Shape Similarity Matching With Octree Representations

Jingsheng Zhang
Human-Computer Interaction Graduate Program,
Iowa State University,
Ames, IA 50011
e-mail: jszhang@iastate.edu

Shana Smith[1]
Department of Mechanical Engineering,
National Taiwan University,
No. 1, Sec. 4, Roosevelt Road,
Taipei, 10617 Taiwan (R.O.C.)
e-mail: ssmith@ntu.edu.tw

To achieve effective 3D shape retrieval, there is a crucial need for efficient shape matching methods. This paper introduces a new method for 3D shape matching, which uses a simplified octree representation of 3D mesh models. The simplified octree representation was developed to improve time and space efficiency over prior representations. The proposed method also stores octree information in extensible markup language format, rather than in a new proprietary data file type, to facilitate comparing models over the Internet. [DOI: 10.1115/1.3197846]

1 Introduction

There are many applications in which storing, retrieving, and reusing 3D information can significantly enhance performance [1–5]. Manually retrieving existing parts is a costly and time-consuming task. Designers typically can spend as much as 60% of their time simply searching for the right information [6]. As a result, a 3D computer-aided design (CAD) shape retrieval system is an important key to reducing design time and allows engineers to spend more time on the creative aspects of their jobs.

1.1 Representations of 3D Objects and Similarity Measures. In general, shape retrieval systems use either object-based or image-based 3D shape representations of objects [7,8]. Early examples of object-based representations include generalized cones, which were introduced by Marr and Nishihara [9], and geon structural descriptions, which were proposed by Biederman [10,11]. Another commonly used object-based representation is based on surface descriptions, which are built on object vertices, edges, and surfaces, in conjunction with their connection relations [12,13]. In general, object-based representations are useful for feature queries. However, when responding to location queries, they require examination of the entire objects.

Image-based representations contain collections of images for different perspectives from which the object can appear to a viewer. The stored views may be either 3D or 2D [14]. Image-based representations are useful for finding objects associated with a particular location or cell (i.e., location queries). However, they require that all cells be examined when determining the location associated with a particular object (i.e., a feature query).

In shape matching applications, object-based representations are sensitive to certain features, which might insignificantly contribute to the overall geometry of the 3D model, but they generally lead to tremendous computation times. Image-based representations usually require either a priori registration of the objects' coordinate systems or searching to find pairwise correspondences during matching [15]. Iyer et al. [16] provided an extensive overview and comparison of prior 3D shape searching methods.

Most feature extraction methods extract features of a shape into a single descriptor [15,17]. Feature extraction methods generally cost less computation time, but they show low stability for some classes of shapes [16] and are not sensitive to changes in feature locations [18]. On the other hand, graph-based methods attempt to extract topology information from a 3D shape and reduce the shape similarity problem to a graph or tree comparison problem [19,20]. In B-Reps, a shape is represented by a graph but large amounts of storage memory are required. Reeb graphs are simpler than B-Reps, but Reeb graph representations are affected by the choices of scalar functions and surface connectivities [16,21]. Skeletal graph-based techniques compute the medial surface for a 3D model as its skeleton and convert it into a skeletal graph as its shape descriptor [16,22–25]. The skeleton-based method requires less storage memory than a corresponding B-Rep. However, it is not easy to extract a skeleton from an arbitrary shape to create a robust, meaningful, and manageable descriptor.

The major advantage of graph-based methods is that they allow representations at multiple levels of detail, and they facilitate partial matching, as well as exact or inexact matching, depending on the noise. However, complex models may have large graphs, which increase comparison costs and dramatically affect efficiency [23]. Tree comparison tends to be fast and easy, but may often oversimplify a shape [16,18]. In this study, a solution to the representation problem was given, which addresses the drawbacks of existing graph-based methods. In particular, an efficient method for mapping a 3D model to a simplified octree was developed. The mapping not only simplifies the original 3D model representation, but also retains important information concerning both local features as well as the global structure of the model.

1.2 Octree Representations of 3D Objects. The classic octree subdivision process creates a tree, the root node that represents the complete 3D object. Leaf nodes are classified as either full or empty, depending on whether their corresponding octants are entirely within or outside the object, respectively. All nonleaf nodes are classified as partial. Most early work in octree encoding methods was mainly focused on B-Rep based solids [26–29]. The methods that were developed for polyhedral B-Rep solids involve elaborate computations, such as face-face intersections, and large numbers of 2D and 3D point classifications. As a result, they are very time-consuming. On the other hand, mesh models and corresponding file formats, such as VRML and 3DS, are now rapidly becoming standards for delivering 3D content over the Internet. Funkhouser et al. [15,30] did an extensive work in mesh modeling and feature extraction. Azernikov et al. [31] proposed grouping a point cloud into an octree for reconstructing the surface. Their approach is an application of classical octrees in 3D modeling for reverse engineering. Wang et al. [32,33] first used octrees to perform 3D model matching. In their method, they used ray tracing to detect object boundaries, which costs much computation time. They used a classical octree representation, in which they labeled and compared the octants as full, empty, or partial [32].

In this paper, we develop a new octree 3D object representation method, which can quickly decide whether to stop or continue octant subdivision. The developed method does not distinguish between outside and inside octants. Compared with a maximal division classical octree [31], our proposed simplified octree has only two classes of nodes: partial and empty. Since our method does not distinguish between outside and inside octants, the time needed for classifying and comparing octrees is substantially reduced. Computational complexity is also reduced by eliminating the need to identify if an octant is inside (full) or outside of an object's surface, with some surface recognition method, such as

[1]Corresponding author.

Contributed by the Computer Aided Product Development (CAPD) Committee of ASME for publication in the Journal of Computing and Information Science in Engineering. Manuscript received October 26, 2007; final manuscript received January 3, 2009; published online August 21, 2009. Editor: S. Gupta.

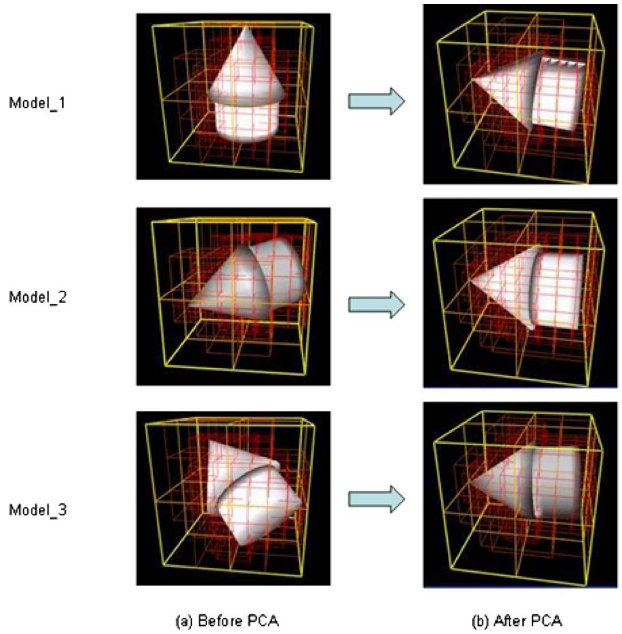

Fig. 1 Example of normalization by PCA

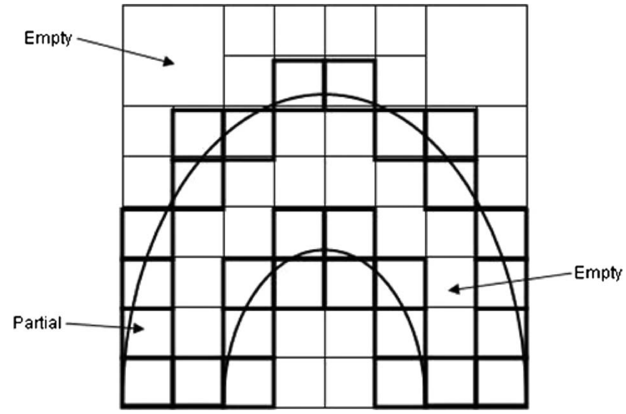

Fig. 2 A 2D shape and its quadtree using our simplified representation method (note that only the quadrants with bold edges contribute to the shape information)

ray tracing. The developed method stores generated octrees in extensible markup language (XML) format files, which are efficient for processing and exchanging data over the Internet, as well as for processing partial (local) comparisons.

2 Normalization

Usually 3D models are given in arbitrary units of measure and in unpredictable positions and orientations in 3D space. To make 3D objects invariant with respect to rotation, translation, and scaling, they must be converted into canonical representations, before converting them into a given shape representation. With octree representations, only tree structures including nodes and link paths are compared when measuring shape similarities. Therefore, due to the nature and generation of octrees, translation and global scaling are not concerns.

To eliminate the rotation differences, the majority of proposed normalization techniques use principal component analysis (PCA) [33–36]. A 3D mesh model is determined by a set of vertex coordinates. Thus, normalization of a 3D mesh model essentially consists of rotating the axes of the object vertex coordinates to align them with the PCs. An example of using PCA to normalize 3D objects is shown in Fig. 1. Three objects—models 1, 2, and 3—were normalized using PCA. The appearance of each model before PCA is shown on the left side of Fig. 1. The appearance of each model after being normalized using PCA is shown on the right side of Fig. 1. The results show that the three objects were aligned well after PCA in the same final orientation.

Similarity between models 1 and 2 and between models 1 and 3 increased after PCA normalization: Similarity increased to 18.6% for model 2 and 31.8% for model 3 (see Table 1) compared with the corresponding similarity scores before PCA normalization.

Table 1 Comparison results for sample models in Fig. 1

Compared with model 1	Scores before PCA	Scores after PCA	Increase in score (%)
Model 2	0.688	0.816	18.6
Model 3	0.607	0.800	31.8

3 Simplified Octree Representation

Our proposed method uses the following assertions when constructing a simplified octree for a 3D object.

- If any vertex of a triangle is on the boundary of the octant, the vertex is considered to be within the octant.
- If none of the vertices of a triangle is within an octant, and the triangle intersects the octant, we neglect the intersection.
- If the resolution of an octree is N vertices per octant, when the number of vertices within an octant is N, or less than N, we consider the octant to be empty; otherwise, the octant is partial. Practically, we set $N=5$ or 10, which was found to be sufficient for our proposed shape matching method.
- If any octant contains more than N vertices when the subdivision procedure reaches a predefined final level, we label it as partial; otherwise, the octant is marked as empty.

Using our octree representation, octants inside and outside of the 3D object are not distinguished in the final octree, and they are marked the same, empty. The rule should not affect the shape matching results, because we only care about surface shapes. In other words, we treat all 3D objects as closed, hollow surface models. Since only surface information rather than volume information is stored, the memory space required for recording shape information is greatly reduced [37]. Figure 2 illustrates our proposed simplified octree representation method for a 2D shape and a corresponding quadtree.

Our algorithm involves recursive calls to a subdivision procedure that subdivides the partial cubes into child cubes. In general, two criteria can be used for stopping octree node subdivision.

- *Criterion 1.* Subdivision can be stopped if a current node has a vertex count that is less than the predefined threshold.
- *Criterion 2.* Subdivision can be stopped at a predefined maximum number of octree levels. The resulting faces are assigned to the leaf nodes and the nodes are marked as partial or empty, depending on the number of vertices contained in the octant.

In our method, for easy shape comparison, we control the resolution of octree subdivision by specifying the number of levels in the octrees. At the given octree level, we then use a vertex density threshold to determine if an octant is partial or empty, where the vertex density is defined to be the number of vertices contained in the octant divided by the total number of vertices in the shape. Therefore, the impact of variations in the resolution for different shapes is minimized.

Creating a new data type would hinder exchange of the octree representations of the 3D models. As a result, for our proposed method, we chose to use XML to represent the model octrees. Using XML, a straightforward linear representation of a simplified octree is produced. With standard XML XQuery or Xpath commands, we can efficiently search and retrieve partial information from the octree, which is extremely useful for local similarity comparisons. Because XML is text-based, applications running on different platforms can easily access, translate, and use XML format information. For example, if a designer looks for an existing alternative part with a similar shape, he/she does not need to provide his/her own exact design model to a parts' provider. Instead, octree information of the original design, in XML format, would be sufficient.

4 Shape Similarity Measure

To measure the shape similarity of the two models, with our method, we can just traverse through the two corresponding octrees and compare the nodes sequentially for exact matching, or nonsequentially for inexact matching, i.e., isomorphic matching. Our proposed similarity measure is given as follows.

Let $s_{i,j}^{(l)}$ be the similarity score between the node i in octree A and the node j in octree B at level l. Then

$$s_{i,j}^{(l)} = \frac{1}{8^l} \times n_{i,j}$$
$$n_{i,j} = \begin{cases} 1 & \text{if the } i\text{th node in } A \text{ has the same label as the corresponding } j\text{th node in } B \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

For exact matching, the nodes at the same level in the two octrees are paired and compared sequentially. The similarity score uses $1/8^l$ as a weighting factor. Therefore, let S be the similarity score between object B and object A, and let N be the total number of nodes, including the root octant. For exact matching,

$$S = \sum_{i=1}^{N} s_{i,i}^{(l)} \tag{2}$$

Our octree-based method can also compare two octrees, which are isomorphic. In Fig. 3, if the two entire trees are compared in order, the two simplified octrees are dissimilar. However, if only the first subtrees from the root to the partial leaf nodes at the bottom of the trees are compared, the two simplified octrees are exactly the same. By calculating the similarity between each pair of nodes between level 1 of octree A and level 1 of octree B ($l=1$), using Eq. (1), we obtain an 8×8 score matrix (see Table 2). The value in each cell represents the similarity score of two level 1 nodes in the two isomorphic octrees. For example, to calculate the value of cell (2,5), we compare node 2 in octree A and node 5 in octree B. At level 1, these two nodes are partial, so the score at level 1 is $\frac{1}{8}$, according to Eq. (1). Then, if the child octants are compared in order, the total score at level 2 is equal to $\frac{1}{64}+\frac{1}{64}+0+\frac{1}{64}+0+\frac{1}{64}+\frac{1}{64}+\frac{1}{64}=\frac{6}{64}$. Therefore, the final score for the two trees is $\frac{1}{8}+\frac{6}{64}=\frac{14}{64}$.

After finding the 8×8 score matrix, we determine the final similarity score for level 1 from the matrix as follows.

1. We find the maximum value in all cells, in this case $s_{2,3}^{(1)}=2/8$, and add it to the total score S. Since each node can only have one best match, we cross out row 2 and column 3, i.e., the second node of octree A and the third node of octree B at level 1.
2. Then, we find the next maximum score in the rest of the cells. For example, in Fig. 3, the next maximum score is $s_{5,5}^{(1)}=2/8$, which is then added to S. Again we mark the fifth node of octree A and the fifth node of octree B at level 1 by crossing out row 5 and column 5.
3. By repeating the above steps until all columns and cells are crossed out, the final matching score for the two octrees at level 1 is obtained by summing up the first eight maximum scores in the matrix. No node is repeated.

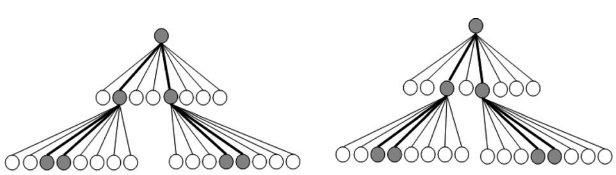

Fig. 3 Two isomorphic trees

The above procedure can be extended to compare octrees with nonisomorphic rotation to resolve the PCA alignment issue mentioned in Sec. 3. By combining model normalization using PCA and isomorphic matching at a given resolution level, our method provides a powerful means for finding general similarities between parts based on local similarities between part features.

5 Experiment and Results

File translations can often lead to losing the common base needed for comparing the efficiency and effectiveness of the different tools and methods. However, to test our method, we used a commonly used and well-defined data set containing 101 CAD models, which were downloaded from Refs. [38,39]. Figure 4 shows that when we queried for the top ten ranked results for an existing model in the data set, the models with similar shapes were found within the data set.

Table 2 An example score matrix for comparing two octrees without order

s	1	2	3	4	5	6	7	8
1	1/8	1/8	0	1/8	0	1/8	1/8	1/8
2	0	0	2/8	0	14/64	0	0	0
3	1/8	1/8	0	1/8	0	1/8	1/8	1/8
4	1/8	1/8	0	1/8	0	1/8	1/8	1/8
5	0	0	14/64	0	2/8	0	0	0
6	1/8	1/8	0	1/8	0	1/8	1/8	1/8
7	1/8	1/8	0	1/8	0	1/8	1/8	1/8
8	1/8	1/8	0	1/8	0	1/8	1/8	1/8

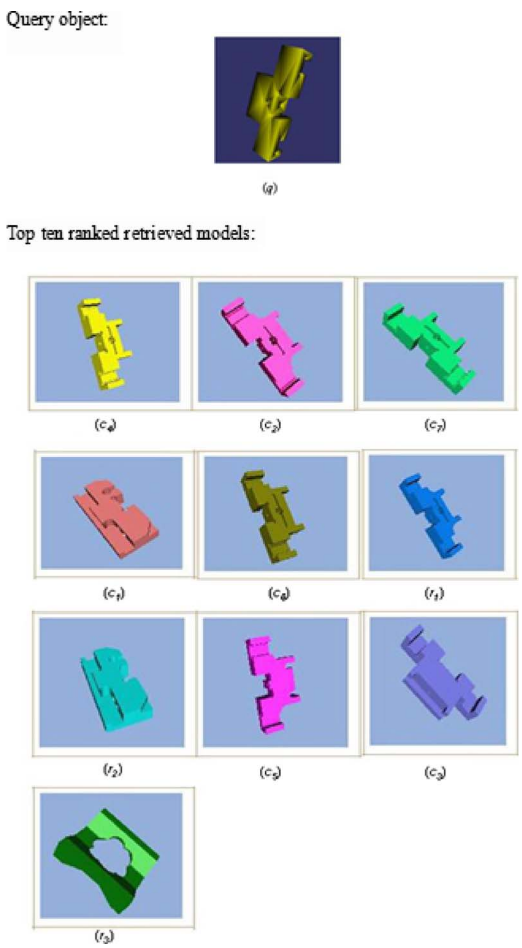

Fig. 4 An example of search results

To evaluate the retrieval performance of our method, we used a precision-recall method, which is a measure used to evaluate search strategies for information retrieval [40]. Recall is the proportion of relevant models retrieved (RR) from the total number of relevant models (REL) in the database. It is usually expressed as a percentage

$$\text{recall} = \frac{\text{RR}}{\text{REL}} \times 100\% \tag{3}$$

Precision is the proportion of relevant models retrieved (RR) to the total number of irrelevant and relevant models retrieved (RET). It is also usually expressed as a percentage

$$\text{precision} = \frac{\text{RR}}{\text{RET}} \times 100\% \tag{4}$$

For example, in Fig. 4, seven objects in the database are relevant to the query model q (REL=$\{c_1, c_2, c_3, c_4, c_5, c_6, c_7\}$). To evaluate our proposed method, using the precision-recall method, we first ranked all objects from the collection Σ using our shape representation and matching method. The set of top ten ranked models from Σ, according to our measured similarity to q, is $\{c_4, c_2, c_7, c_1, c_6, r_1, r_2, c_5, c_3, r_3\}$, where $r_i \in (\Sigma - \text{REL})$ $(i=1,2,3,\ldots)$. The last relevant model lies at position 9 (c_3).

In order to evaluate the performance of our octree descriptor, our 3D retrieval system was compared with SHAPESIFTER, a 3D retrieval system developed by Corney et al. [38,39], using the same 3D model database and the same set of six different queries for six different objects. We took each object from the query set as a query to retrieve models and compute precision-recall values.

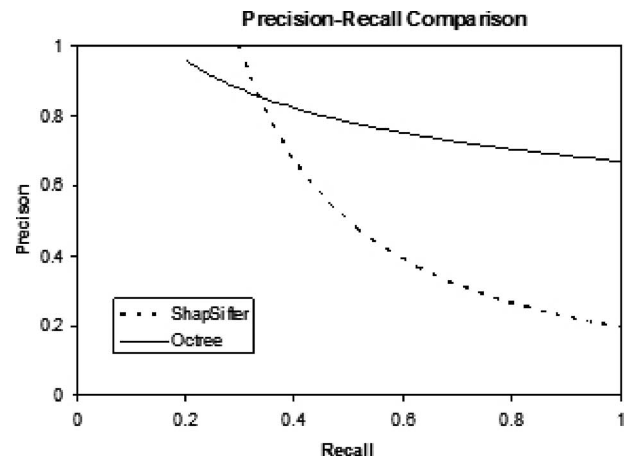

Fig. 5 Precision-recall curves (compared with SHAPESIFTER [38])

Then all the precision-recall data were pooled together and the precision-recall curves were drawn using statistical regression. Statistical results, using a paired t-test, show that the overall precision of our system is higher than that of SHAPESIFTER at a 5% significance level (Fig. 5). The results shows that SHAPESIFTER performs at slightly higher precision levels under low recall settings (<40%). However, with high recall settings (>40%), our proposed system achieves significantly higher relative accuracy.

We also performed the test for models from the engineering shape benchmark (ESB) [41] using the proposed octree shape search engine. The result is shown in Fig. 6. Overall, the search engine with the proposed octree representation performed well on retrieving shapes based on their global features. Using our 3D retrieval system, with a Pentium 4 processor and 512 Mbyte RAM PC, the average computation time for searching the data set of 101 models was approximately 0.8 s. There is also no visible delay when the search engine runs against the ESB data set of 872 models. Prior related studies did not provide the searching speed required for similar queries.

Octree representations allow groups of adjacent elements of a 3D object, which have the same property, to be stored as one octant. This can lead to a substantial reduction in the memory space required for storing a 3D object [42]. For some models, range of feature size is extreme. As a result, octrees with more depth are required to preserve the detailed features. In such cases, the XML file, which contains the octree information, can be serialized in binary format to reduce the storage memory needed. In addition, objects represented by octrees can be processed at multiple resolutions. When a tree is traversed from the root to the

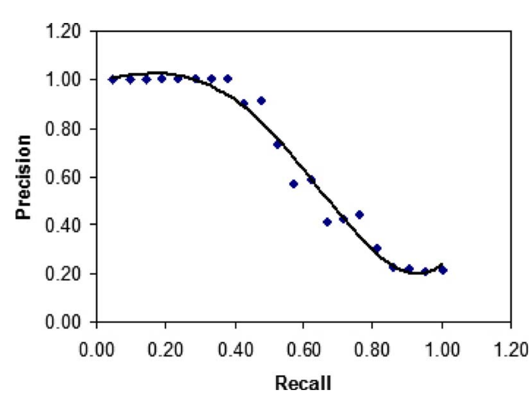

Fig. 6 Precision-recall curve (for models from ESB [41])

leaves, each successive level of the tree represents a progressive refinement of the entire object. Therefore, shape similarity measures can be found for 3D objects at selective levels of resolution.

6 Conclusions

An efficient method for determining the geometrical similarities of 3D mesh objects was developed. The proposed method uses PCA normalization to create canonical representations of 3D mesh models and then converts the canonical representations to simplified octrees. The proposed octrees only store partial and empty information, which reduce computation time dramatically. The resulting octree descriptors are stored in XML file format, which facilitates fast processing during matching, platform-free data exchange, and partial queries. The proposed shape matching method is invariant to scaling, translation, and rotation. The simplified octree representation also supports local and multilevel matching. A similarity measure was also developed. Future work will include shape searching optimization and model keyword integration to provide more efficient mechanical model searching [43].

References

[1] Pope, A., 1994, "Model-Based Object Recognition: A Survey of Recent Research," Department of Computer Science, University of British Columbia, Canada, Technical Report No. 94-04.

[2] Kriegel, H., Brecheisen, S., Kr̋oger, P., Pfeifle, M., and Schubert, M., 2003, "Using Sets of Feature Vectors for Similarity Search on Voxelized CAD Objects," Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '03), San Diego, CA.

[3] Rowe, J., Razdan, A., Collins, D., and Panchanathan, S., 2001, "A 3D Digital Library System: Capture, Analysis, Query, and Display," Proceedings of the Fourth International Conference on Digital Libraries (ICADL), Bangalore, India.

[4] Kastenmuller, G., Kriegel, H., and Seidl, T., 1998, "Similarity Search in 3D Protein Databases," Proceedings of the German Conference on Bioinformatics, Koln, Germany.

[5] Bruno, I. J., Kemp, N. M., Artymiuk, P. J., and Willett, P., 1997, "Representation and Searching of Carbohydrate Structures Using Graph-Theoretic Techniques," Carbohydr. Res., **304**, pp. 61–67.

[6] Leizerowicz, W., Lin, J., and Fox, M. S., 1996, "Collaborative Design Using WWW," Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), University of West Virginia.

[7] Samet, H., 1995, "Modern Database Systems," *The Object Model, Interoperability and Beyond*, W. Kim, ed., ACM, New York, pp. 361–385.

[8] Samet, H., 2004, "Object-Based and Image-Based Object Representations," ACM Comput. Surv., **36**(2), pp. 159–217.

[9] Marr, D., and Nishihara, H. K., 1978, "Representation and Recognition of the Spatial Organization of Three-Dimensional Shapes," Proc. R. Soc. London, **200**, pp. 269–294.

[10] Biederman, I., 1985, "Human Image Understanding: Recent Research and a Theory," Comput. Vis. Graph. Image Process., **32**, pp. 29–73.

[11] Biederman, I., 1987, "Recognition by Components: A Theory of Human Image Understanding," Psychol. Rev., **94**, pp. 115–147.

[12] Thompson, D. W., and Mundy, J. L., 1987, "Three Dimensional Model Matching From an Unconstrained Viewpoint," Proceedings of the IEEE International Conference on Robotics and Automation, Raleigh, NC, pp. 208–220.

[13] Lowe, D. G., 1986, *Perceptual Organization and Visual Recognition*, Kluwer, Boston, MA.

[14] Ullman, S., 1989, "Aligning Pictorial Descriptions: An Approach to Object Recognition," Cognition, **32**, pp. 193–254.

[15] Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D., and Jacobs, D., 2003, "A Search Engine for 3D Models," ACM Trans. Graphics, **22**(1), pp. 83–105.

[16] Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., and Ramani, K., 2005, "Three-Dimensional Shape Searching: State-of-the-Art Review and Future Trends," Comput.-Aided Des., **37**, pp. 509–530.

[17] Ramesh, M., Yip-Hoi, D., and Dutta, D., 2001, "Feature-Based Shape Similarity Measurement for Retrieval of Mechanical Parts," ASME J. Comput. Inf. Sci. Eng., **1**, pp. 245–256.

[18] Cardone, A., Gupta, S., and Karnik, M., 2003, "A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications," ASME J. Comput. Inf. Sci. Eng., **3**, pp. 109–118.

[19] Bespalov, D., Regli, W., Shokoufandeh, A., and Sun, W., 2003, "Scale-Space Representation and Classification of 3D Models," ASME J. Comput. Inf. Sci. Eng., **3**, pp. 315–324.

[20] McWherter, D., Peabody, M., Regli, W., and Shokoufandeh, A., 2001, "Indexing and Clustering of Solid Models: Techniques and Empirical Results," ASME J. Comput. Inf. Sci. Eng., **1**, pp. 300–310.

[21] Reeb, G., 1946, "Sur les Points Singuliers d'une Forme de Pfaff Completement Integrable ou d'une Fonction Numerique (On the Singular Points of a Completely Integrable Pfaff Form or of a Numerical Function)," Acad. Sci. Paris, C. R., **222**, pp. 847–849.

[22] Blum, H., 1967, "A Transformation for Extracting New Descriptors of Shape," *Models for the Perception of Speech and Visual Form*, W. Wathen-Dunn, ed., M.I.T. Press, Cambridge, MA, pp. 362–380.

[23] Iyer, N., Kalyanaraman, Y., Lou, K., Jayanti, S., and Ramani, K., 2003, "A Reconfigurable 3D Engineering Shape Search System Part I: Shape Representation," Proceedings of the ASME DETC '03 Computers and Information in Engineering (CIE) Conference, Chicago, IL.

[24] El-Mehalawi, M., and Miller, R., 2003, "A Database System of Mechanical Components Based on Geometric and Topological Similarity. Part II: Indexing, Retrieval, Matching, and Similarity Assessment," Comput.-Aided Des., **35**, pp. 95–105.

[25] Pu, J., and Ramani, K., 2007, "Priority-Based Geometric Constraint Satisfaction," ASME J. Comput. Inf. Sci. Eng., **7**(4), pp. 322–329.

[26] Kela, A., 1989, "Hierarchical Octree Approximations for B-Rep Based Geometric Models," Comput.-Aided Des., **21**, pp. 355–362.

[27] Novazo, I., Ayala, D., and Brunet, P., 1986, "A Geometric Modeler Based on Exact Octree Representation of Polyhedra," Comput. Graph. Forum, **5**(2), pp. 91–104.

[28] Anand, S., and Knott, K., 1991, "An Algorithm for Converting the Boundary Representation of a CAD Model to Its Octree Representation," Comput. Ind. Eng., **21**, pp. 343–347.

[29] Krishnan, R., Das, A., and Gurumoorthy, B., 1996, "Octree Encoding of B-Rep Based Objects," Comput. Graph., **20**(1), pp. 107–114.

[30] Funkhouser, T., Kazhdan, M., Shilane, P., Min, P., Kiefer, W., Tal, A., Rusinkiewicz, S., and Dobkin, D., 2004, "Modeling by Example," ACM Trans. Graphics, **23**(4), pp. 652–663.

[31] Azernikov, S., Miropolsky, A., and Fischer, A., 2003, "Surface Reconstruction of Freeform Objects Based on Multiresolution Volumetric Method," Proceedings of the Eighth ACM Symposium on Solid Modeling and Applications, Seattle, WA, pp. 115–126.

[32] Wang, C.-S., Shih, T. K., Huang, C.-H., and Chen, J.-F., 2003, "Content-Based Information Retrieval for VRML 3D Objects," 17th International Conference on Advanced Information Networking and Applications, pp. 386–391.

[33] Wang, C.-S., Chen, J.-F., Hung, L.-P., and Huang, C.-H., 2004, "Efficient Indexing and Retrieval Scheme for VRML Database," Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, pp. 130–135.

[34] Vranic, D. V., Saupe, D., and Richter, J., 2001, "Tools for 3D-Object Retrieval: Karhunen–Loeve Transform and Spherical Harmonics," 2001 IEEE Fourth Workshop on Multimedia Signal Processing, pp. 293–298.

[35] Paquet, E., Murching, M., Naveen, T., Tabatabai, A., and Rioux, M., 2000, "Description of Shape Information for 2-D and 3-D Objects," Signal Process. Image Commun., **16**, pp. 103–122.

[36] Kazhdan, M., Funkhouser, T., and Rusinkiewicz, S., 2003, "Rotation Invariant Spherical Harmonic Representation of 3D Shape Descriptors," Proceedings of the Symposium on Geometry Processing, pp. 156–164.

[37] Leifman, G., Katz, S., Tal, A., and Meir, R., 2003, "Signatures of 3D Models for Retrieval," Fourth Israel Korea Bi-National Conference on Geometric Modeling and Computer Graphics, pp. 159–163.

[38] http://www.shapesearch.net.

[39] Corney, J., Rea, H., Clark, J., Pritchard, J., Breaks, M., and MacLeod, R., 2002, "Coarse Filters for Shape Matching," IEEE Comput. Graphics Appl., **22**(3), pp. 65–74.

[40] Baeza-Yates, R., and Ribeiro-Neto, B., 1999, *Modern Information Retrieval*, Addison-Wesley, Reading, MA.

[41] Jayanti, S., Kalyanaraman, Y., Iyer, N., and Ramani, K., 2006, "Developing an Engineering Shape Benchmark for CAD Models," Comput.-Aided Des., **38**(9), pp. 939–953.

[42] Udeshi, T., and Parker, E., 2004, "Exploiting Self-Similarity in Geometry for Voxel Based Solid Modeling," ASME J. Comput. Inf. Sci. Eng., **4**, pp. 49–55.

[43] Udoyen, N., and Rosen, D., 2008, "Description Logic Representation of Finite Element Analysis Models for Automated Retrieval," ASME J. Comput. Inf. Sci. Eng., **8**(3), p. 031002.