

Online learning for automatic segmentation of 3D data

Federico Tombari, Luigi Di Stefano, Simone Giardino

Abstract— We propose a method to perform automatic segmentation of 3D scenes based on a standard classifier, whose learning model is continuously improved by means of new samples, and a grouping stage, that enforces local consistency among classified labels. The new samples are automatically delivered to the system by a feedback loop based on a feature selection approach that exploits the outcome of the grouping stage. By experimental results on several datasets we demonstrate that the proposed *online learning* paradigm is effective in increasing the accuracy of the whole 3D segmentation thanks to the improvement of the learning model of the classifier by means of newly acquired, unsupervised data.

I. INTRODUCTION AND PREVIOUS WORK

Fostered by the increasing availability of accurate and low-cost 3D sensors, a relevant research topic within the computer vision and robotic communities consists in analysis of 3D data. Nowadays, many 3D sensors, based on different techniques (e.g. laser scanners, Time-of-Flight cameras, stereo cameras, ..) are able to compute range maps at a very high frame-rate. In this paper, we address the task of automatically segmenting the sensed 3D data within a pre-defined set of objects or object categories. Such a task, usually referred to as semantic segmentation, has been significantly investigated over the past decade in the case of 2D images, whilst it represents quite novel an issue as regards 3D data, with a few very recent methods proposed in literature [1]–[4]. In fact, so far algorithms related to 3D data analysis have been mostly aimed at computing similarities between surfaces (e.g. [5]–[7]), while segmentation of 3D data among objects or object categories is currently regarded as a particularly challenging open issue, for it requires the ability of handling previously unseen shapes and/or views and to assign correct labels to each region of the scene under analysis.

3D data segmentation is usually dealt with by a machine *learning* approach, i.e. a classifier is first trained off-line in a supervised manner to learn the distinctive shape characteristics of each object or object class, then it is employed on-line to classify the points belonging to the scene that has to be segmented. This approach is motivated by the fact that the generalization ability of common classifiers holds the potential to deal with object deformations, previously unseen vantage points of a given objects as well as with the intra-class variance of each object category. In addition,

F. Tombari is with DEIS, University of Bologna, Bologna, Italy
federico.tombari@unibo.it

L. Di Stefano is with DEIS, University of Bologna, Bologna, Italy
luigi.distefano@unibo.it

S. Giardino is with DEIS, University of Bologna, Bologna, Italy
simone.giardino@studio.unibo.it

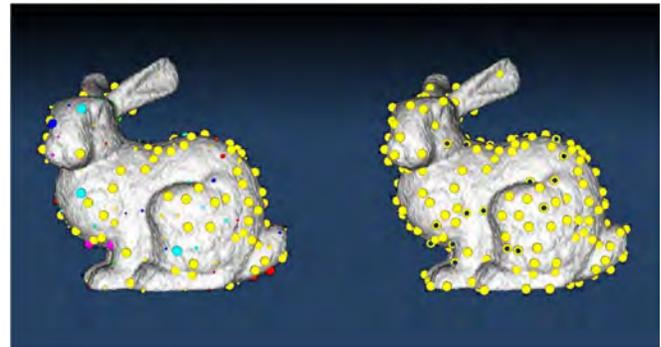


Fig. 1. Some classified feature labels (left) are modified after applying local consistency by a MRF (right): can some of these modified labels (indicated by a black dot) be deployed to improve segmentation of forthcoming data ?

the most recent literature proposals have also tried to exploit the statistical dependence among labels associated to neighboring points, so as to deploy the reasonable practical assumption that neighboring points share always the same label except at object boundaries. This approach requires the use of specific classification techniques usually referred to as *collective classification*. State-of-the-art classification approaches used for segmentation of 3D data [1]–[3], [8]–[10] rely on Associative Markov Networks (AMNs) [11], a variant of the popular discriminative Conditional Random Field (CRF) model. These approaches achieved good performance in automatic segmentation of generic object classes for outdoor and indoor Lidar data, although a main limitation is that the AMN formulation solves a linear classification problem, hence can hardly deal with articulated shapes or large intra-class variations.

In a different approach [12], standard *local* classifiers are first employed to perform a rough classification, successively a grouping stage is applied, that enforces local consistency between classified labels through a Markov Random Field (MRF) formulation over a graph built around the classified features. This approach has the benefit to handle any general-purpose classification technique, hence can be used with non-linear classifiers or with kernels specifically designed to match the data characteristics. The results reported in [12] show that this method is effective with popular local classifiers (Boost, SVM, kNN) on datasets characterized by highly articulated shapes. Also, in [12] it is shown that this method can effectively deal with multiple cues simultaneously, such as color and 3D shape.

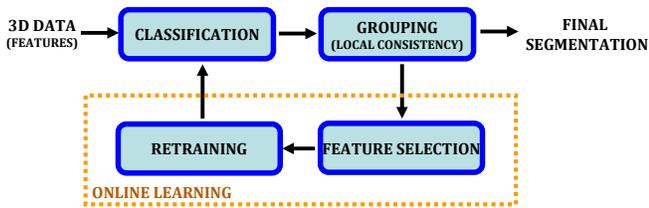


Fig. 2. The proposed approach introduces an online feedback to the learning engine that improves the classification model. This feedback is obtained by analyzing the outcome of the grouping stage that enforces local consistency into the output of the classifier.

II. PROPOSED APPROACH

This work builds upon [12] and aims at defining a method to improve classification by exploiting the data currently being processed in a totally unsupervised way. More specifically, we are interested in investigating on whether the behavior of the MRF-based grouping step can be exploited as a useful source of information to improve classification of forthcoming data. Figure 1 illustrates this concept. The figure shows on the left the “raw” labels assigned to the features of an object by a local classifier. Each color represents a different category, the radius spheres being proportional to the reliability of the class assignment provided by the classifier. It can be seen that many classification errors are present. The enforcement of local consistency by a MRF (shown on the right) significantly improves segmentation. Under this point of view, the grouping stage acts as an additional classification stage, since it performs a novel class assignment based on the initial configuration of the labels given by the local classifier. Hence, the main idea is to deploy the information concerning the features that were assigned to a different class during the grouping stage in order to improve the accuracy of the local classifier. This approach can be deployed in *online*, so as to continuously improve the learning model (hence the final 3D segmentation) by exploiting the information gathered from newly acquired data. Hence, after the initial supervised learning done once at run-time, for each newly processed scene the proposed approach can automatically select a subset of test data to be used for retraining on-line in a fully unsupervised manner the local classifier.

The deployment of new training data gathered on-line can be very useful in many robotic and computer vision applications, for example to deal with the intra-class variation of the learned categories by exploiting the information of objects previously unseen during training, or to include into the learning model the information regarding new vantage points of an object. Another potential benefit of the proposed approach deals with the case of a particularly small training set: the small amount of training data can act as a seed to propagate the unsupervised learning over the newly acquired data. The proposed 3D segmentation algorithm is sketched in Fig. 2. From the figure, it is possible to see the feedback loop that brings back information to the classification module from the outcome of MRF-based grouping stage. The main

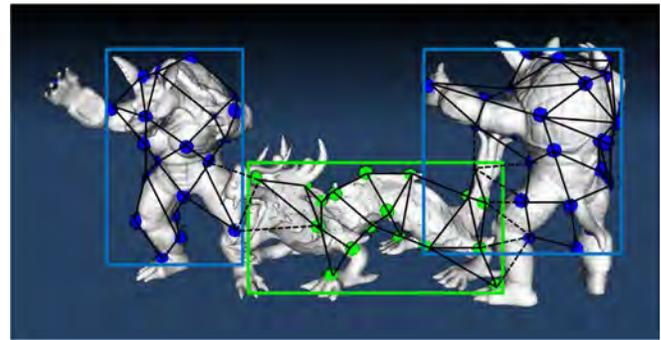


Fig. 3. Connected 3D components are detected in the classified graph.

stages of the algorithm (represented by the blocks in the figure) will now be briefly outlined.

A. Classification and grouping

Once a set of features has been extracted from the 3D data, it is classified by means of a standard classifier. Initially, the classifier is trained off-line by means of a training set. The features can be 3D features (i.e. *shape* features [5]–[7], [13]–[17]), together with additional texture features (e.g. [18], [19]). Successively, the grouping stage builds an undirected graph on the classified features and aims at minimizing an energy functional computed on this graph [12]:

$$E(X) = \sum_i \phi_i(x_i) + \sum_j \phi_{i,j}(x_i, x_j) \quad (1)$$

where x_i is the label assigned to node i and j is one of its neighbors according to a predefined set of cliques. This functional can be interpreted as the *energy* associated with each graph configuration X and is composed of two terms, the former relative to the probability of each feature to belong to a specific class (*unary* term, $\phi_i(x_i)$), the latter measuring the consistency of each feature label with its neighboring features (*pairwise* term, $\phi_{i,j}(x_i, x_j)$). The solution of this functional leads to a *np-hard* problem. By means of an approximated approach (Loopy Belief Propagation [20] was deployed in [12]) a local minimum \tilde{X} of the energy term can be determined in polynomial time:

$$\tilde{X} = \underset{X}{\operatorname{argmin}} E(X) \quad (2)$$

For details on how to build the feature graph and how to define the unary and pairwise terms in (1) the reader is referred to [12].

B. Feature Selection

The first step towards automatic selection of the features that most reliably can be deployed for the retraining procedure is based on the detection of the connected components of the 3D graph determined out of the segmentation stage. This can be useful, as it will be explained, to apply simple heuristics in order to select the subset of features used for retraining.

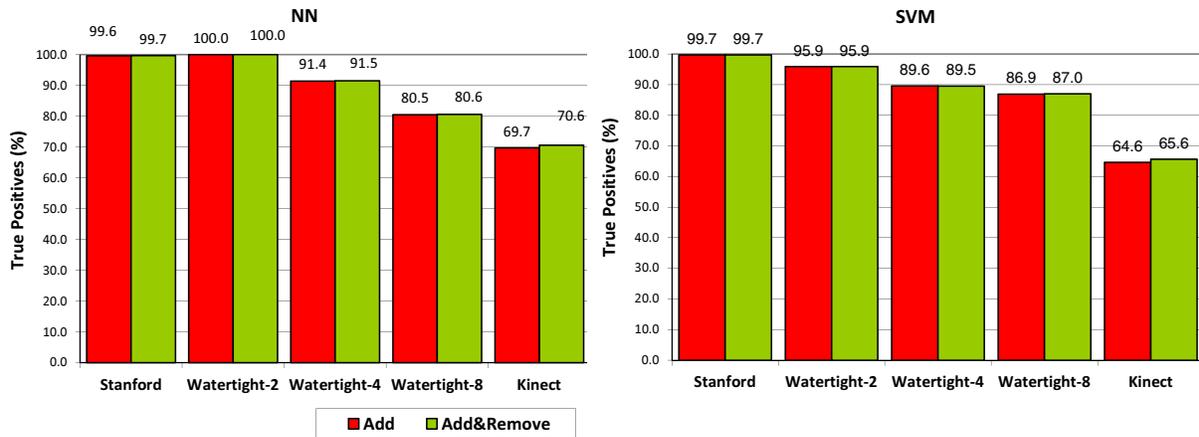


Fig. 4. The % of correctly chosen features yielded the proposed feature selection stage on the experimental datasets used in Section III and relatively to two different classifiers (SVM and NN).

To detect connected components in the scene we use the *breadth-first search* algorithm (BFS) [21], an efficient graph-search algorithm that detects a connected component starting from a seed node and iteratively exploring all the neighboring nodes. The first step of this algorithm consists in cutting all edges that connect two nodes having different labels. Once this is done, starting from any node we add all its neighbors to a First-In First-Out (FIFO) queue and mark them as examined. For all the points in the queue we repeat this procedure adding only the neighbors that have not been examined yet. Once the queue is empty, all the examined points will be included into the same connected component of the graph. To find the other connected components, the algorithm is repeated again, starting from a node that has not been examined yet. An example of the outcome of this stage is depicted in Fig. 3, where the the connected component algorithm is applied over a scene of one dataset used in section III.

Once the connected components have been found, we have to determine which nodes should be used for retraining among those whose class assignment was modified by the grouping stage. The two main heuristics deployed at this step are: i) nodes belonging to small connected components are discarded, since could be easily affected by noise and would hardly represent the surface of a whole object; ii) nodes too close to the border of the connected component are discarded, to avoid the typical boundary effects due to clutter. To enforce these two heuristics, we simply evaluate for each node: i) the minimum number of nodes of its connected component (in all our experiments, it is set to one fiftieth of the total number of nodes of the graph); ii) a minimum distance to the closest connected *bordering* node (in all our experiments, it is set to 1.5 the average distance among the nodes of the graph).

By enforcing these two simple conditions, we are able to determine reliable points that can be provided to the re-training stage. By continuously gathering new samples for the classifier, we incur into the disadvantage that the

training set is continuously incremented with new training features. While this has a marginal significance with certain classifiers such as SVM [22], where the addition of novel training features only changes the number and value of the support vectors, it can be more noticeable on others such as kNN [2], where all training features must be stored in memory and evaluated during the classification stage. In such a case, the addition of novel training features would tend to continuously increase memory occupancy as well as the computation time to search for the best k candidates.

For this reason, we have evaluated a *removal* procedure, dual to that aimed at selecting novel training samples, so as to automatically select a subset of features to be removed from the training set. The main idea is that features reliably detected as "mis-classified" could be discarded from the training set. More specifically, this procedure searches, for each sample selected for retraining, the most similar training sample of the misclassified category. If the similarity is higher than a pre-defined threshold, this sample is permanently discarded from the training set. This *removal* procedure only affects that subset of the training set that was added during the online stage in an unsupervised way, since, in principle, the initial supervised set should be considered reliable. Since applied to all samples used for re-training, this procedure allows us to discard, on the average, the same number of training samples as those added to the training set.

Although simple, the heuristics exploited for the feature selection stage are effective in selecting good candidates for the online learning stage. This is vouched by the charts in Figure 4, which evaluate the performance of the feature selection stage by reporting the number of correctly chosen features (*True Positives*, TP). In other words, since this stage can be seen as a binary classification problem, based on experiments with hand-labeled ground-truth it is possible to compute the number of features that were correctly picked up for retraining (i.e. misclassified by the classifier, correctly classified by the MRF and subsequently selected

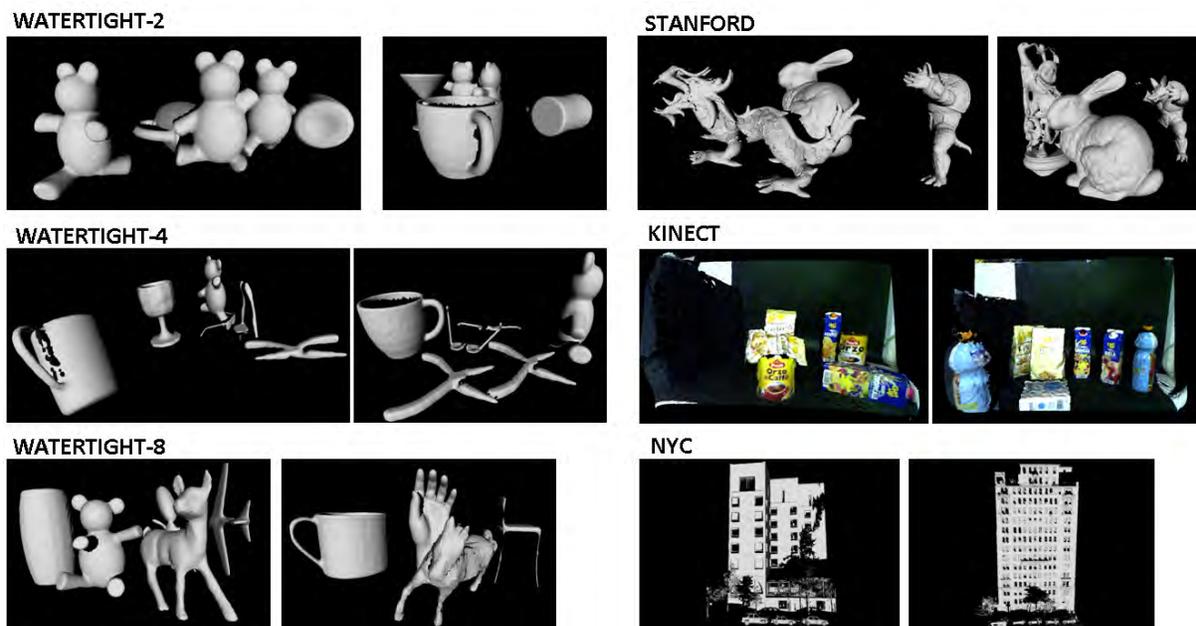


Fig. 5. Snapshots of scenes included in the considered datasets.

for retraining by the feature selection stage).

More specifically, the Figure shows TPs for two different classifiers (SVM and NN), each chart reporting the results yielded by the feature selection stage in the version that only adds retraining samples (referred to as *Add*, in red) as well as in the version that carries out also the *removal* procedure (referred to as *Add&Remove*, in green), on the 5 datasets that will be used in the experimental Section (see Section III). Since the charts only take into account those points that were selected by the algorithm, False Positives are not shown as they would simply be equal to $1 - TP$. From the Figure, it is evident that the great majority of the selected features are correctly chosen over all datasets. Also, it is interesting to note that the performance of the *Add* and of the *Add&Remove* approaches are almost equivalent on all datasets.

C. Retraining

Once a reliable set of features is selected for retraining, it is added to the old training set and used for a new supervised training stage of the classifier. To this purpose, although not deployed yet in our experiments, specific online learning techniques could be usefully deployed to render the overall approach more computationally efficient. For example, in the case of SVM [22] classifiers, the LASVM [23] algorithm may be deployed to rapidly add the new features as training samples for the classifier.

III. EXPERIMENTAL EVALUATION

In this section we present experimental results aimed at evaluating the effectiveness of the proposed approach for automatic segmentation of 3D scenes. We compare the proposed *online learning* approach to the original method (i.e. that proposed in [12]) on several 3D datasets, which

are now briefly introduced. As done in [12], to extract 3D features from the training set and test set we have used the 3D feature detector proposed in [7]. The only exception concerns dataset *Kinect* (see next subsection), since due to the large variation in size of the models belonging to the various classes the difference in the number of features extracted from the various classes by [7] tends to bias the classification towards bigger-sized categories: hence, with *Kinect*, a random sampling of feature points was used (as proposed e.g. in [6]). The extracted 3D features are then described by means of the well-known Spin Image descriptor [6]. The parameter values deployed for feature detection and description are reported in [12].

A. Datasets

The performance of the proposed segmentation algorithm is evaluated over several datasets. The first, referred to as *Stanford*, was already used for the experimental evaluation in [12]. The training set consists of 6 full-3D models ("Armadillo", "Asian Dragon", "Thai Statue", "Bunny", "Happy Buddha", "Dragon") taken from the *Stanford 3D Scanning Repository*¹, while the test and validation sets consists of 36 "2.5D" scenes which were obtained by rendering randomly chosen views of 6 "full 3D" scenes obtained by random rotations and translations of the models composing the training sets. In particular, out of these 36 scenes, 6 are selected for validation and 30 for testing. This is the only dataset that includes a validation set, since all parameters of the algorithms have been tuned on this dataset and then kept constant throughout the remaining experiments. This seems reasonable due to the purpose of our experimental evaluation:

¹<http://graphics.stanford.edu/data/3Dscanrep>

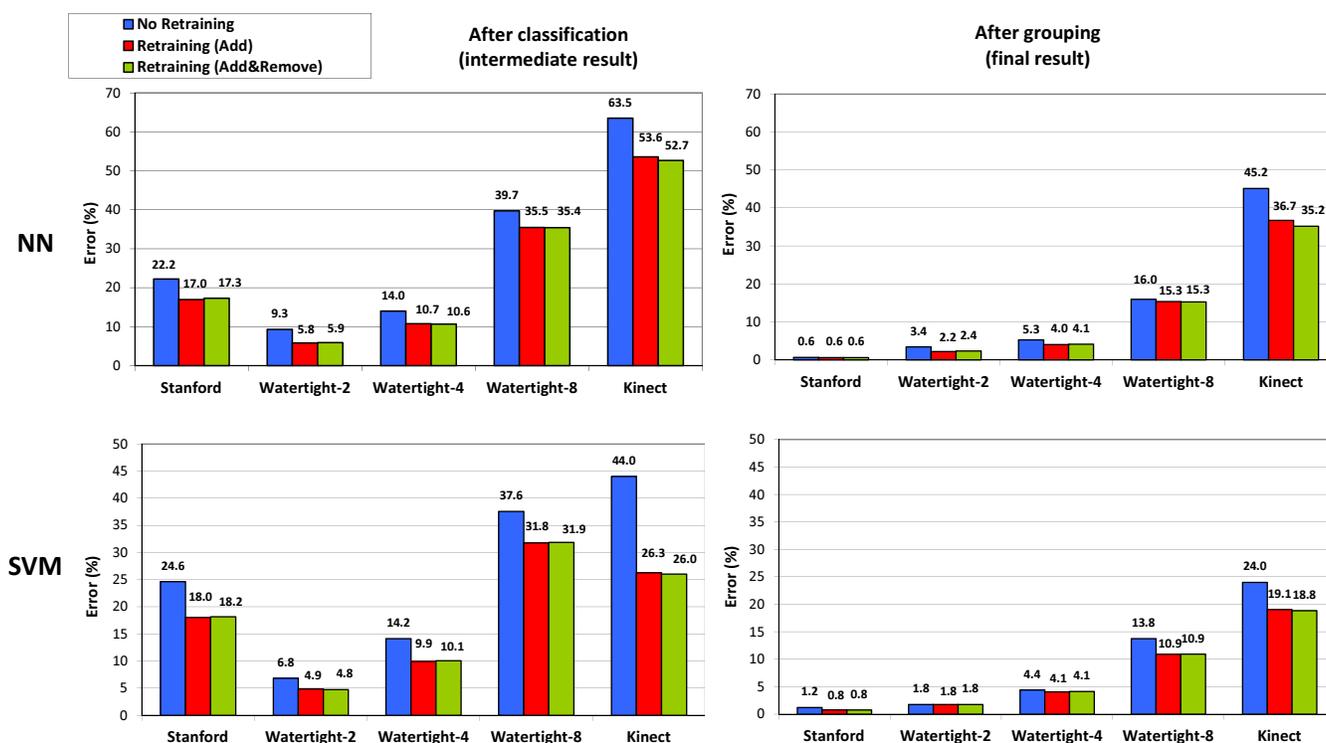


Fig. 6. Results in terms of recognition error evaluating the use of the proposed online learning scheme before (left) and after (center) the grouping stage. Two different classifiers are evaluated (SVM and NN).

since we wish to assess the benefit of the proposed online learning stage with respect to [12], the two methods should share exactly the same parameter values.

A total of 3 other datasets were created from the models available within the *Aim@Shape Watertight* database², which originally contained 400 models, subdivided into 20 categories of 20 models each. The usefulness of this dataset is that, given the amount of different models per each category, the test scenes can be built using models not included in the training set, hence the capabilities of the algorithms in generalizing to categorization and their robustness towards intra-class variations can be evaluated.

The first dataset, dubbed *Watertight-2*, comprises only 2 categories (*Teddy bears* and *cups*). Out of each category, 5 "full 3D" models were used for training and the remaining 15 for testing: using the approach deployed for the *Stanford* dataset, 6 "3D" scenes were created and 3 random "2.5D" views acquired for each scene, for a total of 18 test scenes. Similarly, two other datasets, dubbed *Watertight-4* and *Watertight-8* were created, including respectively 4 and 8 categories. The former includes 5 models of each category in the training set and 30 "2.5D" scenes in the test set, while the latter includes 10 models of each category in the training set and 50 "2.5D" scenes in the test set.

In addition, we acquired a novel dataset with the recently introduced Microsoft *Kinect* device. This device is able to acquire depth and color data (i.e. RGB-D) in real-time.

The dataset we built includes 5 (+ 1 for the background) categories of common grocery products such as packets of biscuits, juice bottles, coffee cans and boxes of salt, of different brands and colors. The training set includes 3 model views for each category, while the testing scenes are 16, including a high degree of clutter and occlusions. Thanks to the deployed device, this dataset includes both color and depth (although the color cue is not exploited in the experiments shown in this paper). Furthermore, by means of a simple approach based on background subtraction, we were able to automatically compute the ground truth (i.e. by assigning to each point of each scene the label of the class it belongs to), hence also this dataset can be used for quantitative evaluation. This dataset is publicly available for quantitative evaluation of 3D segmentation algorithms by other researchers³.

Finally, the performance of the proposed method is also evaluated on the recent *New York City* (NYC) dataset⁴, a publicly available dataset that includes urban data of the city of New York acquired with a Lidar sensor. Since the dataset only includes "raw" 3D data, we selected 3 main semantic classes (*facades*, *vegetation*, *vehicles*) and divided it into a training and a test set composed of, respectively, 5 models for each of the 3 classes and 11 scenes. Since the dataset does not include the ground-truth, regarding this particular dataset we can show here only qualitative results.

²www.aimatshape.net

³www.vision.deis.unibo.it/fede/kinectDataset.html

⁴3D Urban Data Challenge, www.3dimpvt.org/challenge.php

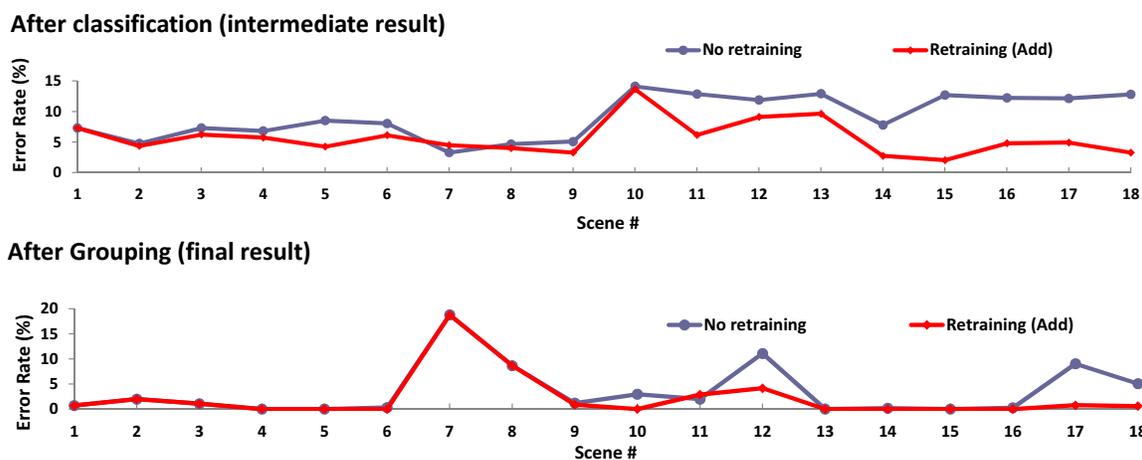


Fig. 7. The data selected for retraining allow for improving the segmentation performance throughout the dataset, both before (top) and after (bottom) the application of the MRF-based grouping stage. Note how, with the proposed online learning approach, the local classification performance (top) increasingly improves as long as more data is provided to the algorithm.

		Inferred Labels									
		humans	cups	fourlegs	airplanes	ants	chairs	teddies	hands	RECALL	
True Labels	Retraining (Add)	humans	207	0	125	0	1	1	0	2	0,62
	cups	0	1.500	5	4	0	31	1	7	0,97	
	fourlegs	29	2	695	0	20	0	2	21	0,90	
	airplanes	0	0	0	565	26	14	0	12	0,92	
	ants	0	4	52	8	548	10	1	2	0,88	
	chairs	3	0	0	20	50	295	14	17	0,74	
	teddies	1	0	2	0	1	21	846	2	0,97	
	hands	22	1	63	2	1	1	0	254	0,74	
	PRECISION		0,79	0,99	0,74	0,94	0,85	0,79	0,98	0,80	

		Inferred Labels									
		humans	cups	fourlegs	airplanes	ants	chairs	teddies	hands	RECALL	
True Labels	No Retraining	humans	205	0	108	0	1	1	0	21	0,61
	cups	0	1.440	5	5	0	88	1	9	0,93	
	fourlegs	10	2	665	0	39	0	2	51	0,86	
	airplanes	1	0	0	536	22	39	0	19	0,87	
	ants	1	4	53	8	505	28	26	0	0,81	
	chairs	3	0	0	30	35	311	13	7	0,78	
	teddies	1	10	0	0	1	21	835	5	0,96	
	hands	16	1	68	2	0	1	0	256	0,74	
	PRECISION		0,86	0,99	0,74	0,92	0,84	0,64	0,95	0,70	

Fig. 8. Confusion matrix for the *Watertight-8* dataset yielded by the segmentation algorithm using a SVM classifier with (top) and without (bottom) the use of the proposed online learning procedure.

Sample snapshots of the scenes included in each dataset are shown in Fig. 5.

B. Results

Fig. 6 shows the segmentation results with and without the deployment of the proposed online learning paradigm. Results refer to the use of two different classifiers (i.e. NN, on the top row, and SVM, on the bottom row). Also, results are shown for both the local classification stage (intermediate result, left column) and the final segmentation after MRF-

based grouping (right column). Each chart compares the classification error reported by the 3D segmentation algorithm without online learning (blue) and with online learning, the latter in the version that only adds retraining samples (*Add*, in red) as well as in the version that also exploits the *removal* procedure (*Add&Remove*, in green).

As it can be seen, the proposed learning approach always improves the overall result on each evaluated dataset, both in terms of the learning model adopted, which is able to improve the accuracy in the classification of the single feature, and in terms of the final result yielded by the algorithm after application of the MRF-based grouping. It is interesting to note that the *Add&Remove* procedure obtains equivalent results to those yielded by the *Add* approach, hence demonstrating that the samples that are selected in this stage can be safely removed since do not help improving the learning model. Also, by comparing the 3 *Watertight* datasets, it seems that the proposed approach is more effective with the SVM classifier if the dataset includes a higher number of categories, conversely with the NN classifier the improvement is higher if the dataset comprises a small number of classes. Finally, in terms of 3D data, the Kinect dataset is the most challenging one, reporting the poorest performance of the algorithms. This appears to be due to the fact that it is also the noisiest sensor among those evaluated. Conversely, Stanford dataset, which includes basically synthetic models, is the one leading to the most accurate final results.

In addition, Fig. 7 compares the recognition error yielded by the proposed online learning method against the original approach in [12] on each scene of dataset *Watertight-2*. From the figure, the behavior of the proposed online learning paradigm can be better evaluated. In particular, it can be noted that continuously deploying new samples for retraining increasingly improves the segmentation, both after the classification stage and after the grouping stage. Given these results, additional improvements yielded by the proposed

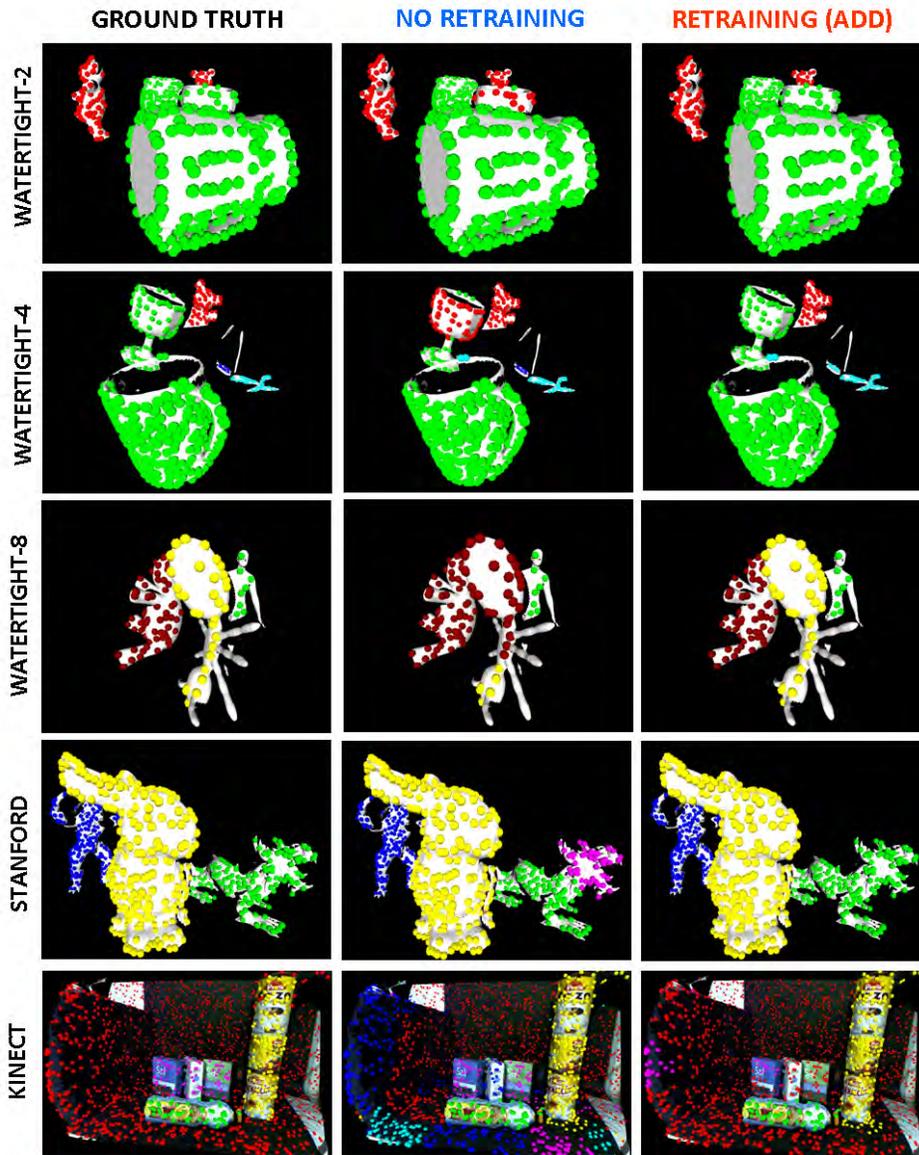


Fig. 9. Qualitative results: some details of the scenes belonging to the datasets with ground-truth used for testing.

method are expectable over longer 3D data sequences.

We can also evaluate in more detail the improvement brought in by the proposed method by analyzing, in Fig 8, the confusion matrix yielded on dataset *Watertight-8* by the proposed online learning and the original approach. For lack of space we can only show the results on one dataset, hence we have selected that including the highest number of classes and intra-class variation. As it can be seen from the Figure, the proposed approach is able to yield improvements over almost all categories, both in terms of *Precision* as well as *Recall*.

Finally, Fig. 9 shows some qualitative results yielded by the two algorithms over some scenes belonging to the considered datasets (all except *NYC*, shown in Fig. 10). The Figure shows also the ground truth. The colored spheres refer to the extracted 3D features, with each color representing a

different category. From the Figure it can be noted that the proposed approach is able to yield overall a more accurate segmentation of the 3D scenes. Fig. 10 shows qualitative results concerning the *NYC* dataset. In this case, the results are compared with the un-segmented scenes, due to the lack of ground-truth. The label colors are, respectively, red for *facades*, green for *vegetation* and blue for *vehicles*. As it can be seen, also in this case the proposed method allows a notable improvement in the segmentation accuracy over several surfaces of the dataset.

IV. CONCLUSIONS

This paper proposed a novel online learning method to be applied for improving segmentation of 3D scenes. The proposed approach is particularly suited for robotic and computer vision applications where one wants to improve the

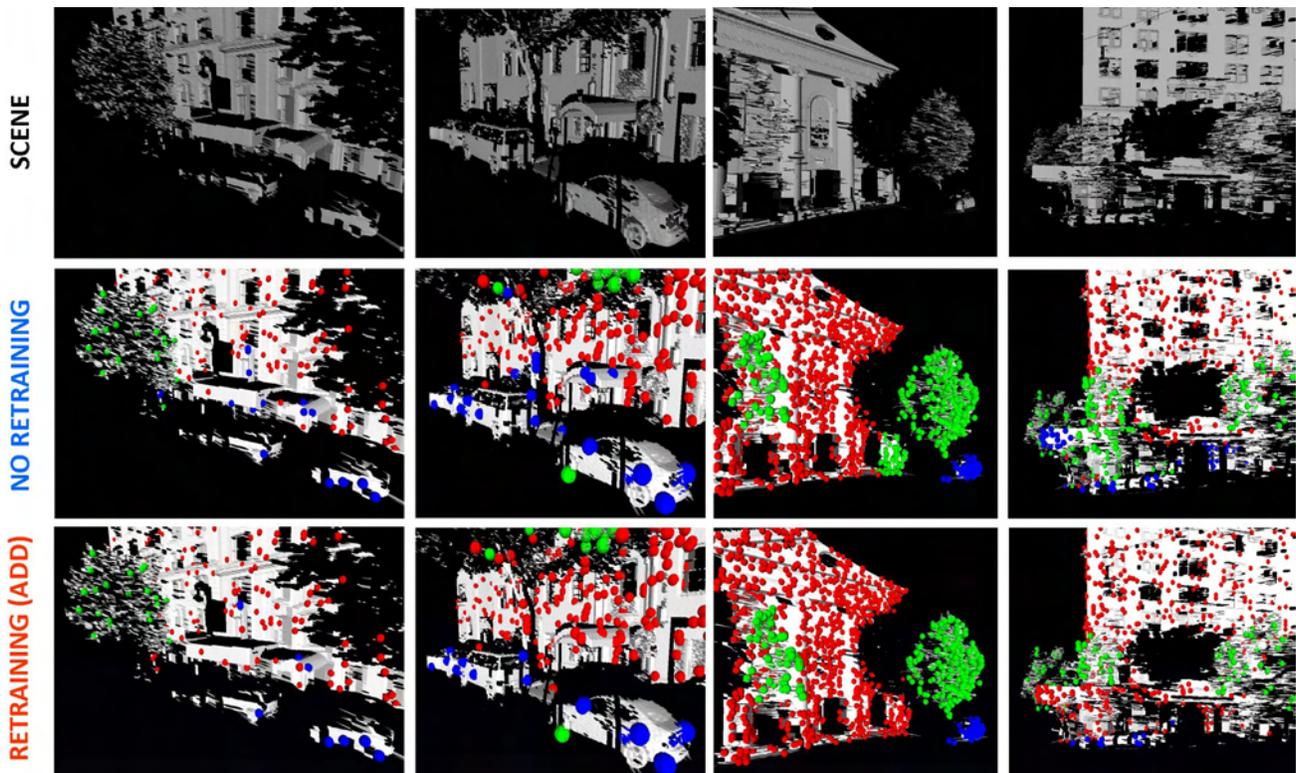


Fig. 10. Qualitative results: some details of the scenes belonging to the NYC dataset.

adopted learning model online without continuously fetching supervised data. The proposed method yielded accuracy improvements over several datasets relative to different 3D sensors (such as Lidar and *Kinect*). It also showed the capability of performing semantic segmentation, by correctly segmenting categories of objects and scene parts. In addition, we proposed a novel dataset with groundtruth for 3D segmentation, acquired with a *Kinect* sensor, which is now publicly available. Future developments include the deployment of the proposed approach in an online robotic application, so as to study the behavior of the method over long time intervals.

REFERENCES

- [1] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng, "Discriminative learning of markov random fields for segmentation of 3-d scan data," in *CVPR*, 2005.
- [2] R. Triebel, R. Schmidt, O. M. Mozos, and W. Burgard, "Instance-based amn classification for improved object recognition in 2d and 3d laser range data," in *Proc. Int. J. Conf. on Art. Intelligence*, 2007.
- [3] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *CVPR*, 2009.
- [4] O. Hadjilaidis and I. Stamos, "Sequential classification in point clouds of urban scenes," in *Proc. 3DPVT*, 2010.
- [5] A. S. Mian, M. Bennamoun, and R. A. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *IJCV*, vol. 89, no. 2-3, pp. 348–361, 2010.
- [6] A. Johnson and M. Hebert, "Using spin images for efficient object recognition in cluttered 3d scenes," *PAMI*, vol. 21, no. 5, pp. 433–449, 1999.
- [7] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *ICCV-WS:3DRR*, 2009.
- [8] R. Triebel, K. Kersting, and W. Burgard, "Robust 3d scan point classification using associative markov networks," in *ICRA*, 2006.
- [9] D. Munoz, N. Vandapel, and M. Hebert, "Directional associative markov network for 3-d point cloud classification," in *3DPVT*, 2008.
- [10] —, "Onboard contextual classification of 3-d point clouds with learned high-order markov random fields," in *ICRA*, 2009.
- [11] B. Taskar, V. Chatalbashev, and D. Koller, "Learning associative markov networks," in *ICML*, 2004.
- [12] F. Tombari and L. Di Stefano, "3d data segmentation by local classification and markov random fields," in *Proc. 3DIMPVT 11*, 2011. [Online]. Available: www.vision.deis.unibo.it/fede/papers/3dim11.pdf
- [13] H. Chen and B. Bhanu, "3d free-form object recognition in range images using local surface patches," *Pattern Recognition Letters*, vol. 28, no. 10, pp. 1252–1262, 2007.
- [14] J. Novatnack and K. Nishino, "Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images," in *ECCV*, 2008, pp. 440–453.
- [15] C. S. Chua and R. Jarvis, "Point signatures: A new representation for 3d object recognition," *IJCV*, vol. 25, no. 1, pp. 63–85, 1997.
- [16] A. Frome, D. Huber, R. Kolluri, T. Bülow, and J. Malik, "Recognizing objects in range data using regional point descriptors," in *ECCV*, vol. 3, 2004, pp. 224–237.
- [17] F. Tombari, S. Salti, and L. Di Stefano, "Unique signatures of histograms for local surface description," in *ECCV*, 2010.
- [18] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004.
- [19] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *CVIU*, vol. 110, no. 3, pp. 346–359, 2008.
- [20] J. Kim and J. Pearl, "A computational model for combined causal and diagnostic reasoning in inference systems," in *Proc. 8th Int. J. Conf. on Artificial Intelligence (IJCAI)*, 1983, pp. 190–193.
- [21] S. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Prentice Hall, 2010.
- [22] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [23] A. Bordes, S. Ertekin, J. Weston, and L. Bottou, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, pp. 1579–1619, 2005.