The Visual Computer manuscript No.
(will be inserted by the editor)

Roberto Toldo ·
Umberto Castellani ·
Andrea Fusiello

The *bag of words* approach on 3D domain: retrieval, partial matching and categorization

Abstract In this paper the effectiveness of the *Bag of Words* framework is exploited for the 3D domain. Such approach provides a part-based representation by partitioning the objects into subparts and by characterizing each segment with different geometric descriptors. In this fashion one object is modeled as an histogram of subparts occurrences which becomes its signature. Therefore, such signature is fed to a Support Vector Machine which is learnt to classify different objects categories. Several examples on the Aim@Shape watertight dataset and on the Google Shape dataset demonstrate the versatility of the proposed method in working with either 3D objects with articulated shape changes or partially occluded or compound objects. In particular an exhaustive experimental section is proposed by focusing on different applications namely i) 3D object retrieval, ii) partial shape matching, and iii) 3D object categorization. Results are encouraging as shown by the comparison with other methods for each of the analyzed scenarios.

1 Introduction

In the last years, the proliferation of large databases of 3D models caused a surge of interest in methods for content-based object retrieval [1–3]. One of major challenges in the context of data retrieval is to elaborate a suitable canonical characterization of the entities to be indexed. In the literature, this characterization is referred to as *descriptor* or *signature*. Since the descriptor serves as a key for the search process, it decisively influences the performance of the search engine in terms of computational efficiency and relevance of the results. Decriptors are *global* or *local*. The former consist in a set of features that effectively and concisely describe the entire 3D model [4]. The latter are instead collections of local features of relevant object subparts [5].

In this paper we exploit the *Bag-of-Words* (BoW) approach in order to combine and merge local information into a global object signature. The BoW framework has been proposed for textual document classification and retrieval. A text is represented as an unordered collection of words, disregarding grammar and even word order. The extension of such approach to visual data requires the building of a *visual vocabulary*, i.e., the set of the visual analog of words. For example, in [6] 2D images are encoded by collecting interest points which represent local salient 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 influence of the free parameters (e.g., the number of bins of the histogram). Finally, in [8] the BoW paradigm has been introduced for human actions categorization from real movies. In this case, the visual words are the quantized vectors of spatiotemporal local features. The extension of the BoW paradigm to 3D objects is non-trivial and has been proposed only in few recent works [9–11]. In [9] range images are synthetically generated from the full 3D model and subsequently treated as 2D (intensity) images. In [10,11] Spin Images are chosen as local shape descriptors after sampling the mesh vertices. Usually local techniques are defined by point-based features rather than by segmentation. Only recently [12] proposed a part-based retrieval method by partitioning an object to meaningful segments and finding analogous parts in other objects.

In our approach a 3D visual vocabulary is defined by extracting and grouping the geometric features of the object sub-parts. Thank to this *part-based* representation of the object we achieve pose invariance, i.e., insensitivity to transformations that change the articulations of the 3D object [13]. In particular, our method is able to discriminate objects with similar skeletons, a feature that is shared by very few other works like [14].

Beside being very effective in object retrival, the BoW representation proved valuable also in the task of 3D object categorization. In particular we devised a *learning-by-example* approach [15]: 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.

In summary, the proposed approach is composed by the following main steps:

Object sub-parts extraction (Sec. 2). Spectral clustering is used for the selection of seed-regions. Being inspired by the *minima-rule* [16], the adjacency matrix is tailored in order to allow convex regions to be-

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

long to the same segment. Furthermore, a multiple-region growing approach is introduced to expand the selected seed-regions, based on a weighted fast marching. 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 segmentation is recovered by combining the seeds selection and the region-growing steps.

Object sub-parts description (Sec. 3). Local region descriptors are introduced to define a compact representation of each sub-part. Working at the part level, as opposed to the whole object, enables a more flexible class representation and allows scenarios in which the query model is significantly deformed. We focus on region descriptors easy to compute and partially available from the previous step (see [5] for an exhaustive overview of shape descriptors).

3D visual vocabularies 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 fact, 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 object parts, but not so large as to discriminate irrelevant variations such as noise.

Object representation and matching (Sec. 5). Each 3D object is encoded by assigning to each object sub-part the corresponding visual word. The BoW representation is defined by counting the number of object subparts assigned to each word. In practice, a histogram of visual words occurrences is built for each 3D object which represent its *global* signature [6]. Matching is accomplished by comparing the signatures.

Object categorization by SVM (Sec. 6). 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.

Finally, the proposed approach has been successfully applied on different applicative scenarios, namely i) 3D object retrieval, ii) partial shape matching, and iii) 3D object categorization.

2 Objects segmentation

The recent survey by [17] and the comparative study by [18] 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, depends on very few parameters and is very fast. It is inspired by the *minima rule* [16]: “for the purposes of visual recognition, the huma visual system divides 3D shapes into parts at negative minima of principal curvature”. Therefore this suggests to cluster in the same set convex regions and to detect boundary parts as concave ones. A concise way to characterize the shape in terms of principal curvatures is given by the *Shape Index* [19].

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

where k_1, k_2 are the principal curvatures of a generic vertex $x \in V$. The Shape Index 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, and (ii) the expansion of these seed-regions using a multiple region growing approach. According to the minima-rule the Shape Index is employed in both phases.

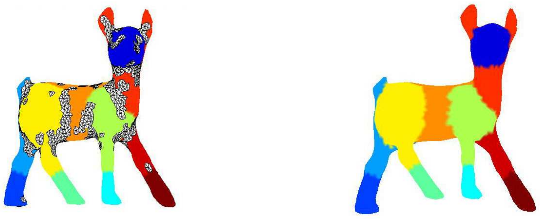
2.1 Seed-regions detection by Spectral Clustering

The extraction of the seed-regions is accomplished with Normalized Graph Cuts [20]. This approach 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 $w(x_i, x_j)$ is built using the Shape Index at each vertex:

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

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

The number of clusters, needed by the Spectral clustering approach, is linked, but not equal, to the number of final segments. Indeed, clusters are not guaranteed to be connected in the mesh. This happens because we do not take into account geodesic distance information at this stage: we cluster only according to the curvature value at each vertex. 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 Figure 1(a).



(a) Seed regions found with spectral clustering.

(b) Final Segmentation.

Fig. 1 An example of segmentation.

2.2 Multiple region growing by weighted fast marching

Once the overall seed regions are found, we must establish a criteria to assign the vertices that don't belong to any initial seed region. The key idea is to expand the initial seeds region using a *weighted* geodesic distance. Again, the weight at each vertex is chosen according to the minima-rule. In formulae, given two vertices $x_0, x_1 \in V$, we define the *weighted geodesic distance* $d(x_0, x_1)$ as

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

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 s:

$$w(x) = e^{\alpha s(x)} \quad (4)$$

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

An example segmentation along with starting seed regions is shown in Figure 1(b). Several other examples of segmentation on different objects are shown in Figure 2. Similar parts seem to be segmented in a similar manner (provided that the parameters of the segmentations are equal).

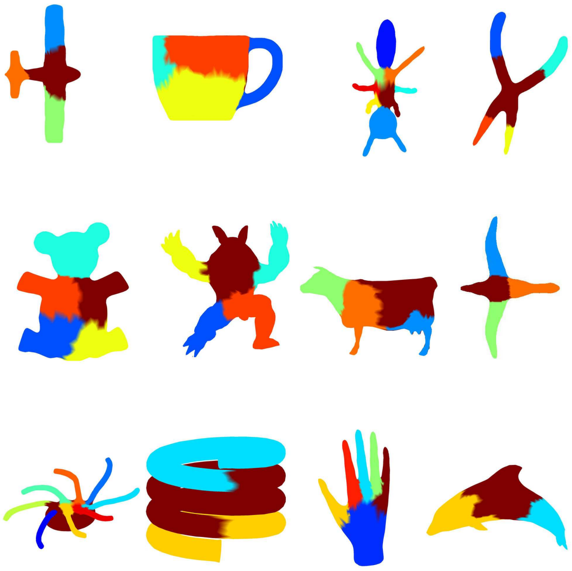


Fig. 2 Examples of segmentation of some objects from the Aim@Shape Dataset.

3 Segment descriptors

We chosed four type of descriptors to represent each extracted region: the Shape Index Histogram (SIH), *Radial Geodesic Distance Histogram* (RGDH), Normal Histogram (NH), and *Geodesic Context* (GC). The first three are defined as the normalized histograms of local measures computed for each point of the region, namely shape index, *radial geodesic distance* and normal. The fourth descriptor depends on the relative positions of the regions and thus it is a context descriptor.

The *radial geodesic distance* measures the geodesic distance of a surface point to the geodesic centroid of the region. In our case, for computation efficiency, we approximate the geodesic centroid as the closest point on the mesh to the Euclidean centroid.

The *Geodesic Context* descriptor for a region is built computing the histogram of the geodesic distance between its centroid and the centroids of the other regions. The *GC* descriptor, defined for regions, resembles the shape context descriptor [21], defined for points.

Please note that the number of bins chosen for each histogram of the four descriptors 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 vocabularies construction

The different sets of region descriptors must be clustered in order to obtain several visual words. Since we start with different segmentations 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 Figure 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 this value.

Once the different clusters are found we retain only their centroids, which are our *visual words*. In Figure 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 $V_s^{d,b,c}$, where:

- s identifies the index of the multiple 3D segmentation (variable segmentation parameter $s \in \{6, 8, 10, 12, 14\}$),
- d identifies the region descriptor types ($d \in \{SI, RG, N, GC\}$),
- b identifies the refined level of the region descriptor (number of histogram bins $b \in \{20, 30, 40, 50\}$),

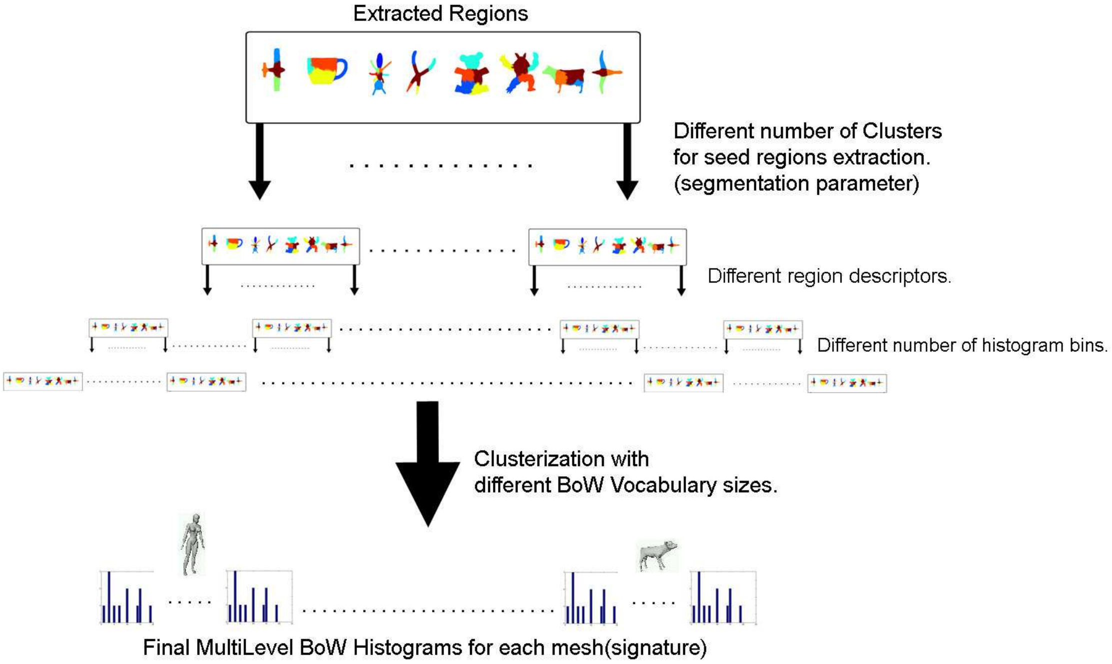

Extracted Regions
Different number of Clusters for seed regions extraction. (segmentation parameter)
Different region descriptors.
Different number of histogram bins.
Clusterization with different BoW Vocabulary sizes.
Final MultiLevel BoW Histograms for each mesh(signature)

Fig. 3 The construction of the vocabularies is performed in a multilevel way. At the beginning we have all region extracted for different numbers of seed regions (variable segmentation parameter). 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.

- c identifies the refined level of the vocabulary construction (number of clusters).

5 3D representation and matching

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 corresponding visual vocabulary. In practice, each segment is assigned to the most similar visual words. The Bag-of-Words representation is obtained by counting the number of segment assigned to each word. The resulting signature is a very sparse vector of occurences. Finally, the objects matching is obtained by comparing their respective signature by using standard metric for histograms.

6 Object categorization by SVM

One of the most powerful classifier for object categorization is the Support Vector Machine (SVM) (see [22] 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:

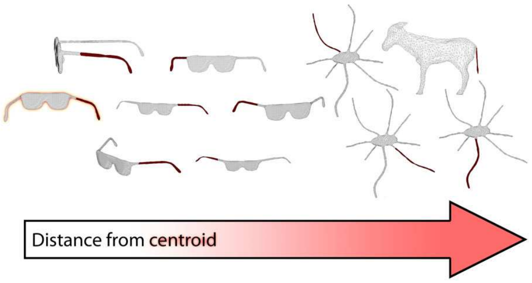


Fig. 4 Example of a Bag-of-Words cluster for SI descriptors. The centroid is highlighted with red and others region in the same cluster are sorted by distance from centroid. Note that sub-parts of meshes from different categories may fall in the same cluster since they share similar shape.

$$K(A,B) = \sum_{s,d,b,c} k(\phi_s^{d,b,c}(A), \phi_s^{d,b,c}(B)), \qquad (5)$$

where (A,B) is a pair of 3D models, and $\phi_s^{d,b,c}(\cdot)$ is a function which returns the Bag-of-Words histogram with respect to the visual vocabulary $V_s^{d,b,c}$. The function

$k(\cdot,\cdot)$ is in turn a kernel which measures the similarity between histograms h^A, h^B:

$$k(h^A, h^B) = \sum_{i=1}^{c} min(h_i^A, h_i^B), \qquad (6)$$

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* 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 sub-parts 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 [15] is followed.

7 Results

In order to prove the effectiveness and the generalization capability of the proposed paradigm we tested it with several different retrieval and categorization tasks, also working with composed or partial mashes. The two datasets employed are the Aim@Shape watertight dataset and the Tosca Dataset. The first is composed of 400 meshes of 20 different classes (see Fig. 5). The dataset is tough since there are many categories and objects inside the same category can be very different. The second is composed of 13 shape classes. In each class, the shape underwent different types of transformations, namely: null (no transformation), isometry, topology (connectivity change obtained by welding some of the shape vertices), isometry+topology, triangulation (different meshing of the same shape) and partiality (missing information,obtained by making holes and cutting parts of the shape). In this case difficulties arise because the categories are very similar each other (see Fig. 6 and Fig. 7 for more details).

7.1 Aim@Shape Watertight

The Aim@Shape Watertight dataset has been used for various retrieval contests [23]. Firstly, we compared our method with the participant of the Aim@Shape Watertight 2007 contest [23]. We used precision and recall to evaluate our results, that are two fundamental measures often used in evaluating search strategies. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database, while precision is the ratio of the number of relevant records retrieved to the size of the return vector [24]. In table 2 the precision and recall of our approach along with the results of the other methods are reported, while in figure

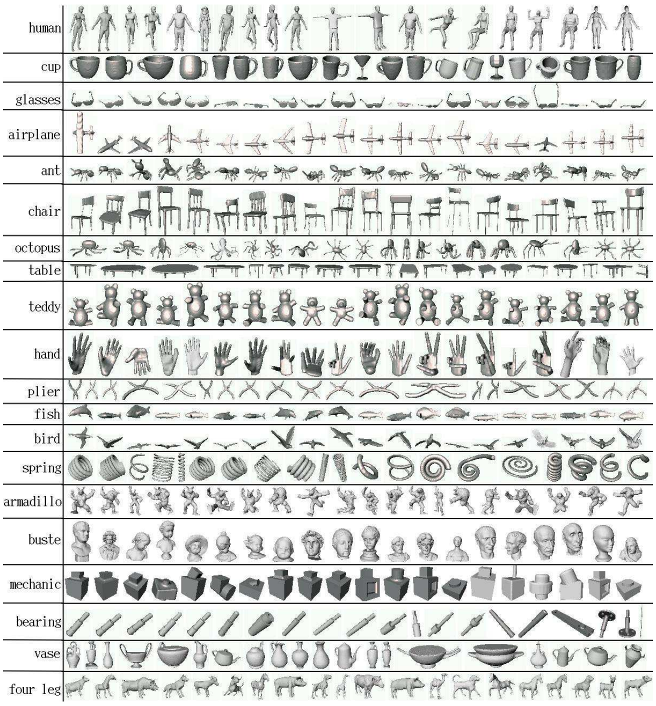


Fig. 5 Aim@Shape Watertight Dataset

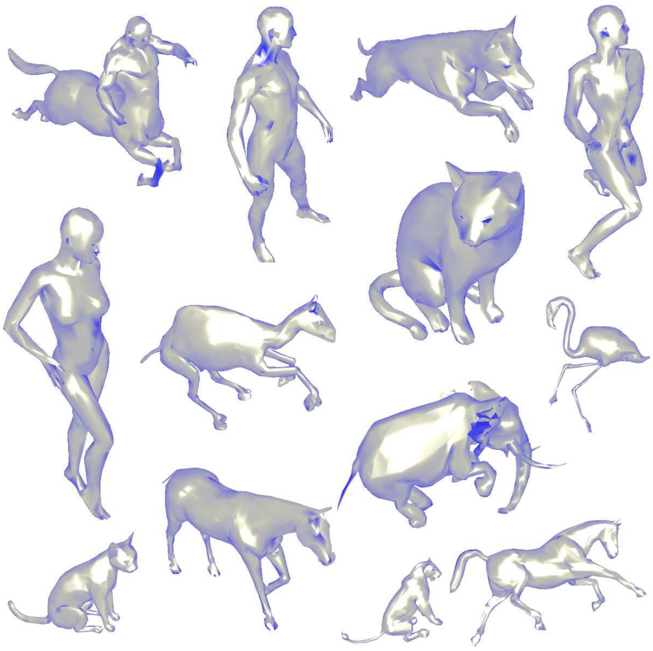


Fig. 6 Example of different kind of objects in the Tosca dataset. The category are 13, namely: centaur,horse,two males,female,two cats,dog, horse, tiger, elephant, dromedary and flamingo

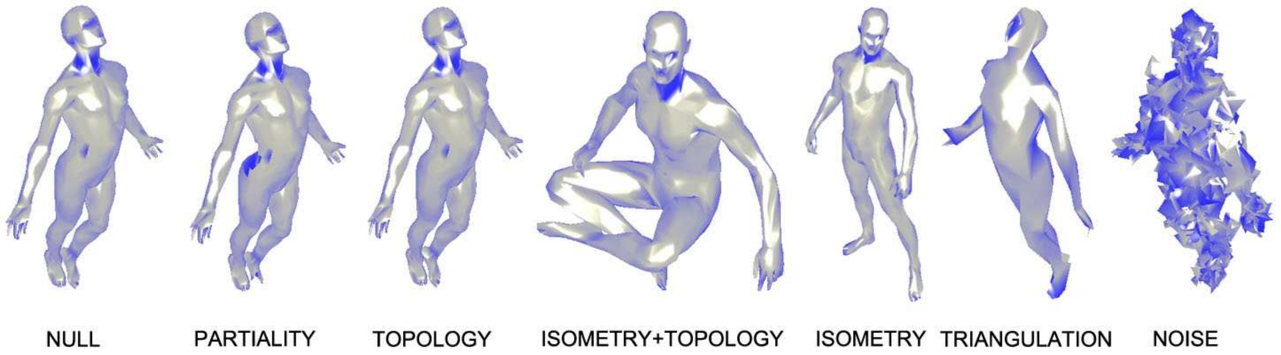


Fig. 7 Example of different type of transformation in the Tosca dataset.

Category	Precision after 20	Category	Precision after 20
Human	0.53	Cup	0.46
Glasses	0.90	Airplane	0.73
Ant	0.92	Chair	0.57
Octopus	0.61	Table	0.52
Teddy	0.94	Hand	0.32
Plier	0.99	Fish	0.8
Bird	0.4	Spring	0.96
Armadillo	0.94	Buste	0.57
Mechanic	0.80	Bearing	0.44
Vase	0.8	Four Legs	0.32

Table 1 Precision for each category of the Aim@Shape dataset after 20 retrieved items.

8 the precision vs recall plot of our method is shown. The results divided by category are shown in figure 1. The algorithm fails with some meshes, but the overall rate of success is still fairly good.

In the second task we tested our method with some query test models that are composed of parts of the original dataset. The query test models are 30 and each query model shares common subparts with (possibly) more than one model belonging to the ground-truth dataset. The query set is shown in figure 9. Again, we compared our method with the participant of the Aim@Shape Partial Matching 2007 contest [23]. In order to evaluate the performance, a set of highly relevant, marginally relevant and non-relevant models belonging to the dataset has been associated to each query model. The performance indicator used is the Normalized Discounted Cumulated Gain vector (NDCG) [25], which is recursively defined as

$$DCG[i] = \begin{cases} G[i] & \text{if i = 1} \\ DCG[i-1] + G[i]\log_2(i)) & \text{otherwise} \end{cases} \quad (7)$$

where G[i] represents the value of the gain vector at the position i. In our case, for a specific query, G(i) equals 2 for highly relevant models, 1 for marginally relevant models and 0 for non-relevant models. The normalized discounted cumulated gain vector NDCG is obtained by dividing DCG by the ideal cumulated gain vector. In figure 10 the NDCG of our approach along with the results of the other methods are reported. We can notice how our method performs better than the other methods considered.

Finally, we tested the dataset in a categorization problem. We performed the test using a Leave-One-Out approach. The Overall success rate is high: **87.25%**. In table the different results for each category are reported.

Precision after	20	40	60	80
Ideal	1	0.5	0.333	0.25
Tung et al.	0.714	0.414	0.290	0.225
Our Approach	**0.648**	**0.379**	**0.270**	**0.210**
Akgul et al.	0.626	0.366	0.262	0.205
Napoleon et al.	0.604	0.366	0.262	0.205
Daras et al.	0.564	0.346	0.252	0.199
Chaouch et al.	0.546	0.329	0.241	0.190
Recall after	20	40	60	80
Ideal	1	1	1	1
Tung et al.	0.714	0.828	0.872	0.902
Our Approach	**0.648**	**0.758**	**0.808**	**0.841**
Akgul et al.	0.626	0.732	0.786	0.821
Napoleon et al.	0.604	0.732	0.788	0.822
Daras et al.	0.564	0.692	0.756	0.798
Chaouch et al.	0.546	0.658	0.724	0.763

Table 2 Precision and Recall after 20, 40, 60 and 80 retrieved items for the Aim@Shape dataset.

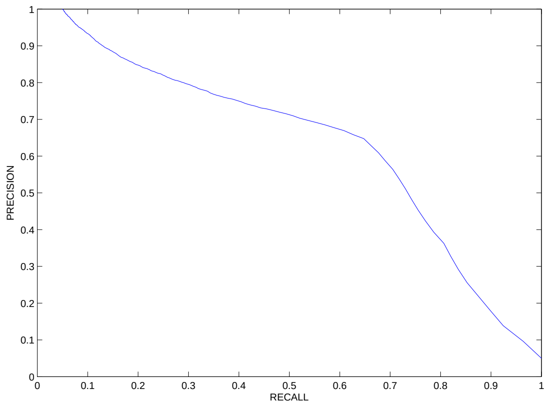
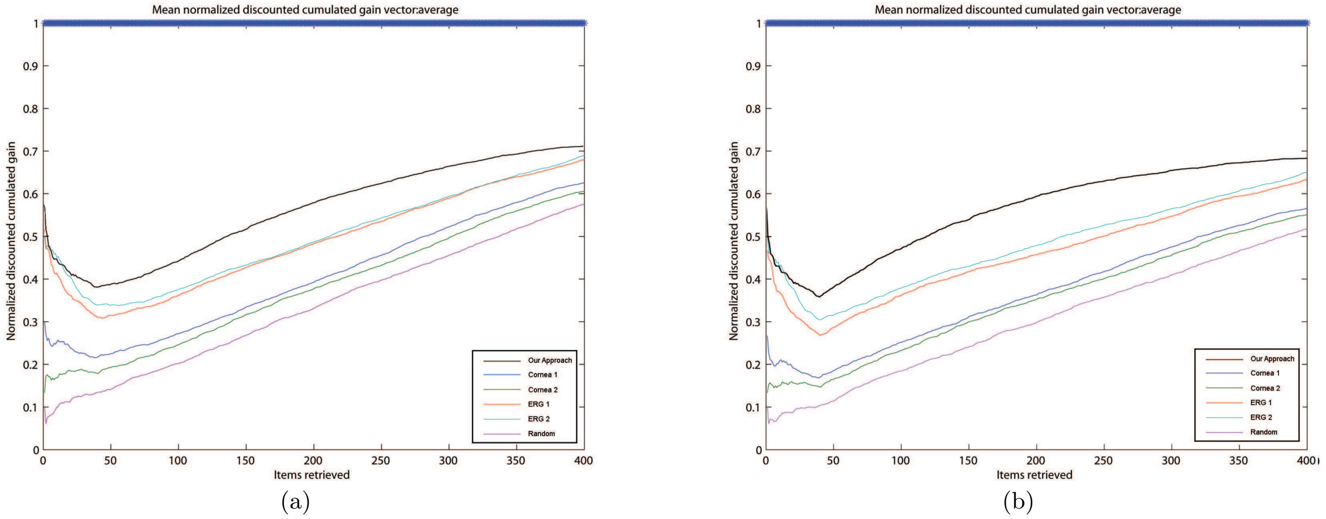


Fig. 8 Precision vs Recall for the Aim@Shape dataset.



(a) (b)

Fig. 10 Overall Normalized Discount Cumulated Gain considering only highly relevant models 10(a) and both highly relevant and marginally relevant models 10(b) for the Aim@Shape partial matching contest.

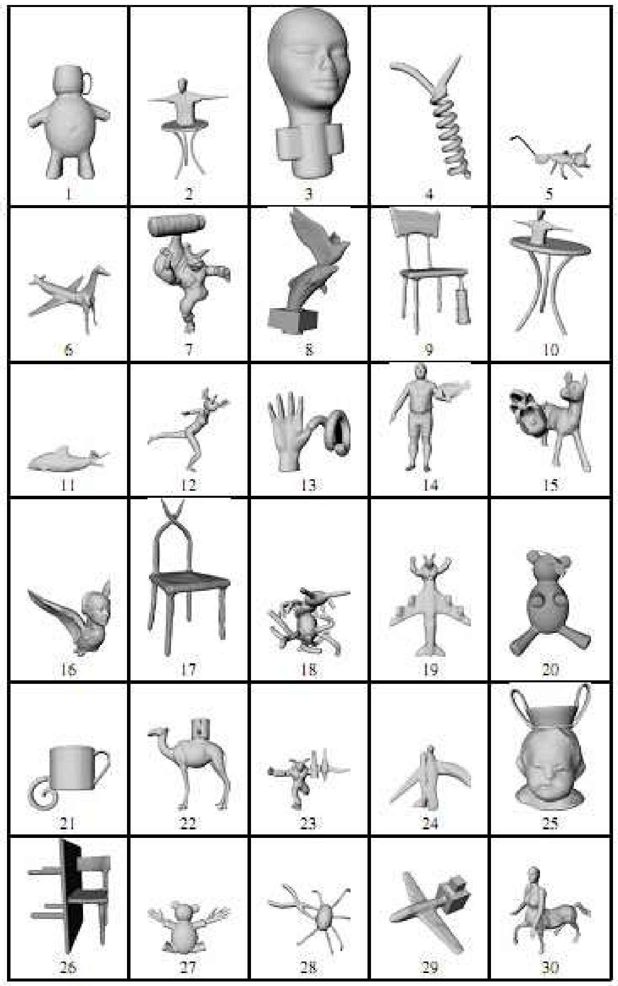


Fig. 9 Aim@Shape Partial Matching query objects.

Category	S.R.	Category	S.R.
Human	0.8	Cup	0.85
Glasses	0.95	Airplane	0.9
Ant	1.0	Chair	0.95
Octopus	0.95	Table	0.8
Teddy	1.0	Hand	0.8
Plier	1.0	Fish	0.85
Bird	0.8	Spring	0.95
Armadillo	1.0	Buste	0.95
Mechanic	0.75	Bearing	0.6
Vase	0.75	Four Legs	0.8

Fig. 11 Success Rate (S.R.) of categorization of the Aim@Shape dataset. The overall rate is **87.25%**.

7.2 Tosca

We tested also the Tosca dataset with a retrieval and a categorization task. In this case we divided the results for the different type of transformation.

Again, for the retrieval task, we measured the performance using the precision and the recall. In this case the number of object per category is variable. The query length have thus been made variable according to size of the specific category, so that 1 is the maximum value of precision obtainable. The overall precision is 0.74%.

For the categorization task, the Leave-One-Out validation have been used. The overall success rate is very high: 0.98%.

The success rate for a precision query and mean success rate for the categorization task, divided for the different transformation are shown in the Tab. 3. In Fig. 12 the plot of the precision vs recall for the retrieval task is shown.

Transformation	S.R. Precision	S.R. Categorization
Isometry	0.77	1.0
Topology	0.44	0.99
Isometry + Topology	0.71	1.0
Noise	0.6	0.8
Null	0.86	1.0
Triangulation	0.58	1.0
Partially	0.68	1.0

Table 3 Success Rate (S.R.) for the precision queries with length equal to the category sizes and Success Rate of the categorization task for the Tosca dataset. The results are divided for type of transformation.

.

7.3 Timing

The entire pipeline is computationally efficient in each stage. 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 is computed in less than 5 seconds, of which $\sim 2.8s$ are necessary to extract all the seed regions, and $\sim 2.1s$ are needed to compute the entire hierarchical segmentation. Region descriptors are computed efficiently: on the average it takes $\sim 0.5s$ to extract all the four descriptors of a single region. As for the k-means clusterization, 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 $\sim 2s$.

8 Conclusions

In this paper the Bag-of-Words paradigm has been proposed for the 3D domain. The main steps of the involved processing pipeline have been carefully designed by focusing on both the effectiveness and efficiency.

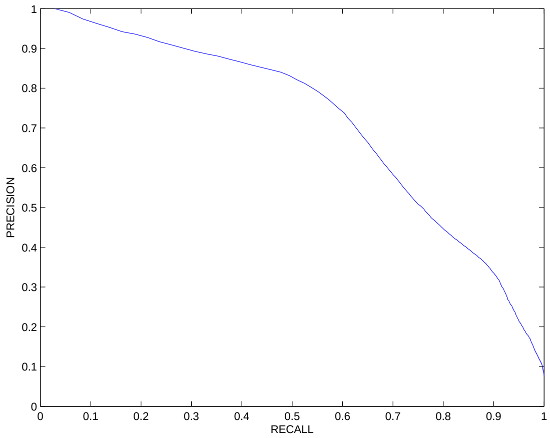


Fig. 12 Precision vs Recall for the Tosca dataset.

.

The Bag-of-Words approach fits naturally with subparts encoding by combining segment descriptors into several visual vocabularies. In this fashion, our methods is able to satisfy query models of composed objects. 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. Our framework is versatile in reporting satisfying performances for different applicative scenarios such as object retrieval, partial matching and shape categorization as shown in the comparison with other methods.

Acknowledgments

This paper was partially supported by PRIN 2006 project 3-SHIRT.

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) (2005) 509–530
2. Funkhouser, T., Kazhdan, M., M., P., Shilane, P.: Shape-based retrieval and analysis of 3D models. Communications of the ACM **48**(6) (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) 145–156
4. Funkhouser, T., Min, P., Kazhdan, M., Chen, J., Halderman, A., Dobkin, D.: A search engine for 3D models. ACM Transactions on Graphics **22** (2003)
5. Shilane, P., Funkhouser, T.: Selecting distinctive 3D shape descriptors for similarity retrieval. In: International Conference on Shape Modelling and Applications, IEEE Computer Society (2006)
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) 1–22
7. Grauman, K., Darrell, T.: The pyramid match kernel: Efficient learning with sets of features. Journal of Machine Learning Research **8**(2) (2007) 725–760
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., k. Osada, 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. Lin, X., Godil, A., Wagan, A.: Spatially enhanced bags of words for 3d shape retrieval. In: ISVC '08: Proceedings of the 4th International Symposium on Advances in Visual Computing. Volume 5358., Springer-Verlag (2008) 349–358
12. Shalom, S., Shapira, L., Shamir, A., Cohen-Or, D.: Part analogies in sets of objects. In: Eurographics Workshop on 3D Object Retrieval. (2008)
13. Gal, R., Shamir, A., Cohen-Or, D.: Pose-oblivious shape signature. IEEE Transaction on Visualization and Computer Graphics **13**(2) (2007) 261–271
14. 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) (2007) 470–482
15. Duda, R., Hart, P., Stork, D.: Pattern Classification. second edn. John Wiley and Sons (2001)
16. Hoffman, D.D., Richards, W.A.: Parts of recognition. Cognition (1987) 65–96
17. Shamir, A.: A survey on mesh segmentation techniques. Computer Graphics Forum (2008)
18. 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 (2006) 7
19. Petitjean, S.: A survey of methods for recovering quadrics in triangle meshes. ACM Computing Surveys **34**(2) (2002)
20. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22**(8) (2000) 888–905
21. Belongie, S., Malik, J.: Matching with shape contexts. Content-based Access of Image and Video Libraries, 2000. Proceedings. IEEE Workshop on (2000) 20–26
22. Burges, C.: A tutorial on support vector machine for pattern recognition. Data Mining and Knowledge Discovery **2** (1998) 121–167
23. 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)
24. Salton, G., M.McGill.: Introduction to modern information retrieval. McGraw Hill (1983)
25. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Trans. Inf. Syst. **20**(4) (2002) 422–446