Learning 3D Part Detection from Sparsely Labeled Data: Supplemental Material

Ameesh Makadia
Google
New York, NY 10011
makadia@google.com

Mehmet Ersin Yumer
Carnegie Mellon University
Pittsburgh, PA 15213
meyumer@cmu.edu

The content that follows was left out of the main paper due to space limitations.

1. Descriptors

In this section are details of the shape and appearance descriptors as described in section 2.1 of the main submission.

Spin Images. Spin Images [9] parameterize surface shape about a reference point into the radial distance to the reference point's surface normal line, and the signed distance to the points tangent plane. A *spin image* is a 2D histogram in this parameter space, and an 8x8 histogram is used in the experiments (the final descriptor representation is a concatenation of each histogram bin into a 64 dimensional vector).

Shape Contexts. Shape Contexts were first developed for contour matching in 2D [1]. For any reference point, every other contour point is represented by its distance and relative orientation to the reference point. A 2D histogram in relative angle and *log*-distance make up the shape context descriptor. A 5x6 histogram (producing a 30 dimensional feature vector) is used for the experiments.

Curvature. For any surface point, principal curvature (k_1 and k_2) is measured using the methods of [7, 6]. A 5 dimensional feature vector is constructed from the following: k_1, k_2, $k_1 * k_2$, $\frac{k_1+k_2}{2}$, $k_1 - k_2$.

PCA. Following [10], the singular values σ_1, σ_2, σ_3 are computed from the covariance matrix of local surface vertices. A 12 dimensional feature descriptor is constructed from the following: $\frac{\sigma_1}{\sigma}$, $\frac{\sigma_2}{\sigma}$, $\frac{\sigma_3}{\sigma}$, $\frac{\sigma_1+\sigma_2}{\sigma}$, $\frac{\sigma_1+\sigma_3}{\sigma}$, $\frac{\sigma_2+\sigma_3}{\sigma}$, $\frac{\sigma_1}{\sigma_2}$, $\frac{\sigma_1}{\sigma_3}$, $\frac{\sigma_2}{\sigma_3}$, $\frac{\sigma_1}{\sigma_2} + \frac{\sigma_1}{\sigma_3}$, $\frac{\sigma_1}{\sigma_2} + \frac{\sigma_2}{\sigma_3}$, $\frac{\sigma_1}{\sigma_3} + \frac{\sigma_2}{\sigma_3}$, where $\sigma = \sigma_1 + \sigma_2 + \sigma_3$.

Average Geodesic Distance (AGD). Following [10], The "isolation" of a vertex is measured by averaging the geodesic distance to all other vertices. An 11-dimensional descriptor is generated from the mean geodesic distance, the squared mean, and the $\{10, 20, \ldots, 90\}$-th percentile of geodesic distances.

Texture. To extract features from the model's texture map at a reference point, the model is orthographically projected along the point's surface normal into an image plane [15]. By rendering the model in this way the model appearance can be processed with traditional feature descriptors for images. Our descriptor is the 128-dimensional Sift [12] descriptor which constructs an orientation histogram of image gradients in the neighborhood of a reference point. Orientation invariance is achieved by rotating the neighborhood patch to align its dominant orientation to a canonical direction.

Scale selection. For 3D shape descriptors, the typical approach to handling scale variance is to extract local descriptors at multiple scales, where the reference size is determined by some global scale characteristic (e.g. diagonal length of the model bounding box, radius of the bounding sphere, median of all-pairs geodesic distances, etc), for example see [11, 10]. In the experiments r is set to be the 30th-percentile of all-pairs geodesic distances. For any scale s, the feature descriptor computed will depend on only the vertices which are within geodesic distance s to the reference point (for the texture feature, the scale s determines the radius of the neighborhood patch in the rendered image). For Sift descriptors, features are extracted at two scales $s \in \{0.05r, 0.10r\}$. For PCA, features are extracted at 5 scales ($s \in \{0.05r, 0.10r, 0.20r, 0.30r, 0.50r\}$), and for Shape Context and Spin Images features are extracted at a single scale $s = r$. Scale selection is not applicable for the curvature descriptor since the approximation utilized is fixed to use only the one-ring of the reference vertex.

The total dimensionality of each feature type (accounting for all scales) is: Spin Images $\in \mathbb{R}^{64}$, Shape Contexts $\in \mathbb{R}^{30}$, Curvature $\in \mathbb{R}^{5}$, PCA $\in \mathbb{R}^{60}$, AGD$\in \mathbb{R}^{11}$, and Sift $\in \mathbb{R}^{256}$. To generate a single descriptor for any vertex, the individual descriptors are $L1$-normalized before being concatenated into a single 426-dimensional descriptor. In all feature computations, geodesic distance is approximated with shortest-path distances.

2. SVM Classifiers

Regarding classification described in section 2.1 of the main submission, we use a common approach for training. We choose to compute c_y as a one-vs-all binary SVM classifier [4] trained over descriptors $\vec{x}$. Specifically we train a polynomial kernel SVM, and c_y is the corresponding decision score of the classifier ($c_y > 0$ is a positive classification for label y). Since we have a large number of negative examples for each class, we employ a data mining approach for hard negative examples as in [8]. SVM parameters are selected through a grid search with 3-fold cross validation. We use the LibSVM [2] implementation in our experiments. Regarding evaluation, since the SVM baseline is not trained to output a sparse labeling we cannot expect it to generate only one detection per object part. Thus, throughout the experimental results we do not penalize the SVM baseline for multiple positive detections of the same object part (*i.e.* for multiple correct detections appearing within the neighborhood of the true location, all but the closest detection are simply ignored).

3. Datasets

3.1. Maybe3D

In section 3 of the main submission we very briefly described two datasets (cameras and video recorders) obtained from maybe3d.com, and examples were given in figure 3 of the main paper. The camera dataset is a collection of 360 models. There are 34 part labels (these parts are listed in the legend of figure 6 in the main paper). Each part appears in 177 models on average, and each model has 20 parts labeled on average (the dataset provided has more than 34 labels but we discard those that don't appear in at least 10 models). The dataset was randomly partitioned into a training (298 models) and test (62 models) set. The triangulated meshes have on average 3237 vertices. Compared to the cameras collection, the video recorders dataset is much smaller, consisting of 64 models (50 train, 14 test), and 12 part labels (these 12 labels are shown in legend of figure 6 in the main paper). Each part appears in 52 models on average, and each model has 10 parts on average. Figure 1 gives an example of the inconsistencies in these datasets that make detection challenging.

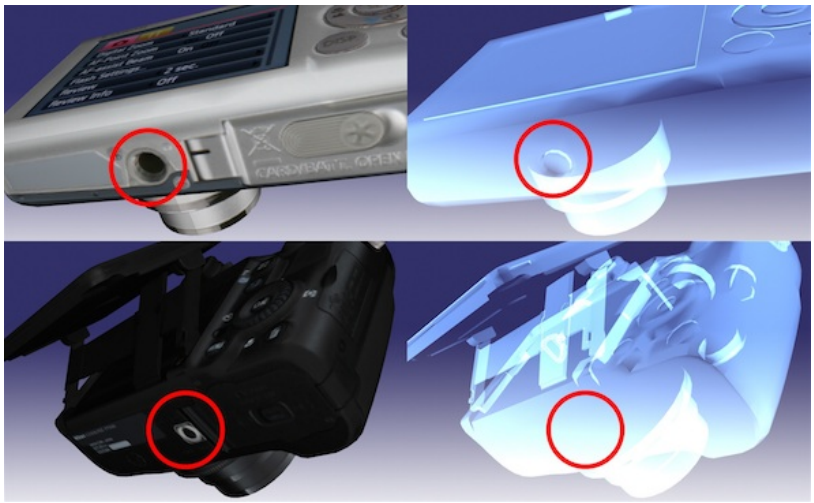


Figure 1. Here we have highlighted the part label *tripod mount* on two models in the dataset. On the top left we see a rendering of a camera model where the tripod mount is highlighted by the red circle. On the top right we see the corresponding surface geometry showing the indentation of the tripod mount. On the bottom we see a model where the tripod mount does not appear in the geometry but only in the texture map. These inconsistencies are typical of this real-world cameras dataset.

3.2. Princeton Segmentation Benchmark

The Princeton Segmentation Benchmark was introduced in [3], which is a collection of human-generated segmentations of 380 models (19 categories, 18 models each). The models were taken from the larger Princeton shape dataset [14]. The segmentations were further processed in [10] to assign labels to segments. From this final dataset we selected 8 categories for evaluation (Airplane, Ant, Armadillo, Bust, Chair, Four Leg, Human, and Vase). For each category, we created a random partition of 14 train and 6 test objects. Figure 2 shows an example model with segmentation from each of the 8 categories.

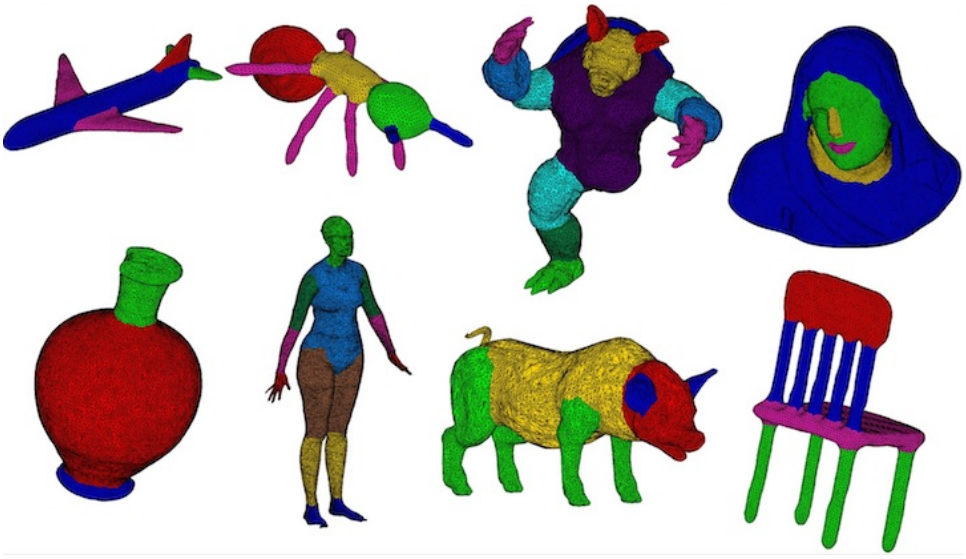
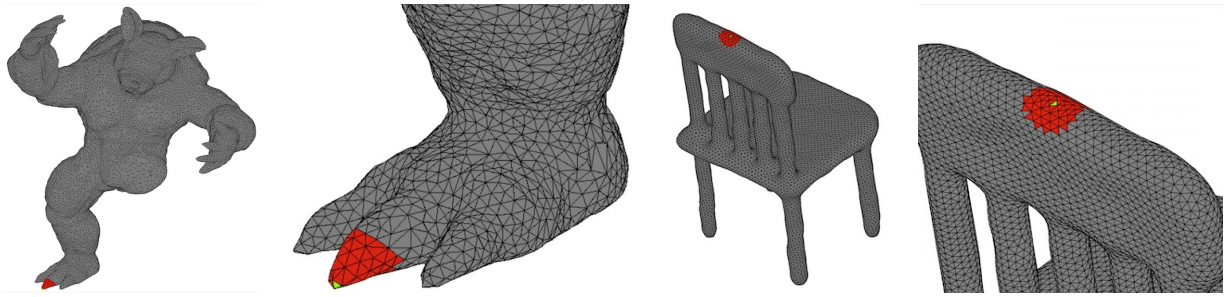


Figure 2. One example from each of the 8 categories of the Princeton segmentation set [3, 10] on which we evaluated our algorithm, colored by their dense labeling. Counter-clockwise from top left: Airplane, Ant, Armadillo, Bust, Chair, Four Leg, Human, and Vase.

In order to simulate a sparse labeling for our experiments we chose a single point within each segment as the sparse annotation. As described in the main submission, we selected points closest to the segment centers (computed as maximum of minimum distances to the segment boundary).



a b c d

Figure 3. Neighborhood sizes. In (a) is an Armadillo model from the Princeton segmentation benchmark. The mesh triangle colored green is the simulated sparse label for the part *foot*. The red triangles denote the triangles which are included in the neighborhood V of the *foot* part. This model has 33,556 mesh triangles. In (b) is a zoomed-in view of (a). Similarly, (c) shows the neighborhood of part *support* of a chair model, and (d) the zoomed-in view. This model has 22,854 mesh triangles. The neighborhood sizes were computed as $0.08m$ where m is the median of all-pairs geodesic distances on the mesh.

At the beginning of section 4 (Experiments) of the main submission we explain that we require a detection to be within a small neighborhood V of the true part location to be considered a correct detection. For the cameras and video recorders we used the median distance between all pairs of adjacent vertices to define the neighborhood radius. This was suitable for the cameras and video recorders which had a consistent mesh resolution of a few thousand triangles. However, this would be too restrictive for models included from the Princeton segmentation set which typically have 20k+ mesh triangles. However, since these models have a dense and roughly uniform triangulation we can use a more robust statistic based on geodesic distances. For these experiments we define the neighborhood V of a vertex as $0.8m$ where m is the median distance between all-pairs geodesic distances on the model. This gives the desirable neighborhood size for evaluation (see figure 3 for two example meshes with different resolutions).

4. Experiments

In the main paper we provided variations of our experimental setup which included **test-vs-train dissimilarity** and **densely labeled models**. Here are some details for each (note, for the **mesh resolution variation** experiments all details are included in the main paper).

4.1. Test-vs-train dissimilarity

We created 8 different test-train partitions of the cameras set based on brand name (*i.e*. where the test and train sets do not share brand names). The top row of figure 5 in the main submission shows the results averaged over all 8 experiments. In table 1 we show the results per brand.

	Canon	Casio	Fujifilm	Nikon	Olympus	**Panasonic**	Pentax	Samsung	**Mean**
SVM	0.18	0.15	0.21	0.19	0.15	**0.19**	0.14	0.16	0.17
$\text{SVM}^c_{I_2}$	0.24	0.19	0.28	0.23	0.20	**0.18**	0.18	0.21	0.22

Table 1. Mean per-label AP for each of the eight brand-specific test-vs-train partitions. The last column corresponds to the results shown in table 5 of the main submission. We can see that the extreme nature of dissimilarity between the test and train sets causes incorrect learning of the spatial layout model in some cases, *e.g*. the model actually hurts performance for the Panasonic brand.

4.2. Densely labeled models

	Airplane	Ant	Armadillo	Bust	Chair	Four Leg	Human	Vase	**Mean**
SVM	0.69	0.65	**0.59**	0.19	**0.58**	**0.46**	0.37	**0.48**	0.51
$\text{SVM}^c_{I_2}$	**0.86**	**0.81**	0.54	**0.36**	0.46	0.39	**0.42**	0.41	**0.54**

Table 2. Mean per-label AP the eight categories from the Princeton segmentation set. This is a breakdown of the composite results show in in table 6 (top row) of the main paper. Since our algorithm is designed to learn a spatial layout model from large collections of sparsely labeled data, this dataset exposes the limitations of our method when learning from limited observations.

We described the densely labeled labeled models from the Princeton segmentation set above. Composite results on 8 categories were reported in table 6 of the main paper (top row). Here in table 2 we provide the per-category results. As we discussed in the main paper, since we are learning a second order spatial layout model it is critical to have enough training observations. Therefore the ability to learn a robust model from a small set of objects (*e.g*. 14 training models per category as we have here) is a limitation of our work.

Note, for the untextured models in the Princeton segmentation set, we omit the Sift features when generating a descriptor ensemble. Also, in the main paper (table 6, second row, last column) we show label accuracy results of the CRF model proposed in [10]. To obtain these results, we follow the algorithm of [10] with two exceptions: (1) we use instead the feature ensemble described in the main paper (minus Sift descriptors), and (2) we omit the multiple steps of retraining and feature stacking and use only a classifier trained directly on the unary shape features. The important aspect of comparison is the CRF layer which we follow closely in our implementation.

References

[1] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *Pattern Anal. Mach. Intell.*, 24(4):509–522, Apr. 2002. 1

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. 2

[3] X. Chen, A. Golovinskiy, and T. Funkhouser. A benchmark for 3D mesh segmentation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 28(3), Aug. 2009. 2, 3

[4] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 2

[5] C. Desai, D. Ramanan, and C. C. Fowlkes. Discriminative models for multi-class object layout. *International journal of computer vision*, 95(1):1–12, 2011.

[6] M. P. do Carmo. *Differential Geometry of Curves and Surfaces*. Prentice-Hall, Englewood Cliffs, NJ, 1976. 1

[7] C.-s. Dong and G.-z. Wang. Curvatures estimation on triangular mesh. *Journal of Zhejiang University Science*, Jan. 2005. 1

[8] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010. 2

[9] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997. 1

[10] E. Kalogerakis, A. Hertzmann, and K. Singh. Learning 3D Mesh Segmentation and Labeling. *ACM Transactions on Graphics*, 29(3), 2010. 1, 2, 3, 4

[11] C. H. Lee, A. Varshney, and D. W. Jacobs. Mesh saliency. In *ACM SIGGRAPH 2005 Papers*, pages 659–666. ACM, 2005. 1

[12] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, Nov. 2004. 1

[13] G. Salton and M. Mcgill. Introduction to modern information retrieval. 1986.

[14] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The Princeton shape benchmark. In *Shape Modeling International*, 2004. 2

[15] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches (vip). *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 1