Using Shape Distributions to Compare Solid Models*

Cheuk Yiu Ip Daniel Lapadat Leonard Sieger William C. Regli †
Geometric and Intelligent Computing Laboratory
Department of Mathematics and Computer Science
Drexel University
3141 Chestnut Street
Philadelphia, PA 19104
http://gicl.mcs.drexel.edu/

ABSTRACT

Our recent work has described how to use feature and topology information to compare 3-D solid models. In this work we describe a new method to compare solid models based on shape distributions. Shape distribution functions are common in the computer graphics and computer vision communities. The typical use of shape distributions is to compare 2-D objects, such as those obtained from imaging devices (cameras and other computer vision equipment). Recent work has applied shape distribution metrics for comparison of approximate models found in the graphics community, such as polygonal meshes, faceted representation, and Virtual Reality Modeling Language (VRML) models. This paper examines how to adapt these techniques to comparison of 3-D solid models, such as those produced by commercial CAD systems. We provide a brief review of shape matching with distribution functions and present an approach to matching solid models. First, we show how to extend basic distribution-based techniques to handle CAD data that has been exported to VRML format. These extensions address specific geometries that occur in mechanical CAD data. Second, we describe how to use shape distributions to directly interrogate solid models. Lastly, we show how these techniques can be put together to provide a "query by example" interface to a large, heterogeneous, CAD database: The National Design Repository. One significant contribution of our work is the systematic technique for performing consistent, engineering content-based comparisons of CAD models produced by different CAD systems.

Categories and Subject Descriptors

H.3.3 [**Information Storage or Retrieval**]: Information Search or Retrieval; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism; J.6 [**Computer Applications**]: Computer-Aided Engineering

General Terms

Management, Experimentation

Keywords

3D Search, Shape Recognition, Shape Matching, Solid Model Databases

*Authors listed alphabetically.

†Also with the Department of Mechanical Engineering and Mechanics. URL: http://www.mcs.drexel.edu/∼regli; E-mail: regli@drexel.edu.

SM'02 June 17-21, 2002, Saarbrucken, Germany.

1. INTRODUCTION

We introduce ideas from computer graphics and computer vision to the problem of content-based retrieval of CAD data. In particular, this work introduces a new approach to manage solid models in database systems that are part of the modern engineering enterprise.

A problem in Computer-Aided Design has been the diversity and heterogeneity of representation formats for the shape information. At a fundamental level, Constructive Solid Geometry (CSG) and Boundary Representation models (BRep) serve as a foundation for most modeling systems and applications. While BReps dominate the CAD industry, the mathematical details of the representation vary widely by system. Hence, even when data translation (via, for example, STEP AP 203) works well, there is little to guarantee that the resulting solid models can be directly compared.

The BReps found "in the wild" come in several species, however two appear to dominate the commercial CAD environment: *NURBS-based BReps* (e.g., SDRC, Pro/E, where NURBS are the primary internal representation) and those dominated by *Analytic Surface BReps* (e.g., Parasolid, ACIS, where analytic surfaces coexist with NURBS). Comparing CAD models for indexing across these formats can be very difficult, requiring considerable amounts of special-case algorithms for each different representation. Even if we work with a STEP AP 203 version of a simple shape, such as a unit cube, the internal representations coming from different systems can be radically different and very hard to compare.

A long-term goal of our work is to develop uniform methodologies to interact with CAD data in engineering information management systems. In our previous work, we have considered three problems in model matching and retrieval: indexing based on design features, indexing based on manufacturing features, and indexing based on model topology. Homogeneity of representation was an underlying assumption for each of these techniques (i.e., they employed extensive geometric reasoning off of an ACIS-based CAD model). In this work we introduce a new technique, based on matching of *shape distributions*, that allows us to compare CAD models regardless of their underlying modeling representation.

Shape models, in particular as found in the computer graphics and computer vision communities, typically are 3D models used for rendering, visualization, simulation and animation. While shape models from these fields sometimes involve exact representations

(implicit surfaces, superquadrics or deformable surfaces), our work leverages recent results in matching approximate shape models, i.e., fine-grained faceted or polyhedral models such as produced by VRML exporters, mesh generators or 3D scanning devices. *Shape distributions* are 2D characterizations of 3D shape based on random statistical sampling of an object's topology. Recent research in computer graphics and computer vision [10] has successfully applied this approach to comparison of shape models found in the graphics domain (chairs, animals, animation figures, etc). This work shows how these ideas can be used with much less diverse sets of 3D models, such as CAD models of mechanical designs.

Given a solid model *S*, our methodology is:

- Obtain a *shape model*, a polyhedral approximation T of the solid model *S*, through faceting, for example, as found in Virtual Reality Modeling Language (VRML) and Stereolithography (`.stl`) translation.

  Generating meshes, triangularizations and voxelizations for CAD and solid models is a well-understood and deeply studied problem. Meshes are used for a number of downstream CAE applications, such as finite element analysis, where forces need to be propagated through solid artifacts. Triangularizations are the basis of `.stl` format, the de-facto standard for solid freeform fabrication processes, and VRML, the 3D viewing/ visualization standard that dominates the Web.

- Compare models using an *enhanced shape distribution* approach. A shape distribution is calculated as follows:

  - Select a shape function. In our work we adopt the $D2$ shape function which measures the distance between two random points on the surface of a model. This shape function has been previously shown to produce metric distance measures that are robust (under varying qualities of faceting) and invariant (under shape-preserving transformations) [10] .
  - Generate set of random sample points evenly distributed on the surface of T;
  - Calculate shape distribution histograms associated with the $D2$ shape distribution function.
  - Compare shape distribution histograms using well-known curve matching techniques.

- Perform empirical validation to determine if shape distribution-based techniques can be used to answer engineering questions. In this paper, we consider an elementary set of tasks (1) automatic inference of part categories and clusters (2) query-by-example browsing of indexed part repositories.

This paper's main contributions include a methodology for comparing solid models based shape distributions that works regardless of the underlying BRep modeling representation. As part of this research, we have shown how to unite several significant research trends in computer vision, computer graphics and databases to create a robust technique for comparison of solid models. Additional contributions of this work include introduction of novel refinements to general shape distribution techniques that enhance their discrimination abilities and enable us to answer meaningful CAD and engineering questions.

2. RELATED RESEARCH

Our research aims to bring information retrieval to CAD databases, enabling them to have indexing and query mechanisms like those beginning to be found in multimedia databases and knowledge management systems. We touch on some of the past work in this area, as well as on the work from computer graphics and computer vision that are related to efforts of this paper.

2.1 Comparing Solid Models

The literature in this area is rather brief, consisting of results from engineering, computer science and, in particular, computer vision communities. Elinson et al. [4] used feature-based reasoning for retrieval of solid models for use in variant process planning. Cicirello and Regli [12, 2] examined how to develop graph-based data structures and create heuristic similarity measures among artifacts; this work was extended in [1] to manufacturing feature-based similarity measurement. Most recently, McWherter et al. [7, 9, 8] have integrated these ideas with database techniques to enable indexing and clustering of CAD models based on shape and engineering properties. Other recent work from the Engineering community includes techniques for automatic detection of part families [11] and topological similarity assessment of polyhedral models [15].

2.2 Comparing Shape Models

The computer vision and computer graphics research communities have typically viewed shape matching as a problem in 2D. This has changed in the past several years with the ready availability of 3D models (usually meshes or point clouds) generated from range and sensor data. A considerable body of work has emerged to interrogate acquired datasets: Thompson et al. [17] examined reverse engineering of designs by generating surface and machining feature information off of range data collected from machined parts. Jain et al. [5] performed some work to index CAD data based on the creation of "feature vectors" from 2D images. The 3D-Base Project [3] converted CAD models into a voxel representation, which is then used to perform comparisons using geometric moments and other features. Sipe, Casasent and Talukder [14, 16, 13] used acquired 2D image data to correlated real machined parts to CAD models and perform classification and pose estimation. Hilaga et al. [6] present a method for matching 3D topological models using multi-resolution Reeb graphs.

The approach most directly related to our research is that of Osada et al. [10]. This method creates an abstraction of the 3D model as a probability distribution of samples from a shape function acting on the model. Specifically, the measure of the similarity between two models is determined by measuring the similarity between their shape distributions. Shape distributions are generated by random sampling of points on the surface of the model. In their paper, they empirically study five different shape functions and conclude (experimentally) that a function they call *D2* (which measures the distance between two random points on the surface of a model) results in the best shape classification method. Their database is a set of over 130 VRML shape models from the Internet.

In general, shape matching-based approaches only operate on the gross-shapes of a single parts and do not work directly on solid models or with semantically meaningful engineering information (i.e., manufacturing or design features, tolerances). Retrieval strategies are usually query-by-example or query-by-sketch paradigms. The Princeton 3D shape database that has been used in a number of these studies [6, 10] contains mainly models from 3D graphics and rendering, and not any models that are specifically engineering, solid modeling or mechanical CAD oriented.

3. PROBLEM FORMULATION

A *shape distribution* can be viewed as a digital signature for a 3D model. While these functions have been studied extensively in

computer vision (where data is approximate) our work represents the first examination of shape distribution in the context of CAD and solid modeling. Our approach is to use distribution-based techniques to perform statistical sampling of the shape properties of solid models and use these samples to generate meaningful comparison metrics among the models. The benefits include:

- **Heterogeneity**: solid models are reduced to a lowest-common-denominator among CAD and modeling formats, thus allowing us to compare models with fundamentally different representations of underlying geometry and topology.
- **Objectivity:** avoid ambiguities regarding which features to recognize and use for indexing by using model comparisons based on fundamental shape properties of the artifact which subsume low-level shape feature information.
- **Perspicacity:** we introduce improvements to shape distributions to enable higher discrimination factors than previous matching techniques. We show how these enhancements can be used to infer model categories and classifications, as well as produce fewer false positives during queries.

Let S be a solid model, let $T = \{t_1, t_2, \ldots t_k\}$ be a set of triangular facets that approximate the topology of S. Note that a triangularization T is not unique for S, nor is T guaranteed to be accurate. For the purposes of this paper, we assume that each of the facets of T are accurate to within ε of the model S (i.e., maximum distance from any point in t_i to the nearest point on the model is $\leq \varepsilon$). The facets of T could be produced by existing mesh generation algorithms, Stereolithography exporters or, as is the case with most of our experimental data, VRML exporters. The facets in T could also be from an active data acquisition system working off of actual models, as in [17].

3.1 Review of the General Approach to Matching with Shape Distributions

Given S and T, matching with shape distributions requires:

1. **Select a shape function.** Following [10], we adopt the $D2$ shape function. $D2$ measures the distance between two random points on the surface of a model.
2. **Sampling of random points:** Generate a sufficiently large number of random sample points on surface of model S.
3. **Calculate shape distribution histograms** associated with the $D2$ shape distribution function.
4. **Compare shape distribution histograms** using well-known curve matching techniques (Minkowski L_N, earth mover's distance, etc.).

An example of two models and their $D2$ shape distribution histograms is given in Figure 1.

Known Issues when Matching with Shape Distributions. These techniques have some limitations when applied directly to matching of solid models such as those found in electromechanical engineering design. Specifically, the technique in Osada et al. [10] is primarily intended for matching gross, or overall, model shapes. In our experiments, as well as their own, pure shape distributions do a very good job of distinguishing models in broad categories: aircraft, boats, people, animals, etc. In this situation, where the models could literally be anything, shape distribution distances generally confirm what one would think of as the intuitive similarity between shapes. However, it can often do poorly when having

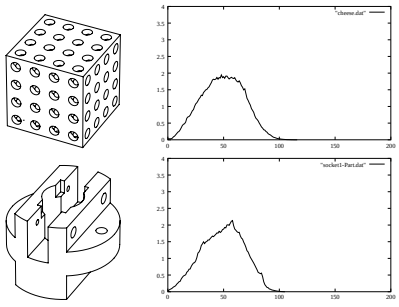


Figure 1: A Swiss-cheese-like part (named "cheese") compared to the classic Gupta Socket example. Note that the $D2$ shape histograms for the two models are very similar—this could lead to query engines generating false positives.

to discriminate between shapes that have similar *gross shape* properties but vastly different *detailed shape* properties. As models get more complex the shape histograms tend toward a bell-shaped, normal distribution. This can lead to models being classified as similar when their topological properties are vastly different. As a result, the technique often yields false positives and, sometimes, false negatives. This is evident in the graphs for parts shown in Figure 1.

Further, it is difficult to implement a general purpose random point generator that operates directly on the surfaces of the solid model S. While one could implement a random point generator that works directly on solid models, it would depend on the underlying surface representations used. Further, this function will need to contain many special cases and, further, these cases will vary from CAD/modeling system to CAD/modeling system.

3.2 Enhanced Shape Distributions for Discrimination of Solid Models

In our approach we compare solid models using a generated set of facets, T, and introduce refinements that improve their performance on CAD data (in particular, on mechanical CAD data).

3.2.1 Construction of Random Points

The generation of m random sample points on the surface of the mesh $T = \{t_1, t_2 \ldots t_k\}$ of the solid model S consists of these steps:

1. Load T into memory, repairing the mesh as needed to guarantee that all polygons in the mesh are triangles;
2. Compute $A(t_i)$, the area of triangle t_i for all $t_i \in T$;
3. Generate `CA[]`, the array of cumulative triangle areas;
4. Generate m sample points on the surface of the model
   (a) Generate a random number r between 0 and the total surface area of T;
   (b) Use r to identify which t_i to place the random point on;
   (c) Generate random point in triangle t

The following sections review the specifics of each of the components of the algorithm.

Read and Fix the Mesh. First, we read the mesh T into memory. The quality and consistency of meshes found "in nature" vary widely. While it is theoretically pleasant to assume that all meshes

are precisely defined with triangles and stored compactly, this simply is not the case for many models produced by VRML and STL exporters. In particular, VRML meshes may contain quadrilaterals, redundant points, and sometimes even redundant topology. When we parse mesh data, redundant points are removed and quadrilaterals are subdivided into triangles. If quadrilaterals are found, they are broken up into two triangles by taking a vertex of the quadrilateral and making three vectors. These vectors are the vectors from the selected vertex to the other three vertices. Then the angles between all three possible pairs of vertices are computed. The largest angle will be between v_1 and v_2, so we know the remaining vector, v_3, is the one we use to split with.

Computing Areas of the Triangles in the Mesh. The areas of each of the triangles, t_i are calculated by computing the lengths of the sides and using Heron's formula (where s is the semi-perimeter of the triangle t_i):

$$A(t_i) = \sqrt{s(s - \|\overline{AB}\|)(s - \|\overline{BC}\|)(s - \|\overline{CA}\|)}$$

As these areas are being computed, they are placed in an array of cumulative areas to be used for placing the random points.

How Many Random Points to Generate. Selecting the number of random points to needed to create a meaningful sample is a difficult matter. The more points used the higher fidelity of the statistical sampling. However, to compute the histogram we will be computing the distances between all pairs of sample points—a task that is quadratic-time complexity. There are trade-offs to consider: given a triangularization $T = \{t_1, t_2 \ldots t_k\}$ of a solid S, one could:

1. Let f be positive integer-valued function over the integers and sample until there are a number $f(k) > 0$ of samples for each t_i. Shape Histogram Complexity: $O((f(k) * k)^2)$.
2. Sample until there are a fixed number $m > 0$ of samples for each t_i. Shape Histogram Complexity: $O((k * m)^2)$.
3. Sample until there are a fixed number $m > 0$ of sample points. Shape Histogram Complexity: $O(m^2)$.

The first thing to observe is that the cost of computing the $D2$ histogram at high levels of fidelity is quite expensive: even VRML meshes for solid models of the most basic artifacts can contain thousands of facets. While there may be an opportunity to employ model and mesh simplification strategies to reduce this complexity, the resulting loss in model quality may eliminate any computational benefits when we use the results for comparisons. Further, facet area can vary greatly and the probability of a facet getting one of the random sample points placed on it is proportional to its area. Hence, to guarantee that every facet gets some number of points > 0 placed on it surface may require generating many more points than there are actual facets.

Given these computational limitations, we use the third approach, selecting to create a fixed number ($m = 1024$) of sample points. Once we locate the m random sample points within the triangles, we compute the $O(m^2)$ pairwise Euclidean distances needed to produce the $D2$ histogram and sort them into the shape distribution histograms. Specifically, this yields $\frac{1024!}{2!*(1024-2)!} = 523,776$ pairs of points to be analyzed when computing the $D2$ histogram.

One additional observation: while intuitively one might like to increase the sample density in "areas of interest" on the model, this would skew the results captured in the shape distribution histogram. Thus, the sampling distribution needs to be completely random.

Generate Random Sample Points. Exact computation off of the solid model's mathematical representation of topology is not needed. Meaningful model comparisons can be made working off of approximate models, such as VRML meshes. For every solid S, we can generate an approximation of topology in the form of a triangle representation T. Using T, we calculate the areas of all the triangles and place the cumulative areas into an array (`CA[]`). We generate a random number between 0 and the total cumulative area, then use binary search to find the triangle t_i to receive the random point. Using two random numbers, r_1 and r_2 [10], a random point on the facet t_i is calculated as:

$$p = (1 - \sqrt{r_1})A + \sqrt{r_1}(1 - r_2)B + \sqrt{r_1}r_2C$$

3.2.2 Calculation of Pairwise Distances

Given the set $\mathscr{P}$ of random points in 3D, the distances between all pairs of points are calculated to create the $D2$ histogram ($O(n^2)$ Euclidean distance calculations).

3.2.3 Classification of Pairwise Distances

The histogram maps *distance probability* versus *measured distance* by counting how many distances fall into fixed size bins. This is the basic approach taken in previous work, and it has the limitations mentioned earlier. We present two significant improvements to the basic sampling procedure:

1. **Classification of Pairwise Point Distances.** The traditional shape distribution approach considers all point pairs as contributing equally to the $D2$ histogram. We will use the term **ALL** to refer to original $D2$ histogram generated by using all of the distance calculations over the entire set of sample points. We introduce three additional histograms, separating the pairs of points based on geometric properties of the line connecting them. Distance measures are classified into three, non-intersecting, groups, as shown in Figure 2:

   - The line connecting the 2 points lies completely *inside* the model: the **IN distances**;
   - The line connecting the 2 points lies completely *outside* the model: **OUT distances**;
   - The line connecting the 2 points passes both *inside and outside* of the model: **MIXED distances**.

   In this way, the sampling creates four $D2$ histograms, one for the cumulative set of distances and one each for those in the distinct categories.

2. **Accumulate Distribution of Classifications.** Given that distance measures are each classified as **IN**, **OUT** and **MIXED** we can make the observation that **IN%**+**OUT%**+**MIXED%**=**ALL%**=100% of the total point-pair distances, so we can use the breakdown of percentages to classify the model. For example, consider the following case of three models: X with a 10%, 30%, 60% distribution of distance categories; Y with a 15%, 27%, 62%; and Z with a 40%, 40%, 20%. It is likely that the correlation of the histograms of X and Y is more significant than any correlation between Z and X or Y. We show how to use these distributions of distances to create a set of weighted comparisons that are more finely tuned than the gross shape comparison techniques.

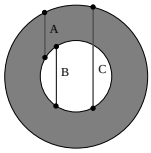


Figure 2: Shown in a 2D example, the classification of point-pair distances: IN (A), OUT (B) and MIXED (C).

3.2.4 Constructing the Shape Distribution Histograms

Four histograms are constructed using a $D2$ shape distribution, one for each of the point-pair categories: **ALL** point pairs, **IN** point pairs, **OUT** point pairs, and **MIXED** point pairs. For each set of point-pair distance data, we compute the average over all of the distances. To create the histogram, we select a bin-width based on this average distance between points in the sample—in this case, $1/50$ of the average distance between pairs of points. Having the bin-width based on the average builds the normalization of the curves into the histogram, hence a separate normalization step is not needed. The histograms are a plot of *probability* vs. *distance*, i.e., the percentage of the distances fall into each bin. In each of these histograms the total area under the curve equals 1, or 100%, of the point-pair distances.

Example. An example of the four classified shape distribution histograms for two solid models are shown in Figure 3. Its clear from the picture that, while the models exhibit a similar "bell-curve" in their **ALL** and **MIXED** distributions, they differ greatly in their **IN** and **OUT** distributions. Further, the distribution of the classifications varies widely: for example, only 3.59% of sampled points are classified as **IN** for the cheese part, whereas over 23% are classified as **IN** for the Socket. Immediately we can tell these two models are very different.

Further, one can correlate the detailed patterns in the **IN** and **OUT** distributions with the *features* of the parts. For example, in the machining domain, features such as slots, holes, pockets etc. are all material removal operations. In a shape distribution, these manifest themselves as a set of surfaces on which random points may be located. The **OUT** point-pairs capture shape properties left by these removal operations; the **IN** point-pairs capture relationships like walls and distances between features. While the correlation is not explicit, the **IN** and **OUT** distributions capture low-level shape properties left by feature instances.

3.2.5 Comparing Shape Distribution Histograms

After the shape distributions are constructed for a set of solid models, models can be compared to produce *dissimilarity measures*. There are a number of different techniques for computing the distances between shape functions (i.e., earth mover's, etc). We have experimentally confirmed Osada's [10] empirical results that dissimilarity measures based on Minkowski L_N norms, in particular the Probability Density Function (pdf) L_1 norm, are the most accurate. The L_1 norm is calculated as:

$$L_1(h_1,h_2) = \sum_{i=0}^{n} |h_{1_i} - h_{2_i}|$$

where $L_1(h_1,h_2)$ is the resulting total distance between shape distribution histograms h_1 and h_2 summed across the i buckets (i.e., total the difference in each of the n buckets of the histogram). Note that, for comparisons of L_1 norms to work there must also be a normalization step to account for differences in scale. For example, variation in the scale of a model S will produce different sets of sample distances—roughly proportional to the difference in the scale of the model. We normalize the shape distribution in our sets of histograms by aligning the mean sample values.

However, different shapes may have different numbers of bins in their $D2$ curves—due to the distribution of distances (i.e., wider distribution of distances usually implies the need for more bins). When two models with different numbers of bins are compared, the histogram for the model with fewer bins is padded with 0 value bins. Due to these differences in numbers of bins, we introduce variation on the L_1 norm distance measure. After the smaller histogram is padded with 0 value bins, we calculate the distance as the PDF L_1 norm between the two histograms *divided* by the number of bins in the histograms. The result is that the distance $D(h_1,h_2)$ between two histograms h_1 and h_2 with n bins is calculated as:

$$D(h_1,h_2) = \frac{L_1(h_1,h_2)}{n}$$

which is just the average difference between corresponding values of the two histograms. This is necessary because, if just the PDF L_1 norm is used, two very similar models each with a large number of bins will have a greater distance measure between them than two other very similar models each with a small number of bins. Dividing by the number of bins to get the average difference eliminates this effect.

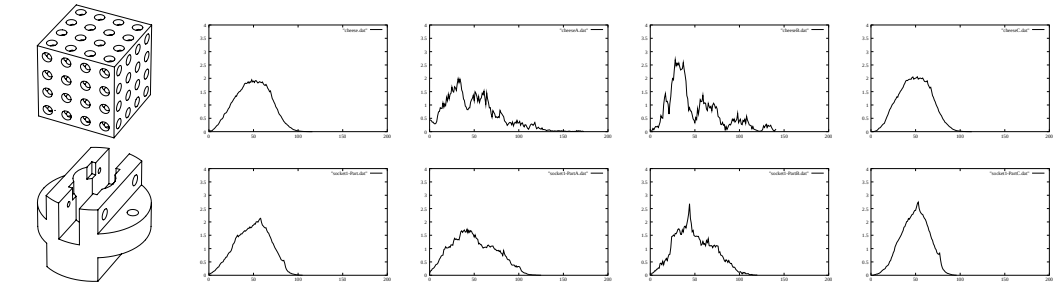
Weighted Histogram Comparisons. There are many cases where, like Figure 1, solids that are not similar have similar shapes distributions when compared under L_1 norms. Our first observation is that, when the **IN**, **OUT** and **MIXED** histograms are considered, the ability to discriminate between solids is greatly increased. This is illustrated in Figure 3. The second is that the distribution of distance classifications is a unique identifier of shape (both global and local) of the solid model. Consider the classification data shown in Table 1 for six of the solid models in Figure 3 and Figure 4. Intuitively,these models fall into several obvious groups: the two Boeing parts, the two TEAM parts, the Socket part and then the cheese part. The classification of the sampling data in Table 1 can be seen to corroborate this intuition, most strongly with the two Boeing parts whose classified shape distributions are each within 1.5% of each other.

Model	% **IN**	% **OUT**	**MIX**
Cheese	03.59%	01.66%	94.74%
Socket	23.92%	04.76%	71.31%
Boeing	16.43%	04.80%	78.75%
Simple Boeing	16.64%	05.76%	77.58%
Team	16.09%	00.79%	83.10%
Team2	25.40%	00.93%	73.66%

Table 1: Distribution of Classifications: Percentages of sampled distances falling into each of the three classifications, IN, OUT, MIXED, for some of our test models.

It is clear that the distributions of classifications should be useful in increasing the accuracy of the distance measurements between models. We have studied several ways to combine the histograms, including using the distributions as weights in the distance measure. Some of our experiments include:

1. **AvgOf3**: the average of the D distances for **IN**, **OUT**, **MIXED**;



Solid Models	**ALL**	**IN**	**OUT**	**MIXED**
*D*2 Distance	3.97	18.62	36.87	13.76
Cheese / Socket	100% / 100%	3.59% / 23.92%	1.66% / 4.76%	94.74% / 71.31%

Figure 3: The ALL points histograms for the Gupta Socket and Cheese models were fairly similar, but note that discrimination is increased with the other three (IN, OUT, MIXED) shape distributions. The actual difference in the *D*2 histogram, as computed with the modified L_1 norm, is shown; as is the distribution of the classifications.

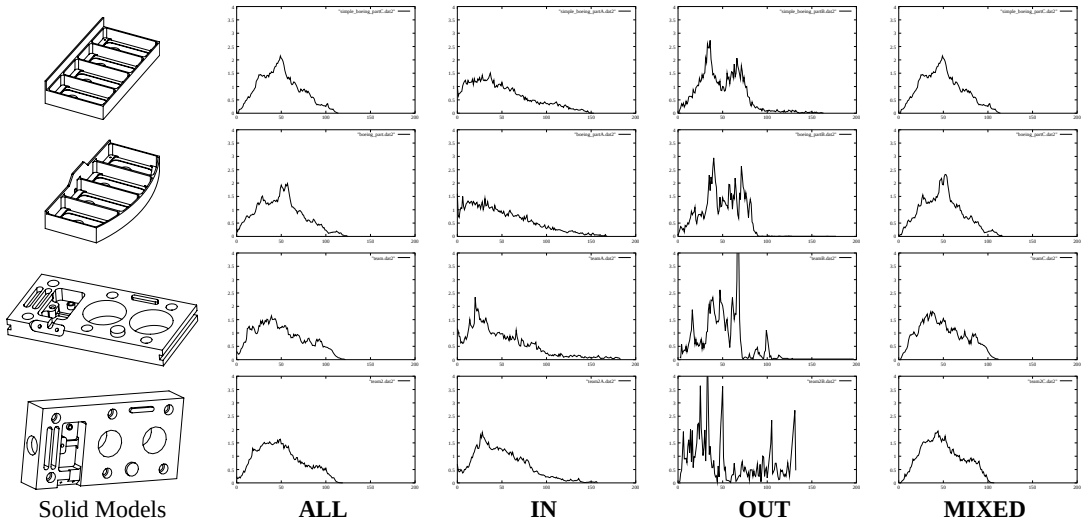


Figure 4: The histograms of four solid models of mechanical parts, broken out into ALL point, IN, OUT, MIXED shape distributions. Its evident that there resemblances are strong between the two aerospace parts (the top two) and the two TEAM parts (the bottom two), but not as strong among the other pairs.

2. **AvgOf4**: the average of the *D* distances for **ALL**, **IN**, **OUT**, **MIXED**;

3. **Weighted**: Similar solid models will have similar distributions of classifications among the **IN**, **OUT** and **MIXED** histograms. An appropriate weighting scheme will bias the comparisons *toward* similarity if the distributions percentages are closer and *away* from similarity if the distributions are greatly different in each of the classifications.

   Therefore, if the distribution percentages for a certain histogram (e.g., for the **IN** histogram) for both models is the same, we weight the distance measure to compare these histograms directly. Hence, a difference of 0 between distribution percentages results in a weight of 1. The greater the difference between distribution percentages for a certain histogram for both models, the greater the weight applied to the distance between the histograms. Also, this distance can not be greater than 1 since **IN**+**OUT**+**MIXED**$= 1$.

Let A and B be two solid models and h_A and h_B be two of their histograms from the same classification. For our experiments in this paper, we selected $\frac{1}{(1-\texttt{Diff}(\%h_A-\%h_B))}$, where $\texttt{Diff}=\texttt{abs}(\%h_A-\%h_B)$ and $\%h_A$ and $\%h_B$ are the percent (e.g., 30% would imply a value of 0.30) of total samples classified in the particular category for models A and B respectively. At $\texttt{Diff}=0$, this function has a value of 1. Note that $\lim_{\texttt{Diff}\to 1}\frac{1}{(1-\texttt{Diff})}=\infty$, hence this function has the appropriate measurement biases. The resulting weighted distance between histograms is calculated as:

$$\frac{1}{1-\texttt{Diff}(\%A_{\text{IN}},\%B_{\text{IN}})}D(A_{\text{IN}},B_{\text{IN}})+$$
$$\frac{1}{1-\texttt{Diff}(\%A_{\text{OUT}},\%B_{\text{OUT}})}D(A_{\text{OUT}},B_{\text{OUT}})+$$
$$\frac{1}{1/\texttt{Diff}(\%A_{\text{MIXED}},\%B_{\text{MIXED}})}D(A_{\text{MIXED}},B_{\text{MIXED}})$$

While our weighting scheme is preliminary, it yields empirically pleasing results. Some of these results are shown in Table 2.

	ALL	AvgOf3	AvgOf4	Weighted
Boeing–Simple_Boeing	4.33	8.72	7.62	26.41
Boeing–Team	13.86	22.35	20.23	69.41
Boeing–Team2	10.65	22.80	26.85	85.10
Simple_Boeing–Team	15.13	19.45	20.89	65.45
Simple_ Boeing–Team2	11.30	23.39	27.42	86.95
Team–Team2	10.53	24.00	28.49	88.80
Cheese–Socket	3.97	18.30	23.08	79.40

Table 2: Differences in $D2$ histograms under a number of different weighting schemes for combining the distribution data with the classification of point-pairs.

3.2.6 *Examples*

Figure 4 shows how our methodology can be used to construct a distance matrix among solid models. Figure 4 shows four solid models, each with their four categorized $D2$ histograms.

4. EXPERIMENTAL RESULTS

We have applied this technique to a set of over 1000 real-world solid models of mechanical parts from the National Design Repository. We use the histograms to create an index to this database of solid models and show how it can support a real-time query-by-example scenario.

The methods have been implemented in Java and run on both Sun Solaris SPARC and Microsoft/Intel-based computers.

Query by Example. We generated histograms for 1094 CAD models in the National Design Repository (`http://www.designrepository.org`), models created by a number of different CAD and modeling systems: ACIS, SDRC I-DEAS, Pro/ENGINEER and Unigraphics among them. We queried this test-set with example models to demonstrate the potential of our technique. The first kind of query corresponds to querying the database with an example model which is present in the database—ideally, we should like to find a "hit". The second kind of query corresponds to querying the database with an example model which is not in the database—ideally, we would like to confirm that the models returned are similar to the query model. Here we present the top five matches for each query.

In the first category, where we are looking for "hits", we selected "goodpart3" from the University of Maryland at College Park from our database for the query. Results are presented in Figure 5 as **Query 1**. The query returns the matched part along with two other very similar parts "goodpart1" and "goodpart2" and two rather dissimilar parts "impleller" and "spacer". The differences among distances in between five returned parts and the query part intuitively represent the degree of similarity. For the "goodpart" parts, the only variation in the solids models is in the existence and size of the fillet and blend surfaces. The distance to the query's matching part in the database returns a difference of less than 5 (recall that the query part is subjected to a new faceting and random point selection, hence its distance to the part in the database is unlikely to be zero). The two similar parts are returned with a distance of approximated 10 and 15, the other dissimilar parts returns a distance of greater than 20. A similar set of results is shown for the Gupta's well-known Socket example in Figure 5 as **Query 2**. For this query, the weighted histogram measurement found the Socket, as well as two other very similar variations on the socket, right at the top of the list of "hits".

For the next set of queries from the Design Repository, we selected models that did not exist in the database. The first model is a simple bracket from UMD-CP which happened to have several variations in the database. The query results, shown in Figure 5 as **Query 3**, returned each of these variations as the nearest matches to the simple bracket. The second model is the compressor housing, shown in Figure 5 as **Query 4**. These results returned a set of models all with a similar dominance of features found in cast-then-machined parts: rounded surfaces with machined finishing features. The third model is "linkage arm 42," from the variable radius Spectrometer assembly from the National Institute of Standards and Technology (NIST). Results are presented in Figure 5 for **Query 5**. In this case, "linkage arm 42" does not exist in the database yet the query returns a very similar part "linkage arm 43"—interestingly, from the same Spectrometer assembly. This time all top 5 matches show a distance about 30.

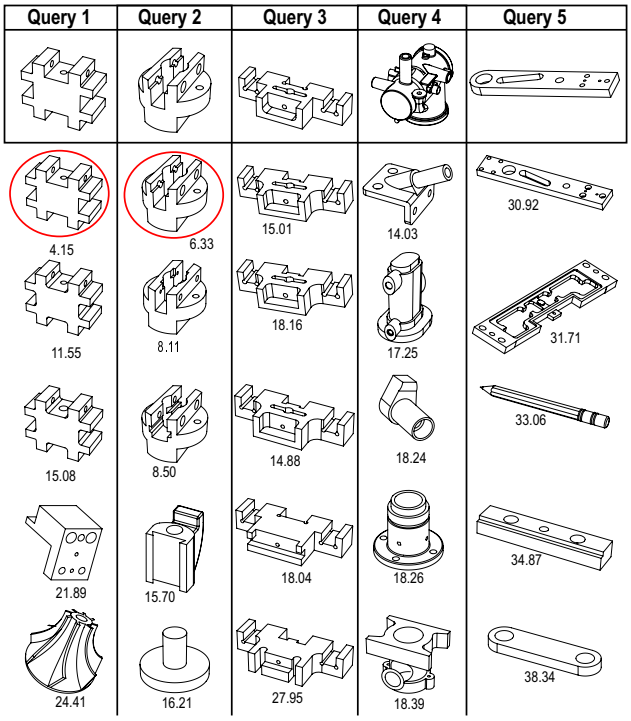


Figure 5: Query by example: Using the weighted distance measure on the $D2$ histograms to find models in the database, as well as similar models. The numbers refer to the weighted $D2$ distance measure on the classified histograms.

5. DISCUSSION AND CONCLUSIONS

This research presents a new approach to compare 3D solid models based on shape distributions. Our methodology works regardless of the underlying BRep modeling representation, allowing us to meaningfully compare models generated by any CAD system. This work is a unique fusion of ideas of research trends in computer vision, computer graphics and databases that has the potential of wide applicability across a broad spectrum of solid modeling application areas. Additional contributions of this work include introduction of novel refinements to general shape distribution techniques that enhance their discrimination abilities and enable us to answer meaningful CAD and engineering questions.

One significant potential application of this work is as a means of

comparing CAD models to real, acquired, physical models. Nearly all previous work based on sensor data has focused on the problem of model reconstruction: generating a CAD model (or approximate CAD model) from range or other sensor data. We believe our technique is the first that could be used to register sensory input against actual CAD models. This could enable new applications, i.e. for discrete parts manufacturing, one could interface sensors on the factory floor to part catalogs and process plan repositories.

This work is also the first technique for comparing solid models regardless of their native representation format. While working off of an approximation of the model (the facets or mesh) is lossy, we show that meaningful questions can be answered with this approach. A potential application of this work is to provide shape-based retrieval tools in product data management systems.

One area for future mathematical study would be to analyze how the quality of the matching (as well as computational complexity) varies with the fidelity, ε, of the faceting. We also intend to explore more sophisticated statistical methods for creating distance measures that combine the four categorized $D2$ shape histograms.

Lastly, we are working to integrate the distribution-based retrieval techniques we presented in this paper with other approaches, based on model topology, design features and machining features, to create a hybrid CAD-model database environment. Our goal is to support a diverse portfolio of database views that are capable of answering a wide variety of practical engineering questions.

Acknowledgements. This work was supported in part by National Science Foundation (NSF) Knowledge and Distributed Intelligence in the Information Age (KDI) Initiative Grant CISE/IIS-9873005; CAREER Award CISE/IIS-9733545 and Office of Naval Research (ONR) Grant N00014-01-1-0618. Additional support has been provided by Honeywell FM&T, AT&T Labs and Lockheed Martin. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation or the other supporting government and corporate organizations.

6. REFERENCES

[1] Vincent Cicirello and William Regli. Machining feature-based comparisons of mechanical parts. In *International Conference on Shape Modeling and Applications*, pages 176–187. ACM SIGGRAPH, the Computer Graphics Society and EUROGRAPHICS, IEEE Computer Society Press, Genova, Italy, May 7-11 2001.

[2] Vincent Cicirello and William C. Regli. Resolving non-uniqueness in design feature histories. In David Anderson and Wim Bronsvoort, editors, *Fifth Symposium on Solid Modeling and Applications*, New York, NY, USA, June 8-11 1999. ACM, ACM Press. Ann Arbor, MI.

[3] Kurt D. Cohen. Feature extraction and pattern analysis of three-dimensional objects. Master's thesis, Dartmouth College, Thayer School of Engineering, Hanover, NH, 1996.

[4] Alexei Elinson, Dana S. Nau, and William C. Regli. Feature-based similarity assessment of solid models. In Christoph Hoffman and Wim Bronsvoort, editors, *Fourth Symposium on Solid Modeling and Applications*, pages 297–310, New York, NY, USA, May 14-16 1997. ACM, ACM Press. Atlanta, GA.

[5] Amarnath Gupta and Ramesh Jain. Visual information retrieval. *Communications of the ACM*, 40(5):71–79, May 1997.

[6] Masaki Hilaga, Yoshihisa Shinagawa, Taku Kohmura, and Tosiyasu L. Kunii. Topology matching for fully automatic similarity estimation of 3d shapes. In *SIGGRAPH*, pages 203 – 212, New York, NY, USA, August 2001. ACM, ACM Press.

[7] David McWherter, Mitchell Peabody, Ali Shokoufandeh, and William Regli. Database techniques for indexing and clustering of solid models. In Deba Dutta and Hans-Peter Seidel, editors, *Sixth ACM/SIGGRAPH Symposium on Solid Modeling and Applications*, pages 78–87. ACM, ACM Press, June 4-8. Ann Arbor, MI 2001.

[8] David McWherter, Mitchell Peabody, Ali Shokoufandeh, and William Regli. Solid model databases: Techniques and empirical results. *ASME/ACM Transactions, The Journal of Computer and Information Science in Engineering*, 1(4):300–310, December 2001.

[9] David McWherter, Mitchell Peabody, Ali Shokoufandeh, and William Regli. Transformation invariant similarity assessment of solid models. In *ASME Design Engineering Technical Conferences*. ASME, ASME Press, September 9-12. Pittsburgh, PA 2001. DETC2001/DFM-21191.

[10] Robert Osada, Thomas Funkhouser, Bernard Chazelle, and David Dobkin. Matching 3d models with shape distributions. In 154-166, editor, *International Conference on Shape Modeling and Applications*. ACM SIGGRAPH, the Computer Graphics Society and EUROGRAPHICS, IEEE Computer Society Press, Genova, Italy, May 7-11 2001.

[11] Madhumati M. Ramesh, Derex Yip-Hoi, and Debasish Dutta. A decomposition methodology for machining feature extraction. In *ASME Design Engineering Technical Conferences, Computers in Engineering Conference*, New York, NY, USA, September 10-13, Baltimore, Maryland 2000. American Assocation of Mechanical Engineers, ASME Press. DETC2000/CIE-14645.

[12] William C. Regli and Vincent Cicirello. Managing digital libraries for computer-aided design. *International Journal of Computer Aided Design*, 32(2):119–132, Februrary 2000. Special Issue on *CAD After 2000*. Mohsen Rezayat, Guest Editor.

[13] M. Sipe and D. Casasent. Global feature space neural network for active object recognition. In *Int'l Joint Conference on Neural Networks*, 1999.

[14] Michael A. Sipe. *Feature Space Trajectory Methods for Active Object Recognition*. PhD thesis, Carnegie Mellon University, Department of Electrical and Computer Engineering, 5000 Forbes Avenue, Pittsburgh, PA 15213, December 1999.

[15] Tien-Lung Sun, Chuan-Jun Su, Richard J. Mayer, and Richard A. Wysk. Shape similarity assessment of mechanical parts based on solid models. In Rajit Gadh, editor, *ASME Design for Manufacturing Conference, Symposium on Computer Integrated Concurrent Design*, pages 953–962. ASME, Boston, MA. September 17-21. 1995.

[16] A. Talukder and D. Casasent. Nonlinear features for classification and pose estimation of machined parts from single views. In *Proc. of the SPIE, Vol. 3522*, pages 16–27. SPIE, November 1998.

[17] W.B.Thompson, J.C. Owen, H.J. de St. Germain, S.R. Stark Jr., and T.C. Henderson. Feature-based reverse engineering of mechanical parts. *IEEE Transactions on Robotics and Automation*, 12(1):57–66, Feburary 1999.