A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications

Antonio Cardone, Satyandra K. Gupta[1]*, and Mukul Karnik*

Mechanical Engineering Department and Institute for Systems Research
University of Maryland
College Park, MD 20742

Abstract

Shape similarity assessment is a fundamental geometric reasoning problem that finds application in several different product design and manufacturing applications. A computationally efficient way to assess shape similarity is to first abstract 3D object shapes into shape signatures and use shape signatures to perform similarity assessment. Several different types of shape signatures have been developed in the past. This paper provides a survey of existing algorithms for computing and comparing shape signatures. Our survey consists of a description of the desired properties of shape signatures, a scheme for classifying different types of shape signatures, and descriptions of representative algorithms for computing and comparing shape signatures. This survey concludes by identifying directions for future research.

[1] Corresponding Author

1 Introduction

Popularity of 3D CAD systems in product design and manufacturing industry is resulting in a large number of CAD models being generated. Availability of these CAD models is opening up new ways in which design information can be archived, analyzed, and used. 3D geometric information is one of the main components of CAD models. Therefore shape similarity assessment is a fundamental geometric reasoning problem that finds application in several different product design and manufacturing applications. The following three examples illustrate how shape similarity assessment can be used to improve product design and manufacturing:

- **Cost Estimation for Machined Parts**. Nowadays, many job shops allow designers to submit a 3D model of the part to be machined over the Internet and provide a cost estimate based on the 3D part model. For some manufacturing domains such as rapid prototyping, reasonably accurate estimates of cost can be achieved by estimating volume or weight of the part. However, for some manufacturing domains such as machining, cost estimate depends on the geometric details of the object and automated procedures are not available for doing accurate cost estimation. Currently in such cases, humans perform cost estimation. In the Internet era, where designers solicit many quotes to make a decision, manual cost estimation is not economical. Cost of manufacturing a new part can be quickly estimated by finding previously manufactured parts that are similar in shape to the new part. If a sufficiently similar part can be found in the database of previously manufactured objects, then the cost of the new part can be estimated by suitably modifying the actual cost of the previously manufactured similar part. Figure 1 shows a new part and a previously manufactured similar part that can be used to provide cost estimate for the new part.

- **Part Family Formation**. In many manufacturing domains such as sheet metal bending, machine tools can be setup to produce more than one type of part without requiring a setup or tool change [1,2]. However, parts need to be shape compatible in order for them to share common tools and setups. Therefore, in order to find common tools and setups, geometrically similar and therefore compatible parts need to be grouped into families. Shared tools and setups can be used to manufacture objects in the same family and therefore result in significant cost savings.

- **Reduction in Part Proliferations By Reusing Previously Designed Parts**. Reusing design/manufacturing information stored would result in a faster and more efficient design process. While designing a new part the designer can refer to existing designs and utilize the components used previously. Let us consider the design of the shaft of a turbine engine. Usually the designer has two options. The first option is to design the shaft from scratch and go through the process and manufacturing planning. The second option is to refer to the database of existing designs, and select an existing shaft and either use it as it is or make minor modifications to it (e.g., drill a few holes or cut a few slots). The process of creating a preliminary concept of the shaft and selecting a shaft similar to the proposed concept is illustrated in Figure 2.

With more and more 3D models being added to databases, a need to organize and index databases of 3D models is imminent. This will provide a systematic and efficient way of retrieving similar models from the database. Group Technology has traditionally been used to categorize parts having similarities in design and manufacturing. Group Technology (GT) involves classifying similar products into groups in order to achieve economies of scale normally associated with high-volume production [3]. In order to implement GT, one must have a concise coding scheme for describing products and a method for grouping (or classifying) similar products, such as the popular Opitz, DCLASS, and MICLASS schemes. In each case the basic idea is for the users to use various tables and rules to capture critical design and manufacturing attributes of a part in an alphanumeric string, or GT code, that is assigned to that part. However, as the classification is done manually, it is subject to individual interpretation. It has been shown that human perception of similarity is subjective [4]. Thus there are possibilities of errors in such classifications.

Direct assessment of similarity between 3D models via Boolean operations is computationally very slow due to difficulty in aligning the models before performing the Boolean operation. Hence, it is often not pragmatic when the number of 3D models being considered is very large. A computationally efficient way to solve this problem is to first abstract 3D shapes into shape signatures and use shape signatures to perform similarity assessment. Shape signatures are abstractions of the actual shapes, and therefore they have lesser shape discrimination capability compared to the complete object model. Over the last few years several papers have been written that describe algorithms for shape similarity assessment. The main body of work can be divided into two different categories: (1) similarity assessment of 2D shapes, and (2) similarity assessment of 3D shapes. Representative work in 2D category includes shape signatures based on Fourier descriptor [5], turning functions [6], bending functions [7], and arch height functions [8]. A comprehensive discussion of 2D shape signatures can be found in [9,10,11]. 2D geometry and 3D geometry have several fundamental differences and unfortunately methods for computing and matching signatures of 2D shapes cannot be easily extended to 3D shapes, in most cases. The goal of this paper is to provide a review of algorithms dealing with 3D shapes and develop a common basis for comparing them. Shape similarity assessment problem has been addressed in many different application domains. Product design and manufacturing is one such domain. This paper primarily focuses on algorithms that are suitable for product design and manufacturing applications[2]. Specialized algorithms for medical [7,12], and computer vision [13,14,15,16,17,18,19] applications have also been developed. However, these algorithms will not be covered in this paper.

This paper is organized in the following manner. Section 2 describes the basic terminology needed to understand this paper. Section 3 provides an overview of the various techniques that are being used to perform similarity assessment along with a classification scheme. Sections 4 to 9 describe various approaches in detail, summarizing their advantages as well as limitations. Finally, Section 10 concludes this survey by summarizing our findings and identifying future research directions.

[2] We have covered few algorithms that were originally published for applications outside product design and manufacturing. But we believe that these algorithms can be utilized in product design and manufacturing applications.

2 Basic Terminology and Background

There are a number of techniques to assess similarity among 2D shapes. These techniques do not extend to 3D models because of the difficulty of extending the parameterization of the boundary of a 2D shape to 3D. Given a 2D shape, its parameterization is straightforward: a 1D curve. In case of the 3D domain, it is necessary to deal with solids of any genus. So most of the 2D similarity assessment methods, based on the 2D parameterization of the boundary, are not extendable to 3D.

Similarity assessment in 3D cases is usually carried out by generating shape signatures from the 3D models and then comparing these signatures using suitable distance functions. Ideally, these signatures should be representation independent and completely describe the features of the 3D model needed for similarity assessment. A shape signature could be a graph, a vector or an ordered collection of numeric values. The features captured by the signatures are usually dependent on the motivation for performing similarity analysis.

After identifying a suitable shape signature, similarity between the 3D models is expressed as the distance between the two signatures. The higher the value of the distance, the more dissimilar are the corresponding CAD models.

Let x and y be two solid models. Let S(x) and S(y) be the shape signatures of the two models. Then $\delta(\mathrm{S}(x),\mathrm{S}(y))$ is the distance between the shape signatures of the two solids. Most papers in literature argue that the shape signatures and the distance function should have the following properties:

Positivity: The distance between two CAD models should be a non-negative value. This is because a negative distance value usually does not carry any physical meaning. The distance function $\delta(\mathrm{S}(x),\mathrm{S}(y))$ should satisfy the following condition:

$$\delta(\mathrm{S}(x),\mathrm{S}(y)) \geq 0$$

Identity (Self-similarity): If the distance $\delta(\mathrm{S}(x),\mathrm{S}(y))$ is zero then the two solid models should be identical. Also if the two CAD models are the same then the distance between them should be zero. Thus the distance function should satisfy the following condition:

$$\delta(\mathrm{S}(x),\mathrm{S}(y)) = 0 \Leftrightarrow x = y$$

Symmetry: If x and y are two solid models with given shape signatures then a symmetric distance function needs to satisfy the following:

$$\delta(\mathrm{S}(x),\mathrm{S}(y)) = \delta(\mathrm{S}(y),\mathrm{S}(x))$$

Symmetry ensures that the distance is not biased to the order of comparison, and yields the same result when x is compared with y and when y is compared with x. Most of the distance functions described in this paper are symmetric.

Triangle Inequality: In some applications, the distance function should also satisfy the triangle inequality. If *x*, *y* and *z* are three solid models then the triangle inequality property requires the distance function to satisfy the following:

$$\delta(\mathrm{S}(x),\mathrm{S}(y))+\delta(\mathrm{S}(y),\mathrm{S}(z))\geq\delta(\mathrm{S}(x),\mathrm{S}(z))$$

Invariance: One of the challenges in assessing similarity among 3D models is the lack of standardization of the data stored for various parts and its representation. The first problem arises due to the existence of multiple different representations for the same part. For example, the boundary of a given object can be represented by either an analytical or a parametric representation. The shape signature should be invariant with respect to the underlying representation. The second problem is due to geometric transformations. The shape signature should be independent of any rotation or translation performed to the object and hence invariant with respect to the transformations.

Robustness and Sensitivity: Given an object, the magnitude of change in the shape signature should reflect the magnitude of change in the shape of the object. If the shape signature changes out of proportion, then the similarity assessment is not robust. This tendency to magnify small changes may lead to a conclusion that the two similar objects are dissimilar. Therefore the shape signature should be robust and should change proportional to the change in the object. On the other hand, if large changes in the shape of the objects result in very small changes in the shape signature, then the similarity assessment is considered not sensitive. Poor sensitivity leads to inadequate discrimination capability.

Computational Efficiency: One of the motivations for performing part similarity assessment is to locate similar parts in a large database or even the Internet. Thus hundreds of comparisons have to be performed within the database. This requires performing part similarity assessment in a very short time. Therefore shape signatures should be such that they can be computed and compared efficiently.

3 Overview of Techniques

Various techniques have been developed to perform similarity assessment of 3D solid models. Similarity assessment between two 3D parts involves two main steps. The first step is to compute the shape signature of the object and the second step is to compare the shape signature by a suitable distance function. Major techniques used in the shape similarity assessment area can be classified on the basis of the type of shape signatures being used. The following types of shapes signatures are currently being used:

- **Features:** Feature-based techniques evaluate the shape signature of an object based on the type, size, orientation, number and other properties of the features and their interactions. Once the features are extracted and their significant characteristics are determined, the comparison is carried out using a suitable distance function. For example, feature-graph signatures are compared by performing graph isomorphism. These techniques discriminate the 3D models based on the features and their characteristics. Hence, they do not consider the gross shape of the object. Feature interactions still pose significant challenges to successful extraction of features. Section 4 describes representative feature-based techniques in detail.

- **Spatial Functions:** These techniques use shape signatures that are spatial functions. An example of a spatial function is the Gaussian map that maps the set of normals of a solid onto a unit sphere. The problem of matching and comparing 2D spatial functions defined over unit sphere involves manipulating two degrees of freedom (the two angles needed to align the surface of a sphere). The main challenge in these techniques is to identify the characteristics to be represented using spatial functions and to determine an efficient matching procedure to compare two shape signatures. Section 5 describes representative spatial function based techniques in detail.

- **Shape Histograms:** These techniques are based on sampling of points on the surface of the 3D models. Several significant characteristics can be extracted from the set of points obtained. Once these characteristics are determined, they are organized in the form of histograms that store the frequency of occurrence of their values. Then, these histograms are compared using a suitable distance function. The accuracy of these signatures depends on the number of points used. Large number of points result in higher accuracy. However, the efficiency of these signatures varies inversely as the number of points. Thus with an increase in the accuracy, the efficiency decreases. Section 6 describes representative shape histogram based techniques in detail.

- **Section Images:** These techniques use sections of the solids as shape signatures. Solids are sectioned at various places and the sections are then analyzed for similarity. This analysis can be carried out using neural network or by using 2D similarity assessment techniques. As these techniques involve sections, they are not invariant to scaling, translation and rotation and can compare objects only with known orientations. They are well suited for rotational parts due to their rotational symmetry. Techniques that use neural networks do not actually compare the two solids but classify the solids into groups based on group technology codes. Based on the images of the sections they determine the group code to which the part belongs. They are robust but involve training of the network to improve the classification and hence require significant time to implement. Also the number of sections affects the accuracy of comparison. If number of sections being considered is small, then small features on the objects may not be recorded. Section 7 describes representative section image based techniques in detail.

- **Topological Graphs:** These techniques use topological graphs as shape signatures to perform similarity analysis. These graphs usually represent the connectivity information of the boundary of the solid such as the adjacency between faces. The nodes and edges of the graph may carry additional information related to the solid model. The comparison can then be carried out by matching the graphs based on relevant characteristics or by graph isomorphism algorithms. However, comparing graphs is not trivial and requires considerable computational time if graph isomorphism algorithm is used. In order to have sufficient discrimination capability, the graphs need to store as much information as possible. But storing excessive information further increases the computational time. Hence there exists a tradeoff between accuracy of comparison and computational time. In some cases, graph properties such as degree of nodes, number of nodes, number of edges, eigenvalues etc. have been used for comparison. Section 8 describes representative topological graph based techniques in detail.

- **Shape Statistics:** Many shape comparison techniques use basic geometric properties in order to perform coarse comparison between solids. They may also be used to reduce the search space. The commonly used properties include volume, surface area, convex hull volume etc. These numerical values representing statistical properties of the shape form the signature of the solid. Such signatures do not carry any topological information. Such methods cannot provide sufficient discrimination power for detailed comparison but are useful as quick and efficient filters. Section 9 describes representative techniques in this category.

4 Feature-Based Shape Signatures

The first step in the technique described in [20] consists of extracting the features from a B-rep model. This is achieved by constructing cells that are portions of space resembling machining features. Once these cells are obtained following a series of rules, they are mapped to machining features. Then, relevant feature characteristics are used to perform the comparison. For this, a *feature class* is defined as a group of geometrically similar features (i.e. identical topology and relative angles between faces). A *T-group* is a group of features in which the features differ from each other only by translation. Similarly a *S-group* is such that the features belonging to it have the same critical dimensions. Seven characteristics are used for comparison. These include feature existence, feature count, feature direction, feature size, directional distribution, size distribution and relative orientation. Feature existence represents the number of different classes of features present in the object and is expressed as a binary vector of dimension *n* where *n* is the total number of feature classes in the two objects being compared. Each element in the vector assumes a value of 1 if the corresponding type of feature is present in the object or else it is 0. Feature count represents the number of instances for every class of feature in a given 3D object. It is expressed as a vector of dimension *n*. Each element denotes the number of instances of the corresponding feature. Feature direction represents the number of T-groups for every class and is expressed as a vector of dimension *n*. Each element indicates the number of T-groups for the corresponding class. Feature size is similar to feature direction and represents the number of S-groups. For every class of features, directional distribution represents the number of instances of features within a T-group belonging to the class considered. Size distribution is similar to directional distribution and is defined for S-groups. Finally, relative orientation represents the relative orientation between T-groups over all the different classes of features. A weighted distance is used to compare two objects. The characteristics considered in the comparison have to be independent of each other. Only planar and cylindrical surfaces are considered. Objects where the cylindrical features intersect other faces non-orthogonally are also ruled out. This technique also does not account for local feature interactions.

Another technique described in [21], involves feature extraction and comparison to determine similarity between mechanical parts. It defines a Model Dependency Graph for each of the two objects being compared and determines the largest common sub-graph between them to assess similarity. The feature extraction is carried out using FBMach System consisting of a library of machining features. After performing feature extraction, the Model Dependency Graph representing the features and their interactions is defined. The nodes of this graph correspond to features and store attributes of the features as ‘labels’ at the nodes. Thus model dependency graph $G = (V, E)$ comprises of a set of nodes $V = \{f_0,\ldots,f_n\}$ where f_i is a machining feature of the solid. An edge between the two nodes exists if the corresponding features f_i and f_j have non-zero intersection between them. Thus $E = \{\{f_i, f_j\}$ such that $\mathrm{vol}(f_i) \cap \mathrm{vol}(f_j) \neq \varnothing\}$. To compare the two solids, the largest common sub-graph (LCS) between the two model dependency graphs needs to be determined. The problem of exactly determining the largest common sub-graph, however, is NP-complete and hence a hill climbing/ gradient descent algorithm is used to obtain a large enough sub-graph. The proposed algorithm involves assigning random mapping between the nodes of the two graphs initially, and then swapping the mappings such that evaluation function assumes the lowest value. The evaluation function H is the count of the number of mismatched edges. The measure of similarity is given as $\mathrm{H}^{*} = \frac{\min\{H_1,...,H_n\}}{|E_1|}$ where $H_1,\ldots,H_n$ are the final values of H from up to n random restarts of the algorithm and $|E_1|$ is the number of edges in the smaller of the two graphs.

This technique provides means for determining objects having similar machining features. However, the Model Dependency Graph generated using this method is not unique for a given solid. This is because the features can be constructed in multiple different ways and in different order as shown in Figure 3. In fact, the issue of multiple feature interaction is a common problem to all existing feature-based similarity assessment methods. This technique considers only feature interaction and does not account for feature size and orientation.

The technique described in [22] is based on a graph representation of the input 3D models. This graph representation is used as the shape signature for the model. Let us consider two objects, m and m'. Then, $c(m)$ and $c(m')$ will be the value of a characteristic for the object m and m'. The equivalence relation $E_i(m, m')$ is true iff $c(m)= c(m')$ (i.e., the two objects are equivalent with respect to the characteristic considered). So, depending on the number of properties or characteristics considered, many different equivalence relations can be defined. Let R_i be an equivalence relation. R_i is valid if R_{i+1} is valid, as the former is contained in the latter. To compare the two objects it is necessary to define $M(m,m')$ as the biggest value of i for which $R_i(m,m')$ holds true. So, if we consider three objects m, m' and m'', and if $M(m,m')> M(m,m'')$ then m is closer to m' than m is to m''. Given the previous definitions, a tree can be built whose leaf nodes represent the parts in the database being compared and the rest of the nodes represent the equivalence relations. Once the tree is obtained, it is possible to obtain the degree of similarity between two objects by calculating the value of M. In [22], the main focus is on the manufacturing aspects of the object represented by the 3D model (see also [23]). For each model a graph called *design signature* is constructed. The nodes represent some characteristics of the design, while the edges represent the relationships among these characteristics. An application is provided in [22], where nodes represent features and edges their interactions. The nodes are labeled with a number of parameters, such as type of feature and machining direction. The edges are labeled depending on the type of intersection occurred: a description of the types of intersections is provided in [22]. The equivalence relation considered in this application is isomorphism between two graphs. It is usually an expensive task, but in this case it is made easier from the labeling of nodes and edges. In fact the labeling allows easier matching of sub graphs.

In the technique described in [24], different attributes of features such as feature type, machining type etc. are stored in Attribute Type table. A qualitative matrix is used to record all the feature interactions. By searching through the Attribute Type table and qualitative matrix similar models can be retrieved.

Techniques described in this section perform similarity assessment based on the features of the parts and their characteristics. Hence the techniques have been primarily developed for product design and manufacturing.

5 Spatial Function Based Shape Signatures

5.1 Local Curvature Distributions Over Mesh Representations

In this technique, a spherical representation that stores the curvature distribution of 3D surfaces of an object is used as signature [25]. The solids to be compared must have a genus of zero. To generate the representation, a tessellated sphere is deformed such that it closely approximates the shape of the object. Each node of the tessellated sphere has three adjacent nodes. A local regularity constraint is introduced during the deformation to ensure that each mesh is similar to others in area. According to this condition, the projection of each node on the triangle formed by the adjacent three nodes should coincide with the centroid of that triangle. This representation then yields the shape signature of the object. Two types of forces are used to perform the deformation. One type of force tries to bring the mesh nodes closer to the surface, while the other helps in maintaining the local regularity constraint. The algorithm for deforming the sphere is based on combining these two forces between the solid model and the spherical mesh. After a uniform surface mesh is obtained, the curvature at every node of the mesh is computed using three discrete nodes in the neighborhood of that node. Once the curvature function is defined, one of the two objects is rotated such that it aligns with the other. The distance between the two objects computed using L_p norm. Alternatively Hausdorff distance may be used to compute the difference [26].

The shape similarity assessment in this case is invariant under translation, rotation and scaling as the curvature depends only on the relative locations of the four nodes that are used to define the local curvature function k. As the distance is computed using L_p norm, it obeys the positivity property. It also satisfies the identity, symmetry, and triangle inequality properties [27]. The distance between the two shapes can be computed in time $O(n^2)$ where n is the number of nodes on the sphere. However, this technique is restricted to solids having zero genus (i.e. solids without holes). This is a serious restriction considering that holes are a common feature in CAD models. Also, the mesh is an approximate representation of the solid and the accuracy depends on the number of tessellations. As the mesh becomes finer, accuracy increases but so does the computational time. A technique similar to this one is described in [28].

The above technique has not been developed for product design and manufacturing, but it could be applied to this domain. In particular, this technique could be used in applications where curvature plays a major role. Nevertheless the restriction to solids without holes is a major limitation.

5.2 Slope Diagram Representations

The technique described in [29] uses the slope diagram representation [30] of a convex polyhedra [31] and uses mixed volumes and volumes based on Minkowski addition [32] to define the similarity measure. The definition and mathematical representation of mixed volume can be found in [29]. A *slope diagram representation* (SDR) is one where a face is represented on the unit sphere by a spherical point, which is an intersection of its normal with the unit sphere. An edge is represented by a spherical arc, which is an arc of a great circle joining the points representing the two faces that share the edge. A vertex is represented by a region of the sphere known as a spherical polygon bounded by the spherical arcs corresponding to the edges sharing the vertex. Let P and Q be the two objects to be compared. Then a rotation r is applied to the SDR of Q while the SDR of P is kept fixed. Such a rotation r can be determined by identifying a set of finite number of critical rotations. Such critical rotations include situations where spherical points of the rotated SDR of Q intersect spherical arcs or points of the SDR of P. These rotations minimize the objective functions defined based on volumes and mixed volumes. This technique, which is defined for convex shapes, is invariant with respect to translation and rotation. However, a considerable computational effort is needed to determine the set of finite critical angles.

The technique described in [29] is not developed for product design and manufacturing, but it could be applied to this domain. However, this technique is restricted to convex polyhedra. Hence, it is limited in scope.

6 Shape Histogram Based Shape Signatures

The technique described in [33] computes shape distributions of solid models using shape functions and then compares these shape distributions to assess similarity. Once a set of random points is obtained on the surface of the solid model, different shape functions are used to compute shape distributions for the solid model. Typical shape functions include

- D1: Computes the distance between a fixed point and a random point. This shape function is not suitable as the chosen fixed point is usually not invariant to rotation or translation.
- D2: Computes the distance between two random points. This function is invariant to rotation and translation and is robust.
- D3: Computes the square root of the area of triangle generated by three random points. This function is also invariant to translation and rotation but not as efficient as D2.
- D4: Computes the cube root volume of the tetrahedron generated by selecting four random points. This method is computationally inefficient even for lesser number of points.
- A3: Computes the angle between three random points. This function is invariant to translation, rotation and scaling but it is not very robust.

Out of these the D2 shape function has been found to be most suitable for computing shape distributions due to its robustness and efficiency along with invariance to rotation and translation. After calculating the distances between random points, they are normalized using the mean distance. The shape distribution is the histogram that measures the frequency of occurrence of distances within a specified range of distance values. Once the shape distributions are generated the distance between the two solid models is computed using L_N norm. Usually L_2 norm is used for comparison. Other distances such as Earth Mover's distance [15], or Match distances [34,35] can also be used.

This technique is robust and efficient. Also there is no restriction on the type of solid models that can be compared. However, as this method involves generating random points on the surface of the solid, it fails to satisfy conditions of identity and symmetry. As the number of points increases the comparison is more robust, but the computational time increases. Furthermore as objects become more and more complex, the shape distributions tend to assume similar shape. This results in inaccurate comparison of solid models. Figure 4 shows three parts and their corresponding D2 shape signatures. Based on our implementation of the algorithm it can be seen that *heat_exchanger1* is more similar to a *grip* than *heat_exchanger2*. Thus this technique has limited discrimination capability.

An extension of the previous technique is described in [36]. The procedure for generating random points on the surface as well as the shape function used is the same. However, instead of computing a single shape distribution for each solid model, this method computes four different shape distributions based on in/out classification of the line joining the random points whose length is the distance measure. The first distribution is the same as in the previous method. The second distribution takes into consideration all the lines joining the random points that lie inside the solid model. The frequency of occurrence of the length of these lines is also measured. The third distribution accounts for all the lines that lie outside the solid. Finally, the fourth distribution includes those lines that lie partially inside the solid and partially outside. The distributions are then compared using L_2 norm. This technique aims at improving the ability of the previous one to distinguish between complex parts having detailed features. However, it fails to satisfy the properties of identity and symmetry for the same reasons as the previous method does. Moreover its computational efficiency is low, as it involves determining whether a line lies inside, outside or partially inside a solid.

The similarity assessment techniques described in this section can detect gross shape similarity. Hence these techniques could be used to perform a pruning on the database to search for similar parts in design and manufacturing applications.

7 Section Images Based Shape Signatures

Manual classification and coding of parts for group technology applications is time-consuming and prone to errors. In [37] a neural network system has been proposed for classifying parts based on bitmaps of the part drawings. The neural network consists of number of layers of neurons, which include an input layer, some hidden layers and an output layer. The theory of neural network systems is described in [38]. The input to the neural network system is a vector I containing bit data that represents the image of a part drawing. For every input i, there is a neuron with a weight vector W^i attached to it. The input to each neuron is the dot product of these two vectors. The output of the neuron is a vector O_i corresponding to the input and activation function f_s. The algorithm for determining the output is described in [39]. The maximum number of neurons that can be used in any given layer is defined using Kolmogorov's theorem [40]. The vector element of O_i with the highest value represents the group to which the part belongs. The output in this case is an Opitz code used to classify rotational parts based on characteristics such as length to diameter ratio. At the beginning, random values are assigned to the weight vectors. The network is then trained using standard inputs for which target outputs have been identified. If the difference between the actual output and target output is above the threshold value then the weight vectors are adjusted such that the error is reduced.

This technique involves classification of part drawings and hence it does not account for rotation or translation of the solid model. It classifies the part drawings using group technology. There is no direct comparison between the part drawings: they are classified based on their characteristics such as L/D ratio, presence of holes etc. However, the solid models available in the databases or the Internet have arbitrary orientation and hence this method will require manual intervention to identify the part drawing with desired orientation.

The technique described above involves classification of rotational parts using neural network system. In [41] this classification has been extended to include 3D parts based on their binarized part drawing image. A back-propagation neural network system has been proposed to classify the 3D parts into a number of predetermined part families. The theory of backpropagation neural network is explained in [42]. Also some concepts related to the neural network such as learning rate, number of neurons in the hidden layer and number of hidden layers are discussed in [41]. The modified technique described in [41] is similar to the one discussed before and uses gradient search procedure to determine the weight vectors such that they reduce the error between the target value and actual value. The learning rate should allow the learning algorithm to converge to minimum error without oscillation of the network and without getting trapped in a local minimum. The local minimum can be avoided by adjusting the value of the momentum. The momentum is similar to physical momentum and allows the network to bounce from a local position and seek a better solution. The formation of part families depends on the predetermined number of part families. Also the learnability of the group increases as the number of hidden neurons increases. This technique involves classification of parts using neural network and hence suffers from the same problems associated with the previous technique. However, it provides an insight into the various parameters that affect the performance of the neural network system used to classify the part drawings.

These techniques have been applied to product design and manufacturing applications. They are specifically used to classify the parts in a database to reuse design information.

8 Topological Graph Based Shape Signatures

8.1 Model Signature Graphs

In [43] Model Signature Graphs have been proposed for topological comparison of solid models. They are an extension of Attribute Adjacency Graphs, mentioned in [44], and are introduced in order to consider curved surfaces. Model Signature Graphs are constructed from boundary representation of the solid. Each node in the model signature graph represents a face of the solid model. There exists an edge between two nodes of the graph if the corresponding faces are adjacent. This graph forms the shape signature of the solid model. Along with the connectivity information between the faces, the identifier for the face (planar, conical, etc.), mathematical representation of the surface, surface area and set of surface normals can also be stored at the nodes. The edge of a Model Signature Graph represents the edge between two adjacent faces of the solid model. Identifier for the edge, concavity/convexity of the edge, mathematical representation of the edge and length of the curve can also be stored at the edge [43]. This additional information helps in more accurate comparison of the solid models. However, in the current implementation the edge angle information is not stored. Thus two simple objects may have the same Model Signature Graph. Once a Model Signature Graph is constructed, the solid models are compared using spectral graph theory. The eigenvalues of the Laplacian matrix [45] are used in the comparison. The eigenvalues of the Laplacian are strongly related to other graph properties such as the graph diameter. The *graph diameter* is the largest number of vertices, which must be traversed, in order to travel from one vertex to another in the graph.

Another technique proposed for comparing the graphs is the use of graph invariance vectors [46]. Graph invariance vectors are vectors whose elements are graph invariants. The vectors are then compared using L_2 norm to determine similarity between the graphs and hence the solid models. The graph invariants that form the graph invariance vectors include node and edge count, minimum and maximum degree of the nodes, median and mode degree of the nodes, and diameter of the graph. The use of graph invariance vectors improves the efficiency of the method. However it results in decrease in the accuracy of comparison.

This technique has been applied to mechanical parts and is applicable to product design and manufacturing domain.

8.2 Multiresolutional Reeb Graphs

In this technique, the skeletal and topological structure of the 3D model is defined by Multiresolutional Reeb Graphs [47], which are used to compare the 3D objects. Reeb graphs have already been used in technical applications such as modeling 3D shapes [48,49]. First, the Reeb Graph is defined on the input object, which is a triangulated solid. It is obtained by defining a suitable function over the 3D object considered. An example of a suitable function is geodesic curvature. In general, the similarity function can be chosen depending on the particular topological properties selected. Then the function value range over the object is split into a number of sub ranges. This number is chosen depending on the desired level of resolution. A part of the object will correspond to each sub range. This part will be made of several connected regions. Every connected region will correspond to a node of the Reeb graph, and the adjacent nodes will be connected by edges. The Reeb graph for the two models is created in $O(V\log(V))$ time, where V is the number of vertices in the mesh of the solid. Now the two corresponding graphs need to be matched. Corresponding nodes are matched in such a way so as to maximize a similarity function. In fact, the function is chosen such that the similarity between the two objects increases with its value. Thus, the best possible matching among the pair of nodes of the two graphs will maximize the value of the similarity function. Once this best matching is found, the value of similarity function between the two objects will yield the degree of similarity. At this stage the similarity function values corresponding to the best matching found are computed for every matched pair of nodes. They are then summed over the two objects, yielding a similarity value for the two objects being compared. The higher is the value, the more similar are the objects. Self-comparison of an object yields a value of 1, which is the maximum possible value. It takes $O(M(N+M))$ to match and assess similarity, where N and M are the number of nodes in the two graphs with $M<N$. So, with the increase in the accuracy of mesh and in the resolution of the Reeb graph, the efficiency decreases. Furthermore, if the function used to define Reeb graph is based on geodesic distance it is not very robust with respect to small deformations on the surface. It is necessary to choose both a robust and efficiently computable function, which is not a trivial task. Finally, from the experimental results reported in [50] it can be observed that this method is not invariant to Euclidean transformations (e.g., rotation, translation, scaling). Thus a given model when compared with its scaled, translated or rotated version will not yield a similarity value of 1.

This technique has not been applied specifically to product design and manufacturing domain. However, the choice of the function to construct the Reeb graph obviously affects the resulting graph. Hence Reeb graph is a flexible tool that can be used to assess similarity in several applications of product design and manufacturing, by choosing an appropriate function.

8.3 Graphs of Aligned Models

In [51] a similarity assessment technique has been described based on the information provided by Brep model and CSG tree termed as T0 tree. T0 tree is a specialized linear tree whose primitives are all sweeps obtained by sweeping a face in space along a profile. Initially, in the preprocessing stage, the T0 tree is used to determine the major sweep directions. Each of the sweep directions is expressed as a double (v_1, v_2), where v_1 is the normal vector of a set of parallel faces and v_2 is the vector indicating the direction by which these parallel faces are organized in space. The set of parallel faces having normal vector v_1 is called layer faces F. Initially, for each pair of matched major sweep directions, the layer faces are matched using their normalized areas and their offsets along v_2. If p_1 is a point on plane P_1 and p_2 a point on P_2 such that $p_2 = p_1 + dv$, where d is a real number and v is a unit vector, then d is called offset from P_1 to P_2 along v. Once the layer faces of major sweep directions are matched such that there exists a one-to-one mapping, the objects are rotated so that the unit vectors v_2 match. Additional pair of faces, which do not have normal along v_2 are matched by attributed string matching algorithm [52], to completely align the two models. After rotating the layer faces to the correct orientation, initial matched sub-graphs of the layer faces are obtained. The nodes in the graph represent faces while the edges represent the edges joining those two faces. Once the layer faces are matched faces adjacent to matched layer faces are analyzed. If they match then they are included in the matched sub-graphs by expanding the sub-graphs. All possible matching sub-graphs are generated for all the major sweep directions and the Brep matching coefficient is computed. This technique has the following restrictions on the models it can compare. All the solid models should have at least one or more major sweep directions. Also the models must be polyhedral.

This technique has been applied to product design and manufacturing on models that comply with the restrictions mentioned above.

9 Shape Statistics

The technique described in [53], uses global shape metrics such as surface area/volume ratio, number of holes, compactness, and crinkliness to perform similarity assessment. These metrics are orientation independent and are extracted from a STL file. Compactness is the non-dimensional ratio of the square of the volume over the cube of the surface area while crinkliness is the surface area of the model divided by surface area of a sphere having the same volume. They are calculated for all the solid models and are stored as searchable entries in a database. To analyze the performance of the search engine, similarity matrices based on human perception of similarity have been generated. In [54], four new filters for shape matching have been proposed. These are based on the coefficient of surface area and convex hull of the solid model. The convex hull based filters include hull crumpliness, hull packing and hull compactness. Hull crumpliness is the ratio of surface area of object to surface area of its convex hull. Hull packing is the percentage of the convex hull volume not occupied by the original object. Hull compactness is the non-dimensional ratio of the cube of the surface area of convex hull to the square of the volume of convex hull.

The filters proposed in this technique are useful for pruning out parts from a large database. They do not have a high discrimination power. These filtering techniques have been applied to large databases of mechanical parts.

10 Conclusions

Table 1 presents summary of representative algorithms described in this paper along with their key features. Based on the survey of algorithms described in this paper, we have come to the following conclusions:

1. Shape signatures are abstractions of 3D shapes, and therefore they often possess reduced discrimination capabilities. Furthermore, computing signatures and matching them requires a certain amount of computational effort. Several different shape signatures have been proposed and their discrimination capabilities and computational performance have been illustrated through datasets selected by their developers. However, many of these signatures were not tested on a particular application so their effectiveness cannot be assessed based on their success in a particular application. Currently, it is very difficult to compare shape discrimination capabilities of two different signatures. Furthermore, it is difficult to predict how well a particular shape signature will perform in a given application. In order to effectively select and utilize existing shape signatures we need to develop an improved understanding of discrimination capabilities of existing signatures.

2. In a given application, usually a single signature will not be good enough to provide adequate discrimination and computational performance. A signature that performs high-degree of abstraction usually has limited discrimination capabilities. On the other hand, a signature that performs lesser degree of abstraction is usually computationally not very efficient. Therefore usually first using a signature that is computationally efficient but produces few false positives, followed by a signature that is computationally less efficient but able to eliminate

false positives is a good strategy. Furthermore, we may sometimes need to simplify models so that signatures can be applied only on the relevant portions of the models.

3. Popularity of 3D scanning and CAD models has resulted in a large number of CAD models that are publicly available. However, they lack the application specific measures of similarity associated with them. Therefore, performing evaluation of existing shape signatures is a difficult task. In order to perform evaluation of shape signatures, we need to develop publicly available benchmarking databases that contain domain specific evaluation metric.

4. As mentioned before, GT coding schemes have been used primarily for classification and retrieval of mechanical parts. Although the GT approach has been used with some success in past, it has several limitations. Describing designs as short strings creates a coarse classification scheme, which limits the kinds of real-world retrieval problems for which the approach can be useful. Moreover, these techniques were developed prior to the advent of inexpensive computer technology; hence, they are not rigorously defined and are intended for human, not machine interpretation—and this can cause difficulty in automating the generation of GT codes.

5. Many previous approaches have favored symmetric distance measures. However, distance measures that are not symmetric in nature are of interest as well because of the following reasons. Let *A* and *B* be two objects. Let *A* contain subset of features of *B*. In this case, *B* can be used to estimate cost of *A* by simply deleting extra processing steps (i.e., steps corresponding to features that are not present in *A*) from *B*. So distance of *A* from *B* should be small. On the other hand *A* cannot be used to estimate cost for *B*. So distance of *B* from *A* should be very large. Therefore various asymmetric distance measures should be studied to better understand their properties.

6. Most researchers only report worst-case asymptotic complexity for computation and matching algorithms for shape signatures. However, in practice to select a particular shape signature we will need to know average case run-time for the problems on which we plan to run the computation and matching algorithms. Therefore, the performance of both computation and matching algorithms needs to be evaluated by running them on a wide variety of models of differing sizes and complexity.

We believe that 3D shape similarity assessment is a fundamental problem of growing importance due to pervasive use of 3D models. Solution to this problem will provide solution to many existing needs and enable development of many new applications.

Acknowledgements. This research has been supported in part by the NSF Grant SES9906265. Opinions expressed in this paper are those of authors and do not necessarily reflect opinion of the National Science Foundation.

References

[1] U. Alva and S.K. Gupta. Automated design of sheet metal punches for bending multiple parts in a single setup. *Robotics and Computer Integrated Manufacturing*, 17(1/2):33-47, 2001.

[2] S.K. Gupta and D.A. Bourne. Sheet metal bending: Generating shared setups. *ASME Journal of Manufacturing Science and Engineering*, 121:689--694, November 1999.

[3] J.L. Burbidge. The Introduction of Group Technology. Heinemann, London ,1975.

[4] S. Santini and R. Jain. Similarity matching. In *2nd Asian Conference on Computer Vision*, Singapore, December 1995.

[5] K. Arbter, W.E. Snyder, H. Burkhardt, and G. Hirzinger. Application of affine-invariant fourier descriptors to recognition of 3-D objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):640-647, July 1990.

[6] E.M. Arkin, L.P. Chew, D.P. Huttenlocher, K. Kedem, and J.S. Mitchell. An efficiently computable metric for comparing polygonal shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(3):209-216, March 1991.

[7] I. Young, J. Walker, and J. Bowie. An analysis technique for biological shape. *Information and Control*, 25(4):357-370, August 1974.

[8] Y. Lin, J. Dou, and H. Wang. Contour shape description based on arch height function. *Pattern Recognition*, 25(1):17-23, 1992.

[9] H. Alt and L.J. Guibas. Discrete geometric shapes: Matching, interpolation, and approximation: A survey. Technical Report B96-11, EVL-1996-142, Institute of Computer Science, Freie Universität Berlin, December 1996.

[10] S. Loncaric. A survey of shape analysis techniques. *Pattern Recognition*, 31(8):983-1001, August 1998.

[11] R.C. Veltkmap. Shape matching: Similarity measures and algorithms. In *International Conference on Shape Modeling and Applications*, Genova, Italy, May 2001.

[12] D.A. Keim. Efficient geometry-based similarity search of 3D spatial databases. In *ACM SIGMOD International Conference on Management of Data*, Philadephia, PA, June 1999.

[13] F. Arman and J. Aggrawal. Model-based object recognition in dense-range images - A review. *ACM Computing Surveys*, 25(1):5-43, March 1993.

[14] L.P. Chew, M. T. Goodrich, D.P. Huttenlocher, K. Kedem, J. M. Kleinberg, and D. Kravets. Geometric pattern matching under Euclidean motion. *Computational Geometry*: *Theory and Application*, 7(1-2):113-124, January 1997.

[15] Y. Rubner, C. Tomasi, and L. Guibas. A metric for distributions with applications to image databases. In *6th International Conference on Computer Vision*, Bombay, India, January 1998.

[16] K. Siddiqi, A. Shokoufandeh, S. J. Dickenson, and S.W. Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):13-32, November 1998.

[17] N. Thacker, P. Riocreux, and R. Yates. Assessing the completeness properties of pairwise geometric histograms. *Image and Vision Computing*, 13(5):423-429, June 1995.

[18] D. Zhang and M. Hebert. Harmonic maps and their applications in surface matching. In *IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, June 1999.

[19] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Kauai Marriott, Hawaii, December 2001.
[20] M. Ramesh, D. Yip-Hoi, and D. Dutta. Feature-based shape similarity measurement for retrieval of mechanical parts. *Journal of Computing and Information Science in Engineering*, 1(3):245-256, September 2001.
[21] V.A. Cicirello and W.C Regli. Machining feature-based comparisons of mechanical parts. In *International Conference on Shape Modeling and Applications,* Genova, Italy, May 2001.
[22] A. Elinson, D. S. Nau, and W. C. Regli. Feature-based similarity assessment of solid models. In *4th Symposium on Solid Modeling and Applications*, Atlanta, GA, May 1997.
[23] A. Elinson, D. Nau, and W. C. Regli. Solid similarity measurements. Technical report ISR-TR96-63, Institute for Systems Research, University of Maryland. 1996.
[24] G. Srinivas, D.E. Fasse, and M.M. Marefat. Retrieval of similarly shaped parts from a CAD database. In *IEEE International Conference on Systems, Man, and Cybernetics,* San Diego, CA, October 1998.
[25] M. Hebert, K. Ikeuchi, and H. Delingette. A spherical representation for recognition of free-form surfaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(7):681--690, July 1995.
[26] D. Huttenlocher and K. Kedem. On computing the minimum Hausdorff distance for point sets under translation. In *6th ACM Symposium Computational Geometry*, Berkeley, CA, June 1990.
[27] H. Shum, M. Hebert, and K. Ikeuchi. On 3d shape similarity. In *IEEE Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, June 1996.
[28] J. Schwarz and M. Scharir. Identification of partially obscured objects in two and three dimensions by matching noisy characteristic curves. *International Journal Robotics Research*, 6(2):29-44, Summer 1987.
[29] A. V. Tuzikov, J.B.T.M. Roerdink, and H.J.A.M. Heijmans. Similarity measures for convex polyhedra based on Minkowski addition. *Pattern Recognition*, 33(6):979-995, June 2000.
[30] P. K. Ghosh and R. M Haralick. Mathematical morphological operations of boundary-represented geometric objects. *Journal of Mathematical Imaging and Vision*, 6(2-3):199-222, June 1996.
[31] B. Grunbaum. *Convex polytopes*. Interscience publishers, 1967.
[32] P. K. Ghosh. A unified computational framework for Minkowski operations, *Computers and graphics*, 17(4):357-378, 1993.
[33] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. In *International Conference on Shape Modeling and Applications*, Genova, Italy, May 2001.
[34] H. C. Shen and A.K.C. Wong. Generalized texture representation and metric. *Computer Vision, Graphics and Image Processing*, 23(2):187-206, August 1983.
[35] M. Werman, S. Peleg, and A. Rosenfeld. A distance metric for multidimensional histograms. *Computer Vision, Graphics and Image Processing*, 32(3):328-336, December 1985.

[36] Cheuk Yiu Ip, D. Lapadat, L. Sieger, and W.C. Regli. Using shape distributions to compare solid models. In *7th Syposium on Solid Modeling and Applications*, Saarbrücken, Germany, June 2002.
[37] S. Kaparthi and N.C. Suresh. A neural network system for shape based classification and coding of rotational part. *International Journal of Production Research*, 29(9):1771-1784, 1991.
[38] T. Khanna. *Foundations of Neural Networks*. Addison Wesley, 1990.
[39] R.P. Lippmann. Pattern classification using neural networks. *IEEE Communications Magazine*, 27(11):47-64, November 1989.
[40] J. A. Anderson and E. Rosenfeld. *Neurocomputing – Foundations of Research*. MIT Press, Cambridge, MA, 1988.
[41] Y. Chung and A. Kusiak. Grouping parts with a neural network. *Journal of Manufacturing System*, 13(4):262 – 275, 1994.
[42] R. Hecht-Nielsen. Theory of backpropagation neural networks. In *IEEE International Joint Conference on Neural Networks*, Washington D.C., June 1989.
[43] D. McWherter, M. Peabody, A. Shokoufandeh, and W.C. Regli. Database techniques for archival of solid models. In *6th ACM/SIGGRAPH Symposium on Solid Modeling and Applications*, Ann Arbor, MI, June 2001.
[44] S. Joshi, and T. C. Chang. Graph-based heuristics for recognition of machined features from a 3D solid model. *Computer Aided Design*, 20:58-66, 1988.
[45] F.R.K. Chung. Spectral graph theory. In *American Mathematical Society's Regional Conference Series in Mathematics No. 92*, Providence, RI, 1997.
[46] D. McWherter, M. Peabody, W.C. Regli, and A. Shokoufandeh. Transformation invariant shape similarity comparison of solid models. In *ASME 6th Design for Manufacturing Conference*, Pittsburgh, PA, Sept 2001.
[47] G. Reeb. On the singular points of a completely integrable Pfaff form or of a numerical function. *Comptes Rendus Acad. Sciences*, Paris, France, 222:847-849, 1946.
[48] F. Lazarus and A. Verroust. Level set diagrams of polyhedral objects. In *ACM Solid Modeling*, Ann Arbor, MI, June 1999.
[49] S. Takahashi, Y. Shinagawa, and T. L. Kunii. A feature-based approach for smooth surfaces. In *4th ACM Symposium on Solid Modeling and Applications*, Atlanta, GA, May 1997.
[50] M. Hilaga, Y. Shinagawa, Taku Kohmura, and T. L. Kunii. Topology matching for fully automatic similarity estimation of 3D shapes. In *ACM SIGGRAPH*, Los Angeles, CA, August 2001.
[51] T.L. Sun, C.J. Su, R.J. Mayer, and R.A. Wysk. Shape similarity assessment of mechanical parts based on solid models. In *Design for Manufacturing Symposium, ASME Design Technical Conference*, Boston, MA, September 1995.
[52] W.H. Tsai and S.S. Yu. Attributed string matching with merging for shape recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):453-462, July 1985.
[53] H J Rea, J R Corney, DER Clark, J Pritchard, ML Breaks, and RA MacLeod. Part-Sourcing in a Global Market. In *2001 International Conference on eCommerce Engineering, IceCE 2001*. Xi'an, P.R. China, September 2001.
[54] R Sung, H J Rea, J R Corney, DER Clark, J Pritchard, ML Breaks, and RA MacLeod. Assessing the Effectiveness of Filters for Shape Matching. In *2002 ASME International*

Mechanical Engineering Congress & Exposition, IMECE '02. New Orleans, LA, November 2002.

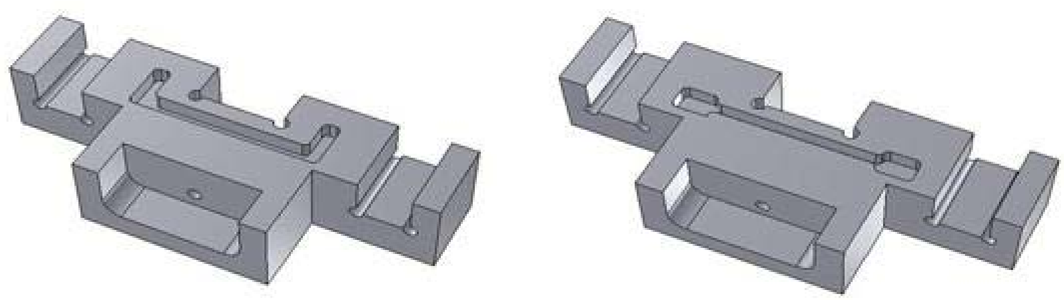
Captions for Figures

Figure 1: An Example of Using a Similar Part for Cost Estimation
Figure 2: An Example of Using an Existing Shaft for Reducing Part Proliferation
Figure 3: An Example Depicting Different Ways of Representing Features
Figure 4: An Example Indicating the Low Discrimination Capability of Shape Distributions

Table 1 Summary of Comparison of Representative Algorithms

No	Shape Signature	Methods	Id*	Sym*	Inv*	Robustness/ Sensitivity	Efficiency	Restrictions
1.	Feature Based	Technique described in [20]	No	Yes**	Yes	Robust but not sensitive to changes in feature locations	ND	Planar and cylindrical surfaces only
		Technique described in [21]	No	Yes**	Yes	Robust but not sensitive to changes in feature locations	Not efficient if graph matching is done exactly Efficient if graph matching is done approximately	Only works if feature extraction is successful
		Technique described in [22]	No	Yes	Yes	Robust but not sensitive to changes in feature locations	Not efficient if graph matching is done exactly Efficient if graph matching is done approximately	Only works if feature extraction is successful
2.	Spatial Function Based	Local Curvature Distributions Over Mesh Representations [25]	Yes	Yes	Yes	Not robust with respect to addition of mechanical features such as slots and blind holes	Efficient for coarse mesh	Solids of zero genus
		Slope Diagram Representations [29]	Yes	Yes	Yes	ND	Not efficient due to computation of critical angles in model alignment	Convex polyhedra
3.	Shape Histogram Based	Shape Distributions [33]	No	No	Yes	Robust but not sensitive for parts with intricate details	Efficient for small number of points on surface	Low discrimination capability
		Shape Distributions [36]	No	No	Yes	Robust and sensitive	Not efficient due to line in/out classifications	Not clear how changes in shape distributions relate to changes in shapes
4.	Section Images Based	Neural Network based technique [37]	NA	NA	No	ND	Implementation required for determining efficiency and time for training network	Works with 2D drawings of rotational parts
		Neural Network based technique [41]	NA	NA	No	ND	Implementation required for determining efficiency and time for training network	Works with 2D drawing
5.	Topological Graph Based	Model Signature Graphs [43]	No	Yes	Yes	Not robust to minor feature changes like chamfering and also not very sensitive to changes in feature location	Reasonably efficient	Limited discrimination capability
		Multiresolutional Reeb Graphs [47]	No	Yes	No	Not sensitive	Efficient for small number of vertices	Limited discrimination capability
		Graphs of Aligned Models [50]	No	No	Yes	ND	Reasonably efficient	Polyhedral solids with one or more sweeps
6.	Shape Statistics	Technique described in [53]	No	Yes	Yes	Robust but not sensitive to feature location	Efficient	Low discrimination capability
		Technique described in [54]	No	Yes	Yes	Robust but not sensitive to feature location	Efficient	Low discrimination capability

* - Id = Identity, Sym = Symmetry, Inv = Invariance
** - Depends on the feature extraction method used
NA – Not Applicable
ND – Not Described



(a) New Object (b) Previously Machined Object

Figure 1 An Example of Using a Similar Part for Cost Estimation

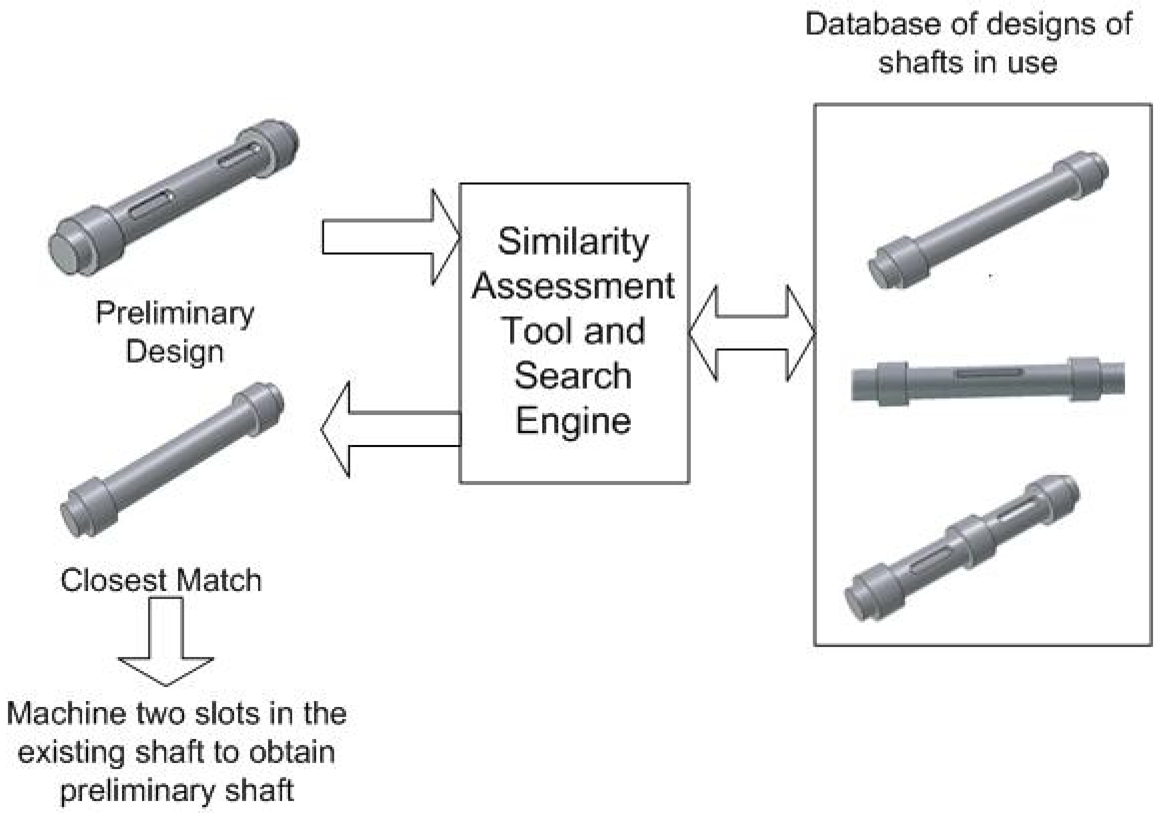


Figure 2 An Example of Using an Existing Shaft for Reducing Part Proliferation

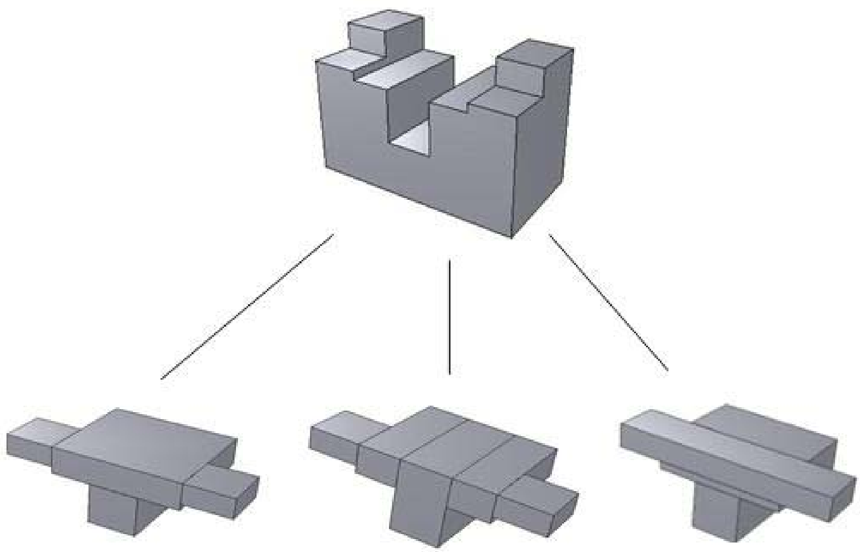


Figure 3 An Example Depicting Different Ways of Representing Features

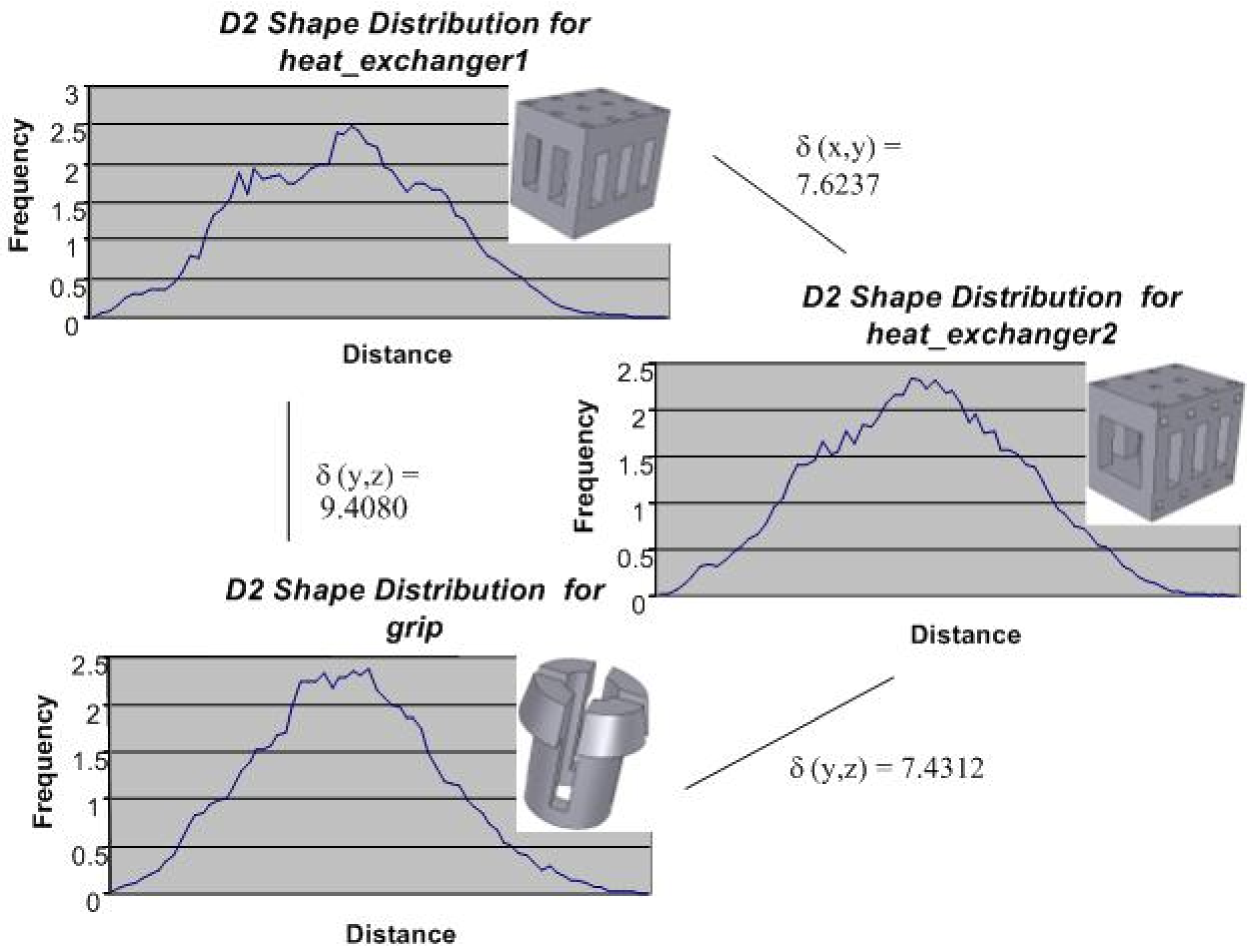


Figure 4 An Example Indicating the Low Discrimination Capability of Shape Distributions