

# A Learning Approach to 3D Object Representation for Classification

Indriyati Atmosukarto and Linda G. Shapiro

University of Washington,  
Department of Computer Science and Engineering, Seattle, WA, USA  
{indria, shapiro}@cs.washington.edu

**Abstract.** In this paper we describe our 3D object signature for 3D object classification. The signature is based on a learning approach that finds salient points on a 3D object and represent these points in a 2D spatial map based on a longitude-latitude transformation. Experimental results show high classification rates on both pose-normalized and rotated objects and include a study on classification accuracy as a function of number of rotations in the training set.

**Key words:** 3D Object Classification, 3D Object Signature

## 1 Introduction

Advancement in technology for digital acquisition of 3D models has led to an increase in the number of 3D objects available in specific application databases and across the World Wide Web. This has motivated recent research in two major areas. The first research area is 3D object retrieval where the goal is to retrieve 3D objects from a database that are similar in shape to a given 3D object query. The second research area is 3D object classification where previously unseen objects are assigned to one of a set of previously defined object classes. Classification helps in labeling new unseen objects and can be applied to a number of fields including engineering, architecture, and medical field. The two research areas are complementary as 3D object classification helps in organizing and retrieving objects from a database. Classifying a query object before retrieving similar objects from the database helps to speed up and potentially improve the retrieval process.

Classification of 3D objects requires them to be represented in a way that captures the global and local characteristics of the object. This is achieved by creating a 3D descriptor or signature that summarizes the important shape properties of the object that can be used in classification. In this paper, we describe our method of representing a 3D object and using the representation for 3D object classification. Our 3D object signature is based on a learning approach that finds the salient points on a 3D object and represent these points in a 2D spatial map based on longitude-latitude transformation. A classifier is trained on the signatures of the objects for a number of classes and can then be used to

classify new unseen objects. Motivated by our work on medical applications [2], we wanted to explore the performance of the signatures in classifying objects that have a similar global shape. For the experiments in this paper, we chose to classify head shapes including human, dog, cat, rabbit, horse, wild cats, and bear heads.

The rest of the paper is organized as follows. We first discuss existing classification methods and existing 3D object signatures. We then describe our data acquisition process. Next we describe our method for learning the salient points of a 3D object and then describe the longitude-latitude map signature that captures the patterns of the salient points. Finally, we explain and analyze the results from our experiments and provide a summary and suggestions for future work.

## 2 Related Work

Biasotti et al. [3] presented a comparative evaluation study for 3D shape classification. In their work, they compared the performance of five similarity measures on four different shape descriptors in classifying 3D objects, and concluded that using the nearest neighbor similarity measure performed the best for all four different shape descriptors. Huber et al. [4] presented a parts-based method for 3D object classification. Their method extracted parts from an object and learned part classes and a mapping from the part classes to the object classes using probability.

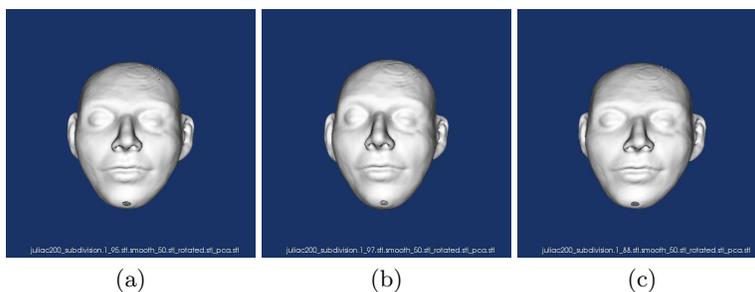
Lee et al. [9] used the curvature information at every mesh vertices to find the salient points. Their method then computed the difference between mean curvature at different scales. The final saliency value was the aggregate of the saliency at all scales with a non-linear normalization. Ohbuchi et al. [11] rendered multiple views of a 3D model and extracted local features from each view image using the SIFT algorithm. The local features are then integrated into a histogram using the bag-of-features approach. Novatnack et al. [10] derived corner and edge detectors to extract salient points at each scale.

Assfalg et al. [1] captured the shape of a 3D object using the curvature map of the object's surface. The model surface is first warped into a sphere, and the curvature information is then mapped onto a 2D image using the Archimedes projection. Retrieval of 3D objects is performed by comparing the 2D map of the query object against the 2D maps of the database models using a histogram-based search as one method and a weighted walk-through of the map regions as another. Our method differs from theirs in that it does not use curvature directly, but instead uses a classifier to find salient points and labels them according to the classifier prediction score. It then uses the longitude and latitude position of the salient points on the object's surface to create a 2D map signature. A second classifier is trained on these 2D map signatures for object classification.

### 3 Data Acquisition

The 3D objects used in our experiments were obtained by scanning hand-made clay toys using a Roldand-LPX250 laser scanner with a maximal scanning resolution of 0.008 inches for plane scanning mode [6]. Raw data from the scanner consisted of clouds of 3D points that were further processed to obtain smooth and uniformly sampled triangular meshes of 0.9-1.0 mm resolution.

To increase the number of objects for training and testing our method, new objects were created by deforming the original scanned 3D models in a controlled fashion using 3D Studio Max [5]. Global deformations of the models were generated using morphing operators such as tapering, twisting, bending, stretching and squeezing. The parameters of each of the operators were randomly chosen from given ranges that were determined empirically. Each deformed model included at least five different deformations applied in a random sequence. Figure 1 shows an example of an original scanned human head model and two different morphed versions. The first morphed model was generated by squeezing the original model resulting in a slimmer head, while the second morphed model was generated by twisting the original model resulting in a slightly asymmetrical face.

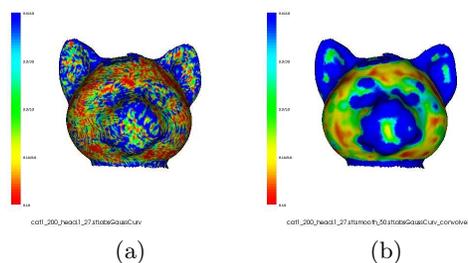


**Fig. 1.** (a) Original scanned human head model and (b,c) two different morphed versions of the model.

Fifteen objects were originally scanned to create a 7-class database. In this work we focus on heads of different classes and shapes to emphasize the repeatability of the method. The seven classes are: cat head, dog head, human head, rabbit head, horse head, tiger head and bear head. Each of the fifteen original objects were randomly morphed to increase the database size. A total of 250 morphed models per original object were created. For this work, we used 75 morphed models from each of the original objects for training and testing the classifier. Points on the morphed models are in full correspondence with the original models from which they were constructed.

## 4 Learning Salient Points

Our methodology starts by applying a low-level operator to every point on the surface mesh. The low-level operators extract local properties of the surface points by computing a single feature value  $v_i$  for every point  $p_i$  on the mesh surface. Five different low-level operators were implemented to test the algorithm’s performance. The five low-level features are Gaussian curvature, mean curvature, shape index, normal magnitude, and curvedness. In this work, we will use the absolute values of the Gaussian curvature for our experiments. The low-level feature values are convolved with a Gaussian filter to reduce the noise. Figure 2a shows an example of the absolute Gaussian curvature values of a 3D model, and Figure 2b shows the results of applying a Gaussian filter over the low-level values.

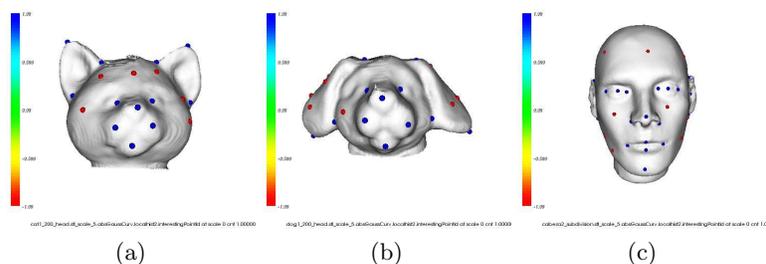


**Fig. 2.** (a) Absolute Gaussian curvature low-level feature value and (b) Smooth low-level feature values after convolution with the Gaussian filter.

After this first step, every point  $p_i$  on the surface mesh will have a low-level feature value  $v_i$ . The second step performs mid-level feature aggregation to compute an aggregate vector for a given neighborhood of every point  $p_i$  on the surface mesh. In this work, we use local histograms to aggregate the low-level feature values of each point. The histograms are computed by taking a neighborhood around each point and accumulating the low-level features in that neighborhood. The size of the neighborhood is determined by multiplying a constant  $c$ ,  $0 < c < 1$ , with the diagonal of the object’s bounding box. This ensures that the size of the neighborhood is scaled according to the object’s size and that the results are comparable across objects. The aggregation results in a  $d$ -dimensional vector  $f_i$  for every point  $p_i$  on the surface mesh where  $d$  is the number of histogram bins. For our experiments, we used  $d = 250$  and  $c = 0.05$ .

Because we were not satisfied with the salient points computed by standard interest operators (eg. Kadir’s entropy operator [7] or Lowe’s SIFT operator [8] applied to 3D), we chose to teach a classifier the characteristics of points that we regard as salient. Histograms of low-level features are used to train a Support Vector Machine (SVM) classifier [12] [13] to learn the salient points on the 3D surface mesh. We used the SVM implemented in WEKA for our experiments [14]. The training data for supervised learning for the classifier are

obtained by manually marking salient and non-salient points on the surface of each training object. An average of 12 salient points and 12 non-salient points were marked on the training objects. Since our current database contains head shapes (human heads, wildcat heads, bear heads, etc.), salient points that were chosen included the tip of the nose, corner of the eyes, corner and midpoints of the lips, etc. The histogram of low-level features of each of the marked points are saved and used for the training. Figure 3 shows examples of manually marked salient and non-salient points on the training data. Since the morphed models were in total correspondence with the original model, we only had to manually mark the original 15 models.

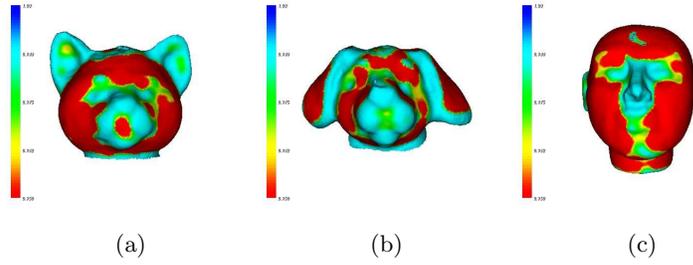


**Fig. 3.** Examples of manually marked salient (blue color) and non-salient (red color) points on (a) cat head model, (b) dog head model, and (c) human head model.

A small training set was used to train the classifier to learn the salient points. The training data for the classifier consisted of the histogram of low-level features for the manually marked salient and non-salient points of 25 instances of the cat head model, 25 instances of the dog head model and 50 instances of the human head models. The classifier was trained with 10-fold cross validation. The classifier does not learn the exact point positions, but instead learns the characteristics of the salient training points in terms of the histograms of their low-level features.

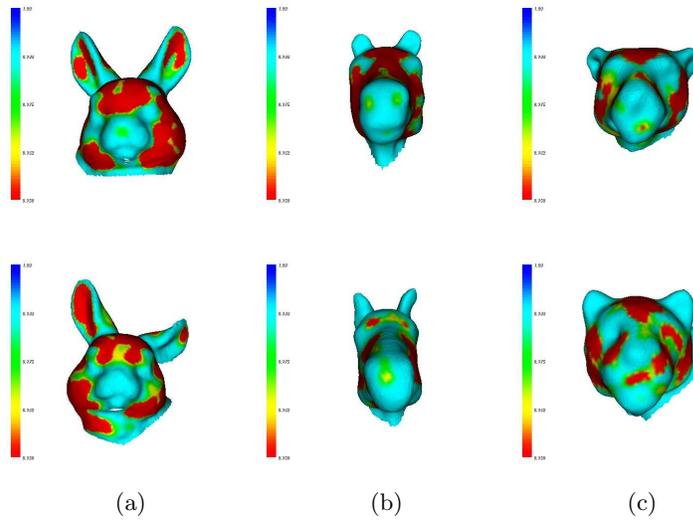
To test the classifier, the histogram of low-level features of all the points of an object are given as its input. The classifier labels each of the points of the 3D object as either salient or non-salient and provides a confidence score for its decision. A threshold  $T$  is applied to the confidence scores for the salient points. In our experiments, we used  $T = 0.95$  to keep only the salient points with high confidence scores from the classifier. Figure 4 shows results of the salient points predicted on instances of the cat, dog and human head class. The salient points are colored according to the classifier confidence score assigned to the point. Non-salient points are colored in red, while salient points are colored in different shades of blue with dark blue having the highest prediction score.

The salient point prediction model was also tested using object classes that were not included in the training phase. Figure 5 shows the predicted salient points on these new object classes. The results show that the salient points



**Fig. 4.** Salient point prediction for (a) cat head class, (b) dog head class, and (c) human head class. The three different columns of models shows the repeatability of the salient point prediction for models of the same 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 score.

predicted are repeatable across objects of the same class, and across objects of different classes.

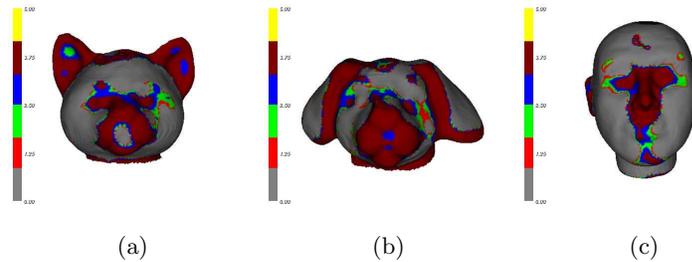


**Fig. 5.** Salient point prediction for (a) rabbit head class, (b) horse head class, and (c) leopard head class. Even though all three classes were not included in the training, the training model was able to predict repeatable salient points across the classes.

## 5 2D Longitude-Latitude Map Signature

Most 3D object classification methods require the use of a 3D descriptor or signature to describe the shape and properties of the 3D objects. Our signature is based on the salient point patterns of the 3D object mapped onto a 2D plane via a longitude-latitude transformation.

Before mapping the salient point patterns onto the 2D plane, the salient points are assigned a label according to the classifier confidence score assigned to the point. The classifier confidence score range is discretized into a number of bins. For our experiments, at confidence level 0.95 and above, we chose to discretize the confidence score range into 5 bins. Each salient point on the 3D mesh is assigned a label based on the bin into which its confidence score falls. Figure 6 shows the discretized salient point patterns on the 3D objects of Figure 4.



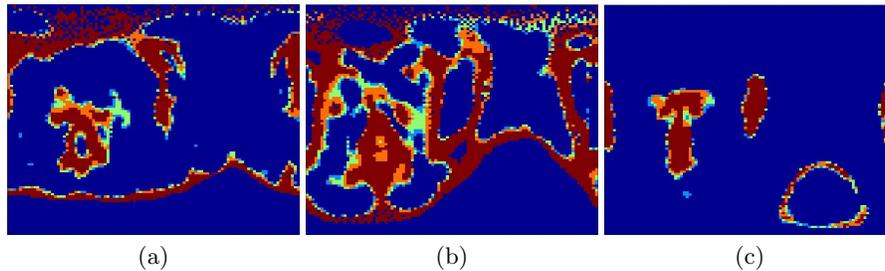
**Fig. 6.** Salient point patterns on 3D objects of Figure 4. Salient points with confidence 0.95–0.96: red, 0.96–0.97: green, 0.97–0.98: blue, 0.98–0.99: dark red and 0.99–1.0: yellow. Gray points are non-salient points.

To obtain the 2D longitude-latitude map signature for an object, we calculate the longitude and latitude positions of all the 3D points on the object’s surface. Given point  $p_i(p_{ix}, p_{iy}, p_{iz})$ , the longitude position  $\theta_i$  and latitude position  $\phi_i$  of point  $p_i$  are calculated as follows:

$$\theta_i = \arctan\left(\frac{p_{iz}}{p_{ix}}\right) \quad \phi_i = \arctan\left(\frac{p_{iy}}{\sqrt{(p_{ix}^2 + p_{iz}^2)}}\right)$$

where  $\theta_i = [-\pi, \pi]$  and  $\phi_i = [-\frac{\pi}{2}, \frac{\pi}{2}]$

A 2D map of the longitude and latitude positions of all the points on the object’s surface is created by binning the longitude and latitude values of the points into a fixed number of bins. A bin is labeled with the salient point label of the points that fall into that bin. If more than one label is mapped to a bin, the label with the highest count is used to label the bin. Figure 7 shows the 2D map signature for the cat head, dog head, and human head model of Figure 6.



**Fig. 7.** 2D longitude-latitude map signatures for (a) cat head, (b) dog head, and (c) human head from Figure 6. All three objects were registered to face forward.

## 6 3D Object Classification

By creating a signature for each of the 3D objects, we are able to perform classification for all the objects in the database. Our experiments fall into two categories: pose normalized and rotation invariant.

### 6.1 Pose Normalized

For the first part of the classification experiments, all objects in the database were rotated to face forward and have the same orientation. Classification of the 3D objects in the database was performed by training a SVM classifier on the salient point patterns of each class using the 2D longitude-latitude signature of the objects in the class. We trained the classifier with the signatures of 25 objects from each class for all seven classes in the database and tested the classifier model with a new training set consisting of 50 objects per class for each of the seven classes. The classifier achieved 100% accuracy in classifying all the pose-normalized objects in the database.

### 6.2 Rotation Invariant

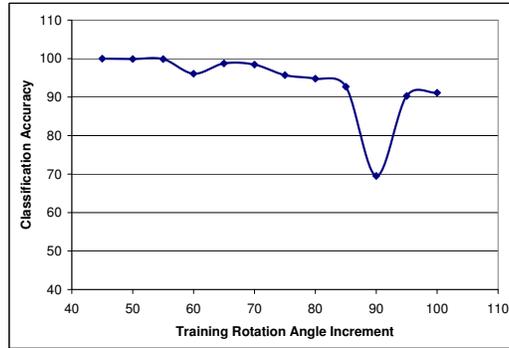
Since 3D objects may be encountered in the world in any orientation, rotation invariant classification is desirable. To achieve rotation invariance for classification, we trained the classifier with a number of rotated versions of the 2D map longitude-latitude signature for each training object. Due to WEKA space and memory constraints, we reduced the database from 7 classes to 3 classes for these tests: human head, wild cat head and bear head, and selected 5 objects per class as the base training set.

For the rotation invariance study, we ran three sets of experiments. We first tested the classification accuracy by training a classifier with rotated versions of the training data signatures at 45 degree increments for all three axes. This resulted in  $8 \times 8 \times 8$  rotated signatures for each object in the database. The classifier was tested on four new objects per class. Rotated versions of the testing data signatures were generated using the same rotation degree increments. The

classifier achieved 100% classification accuracy when classifying objects that were rotated in the same increments as the training data.

In the second experiment, the classification method was tested on 15 new testing instances per class that were rotated randomly. For example, a rotation of  $(250,4,187)$  was one of the random rotations that did not match any of the training rotations. The classifier was still able to achieve 100% classification accuracy.

Our third set of experiments was to explore the degradation in the classification accuracy by varying the training rotation angle increment when generating the signatures for the training data. Figure 8 shows the degradation in the classification accuracy as the training angle increment increases and the number of rotated training signature instances decreases. The graph shows that the classification accuracy steadily decreases as the number of rotated training signature decreases. In addition, there is a big dip in the classification performance when the training signatures are generated at 90 degree angle increments. This is because the signatures produced at 90 degree increments are not representative of angles in between the multiples of 90 degrees. Note that the classifier is still able to achieve 91% classification accuracy with training signatures generated at 100 degree increments with only  $3 \times 3 \times 3 = 27$  rotated training signatures per training object, which is much better than the  $8 \times 8 \times 8 = 512$  signatures that we started with.



**Fig. 8.** Classification accuracy vs training rotation angle increment.

## 7 Conclusions

We have described a methodology that uses a learning approach to identify salient points on 3D objects. The patterns of these salient points are then used

to train a classifier for object class recognition by transforming the patterns onto a 2D map using longitude-latitude transformation. The first set of experiment results show that the classifier is able to achieve high accuracy when classifying objects that are pose normalized and are of the same orientation. We extended the approach to achieve rotation invariance by generating a number of rotated version of each of the training object signatures and trained a classifier on these rotated signatures to learn possible rotation combination and their respective maps. Experimental results show that the classifier was able to achieve 100% classification accuracy when trained on rotated signatures generated at 45 degree angle increments and can still achieve 91% classification accuracy when trained on rotated signatures generated at 100 degree angle increments.

We are investigating clustering the rotated training signatures to further reduce space usage and computation time so that we will be able to include many more models in the database. By clustering the training signatures, we will be able to reduce the training data by using only the cluster centroids. In addition, we are also working on finding a mapping function between two signatures. This will eliminate the need to generate rotated version of the signatures and will greatly improve space and computation efficiency.

## References

1. J. Assfalg, A. D. Bimbo, and P. Pala. Content-based retrieval of 3d models through curvature maps: a cbr approach exploiting media conversion. *Multimedia Tools Appl.* 31:29–50 (2006)
2. I. Atmosukarto, R. Travillian, J. Franklin, L. Shapiro, J. Brinkley, and D. Suci. A unifying framework for combining content-based image retrieval with relational database queries for biomedical applications, SIIM (2008)
3. S. Biasotti, D. Giorgi, S. Marini, M. Spagnuolo, and B. Facidieno. A Comparison Framework for 3D Object Classification Methods. *MRCS*, 314–321 (2006).
4. D. Huber, A. Kapuria, R. R. Donamukkala, and M. Hebert. Parts-based 3D object classification. In *Proc. IEEE CVPR* (2004).
5. Autodesk 3d studio max, <http://usa.autodesk.com>
6. S. R. Correa, L. Shapiro, M. Meila, G. Berson, M. Cunnigham, and M. Sze. Symbolic Signatures for Deformable Shapes. *IEEE Trans. PAMI*, 28(1):75–90 (2004)
7. T. Kadir and M. Brady. Scale, Saliency and Image Description. *IJCV*, 45(2):83-105 (2001)
8. D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91-110 (2004)
9. C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. *ACM Trans. Graph*, 24(3):659–666 (2005)
10. J. Novatnack and K. Nishino. Scale-dependent 3d geometric features. In *Proc. ICCV* (2007)
11. R. Ohbuchi, K. Osada, T. Furuya, and T. Banno. Salient local visual features for shape-based 3d model retrieval. In *Proc. IEEE SMI* (2008)
12. B. Scholkopf and A. J. Smola. *Learning with kernels*. Cambridge Uni. Press (2002)
13. V. V. Vapnik. *Statistical Learning Theory*. John Wiley and Sons (1998)
14. I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann San Fransisco, 2nd edition (2005)