3D Object Retrieval Using Salient Views

Indriyati Atmosukarto, Linda G. Shapiro
Dept. Computer Science and Engineering
University of Washington
Seattle, WA, USA
indria, shapiro@cs.washington.edu

ABSTRACT

This paper presents a method for selecting salient 2D views to describe 3D objects for the purpose of retrieval. The views are obtained by first identifying salient points via a learning approach that uses shape characteristics of the 3D points [4, 3]. The salient views are selected by choosing views with multiple salient points on the silhouette of the object. Silhouette-based similarity measures from [6] are then used to calculate the similarity between two 3D objects. Experimental results show that the retrieval results using the salient views are comparable to the existing light field descriptor method [6], and our method achieves a 15-fold speedup in the feature extraction computation time.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing

General Terms

Algorithms, Performance

Keywords

3D object retrieval, 3D object signature, salient points

1. INTRODUCTION

Advancement in technology for digital acquisition and graphics hardware has led to an increase in the number of 3D objects available. Three-dimensional objects are now frequently used in a number of areas including the games industry, manufacturing companies, medical institutions, and architectural companies. The widespread integration of 3D models in all these fields motivates the need to be able to store, index, and retrieve 3D objects automatically. Current techniques for text retrieval, 2D image retrieval, and video retrieval cannot be directly translated and applied to 3D object shape retrieval, as 3D objects have different data characteristics from other data modalities.

Shape-based retrieval of 3D objects is an important area of research. The accuracy of a 3D shape-based retrieval system requires the 3D object to be represented in a way that captures the local and global shape characteristics of the objects. This is achieved by creating 3D object descriptors that encapsulate the important shape properties of the objects. This process is not a trivial task.

This paper presents our method of selecting salient 2D views to describe a 3D object. First salient points are identified by a learning approach that uses the shape characteristics of each point. Then 2D salient views are selected as those that have multiple salient points on or close to their silhouettes. The salient views are used to describe the shape of a 3D object. The similarity between two 3D objects uses view-based similarity measure developed by Chen et al. [6] for which two 3D objects are similar if they have similar 2D views.

The remainder of this paper is organized as follows. First, existing shape descriptors and their limitations are discussed. Next, we describe the datasets acquired to develop and test our methodology. The method for finding the salient points of a 3D object is described next. Then, selection of the salient views based on the learned salient points is defined. In the experimental results section, the evaluation measures are first described, and a set of retrieval experiments is described and analyzed. Finally, a summary and suggestions for future work are provided.

2. RELATED LITERATURE

Three-dimensional object retrieval has received increased attention in the past few years due to the increase in the number of 3D objects available. A number of survey papers have been written on the topic [15, 8, 13]. An annual 3D shape retrieval contest was also introduced in 2006 to try to introduce an evaluation benchmark to the research area [14].

There are three broad categories of ways to represent 3D objects and create a descriptor: feature-based methods, graph-based methods, and view-based methods. The feature-based method is the most commonly used method and is further categorized into global features, global feature distributions, spatial maps, and local features. The graph-based method describes the shape of a 3D object using the topological information of the object. The view-based method defines the shape of a 3D object using a set of 2D views taken from various angles around the object. The most effective view-based descriptor is the Light Field

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
MIR'10, March 29–31, 2010, Philadelphia, Pennsylvania, USA.
Copyright 2010 ACM 978-1-60558-815-5/10/03 ...$10.00.

Descriptor (LFD) developed by Chen et al. [6]. A light field around a 3D object is a 4D function that represents the radiance at a given 3D point in a given direction. Each 4D light field of a 3D object is represented as a collection of 2D images rendered from a 2D array of cameras distributed uniformly on a sphere. Their method extracts features from 100 2D silhouette image views and measures the similarity between two 3D objects by finding the best correspondence between the set of 2D views for the two objects.

The Light Field Descriptor was evaluated to be one of the best performing descriptors on the Princeton and SHREC benchmark databases. Ohbuchi et al. [12] used a similar view-based approach, however their method extracted local features from each of the rendered image and used a bag-of-features approach to construct the descriptors for the 3D objects. Wang et al. [16] used a similar view-based approach by projecting a number of uniformly sampled points along six directions to create six images to describe a 3D object. Zhenbao et al. [10] also generated six view planes around the bounding cube of a 3D object. However, their method further decomposed each view planes into several resolution and applied wavelet transforms to the extracted features from the view planes. Both these methods require pose-normalization of the object, however pose-normalization methods are known not to be accurate and objects in the same class are not always pose-normalized into the same orientation. Yamauchi et al. [17] applied a similarity measure between views to cluster similar views and used the centroid of clusters as the representative views. The views are then ranked based on a mesh saliency measure [9] to form the object's representative views. Ansary et al. [1, 2] proposed a method to optimally select 2D views from a 3D model using adaptive clustering algorithm. Their method used a variant of K-means clustering and assumed the maximum number of characteristic views was 40. Cyr et al. [7] presented an aspect graph approach to 3D object recognition by using 2D shape similarity metric to group similar views into aspects and to compare two objects.

We propose a method to select salient 2D silhouette views of an object and construct a descriptor for the object using only the salient views extracted. The salient views are selected based on the salient points learned for each object. Our method does not require any pose normalization nor clustering of the views.

3. DATASETS

We obtained two different datasets to develop and test our methodology. The Heads database contains head shapes of different classes of animals, including humans. The digitized 3D objects were obtained by scanning hand-made clay toys using a laser scanner. To increase the number of objects for training and testing our methodology, we created new objects by deforming the original scanned 3D models. Global deformations of the models were generated using morphing operators such as tapering, twisting, bending, stretching and squeezing. Fifteen objects representing seven different classes were scanned. The seven classes are: cat head, dog head, human head, rabbit head, horse head, tiger head and bear head. A total of 250 morphed models per original object were generated. Points on the morphed model are in full correspondence with the original models from which they were constructed.

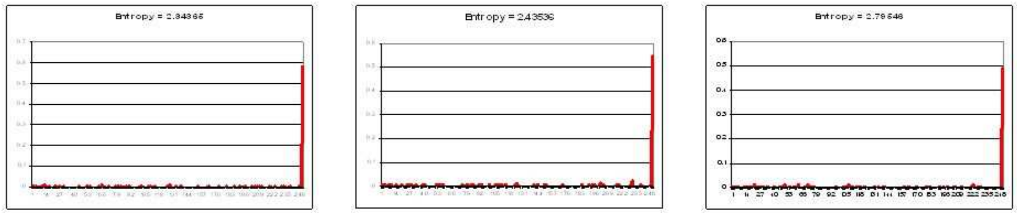
The SHREC 2008 classification benchmark database was obtained to further test the performance of our methodology. The dataset consisted of 425 pre-classified objects with a high level of shape variability. The models in the database were classified using three different levels of categorization: coarse, intermediate, and fine level. For our experiments, we used the intermediate categorization as the groundtruth in which objects are classified according to functionality and shape. In this categorization, the 425 objects in the dataset were pre-classified into 39 classes.

4. FINDING SALIENT POINTS

Our application was developed for single 3D object retrieval and does not handle objects in cluttered 3D scenes nor occlusion. The base framework of our methodology starts by applying a low-level operator to every point on the surface mesh [4, 3]. The low-level operators extract local properties of the surface mesh points by computing a low-level feature value v_i for every surface mesh point p_i. In this work, we use absolute values of Gaussian curvature and Besl-Jain surface curvature characterization [5] as the low-level surface properties. The low-level feature values are convolved with a Gaussian filter to reduce noise. After the first phase, every surface mesh point p_i will have a low-level feature value v_i depending on the operator used. The second phase of the base framework performs mid-level feature aggregation to compute a number of values for a given neighborhood of every surface mesh point p_i. Local histograms are used to aggregate the low-level feature values of each mesh point. The histograms are computed by taking a neighborhood around each mesh point and accumulating the low-level feature values in that neighborhood. The size of the neighborhood is the product of a constant c, $0 < c < 1$, and the diagonal of the object's bounding box; this ensures that the neighborhood size is scaled according to the object's size. The feature aggregation results of the base framework are used to determine salient points of an object using a learning approach.

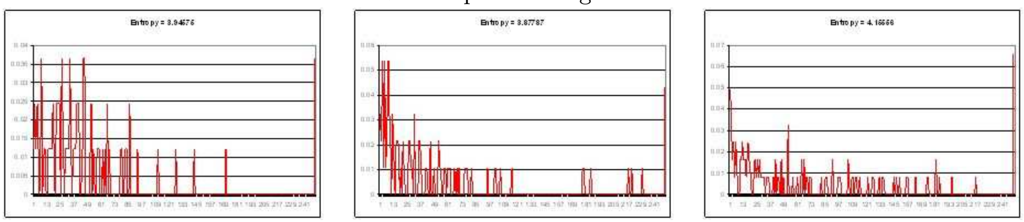
Our methodology identifies interesting or *salient points* on the 3D objects. Initially motivated by our work on medical craniofacial applications, we developed a salient point classifier that detects points that have a combination of high curvature and low entropy values. As shown in Figure 1, the salient point histograms have low bin counts in the bins corresponding to low curvature values and a high bin count in the last (highest) curvature bin. The non-salient point histograms have medium to high bin counts in the low curvature bins and in some cases a high bin count in the last bin. The entropy of the salient point histograms also tend to be lower than the entropy of the non-salient point histograms. To avoid the use of brittle thresholds, we used a learning approach to detect the salient points on each 3D object [4]. This approach was originally developed for craniofacial image analysis, so the training points were anatomical landmarks of the face, whose curvature and entropy properties are useful for objects in general.

The learning approach teaches a classifier the characteristics of points that are regarded as salient. Histograms of low-level feature values obtained in the base framework are used to train a Support Vector Machine (SVM) classifier to learn the salient points on the 3D surface mesh. The training data points for the classifier's supervised learning are obtained by manually marking a small number of salient and non-salient points on the surface of each training object.



E = 0.348 E=2.435 E=2.79

Salient point histograms

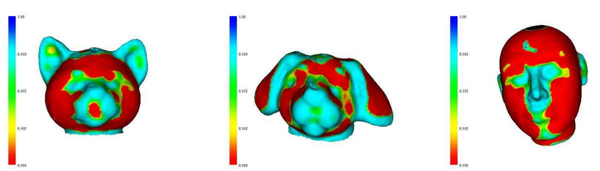


E=3.95 E=3.877 E=4.185

Non-salient point histograms

Figure 1: Example histograms of salient and non-salient points. The salient point histograms have a high value in the last bin illustrating a high curvature in the region, while low values in the remaining bins in the histogram. The non-salient point histograms have more varied values in the curvature histogram. In addition, the entropy E of the salient point histogram is lower than the non-salient point histogram (listed under each histogram).

For our experiments, we trained the salient point classifier on 3D head models of the Heads database. The salient points marked included the tip of the nose, corners of the eyes, and both corners and midpoints of the lips. The classifier learns the characteristics of the salient points in terms of the histograms of their low-level feature values. After training, the classifier is able to label each of the points of any 3D object as either salient or non-salient and provides a confidence score for its decision. A threshold is applied to keep only salient points with high confidence scores (≥ 0.95). While the classifier was only trained on cat heads, dog heads, and human heads (Figure 2), it does a good job of finding salient points on other classes (Figure 3).



(a) (b) (c)

Figure 2: Salient point prediction for (a) cat head class, (b) dog head class, and (c) human head class. Non-salient points are colored in red, while salient points are colored in different shades ranging from green to blue, depending on the classifier confidence score assigned to the point. A threshold ($T = 0.95$) was applied to include only salient points with high confidence scores.

The salient points identified by the learning approach are quite dense and form regions. A clustering algorithm was applied to reduce the number of salient points and to produce more sparse placement of the salient points. The al-

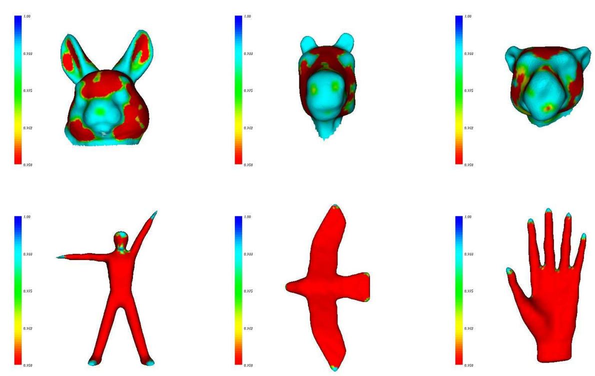


Figure 3: (Top row) Salient point prediction for rabbit head, horse head, and leopard head class from the Heads database. (Bottom row) Salient point prediction for human, bird, human head class from the SHREC database. These classes were not included in the salient point training.

gorithm selects high confidence salient points that are also sufficiently distant from each other. The algorithm follows a greedy approach. Salient points are sorted in decreasing order of classifier confidence scores. Starting with the salient point with the highest classifier confidence score, the clustering algorithm calculates the distance from this salient point to all existing clusters and accepts it if the distance is greater than a neighborhood radius threshold. For our experiments, the radius threshold was set at 5. Figure 4 shows the selected salient points on the cat, dog, and human head objects from Figure 2. It can be seen that objects from the same class (heads class in the figure) are marked with salient points in similar locations, thus illustrating the repeatability of the salient point learning and clustering method.

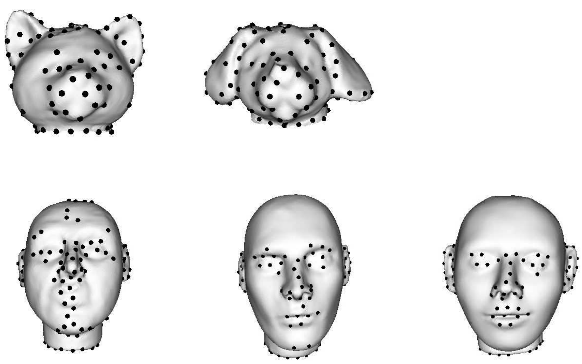
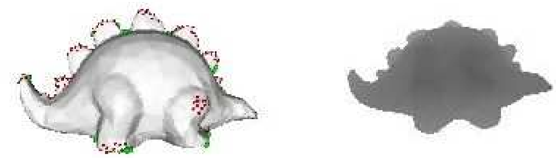


Figure 4: Salient points resulting from clustering.

5. SELECTING SALIENT VIEWS

Our methodology is intended to improve the Light Field Descriptor [6] and uses their concept of similarity. Chen et al. [6] argue that if two 3D models are similar, the models will also look similar from most viewing angles. Their method extracts light fields rendered from cameras on a sphere. A light field of a 3D model is represented by a collection of 2D images. The cameras of the light fields are distributed uniformly and positioned on vertices of a regular dodecahedron. The similarity between two 3D models is then measured by summing up the similarity from all corresponding images generated from a set of light fields.

To improve efficiency, the light field cameras are positioned at 20 uniformly distributed vertices of a regular dodecahedron. Silhouette images at the different views are produced by turning off the lights in the rendered views. Ten different light fields are extracted for a 3D model. Since the silhouettes projected from two opposite vertices on the dodecahedron are identical, each light field generates ten different 2D silhouette images. The similarity between two 3D models is calculated by summing up the similarity from all corresponding silhouettes. To find the best correspondence between two silhouette images, the camera position is rotated resulting in 60 different rotations for each camera system. In total, the similarity between two 3D models is calculated by comparing $10 \times 10 \times 60$ different silhouette image rotations between the two models. Each silhouette image is efficiently represented by extracting the Zernike moment and the Fourier coefficients from each image. The Zernike



(a) (b)

Figure 5: (a)Salient points must appear on the contour of the 3D objects for a 2D view be considered a 'salient' view. The contour salient points are colored in green, while the non-contour salient points are in red. (b) Silhouette image of the salient view in (a).

moments describe the region shape, while the Fourier coefficients describe the contour shape of the object in the image. There are 35 coefficients for the Zernike moment descriptor and 10 coefficients for the Fourier descriptor.

Like the Light Field Descriptor, our proposed method uses rendered silhouette 2D images as views to build the descriptor to describe the 3D object. However, unlike LFD, which extracts features from 100 2D views, our method selects only *salient views*. We conjecture that the salient views are the views that are discernible and most useful in describing the 3D object. Since the 2D views used to describe the 3D objects are silhouette images, some of the salient points present on the 3D object must appear on the contour of the 3D object (Figure 5).

A salient point $p(p_x, p_y, p_z)$ is defined as a *contour salient point* if its surface normal vector $v(v_x, v_y, v_z)$ is perpendicular to the camera view point $c(c_x, c_y, c_z)$. The perpendicularity is determined by calculating the dot product of the surface normal vector v and the camera view point c. A salient point p is labeled as a contour salient point if $|v \cdot c| <= T$ where T is the perpendicularity threshold. For our experiments, we used value $T = 0.10$. This value ensures that the angle between the surface normal vector and the camera view point is between 84° and 90°.

For each possible camera view point (total 100 view points), the algorithm accumulates the number of contour salient points that are visible for that view point. The 100 view points are then sorted based on the number of contour salient points visible in the view. The algorithm selects the final top K salient views used to construct the descriptor for a 3D model. In our experiments, we empirically tested different values of K to investigate the respective retrieval accuracy.

A more restrictive variant of the algorithm selects the top K *distinct salient views*. In this variant, after sorting the 100 views based on the number of contour salient points visible in the view, the algorithm uses a greedy approach to select only the distinct views. The algorithm starts by selecting the first salient view, which has the largest number of visible contour salient points. It then iteratively checks whether the next top salient view is too similar to the already selected views. The similarity is measured by calculating the dot product between the two views and discarding views whose dot product to existing distinct views is greater than a threshold P. In our experiments, we used value $P = 0.98$. Figure 6(top row) shows the top 5 salient views, while Figure 6(bottom row) shows the top 5 distinct salient views for a human ob-

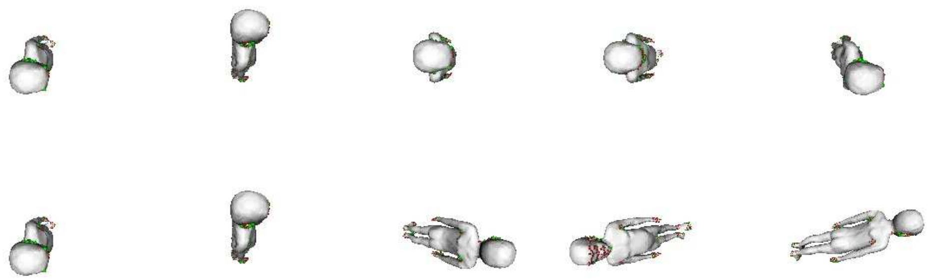


Figure 6: Top 5 salient views for a human query object (top row). Top 5 distinct salient views for the same human query object (bottom row). The distinct salient views capture more information regarding the object's shape.

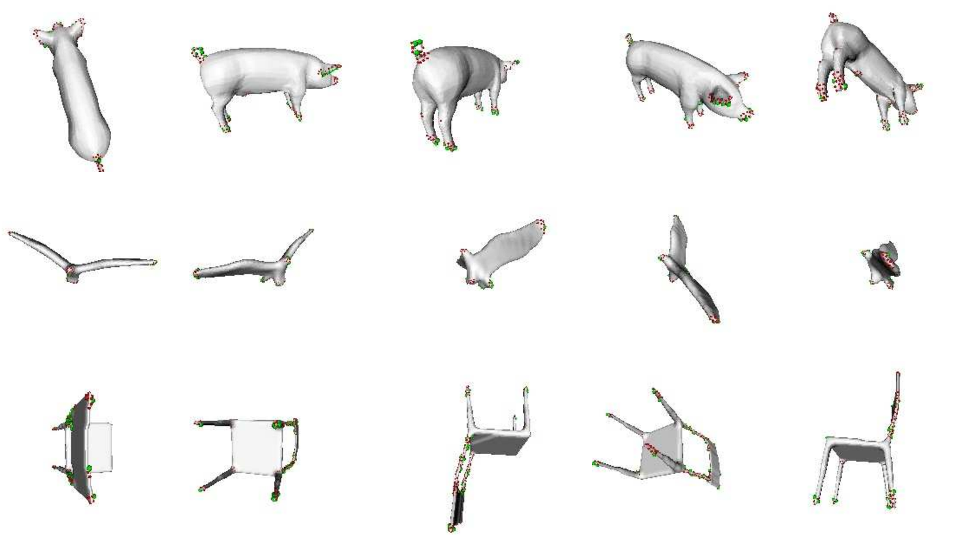


Figure 7: Top 5 distinct salient views of animal class (top row), bird class (middle row), and chair class (bottom row) from the SHREC database.

ject. It can be seen in the figure that the top 5 distinct salient views more completely capture the shape characteristics of the object. Figure 7 shows the top 5 distinct salient views for different classes in the SHREC database.

6. EXPERIMENTAL RESULTS

We measured the retrieval performance of our methodology by calculating the average normalized rank of relevant results [11]. The evaluation score for a query object was calculated as follows:

$$score(q) = \frac{1}{N \cdot N_{rel}} \left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel}+1)}{2} \right)$$

where N is the number of objects in the database, N_{rel} is the number of database objects that are relevant to the query object q (all objects in the database that have the same class label as the query object), and R_i is the rank assigned to the i-th relevant object. The evaluation score ranges from 0 to 1, where 0 is the best score as it indicates that all database objects that are relevant are retrieved before all other objects in the database. A score that is greater than 0 indicates that some non-relevant objects are retrieved before all relevant objects.

The retrieval performance was measured over all the objects in the dataset using each in turn as a query object. The average retrieval score for each class was calculated by averaging the retrieval score for all objects in the same class. A final retrieval score was calculated by averaging the retrieval score across all classes.

A number of experiments were performed to evaluate the performance of our proposed descriptor and its variants. The first experiment explores the retrieval accuracy of our proposed descriptor. The experiment shows the effect of varying the number of top salient views used to construct the descriptors for the 3D objects in the dataset. As shown in Figure 8, the retrieval performance improves (retrieval score decreases) as the number of salient views used to construct the descriptor increases. Using the top 100 salient views is equivalent to the existing LFD method. For the absolute Gaussian curvature feature (blue line graph), LFD with 100 views has the best retrieval score at 0.097; however, reducing the number of views by half to the top 50 salient views only increases the retrieval score to 0.114. For the Besl-Jain curvature feature (pink line graph), the trend is similar with a smaller decrease in performance as the number of views is reduced.

In the second experiment, the algorithm selects the top salient views which are distinct. Table 1 shows the average retrieval scores across all classes in the dataset as the number of views and number of distinct views are varied. Comparing the results, it can be seen that the retrieval scores for the top K distinct views is always lower (better) than that for the top K views. For example, using the top 5 distinct salient views achieves an average retrieval score of 0.138 compared to using the top 5 salient views with retrieval score of 0.157. In fact, using the top 5 distinct salient views achieves similar retrieval score to using the top 20 salient views, and using the top 10 distinct salient views produces a similar retrieval score as to using the top 50 salient views. Each object in the dataset has its own number of distinct salient views. The average number of distinct salient views for all the objects in the dataset is 12.38 views. Executing the retrieval with the maximum number of distinct salient views for each object query achieves a similar average retrieval score to the retrieval performed using the top 70 salient views.

The third experiment compares the retrieval score when using the maximum number of distinct salient views to the retrieval score of the existing LFD method. Table 2 shows the average retrieval score for each class using the maximum number of distinct salient views and the LFD method. Over the entire database, the average retrieval score for the maximum number of distinct salient views was 0.121 while the average score for LFD was 0.098. To better understand the retrieval scores, a few retrieval scenarios are analyzed. Suppose that the number of relevant objects to a given query is N_{rel} and that the total number of objects in the database is $N = 30$; the retrieval score is dependent on the rank of the N_{rel} relevant objects in the retrieved list. The same retrieval score can be achieved in two different scenarios. When $N_{rel} = 10$ a retrieval score of 0.2 is attained when 3 of the relevant objects are at the end of the retrieved list, while the same score value is obtained in the case of $N_{rel} = 5$ when only 1 of the relevant objects is at the end of the list. This shows that incorrect retrievals for classes with small N_{rel} value are more heavily penalized, since there are fewer relevant objects to retrieve. In Table 2 it can be seen that for classes with small N_{rel} values ($N_{rel} < 10$, the average class retrieval scores using the maximum number of distinct views are small and similar to retrieval using LFD (scores < 0.2), indicating that the relevant objects are retrieved at the beginning of the list. For classes with bigger N_{rel} values, the retrieval scores for most classes are < 0.3 indicating that in most cases the relevant objects are retrieved before the middle of the list. The worst performing class for both methods is the spiral class with a score of 0.338 using maximum distinct salient views and 0.372 using LFD; this most probably is due to the high shape variability in the class. The retrieval score using our method is quite similar to the retrieval score of LFD with only small differences in the score values suggesting that the retrievals slightly differ in the ranks of the retrieved relevant objects, with most relevant objects retrieved before the middle of the list. However, our method greatly reduces the computation time for descriptor computation.

The last experiment investigates the run time performance of our methodology and compares the run time speed of our method to the existing LFD method. These experiments were performed on a PC running Windows Server 2008 with Intel Xeon dual processor at 2GHz each and 16GB RAM. The run time performance of our method can be divided into three parts: (1) salient views selection, (2) feature extraction, and (3) feature matching. The salient view selection phase selects the views in which contour salient points are present. This phase on average takes about 0.2s per object. The feature matching phase compares and calculates the distance between two 3D object. This phase on average takes about 0.1s per object. The feature extraction phase is the bottleneck of the complete process. The phase begins with a setup step that reads and normalizes the 3D objects. Then, the 2D silhouette views are rendered and the descriptor is constructed using the rendered views. Table 3 shows the difference in the feature extraction run time for one 3D object between our method and the existing LFD method. The results show that feature extraction using the selected salient views provides a 15-fold speedup compared to using

Table 2: Retrieval score for each SHREC class using the maximum number of distinct views versus using all 100 views (LFD).

No	Class	# Objects	Avg # distinct salient views	Max distinct salient views score	LFD score
1	human-diff-pose	15	12.33	0.113	0.087
2	monster	11	12.14	0.196	0.169
3	dinosaur	6	12.33	0.185	0.169
4	4-legged-animal	25	12.24	0.274	0.186
5	hourglass	2	11.50	0.005	0.001
6	chess-pieces	7	12.14	0.085	0.085
7	statues-1	19	12.16	0.267	0.250
8	statues-2	1	13.00	0.000	0.000
9	bed-post	2	12.00	0.124	0.008
10	statues-3	1	12.00	0.000	0.000
11	knot	13	12.00	0.006	0.003
12	torus	18	11.77	0.194	0.161
13	airplane	19	12.42	0.101	0.054
14	heli	5	11.60	0.204	0.158
15	missile	9	12.00	0.306	0.241
16	spaceship	1	13.00	0.000	0.000
17	square-pipe	12	12.31	0.026	0.017
18	rounded-pipe	15	11.8	0.221	0.184
19	spiral	13	12.46	0.338	0.372
20	articulated-scissors	16	12.06	0.027	0.005
21	CAD-1	1	12.00	0.000	0.000
22	CAD-2	1	12.00	0.000	0.000
23	CAD-3	1	13.00	0.000	0.000
24	CAD-4	1	12.00	0.000	0.000
25	CAD-5	1	11.00	0.000	0.000
26	glass	7	11.86	0.144	0.245
27	bottle	17	12.12	0.093	0.081
28	teapot	4	11.50	0.075	0.015
29	mug	17	12.06	0.035	0.004
30	vase	14	12.21	0.166	0.149
31	table	4	11.50	0.099	0.153
32	chairs	28	12.04	0.173	0.123
33	tables	16	11.88	0.254	0.183
34	articulated-hands	18	11.94	0.226	0.146
35	articulated-eyeglasses	13	12.00	0.161	0.156
36	starfish	19	12.26	0.158	0.102
37	dolphin	23	12.35	0.071	0.053
38	bird	17	12.12	0.239	0.211
39	butterfly	2	12.00	0.166	0.009
Mean			12.38	0.121	0.098

Table 3: Average feature extraction run time per object.

Method	Setup	View rendering	Descriptor construction	Total time
Max distinct views	0.467s	0.05s	0.077s	0.601s
LFD 100 views	0.396s	4.278s	4.567s	9.247s

all 100 views for the LFD method.

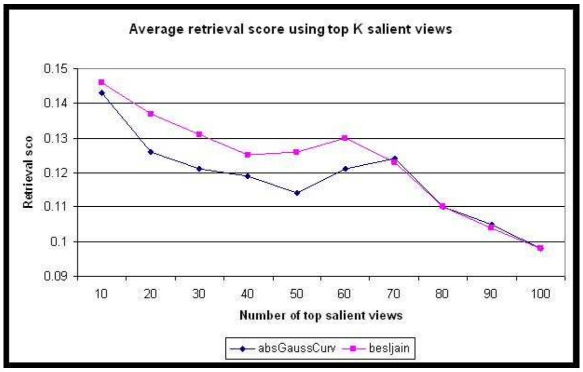


Figure 8: Average retrieval scores across all SHREC classes in the database as the number of top salient views used to construct the descriptor is varied. Learning of the salient points used two different low-level features: absolute Gaussian Curvature and Besl-Jain curvature.

Table 1: Average retrieval scores across all SHREC classes as the number of top salient views and top distinct salient views are varied. Absolute Gaussian curvature was used as the low-level feature in the base framework. The average maximum number of distinct salient views is 12.38, hence there is no score available for K > 13 when using the top K distinct views.

K	score for top K views	score for top K distinct views
1	0.207	0.207
2	0.186	0.174
3	0.172	0.163
4	0.162	0.151
5	0.157	0.138
6	0.155	0.134
7	0.152	0.131
8	0.152	0.129
9	0.146	0.127
10	0.143	0.128
11	0.137	0.127
12	0.134	0.121
20	0.126	-
30	0.121	-
40	0.119	-
50	0.114	-
60	0.121	-
70	0.124	-
80	0.110	-
90	0.105	-
100	0.098	-

7. CONCLUSION

We have developed a new methodology for view-based 3D object retrieval that uses the concept of salient 2D views to speed up the computation time of the light field descriptor algorithm. Our experimental results show that the use of salient views instead of 100 equally-spaced views can provide similar performance, while rendering many fewer views. Furthermore, using the top K distinct salient views performs much better than just the top K salient views. Retrieval scores using the maximum number of distinct views for each object are compared to LFD and differences in retrieval scores are explained. Finally, a timing analysis shows that our method can achieve a 15-fold speedup in feature extraction time over the LFD.

Future work include investigating other methods to obtain the salient views. One way is to generate salient views using a plane fitting method with the objective of fitting as many salient points on the surface of the 3D object. This approach may be more computationally expensive as it may require exhaustive search in finding the best fitting plane, however some optimization method may be used to reduce the search space.

8. ACKNOWLEDGMENTS

This research is supported by the National Science Foundation under grant number DBI-0543631 (PI: L. Shapiro).

9. REFERENCES

[1] T. Ansary, M. Daoudi, and J.-P. Vandeborre. 3d model retrieval based on adaptive views clustering. In *ICAPR LCNS 3687*, pages 473–483, 2005.

[2] T. F. Ansary, J.-P. Vandeborre, and M. Daoudi. On 3d retrieval from photos. In *3DPVT*, 2006.

[3] I. Atmosukarto and L. G. Shapiro. A learning approach to 3d object representation for classification. In *International Workshop on Statistical + Structural and Syntactic Pattern Recognition*, 2008.

[4] I. Atmosukarto and L. G. Shapiro. A salient-point signature for 3d object retrieval. In *ACM Multimedia Information Retrieval*, 2008.

[5] P. J. Besl and R. C. Jain. Three-dimensional object recognition. *Computing Surveys*, 17(1), 1985.

[6] D. Chen, X. Tian, Y. Shen, and M. Ouhyoung. On visual similarity based 3d model retrieval. *Computer Graphics Forum*, 22(3), 2003.

[7] C. Cyr and B. Kimia. 3d object recognition using shape similarity-based aspect graph. In *ICC*, 2001.

[8] 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, 2005.

[9] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Trans. Graph.*, 24(3):659–666, 2005.

[10] Z. Liu, J. Mitani, Y. Fukui, and S. Nishihara. Multiresolution wavelet analysis of shape orientation for 3d shape retrieval. In *ACM Multimedia Information Retrieval*, 2008.

[11] H. Müller, S. Marchand-Maillet, and T. Pun. The truth about corel - evaluation in image retrieval. In

CIVR, 2002.

[12] R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3d model retrieval. In *SMI*, 2008.

[13] Z. Qin, J. Jia, and J. Qin. Content based 3d model retrieval: A survey. In *International Workshop CBMI*, pages 249–256, 2008.

[14] SHREC. http://www.aimatshape.net/event/shrec, 2006.

[15] J. Tangelder and R. Veltkamp. A survey of content based 3d shape retrieval methods. *Multimedia Tools Application*, 2007.

[16] Y. Wang, R. Liu, T. Baba, Y. Uehara, D. Masumoto, and S. Nagata. An images-based 3d model retrieval approach. In *Multimedia Modeling*, 2008.

[17] H. Yamauchi, W. Saleem, S. Yoshizawa, Z. Karni, A. Belyaev, and H.-P. Seidel. Towards stable and salient multi-view representation of 3d shapes. In *Shape Modeling International*, 2006.