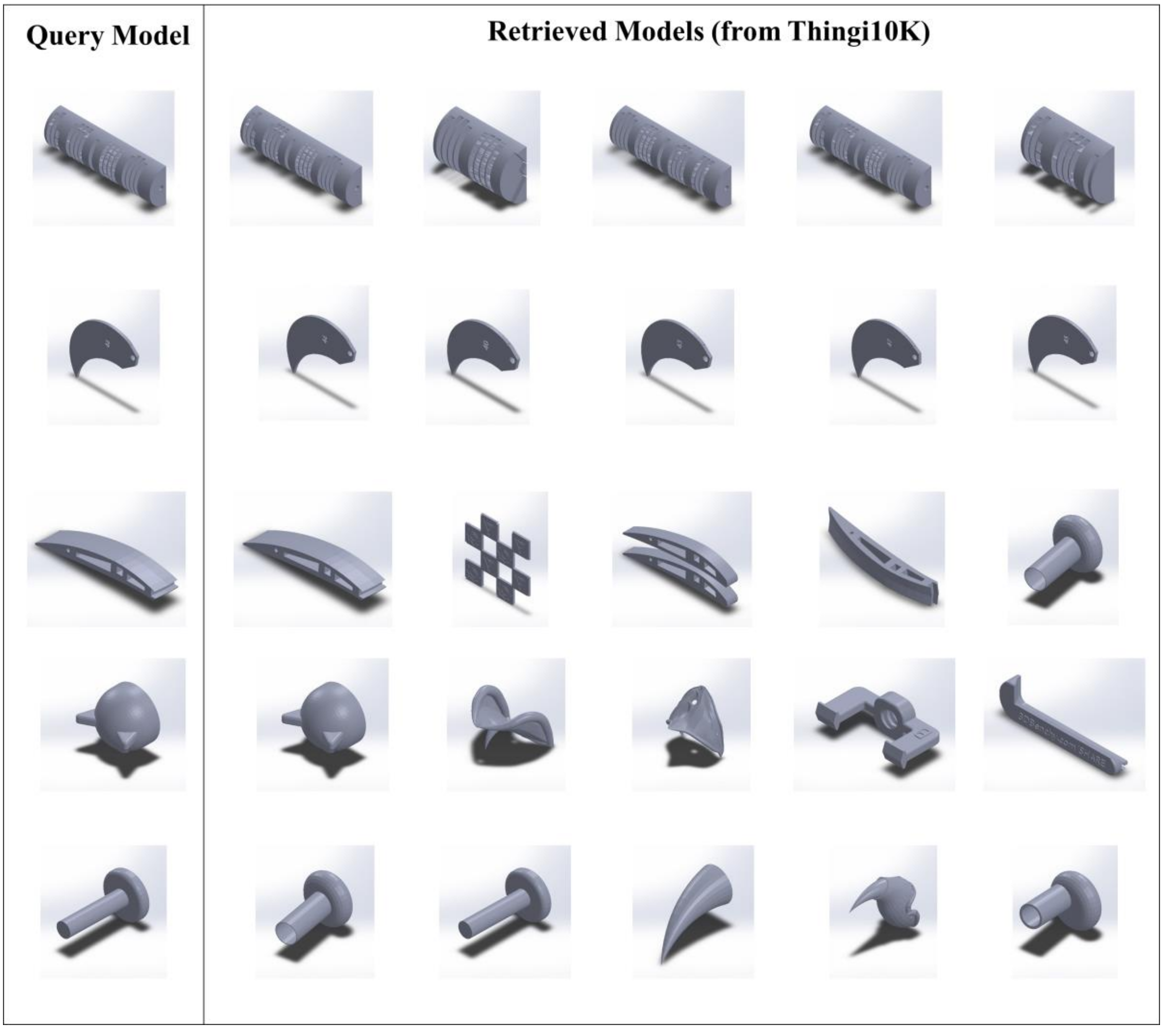


Figure 12: Search and Retrieval Examples from Thingi10K.