A *Bag of Words* Approach for 3D Object Categorization

Roberto Toldo, Umberto Castellani, and Andrea Fusiello

Dipartimento di Informatica, Università di Verona,
Strada Le Grazie 15, 37134 Verona, Italy
roberto.toldo@univr.it
umberto.castellani@univr.it
andrea.fusiello@univr.it

Abstract. In this paper we propose a novel framework for 3D object categorization. The object is modeled it in terms of its sub-parts as an histogram of 3D visual word occurrences. We introduce an effective method for hierarchical 3D object segmentation driven by the minima rule that combines spectral clustering – for the selection of seed-regions – with region growing based on fast marching. The front propagation is driven by local geometry features, namely the Shape Index. Finally, after the coding of each object according to the Bag-of-Words paradigm, a Support Vector Machine is learnt to classify different objects categories. Several examples on two different datasets are shown which evidence the effectiveness of the proposed framework.

1 Introduction

The availability of large collections of 3D models has increased the interest in content-based 3D search and retrieval [1–3]. Typical object retrieval systems require the user to define a query-model which output is a set of its most similar objects in the database. In general, such approach requires the comparison of the query-model with *all* the objects in the dataset according with a given matching criterion, after the coding of the object with respect to some indexing technique. Shape signatures [4] are commonly utilized as a fast indexing mechanism for shape retrieval.

In this paper we present a 3D object categorization method based on a *learning-by-example* approach [5]. Geometric features representing the query-model are fed into a Support Vector Machine (SVM) which, after a learning stage, is able to assign a *category* (or a *class*) to the query-model without an explicit comparison with all the models of the dataset. Our approach is inspired to the *Bag-of-Words* framework for textual document classification and retrieval. In this approach, a text is represented as an unordered collection of words, disregarding grammar and even word order.

The extension of such approach to non-textual data requires the building of a *visual vocabulary*, i.e., the set of all the visual analog of words. For example, in [6] images are encoded by collecting interest points which represent local salient

A. Gagalowicz and W. Philips (Eds.): MIRAGE 2009, LNCS 5496, pp. 116–127, 2009.

regions. This approach has been extended in [7] by introducing the concept of *pyramid* kernel matching. Instead of building a fixed vocabulary, the visual words are organized in a hierarchical fashion in order to reduce the conditioning of the free parameter definition (i.e., the number of bins of the histogram). Recently, in [8] the Bag-of-Words paradigm has been introduced for human actions categorization from real movies. In this case, the visual words are the vector quantization of spatiotemporal local features. The extension to 3D objects have been proposed in few work [9, 10], to the best of our knowledge. In [9] range images are synthetically generated from the full 3D model, then salient points are extracted as for the 2D (intensity) images. In [10] Spin Images are chosen as local shape descriptors after a random samples of the mesh vertices.

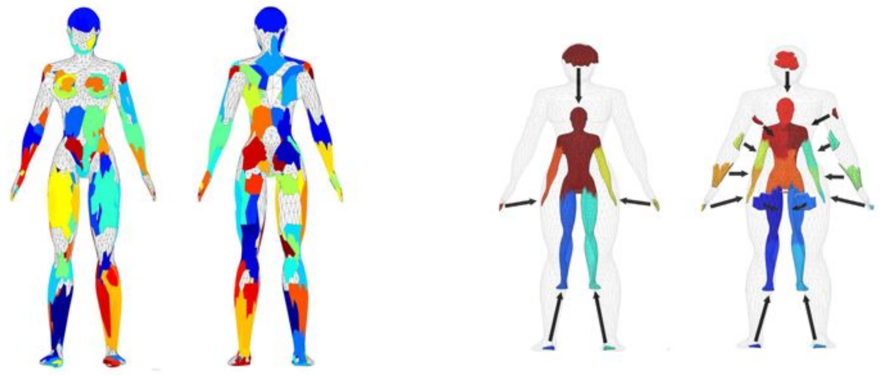
In our approach a 3D visual vocabulary is defined by extracting and grouping the geometric features of the object sub-parts from the dataset, after a hierarchical 3D object segmentation. Thank to this *part-based* representation of the object we achieve pose invariance, i.e., insensitivity to transformation which change the skeletal articulations of the 3D object [11]. Moreover, our approach is able to discriminate objects with similar skeletons, a feature that is shared by very few other works [12]. Its main steps are:

Object segmentation (Sec. 2). Spectral clustering is used for the selection of seed-regions. Being inspired by the *minima-rule* [13], the adjacency matrix is defined purposely in order to allow convex regions to belong to the same segment. Furthermore, a multiple-region growing approach is introduced to expand the selected seed-regions. In particular, a weighted fast marching is proposed by guiding the front propagation according to local geometry properties. In practice, the main idea consist on reducing the speed of the front for concave areas which are more likely to belong to the region boundaries. Then, the hierarchical segmentation is recovered by combining recursively the seeds selection and the region-growing steps.

Object sub-parts description (Sec. 3). Local region signature are introduced to define a compact representation of each sub-part. Working at the part level, as opposed to the whole object level, enables a more flexible class representation and allows scenarios in which the query model is significantly transformed (e.g., deformed) to be classified. We focus on region signatures easy to compute and partially available from the previous step (see [4] for an exhaustive overview of shape descriptors).

3D visual vocabulary construction (Sec. 4). The set of region descriptors are properly clustered in order to obtain a fixed number of 3D visual *words* (i.e., the set of clusters centroids). In practice, the clustering defines a vector quantization of the whole region descriptor space. Note that the vocabulary should be large enough to distinguish relevant changes in image parts, but not so large as to distinguish irrelevant variations such as noise.

Object categorization by SVM (Sec. 5). Each 3D object is encoded by assigning to each object sub-part the corresponding visual word. Indeed, a Bag-of-Words representation is defined by counting the number of object sub-parts assigned to each word. In practice, a histogram of visual words



(a) Seed regions after spectral clustering.

(b) Segmentation at depth 2 (left) and depth 3 (right) of the hierarchy, with corresponding seed regions.

Fig. 1. Segmentation

occurrences is build for each 3D object which represent its *global* signature [6]. Then, a SVM is trained by adopting a learning by example approach. In particular, a suitable kernel function is defined in order to implicitly implement the sub-part matching.

2 Objects Segmentation

Due to its wide ranging applications, 3D object segmentation has received a great attention lately. The recent survey by [14] and the comparative study by [15] have thoroughly covered the several different approaches developed in literature.

In the following we present a novel mesh segmentation technique that provides a consistent segmentation of similar meshes complying with the cognitive *minima rule* [13]. In addition, the final segmentation is extracted in a hierarchical structure in order to improve the flexibility in modeling the object sub-parts.

The segmentation proceeds top-down: starting with a root node corresponding to the whole mesh, the segmentation is recursively created by partitioning the current leaf nodes into two or more child nodes. The minima rule states that human perception usually divides a surface into parts along the concave discontinuity of the tangent plane [13]. Therefore this suggest to cluster in the same set convex regions and to detect boundary parts as concave ones. A concise way to express the type of shape in terms of principal curvatures is given by the *Shape Index* (SI) [16].

$$SI = -\frac{2}{\pi} \arctan\left(\frac{k_1 + k_2}{k_1 - k_2}\right) \qquad k_1 > k_2 \tag{1}$$

where k_1, k_2 are the principal curvatures of a generic vertex $x \in V$. The SI varies in $[-1, 1]$: a negative value corresponds to concavities, whereas a positive value represents a convex surface.

The key idea behind our algorithm is the synergy between two main phases: (i) the detection of similar connected convex regions as *seed*-region, and (ii) the expansion of these seed-regions using a multiple region growing approach. According to the minima-rule the SI is employed for both the phases.

2.1 Seed-Regions Detection by Spectral Clustering

The extraction of the seed-regions is accomplished with Normalized Graph Cuts [17]. It has been firstly applied to image segmentation although it is stated as a general clustering method on weighted graphs. In our case, the weight matrix is built using the SI at each vertex:

$$W(x_i, x_j) = e^{-|SI(x_i) - SI(x_j)|} \tag{2}$$

where the vertices with negative SI – i.e., those corresponding to concave regions – have been previously discarded. In this way we cluster together vertices representing the same convex shape.

Final clusters are not guaranteed to be connected. This happens because we don't take into account any (geodesic) distance information at this stage. Hence, we impose connection as a post-processing step: the final seed regions are found as connected components in the mesh graph, with vertices belonging to the same cluster. An example of seed regions found by the algorithm is shown in Fig. 1(a).

2.2 Multiple Region Growing by Weighted Fast Marching

Once the overall seed regions are found we must establish a criteria to select the starting seed regions of each node of the hierarchical segmentation tree. For each tree node we consider only the seed regions that are contained in the parent segmentation. We firstly find the two farthest seed regions. We then add more regions until the distance from the regions already added is less than half the two farthest seed regions. As explained next, the distance between two regions can be efficiently calculated with the Fast Marching algorithm [18, 19]. In particular, when the seed regions of the current tree node are found, we expand them using a *weighted* geodesic distance. In formulae, given two vertices $x_0, x_1 \in V$, we define the *weighted geodesic distance* $d(x_0, x_1)$ as

$$d(x_0, x_1) = min_\gamma \left\{ \int_0^1 \|\gamma'\| w(\gamma(t)) dt \right\} \tag{3}$$

where $w(\cdot) =$ is a weight function (if $w(\cdot) = 1$ this is the classic geodesic distance) and γ is a piecewise regular curve with $\gamma(0) = x_0$ and $\gamma(1) = x_1$. Our weight function is based on the Shape Index SI:

$$w(x) = e^{\alpha SI(x)} \tag{4}$$

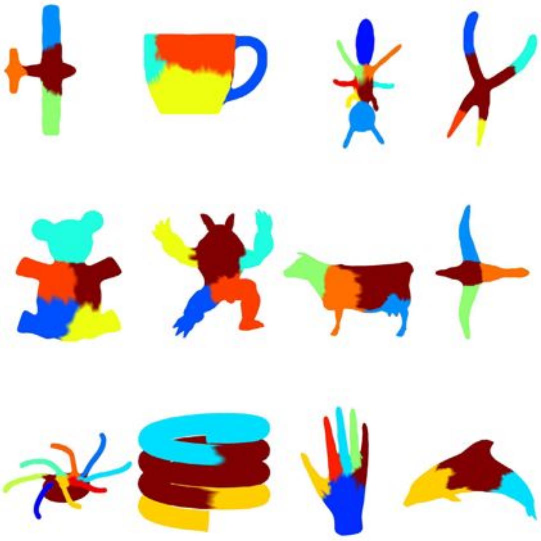


Fig. 2. Examples of segmentation extracted on several meshes of the Aim@Shape Dataset

where α is an arbitrary constant. An high α value heavily slow down the front propagation where the concavity are more prominent. In our categorization paradigm we used a fixed $\alpha = 5$ to obtain consistent segmentations.

An example segmentation along with starting seed regions is shown in Fig. 1(b). Several other examples of segmentation on different objects are shown in Fig. 2.

The overall hierarchical algorithm is summarized below:

Algorithm 1. Hierarchical clustering

```
1. Find all seed-regions $S$.
2. Initialize $C$ as the entire mesh and place in the priority queue $Q$.
3. Get the current top cluster $C \in Q$ and remove it from $Q$.
4. Find starting regions $S_C \in S \bigcap C$.
   If the starting regions are more than one go to next step else go to step 6.
5. Find final cluster starting from $S_C$ trough weighted geodesic distance and
   add them to $Q$.
   These are child cluster of $C$ in the hierarchical tree.
6. If $Q$ is empty stop, else go to step 3.
```

3 Segment Descriptors

We chose four type of descriptors to represent each extracted region. The first two are based on SI and *Curvedness* [16]. Both encode local surface geometry properties for each vertex. In particular, the SI allows the classification of the surface among peek, valley, saddle, and so on. The Curvedness CU instead, is a concise way to measure the size of a local patch:

$$CU = \frac{2}{\pi} \ln \sqrt{\frac{k_1^2 + k_2^2}{2}} \tag{5}$$

The two descriptors SIH and CUH are defined as the normalized histograms of the observed SI and CU values in the region vertices, respectively.

The other two descriptors are normalized region histograms of vertex-distances derived directly from our segmentation algorithm. The idea is to describe the shape of a region in relationship with its starting seed. In practice, we compute the geodesic distance and the weighted geodesic distance of each vertex of a segment to its seed region. The point-to-seed-region distance is defined as the geodesic distance between the point itself and its closest point belonging to the seed region. The two descriptors GD and GDW are the normalized histograms of such distances (respectively) over the vertices of the segment.

Note that GD can be interpreted as an approximation of the eccentricity [20], and GDW, implicitly encodes also the local surface geometry information since the weight function depends on the SI, according to Eq. (4).

Note further, that the number of bins chosen for each histogram is a critical choice. A small number reduce the capability of the region descriptor in discriminating among different segments. On the other hand, a high number increases the noise conditioning. Hence we introduce, for each descriptor, histograms with different number of bins in order to obtain a *coarse-to-fine* regions representation.

4 3D Visual Vocabulary Construction

The different sets of region descriptors must be clustered in order to obtain several visual words. Since we start with a hierarchical segmentation and different types of descriptors, we adopted a multi-clustering approach rather than merging descriptors in a bigger set. Before the clusterization, the sets of descriptors are thus split in different subsets as illustrated in Fig. 3. The final clusters are obtained with a k-means algorithm. Again, instead of setting a fixed free parameter k, namely the number of cluster, we carry out different clusterizations while varying its value.

Once the different clusters are found we retain only their centroids, which are our *visual words*. In Fig. 4 an example of descriptors subset clusterization with relative distance from centroid is shown. Note that object sub-parts from different categories may fall in the same cluster since they share similar shape.

More in details, at the end of this phase we obtain the set of visual vocabularies $V_{l,s}^{d,b,c}$, where:

- l identifies the region level of the hierarchical 3D segmentation ($l \in \{2, 3\}$),
- s identifies the index of the multiple 3D segmentation (variable segmentation parameter $s \in \{4, 8, 12\}$),

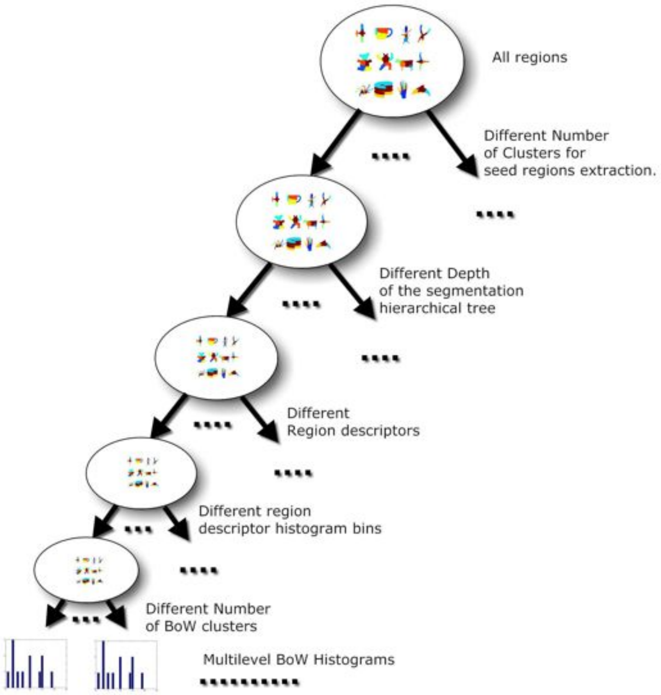

All regions
Different Number of Clusters for seed regions extraction.
Different Depth of the segmentation hierarchical tree
Different Region descriptors
Different region descriptor histogram bins
Different Number of BoW clusters
Multilevel BoW Histograms

Fig. 3. The Vocabulary construction is performed in a multilevel way. At the beginning we have all region extracted for different numbers of seed regions (variable segmentation parameter). The regions are divided by the different segmentations and by the different depth of the segmentation tree. For every region, different descriptors are attached. The different region descriptors are divided by the type of descriptor and its number of bins. The final clusterizations are obtained with varying number of clusters. At the end of the process we obtain different Bag-of-Words histograms for each mesh.

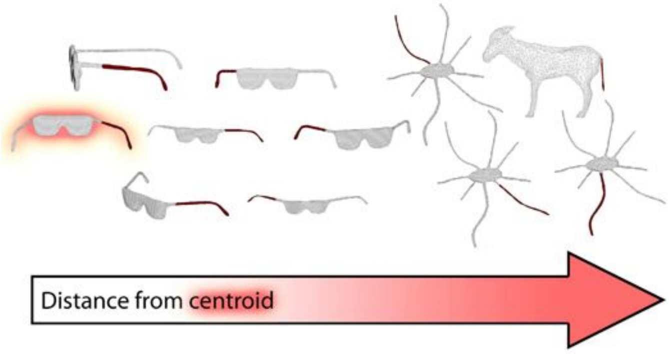


Fig. 4. Example of a Bag-of-Words cluster for SI descriptors. The centroid is highlighted with red and others region in the same cluster are sorted by distance from centroid. Note that sub-parts of meshes from different categories may fall in the same cluster since they share similar shape.

- d identifies the region descriptor types ($d \in \{SIH, CUH, GD, GDW\}$),
- b identifies the refined level of the region descriptor (number of histogram bins $b \in \{10, 25, 50, 100, 200\}$),
- c identifies the refined level of the vocabulary construction (number of clusters $c \in \{20, 40, 60, 80\}$).

In order to construct a Bag-of-Words histogram of a new 3D object, we compare its regions descriptors with the visual words of the associated visual vocabularies.

5 Object Categorization by SVM

One of the most powerful classifier for object categorization is the Support Vector Machine (SVM) (see [21] for a tutorial). The SVM works in a vector space, hence the Bag-of-Words approach fits very well, since it provides a vector representation for objects. In our case, since we work with multiple vocabularies, we define the following positive-semi-definite kernel function:

$$K(A, B) = \sum_{l,s,d,b,c} k(\phi_{l,s}^{d,b,c}(A), \phi_{l,s}^{d,b,c}(B)), \tag{6}$$

where (A, B) is a pair of 3D models, and $\phi_{l,s}^{d,b,c}(\cdot)$ is a function which returns the Bag-of-Words histogram with respect to the visual vocabulary $V_{l,s}^{d,b,c}$. The function $k(\cdot, \cdot)$ is in turn a kernel which measures the similarity between histograms h^A, h^B:

$$k(h^A, h^B) = \sum_{i=1}^{c} min(h_i^A, h_i^B), \tag{7}$$

where h_i^A denotes the count of the i^{th} bin of the histogram h^A with c bins. Such kernel is called *histogram intersection* function and it is shown to be a valid kernel [7]. Histograms are assumed to be normalized such that $\sum_{i=1}^{n} h_i = 1$. Note that, as observed in [7] the proposed kernel implicitly encodes the subparts matching since corresponding segments are likely to belong to the same histogram bin. Indeed, the histogram intersection function counts the number of sub-parts matching being intermediated by the visual vocabulary.

Finally, since the SVM is a binary classifier, in order to obtain an extension to a multi-class framework, a one-against-all approach [5] is followed.

6 Results

We tested our categorization paradigm with two different datasets. For each dataset we performed a Leave-One-Out cross validation [5].

6.1 TOSCA Non Rigid Shape Dataset

The TOSCA dataset [22–24], publicly available[1], contains various non-rigid shapes in a variety of poses divided by category. The dataset is composed by: 9 cats, 8 men, 9 dogs, 21 gorillas, 17 horses and 9 women. The meshes used are shown in Fig. 5. Please note that each category is composed by the same model with different pose. Furthermore, some classes are very similar, e.g. men and women, and contains a number of elements very variable.

In this case, our categorization algorithm works perfectly in each category with a rate of success of 100%. This experiment shows that our system copes finely with the categorization of objects that present high inter-class similarity. Nevertheless, the methods is robust with objects that appear with different poses, by varying strongly their skeletal articulations (e.g., the gorillas).

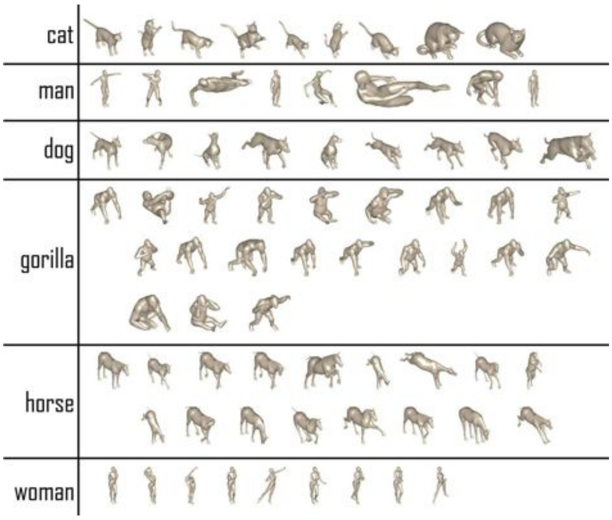
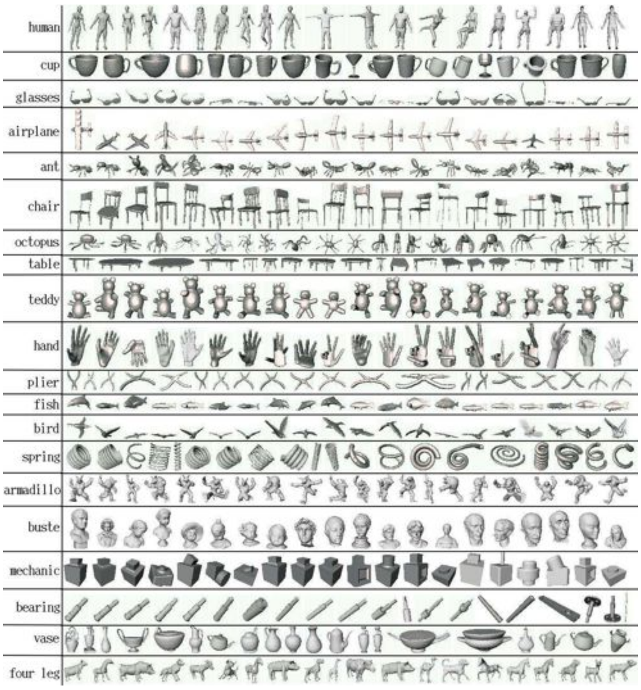


Fig. 5. TOSCA Non rigid Shape Dataset models, divided by category. Overall success rate of categorization is **100%**.

6.2 Aim@Shape Watertight

The Aim@Shape Watertight dataset has been used for various retrieval contests [25]. This dataset contains 20 categories each composed of 20 meshes. The entire dataset is shown in Fig. 6 together with the categorization results. In this case the algorithm fails with some meshes, but the overall rate of success is still fairly good. The dataset is tough since there are many categories and objects inside the same category can be very different. We can notice that the system is less accurate when the shapes are CAD-like (e.g. *mechanics, bearings* and *tables*).

[1] `http://tosca.cs.technion.ac.il/data_3d.html`



Category	% of success
Human	80
Cup	85
Glasses	95
Airplane	90
Ant	100
Chair	95
Octopus	95
Table	80
Teddy	100
Hand	80
Plier	100
Fish	85
Bird	80
Spring	95
Armadillo	100
Buste	95
Mechanic	75
Bearing	60
Vase	75
Four Legs	80

Fig. 6. Aim@Shape Watertight Dataset objects, divided by category and success rate of categorization. Overall the rate is **87.25%**.

This suggests that the descriptors based on curvature may not discriminate enough these kind of regions. Future improvements of the system can be obtained by adding more descriptors.

6.3 Timing

The categorization pipeline is computationally efficient in each sub-part. We used an entry level laptop at $1.66Ghz$ to perform tests. The code is written in Matlab with some parts in C. An entire mesh segmentation of 3500 vertices, with a maximum hierarchical depth of four is computed in less than a minute. Precisely, $\sim 8s$ are necessary to extract all the seed regions, while $\sim 50s$ are needed to compute the entire hierarchical segmentation. Region descriptors are computed efficiently. On the average it only takes $\sim 0.2s$ to extract all the four descriptors of a single region. Also the k-means clusterizations are not time consuming. For example 10 clusters for 300 points each composed of 200 feature are extracted in less than one second. Finally, the time needed to train a SVM with 400 elements is $\sim 80s$, while the time needed to validate a single element is about $\sim 2s$.

7 Conclusions

In this paper a new approach for 3D object categorization is introduced basing on the Bag-of-Words paradigm. The main steps of the involved categorization pipeline have been carefully designed by focusing on both the effectiveness and efficiency.

The Bag-of-Words approach allows naturally the object sub-parts encoding by combining effectively sub-part descriptors into several visual vocabularies. Moreover, we have proposed a Learning-by-Example approach by introducing a local kernel which implicitly performs the object sub-parts matching. In particular, the object categories are inferred without an exhaustive pairwise comparison between all the models.

The experimental results are encouraging. In particular, our framework is able to categorize objects which heavily deform their shape and change significantly their pose. Nevertheless, the method is able to distinguish also categories with the same skeletal structure (e.g., a man from a woman).

Acknowledgments

This paper was partially supported by PRIN 2006 project 3-SHIRT. Thanks to the anonymous reviewers for useful remarks and suggestions.

References

1. Iyer, N., Jayanti, S., Lou, K., Kalynaraman, Y., Ramani, K.: Three dimensional shape searching: State-of-the-art review and future trend. Computer Aided Design 5(37), 509–530 (2005)
2. Funkhouser, T., Kazhdan, M., Patrick, M., Shilane, P.: Shape-based retrieval and analysis of 3D models. Communications of the ACM 48(6), 58–64 (2005)
3. Tangelder, J.W., Veltkamp, R.C.: A survey of content based 3d shape retrieval methods. In: International Conference on Shape Modelling and Applications (2004)
4. Shilane, P., Funkhouser, T.: Selecting distinctive 3D shape descriptors for similarity retrieval. In: International Conference on Shape Modelling and Applications. IEEE Computer Society Press, Los Alamitos (2006)
5. Duda, R., Hart, P., Stork, D.: Pattern Classification, 2nd edn. John Wiley and Sons, Chichester (2001)
6. Cruska, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: ECCV Workshop on Statistical Learning in Computer Vision (2004)
7. Grauman, K., Darrell, T.: The pyramid match kernel: Efficient learning with sets of features. Journal of Machine Learning Research 8(2), 725–760 (2007)
8. Laptev, I., Marsza, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: IEEE Conference on Computer Vision and Pattern Recognition (2008)
9. Ohbuchi, R., Osada, k., Furuya, T., Banno, T.: Salient local visual features for shape-based 3d model retrieval. In: International Conference on Shape Modelling and Applications (2008)

10. Li, Y., Zha, H., Qin, H.: Sapetopics: A compact representation and new algorithm for 3d partial shape retrieval. In: International Conference on Computer Vision and Pattern Recognition (2006)
11. Gal, R., Shamir, A., Cohen-Or, D.: Pose-oblivious shape signature. IEEE Transaction on Visualization and Computer Graphics 13(2), 261–271 (2007)
12. Tam, G.K.L., Lau, W.H.R.: Deformable model retrieval based on topological and geometric signatures. IEEE Transaction on Visualization and Computer Graphics 13(3), 470–482 (2007)
13. Hoffman, D.D., Richards, W.A.: Parts of recognition. Cognition, 65–96 (1987)
14. Shamir, A.: A survey on mesh segmentation techniques. Computer Graphics Forum 27, 1539–1556 (2008)
15. Attene, M., Katz, S., Mortara, M., Patane, G., Spagnuolo, M., Tal, A.: Mesh segmentation - a comparative study. In: Proceedings of the IEEE International Conference on Shape Modeling and Applications. IEEE Computer Society Press, Los Alamitos (2006)
16. Petitjean, S.: A survey of methods for recovering quadrics in triangle meshes. ACM Computing Surveys 34(2), 211–265 (2002)
17. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22(8), 888–905 (2000)
18. Sethian, J.: A fast marching level set method for monotonically advancing fronts. In: Proceedings of the National Academy of Sciences, vol. 93 (1996)
19. Kimmel, R., Sethian, J.: Computing geodesic paths on manifolds. In: Proceedings of the National Academy of Sciences, vol. 95 (1998)
20. Ion, A., Artner, N.M., Peyr, G., Marmol, S.B.L., Kropatsch, W.G., Cohen, L.: 3d shape matching by geodesic eccentricity. In: Proceedings of S3D Workshop (2008)
21. Burges, C.: A tutorial on support vector machine for pattern recognition. Data Mining and Knowledge Discovery 2, 121–167 (1998)
22. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical geometry of non-rigid shapes. Springer, Heidelberg (2007)
23. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Calculus of non-rigid surfaces for geometry and texture manipulation. Transactions on Visualization and Computer Graphics 13(5), 902–913 (2007)
24. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Efficient computation of isometry-invariant distances between surfaces. SIAM Journal of Scientific Computing 28(5), 1812–1836 (2006)
25. Veltkamp, R.C., ter Haar, F.B.: Shrec 2007 3d retrieval contest. Technical Report UU-CS-2007-015, Department of Information and Computing Sciences (2007)