**Proceedings of the TMCE 2004**, April 12-16, 2004, Lausanne, Switzerland, Edited by Horváth and Xirouchakis


# A MULTI-SCALE HIERARCHICAL 3D SHAPE REPRESENTATION FOR SIMILAR SHAPE RETRIEVAL

**Natraj Iyer**
PRECISE, School of Mechanical Engineering
Purdue University, West Lafayette IN
United States
iyern@purdue.edu

**Subramaniam Jayanti**
**Kuiyang Lou**
**Yagnanarayanan Kalyanaraman**
**Karthik Ramani**
PRECISE, School of Mechanical Engineering
Purdue University, West Lafayette IN
United States
{jayantis, kuilou, yagna, ramani}@purdue.edu

## 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 proposes a novel shape representation that can be used for similar 3D shape retrieval. The shape representation is hierarchical in nature and represents the shape at multiple levels of detail. It also preserves geometry and topology of 3D models with good fidelity. Additionally, the representation has local shape information such as lengths and angles which can be used for finer discrimination between 3D shapes. An example demonstrating similar shape retrieval is provided.*

## KEYWORDS

Shape Similarity, Design Reuse, Design Repository, Shape Representation, CAD, Skeleton, Graph

## 1. INTRODUCTION

Computer-aided design (CAD) and solid modeling have become essential elements in modern design and manufacturing. 3D CAD models of product information are ubiquitous throughout the lifecycle of a product. In order for companies to remain competitive, reducing product development costs and lead time are critical success factors. Conservative estimates suggest that about 75% of all design activity is case-based design, i.e. reuse of previous design knowledge to address a new design problem. Two types of knowledge reuse have been identified:

a. Internal knowledge reuse: a designer reusing knowledge from his/her own personal experiences (internal memory)

b. External knowledge reuse: a designer reusing knowledge from an external knowledge repository (external memory).

Whereas internal knowledge reuse is effective, external knowledge reuse often fails. One of the key reasons is that there are no mechanisms from both the information technology and organizational viewpoints for capturing, finding, and retrieving reusable knowledge (Fruchter, R., and Demian, P., 2002). It is estimated that designers spend about 60% of their time searching for the "right" information (Leizerowicz, W., et al., 1996). 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." Nevertheless, fire-fighting has become the de facto process for developing new products in industry (Repenning, N.P., 2001). Design and associated knowledge reuse is the key to reducing new product development time and cutting down costs.

Text based searching of 3D models is not robust primarily for the following reasons:

a. All models will not have a well-defined attached context.

b. Keywords such as project names or part names may be unknown to the user.

c. Context may be too narrow or too broad to retrieve relevant models.

d. 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. The motivation for this research is based on enabling new possibilities and associated applications, specifically:

a. Allowing for the retrieval of past knowledge and previous parts that are similar.

b. Enabling better quotation support, including reducing risk and improving time of response.

c. Being able to locate suppliers through neutral secure locations where geometry-related manufacturing capabilities can be determined without revealing part geometry.

d. Being able to overcome limitations of knowing part history, part names, project names, and context that are often forgotten.

e. Using distributed repositories at other locations whose histories one may not be familiar with.

f. Reducing the search time for parts and even finding unknown relations or knowledge from earlier projects across the extended enterprise.

g. Enabling the association of context by relating the shape and text-based index structures in multi-dimensional queries.

## 1.1. Previous Work

All related methods for matching 3D shapes decompose a shape into a shape representation. Based on the methods used to convert a shape into a shape representation, they can be classified into the following categories (Iyer, N., et al., submitted):

1. *Invariant/Descriptor based*: These methods use invariants or descriptors of the 3D shape such as volume, surface area, aspect ratio, higher order moments or moment invariants as signatures.

2. *Harmonics based*: These approaches use a set of harmonic functions of a shape as its signature. Spherical or Fourier functions are usually used to decompose a discrete 3D model into an approximate sum of its (first n) harmonic components.

3. *Statistics/Probability based*: Osada, R. et al. (2001) use shape functions and construct a shape distribution by random sampling of points. Ankerst, M. et al. (1999) use shape histograms to approximate and search for a 3D model.

4. *3D Object Recognition based*: Some 3D object recognition approaches that have been used for 3D shape searching are Aspect Graphs (Cyr, C.M., and Kimia, B.B., 2001) and Geometric Hashing (Lamdam, Y., and Wolfson, H.J., 1988).

5. *Graph based*: Graph based approaches have employed subgraph isomorphism for matching B-Rep graphs (El-Mehalawi, M., and Miller, R., 2003) and matching eigenvalues of a model signature graph (MSG) constructed from the B-Rep graph (McWherter, D., et al., 2001).

6. *Feature Recognition based*: Ramesh, M. et al. (2001) decompose a part into cells which are further processed to identify machining features and their spatial relationships to calculate similarity between parts.

7. *Group Technology based*: A two step Group Technology (GT) method was developed in (Iyer, S., and Nagi, R., 1997) to compare similarity between parts.

An extensive review of various approaches to 3D shape searching is available in (Cardone A., et al., (2003) and Iyer, N., et al., submitted). Most previous approaches are bipolar - too granular (categories 4, 5, 6) or too lumped (categories 1, 2, 3, 7). Granular approaches represent the shape in great detail making the search intractable, while lumped approaches combine all shape characteristics into a single quantity. The input to most systems is a detailed model or a detailed drawing, which is unrealistic in an engineering design situation. The designer, depending on the stage of design may not know the detailed shape of the model he/she is searching for. Importantly, an application domain is not considered in most methods. The notion of similarity for engineering shapes is cognitively different than that for other shapes such as in bioinformatics or computer vision.

## 2. DEFINITIONS

The physical shape of any object can be regarded at least from a topological, a morphological, and a geometric point of view. Although, these aspects of a shape are a synergetic entirety, definitions can be formulated that identify related properties.

### 2.1. Informal description

Shape has been studied by philosophers, psychologists, scientists and engineers. However, there has been no unique definition of shape proposed as yet. Merriam-Webster Dictionary defines the noun shape as: "the visible makeup characteristic of a particular item or kind of item", "a spatial form or contour" or "a standard or universally recognized spatial form". Although this describes what is commonly understood by the word 'shape' in the English language, it is not sufficient in the context of analysis and representation of shapes. Marr, D. (1982) described a shape as: "We shall reserve the term shape for the geometry of an object's physical surface. Thus, two statues of a horse, cast from the same mold have the same shape." Another definition of shape was proposed by Kendall, D.G. (1977): "Shape is all the geometrical information that remains when location, scale, and rotational effects are filtered out from an object." Another way to state the above is to say that a shape is all information contained in a geometry that is invariant to Euclidean transformations.

### 2.2. Formal definition

Suppose we have a database of designs D and want to estimate the similarity between two designs $d_1$ and $d_2$ from the database. The similarity between two designs can be calculated based on various design attributes. As an example, let $A=\{a_1,a_2,...,a_n\}$ be the set of attributes that are stored in the database. The set of design attributes for a design can consist of various features including geometry of the design, materials used in the design, processes employed in the design, supplier information, tooling, cost data, assembly instructions and recycling information. We will use $a_i\left(d_j\right)$ to denote the value of the attribute $a_i$ for design $d_j$. The similarity between $d_1$ and $d_2$ based on a particular design attribute $a_i$ can be found by comparing the attributes of both designs $a_i\left(d_1\right)$ and $a_i\left(d_2\right)$. The overall similarity between both designs is simply $\sum_{i=1}^{n}\left|a_i\left(d_1\right)\sim a_i\left(d_2\right)\right|$ where the $\sim$ operator denotes comparison between two attributes.

The geometry or "shape" of the design is an important design attribute. In this paper we describe a method to evaluate similarity between two designs based on their shape. The characterization of a physical shape needs the simultaneous consideration of geometrical, morphological and topological aspects of the shape. Some definitions regarding the above aspects are given in (Horváth, I., 1998).

<u>Definition 1</u>: Geometric shape $S_G$ related to an object $O$ is the set $S_G$ of all points $p_j \in \Re^3$ that build up the physical extent of a shape, that is $S_G = \bigcup_{p_j \in O} p_j$.

<u>Definition 2</u>: Morphological shape $S_M$ related to an object $O$ is the composition of a family $\{z_k\}$ of subsets $z_k$ of points $p_j$ of $E^3$, so that $S_M = \bigcup_{k=1}^{N} z_k$ forms a weak morphological covering of the physical shape and $z_k = \bigcup_{p_j \in z_k} p_j$.

<u>Definition 3</u>: Topological shape $S_T$ related to an object $O$ is a Hausdorff space $S_T$ in which each point $t_i \in S_T$ has a neighborhood homeomorphic to $\Re^3$, so that $S_T = \bigcup_{i=1}^{\infty} t_i$, such that $U(t_i) \geq 0$, where $U(t_i)$ is the number of neighbors of point $t_i$.

According to these definitions, the lowest level of feature information is related to the geometric shape. To depict shape information on a part level, besides the geometric and morphological interpretations, topological shape has to be also considered in order to provide continuity.

# 3. SHAPE REPRESENTATIONS

Any method employed for shape matching and analysis reduces a shape into a simpler *shape representation*. Detailed reviews of various shape representation techniques are available in Loncaric, S. (1998) and Alt, H., and Guibas, L.J. (1996). Below we describe the difference between a shape representation and a shape description.

Marr, D. and Nishihara, H.K. (1978) define a shape representation as: "A formal scheme for describing shape or some aspects of shape together with rules that specify how the scheme is applied to any particular shape. The result of using a *representation* to describe a given shape is a *description* of the shape in that representation. A *description* may specify a shape only roughly or in fine detail."

Loncaric, S. (1998) distinguishes between a shape representation and a description in this way: "Shape representation methods result in a non-numeric representation of the original shape (e.g. a graph) so that the *important* characteristics of the shape are preserved. The word *important* in the above sentence typically has different meanings for different applications. Shape description refers to the methods that result in a numeric descriptor of the shape and is a step subsequent to shape representation. A shape description method generates a shape descriptor vector (also called a feature vector) from a given shape. The goal of description is to uniquely characterize the shape using its shape descriptor vector."

Woodham, R.J. (1987) discriminates between shape representation and description thus: "The term *representation* is used to identify a formalism, or language, for encoding a general class of shapes. The term *description* is restricted to mean a specific expression in the formalism that identifies an instance of a particular shape, or class of shapes, in the representation."

In other words, *shape description* is an instantiation of a *shape representation*. We will use this definition for a shape representation.

## 3.1. Criteria for a Shape Representation

The following criteria have been commonly cited by researchers such as Marr, D. and Nishihara, H.K. (1978), Woodham, R.J. (1987) and Brady, M. (1983) regarding criteria while formulating or evaluating a shape representation:

*(a) Scope*: The shape representation must be able to describe all classes of shapes. For example, a shape representation designed to describe planar surfaces and junctions between perpendicular planes would only have cubical solids within its scope, but would be unable to describe spherical solids or solids with curved surfaces.

*(b) Uniqueness*: There should be a one-to-one mapping between shapes and descriptions of shape within a representation. This is particularly important because at some point during searching, the difficult problem of deciding whether two shape descriptors represent the same shape would arise.

(c) *Stability*: For a particular shape representation, the shape descriptor must be stable to small changes in shape. In other words, small changes in shape must produce small changes in the description.

(d) *Sensitivity*: The shape representation must be capable of capturing even subtle details of the shape. This is a contradictory condition to the stability criterion above. These opposing conditions can only be satisfied if it is possible to de-couple the stable information that captures the more general and less varying properties of a shape from information that is sensitive to the finer distinctions between shapes (Marr, D., 1982).

(e) *Efficiency*: It must be possible to efficiently compute and compare descriptors within a shape representation from input data. In the context of shape searching, it may either be comparison of feature vectors or comparison of other data structures.

(f) *Multi-scale Support*: The representation must describe shape at multiple scales as a hierarchical structure. Details are suppressed until they are required. Further, hierarchical structures are also efficient in terms of storage and are rich in information. The representation should define a natural semantic segmentation and coarse-to-fine levels of detail. As an example, a pinhole in a metal casting is not significant when the task is part identification. But it is significant in identifying defects in parts.

(g) *Local Support*: The representation must be information preserving and, if required, should be able to be computed locally. Local means the scale at which the representation is computed. This may be essential for detailed inspection.

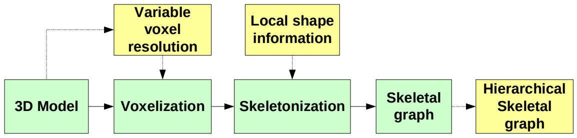


**Figure 1** Conversion of a 3D model into a skeletal graph and a hierarchical skeletal graph representation

Both geometry and topology have to be considered separately to completely describe a shape. Additionally, multi-scale support and local shape information are important for similarity comparisons. The next section describes a hierarchical skeletal graph representation which is capable of simultaneously handling geometry and topology along with multi-scale and local support.

## 4. HIERARCHICAL SKELETAL GRAPH

A Client-Server-Database architecture for 3D shape searching was proposed in our earlier papers. A skeletal graph representation was used as the shape representation for a 3D model (Iyer, N., et al., 2003, Lou, K., et al., 2003). This paper develops the skeletal graph into a hierarchical representation rich with local shape information. The process of converting a 3D model into a skeletal graph is explained in Figure 1.

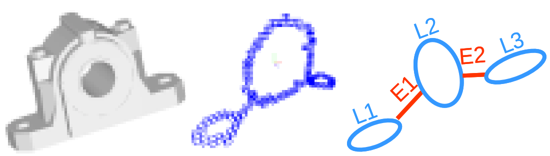


**Figure 2** (a) 3D model, (b) Skeleton, and (c) Skeletal graph for a bearing block. L represents loop entity while E represents line entity in the skeletal graph

A 3D model is converted into a voxel model. Voxelization is defined as the process of converting a geometric representation of a synthetic model into a set of voxels (volume elements) that best represent the synthetic model within the discrete model space. The voxel model is then converted into a skeleton model (Palagyi, K., and Kuba, A., 1998). A marching algorithm is used to identify the various entities in the skeleton model (Iyer, N., et al., 2003). Formally, a *skeleton* is an object $S$ in 3D digital space, composed of *skeletal points* and *skeletal entities.* A *skeletal entity* can be a *line* or a *loop* entity. Below, we define basic terminology used in defining a skeleton.

**Skeletal Points:** A skeletal point $p_i \in \mathbb{Z}^3$ is a point with integer coordinates in 3D digital space. A skeletal point can be a junction point or an end point. A junction point is a point in 3D space where two or more skeletal entities meet. An end point is a free end of a skeletal entity. Let $p = (p_1, p_2, p_3)$ and $q = (q_1, q_2, q_3)$ be two points in 3D digital space and let us consider the Euclidean distance $\|p-q\| = \sqrt{\sum_{i=1}^{3}(p_i - q_i)^2}$. The points $p$ and $q$ are said to be 26-adjacent if $\|p-q\| \le \sqrt{3}$. $N_{26}(p_i)$ is the set of points 26-adjacent to $p_i$ and is called the 26-neighborhood of $p_i$. A set of points $P = \{p_i\}$ is continuous if $\forall p_i \in P\ \exists\, p_j \left| \left(p_j \in N_{26}(p_i),\ p_j \in P\right)\right.$.

**Line Entities:** Line entities can be characterized by their curvature. Every line entity is defined in terms of a parametric equation $\vec{R}(s) = [x(s), y(s), z(s)]$. We define the curvature function $\kappa(s) = \|\vec{R}''(s)\|$ as a function of arc length $s$ of the line entity. If $\kappa(s) = 0$ for all points belonging to an entity, we characterize it as a straight entity. If $\kappa(s) \neq 0$ for all points belonging to an entity, we characterize it as a curved entity.

<u>Straight Entities: </u>A straight entity $\varepsilon^S$ is a collection of continuous skeletal points $p_i$ that form a skeletal entity with zero curvature, that is $\varepsilon^S = \bigcup_{p_i \in S} p_i$

such that $\kappa(p_i) = 0\ \forall\ p_i \in \varepsilon^S$ and

$\forall p_i \in \varepsilon^S\ \exists\, p_j \left| \left(p_j \in N_{26}(p_i) \wedge p_j \in \varepsilon^S\right)\right.$.

<u>Curved Entities</u>: A curved entity $\varepsilon^C$ is a collection of continuous skeletal points $p_i$ that form a skeletal entity with non-zero constant or varying curvature, that is $\varepsilon^C = \bigcup_{p_i \in S} p_i$ such that $\kappa(p_i) = \theta_i\ \forall\ p_i \in \varepsilon^C$ where $\theta_i$ is a non-zero constant, or a function of $p_i$ not zero at all points, and $\forall p_i \in \varepsilon^C\ \exists\ p_j \left| \left(p_j \in N_{26}(p_i) \wedge p_j \in \varepsilon^C\right)\right.$. The set of

all straight entities is $E^S = \bigcup_{\varepsilon_i^S \in S} \varepsilon_i^S$ and the set of all curved entities is $E^C = \bigcup_{\varepsilon_j^C \in S} \varepsilon_j^C$.

**Loop Entities**: A loop entity is a closed continuous skeletal entity which connects a skeletal point to itself. A loop entity $L$ can contain a set of line entities, that is $L = \bigcup_{E_i \in E} E_i$ where $E = E^S \cup E^C$.

**Skeletal Graph:** A skeletal graph $G$ is an undirected graph, represented as a tuple $G = (V, E, v, e)$ where $V$ is a finite set of vertices, $V \neq \phi$, $E \subseteq V \times V$ is a finite set of edges, $v: V \rightarrow W_V$ a mapping for assigning attributes to the vertices ($W_V$ is a finite set of attributes for vertices), $e: E \rightarrow W_E$ a mapping for assigning attributes to the edges ($W_E$ is a finite set of attributes for the edges). Vertices $V$ represent skeletal (line and loop) entities, while edges $E$ represent connectivity between skeletal entities. The set of attributes $W_V$ represent the local properties of skeletal entities such as length, curvature, moments, distances and volumes. The set of attributes $W_E$ represent relational information between connected entities such as angle, relative length ratios and relative volume ratios.

Figure 2 shows an example of a 3D model together with its skeleton model and skeletal graph. Similarity between two 3D models is determined by matching their skeletal graphs thereby yielding a similarity score as well as the correspondences between the various entities.

### 4.1. Levels of detail

In our technique a 3D model is represented at different levels of resolution by a hierarchical set of skeletal graphs. The various hierarchical skeletal graphs are generated by varying the voxel resolution for the 3D model as described in Figure 1. We determine the number of voxels for a 3D model based on the smallest dimension of its bounding box. The voxel size is calculated as $d/2^n$ where $d$ is the smallest bounding box dimension and $n$ is the level of resolution desired. Figure 3 shows some skeletons generated by varying the value of $n$.

Hierarchical structures are well supported by studies in human cognition. Koenderink, J. J., and van Doorn, A. J., (1986) suggest that that the perceptual approach to shape organization is dynamic. A partial order is apparent that relies on a hypothetical evolution or morphogenesis that is an integral part of the shape description. In the conventional approach, shapes that are visually not all that different end up in different ball parks. In general, comprehension of an objects shape by humans follows the principle of “from remote to closer”, “from outer to inner”, “from total to detail” (Wang, C., et al., 2002).

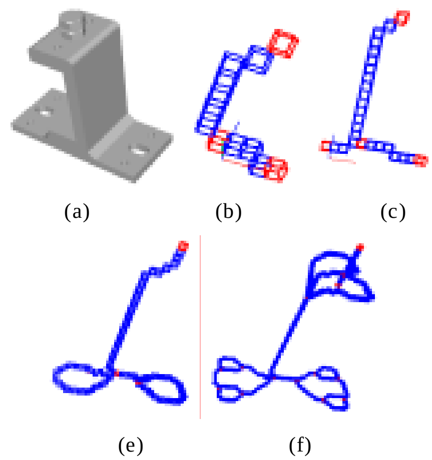


**Figure 3** (a) 3D model, (b) Levels of detail (LOD) for n = 2, (c) LOD for n = 3, (d) LOD for n = 4, (d) LOD for n = 5

The hierarchical skeletal graph structure is an example of a dynamic approach to shape description. The advantage of using a dynamic approach is that similar shapes can be detected at different resolutions. For example, consider the shapes in Figure 4. Although these shapes can be perceived as visually similar, conventional approaches cannot detect them as similar. The skeletons for these shapes at the same voxel resolution will also be different. However, they will yield similar skeletons at individually different voxel resolutions. Thus, we will be able to detect them as similar at some level of detail.

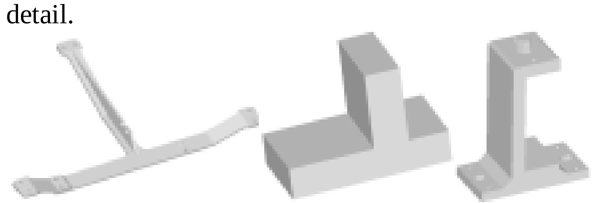

**Figure 4** Models that may be perceived as visually similar but cannot be detected as similar by conventional approaches

## 4.2. Local Shape Information

Our hierarchical graph structure stores local shape information such as normalized entity lengths and curvature distribution. Topological as well as geometric relationships among entities such as internal angles as well as degree and length ratios are used to enrich the graph structure. These attributes are used in the graph matching process in a weighted fashion.

In our future implementations we plan to use other local information including local distances obtained through the use of distance transform for each skeletal voxel, normalized volumes, volume distribution along each entity and principal moments for each entity. We believe that the use of such local properties will restore most of the shape-related information lost during skeletonization, without affecting the complexity of graph matching considerably.

In summary, by using a hierarchical skeletal graph structure with local information we have tried to explicitly address the issues of multi-scale support and local support as detailed in the criteria for shape representation (see Section 3).

# 5. SKELETAL GRAPH MATCHING

The process of converting 3D models into skeletal graphs converts the problem of matching shapes into one of matching their skeletal graphs. Graph matching has been well-studied for various theoretical as well as real-world applications including computer vision and pattern recognition (Christmas, W.J., et. al., (1999), and Myers, R., and Hancock, E. R., (1999)). As a result, a large number of algorithms have been proposed many of which compare a query graph with each of the database graphs providing a measure of similarity between them (Jolion, J. M., (2001)). Messmer, B. T., and Bunke, H. J., (1985) proposed a decision-tree based approach for indexing all graphs in a database, thereby making similarity a decision, whereby traversal of a query graph through the tree resulted in a set of similar models. This approach was employed in Lou, K. et al. (2003) for indexing and searching from a set of skeletal graphs of engineering shapes. Although the search time is linearly proportional to the size of the query graph, this approach becomes intractable when indexing skeletal graphs with sizes larger than 10 nodes.

In this paper, we have made use of the association graph technique that converts the graph matching problem into one of finding the maximum weighted clique in a composite graph called an association graph. Details of this approach are provided in the subsections below (also see Ballard, D., and Brown. C., (1982) and, Horaud, R., and Skordas, T., (1989)). Subsequent to forming an association graph, a heuristics-based Genetic Algorithm (GA) developed by Marchiori, E., (1998) is employed to determine the maximum weighted clique of the association graph. The maximum weight of the clique therefore represents the 'best similarity' between the two graphs.

## 5.1. Topology and local shape matching

We devote this section to describe the association graph based shape matching algorithm in detail. An *association graph* relates two graphs in order to find homeomorphism between both graphs. It is composed of nodes, and arcs connecting nodes. Although the terms 'vertex' and 'node' as well as 'edge' and 'arc' are interchangeably used in graph theory, for the sake of clarity, we describe a *skeletal graph* as being made up of *vertices* and *edges*, while an *association graph* as composed of *nodes* and *arcs*.

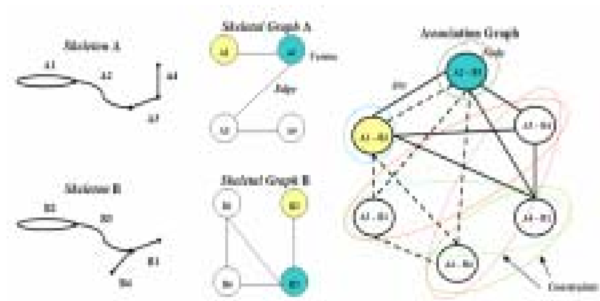

**Figure 5** Association graph construction for determining similarity between two skeletal graphs

The association graph $\Gamma$ between two graphs $G_1 = (V_1, E_1, v_1, e_1)$ and $G_2 = (V_2, E_2, v_2, e_2)$ is defined as the complete graph $\Gamma = (N_A, A_A, \eta_A, \alpha_A)$ with nodes $N_A = V_1 \times V_2$, arcs $A_A = N_A \times N_A$, node attributes $\eta_A$ and arc attributes $\alpha_A$. Node attributes represent a measure of similarity between the skeletal vertices it represents. For example, a node $n_A^i$ representing the association between the vertices $V_1^l$ and $V_2^m$ of two skeletal graphs $G_1$ and $G_2$, respectively, will represent the similarity between their attributes $v_1^l$ and $v_2^m$. The similarity measure is formed as a weighted sum of the individual attribute similarities for the corresponding skeletal vertices. An arc $a_A^{ij}$ between two nodes $n_A^i$ and $n_A^j$ has positive attributes if the two nodes are compatible (subject to constraints), and negative otherwise.

Similarity between the two graphs $G_1$ and $G_2$ now is a question of finding the homeomorphism of the association graph $\Gamma$. A graph homeomorphism h is a mapping $h: V_1 \to V_2 \mid \forall v_1 \in V_1,\ \forall\ v_2 \in V_2$ , such that if $(v_1, v2) \in E_1$, then $(h(v_1), h(v_2)) \in E_2$. In terms of the association graph, a homeomorphism is a subgraph $\Gamma' = (N_A', A_A', n_A', a_A')$ of the association graph, which forms the maximal-weighted clique if the attributes are considered as weights. There exist many such homeomorphism mappings between any two *skeletal graphs*. However, the best similarity between the two graphs is the homeomorphism yielding the highest value of similarity among skeletal vertices and edges. Hence, the problem is one of maximizing the combined weight of a clique, also called a '*similarity function'*, S(Γ'), as described below:

$$S(\Gamma') = \frac{\omega_v}{|N'|} \sum_{(n_1,n_2) \in N'} nodeSim(n_1, n_2) + \frac{(1-\omega_v)}{|E'|} \sum_{a \in A'} arcSim(a)$$

where $a$ is the arc representing the relationships between edge $(n_1, m_1)$ of $G_1$ and edge $(n_2, m_2)$ of $G_2$. The term *arcSim(a)* represents the compatibility value of the arc $a$, $a = ((n_1, n_2), (m_1, m_2))$, while *nodeSim*$(n_1,n_2)$ represents the similarity between nodes $n_1$ and $n_2$.

Compatibility constraints ensure the validity of graph matching with constraints such as one-to-one correspondence between skeletal graph vertices, matching nodes only if the corresponding edges of the skeletal graph are sufficiently similar, etc.

## 5.2. Similarity determination

The problem of finding the 'best similarity' between two models is essentially a combinatorial optimization problem where the objective function is the similarity measure. We have utilized a heuristic-based Genetic Algorithm approach (Marchiori, E., 1998) to explore the maximal weighted clique of the association graph. The algorithm starts by initially picking a number of random subgraphs of the association graph. A heuristic algorithm is then used to randomly prune and relax the subgraph in order to obtain a maximal weighted clique. Subsequently, this population of cliques serves as the input to a GA that searches for better solutions with the aid of a set of crossover and mutation operations. Although globally optimal solutions are not guaranteed for large graphs, this method avoids exhaustive search which becomes intractable for medium to large graph sizes. However, this algorithm is capable of finding near-optimal solutions for fairly large graphs. The reasonably small sizes of our skeletal graphs guarantee accurate similarity determination with high probability.

## 6. RESULTS AND DISCUSSION

Figures 6 and 7 present some search results for two queries on a database of 200 parts. For both sets of results, equal weights were given to geometry and topology during graph matching. Addition of geometric information to the skeletal graph improves the similarity between models and retrieves more similar models while discarding dissimilar models.

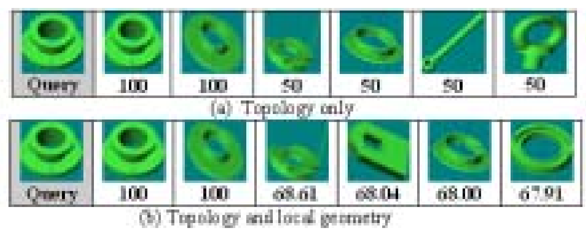


**Figure 6** Effect of local geometry for Query model 1

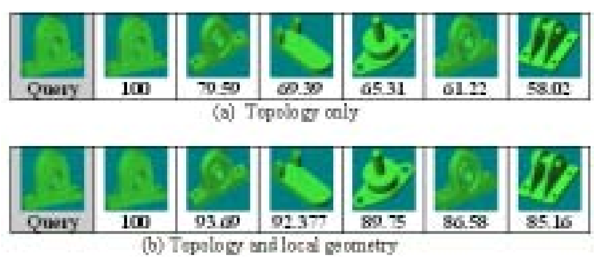


**Figure 7** Effect of local geometry for Query model 2

Inclusion of local geometry in the skeletal graph further refines the similarity values to account for noise. Therefore, for some queries (e.g. Figure 6) more similar models are retrieved even in the presence of noise. However, for other queries this refinement does not alter the ordering of search results significantly (e.g. Figure 7), although the similarity values are more in correspondence with reality. However, a systematic study of the effectiveness of the local geometry is needed to assert the effectiveness of such refinement. We believe that including additional geometric properties such as local volumes will further refine the similarity metric and make it more consistent with human perception of similarity.

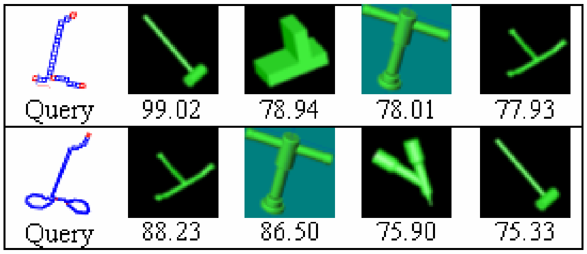


**Figure 8** Search results for two queries using two levels of detail for a model using both geometry and topology of the graph

We also tested the effectiveness of using graphs at various levels of detail for shape matching. Separate graphs at two different levels of detail for the model shown in Figure 3 were stored in the database and were tested for similarity against other models. Figure 8 shows some representative search results. As observed from the figures, different models are retrieved for the same query model with different LODs, thereby providing flexible search intent and multi-scale support to the user. However, skeletal graph matching based on LOD needs a lot of refinement. Specifically, the similarity metric needs to be modified to consider what level of detail matches best with the query model.

Our current implementation of graph matching controls the number of iterations during optimization based on the corresponding size of the association graph. In essence, the similarity metric is not an absolute, but an approximate measure. The search time for a query is dependent not only on the size of the database but also the sizes of the skeletal graphs in the database. For example, the times taken for the queries in Figure 6 and 7 were 2 seconds and 92 seconds respectively.

## 7. CONCLUSIONS

The hierarchical shape representation proposed in this paper was based on human cognition. Parts that are visually similar but could not be retrieved using skeletal graphs at a single level of detail were retrieved as similar with this new representation. Additionally, local shape information is used to finely discriminate between shapes. It helped reduce the semantic gap, which is the difference in perception of similarity between the system and a user.

As part of our future work, we propose to use this method for a larger database of parts. Careful studies of time taken for skeletal graph matching will also be conducted along with database pruning to reduce the number of graph comparisons. We expect that the use of additional local information will greatly enrich the hierarchical graph, thereby reducing the semantic gap further.

## ACKNOWLEDGMENTS

The initial funding for this project came from the 21st Century Research and Technology Fund award from the state of Indiana. We acknowledge Professor Karthik Ramani's University Faculty Scholar award from Purdue University, which seeded this project. Supplemental support from the e-Enterprise Center at Discovery Park, Purdue University is acknowledged. We also recognize the Innovation Realization Laboratory at Purdue University for supporting Natraj Iyer.

## REFERENCES

Alt, H., and Guibas, L.J., (1996), “Discrete Geometric Shapes: Matching, Interpolation, and Approximation: A Survey,” Technical Report B 96-11, EVL-1996-142, Institute of Computer Science, Freie Universität, Berlin.

Ankerst, M., Kastenmüller, G., Kriegel, H.-P., Seidl, T., (1999), “3D Shape Histograms for Similarity Search and Classification in Spatial Databases,” Lecture Notes in Computer Science, Vol. 1651, Springer, pp. 207-226.

Brady, M., (1983), “Criteria for representations of shape”, In Human and Machine Vision, eds. Beck, J., Hope, B., and Rosenfeld, A., Academic Press, pp. 39-84.

Ballard, D., and Brown, C., (1982), Computer Vision. Prentice-Hall.

Cardone, A., Gupta, S.K., and Karnik, M., (2003), “A Survey of Shape Similarity Assessment Algorithms for Product Design and Manufacturing Applications”, ASME Journal of Computing and Information Science in Engineering, Vol. 3, No. 2, pp. 109-118.

Christmas, W. J., Kittler, J., and Petrou, M., (1995), “Structural Matching in Computer Vision Using Probabilistic Relaxation,” IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 17, No. 8, pp. 749-764.

Cyr, C. M., and Kimia, B. B., (2001), “3D object recognition using shape similarity-based aspect graph”, in Proceedings of the International Conference on Computer Vision, pp. 254-261.

El-Mehalawi, M., and Miller, R., (2003), “A Database System of Mechanical Components Based on Geometric and Topological Similarity. Part I: Representation and Part II: Indexing, Retrieval, Matching, and Similarity Assessment,” Computer Aided Design, Vol. 35, pp. 83-105.

Fruchter, R., and Demian, P., (2002), “Knowledge Management for Reuse”, Proceedings of CIB w78 Conference, Aarhus School of Architecture, Denmark.

Horaud R. and Skordas T., (1989), “Stereo Correspondence through Feature Grouping and Maximal Cliques,” IEEE Journal on Pattern Analysis and Computer Vision, Vol. 11, No.11, pp. 1168-1180.

Horváth, I., (1998). “Theoretical Fundamentals of Natural Representation of Shapes Generated with Gestural Devices”, Proceedings of TMCE’ 98, Manchester, England, pp. 393-409.

Iyer, N., Kalyanaraman, Y., Lou, K., Jayanti, S., and Ramani, K., (2003), “A Reconfigurable 3D Engineering Shape Search System Part I: Shape Representation”, accepted for ASME DETC 03 Computers and Information in Engineering (CIE) Conference, Chicago, Illinois.

Iyer, N., Lou, K., Jayanti, S., Kalyanaraman, Y., and Ramani, K., “Three-dimensional Shape Searching: State-of-the-art review and Future Trends”, submitted to Computer Aided Design.

Iyer, S. and Nagi, R., (1997), “Automated Retrieval and Ranking of Similar Parts in Agile Manufacturing”, IIE Transactions on Design and Manufacturing, Vol. 29, pp. 859-876.

Jolion, J. M., (2001), “Graph matching: what are we talking about? ,” in J.M. Jolion, W. Kropatsch, and M. Vento eds, Proc. of Third IAPR-TC15 Workshop on Graphbased Representations in Pattern Recognition, May 23-25, 2001.

Kendall, D.G., (1977), “The diffusion of shape," Advances in Applied Probability, Vol. 9, pp. 428–430.

Koenderink, J.J., and van Doorn, A.J., (1986), “Dynamic Shape”, Biological Cybernetics, Vol. 53, pp. 383-396.

Lamdam, Y. and Wolfson, H.J., (1988), “Geometric Hashing: a General and Efficient Model Based Recognition Scheme,” Proceedings of International Conference on Computer Vision, Tampa, Florida, pp. 238-249.

Leizerowicz, W., Lin, J., and Fox, M.S., (1996), “Collaborative Design Using WWW,” Proceedings of WET-ICE '96, CERC, University of West Virginia.

Loncaric S., (1998), “A Survey of Shape Analysis Techniques”, Pattern Recognition, Vol. 31, No. 8, pp. 983-1001.

Lou, K., Jayanti, S., Iyer, N., Kalyanaraman, Y., Ramani, K., and Prabhakar, S., (2003), “A Reconfigurable 3D Engineering Shape Search System Part II: Database Indexing, Retrieval and Clustering”, accepted for ASME DETC 03 Computers and Information in Engineering (CIE) Conference, Chicago, Illinois.

Marchiori, E., (1998), “A simple heuristic based genetic algorithm for the maximum clique problem,” Proceedings of ACM Symposium on Applied Computing, pp. 366–373.

Marr, D., (1982), “Vision: A Computational Investigation into the Human Representation and Processing of Visual Information,” W.H. Freeman, San Francisco.

Marr, D., and Nishihara, H.K., (1978), “Representation and Recognition of the Spatial Organization of Three-dimensional Shapes,” Proceedings of the Royal Society of London: Part B, Vol. 200, No. 1140, pp. 269-294.

McWherter, D., Peabody, M., Regli, W. C., and Shoukofandeh, A., (2001), “An approach to Indexing Databases of Graphs,” Technical Report DU-MCS-01-01, Department of Mathematical and Computer Science, Drexel University, Philadelphia, PA.

Messmer, B.T., and Bunke, H., (1995), “Subgraph Isomorphism in Polynomial Time,” Technical Report IAM 95-003, University of Bern, Institute of Computer Science and Applied Mathematics, Bern, Switzerland.

Myers, R., and Hancock, E. R., (2000), “Selection Strategies for Ambiguous Graph Matching by Evolutionary Optimization,” Proceedings of SSPR/SPR 2000, pp. 397-406.

Osada, R., Funkhouser, T., Chazelle, B., and Dobkin, D., (2001), “Matching 3D models with shape distributions”, in Proceedings of Shape Modeling International, Genova, Italy, pp. 154-166.

Palagyi, K., and Kuba, A., (1998), “A 3D 6-subiteration Thinning Algorithm for Extracting Medial Lines,” Pattern Recognition Letters, Vol. 19, pp. 613-627.

Ramesh, M., Yip-Hoi, D., Dutta, D., (2001), “Feature based Shape Similarity Measurement for Mechanical Parts”, ASME Journal of Computing and Information Science, Vol. 1, No. 3, pp. 245-256.

Repenning, N.P., “Understanding Fire Fighting in New Product Development,” Journal of Product Innovation Management, Vol. 18, pp.285-300.

Wang, C., Horváth, I., Vergeest, J.S.M., (2002), “Towards the Reuse of Shape Information in CAD”, Proceedings of TMCE’ 00, Wuhan, P.R. China, pp.103-116.

Woodham, R.J., (1987), “Stable representation of shape.” In Computational Processes in Human Vision, ed. by Z. Pylyshyn, Norwood, N.J.