Selecting Distinctive 3D Shape Descriptors for Similarity Retrieval

Philip Shilane and Thomas Funkhouser
Department of Computer Science, Princeton University
35 Olden Street, Princeton, NJ 08540
{pshilane,funk}@cs.princeton.edu

Abstract

Databases of 3D shapes have become widespread for a variety of applications, and a key research problem is searching these databases for similar shapes. This paper introduces a method for finding distinctive features of a shape that are useful for determining shape similarity. Although global shape descriptors have been developed to facilitate retrieval, they fail when local shape properties are the distinctive features of a class. Alternatively, local shape descriptors can be generated over the surface of shapes, but then storage and search of the descriptors becomes unnecessarily expensive, as perhaps only a few descriptors are sufficient to distinguish classes. The challenge is to select local descriptors from a query shape that are most distinctive for retrieval.

Our approach is to define distinction as the retrieval performance of a local shape descriptor. During a training phase, we estimate descriptor likelihood using a multivariate Gaussian distribution of real-valued shape descriptors, evaluate the retrieval performance of each descriptor from a training set, and average these performance values at every likelihood value. For each query, we evaluate the likelihood of local shape descriptors on its surface and lookup the expected retrieval values learned from the training set to determine their predicted distinction values. We show that querying with the most distinctive shape descriptors provides favorable retrieval performance during tests with a database of common graphics objects.

Keywords: shape retrieval, geometric matching, shape database, shape distinction.

1 Introduction

Databases of 3D shapes have become widespread in a variety of applications including computer graphics, mechanical CAD, molecular biology, and medicine. Searching databases for similar shapes is a fundamental task that can sometimes speed up design or discovery processes, and thus it is an active research problem.

The goal of this project is to develop an effective shape-based method to retrieve similar objects from a database of 3D shapes. The key intuition behind our work is the importance of focusing the shape matching process on local features of shapes that are both consistent within a class and distinguishing from other classes. Many previous approaches have represented each shape with a single shape descriptor for the entire shape. During retrieval, the global shape descriptor for the query shape is compared against the descriptors for all of the shapes in the database, and the results are sorted by similarity to the query descriptor. A drawback of using global descriptors is the underlying assumption that objects within a class have overall similar shape. Instead, other researchers have matched multiple local shape descriptors accounting for the fact that certain classes have articulations or optional parts. We focus on analyzing a database of shapes to determine which local features are distinctive and will therefore provide the best retrieval performance. As shown in Figure 1, some local regions of shape are more useful than others for retrieving objects in the same class – our goal is find them.

Our work is based on using local features of each shape to improve retrieval performance [4, 14]. An object is represented by a set of local shape descriptors, each representing a (possibly overlapping) region of the shape. Then, the similarity between two shapes is related to the similarity between the two sets of local descriptors, so a chair without armrests may match well to a chair with armrests because of consistency among the other local features. Unfortunately, using many local shape descriptors dramatically increases the size of the descriptor database and slows retrieval time. Therefore, we are motivated to investigate methods for selecting a small subset of shape descriptors to be used for matching 3D objects.

Previous methods have described how to select subsets of descriptors for matching based on saliency [9] or likelihood [14], for example. In contrast, our approach is to select features that are distinctive of a class. We map shape descriptors into a space parameterized by their likelihood and learn (from a training set) the retrieval value of each likelihood – i.e. how distinctive they are. Then, we predict the expected retrieval performance of each query descriptor by evaluating the likelihood of the descriptor and looking up the expected retrieval performance associated with the likelihood value in the training set. We select a subset of descriptors from the query with high expected retrieval per-

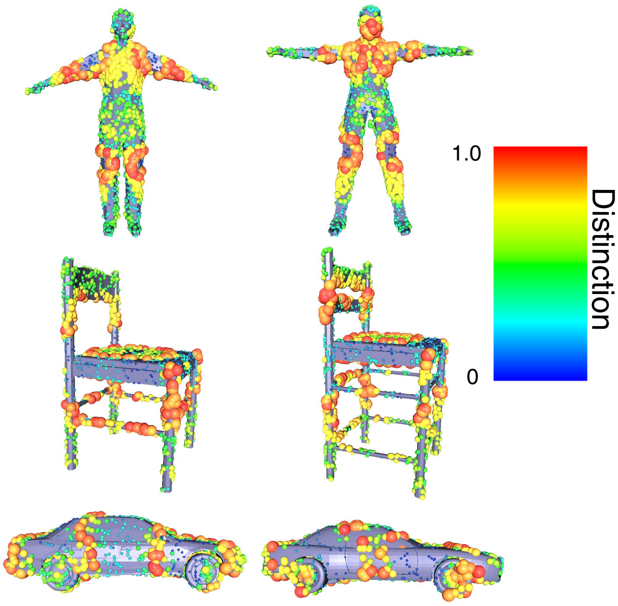


Figure 1. Predicted distinction scores are shown across three classes of shapes with higher scores shown in red. We find that selecting a small subset of the most distinctive features provides retrieval performance equivalent to using all local features at significantly less cost.

formance, i.e. the ones expected to be distinctive of the object class.

In this paper, we address the research problem of predicting shape distinction and using it for retrieval. Specifically, we make the following contributions: (1) the definition of a mapping function for shape descriptor likelihood that separates descriptors with good retrieval performance, (2) an algorithm for learning the retrieval performance of descriptors from a training set, and (3) a method for matching shapes using only the most distinctive shape descriptors. We find that using a small set of distinctive descriptors for retrieval performs almost as well as using the full set of descriptors while being significantly faster.

The remainder of this paper is organized as follows: The next section provides a summary of previous work for local shape descriptors and shape distinction. Section 3 describes how shape distinction can be used to improve local matching for retrieval. In Section 4, we define a mapping function based on the likelihood of shape descriptors and then show how to learn the retrieval performance of each descriptor from a training set in Section 5. We show how to select a small set descriptors for each shape that are predicted to be distinctive in Section 6 and how to use multiple descriptors for retrieval in Section 7. In Section 8, we provide empirical results demonstrating our definition of shape distinction is useful for retrieval. We summarize our results and describe future work in Section 9.

2 Related Work

In this overview of previous work, we present two main categories: (1) shape descriptors for various properties of shapes and (2) techniques for selecting important local descriptors for retrieval. For a thorough summary of shape matching and a comparison of techniques, see [12, 25].

3D Shape Descriptors: The shape matching problem has largely focused on whole-to-whole shape matching. Various descriptors have been proposed for representing the distribution of a shape's surface area [1, 3, 8, 14, 22, 26]. As two examples, the Mass per Shell Shape Histogram descriptor [1] creates a histogram of the amount of surface area in concentric shells, and the Spherical Harmonic Descriptor [16] represents the distribution of surface area in each shell as a series of spherical harmonic coefficients. Although global shape descriptors have shown good performance on many data sets, they have an underlying assumption that shapes from the same category have similar overall shape.

Part-to-part matching addresses the problem of articulated or missing parts by using local descriptors of the shape. Spin images [14] were developed for scanned objects by creating a cylindrical projection of local sets of surface points represented as an image. Local surface curvatures [4] and Shape Contexts [2, 7, 17] have also been used for describing the shapes of local regions. A drawback with these approaches is that matching time is related to the number of descriptors, and thus retrieval times are quite slow for large databases.

Distinctive Features: Selecting a subset of local descriptors is a well known technique for speeding up retrieval, and several researchers have proposed different methods for this task.

The simplest technique is to select local descriptors from the query shape randomly [7, 20]. Mori et al. [20] found that randomly selecting several 2D Shape Context descriptors had low pruning error. Unfortunately, with random selection, there is no consideration of whether the selected descriptor is useful for discrimination, and thus a greater number of descriptors may be required to achieve the same retrieval performance.

There have been several attempts to select regions that humans find salient for recognition. Early motivation for research on salient features comes from psychophysical experiments, which showed that the human visual system decomposes complex shapes into parts based on curvature and processes salient features before higher level recognition [10]. These findings from human vision research were applied in a scene recognition system [6] using a combination of filters measuring edges and local maxima to focus search on a small portion of the scene. Gal et al. [9] augmented part-in-whole matching by considering salient features based on curvature properties. A center-surround filter of curvature across multiple scales on a shape was also considered [18] for selecting salient regions for mesh simplification and view point selection. A similar approach was used to select salient regions for shape matching [21]. A difficulty with these approaches is the focus on curvature, which typically requires a manifold surface or dense point samples. Also, while these previous projects have defined saliency based on human vision, it is not clear that a computer vision system would find the same features to be distinctive for shape descriptor matching.

Other previous work has focused on rare descriptors, under the assumption that these features are distinctive for retrieval. Using local spin images, distinctive regions were defined by Shan et al. [23] where each spin image was used as a query. A query could match multiple times to each shape in the database, and the number of matches was recorded. Descriptors that mostly matched to a single shape were considered distinctive as opposed to common descriptors that matched all shapes equally well. This approach has the advantage of being independent of the underlying type of shape descriptor but does require using each descriptor for retrieval to determine its distinction. Work by Chua at al. [4] found "selective points" by comparing local descriptors from the query to each other and selecting descriptors that failed to match other regions of the shape.

The approach most similar to our own is that of Johnson [15], where the likelihood of each descriptor was calculated based on a Gaussian distribution of the descriptors within each query model, and only the least likely descriptors were used for surface matching. We augment this approach in two important ways. First, we compute likelihood with respect to all shape descriptors in an entire database, rather than just for the query model. Second, we predict the retrieval performance for every likelihood value and then select only the shape descriptors with highest expected retrieval performance for shape matching. These differences allow our shape matching method to achieve higher performance than previous related methods (see Section 8).

3 Overview of the Approach

The goal of our project is to predict which shape features are distinctive and focus similarity retrieval on those features. Our approach is to compute shape descriptors for several regions of each shape, map them into a space parameterized by their likelihood, predict their retrieval performance based on a training set of labeled descriptors, and then select only the most distinctive descriptors to be used during retrieval.

The organization of our system is shown in Figure 2. During a training phase, a distinction function is learned. First, the shapes are normalized for scale, and then random points are generated across the surface of each shape. A shape descriptor is created, centered at each random point. Then, the likelihood of each descriptor is evaluated along with its retrieval performance in the classified training database. A histogram of retrieval performance scores is built for different descriptor likelihood values.

When a user presents a query shape to the system, distinction values are predicted for local descriptors on the query shape. First, local descriptors are generated across the surface in a manner similar to the training phase. The likelihood of each descriptor relative to the training database is

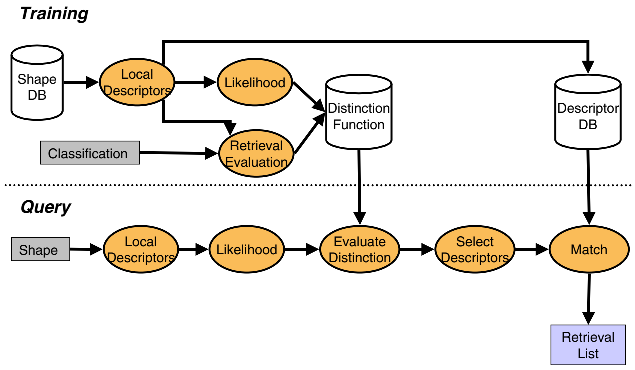


Figure 2. Diagram of training and query phases.

calculated. Then, the distinction scores of the descriptors are predicted based on their likelihood values and the distinction function learned during the training phase. A small set of the k most distinctive descriptors are then selected for the query. Each selected descriptor is matched against all descriptors in the database, and then the objects with the best sum of match scores for all k selected query scores are returned as the retrieval result.

The key step in this process is the way in which we predict distinction for every shape descriptor based on the average retrieval performance of descriptors with the same likelihood in a training set. More formally, predicted distinction function D maps descriptor $\vec{d}$ with likelihood function map into a bin representing descriptors from a training database with the same likelihood value as $\vec{d}$. We represent these training descriptors with the same likelihood as $\vec{d}$ as the set F. The predicted distinction value for $\vec{d}$ is the average retrieval performance of the descriptors $\vec{f} \in F$.

$$D(map(\vec{d})) = \frac{1}{|F|} \sum_{\vec{f} \in F} RetrievalPerf(\vec{f})$$

There are several advantages to this approach. The main advantage is that our predicted distinction function D is based on the retrieval performance of descriptors from the training database. This produces more accurate predictions than considering only descriptor likelihood. Another advantage is that D is independent of the type of descriptor, so it can be applied to many real-valued descriptors. Also, by defining a predicted distinction function in terms of descriptors mapped by likelihood, we have created a one-dimensional parameterization. This allows for a compact representation of predicted distinction as a table of average retrieval scores computed from a training set. The query descriptor with likelihood having the highest predicted distinction can be used as the query into the database. If multiple descriptors for the query shape will be used for retrieval, D provides an ordering of the descriptors. Alternatively, while descriptors are being calculated for the query shape, predicted distinction can be determined for each descriptor, and the process can end early when a descriptor with a sufficiently high distinction value is found. As such, we have a quick way to select the most distinctive descriptors for a query.

In the following sections, we investigate several research problems for creating the distinction function. We first define a likelihood model for shape descriptors and then show how to use a training set to evaluate retrieval performance. We then explain how to select a subset of the most distinctive descriptors for a query shape and use the subset during retrieval. We then evaluate D against other alternatives. Since the focus of our project is on a descriptor-independent distinction function, we rely on previous research into shape descriptors, which is not a contribution of our work.

4 Mapping from Descriptors to Likelihood

The first issue in implementing our approach is to define a mapping function that clusters shape descriptors based on their retrieval performance. The challenge is to define a mapping such that descriptors near each other in the mapped space will have similar retrieval scores and be well separated from descriptors with different scores. There are many options for a mapping function. One approach is to use the full descriptors directly, though this would affect the lookup time during prediction. Other mapping functions could use the local curvature or the descriptors' positions relative to a coordinate system such as the shape's center of mass.

We define a mapping function of shape descriptors using likelihood based on the work of [4, 15]. A rationale for this approach is that rare features (such as the wingtips and tail of the plane in Figure 3) may be discriminating for retrieval, while common areas (such as the flat portions of the wings) may match numerous categories of shapes. Likelihood mapping has the advantage of being independent of the underlying real-valued feature vector used as a shape descriptor. After descriptor statistics are estimated from the training set, the likelihood function can be evaluated quickly for queries.

A key question is then how to map descriptors to likelihoods. In previous work, Johnson et al. used a mixture of Gaussian distributions to estimate descriptor likelihoods. However, if the distribution of our descriptors is normal, then perhaps we can use a single Gaussian distribution to achieve the same performance at less cost. Based on the assumption of a normal distribution of shape descriptors, the probability density of descriptor $\overrightarrow{x}$ can be modeled by a multivariate normal distribution [5]:

$$\text{density}(\overrightarrow{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\overrightarrow{x}-\overrightarrow{\mu})^t \Sigma^{-1}(\overrightarrow{x}-\overrightarrow{\mu})}$$

with the mean $\overrightarrow{\mu}$ and covariance Σ estimated from a training set and d equal to the dimensionality of the shape descriptor.

Under floating point arithmetic, the exponential function rounds to zero for descriptors far from $\overrightarrow{\mu}$, so we work with

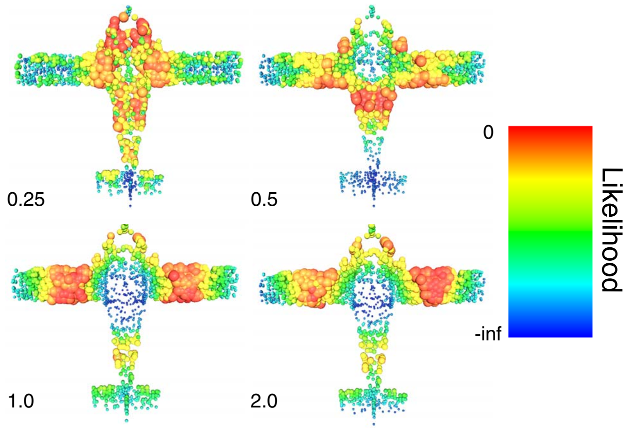


Figure 3. The likelihood of the descriptors is color coded with red indicating the most likely descriptors. Notice that the likelihood of the descriptors changes with the scale of the descriptor.

the natural log of the density function. We also drop the normalization term since we are interested in the relative density of descriptors as opposed to their exact values. We refer to this function, p, as the likelihood of a descriptor:

$$p(\overrightarrow{x}) \propto \ln(\text{density}(\overrightarrow{x}))$$
$$p(\overrightarrow{x}) = -\frac{1}{2}(\overrightarrow{x} - \overrightarrow{\mu})^t \Sigma^{-1} (\overrightarrow{x} - \overrightarrow{\mu})$$

In practice, we calculate distinction function D from the training set with p as the mapping function, therefore $map \equiv p$. Bins partitioning the likelihood space hold the average retrieval performance of the training set descriptors. Since the distribution has a long tail of low likelihood, a threshold is selected and a bin represents all descriptors with likelihood below the threshold.

To evaluate this normality hypothesis, we generated 200,000 local descriptors on 100 shapes from the Princeton Shape Benchmark (PSB) [24]. For this experiment we used a version of the Spherical Harmonic Descriptor [16] representing a local region of each shape with 512 values. We compared the distribution of these descriptors against 200,000 feature vectors randomly generated with distribution $N(0,\ 1)$ and 512 dimensions. Since our definition of likelihood incorporates a covariance matrix that accounts for correlated features, we evaluated the shape descriptors with a diagonal covariance matrix for this experiment. Figure 4 shows a quantile-quantile plot [11] comparing the shape descriptor distribution against the randomly generated feature vectors. A quantile-quantile plot is a visualization of the relationship between two distributions of data. For each $+$ marker, the horizontal position indicates the likelihood value for a quantile of the randomly generated data, and the vertical position for the maker indicates the likelihood for an equal quantile of the measured shape descriptor data. The straight line indicates the line of best fit between the distributions, which corresponds to a normal distribution with different mean and variance. While the shape descriptor distribution varies from the line of best fit, a normal distribution is a reasonable model for the majority of shape descriptors.

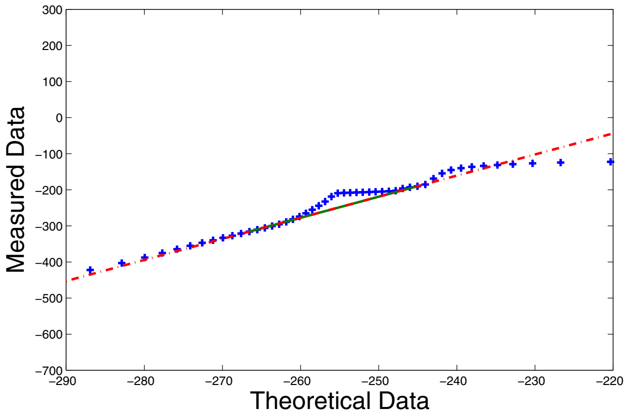


Figure 4. Quantile-Quantile plot of the likelihood of SHD descriptors against a randomly generated N(0, 1) distribution. The $+$ markers indicate the relationship between the measured and randomly generated data. A normal distribution provides a good model of the shape descriptor distribution.

5 Mapping from Likelihood to Distinction

The second step is to define a distinction function that maps descriptor likelihood to an expected retrieval score. For this step, we evaluate the retrieval performance of every local shape descriptor in a training set and build a histogram of average retrieval performance as a function of likelihood.

During a training phase, each query shape is presented to a retrieval system, and local descriptors are calculated over the shape. Consider shape X consisting of a set of N shape descriptors, $X = \{\overrightarrow{X_1}, ..., \overrightarrow{X_N}\}$. We define the correspondence between the ith feature of shape X to shape Y as the minimal distance between descriptor $\overrightarrow{X_i}$ and all of the descriptors of Y.

$$C(\overrightarrow{X_i}, Y) = \min_j |\overrightarrow{X_i} - \overrightarrow{Y_j}|$$

Given the distance from the ith feature of X to every other shape in the database, we sort the distances from smallest to largest. This sorted list is typically called the retrieval list for $\overrightarrow{X_i}$ and represents the order of retrieval results.

Then, we evaluate the quality of the retrieval list with any standard retrieval metric [19]. We prefer to use the Discounted Cumulative Gain (**DCG**) [13] because it incorporates the entire retrieval list. Correct results near the front of the retrieval list are weighted more heavily than correct results near the end under the assumption that a user is most interested in the first results.

To calculate the DCG for a descriptor, the retrieval list R is converted to a list G, where element G_i has value 1 if element R_i is in the correct class and value 0 otherwise. Discounted cumulative gain is then defined as:

$$DCG_i = \left\{ \begin{array}{ll} G_1, & i = 1 \\ DCG_{i-1} + \frac{G_i}{\log_2(i)}, & otherwise \end{array} \right\}$$

This result is then divided by the maximum possible DCG, which is achieved if the first C elements are in the correct class, where C is the size of the class:

$$DCG = \frac{DCG_M}{1 + \sum_{j=2}^{|C|} \frac{1}{\log_2(j)}}$$

where M is the number of shapes in the database. The DCG for a descriptor is between zero and one, with better retrieval performance corresponding to DCG values closer to one (Figure 5).

For every query in the training set, we evaluate both its likelihood and its DCG retrieval performance. Then, we cluster descriptors into regular bins by likelihood and average the DCG scores for all descriptors in the same likelihood bin. The result is a histogram of average DCG scores indexed by likelihood that can be used as a map from likelihood to distinction.

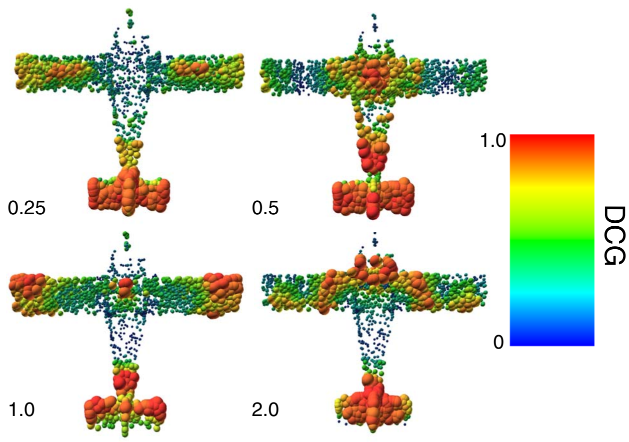


Figure 5. The retrieval performance for local descriptors over the surface is shown with red indicating the best performance. Across multiple descriptor scales, the tail region of the plane has distinctive descriptors.

6 Selecting Distinctive Descriptors

When performing a retrieval task, the dissimilarity score between the query and each shape in the database should be related to the difference between their sets of local shape descriptors. Our goal is to select the k most distinctive descriptors from each query shape to focus the matching process. One issue is that geometrically close descriptors provide little additional information about the shape, so the shape is better represented by selecting distinctive descriptors that are spread apart. Figure 6 shows that predicted distinction scores are clustered, so naively selecting descriptors with the highest scores will tend to pick descriptors near each other. There are several possible techniques for spreading out the selected descriptors such as non-maximum suppression or enforcing distance constraints.

We found that requiring a minimum distance between selected descriptors works well in practice. We sort the query descriptors based on their distinction scores and select the k descriptors with the highest scores that are a minimum distance apart. Descriptors are selected in a greedy manner, so the first descriptor in each list–with the highest predicted distinction–is always selected.

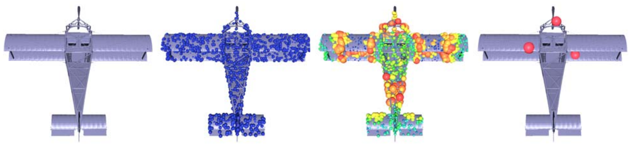


Figure 6. When a new query mesh is presented, shape descriptors are created at random positions, the predicted distinction scores D are calculated based on the likelihood of each descriptor, and a subset of distinctive descriptors is selected to be used during retrieval.

7 Querying with Selected Descriptors

The final step is to evaluate the difference between query shape X and each database shape Y. The goal is to create a distance metric that accounts for differences between local features of each shape. While there are distance metrics that account for spatial distribution of the descriptors such as [2, 21], we take a simple approach in this study: we measure the sum of distances between all k descriptors from X (represented as X^k) and the closest descriptors of Y:

$$|X^k - Y| = \sum_{i}^{k} C(\overrightarrow{X_i^k}, Y)$$

Although this distance function does not consider the amount of deformation necessary to bring the corresponding regions of the shape into alignment, it is fast to compute, and it can be considered a lower-bound on more complex geometric distance functions.

8 Results

In this section, we evaluate the value of selecting distinctive descriptors based on likelihood and learned retrieval performance. We first describe the shape database and set of shape descriptors used for our experiments, and then we address the following research questions with empirical results.

- Does mapping descriptors by their likelihood group those with similar retrieval performance?
- Can a few distinctive local descriptors improve retrieval over a single global descriptor?
- Can we speed up retrieval with a few distinctive descriptors compared to using the full set of local descriptors, while maintaining retrieval performance?
- How well do distinctive descriptors compare against other alternative approaches for selecting local descriptors?

8.1 Shape Database

We evaluated our experiments on 100 models[1] from the Princeton Shape Benchmark [24], which consists of a set of classified graphics objects. The 100 shapes, evenly divided into ten classes, represent classes that are in different branches of the hierarchical classification, so a diverse set of classes was included.

During the preprocessing phase, all shapes were scaled so that average surface points were 0.5 units from the center of mass. Then 2,000 points were generated randomly over the surface with uniform distribution to serve as the center points for local shape descriptors. Each shape descriptor was computed to include the portion of the shape that fell within a local radius of support. We experimented with radii of $\{0.25, 0.5, 1.0, 2.0\}$, where the smallest radius (0.25) represented a reasonably local shape region while the largest radius (2.0) covered the entire surface when the descriptor was positioned on a point sample on an extremity. Unless otherwise noted, all of the reported experiments are for a scale of 1.0, which generally included about 30% of the shape when positioned on an extremity.

8.2 Shape Descriptor Types

We experimented with two different shape descriptor types:

- The Mass per Shell Shape Histogram (SHELLS) [1] descriptor is a 1D histogram of the distribution of surface area relative to the center of mass. Thirty-two evenly spaced bins were used.

[1] The shape classes include: biplane, spider, human with arms out, dome church, dining chair, rectangular table, ice cream, potted plant, sedan, and tank.

- The Spherical Harmonic Descriptor (SHD) [16] is a 3D function, where the values are the composition of a Gaussian with the Euclidean distance transform of the model's surface. Then, concentric spheres are restricted to the function, and the norm of the spherical harmonic frequency is stored, making the SHD rotation invariant by construction. Thirty-two shells and sixteen frequencies are used for all experiments.

These descriptors were chosen because they are simple to compute, invariant to rotations (which simplifies matching), and they are used in several previous studies (e.g., [7]).

8.3 Mapping Functions

We first evaluated whether mapping descriptors based on their likelihood effectively groups descriptors with similar retrieval performance. For every shape descriptor, we performed a query into the database of descriptors for the 100 shapes and evaluated the likelihood of the descriptor and its retrieval performance. Figure 7 shows the resulting average retrieval performance as vertical bars for each likelihood value. The horizontal axis shows the likelihood. The left vertical axis is retrieval performance as measured by DCG, with 1 standard deviation error bars shown in cyan. The magenta line indicates the percentage of descriptors that falls within each likelihood bin. Note that the axis for the magenta line is on the right side of the plot.

We found that the most likely bin of the histogram (with 40% of the descriptors) contains descriptors with nearly the worst retrieval performance. We also find that grouping shape descriptors based on their likelihood effectively clusters descriptors with similar retrieval performances. Using a t-test, there is 99% confidence that the bin with the best performance varies significantly from the most common bin.

For comparison sake, we considered alternative mappings, such as the amount of surface area within the descriptor's radius, as well as the position of the descriptor relative to the shape's center of mass, in studies not reported here in detail. However, both alternatives failed to group descriptors with similar retrieval scores as well as likelihood.

8.4 Retrieval Results

We next evaluated whether using distinctive local descriptors can improve retrieval performance over competing methods. We performed leave-one-out experiments where we held out one model as a query and trained the distinction function over the remaining models (this maximizes the size of our training and test sets, since each of the 100 models serves as a query once and the training set has the remaining 99 models). For each query, we matched its k best descriptors to all the descriptors of the other 99 models, and then we returned the models in a ranked retrieval list in order of the sum of descriptor match scores, as described in Section 7.

Comparison to Global Shape Descriptors: Figure 8 shows a precision recall plot comparing retrieval with a single global descriptor versus using 10 descriptors with high distinction values. Recall is the percentage of shapes retrieved from the same class as the query, while precision is the percentage from the retrieval list that are from the correct class for a given recall level. Higher lines indicate better retrieval performance. For this experiment, 10 descriptors were used and the distance threshold was 0.1 on the scaled shapes. Using these distinctive descriptors improves retrieval performance beyond a single global descriptor. To be fair, shape matching with a global descriptor is faster than with local descriptors (Table 1), but the improved retrieval accuracy may be worth the extra time for certain applications.

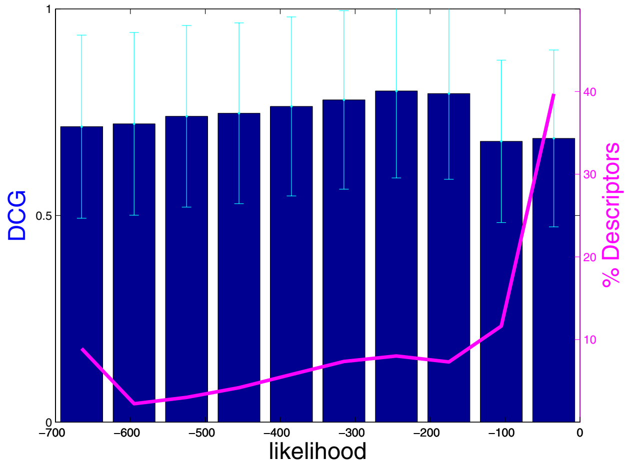


Figure 7. Using a likelihood mapping of SHD descriptors, the majority of descriptors fall within a poor retrieval group to the right. An area between the least likely and most likely descriptors tends to be better for retrieval.

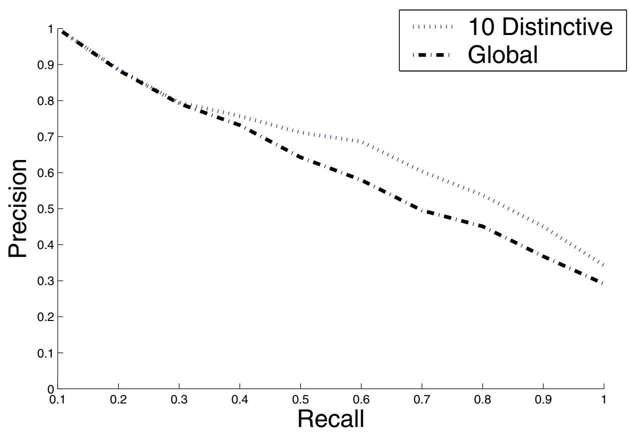


Figure 8. Using 10 distinctive SHD descriptors improves retrieval compared to using a single global descriptor.

Effect of selecting fewer descriptors: We also considered how the retrieval performance varies with k, the number of descriptors selected for each query model. Figure 9 shows the retrieval performance when using different numbers of query descriptors. For most values of $k > 3$, retrieval performance remains almost as high as when using all 2,000 descriptors. This result shows that using a small number of distinctive descriptors can approximate the retrieval result of using the full set. Meanwhile, Table 1 shows that comparing a query shape against a shape in the database using only the 3 most distinctive descriptors only takes $\frac{1}{350}$ of the time for using all 2,000 (Table 1). This combination provides a significant time savings with minimal loss of retrieval precision.

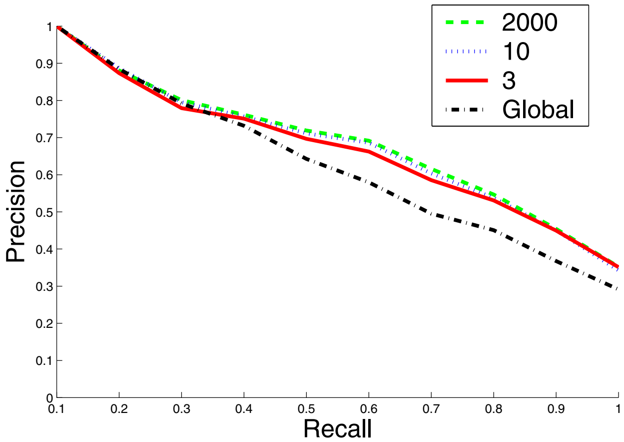


Figure 9. Performance decreases gradually as the number of distinctive SHD descriptors is reduced.

Comparison to other selection methods: We next evaluated how well our predicted distinction function compares to previous techniques for selecting local descriptors. We compare against three alternative approaches:

- Least Likely DB: For each model, the descriptors are sorted based on their likelihood as calculated based on the distribution for the entire database.
- Least Likely Model: For each model, the descriptors are sorted based on their likelihood as calculated based on the distribution of descriptors for the model.
- Random: The descriptors are randomly sorted.

Figure 10 shows the retrieval performance when combining descriptors with $k = 3$. In this plot, the vertical axis shows the percentage improvement over Random. Results for both SHELLS and SHD descriptors are shown. Selecting the k descriptors with highest predicted distinction scores outperforms Global as well as Random, Least Likely DB, and Least Likely Model for most recall values. The minimum distance constraint was set to 0.2 for the SHELLS descriptor and 0.1 for the SHD descriptor. Since the SHELLS descriptor has less descriptive power, it is necessary to require the descriptors to be farther apart. The distance constraint was also applied to the alternative techniques. It should be noted that as k increases, the difference between all of the techniques decreases, since each shape becomes fully represented with the local descriptors.

This result demonstrates that distinctive features are generally better for retrieval than other approaches that focus on likelihood without consideration of how likelihood relates to retrieval performance. While this is the only retrieval result shown for the SHELLS descriptor, our results on other experiments are consistent for both the SHELLS and SHD descriptors.

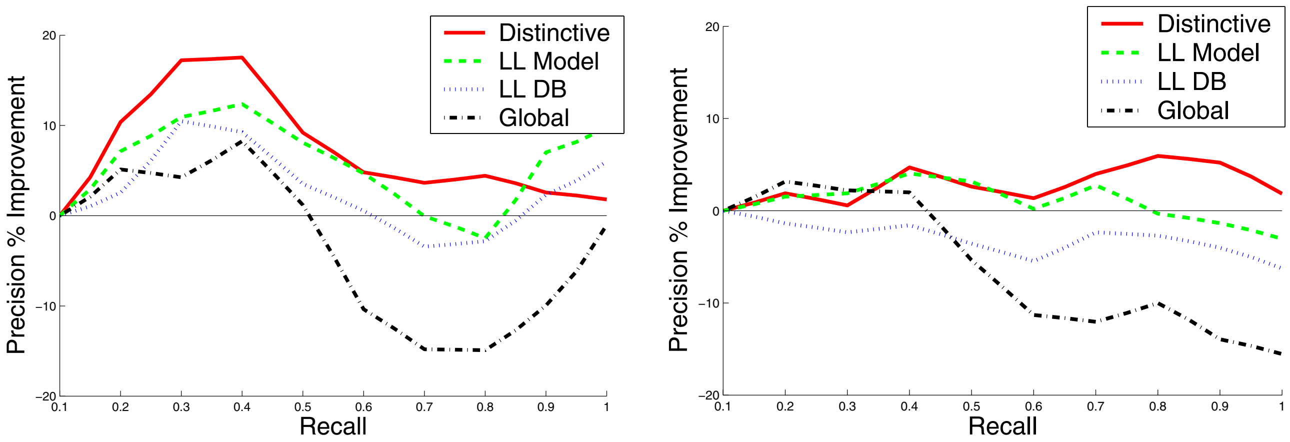
	Timing Results			
Descriptors	Generate Descriptors	Calculate Likelihood	Compare Descriptors	Retrieval DCG
Global SHD	0.35s	NA	0.000009s	0.762
3 SHD	81.5s	3.7s	0.0057s	0.785
10 SHD	81.5s	3.7s	0.018s	0.794
2,000 SHD	81.5s	NA	2.18s	0.796
Global SHELLS	0.35s	NA	0.000001s	0.638
3 SHELLS	68.7s	0.1s	0.0007s	0.679
10 SHELLS	68.7s	0.1s	0.0016s	0.718
2,000 SHELLS	68.7s	NA	0.56s	0.735

Table 1. Using a few distinctive features provides better matching results than a global descriptor and is faster than using the full set of local descriptors, with a modest decrease in retrieval accuracy. All timing results are for experiments on a computer running the Windows XP operating system on an Intel Pentium 4 processor running at 3 GHz with 1 GB of RAM.

9 Conclusion and Future Work

The main contribution of our work is a method for selecting a subset of local shape descriptors to use during matching based on shape distinction. We map descriptors based on their likelihood and calculate the average DCG for each descriptor within a likelihood bin. From this training data, we can efficiently predict the distinction score for descriptors from a query based on their likelihoods.

During our experiments, we have demonstrated several important properties of distinctive descriptors. Descriptors with similar likelihoods have similar retrieval performance. However, the least likely descriptors do not have the best retrieval performance – although they are rarest, they are not the most distinctive. Rather, ones with intermediate likelihoods provide the best retrieval performance, and thus it is valuable to store a mapping from likelihood to retrieval performance and to use that mapping for selecting query descriptors during shape matching. We find that distinctive descriptors can be combined to improve retrieval over using a single global descriptor, and a small subset of distinctive descriptors can approximate the retrieval performance of the full set while dramatically improving retrieval times. We also found that distinctive descriptors are better for retrieval than alternative approaches such as either selecting randomly or selecting the least likely descriptors.



SHELLS SHD

Figure 10. Distinctive descriptors have better retrieval performance than using randomly selected descriptors, least likely (LL) descriptors, or a global descriptor. Precision values are shown as improvement over randomly selected descriptors.

There are several limitations to our work that should be considered. Although descriptors with high predicted distinction scores are better for retrieval than those selected randomly or with other metrics, there is less improvement as the number of selected descriptors thoroughly covers the shape. This does confirm the importance of using multiple regions of each shape for matching. Assuming a normal distribution and using that parameterization of descriptors is an initial mapping approach but also suggests a range of alternatives. Other distribution models may more accurately reflect the true likelihood, but all likelihood mappings condense the descriptors to one dimension parameterized by likelihood. This has the drawback of grouping all descriptors within a shell of equal likelihood, even if there is a large variation of retrieval performance within each shell. Other groupings of descriptors may better separate those with higher retrieval scores, though increasing the dimensionality of the mapped space can adversely affect distinction calculation for query descriptors.

We feel that our general framework, mapping descriptors to a space where training performance is recorded, can be extended in several ways. We plan to consider the feature space of the descriptors directly, which is likely to provide a better clustering of retrieval performance. Finally, we believe our framework can be extended beyond retrieval to such tasks as mesh alignment and classification applications.

10 Acknowledgements

The authors would like to thank Joshua Podolak, Szymon Rusinkiewicz, and the Princeton Graphics group for useful discussions about this project. The authors would also like to thank the National Science Foundation, who provided partial funding for this work under grants CCR-0093343 and 11S-0121446, and the Air Force Research Laboratory, who provided partial funding under grant FA8650-04-1-1718.

References

[1] M. Ankerst, G. Kastenmüller, H.-P. Kriegel, and T. Seidl. Nearest neighbor classification in 3D protein databases. In *Proc. ISMB*, 1999.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522, 2002.

[3] D.-Y. Chen, M. Ouhyoung, X.-P. Tian, and Y.-T. Shen. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, pages 223–232, 2003.

[4] C. Chua and R. Jarvis. Point signatures: A new representation for 3D object recognition. *International Journal of Computer Vision*, 25(1):63–85, 1996.

[5] R. Duda, P. Hart, and D. Stork. *Pattern Classification, Second Edition*. John Wiley & Sons, New York, 2001.

[6] S. Frintrop, A. Nchter, H. Surmann, and J. Hertzberg. Saliency-based object recognition in 3D data. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, sept 2004.

[7] A. Fromme, D. Huber, R. Kolluri, T. Bulow, and J. Malik. Recognizing objects in range data using regional point descriptors. In *Proceedings of the European Conference on Computer Vision*, May 2004.

[8] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderm, D. Dobkin, and D. Jacobs. A search engine for 3D models. *Transactions on Graphics*, 22(1):83–105, 2003.

[9] R. Gal and D. Cohen-Or. Salient geometric features for partial shape matching and similarity. In *ACM Transactions on Graphics*, January 2006.

[10] D. D. Hoffman and M. Singh. Salience of visual parts. *Cognition*, 1997.

[11] R. Hogg and E. Tanis. *Probability and Statistical Inference*. Pearson Education, sixth edition, 2001.

[12] N. Iyer, S. Jayanti, K. Lou, Y. Kalyanaraman, and K. Ramani. Three dimensional shape searching: State-of-the-art review and future trends. *Computer-Aided Design*, 37:509–530, April 2005.

[13] K. Jarvelin and J. Kekalainen. IR evaluation methods for retrieving highly relevant documents. In *23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2000.

[14] A. Johnson and M. Hebert. Using spin-images for efficient multiple model recognition in cluttered 3-D scenes. *IEEE PAMI*, 21(5):433–449, 1999.

[15] A. Johsnon. Surface landmark selection and matching in natural terrain. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 413–420, June 2000.

[16] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz. Rotation invariant spherical harmonic representation of 3D shape d escriptors. In *Symposium on Geometry Processing*, June 2003.

[17] M. Krtgen, G.-J. Park, M. Novotni, and R. Klein. 3D shape matching with 3D shape contexts. In *7th Central European Seminar on Computer Graphics*, April 2003.

[18] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Transactions on Graphics (SIGGRAPH 2005)*, aug 2005.

[19] G. Leifman, S. Katz, A. Tal, and R. Meir. Signatures of 3D models for retrieval. pages 159–163, February 2003.

[20] G. Mori, S. Belongie, and J. Malik. Shape contexts enable efficient retrieval of similar shapes. *IEEE Computer Vision and Pattern Recognition (CVPR)*, December 2001.

[21] M. Novotni, P. Degener, and R. Klein. Correspondence generation and matching of 3D shape subparts. Technical Report CG-2005-2, Universität Bonn, June 2005.

[22] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin. Matching 3D models with shape distributions. *Shape Modeling International*, pages 154–166, May 2001.

[23] Y. Shan, B. Matei, H. S. Sawhney, R. Kumar, D. Huber, and M. Hebert. Linear model hashing and batch RANSAC for rapid and accurate object recognition. *IEEE International Conference on Computer Vision and Pattern Recognition*, 2004.

[24] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton Shape Benchmark. In *Proc. Shape Modeling International*, 2004.

[25] J. Tangelder and R. Veltkamp. A survey of content based 3D shape retrieval methods. In *Shape Modeling International*, June 2004.

[26] D. V. Vranic. An improvement of rotation invariant 3D shape descriptor based on functions on concentric spheres. In *IEEE International Conference on Image Processing (ICIP 2003)*, volume 3, pages 757–760, September 2003.