International Engineering Journal For Research & Development

E-ISSN : 2349-0721

Volume 1 : Issue 3

3-Dimensional Engineering Shape Search (3DESS) System

Mayuresh Gadge

Assistant Professor JCOET,Yavatmal

mayugadge@gmail.com

ABSTRACT

Twenty first century product development is driven by reduction in cost/time, specialization of industry, globalization, outsourcing and geographically distributed companies. Designers spend a significant amount of time searching for information that is available but cannot be located through traditional methods. Rectification of errors that have been committed due to lack of information is a costly way to learn. Nevertheless this has become a de facto process for new product design. A significant amount of information generated during the lifecycle of a product is associated with 3D models. Reuse of this information can significantly shorten lead times and reduce costs during a product's lifecycle. Since design knowledge and context are intimately related to 3D geometry, text-based search cannot satisfy many requirements. This paper presents a brief overview of a novel approach to search for 3D models.

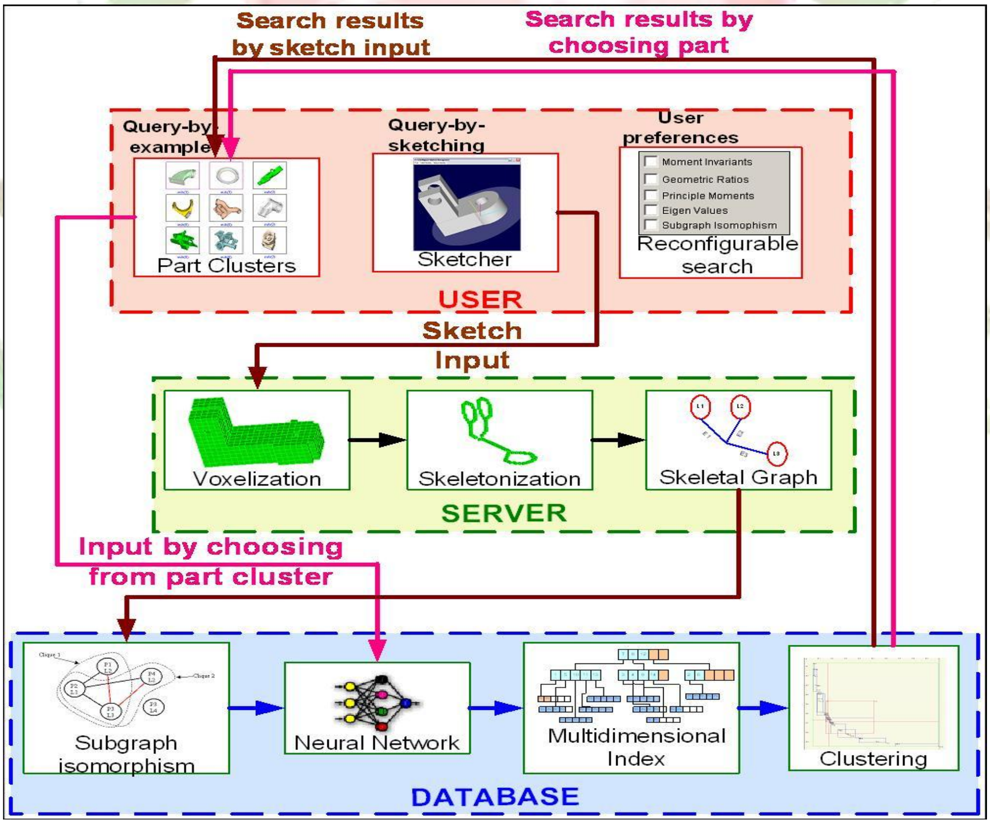
INTRODUCTION

Designers spend about 60% of their time searching for the right information, which is rated as the most frustrating of engineers' activities. Often designers have to make "assumptions" while designing, which may lead to problems at a later stage of the design. Such unforeseen errors can lead to allocation of scarce resources for solving unanticipated problems or putting out "fires." In the product development context, fire-fighting describes the unplanned allocation of engineers' time and other resources to fix problems discovered late in a product's development cycle. Re-learning from errors is a costly method of learning. Nevertheless, fire-fighting has become the *de facto* process for developing new products in industry. For example, in engineering, it is conservatively estimated that more than 75% of design activity comprises case-based design - i.e. reuse of previous design knowledge to address a new design problem. However, physical inventories such as tooling and associated knowledge are often not located or reused resulting in significant losses. Twenty-first century product development will mainly be driven by the need to reduce cycle times and costs. Design and associated knowledge reuse is the key to reducing new product development time as well as fire-fighting.Engineering design and manufacturing has progressed extensively from 2D to 3D in the last decade. This includes development of large-scale computer-aided design (CAD) and manufacturing software used throughout a product's lifecycle. Advances in 3D graphics hardware have also contributed to the widespread use of 3D models. The use of 3D models is especially high in net-shape manufacturing processes such as injection moulding and casting. In these processes, it is critical to visualize and understand the part accurately before building expensive tooling, including dies and molds. For example, approximately 66% of CAD modeling in the mold making industry was being done in 3D in 2001. This has increased to 80% in 2003. As a result, the shapes of products and associated tooling have increased considerably in complexity and number, contributing to the *3D model explosion.* For example, a cursory study reveals that the Boeing 777 has about 3 million parts, of which about 132,500 parts are uniquely engineered.

1. All models will not have a well-defined attached context
2. Keywords such as project names or part names may be unknown to the user.
3. Context may be too narrow or too broad to retrieve relevant models.
4. Context changes with time, such as when designers or naming conventions change.

The Internet has facilitated newer business models along with geographically-distributed design and manufacturing. Hence, 21st century designers may not be familiar with design history and context, making a keyword-based search an unattractive option. Although large parts of our brains are devoted to the processing of shape, the visual channel has not been exploited for information retrieval in engineering.

Thus, a search system which is capable of retrieving similar 3D models based on their shape will retrieve shape and related knowledge that would not be discovered by other means. Furthermore, designers relate products to 3D shapes more than to meta-data and, hence, only a shape-based system can provide an answer when other methods fail.



The Shape Search System

The search system consists of a Client-Server-Database architecture. The Client end consists mainly of the user interface which enables query creation and display of results. The Server side essentially takes in the shape input from the Client and converts it into multiple representations (voxel, skeleton, feature vectors and skeletal graph), which are then stored into the Database. This approach makes the following major contributions:

1. Customized user query construction.
2. Only "essential" shape features are captured
3. Represents shape as a combination of both geometry and topology.
4. Multi-level shape representation.
5. Accurate and computationally efficient.

User Interface

The function of the user interface is to allow a user to create a custom query, either by example or by sketching, and to display the search results. The primary function of the user interface is to enable the following user inter-actions:

1. *Query-by-example:* A query is initiated by choosing a part most similar to that the user desires. Part choice will be enabled by the user interface that will enable a user to drill down through clusters of similar parts to find a part to start searching with.

2. *Query-by-sketching:* A query is initiated by the user by quick sketching of an approximate 3D mock-up of the model he/she is interested in searching for. Alternatively, a user could also specify the location of a CAD file at the Client end. The sketch interface is based on a web-based collaborative system implemented in Java™ and uses an AC IS modeling kernel.

3. *Feature Vector Choice Interface:* The user can customize the feature vectors to be used in the search. This enables the user to "reconfigure" the search at any time.

Server

The first step in converting 3D models to a searchable representation is Voxelization, prior to which the model is normalized into a canonical form independent of position, orientation and scaling.

Voxelization

Voxelization is defined as the process of converting a geometric representation of a synthetic model into a set of voxels (volume elements) that best represents the synthetic model within the discrete model space. In recent years, a number of curve, surface and polygon mesh voxelization algorithms have been proposed. Most algo-rithms provide efficient ways to extend 2D scan conversion methods to the 3D domain. The important differ-ence between 2D scan conversion and voxelization is that voxelization is decoupled from the rendering process and hence is a one-time process. The most dominant solid representation methods are Boundary Representation (B-Rep) and Constructive Solid Geometry (CSG). B-Rep describes a part in terms of its vertices, edges and faces. CSG describes the part in terms of a set of Boolean operations applied to primitive geometric entities such as cubes and cylinders.It is difficult to voxelize a B-Rep solid because the interior is not explicitly defined. However, advances in parallel computation, hardware and computational power have made voxelization of B-Rep solids almost real-time. Most 3D CAD systems such as Pro/Engineer™, SolidWorks™ and IDEAS™ use a B-Rep representation as their internal data structure. Furthermore, there is no unique CSG representation standard for a part. Therefore, only B-Rep models have to be considered, since most engineering parts are either available in B-Rep formats or can be easily translated into a neutral B-Rep representation.

Skeletonization

Skeletonization is the process of extracting a skeleton from a 3D binary image (voxel model). The model can be converted into a binary 3D digital model. However, in digital spaces, only an approximation to the "true skeleton" can be extracted. The two requirements for a skeleton are:

1. Topological**:** A skeleton must retain the topology of the original object.

2. Geometrical**:** A skeleton must be in the "middle" of the original object and must be invariant to translation, rotation and scaling.

The three major Skeletonization techniques are:

1. Voronoi-based
2. Distance transform based
3. Thinning

Thinning is the preferred Skeletonization method because it offers advantages over Distance transform and Voronoi based skeletons. Distance transform skeletons satisfy "geometrical" conditions but may not satisfy "topological" conditions. Voronoi based skeletons satisfy both requirements, but are very expensive to compute especially for realistic engineering models. Furthermore, Voronoi skeletons have unwanted appendages which require pruning as an additional process. Thinning satisfies the "topological" requirements but does not always satisfy "geometrical" requirements. For our application, topological correctness is more important than geometrical correctness.

Skeletal Graph Generation

The skeleton is converted into a hierarchical skeletal graph, for storing in database. This graph can be analyzed at different levels depending on information content. The main approaches in the past have been to convert the skeleton into a voxel graph or into a skeleton tree. In this approach, skeleton is converted into a skeletal graph, that is made up of the following basic skeletal entities:

1. Node - the voxel situated at the ends of the edges

2. Edge - Set of voxels forming a single geometric entity

3. Loop - One or more edges forming a closed entity

Database

A similarity measure is a function for quantifying the similarity between two models. It takes the feature vectors of the query model and that of a model in the database and outputs a real number that reflects the degree of similarity between the two models.The Euclidean distance between points in feature space is used to indirectly represent the similarity measure. Clearly, this similarity measure is a metric and is the primary measure of similarity for, models based on features.

Database Indexing

Index structures are used to speed up searching in large databases, and the design of these structures is a critical issue for any realistic 3D shape search system.The one-dimensional index structure is not sufficient for 3D shape search systems. The fundamental problem is that the feature vectors representing 3D models are complex data types. Searching is usually based on overall similarity *(similarity query)* rather than the similarity of individual attributes *(attribute query). A*ll the attributes have to be simultaneously to determine the similarity. In other words, a model cannot be discarded from the candidate list only because some attributes do not match the query model. Thus, in order to search similar 3D models efficiently, there needs to be an index structure with properties such that:

Graph-based indexing

In addition to similarities between feature vectors, similarity based on skeletal graphs is also investigated. Skeletal graphs are undirected entity graphs and are represented as adjacency matrices. Domain knowledge is incorporated into the graph data structure, thereby enriching it and enhancing the search capabilities. The adjacency matrix is formulated to capture the structural properties of the skeleton, such as loop-loop, edge-loop connections, etc. This representation reflects the topology and the high level geometry of the skeletons. The goal of the search system is to retrieve all models that are 'similar' and not only those that match exactly with the query model. In graph similarity search, this translates to finding models whose skeletons have isomorphism as well as sub graph isomorphism with the query model.

Implementation

The voxelization, skeletonization, and skeletal graph extraction algorithms were implemented in C++ using ACIS as the geometric modelling kernel, and the user interface was implemented in Java. The prototype was tested only for prismatic parts since skeletonization methods for prismatic parts have been covered well in the literature. The same architecture can be used to search for non-prismatic parts. Figure 1 shows a bearing and its associated skeleton. It can be seen that the skeleton has unwanted appendages as Experiments were designed and implemented to test the approaches to deal with challenges for 3D shape search systems, i.e. subjectivity of shape similarity, semantic gap and efficiency. The tests were conducted on a DELL Pentium 2.66 GHz PC with 1.0 GB RAM. The voxclization and skeletonization algorithms were implemented in C++ with ACIS libraries, while the database systems were implemented in Java. In actual deployment, the database will be constantly updated if a change is made to the CAD model, and the server side processes of voxelization, skeletonization, skeletal graph and feature vector extraction are repeated. The ability of the multidimensional R-tree index as well as SOMs to cluster similar models with a real 3D model database, and the efficiency of search operations using the R-tree index was tested using both synthetic and real datasets. The database of the real 3D models currently consists of about 150 models. Although the real database is being expanded, the size is still relatively small as compared to typical design repositories. The sizes of synthetic datasets vary from 50 to 1,000,000 data records. The effects of database *size, dimensionality* of feature vector, and *node volume* to the efficiency of the feature vector based search were studied with the synthetic datasets. The synthetic database was created using a random number generator. The utility of relevance feedback to enable a customizable similarity definition and bridge the semantic gap was tested with the real 3D database.

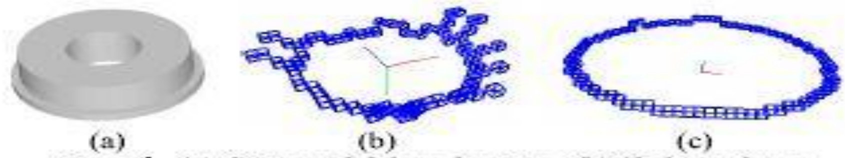


Figure 1 (a) CAD model for a bearing, (b) Skeleton for a voxel size of 1/8 (c) Skeleton for a voxel size of 1/32

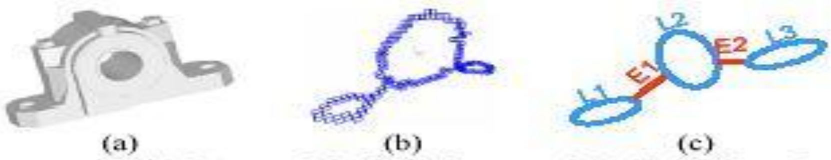


Figure 2 (a) CAD model, (b) Skeleton, and (c) Skeletal graph for a bearing block containing 182 faces

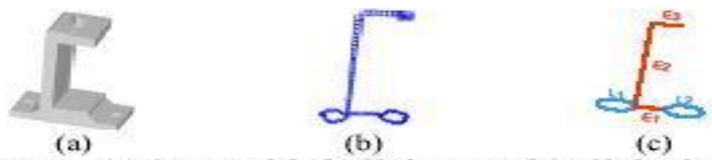


Figure 3 (a) CAD model, (b) Skeleton, and (c) Skeletal graph for a clamp containing 82 faces

REFERENCES

-Iyer N., Kalyanaraman Y., Lou K., Jayanti S., and Ramani K., (2003), "Early results with a 3D Engineering Shape Search System", International Symposium on Product Lifecycle Management (PLM '03), Indian Institute of Science, Bangalore, India.

-Website of Purdue University.

-Website of Imaginestics Inc.