3D data segmentation by local classification and Markov Random Fields

Federico Tombari
DEIS-ARCES
University of Bologna
Bologna, Italy
federico.tombari@unibo.it

Luigi Di Stefano
DEIS-ARCES
University of Bologna
Bologna, Italy
luigi.distefano@unibo.it

***Abstract*—Object segmentation in 3D data such as 3D meshes and range maps is an emerging topic attracting increasing research interest. This work proposes a novel method to perform segmentation relying on the use of 3D features. The deployment of a specific grouping algorithm based on a Markov Random Field model successively to classification allows at the same time yielding automatic segmentation of 3D data as well as deploying non-linear classifiers that can well adapt to the data characteristics. Moreover, we embed our approach in a framework that jointly exploits shape and texture information to improve the outcome of the segmentation stage. In addition to quantitative results on several 3D and 2.5D scenes, we also demonstrate the effectiveness of our approach on an online framework based on a stereo sensor.**

***Keywords*-3D segmentation; feature classification; MRF; 3D computer vision**

I. Introduction and previous work

The increasing availability of 3D sensors is fostering research on computer vision and machine learning techniques aimed at processing range maps and 3D meshes. An important task in this field deals with automatic classification of the data gathered by a 3D sensor (such as e.g. a laser scanner, a Time-of-Flight camera or a stereo camera) within a predefined set of objects or object classes. This task is key for several important applications such as scene understanding, robot localization and navigation, map registration, robot manipulation and grasping.

The problem of object categorization and semantic segmentation has been addressed mainly in the 2D field (i.e. for images), while so far most algorithms related to analysis of 3D data have been aimed at computing similarities between surfaces, i.e. *surface matching*. In such a field, building upon successful research on image features, the approach based on detection, description and matching of local 3D features is gaining increasing popularity [1]–[10]. On the other hand, 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. In principle, the problem may be tackled in a conventional *supervised* way, i.e. employing a classifier to learn off-line the local shape characteristics of each object or object class and then achieving segmentation by classification of the points belonging to newly acquired scenes. In fact, classifiers' generalization skills hold the potential to deal with object deformations, previously unseen vantage points of a given objects as well as, in the case of semantic segmentation, with the intra-class variance of each object category.

However, the most recent literature proposals try 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*. In particular, state-of-the-art classification approaches used for segmentation of 3D data [11]–[16] rely on Associative Markov Networks (AMNs) [17], a variant of the popular discriminative Conditional Random Field (CRF) model. Although AMNs have been shown to outperform linear SVMs for 3D scene classification [11], [14], one major issue of the former method is represented by the notably high computational cost of the training stage, which relies on quadratic programming optimization. To deal with this aspect, more efficient approaches using subgradient optimization [14] or adaptive training data reduction [12] have been proposed. Recently, the use of high-order models has also been proposed [15], [16], while [18] presents a refinement of the AMN formulation based on deploying additional information concerning the global shape of the objects to be classified. A different approach is represented by the use of an Hidden Markov Model (HMM) formulation specifically conceived to classify points among three categories (vegetation, horizontal non-vegetation, vertical non-vegetation) [19].

Generally speaking, a main limitation of above referenced methods is that the AMN formulation solves a linear classification problem, thus resulting in labels that tend to be concentrated in the same area but still yielding a high misclassification errors due to data underfitting. In turn, [13] shows that the AMN approach is overall outperformed by a much simpler NN classifier. Moreover, the proposed AMN-based methods are characterized by an intrinsic slow inference also at testing time [15], which coupled with formulations that attempts to classify each individual 3D

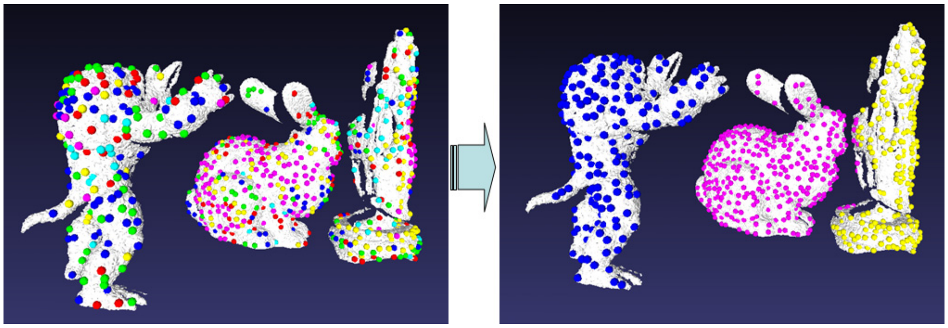

Figure 1. Initial segmentation by local classification of 3D features (left), final result after enforcing local consistency by the MRF (right)

point results in a dramatic computational complexity of the segmentation process. The computational issue is unlikely to be dealt with easily in a general way within the AMN framework, though very specific methods capable of providing a fast approximate solution have been devised [15].

In this paper we propose a general framework for 3D segmentation which, in principle and as most of the above mentioned methods, can handle segmentation of 3D data into rigid/deformable object instances as well as object categories. Our method differs from most recent proposals for it employs standard, local classifiers rather than collective classifiers and enforces local consistency between classified labels through a Markov Random Field (MRF) formulation. Thanks to this approach, and unlike the AMN formulation, it is possible to rely on standard general-purpose classification techniques, that render our method more flexible and more general. In particular, non-linear classifiers can be adopted seamlessly so to be less prone to data underfitting. In addition, kernel-based classifiers may also be employed, allowing the choice of the kernel to match the data characteristics. An additional advantage of the proposed approach is that the training time can be significantly reduced by choosing a suitable classifier (e.g. a NN classifier, should one wish to minimize such a time). In fact, the MRF algorithm does not need to be trained and hence it does not introduce any overhead with respect to the training of the local classifier. This compares favorably to collective classifier techniques that generally require a time-consuming and memory intensive training stage [15].

Another key original element of our proposal consists in the deployment of local 3D features to attain segmentation of the salient keypoints within the 3D data only, unlike all the above referenced proposal that instead are aimed at classifying each individual 3D point. In fact, we believe that in many applications a computationally efficient sparse segmentation can be a viable and attractive alternative to the dense but slow segmentation typically provided by collective classifiers such as AMN. An example of the sparse segmentation achievable by our method is shown in Fig. 1, with classified 3D features denoted as small colored spheres.

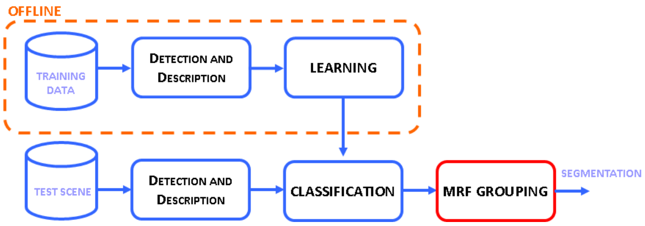

Figure 2. Flow diagram of the proposed 3D Segmentation algorithm.

Finally, as an additional contribution, we propose to jointly exploit, when available, multiple cues. In particular, we show how to seamlessly embed shape and texture cues into our framework and how this significantly improves the final segmentation. It is worth pointing out that all of the above referenced methods can deal with the shape cue only. Instead, we argue that a multi-cue framework will become increasingly relevant in the near future since several sensing technologies available today allow associating accurately registered texture information to range maps. For instance, the novel *Kinect* camera by Microsoft is an example of a particularly cost-effective device that would allow for synergistic deployment of depth and texture.

II. Description of the framework

This Section briefly outlines the proposed framework for 3D data segmentation. The flow diagram sketching the algorithm pipeline is shown in Figure 2.

First of all, we assume a *learning* stage, performed *offline*, aimed at training a multi-class classifier by means of labelled training data. As the first step, 3D feature detection and description is carried out on each 3D model to obtain descriptions of salient 3D keypoints. The use of a detector also brings in benefits for what concern the robustness of the classification. In fact, since the main goal of a detector is to find highly-repeatable distinctive features, the classifier only needs to focus over salient 3D points rather than trying to model also less discriminative shapes that are more prone to be in common with several classes.

Optionally, a dimensionality reduction step (such as PCA) may follow feature description to reduce the dimensionality of description vectors. Successively, extracted descriptors are fed to a multi-class classifier as training data.

During the *on-line* stage, 3D feature detection and description is performed on the 3D scene to be segmented. Then, feature points are classified based on their descriptors. If dimensionality reduction is used, it is applied previously to classification. Next, labeled features are fed to the MRF-based algorithm, which attains the final segmentation starting from the initial labeling provided by the local classifier.

As it will be explained in Section III, the MRF algorithm requires each initial class label to come together with a reliability - or probability - score concerning that particular label assignment. This value, hereinafter denoted as $\nu(x_i)$ (x spanning all possible class labels and i being an index associated with each feature, see next Section), ranges in the unit interval: the closer to 1, the higher the confidence of the classifier in choosing that label assignment, this confidence measure being quite easy to obtain from the output of most classifiers.

In the experimental results presented in Section IV, our approach is evaluated with three widely used classifiers, i.e. Support Vector Machines (SVM) [20], Boosting [21] and NN [13] based on the Euclidean distance *d*. As for SVM, we use as reliability term the multi-class probability estimates computed according to the method described in [22] and implemented in *libSVM* [23]. As far as Boosting is concerned, we have deployed the binary-class implementation included in the OpenCV library [24]. First, we have extended it to the multi-class case using a one-versus-one (*OvO*) methodology [25]. Then, each probability estimate $\nu(x_i)$ is computed as the ratio between classifiers that voted for class x_i over all classifiers concerning class x_i. Finally, for the NN classifier the reliability computed for a feature i assigned to class x is obtained as:

$$\nu(x_i) = 1 - \frac{1}{2} \cdot \frac{d_{x_i}}{\tilde{d}_{x_i}} \tag{1}$$

where d_{x_i} and $\tilde{d}_{x_i}$ denote the Euclidean distance between the feature i and, respectively, its closest and second-closest training feature. Then, all other classes different from x are given a reliability equal to $1 - \nu(x_i)$.

A. Use of multiple cues

Several 3D sensors integrate the information on reconstructed surfaces with information on the appearance of the scene. This is the case, e.g., of stereo cameras, of most TOF cameras as well as of certain devices based on the projection of coded light, such as the new *Kinect* camera. Appearance information can represent a valuable additional cue to discriminate between different objects when objects belonging to different classes show, locally, similar surface shapes. For this reason, we aim at including seamlessly texture information into our 3D segmentation framework. Thus, when available, we extract and describe also 2D features from the pixel lattice associated with the range map using the well-known SURF method [26]. A specific classifier is then trained with the extracted "2D features". During the online stage, classified 2D features are added to the 3D features to form a super-set of 2D-3D features, each associated with a specific label class. This feature super-set is then fed to the MRF algorithm described in the next Section.

III. MRF FORMULATION FOR 3D OBJECT SEGMENTATION

Markov Random Field theory stems from a formalization of the Ising-Potts model for particle interaction. Let us assume there is a set of nodes, S within an undirected graph Ω. In the specific case of 3D object segmentation, each node represents a feature extracted from the 3D data. As pointed out in Section II-A, should also texture information be available, the super set of features would consist of the union of the 2D and the 3D feature sets. Each node i is denoted with a label $x_i \in C$, C being the finite discrete set of possible object or categories in which the scene could be subdivided into. We will refer here to the specific case of Pairwise MRF, i.e. only cliques of size 2 are taken into consideration.

In order to define the graph on which to apply the MRF formulation, we need to draw arcs between features in the 3D space so to define, for each node i, a set of neighbors $N(i)$. To this purpose, we have considered two different approaches. The first, referred to as *k-Nearest Neighbor* (kNN) [27], connects to each node of the graph its closest k features, the distance given by the L_2 norm (Euclidean distance). It is easy to infer that, at the end of the graph building process, each node might have more than k neighbors, since it might be a k-NN for additional nodes other than its k neighbors. The other approach, referred to as ϵ-ball [27], defines as neighbors of a node all the features falling within a sphere of a specific radius centered at the node. To avoid isolated nodes, at least one neighbor is selected for each node. Based upon preliminary experiments, we have selected the ϵ-ball technique, since in our experiment it has shown to perform overall better than the kNN approach. Though not yet investigated in our work, another possible approach would have been a Minimum Spanning Tree (MST) [28] applied not to all points of the mesh but only to the subset of feature points.

According to the Hammersley-Clifford theorem, the posterior probability associated with the labeling configuration $X = \{x_i \in S\}$ takes the form of a Gibbs distribution:

$$P(X) = \frac{1}{Z} \prod_{i \in S} e^{-\phi_i(x_i)} \prod_{i \in S} \prod_{j \in N(i)} e^{-\phi_{i,j}(x_i,x_j)} \tag{2}$$

where Z is a normalization constant (*partition* function), $e^{-\phi_i(x_i)}$ is the *likelihood* term associated with node i having the label x_i, and $e^{-\phi_{i,j}(x_i,x_j)}$ is the prior term for the two neighboring nodes i and j of having label x_i and x_j. These two latter terms are usually referred to with different names, such as *unary* term, *evidence* or *fidelity* the former and *pairwise* term, *compatibility* or *regularization* the latter. The dependence of the prior term only from the neighbors of i is due to the Markov property.

Hence, according to the above *Maximum-a-Posteriori* (MAP) formulation, the optimal configuration $\tilde{X}$ is that maximizing the posterior probability:

$$\tilde{X} = \underset{X}{\operatorname{argmax}}\, P(X) \tag{3}$$

From a computational complexity point of view, it is more convenient to minimize the log of (2):

$$\tilde{X} = \underset{X}{\operatorname{argmin}}\, E(X) \tag{4}$$

where

$$E(X) = \sum_i \phi_i(x_i) + \sum_i \phi_{i,j}(x_i, x_j) \tag{5}$$

E(X) can be interpreted as the *energy* associated with each configuration X and is composed of two terms, corresponding to the above mentioned unary and pairwise terms.

In our specific case, the evidence has to be correlated with the reliability, previously referred to as $\nu_(x_i)$, assigned to each class label by the classification stage. Since the energy must be minimized, $\phi_i(x_i)$ has to be in the form of a cost, thus the evidence term is defined as:

$$\phi_i(x_i) = \lambda(1 - \nu(x_i)) \tag{6}$$

where λ is a regularization parameter that weights the importance given to the evidence with respect to the compatibility. Hence, the evidence ranges between 0 and λ, being minimum when the reliability is maximum and viceversa. As for the compatibility, we propose a definition derived from the Potts model [29]:

$$\phi_{i,j}(x_i, x_j) = \begin{cases} 0 & \text{if } x_i = x_j \\ e^{\frac{-||p_i - p_j||_2}{\sigma_c}} & \text{otherwise} \end{cases} \tag{7}$$

where p_i represents the 3D coordinate vector assigned to node i. Hence, the compatibility term is null if two neighboring nodes are given the same label and, in general, it assigns a penalty that is inversely proportional to the Euclidean distance between the two nodes. More specifically, the penalty function is defined as an inverse exponential regulated by parameter σ_c.

Since the minimization of (5) over an undirected graph turns out to be a NP-hard problem, an approximated method has to be employed. When the order of the cliques is 2, an algorithm that has been shown to yield good convergence to the global minimum is Loopy Belief Propagation (BP) [30]. Graph Cuts (GC) [31] is another popular approach, though it is also not guaranteed to converge to the global minimum when the cardinality of C is higher than 2. In our approach, we have employed a standard implementation of the BP algorithm to apply regularization over the undirected feature graph. The algorithm iteratively applies BP-based message-passing for a pre-defined number of iterations. A threshold on the derivative of the total energy $E(X)$ may also be used as a alternative termination criterion for the BP iterative process.

IV. Experimental results

This Section proposes experimental results aimed at validating the capabilities of the proposed approach to effectively segment 3D data into different objects. We divide the Section into two main parts. First, we show results aimed at assessing quantitatively on different 3D datasets both the effectiveness of the proposed MRF-based grouping approach and the usefulness of employing a 3D feature detector. In addition, we show qualitative results concerning an online segmentation framework based on stereo data and simultaneous deployment of shape and texture cues.

Throughout our experiments we have used the 3D feature detector proposed in [3], referred to here as *ISS*, and the well-known Spin Image descriptor [6].

Table I
TUNED PARAMETER VALUES FOR DATASETS *Stanford-3D* AND *Stanford-2.5D*. EXCEPT FOR PARAMETER "W" OF SPIN IMAGES, ALL PARAMETERS ARE IN UNITS OF AVERAGE MESH RESOLUTION.

	ISS detector		Spin Images	
Validation	Support	Non-max. Radius	W	b
Stanford 3D - SVM	6	3	15	3
Stanford 3D - Boost	6	5	15	3
Stanford 3D - NN	6	5	15	2
Stanford 2.5D- SVM	5	2	25	1
Stanford 2.5D - Boost	6	3	20	1
Stanford 2.5D - NN	4	2	30	1

A. Quantitative evaluation

As for these experiments, we evaluate the performance of the proposed approach on 2 different datasets. The first, recalled as *Stanford-3D*, has been created using 3D models taken from the *Stanford 3D Scanning Repository* [1]. The training set consists of 6 full-3D models ("Armadillo", "Asian Dragon", "Thai Statue", "Bunny", "Happy Buddha", "Dragon") taken from the repository. Then, 45 scenes have been built up by randomly rotating and translating different subsets of the model set so as to create clutter[2]. Out of these 45 scenes, 10 have been randomly selected for validation, while the remaining 35 have been used for testing. The second dataset, dubbed *Stanford-2.5D*, shares with the previous one the same 3D models as training set. However, 36 2.5D scenes are then obtained by rendering randomly chosen views of 6 of the scenes belonging to the previous

[1] http://graphics.stanford.edu/data/3Dscanrep

[2] 3 sets of 15 scenes each, containing respectively 3, 4 and 5 models

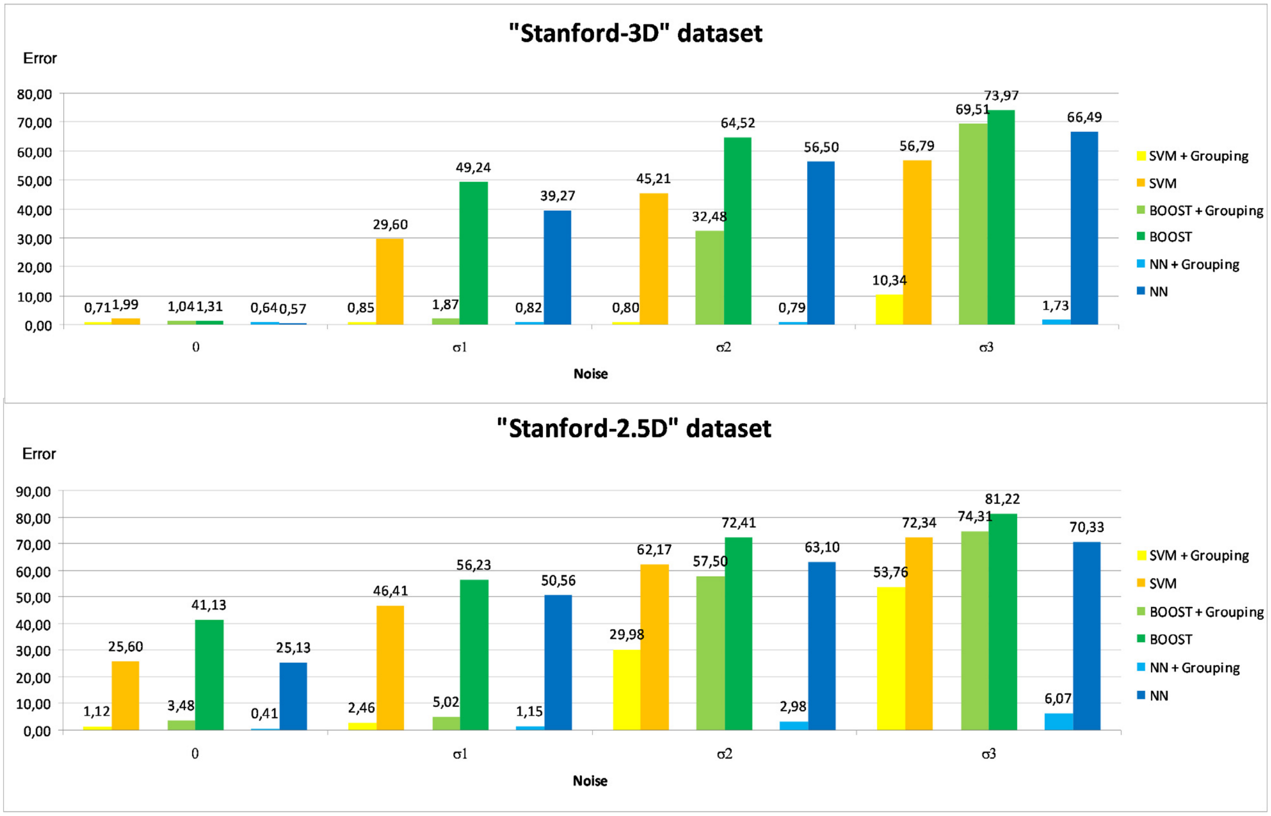

Figure 3. Quantitative results of the proposed approach using different classifiers over *Stanford-3D* and *Stanford-2.5D* datasets

dataset. Out of these 36 scenes, 6 are selected for validation and 30 for testing. A sample scene within the *Stanford-2.5D* dataset was depicted in Fig. 1. The tuned parameter values for the ISS detector and the Spin Images descriptor used for these two datasets are reported in Table I. The parameters not specified in the table were set to the default values proposed in the original papers [3], [6].

In order to render the evaluation more realistic and challenging, similarly to the methodology proposed in [27] we add Gaussian noise with increasing standard deviation, namely σ_1, σ_2 and σ_3 at respectively 10%, 20% and 30% of the average mesh resolution computed on the training set.

Figure 3 aims at evaluating the effectiveness of the proposed MRF-based grouping algorithm, and shows the classification error before and after the application of the grouping process over classified 3D features for both datasets and at different levels of noise. More specifically, each chart report the mean recognition error averaged over all scenes of the evaluated dataset. In order to prove the fact that our grouping approach is effective over classified data of different nature, in this experiment we have considered three different classifiers: SVM, Boost, and NN. As shown by the figure, the grouping approach is always capable of improving the classification results, that is with all classifiers and at all noise levels. More importantly, in many cases it is able to dramatically decrease the recognition error by turning a poor classification (i.e. $> 50\%$) into an accurate one (i.e. $< 10\%$). Overall the whole algorithm shows good robustness to occlusions and noise: e.g. in the NN case, it yields very low error rates (1.73, 6.07 respectively on the two datasets) even with high noise levels (σ_3).

Figure 4 evaluates the usefulness of a 3D detector for training and classification of 3D data. In particular, it compares the performance of the proposed approach, which deploys the feature detector described in [3], with that yielded by the same pipeline in which however the feature detector is substituted by randomly sampling interest points in both the training and testing data. As pointed out by the Figure, when the classification error is below 50%, the use of a feature detector always outperform the use of random sampling. The only case when this does not happen is with the *Stanford-2.5D* dataset and high noise levels (σ_2, σ_3), where Boost and SVM provide notably poor performance and instead a NN approach turns out to be the best classification method. Indeed, with NN and high noise levels the use of a feature detector significantly improves classification results with respect to random selection of interest points.

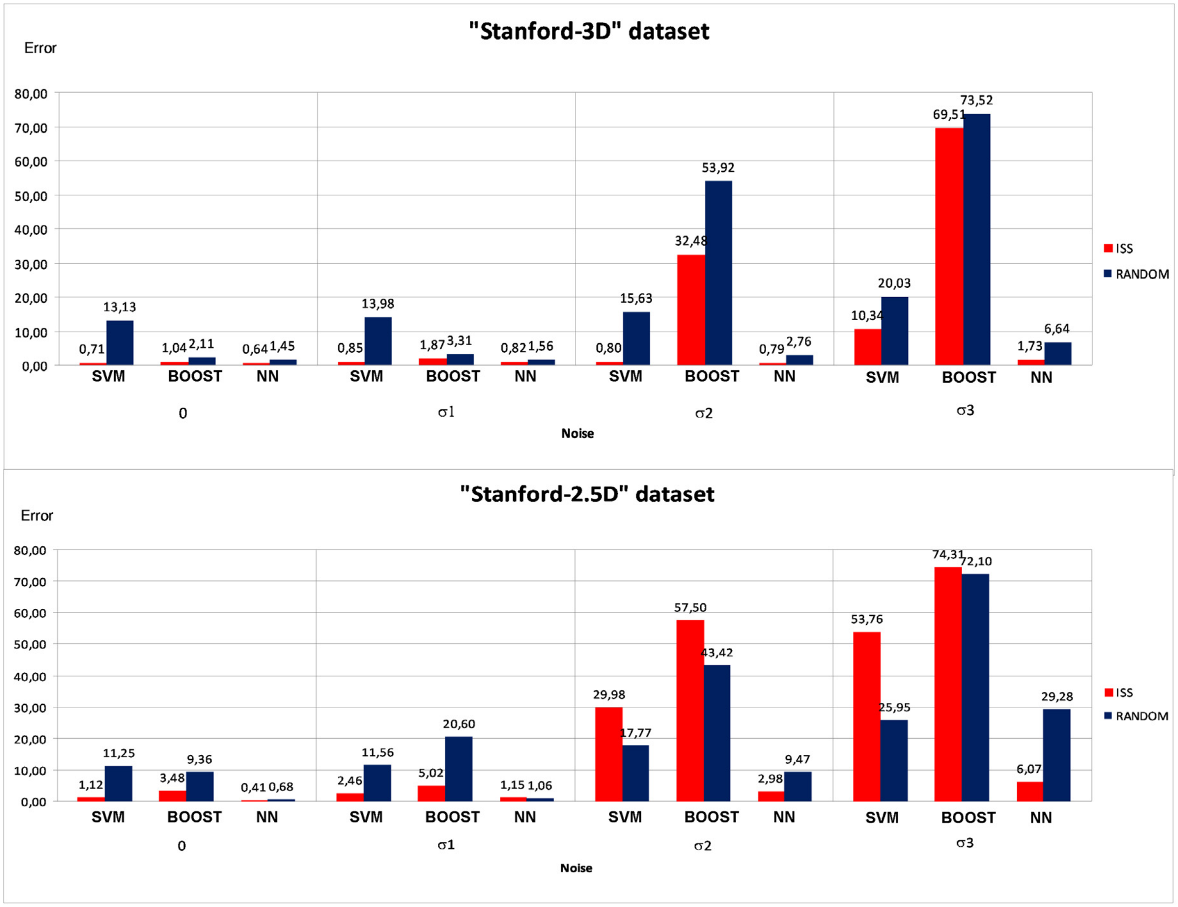

Figure 4. Quantitative results of the proposed approach comparing different feature extraction modalities over *Stanford-3D* and *Stanford-2.5D* datasets

B. *Qualitative evaluation*

We present here a qualitative performance analysis of our method based on data obtained from a stereo vision sensor. Since this data includes both shape and texture, we deploy the methodology outlined in Subsection II-A to include both cues in our segmentation framework. As for the stereo setup, we use 2 calibrated off-the-shelf webcams to get rectified image pairs, and a standard block-based dense stereo algorithm to compute the range maps. Given the well-known limitations of standard dense stereo matching algorithms within low textured regions, we use an off-the-shelf projector to augment the scene with a random, black-and-white pattern. In addition, to obtain training data we rely on a simple background subtraction approach algorithm and insert one object at a time in the scene in order to automatically get labeled training data of each class.

Results are provided in Fig. 5. In this experiment we use 5 classes (4 toy objects and 1 background class, shown on the leftmost column, top, together with the colors used to denote the different classes) and we train our data with different frontal and lateral views of the objects (approximately 10 range maps per class are acquired). Then, several scenes, shown in the column next to the leftmost one, are built up by randomly placing the objects on a table so as to cause occlusions and clutter. Classification results are displayed in the remaining four columns as colored spheres, each color denoting a class. In particular, from left to right, we show the 3D classification, the 2D classification, the 2D+3D feature merging, and, in the rightmost column the final results attained by the MRF-based grouping stage. As it can be seen, despite the challenging scenario the proposed grouping approach is capable of dramatically improving the quality of both 2D and 3D classification. Overall the proposed framework yields a reliable segmentation of the 3D data provided by the stereo sensor.

To provide an indication of the challenging scenario our approach is able to cope with, the figure also reports, in the bottom of the leftmost column, some sample disparity maps attained by the standard block-based stereo algorithm on the acquired scenes. As it may be expected, many object parts are missing due to occlusions and disparities turn out inaccurate (especially at object borders) and noisy.

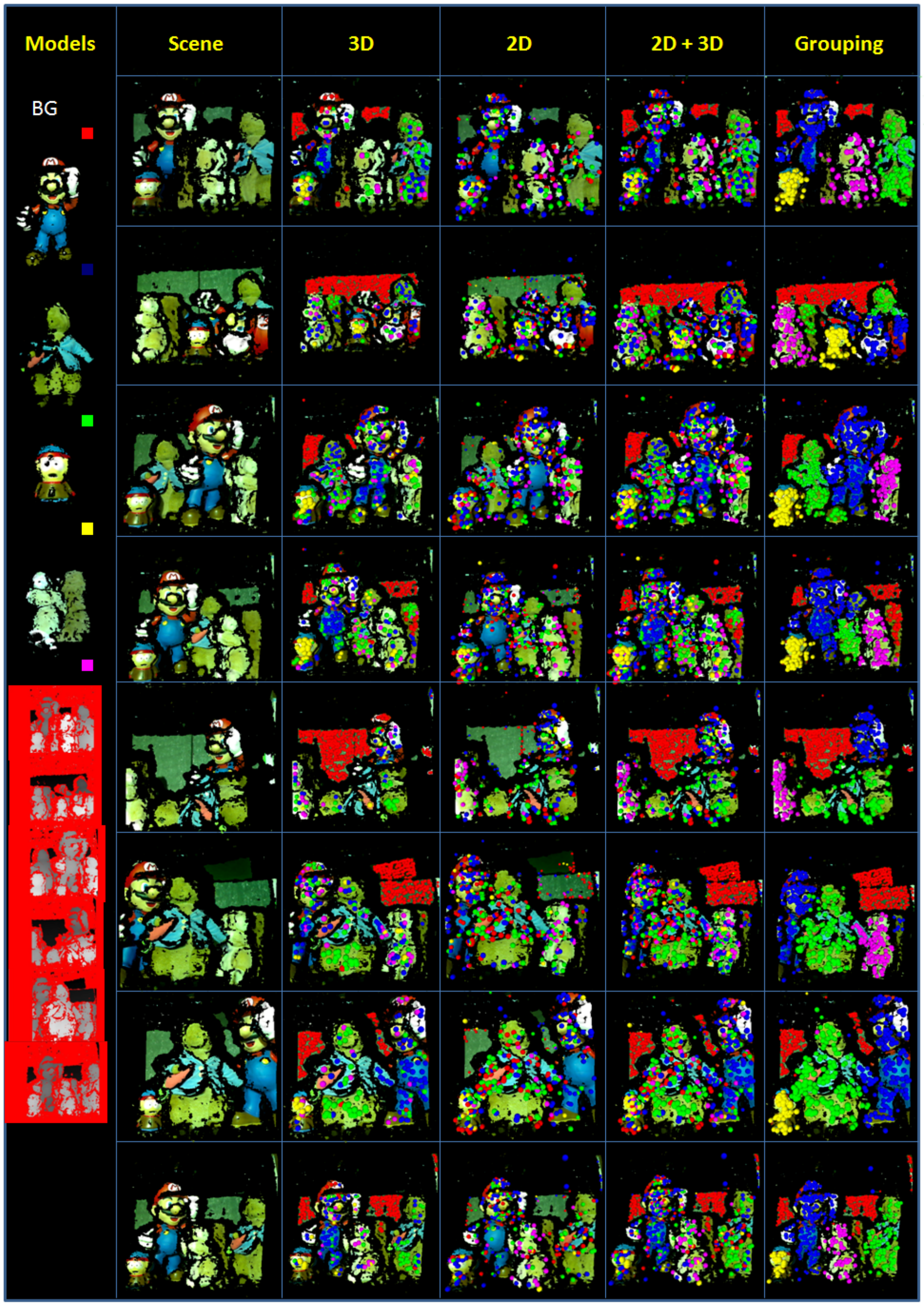

Figure 5. Qualitative results of the proposed segmentation approach with stereo data acquired from a pair of webcams.

Additional qualitative results achieved by our method on this challenging online stereo-based framework are included in the supplementary material.

From the computational point of view, in the current implementation the full pipeline of the proposed approach requires approximately 25-30 seconds per scene, including stereo acquisition and matching (30%), 2D and 3D feature detection and description (35%), classification (5%) and MRF-based grouping (30%).

V. Conclusion

Results provided on standard benchmark data as well as on a more challenging on-line stereo-based scenario have demonstrated the validity of the proposed contributions. The use of a MRF-based grouping stage following local classification of 3D feature descriptions allows robust segmentation of 3D data into different objects by notably reducing the error that would be yielded by using only the initial classification stage. Also, seamless introduction of multiple cues in our framework has shown to provide significant benefits. Extending the validation of the method to semantic segmentation scenarios, in particular considering outdoor data obtained with 3D sensors such as Lidars or laser scanners is one major direction of our future research.

References

[1] A. Mian, M. Bennamoun, and R. Owens, "On the repeatability and quality of keypoints for local feature-based 3d object retrieval from cluttered scenes," *Int. J. Computer Vision*, p. to appear, 2009.

[2] 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.

[3] Y. Zhong, "Intrinsic shape signatures: A shape descriptor for 3d object recognition," in *Proc. 3DRR Workshop (in conj. with ICCV)*, 2009.

[4] J. Novatnack and K. Nishino, "Scale-dependent 3d geometric features," in *Proc. ICCV*, 2007.

[5] ——, "Scale-dependent/invariant local 3d shape descriptors for fully automatic registration of multiple sets of range images," in *Proc. ECCV*, 2008.

[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] C. Chua and R. Jarvis, "Point signatures: a new representation for 3d object recognition," *IJCV*, vol. 25, no. 1, pp. 63–85, 1997.

[8] 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.

[9] F. Stein and G. Medioni, "Structural indexing: Efficient 3-d object recognition," *PAMI*, vol. 14, no. 2, pp. 125–145, 1992.

[10] A. Mian, M. Bennamoun, and R. Owens, "A novel representation and feature matching algorithm for automatic pairwise registration of range images," *IJCV*, vol. 66, no. 1, pp. 19–40, 2006.

[11] 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 *Proc. CVPR*, 2005.

[12] R. Triebel, K. Kersting, and W. Burgard, "Robust 3d scan point classification using associative markov networks," in *Proc. ICRA*, 2006.

[13] 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.

[14] D. Munoz, N. Vandapel, and M. Hebert, "Directional associative markov network for 3-d point cloud classification," in *Proc. 3DPVT*, 2008.

[15] ——, "Onboard contextual classification of 3-d point clouds with learned high-order makov random fields," in *Proc. ICRA*, 2009.

[16] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert, "Contextual classification with functional max-margin markov networks," in *Proc. CVPR*, 2009.

[17] B. Taskar, V. Chatalbashev, and D. Koller, "Learning associative markov networks," in *Proc. Int. Conf. on Machine Learning*, 2004.

[18] A. Agrawal, A. Nakazawa, and H. Takemura, "Mmm-classification of 3d range data," in *Proc. ICRA*, 2009.

[19] O. Hadjiliadis and I. Stamos, "Sequential classification in point clouds of urban scenes," in *Proc. 3DPVT*, 2010.

[20] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[21] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. EuroCOLT*, 1995, pp. 23–37.

[22] T. Wu, C. Lin, and R. Weng, "Probability estimates for multi-class classification by pairwise coupling," *J. Machine Learning Research*, vol. 5, pp. 975–1005, 2003.

[23] www.csie.ntu.edu.tw/~cjlin/libsvm.

[24] http://opencv.willowgarage.com.

[25] U. Kreel, "Pairwise classification and support vector machines," in *Advances in Kernel Methods: Support Vector Learning*, S. A. Schlkopf B, Burges CJC, Ed. Cambrige, MA, USA: MIT Press, 1999, pp. 255–268.

[26] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[27] R. Unnikrishnan and M. Hebert, "Multi-scale interest regions from unorganized point clouds," in *CVPR Workshop on Search in 3D*, 2008.

[28] M. . Carreira-perpin and R. S. Zemel, "Proximity graphs for clustering and manifold learning," in *Neural Information Processing Systems (NIPS)*, 2005, pp. 225–232.

[29] R. Potts, "Some generalized order-disorder transformations," *Proc. Cambridge Phil. Soc.*, vol. 48, 1952.

[30] 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 83)*, 1983, pp. 190–193.

[31] V. Kolmogorov and R. Zabih, "What energy functions can be minimized using graph cuts?" *Trans. PAMI*, 2002.