Title:
**Solid Model Similarity for Engineering Applications using Congruence of Triangles**

Authors:
Christopher Sousa, clsousa@g.clemson.edu, Clemson University
Rahul S. Renu, rrenu@fmarion.edu, Francis Marion University

Keywords:
Solid Model Similarity, Information Retrieval, CAD Similarity, Design Reuse

DOI: 10.14733/cadconfP.2020.66-70

Introduction:

The ability to accurately recall existing solid models from a database in order to reuse 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, 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 (stemming from language differences and synonym use), and requires immense efforts to create and assign appropriate tags. Therefore, query-by-solid-model (see Fig. 1) is a preferred approach.

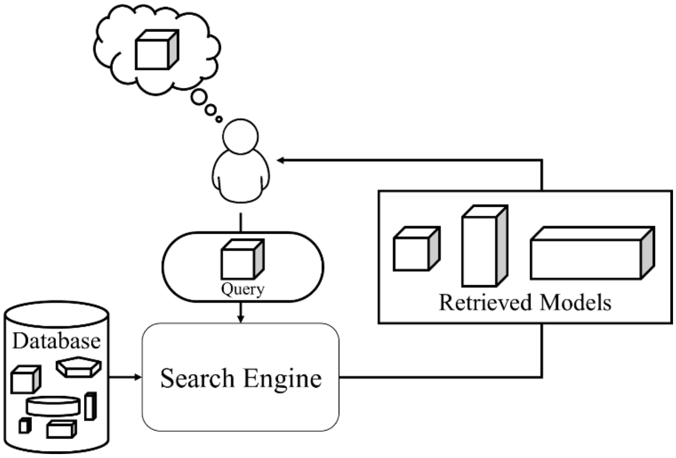


Fig. 1: Query-by-Solid-Model Approach.

The work presented in this paper compares triangular mesh solid models based on the congruence of their constituent triangles. The rationale and motivation for this research stems from the fact that triangles are ubiquitous, elementary building blocks of three-dimensional shapes. 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.

<u>Main Idea:</u>
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. 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 individual side lengths of each tessellation are calculated in order to build a profile of every triangle that comprises the model. These side length triplets 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. Two triangles are said to be congruent if all three sides are equal. The complexity of the algorithm, for a given pair of solid models is O (N). 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 1.
4. Normalize similarity scores using Equation 2.
```

$$Similarity = \frac{Number\ of\ congruent\ triangles\ across\ both\ models}{Total\ number\ of\ triangles\ across\ both\ models} \tag{1}$$

$$Normalized\ S = \frac{S_d - Min\ S}{Max\ S - Min\ S} \qquad \forall\, d \in D \tag{2}$$

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

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 [3]. 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) [2]. Jayanti and colleagues [2] 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 [1] 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 (
Fig. 2). It must be noted that ten sets of ESB files were generated, and testing was performed within (and not across) these sets.

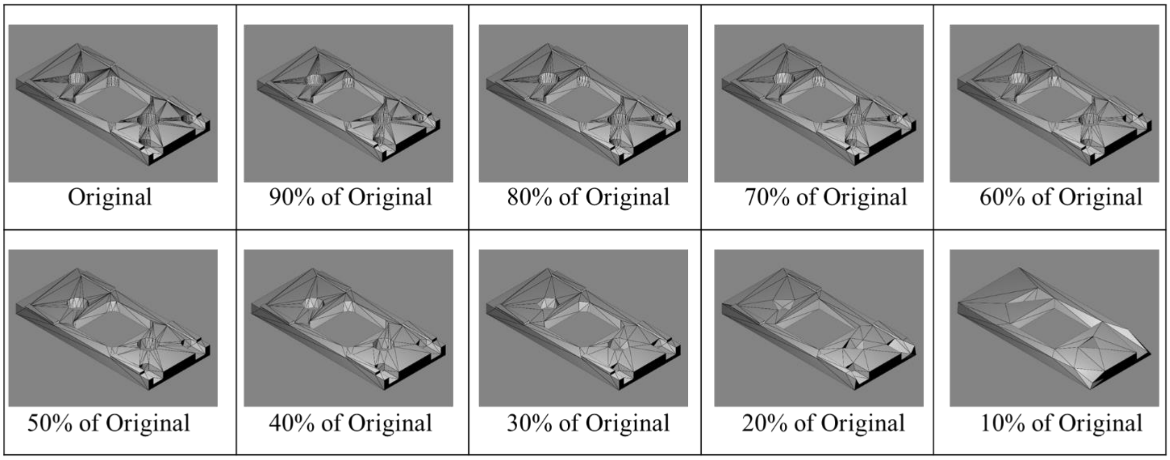


Fig. 2: Example of Reduced Resolution STL File.

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 Precision-Recall and Precision at Retrieval Size of Five.

<u>Conclusions:</u>

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. It is found 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 (1), and thus having a higher similarity score. It was also found that performance shares an inverse-U relationship with resolutions. A medium tessellation resolution (70%) is therefore recommended along with the use of one decimal place to compute side lengths.

Using these recommendations, PR curves were constructed for the ESB subcategories. Select curves were then superimposed on all available PR curves from [2], and the resulting data is presented in Fig. 3. It is seen that, except in the case of Prismatic Parts, the CTS method outperforms most methods from literature. This evaluation in conjunction with (1) basic requirement of tessellated solid model representations (and no requirement of BREP data), and (2) low complexity of the algorithm (O(N)) makes the CTS method viable to be used for search and retrieval of engineering solid models.

Fig. 4 shows examples of query and retrieved solid models using the recommended parameters for CTS. All PR curves, and additional retrieval examples can also be found at the following website: https://github.com/rahulrenu/CongruentTriangleSimilarity

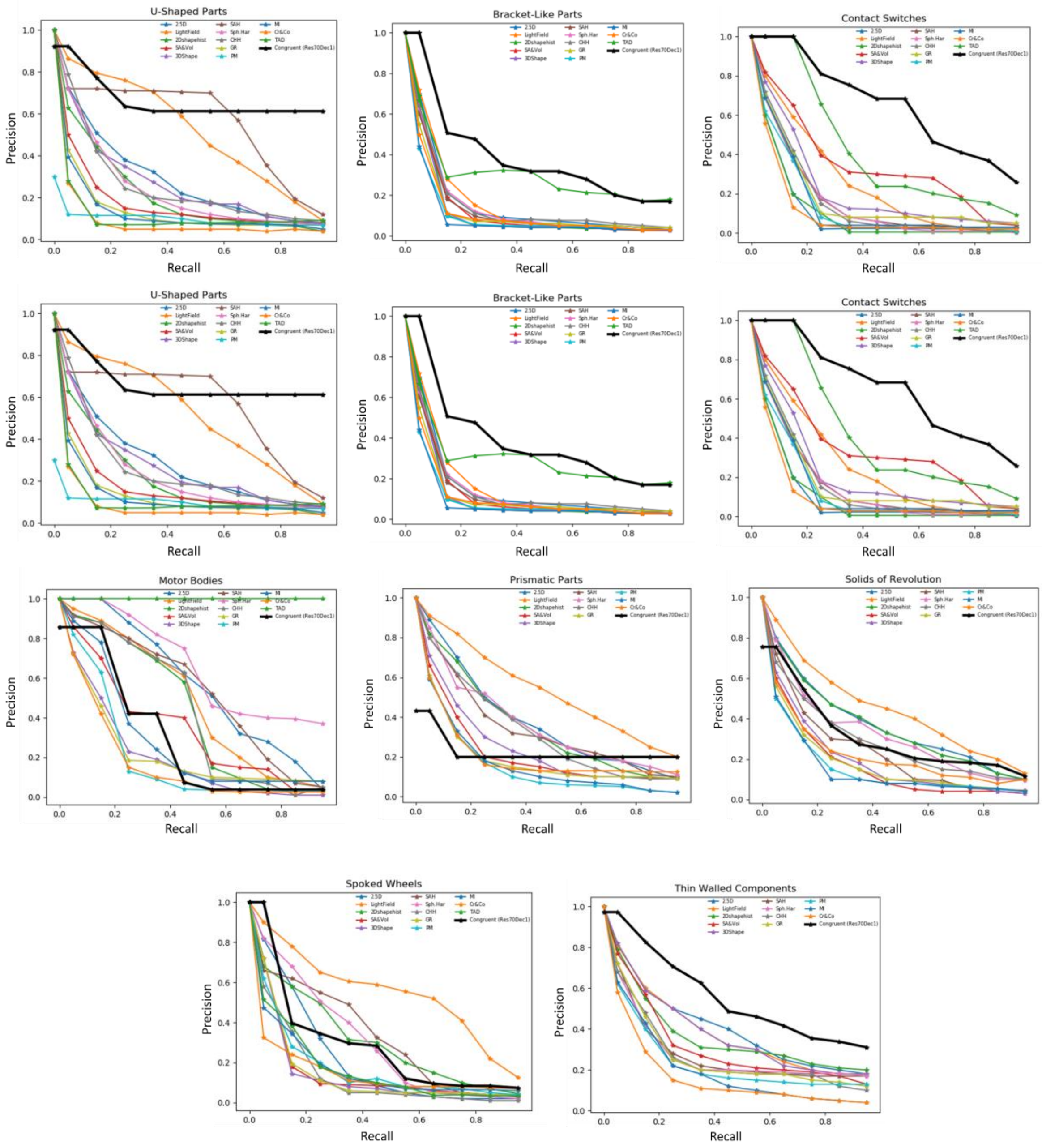


Fig. 3: PR Curves Comparing CTS and Methods from Literature.

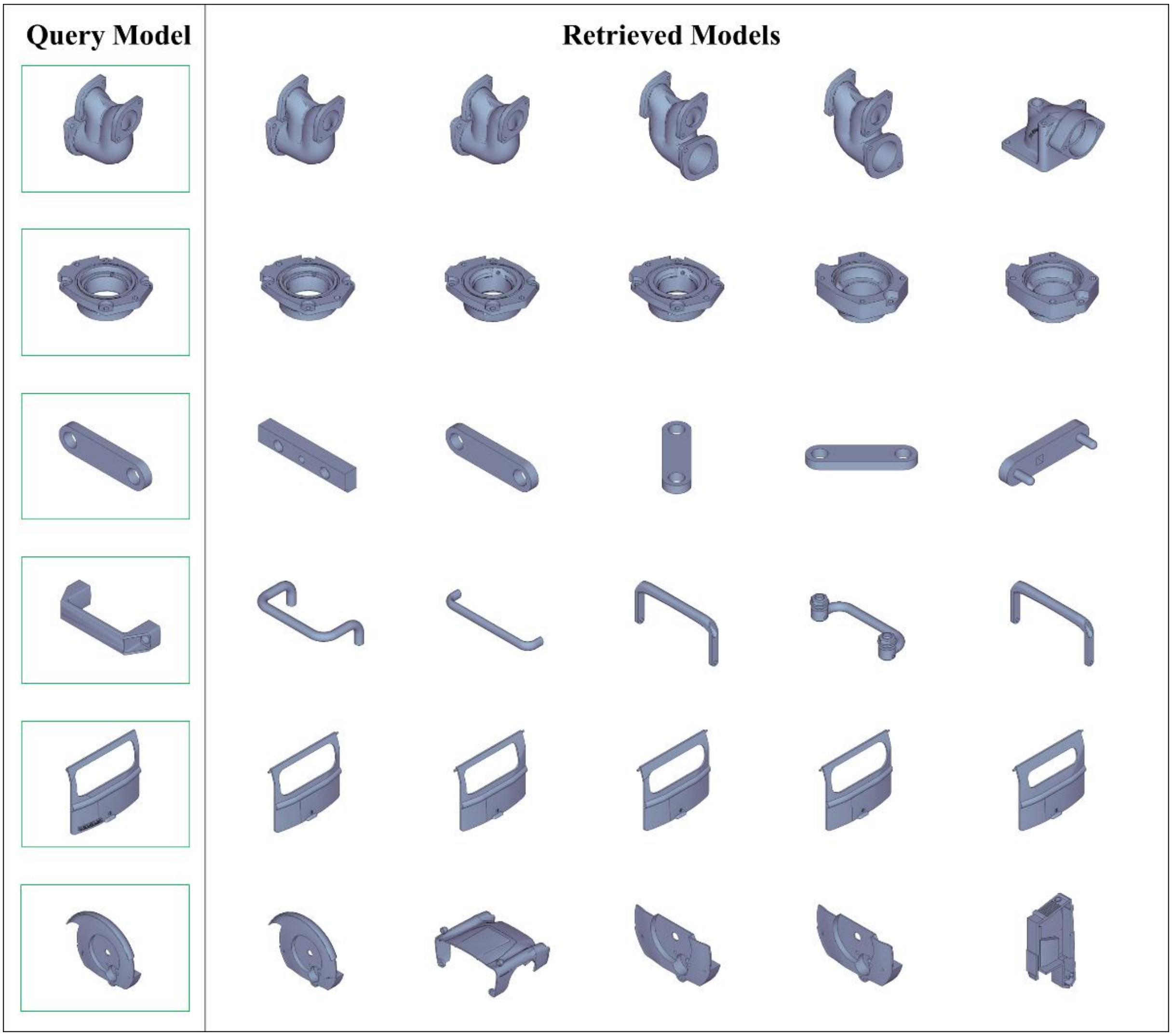


Fig. 4: Example Search and Retrieval Performed Using CTS and Query-by-Model.

References:

[1] 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.

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

[3] Renu, R.; Mocko, G.: Retrieval of Solid Models Based on Assembly Similarity, Computer Aided Design and Applications, 13(5), 2016, pp. 628–636. https://doi.org/10.1080/16864360.2016.1150708