

Harmonic Shape Histograms for 3D Shape Classification and Retrieval

J. Fehr, H. Burkhardt
 Chair of Pattern Recognition and Image Processing
 Albert-Ludwig-University
 Freiburg, Germany

Abstract

In this paper, we present a novel approach towards 3D shape recognition and retrieval using histograms of rotation invariant local features. Features are extracted for every point of voxelized 3D shape objects by use of functions on spheres which are invariant towards rotation of the object. The fast computation of the local features is performed via convolution methods in frequency space. Histograms of these features describe an object in terms of distributions of local geometric properties such as orientation and angle of edges, distances and convexity. Object classification is performed by Support Vector Machines with histogram-intersection kernels. In experiments on the Princeton Shape Benchmark [1], our approach outperformed many existing methods in several classification and retrieval tasks.

1. Introduction

Retrieval and matching of shape objects is a wide spread problem with many applications in computer vision, CAD, computer graphics and medicine. Many algorithms and various rotation invariant shape representations have been published so far. Matching of two identical (under the group of rotation, translation and scaling) shapes can be achieved with high accuracy for most cases. On the other hand, the more challenging, and practical relevant task of object classification/retrieval of more or less semantic classes with a high intra-class variance is far from being solved. In this paper we argue to focus more on local rather than global features of shapes in order to enable classifiers to handle higher order classification and retrieval tasks as presented in the Princeton Shape Benchmark [1].

We restrict our brief overview of related work to methods for 3D shape classification based on rotation invariant features using functions on spheres, and those which performed best on the PSB. The very basic **Global Shape Histogram (SHAPEHIST)** computes the distance for each point on the shape to all other points on the shape and stores the results in a histogram. **Global Shell Histogram (SECSHELL)**: [2] divide an object in 3D shells and sectors located around the center of mass using Histograms over the subdivisions as global object features. The **Spheri-**

cal Extent Function (EXT) [4] computes global rotational invariant features as absolute values of coefficients of a spherical harmonic approximation of the object. **Radialized Spherical Extent Function (REXT)** [5] extends the (EXT) approach to the spherical harmonic approximation of the object's inner structure on several concentric spheres combined with PCA. The **Spherical Harmonic Descriptor (SHD)** [3] is similar to (REXT). **Light Field Descriptor (LDF)** [7] uses multiple 2D views of 3D shapes. Rotation invariance is achieved by a collection of 100 2D views per object, which are rendered orthogonal to a sphere containing the object.

All of these methods have in common that they try to model an object shape at a global level which has the disadvantage that the assumption that objects of the same class are sharing the same base shape is not always adequate - especially when one considers more semantic groupings with high intra-class variance. The (EXT),(REXT) and (SHD) methods use spherical harmonic representations. In order to gain rotation invariance, only the absolute values of the harmonic coefficients are used for feature computation. This neglects the entire phase information of the complex coefficients, which can cause ambiguous feature representations and weaken separability.

2. Local Features

To overcome the existing drawbacks mentioned in the last section, we propose harmonic shape histograms as local rotation invariant features for object shape recognition. The generation of these features follows three steps: first, the 3D shape model is rendered into a volume V , with voxels $\mathbf{v} \in V$ and $\mathbf{v} = 1$ inside the object contour and $\mathbf{v} = 0$ outside. All objects are rendered to a fixed size, providing scale invariance. Second, voxel-wise features are extracted for the entire volume, and third, stored in a histogram over each feature. For each object, feature-histograms are concatenated to a single 1D histogram.

2.1. 2-Point Features

2-Point Features are simple, yet powerful local shape features. In many ways they could be considered to be the local equivalent to global shape histograms. For each voxel

$\mathbf{v} \in V$ all neighboring voxels with distance r inside the object are counted. This rotation invariant feature $T_r(\mathbf{v})$ is derived by Haar-Integration over the spherical neighborhood parameterized by the angles φ_1, φ_2 and $\mathbf{q}_{\varphi_1, \varphi_2}$ with $\|\mathbf{q}\| = r$.

$$T_r(\mathbf{v}) := \int_0^\pi \int_\pi^{-\pi} V(\mathbf{v} + \mathbf{q}_{\varphi_1, \varphi_2}) d\varphi_1 d\varphi_2 \quad (1)$$

This can be directly implemented via fast convolution with the surface of a sphere S_r with radius r (Fig. 1):

$$T_r(\mathbf{v}) := (V * S_r)(\mathbf{v}), \quad \text{with } S_r(\mathbf{v}) := \delta(\|\mathbf{v}\| - r) \quad (2)$$

The name "2-Point" derives from the property that the fea-

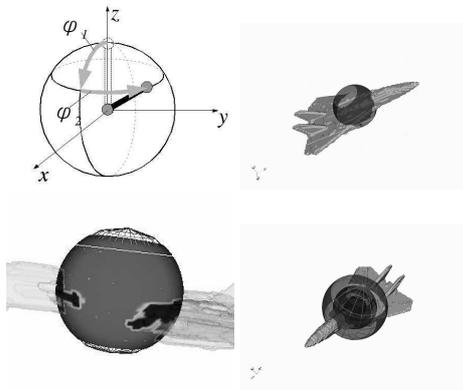


Figure 1: 2-Point features: top,left: parameterization on the sphere; right: convolution sphere for one voxel, bottom, left: integration via convolution, the black areas indicate voxels which are integrated. right: multiple spheres for the histogram representation of one voxel.

ture encodes the pairwise relations of \mathbf{v} and its neighbors with distance r . This approach already leads to quite reasonable results (see Fig. 2), which can be notably improved by two minor extensions: first, so far we only consider voxels inside the shape - by applying a gauss filter \mathcal{G} to V before feature extraction, voxels close to the shape contribute to the local histograms depending on their gaussian weight. Second, in order to gain more control over the local weighting, we introduce two arbitrary non-linear functions f_a and f_b , which are applied to \mathbf{v} and V :

$$T_{r, f_a, f_b}(\mathbf{v}) := f_a(\mathbf{v}) \cdot (f_b(\mathcal{G}(V)) * S_r)(\mathbf{v}) \quad (3)$$

Suitable choices for $f_a(x)$ and $f_b(x)$ were empirically found to be simple mappings like x^2, x^3 or \sqrt{x} .

2.2. 3-Point Features

The 2-Point features provide useful local distance distributions and along with that some implicit local shape prop-

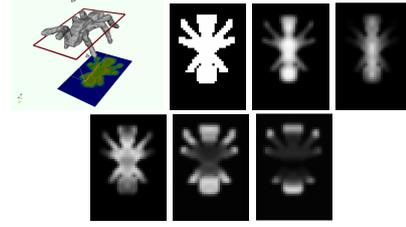


Figure 2: Example results of 2p features in a 2D xy-slice. Top: slice scheme, original data, followed by results of features with growing radii.

erties like thickness or convexity. But the local discrimination power is quite limited. For example, it is not possible to explicitly distinguish between edges and corners. Since we are especially interested in such "higher order" properties, we extend the original 2-point approach to Haar-Integrals over the degrees of freedom of the relation of three points. This will increase the local discrimination power to the needed extend. Going from two to three points appears to be straight forward, but in fact this turns out to be quite challenging. This is due the fact, that we now have to integrate over three degrees of freedom ($\varphi_1, \varphi_2, \varphi_3$) in order to cover all possible constellations of three points ($\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$) with radii r_1 and r_2 (fig. 3). With this third degree of

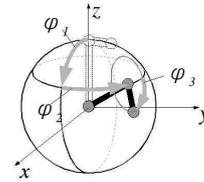


Figure 3: Parametrization of the 3-Point feature, for visual simplification the special case of $r_2 = r_1$ is displayed.

freedom, our fast convolution algorithm can not be applied anymore, but a direct computation would be too expensive. To overcome this, we first consider that for a given center voxel \mathbf{v}_0 , all voxels which could possibly contribute to the Haar-Integral, lie on the surfaces of two concentric spheres with radii r_1 and r_2 . We denote these spheres D_{r_1} and D_{r_2} , which are carrying the original data on the surface. Now we expand these spheres in terms of spherical harmonics, which provide an orthogonal basis for functions on the 2-Sphere analog to the Fourier transform. So every spherical function on a sphere can be represented by the sum of its harmonics:

$$f(\theta, \phi) = \sum_{l=0}^{\infty} \sum_{m=0}^l a_{lm} Y_l^m(\theta, \phi) \quad (4)$$

where l denotes the band of expansion, m the components for the l -th band and a_{lm} the harmonic coefficient. The

transformation to harmonic representation of D_{r_1} and D_{r_2} can be computed efficiently via the dot-product :

$$\widehat{D}_r(l, m) = Y_l^m \cdot D_r \quad (5)$$

The harmonic representation now enables us to compute the integral over all degrees of freedom very efficiently: First, we consider rotation around φ_3 : All voxels contributing to the integration over this angle, lie on a circle C on the outer sphere D_{r_2} with its center on the extension of \mathbf{v}_1 and radius r_c (fig. 3). Due to the spherical harmonic addition theorem, we can rotate the entire expansion to center C at the "north pole". Now Haar-Integration over C on the sphere D_{r_2} can be computed via left-convolution of D_{r_2} with C . We define the left-convolution in the harmonic domain as $D_{r_2} \otimes C$:

$$\widehat{D}_{r_2} \otimes (l, m) = 2\pi \sqrt{\frac{4\pi}{2l+1}} \widehat{D}_{r_2}(l, m) \cdot \widehat{C}(l, 0) \quad (6)$$

This has the nice property, that since for our case C is symmetric and reduced to a "latitude" of D_{r_2} , all harmonic coefficients $\widehat{C}(l, m)$ are zero for $m \neq 0$ and equal to the associated Legendre polynomial $P_l^0(\cos \alpha)$ for $m = 0$. This way the left-convolution in the harmonic domain is reduced to a simple point-wise multiplication with a scalar value for each band of expansion. The other two degrees of rotational freedom, ϕ_1 and ϕ_2 , are then eliminated by the computation of the dot-product between the harmonic representations of the spheres D_{r_1} and D_{r_2} . This leads to the following final formulation for three points case analog to the 2-Point case:

$$T(\mathbf{v}) := f_a(v) \cdot \sum_l \sum_m \widehat{f_b(D_{r_1})}(l, m) \cdot (\widehat{f_c(D_{r_2})} \otimes C)(l, m) \quad (7)$$

A gaussian filter \mathcal{G} can be inserted in this formulation as in the 2-Point case. Figure (4) shows a schematic overview of the computation procedure, illustrating that the local 3-Point features can entirely be computed via fast global convolution operations. Note, that for our features rotation invariance is achieved without totally neglecting the phase information, which is here implicitly preserved. The invariance of our features is independent of the maximum band of the harmonic expansion - it can be chosen depending on the desired level of detail to be detected by the features.

3. Training and Classification

We use Support Vector Machines (SVM) [6] for training and classification. During training procedure, we compute a large amount of 2 and 3-Point features with various combinations from parameter space $P = \{f_a, f_b, f_c, r_1, r_2, r_3, \mathcal{G}\}$. We apply a Maximum Marginal Diversity algorithm [9] to select the most discriminating features. Then, we compute gray-scale histograms of the derived feature responses and combine these to one single feature vector - the **Harmonic Shape Histogram**.

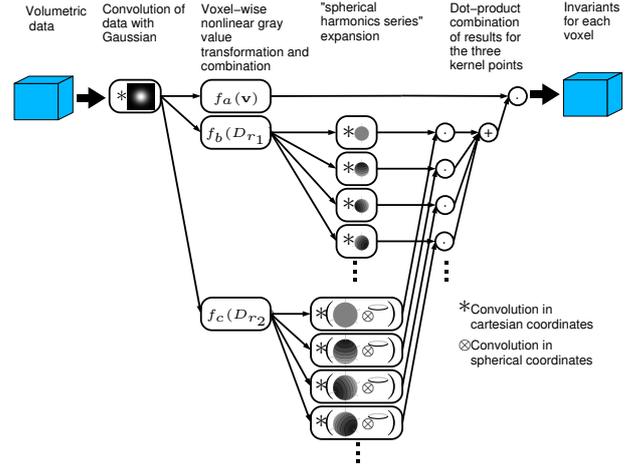


Figure 4: Schema of the 3-Point feature computation.

SVM kernel parameters are determined via gridsearch over the parameters using a "leave one out" evaluation procedure. All feature settings and parameters determined on the training-set are then applied for classification or retrieval tasks on the test-set. Due to the histogram representation of the features, the usage of a Histogram-Intersection kernel implying a \mathcal{L}_1 -norm distance measure is straight forward and empirically delivered the best results. Before kernel evaluation, the feature histograms are normalized to minimum and maximum values.

4. Experiments

Our approach was evaluated on the "The Princeton Shape Benchmark" (PSB) [1]. The PSB contains 1814 3D shape models collected from various Internet sources, including very different object types from fighter jets to chairs or human models. The objects are grouped to classes in four different levels of semantic abstraction: *base*, *coarse1*, *coarse2* and *coarse3*. The distinguished classes with low intra-class variance at the *base* level are combined to more and more semantic classes at following levels. The results published for the PSB (see Table 2) were presented mainly as retrieval results. Direct classification results are only known for the *base* level provided by a $k = 1$ nearest-neighbor classifier (NN) (see Table 2). We performed three different experiments: classification with SVMs, retrieval with SVMs and retrieval using the NN framework provided by the PSB. For the classification task (results see Fig. 1), the SVM was trained on the training set and then applied to the test set. A rejection class was used to handle classes in the test set which are not part of the training set. In order to produce somewhat comparable results in the SVM retrieval task, we used the training-sets only for feature- and model selection as described in section 3. Then, we per-

Features	Base	Coarse1	Coarse2	Coarse3
HSH_2	73.78%	61.43%	63.07%	88.42%
HSH_{23}	78.66%	51.4%	63.51%	86.99%
$HSH_{23}+SHD$	80.27%	50.33%	65.6%	87.85%

Table 1: Results of our features with SVM classification on the PSB test set.

Features	Base level nearest neighbor
LFD	65.7%
$HSH_{23}+SHD$	63.52%
REXT	60.2%
HSH_{23}	59.8%
SHD	55.6%
EXT	54.9%
SECSHELL	54.6%
SHAPEHIST	22.7%

Table 2: Reference results of nearest neighbor classification on the PSB. Taken from [1].

formed a leave-one-out procedure on the test-set. These results (see Table 3) can be compared to those in [1], since this way the same information input is used to estimate performance. Of course, comparing the results of a SVM with a nearest-neighbor classifier (NN) might influence the interpretation of feature performance - so in the last experiment, we provide some NN results of our features (see Table 2). We used the Histogram-Intersection kernel as distance measure for the nearest-neighbor classification. For the experiments we extracted 223 2- and 3-Point Harmonic Shape Histogram features (HSH), and computed 100 bin histograms over each feature. Results were computed for all tasks using only 2-Point (HSH_2), only 3-Point (HSH_3) and combined 2- and 3-Point features (HSH_{23}). Finally we combined our local features with the global SHD features.

4.1. Discussion

For the previous retrieval results on the PSB, published in [1] (see Table 2), the shape features were compared measuring the retrieval performance by means of the n closest samples in feature space (in terms of $k = 1$ Nearest Neighbor). Even though this is appropriate for many retrieval

Features	Base	Coarse1	Coarse2	Coarse3
SHD	55.42%	-	-	-
HSH_2	60.75%	71.6%	78.83%	94.28%
HSH_3	51.0%	53.1%	61.0%	83.79%
HSH_{23}	68.36%	74.17%	73.43%	91.73%
$HSH_{23}+SHD$	72.88%	79.38%	80.15%	93.27%

Table 3: Results of our features with SVM leave-one-out on the PSB test set.

tasks, we stress that feature comparison has to consider the appropriate distance measure (classifier) for each feature. In our case, HSH performs very well using the NN measure (Table 2), but the true performance is revealed by the use of the SVM (Table 3). This is due to the high dimensionality of the HSH features and their local nature: different shapes can have many very similar areas and only a few, but very distinctive areas. On the other hand, not all features can take advantage of the SVM, table (3) shows that the SHD features even performed worse compared to the NN approach. The classification results shown in Table (1) can not be compared to any previous retrieval results published for the PSB. Looking at the results for the different levels of coarseness (semantic abstraction), all experiments indicate that HSH_2 is more suitable for classes of higher abstraction levels than HSH_{23} or $HSH_{23} + SHD$. This is not surprising since the high amount of detail encoded by the edge driven HSH_3 features is only distinctive for classes at a lower level.

References

- [1] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser, "The Princeton Shape Benchmark" *Shape Modeling International*, Genova, Italy, June 2004.
- [2] Mihael Ankerst, Gabi Kastenueller, Hans-Peter Kriegel, Thomas Seidl, "Nearest Neighbor Classification in 3D Protein Databases" in *Proc. ISMB*, 1999.
- [3] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Rotation invariant spherical harmonic representation of 3D shape descriptors." in *Symposium on Geometry Processing*, June 2003.
- [4] D. Saupe and D. V. Vranic, "3D model retrieval with spherical harmonics and moments." in *B. Radig and S. Florczyk, editors, DAGM 2001*, pages 392-397, September 2001.
- [5] 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.
- [6] Cortes, C. and Vapnik, V, "Support-Vector Networks." *Machine Learning Journal*, 20:273-297, 1995.
- [7] 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.
- [8] Groemer, H. "Geometric Applications of Fourier Series and Spherical Harmonics." *Cambridge University Press*, 1996.
- [9] N. Vasconcelos, "Feature Selection by Maximum Marginal Diversity: optimality and implications for visual recognition" *CVPR*, 2003