

# Introducing 3D Model Search into Case Study Library Management for Engineering Education

Jiale Wang, Liu Yang, Jun Yin

College of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China

wjl8026@zjgsu.edu.cn, liu\_yang@zjgsu.edu.cn, junyin2@zjgsu.edu.cn

**Keywords:** 3D modelling, CAD model, Content-Based Retrieval

**Abstract.** This paper discusses using 3D model search techniques to build and access case study libraries that act as an interdisciplinary tool for engineering education. After introducing the concept and benefits of managing case study libraries by 3D model search techniques, we present a prototype system of a case study library for mechanical parts development. Several aspects of the prototype system are discussed, including user interface, shape comparison, case browsing and user feedback. The paper ends with some examples of cases retrieved from our library.

## Introduction

Today's distributed product development infrastructure makes people work at one engineering function is unaware of the intricacies of another. For example, it is common that product design and manufacturing take place in different countries. The interaction between design and manufacturing is little, and the gap between them is wider. However, for engineering designers, one of those skills to keep themselves competent in a globalized work place (or a flat world) is: to be holistic and multidisciplinary [1].

Contrast to traditional classroom courses that focus on individual subject in depth, case studies expose all subjects involved in an authentic product development project to students. For example, in mechanical part development, it involves shape design, model analysis, material selection, tooling and testing, etc. Thus, case studies are an important instrument to teach students multidisciplinary or interdisciplinary knowledge.

But the difficulties in information management are a barrier to setting up and using case study libraries. Product databases of enterprises are possible case study resources. But how to retrieve wanted cases, and how to search the needed information in libraries? During the product development process, a large amount of information is generated at different stages, in forms of drawings, data and text. This product-related information is scattered, or just loosely coupled through similar file names. Identifying product information by textual names has the following shortages: 1. A short piece of text is not descriptive enough to characterize a product. 2. People may have no idea of what name the product he or she wants is. 3. Existing product information may be not properly named.

Two technological trends in engineer design and computer science provide a new way to manage the product information. One started many years ago: the intensively using of 3D CAD software in design and manufacturing in the last decade. Another is just emerging: the 3D model searching technique.

With the popularity of 3D CAD systems in industry, 3D CAD models of products serve as a virtual prototype for engineering analysis and manufacturing planning. Throughout the design and manufacturing process, all kinds of product information are generated and attached to the CAD model of a product. Most of these information is shape-related (manufacturing process details), or shape-dependent (material, cost, analysis data, etc.). As a result, CAD models become an important engineering object. Recent years, CAD research community has developed methods to search 3D CAD models by shape [2]. Users can upload an example CAD model, and then the search system returns similar ones. Since the 3D CAD model is tightly coupled with other product data, we can use

CAD model as a thread to retrieve and access the information generated at different stages of a product development process.

This paper discusses using 3D model search techniques to build and access case study libraries that act as an interdisciplinary tool for engineering education. After introducing the concept and benefits of managing case study libraries by 3D model search techniques, we present a prototype system of a case study library for mechanical parts development. Several aspects of the prototype system are discussed, including user interface, shape comparison, case browsing and user feedback. The paper ends with some examples of cases retrieved from our library.

### **Query by 2D View Sketches**

Query by 2D view sketches means that a user can draw sketches of three 2D views to represent the model he or she wants to retrieve. The following two sub-sections will explain the modules to implement this function in detail.

2D view generation is to obtain a series of projection views from a 3D model in database. These views will be used to compare with user's 2D view sketches.

The projection of a 3D model can be defined by a projection box. Each face of the box is a projection plane. A 3D model may have arbitrary orientation in space. Along which direction projecting the 3D model will be accordant with user's sketches? In many cases, a method called principal component analysis (PCA) [3] is used to normalize the orientation of 3D objects, but PCA based methods are not robust and not good enough at finding out the orientations accordant with human perception [4]. This paper proposes a method for 2D view generation by using multiple projection boxes. Our method is similar with the LFD (Light Field Descriptor) [5]. But LFD is dedicated to model-to-model matching. It is complicate and computational consuming. The method we proposed here is aimed to address the problem of matching a 3D model with its three views (model-to-views matching). There are some specific issues for this problem, and the resolving algorithm turns out to be simpler than LFD.

Our method applies a number of projection boxes with different rotating angles to a 3D model. To cover every possible viewing angle, the bounding boxes needed to be evenly distributed around the 3D model.

This algorithm rotates each bounding box around x-, y- and z- axis by an offset angle respectively. By applying these projection boxes to a 3D model, views from different viewing angles can be obtained.

Once the projection boxes are positioned, the next step is to generate the views from the 3D model. We use silhouette as the view shape of a 3D model [6]. For a 3D polygonal object, silhouette is defined as the edges that share a front- and a back-facing polygon. Intuitively, silhouette represents the outline of a 3D object.

Once the views of a 3D model have been generated, we can measure the similarity between views of 3D model and user's 2D view sketches.

Due to the reflective symmetry of silhouette, projections along opposite directions are identical. Thus, for a projection box, there are only three different views being generated. However, the corresponding relationship between the three views and user's three 2D view sketches is unknown. Each permutation of three views of a projection box is a possible match to the three sketches of a user. We compare user's three sketches with each permutation of three views of a projection box, and define the minimal dissimilarity distance between three sketches and three views over all permutations as the distance between a projection box and user's 2D view sketches.

In the past decades, the CIBR (Content-Based Image Retrieval) community has developed many methods for this purpose. Here we use the algorithm proposed by Chalechale et al [7] to address our problem.

Finally, to assess the similarity between a user's 2D view sketches and a 3D model, we compute the minimal over all projection boxes. Retrieval performance will be promoted with the increasing

of L. In our experiment, it's found that applying 216 projection boxes (L=6) to a 3D model is good enough.

### Query by Skeleton Sketch

Skeleton is a “central-spine” like 1D line representation of a 3D object. Skeleton retains the topology of the original object [8]. Query by skeleton sketch allows users conveying the topological information of a 3D model to the retrieval algorithm by drawing the skeleton of the model through query interface. The following sub-sections will explain the modules to implement this function.

To extract the skeleton, a 3D model needs to be represented as a voxel model [9]. A voxel model is composed by a set of volume elements (voxels) in a discrete 3D space. A voxel represents a cubic unit of volume, and it can be seen as the 3D counterpart of the 2D pixel that represents a unit of area. Compared with other skeleonization methods, the advantages of the thinning algorithms for our work are:

- 1) Producing one voxel width skeleton, which is convenient for the later skeleton graph generation.

- 2)Efficient to compute and relatively easy to implement.

The process of extracting the skeleton from a voxel model is called skeletonization. Researchers in the computer graphics community have proposed lots of skeletonization algorithms. The skeletonization method we use here is a kind of the so-called thinning algorithm [10]. Thinning is an iterative object reduction technique. It produces one voxel width skeleton by deleting points in the voxel model if they satisfy deletion conditions that preserve topology.

After the skeleton of a model has been extracted, we convert it to a graph data structure. A skeleton graph is generated by this way:

- 1) Node: a node represents the voxel at the end of a line, or the voxel at a cross.

- 2) Edge: an edge represents the connection between nodes.

A user's skeleton sketch is represented as one-pixel width binary image. The skeleton sketch is converted into graph in a similar way to that of the skeleton of a 3D model, except that the voxels are now pixels. When users draw a skeleton sketch, they must make sure that there are no lines overlapping with each other.

Once a graph is generated, we perform a loop detection algorithm on it. The algorithm will find each loop in a graph and replace it with a node. An attribute value is attached to every node to mark it, L for nodes of loop, N for other nodes. Loop detection and elimination will bring out two benefits: 1. A graph is reduced to a tree-like structure by removing internal loops. This will make the graph comparing more rapid. 2. A loop in a graph denotes a hole in the original model. Hole is an important feature for mechanical parts and engineering models. Identifying loop is to recognize this feature. Users can search for models with holes by searching for graphs with L nodes.

For two give graphs, graph isomorphism is a bijective mapping from the nodes of a graph to the nodes of the other graph that preserves the structure of all edges. Isomorphism suggests that two graphs have the same number and types of nodes and the same relationship between nodes. If two graphs are isomorphic, they are identical. A maximum common subgraph is a graph that is a subgraph of both and with the maximum number of nodes over all possible common subgraphs. Maximum common graph also can be seen as the largest local parts of two graphs that are isomorphic. More similar are two graphs, closer will be the ratio to 1

Although the general graph isomorphism algorithms are NP-complete, it is feasible for small graphs of up to 20-30 nodes. In our experiment, the skeleton graphs of 80% CAD models have less than 15 nodes.

### Similarity Degree Combination

Now, we get two similarity degrees measured by two metrics: the similarity degree of “model-to-2D view sketches” and the similarity degree of “model-to-skeleton sketch”. A simple

approach for combining the similarity degrees measured by different metrics is to compute the weighted sum of them. However, it is not a good approach, for the difficulty to choose the weights that provide a good retrieval performance in general. Our method for resolving this problem is to treat the weights as probability and use the information from Relevance Feedback (RF) to estimate them. RF makes the search process as an interaction between the computer and users. For a RF based search process, the system first retrieves similar models and returns them to user. Then, the user provides feedback regarding the relevance of some of the retrieval results (user mark the relevant objects in the results and submit them back to the search system). Finally, the system uses the feedback information to improve the performance in the next iteration.

## Implementation and Experiment

In order to evaluate the validity and performance of our method, we implemented a prototype system and conduct some experiments on it.

To evaluate the retrieval performance of a search system, a ground-truth database is needed to assess the relevance of models. We designed a ground-truth database including 421 3D CAD models and 39 classes for the test. The models in the database are classified manually by three of our research assistant students.

The performance of our prototype system is compared with that of three other widely discussed 3D shape retrieval methods:

- Sphere harmonic descriptor: a rotation invariant representation of shape feature obtained by computing the norm of each harmonic frequency of a function on concentric spheres. [11]
- D2 Shape Distribution: a histogram of distances between pairs of points on the surface of a 3D object [12]
- Shape histogram: a histogram of distances from the center of mass to points on the surface of a 3D object [13]

The precision-recall curves are illustrated in Figure 1. It can be seen that our proposed method has the best retrieval performance.

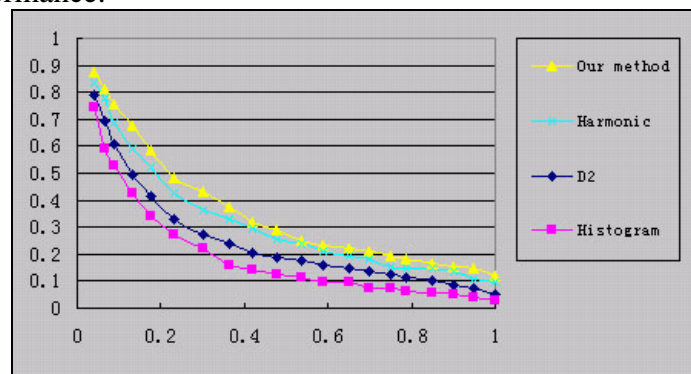


Fig. 1 Precision and recall plot of 3D model retrieval algorithms

Figure 2 shows the retrieval results of a query after a RF iteration. The retrieval results are listed in decreasing order of similarity degree from left to right and top to bottom. The models with black frame are relevant models.

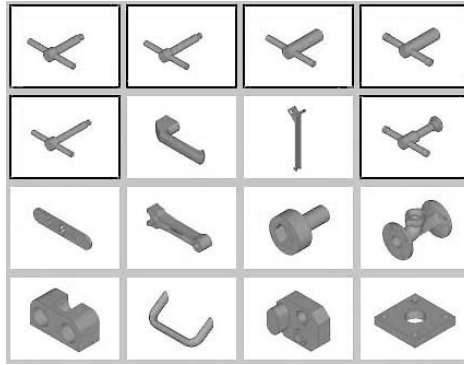


Fig. 2 Retrieval results

## Conclusions

This paper discusses using 3D model search techniques to build and access case study libraries that act as an interdisciplinary tool for engineering education. We present a prototype system of a case study library for mechanical parts development. Several aspects of the prototype system are discussed, including user interface, shape comparison, case browsing and user feedback. Some experiments results is presented to demonstrate the effectiveness of the prototype system.

## Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.61202200) and the Scientific Research Fund of Zhejiang Provincial Education Department (No.Y201018008).

## References

- [1] Felder R.M. A Whole New Mind for a Flat World. *Chem. Engr. Education*, 40(2), 96-97.
- [2] Iyer N., Lou K., Jayanti S., Kalyanaraman Y., Ramani K. Shape-based Searching for Product Life Cycle Applications. *Computer-Aided Design*, Vol 37 (2005) 1435-1446.
- [3] Jolliffe IT. *Principal Component Analysis*, 2nd edition. Springer, 2002.
- [4] Funkhouser T, Min P, Kazhdan M, Chen J, Halderman A, et al. A Search Engine for 3D Models. *ACM Transactions on Graphics*; 2003, 22(1):83-105
- [5] Chen DY, Tian XP, Shen YT, Ouhyoung M. On Visual Similarity Based 3D Model Retrieval. *Computer Graphics Forum*; September 2003, 22(3): 223-232.
- [6] Isenberg T, Freudenberg B, Halper N, Schlechtweg S, Strothotte T. A Developer's Guide to Silhouette Algorithms for Polygonal Models. *Computer Graphics and Applications*; 2003 23(4):28 - 37
- [7] Chalechale A, Naghdy G, Mertins, A. Sketch-based image matching Using Angular partitioning. *Systems, Man and Cybernetics, Part A, IEEE Transactions on*; 2005, 35(1):28-41
- [8] Blum H. A Transformation for Extracting New Descriptors of Shape. *Models for the Perception of Speech and Visual Form*, eds. E. Dunn, Cambridge, MA; 1967, p. 362-380
- [9] Huang J, Yagel R, Filippov V, Kurzion Y. An Accurate Method To Voxelize Polygonal Meshes. *Proc. IEEE Symposium on Volume Visualization*; 1998, p. 119-126
- [10] Wenjie X, Thompson RP, Perucchio R. A Topology-Preserving Parallel 3D Thinning Algorithm for Extracting the Curve Skeleton. *Pattern Recognition*; July 2003, 36(7):1529-1544

- [11] Kazhdan M, Funkhouser T, Rusinkiewicz S. Rotation invariant spherical harmonic representation of 3D shape descriptors. In Symposium on Geometry Processing, June 2003.
- [12] Osada R, Funkhouser T, Chazelle B, Dobkin D. Matching 3D models with shape distributions. Shape Modeling International; May 2001, p.154–166
- [13] Ankerst M, Kastenmuller G, Kriegel HP, Seidl T. Nearest neighbor classification in 3D protein databases. In Proc. 7th Int. Conf. on Intelligent Systems for Molecular Biology; 1999, Heidelberg, Germany, AAAI Press, p. 34-43