

# Unsupervised Range Image Segmentation and Object Recognition Using Feature Proximity and Markov Random Field

Dario Lodi Rizzini, Fabio Oleari, Andrea Atti, Jacopo Aleotti and Stefano Caselli

**Abstract** In this paper, we propose a framework for unsupervised range image segmentation and object recognition that exploits feature similarity and proximity as leading criteria in the processing steps. Feature vectors are distinctive traits like color, texture and shape of the regions of the scene; proximity of similar features enforces classification and association decisions. Segmentation is performed by dividing the input point cloud into voxels, by extracting and clustering features from each voxel, and by refining such segmentation through Markov Random Field model. Candidate objects are selected from the resulting regions of interest and compared with the models contained in a dataset. Object recognition is performed by aligning the models with the refined point cloud clusters. Experiments show the consistency of the segmentation algorithm as well as the potential for recognition even when partial views of the object are available.

## 1 Introduction

The diffusion of cheap and relatively accurate 3D sensors has popularized scene interpretation and point cloud processing. Motion planning, human-robot interaction, manipulation and grasping [1] have taken advantage from these advancements in perception. In particular, identification of objects in a scene is a fundamental task when the robot operates in unpredictable human-populated environments.

Different formulations of this problem depend on the availability of shape or color data as well as on specific prior knowledge about the setup or the object. Object detection and recognition are commonly achieved by extracting features that represent a signature for a point neighborhood. We can roughly distinguish between

---

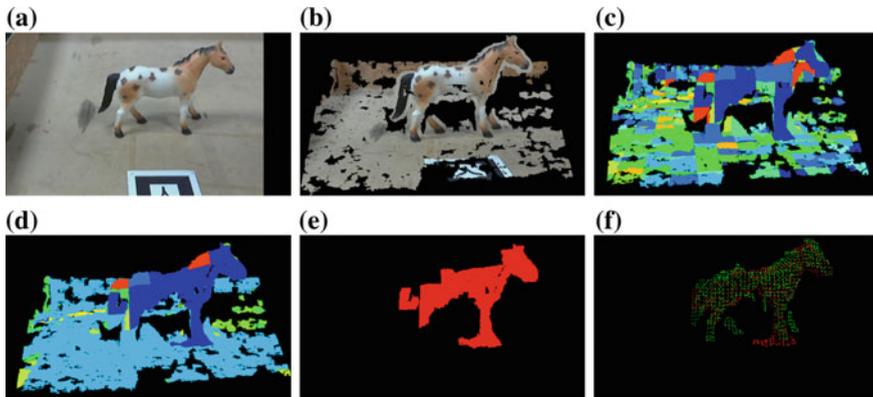
D. Lodi Rizzini (✉) · F. Oleari · A. Atti · J. Aleotti · S. Caselli  
RIMLab-Robotics and Intelligent Machines Laboratory, Dipartimento di Ingegneria dell'Informazione, University of Parma, Parma, Italy  
e-mail: dlr@ce.unipr.it

F. Oleari  
Elettric80 S.p.a., Viano, Italy

point feature descriptors [2–6] and simple geometric or visual features extracted from local patches [7]. The classification of point neighborhoods (e.g. to label the point neighborhoods belonging to the same object) may take into account both their relative position in space and the similarity of their features. The similarity of two feature vectors is measured by their distance, and accordingly two point neighborhoods are similar if their corresponding features are close in feature space. Thus, *feature similarity* and *proximity* in Euclidean space are the two leading criteria in scene interpretation problems.

Graphical model frameworks have been used to encode neighborhood relations and to label nodes through potential functions in 3D object detection problems [8–11]. Each node of a graphical model corresponds to a point neighborhood, which may correspond in turn to a voxel, another type of cell, or a patch. The classification of each node is initially based only on its features and then refined by weighting the results of neighbor cells. These techniques produce a satisfactory segmentation of the scene when the chosen label set is consistent with the observation. For example a label should be defined for every region of interest (ROI) with homogeneous features each roughly corresponding to an object or a class of objects. In many works the initial classification is performed by supervised classifiers that associate features to labels defined a priori. Unsupervised detection algorithms are more general, but the label set must be assessed from feature distribution so that candidate objects can be distinguished avoiding over segmentation. However, labels are usually computed according to a single criterion, either by clustering items in feature space (similarity) or by region growing from a randomly chosen seed to the similar neighbors (proximity).

In this paper, we present a complete system that performs unsupervised range image segmentation and object recognition by jointly exploiting feature similarity and proximity. Figure 1 provides an outline of the algorithm steps. The point cloud representing the observed scene is divided and indexed by octree voxel cells; features related to color, texture and shape are extracted from each voxel. An original contribution of this work is a region growing algorithm that estimates the number of feature classes by visiting neighbor voxels. All the voxels are visited starting from a seed recursively expanding over unvisited neighbors; then the algorithm builds a set of centroids in feature space. A new centroid in feature space is added only if a very different feature is found. The  $k$ -means algorithm initialized with these centroids detects clusters in feature space and is applied to attach a label to each voxel. Since the initial centroids of  $k$ -means algorithm have been chosen according to both feature similarity and spatial adjacency, the spatial distribution of the resulting labels is concentrated into groups even before the application of a graphical model method. The Markov Random Field method further reduces segmentation discontinuities. Each connected cluster of points with the same label is a potential candidate object. Clusters are further refined by splitting weakly connected regions and then may be merged and selected as candidate objects according to their size or relative position in the scene. Finally, clusters representing candidate objects are compared with a model dataset of point clouds corresponding to objects to be recognized. Object recognition is performed through alignment. An initial alignment is achieved by matching Fast Point



**Fig. 1** The steps of the range image segmentation and object recognition algorithm: the original image (a) and the corresponding point cloud (b), the segmented cloud before (c) and after Markov Random Field optimization (d), the selected cluster (e) and the alignment with the best matching model (f)

Feature Histogram features between the candidate object and each model. Alignment is refined by iterative closest point (ICP) technique. The distance between the two aligned point clouds is measured by the percentage of matching points between two point clouds.

The paper is organized as follows. Section 2 reviews the state of the art in range sensing for object detection and recognition focusing on tridimensional features and graphical models for segmentation. Section 3 illustrates the algorithms for range image segmentation, candidate object selection and object recognition. Section 4 presents the experiments performed to assess the effectiveness of the approach, and Sect. 5 discusses the results.

## 2 Related Works

Scene interpretation has been addressed operating on different scale and environments (outdoor, room-level indoor, etc.), setups (manipulators, mobile robots, etc.), input sensor data (images, RGBD data, point cloud, etc.). Feature extraction is a preliminary operation common to most detection and recognition techniques and is often applied to achieve segmentation, object detection and recognition.

Several 3D features to be extracted from point clouds or other representations have been proposed during the years. Spherical harmonic invariants [2] are computed on parametrized surfaces as values invariant to translation and rotation of such surfaces. Spin images [3] are obtained by projecting and binning the object surface vertices on the frame defined by an oriented point on the surface. Curvature map method [4] computes a signature based on curvature in the neighborhood of each

vertex. More recently, point feature descriptors like Point Feature Histogram (PFH) and Fast Point Feature Histogram (FPFH) [5] have been proposed. FPFH are computed as histograms of the angle between the normal of a point and the normals of the points in its neighborhood. Several features have been proposed and implemented in *Point Cloud Library* (PCL) [6]. These methods usually provide a parameter vector that describes the local shape. Such descriptors allow object recognition of known objects by matching a model and the observed point cloud.

Other algorithms operate on patches or voxels of the point cloud instead of extracting local descriptors in the neighborhood of each point [12]. Straight scene segmentation may be achieved either by investigating 3D points distribution [13] or using simple features like color, planarity, etc. from each cell or voxel. The method described in [14] detects smooth surface patches in the range image and combines these low-level segments into high-level object segments.

A graphical model framework is usually applied to refine the labels or to improve classification under the hypothesis that spatial neighbors in the dataset tend to have similar labels. The graphical model category includes Conditional Random Fields (CRF), Associative Markov Networks (AMN) and Markov Random Fields (MRF). In [8] segmentation and classification are performed simultaneously on laser range data using AMN. The minimization of an energy function generates uniform labels among neighbors and enables the estimation of best separating hyperplanes in feature space for classification. The clusterization method proposed in [15] is based on features and MRF. This method associates a label for each point and defines the MRF graph on the neighborhood obtained from the mesh built from the point cloud. A minor similarity with our work lies in the update procedure of the centroids after the fusion of an incoming cloud with the cumulated one. Graphical models may also be used to learn models from feature point descriptors as shown in [9]. The segmentation method described in [10] uses local classifiers that operate on shape signature and spin images and MRF to improve local consistency. Herbst et al. perform object discovery using the images collected from several views and applying MRF [16]. Object detection may also be performed using classifiers trained to recognize an object from a specific view and MRF to enforce dependency among neighbor voxels [11].

### 3 Object Detection and Recognition

This section illustrates the algorithms developed in this work to detect and recognize objects in a depth image. The depth image is initially partitioned into voxels and each voxel is described by a feature vector. Feature vectors are partitioned into clusters (each associated to a label) according to their similarity, but also taking into account the spatial distribution of the corresponding cells. The principle that spatial neighbors are likely to belong to the same cluster is exploited to refine the labels with MRF. Objects are selected from the cluster labels according to cluster description and attention-based criteria. Finally, the system checks whether each

selected object corresponds to a target object belonging to a given dataset. All these steps are discussed in the next sections.

### 3.1 Scene Segmentation

The segmentation algorithm is designed to detect homogeneous regions of interest (ROI) of the point cloud. These regions correspond to candidate objects or other recognizable background entities. The cloud is partitioned into voxels whose size is a trade-off between resolution and information content of each voxel. Each voxel corresponds to a cell of the space grid subdivision. A large voxel size corresponds to a coarse subdivision of the range image, but each voxel contains more points and several descriptive features. In the experiments reported in the paper, the voxel side length is 2 cm.

Formally, let  $\mathcal{P}$  be the input point cloud and  $\mathcal{V} \subset 2^{\mathcal{P}}$  be the set of the cells.  $\mathcal{V}$  is a partition of  $\mathcal{P}$ , i.e.  $\cup_{j=1}^{|\mathcal{V}|} v_j = \mathcal{P}$  and  $v_{j1} \cap v_{j2} = \emptyset$ . With an abuse of notation  $v : \mathcal{P} \rightarrow \mathcal{V}$  is also the map that associates each  $p_i \in \mathcal{P}$  to its container cell  $v(p_i) = v_j$ . Each point  $p_i$  consists of its Cartesian coordinates  $p_{i,pos}$  w.r.t. a specific frame (e.g. the sensor reference frame) and of the color information  $p_{i,col}$ . A cell  $v_j$  has neighboring cells that are represented by its neighborhood set  $\mathcal{N}_j \subset \mathcal{V}$ . The cells used in this system are *voxels* indexed by an *octree* data structure which quickly accesses the neighbors. For neighbor sets on octree reflexivity is granted, i.e.  $v_k \in \mathcal{N}_j$  implies  $v_j \in \mathcal{N}_k$ .

The system performs scene segmentation by grouping similar voxels. Similarity between voxels is computed in term of distance between their corresponding features. In this work, a cell  $v_j$  is described by color, texture and shape and such information is encoded in a feature vector  $f_j = [f_{j,col}^T, f_{j,tex}^T, f_{j,shape}^T]^T$ .

- *Color features* The color features of the voxel  $v_j$  are computed by binning the HSV color components of its points. In particular, a two-dimensional color histogram is computed on hue (H) and saturation (S) components with  $10 \times 10$  bins and a unidimensional histogram is computed with 10 bins for value (V). Hence, color features contribute to the feature vector with  $|f_{j,col}| = 10 \times 10 + 10 = 110$  entries.
- *Texture features* Texture information is computed on the grey-scale image patches corresponding to the cells after the application of a set of Gabor filters. We have combined 24 filters which are obtained by using 6 orientations and 4 scales. A 10 bin histogram is obtained from each of the 24 images. Hence, texture components contribute with  $|f_{j,tex}| = 6 \times 4 \times 10 = 240$  feature components.
- *Shape features* The shape of each cell is defined by several parameters: the 3 normalized eigenvalues  $l_1 \geq l_2 \geq l_3$  of the point coordinate covariance matrix, the curvature index defined as the ratio  $l_3 / \sum_{i=1}^3 l_i$ , the two-dimensional histogram of  $10 \times 10$  bins computed on the two polar coordinates of the normals. Thus, the shape is described by  $|f_{j,shape}| = 3 + 1 + 10 \times 10 = 104$  parameters.

**Algorithm 1: Initial Clustering**


---

**Data:**  $\mathcal{P}$ : point set;  $\mathcal{V}$ : cell set;  $\alpha$ : confidence interval;  
**Result:**  $\mathcal{C}$ : centroids in feature space;

```

1  $\bar{\mathcal{V}} \leftarrow \mathcal{V}$ ; /* set of unvisited cells */
2  $\mathcal{Q} \leftarrow \emptyset$ ; /* queue of cells to be visited */
3  $\mathcal{C} \leftarrow \emptyset$ ; /* centroid list */
4 while  $\mathcal{Q} \neq \emptyset$  or  $\bar{\mathcal{V}} \neq \emptyset$  do
5   if  $\mathcal{Q} \neq \emptyset$  then
6     | pop  $v_j$  from  $\mathcal{Q}$  and remove it from  $\bar{\mathcal{V}}$ ;
7   else
8     | extract an unvisited node  $v_j$  from  $\bar{\mathcal{V}}$ ;
9   end
   /* updates the nearest centroid */
10  compute feature vector  $f_j$  for  $v_j$ ;  $d_i = \min_{s \in \mathcal{C}} (f_j - \mu_s)^T \Sigma_s^{-1} (f_j - \mu_s)$ ;
11  if  $d_i < \chi_{\alpha, d}^2$  then
12    |  $\hat{\mu}_i \leftarrow \mu_i$ ;
13    |  $\mu_i \leftarrow \frac{n_i \mu_i + f_j}{n_i + 1}$ ;
14    |  $\Sigma_i \leftarrow \frac{n_i \Sigma_i + n_i \hat{\mu}_i \hat{\mu}_i^T + f_j f_j^T - (n_i + 1) \mu_i \mu_i^T}{n_i + 1}$ ;
15    |  $n_i \leftarrow n_i + 1$ ;
16  else
17    | initialize centroid  $i = |\mathcal{C}| + 1$ ;
18    |  $\mu_i \leftarrow v_j$ ;  $\Sigma_i \leftarrow 0$ ;  $n_i \leftarrow 1$ ;
19    |  $\mathcal{C} \leftarrow \mathcal{C} \cup \{(n_i, \mu_i, \Sigma_i)\}$ 
20  end
   /* visit neighborhood  $\mathcal{N}_j$  of  $v_j$  */
21  sort  $\mathcal{N}_j$  by distance to  $(n_i, \mu_i, \Sigma_i)$ ;
22  foreach  $v_k \in \mathcal{N}_j$  do
23    | if  $v_k \notin \bar{\mathcal{V}}$  then push  $v_k$  in  $\mathcal{Q}$ ;
24  end
25 end
   /* performs  $k$ -means starting from the initialized centroids */
26  $k \leftarrow |\mathcal{C}|$ ;
27  $\mathcal{C} \leftarrow \text{kmeans}(\mathcal{V}, \mathcal{C})$ ;

```

---

Scene segmentation is performed by associating a label to each voxel cell  $v_j$  according to the feature vector  $f_j$  and the proximity of similar cells. An unsupervised classifier has been adopted to perform an initial classification, which is refined using a Markov Random Field, as described in the next section.

### 3.2 Classification and Markov Random Field Inference

Scene segmentation is formulated as an unsupervised classification problem: given the cell set  $\mathcal{V}$  described before, the aim is to find a label set  $\mathcal{L}$  and a labeling function

$\lambda : \mathcal{V} \rightarrow \mathcal{L}$ . The labeling function  $\lambda(\cdot)$  should weight both the self-organization of features  $f_j$ , i.e. the existence of clusters in feature space, and the dependence among neighbor cells, i.e. the cell context. The  $k$ -means algorithm allows the a priori estimation of the  $k$  clusters, while a graphical model technique refines the initial classification through context-oriented inference.

Estimating the number of labels  $|\mathcal{L}|$  is an important operation that affects the detection of objects. Several criteria could be used such as Akaike's Information Criterion (AIC) [17] or Davies-Boulding Index (DBI) [18]. However, these indices are not related to the geometric structure of the 3D scene and the resulting clusters may turn out to be discontinuous. Algorithm 1 searches candidate cluster centroids taking into account both features similarity and their spatial distribution. In particular, the nodes of the graph defined by the neighborhood of each cell are visited according to a breadth-first search. Under the assumption that neighbor cells are likely to have similar feature, the feature values found during expansion are used to initialize or update a set of centroids  $\mathcal{C}$ . Such centroids represent subsets of points with similar features resulting from proximity. The search starts from a new seed for each connected component and is handled by a FIFO queue  $\mathcal{Q}$ . At each iteration, an unvisited item  $v_j$  is extracted from the queue or picked from the unvisited items in  $\tilde{\mathcal{V}}$ , when a connected component has been visited. If the closest centroid to the feature  $f_j$  of cell  $v_j$  is less than threshold  $\chi_{\alpha,d}^2$  (gaussian distribution is assumed and  $\alpha = 0.85$ ), then the closest centroid is updated (line 11) or a new centroid is initialized (line 16). This procedure adds a new centroid when a feature discontinuity between neighbors is detected and no other similar centroid has been detected before. After the initial clustering, the centroids obtained by the algorithm are used as the input values for  $k$ -means method. After the centroids have been estimated the label function  $\lambda(\cdot)$  is defined by applying to each cell  $v_j$  the label associated to its closest centroid in feature space. The label of a cell is also inherited by the points contained in the cell.

The assumption that spatial neighbors in the dataset tend to have the same labels has already been used to estimate the number of clusters and to generate an initial classification. However, the unsupervised classification of each  $v_i$  is independent from its neighbors. MRF techniques allow the refinement of the classification based only on individual similarity to the centroids. The implicit graph defined by neighbor sets represents relations among the random variables representing the labels of each cell. Let  $\lambda_j \in \mathcal{L}$  be the random variable representing the label of  $v_j$ ,  $\lambda = [\lambda_1, \dots, \lambda_n]^T \in \mathcal{L}^n$ ,  $\psi_j(\lambda_j)$  the unary potential associated to the label, and  $\phi_{jk}(\lambda_j, \lambda_k)$  the cross-potential between two neighbors. The new label vector  $\lambda$  is computed in order to minimize the energy function

$$E(\lambda) = \sum_{v_j \in \mathcal{V}} \psi_j(\lambda_j) + \sum_{v_j \in \mathcal{V} \wedge v_k \in \mathcal{N}_j} \phi_{jk}(\lambda_j, \lambda_k) \quad (1)$$

The function  $\psi_j(\lambda_j)$  depends on the initial classification operated on the single voxel  $v_j$  as discussed above. In this work, the value of the potential for a specific label  $l$

depends on the distance between the feature  $f_j$  and the centroid  $\mu_l$  corresponding to  $l$ . Instead the cross-potential has been defined according to the standard *Potts model* [19]. In particular, the analytical expression of the two terms is

$$\psi_j(\lambda_j = l) = \omega_u \log \left( 1 - \frac{d_{jl}}{\sum_{l'} d_{jl'}} \right) \quad (2)$$

$$\phi_{jk}(\lambda_j, \lambda_k) = \omega_c (1 - \delta_{jk}) \quad (3)$$

where  $d_{jl} = \|f_j - \mu_l\|$  is the feature to centroid distance,  $\delta_{jk}$  the Kronecker delta,  $\omega_u$  and  $\omega_c$  two proper constants. In our experiments these constants have been set to  $\omega_u = 1.0$  and  $\omega_c = 0.275$ . The cross-potential is equivalent to a smoothness function since it adds a penalty when the neighbor nodes have different labels.

The labels are computed by minimizing the energy function in Eq. (1). The computation has been performed using the Graph-Cut Optimization (GCO) library [20] with  $\alpha\beta$ -swap method. The application of MRF produces a more homogeneous distribution of labels in the same region of space as shown in Fig. 1c–d.

### 3.3 Selection

The result of segmentation is a labelled point cloud that can be partitioned into connected sets of voxels with the same labels. Such connected components will be called cluster in the remaining of the paper. In an ideal case, an object is represented by a single cluster and such cluster does not include voxels belonging to the background or to other objects. However, one of the following cases may arise:

1. an object is partitioned into more than one cluster;
2. a cluster covers more than an object and/or part of the background.

The first case is not critical for recognition if the cluster covers a significant portion of the object, since the object recognition algorithm can operate with partial object representations as shown in Sect. 4. The second case is addressed by the selection algorithm illustrated in this section. First, all the cluster are computed by performing Euclidean clustering according to the labels. Second each cluster is further divided by detecting loosely connected components, which usually correspond to single objects. Weak connections are found by estimating the point density in the spherical neighborhood around each point. Finally, the connected components containing more points than a given threshold are selected as candidate objects.

### 3.4 Recognition

Recognition of clusters is the last step in the described pipeline and aims at matching a selected cluster with an entry in a dataset of known models. The dataset consists of



**Fig. 2** Objects used by the recognition algorithm (*left*) and multiple models obtained from different PoV for an example object (*right*)

a variable number of views for each object taken from several viewpoints as shown in Fig. 2 (right). Each model is obtained by accumulating points from multiple frames in order to fill gaps of the cloud produced by stereo vision. Then a voxel grid filter is applied to achieve a uniformly sampled point cloud. The recognition algorithm is based on point clouds alignment. The two clouds of the  $i$ th model  $\mathcal{P}_i^{\text{mod}}$  and the current object  $\mathcal{P}^{\text{obj}}$  in 3D space need to be registered or aligned in order to be compared. The registration procedure computes the rigid geometric transformation that should be applied to  $\mathcal{P}_i^{\text{mod}}$  to align it to  $\mathcal{P}^{\text{obj}}$ . Registration is performed in three different steps:

- *Remove dependency on external reference frame*  
 $\mathcal{P}_i^{\text{mod}}$  and  $\mathcal{P}^{\text{obj}}$  are initially expressed in the reference of the respective centroids.
- *Perform initial alignment*  
 The algorithm estimates an initial and sub-optimal alignment between point clouds. This step is performed with the assistance of a RANSAC method that uses FPFH descriptors [5] as parameters for the function of consensus.
- *Refine the alignment*  
 The initial alignment is then refined with an ICP algorithm that minimizes the mean square distance between points.

The procedure is detailed in Algorithm 2 and an example result is shown in Fig. 1f.

Recognition is then performed by computing a *fitness* value that evaluates the overall quality of the alignment between  $\mathcal{P}_{i,\text{aligned}}^{\text{mod}}$  and  $\mathcal{P}^{\text{obj}}$ . For each point of  $\mathcal{P}^{\text{obj}}$ , the algorithm calculates the square mean distance from the nearest point of  $\mathcal{P}_{i,\text{aligned}}^{\text{mod}}$  and retrieves the percentage of points whose distance is below a fixed threshold  $\delta_{\text{th}}$ :

$$Q = \left\{ p_i \in \mathcal{P}^{\text{obj}} : \|p_j - p_i\|^2 \leq \delta_{\text{th}}, p_j \in \mathcal{P}_{i,\text{aligned}}^{\text{mod}} \right\} \quad (4)$$

$$\text{fitness}(\mathcal{P}^{\text{obj}}, \mathcal{P}_{i,\text{aligned}}^{\text{mod}}) = \frac{|Q|}{|\mathcal{P}^{\text{obj}}|} \quad (5)$$

Maximum fitness, equal to 100 %, is obtained when all points of  $\mathcal{P}^{\text{obj}}$  have a neighbour in  $\mathcal{P}_{i,\text{aligned}}^{\text{mod}}$  within  $\delta_{\text{th}}$  (in our experiments  $\delta_{\text{th}} = 1$  cm).

**Algorithm 2:** Registration procedure

---

**Data:**  
 $\mathcal{P}_i^{mod}$ : Point cloud of i-th model;  
 $\mathcal{P}^{obj}$ : Point cloud of the object to be recognized;

**Result:**  
 $\mathcal{P}_{i,aligned}^{mod}$ : Aligned point cloud of the model;

- 1  $\mathcal{P}_c^{obj} \leftarrow \text{shiftToCentroid}(\mathcal{P}^{obj});$
- 2  $\mathcal{P}_{i,c}^{mod} \leftarrow \text{shiftToCentroid}(\mathcal{P}_i^{mod});$
- 3  $\mathcal{F}_o \leftarrow \text{computeFPFH}(\mathcal{P}_c^{obj});$
- 4  $\mathcal{F}_m \leftarrow \text{computeFPFH}(\mathcal{P}_{i,c}^{mod});$
- 5  $\mathcal{P}_{i,sac}^{mod} \leftarrow \text{getRANSACAlignment}(\mathcal{P}_c^{obj}, \mathcal{F}_o, \mathcal{P}_{i,c}^{mod}, \mathcal{F}_m);$
- 6  $\mathcal{P}_{i,aligned}^{mod} \leftarrow \text{getICPAlignment}(\mathcal{P}_{i,sac}^{mod}, \mathcal{P}^{obj});$

---

**Algorithm 3:** Overall recognition procedure

---

**Data:**  
 $\mathcal{P}^{mod}[\cdot]$ : List of point cloud models;  
 $\mathcal{P}^{obj}$ : Point cloud of the object to be recognized;

**Result:**  
*name*: Name of the recognized object;

- 1  $\mathcal{F}_{max} \leftarrow 0;$
- 2 **foreach**  $\mathcal{P}_i^{mod} \in \mathcal{P}^{mod}[\cdot]$  **do**
- 3      $\mathcal{P}_{i,aligned}^{mod} \leftarrow \text{performRegistration}(\mathcal{P}_i^{mod}, \mathcal{P}^{obj});$
- 4      $\mathcal{F}_i \leftarrow \text{getFitness}(\mathcal{P}^{obj}, \mathcal{P}_{i,aligned}^{mod});$
- 5     **if**  $\mathcal{F}_i > \mathcal{F}_{max}$  **then**
- 6          $\mathcal{F}_{max} \leftarrow \mathcal{F}_i;$
- 7         *name*  $\leftarrow$  name of  $\mathcal{P}_i^{mod};$
- 8     **end**
- 9 **end**

---

The algorithm is iterated for each model in the dataset and returns the recognized model with the higher fitness as shown in Algorithm 3.

## 4 Results

This section presents the experiments performed to assess both the segmentation and the object recognition algorithms illustrated in the previous section. The detection, selection and recognition modules have been implemented as separated components using ROS (Robot Operating System) framework. The input range images are acquired using a stereo vision system consisting of two vertically aligned Logitech C270 cameras [21].

**Table 1** Precision, recall and V-measure in segmentation

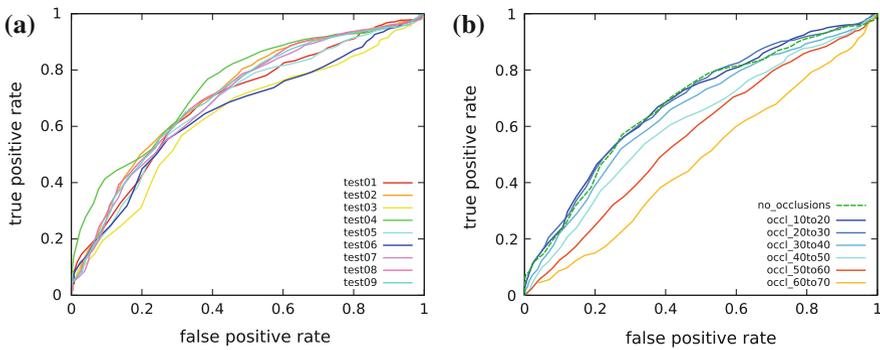
Algorithm	Precision (%)	Recall (%)	V-Measure
ANN	61.3	61.1	–
ANN+MRF	64.9	70.3	–
SVM	70.4	68.7	–
SVM+MRF	59.7	71.1	–
K-means	61.3	61.3	0.17
K-means+MRF	58.2	76.7	0.35

The first set of experiments is aimed at evaluating the performance of the proposed segmentation method. The used dataset consists of 13 point clouds within the annotated images representing the groundtruth. The annotated image is partitioned into background elements (floor, wall, door) and objects. The proposed unsupervised  $k$ -means clustering is compared with two supervised classifiers trained on the four categories: an artificial neural network (ANN) and a support vector machine (SVN). Each supervised multi-class classifier consists of four binary classifiers, one for each category, and the final decision is taken by choosing the binary classifier with the highest score. On the other hand, the output of the unsupervised classification consists of labels that cannot be directly associated to the groundtruth categories. Hence, each label  $\lambda_j$  has been associated to the groundtruth region  $r$  that contains more voxels labeled as  $\lambda_j$ . Of course, such correspondence is an approximation: the unsupervised algorithm is not aimed at classifying voxels into the classes defined a priori, but only at discovering distinctive similar voxels. It may occur that two connected clusters belonging to different groundtruth regions have the same label due to their feature similarity. In spite of these limitations, such comparison measures the internal consistency of the segmentation achieved by the unsupervised algorithm. The precision and the recall obtained in the experiments by the three methods with or without MRF refinement, are shown in Table 1. The values of precision and recall of  $k$ -means are comparable with those obtained with the two supervised classifiers, while the recall is slightly greater. The application of MRF tends to reduce the precision, but increases the recall. An assessment restricted to the unsupervised cases is given by the V-measure [22] which is equal respectively for K-means and K-means+MRF to 0.17 and 0.35. Hence, MRF significantly improves the V-measure, which is a cluster evaluation measure for unsupervised classification methods.

The second set of experiments is designed to assess the object recognition algorithm, in particular when only partial point clouds of the object are available due to noisy segmentation and occlusions. Test set consists on a fixed sequence of 2241 object point clouds taken from random viewpoints. The dataset consists of 61 models representing the 8 objects in Fig. 2 (8 views for each object on average). The first test series takes into account parameters of the algorithm like the number of model views in the dataset, the number of RANSAC iterations, the number of ICP iterations and

**Table 2** Experimental results for the recognition algorithm

	Test01	Test02	Test03	Test04	Test05	Test06	Test07	Test08	Test09
# Dataset	61	32	61	24	61	61	32	32	32
Radius (mm)	3, 5, 10, 20, 30, 50	3, 5, 10, 20, 30, 50	3, 10, 30	3, 5, 10, 20, 30, 50	5, 15	3, 10, 30	5, 15	3, 10, 30	5, 15
RANSAC it.	200	200	200	200	200	100	200	100	100
ICP it.	200	200	200	200	200	150	200	150	150
True Pos. (%)	80	70	83	60	81	81	73	70	69
False Pos. (%)	20	30	17	40	19	19	27	30	31

**Fig. 3** ROC curves for tests test 01–09 (a) and for tests with occlusions (b)

the search radius used to compute the FPFH (trials are called *test01*, *test02*, etc. in Table 2).

Experimental results show the importance of including dataset models taken from multiple viewpoints. Keeping fixed all the parameters while decreasing the dataset size, the percentage of true positives decreases (see *test01*, *test02* and *test04*). Results also show that, even limiting the search radius values, recognition rate is not negatively affected. Moreover, reducing RANSAC and ICP iterations marginally decreases performance (see *test03* and *test06*). The Receiver Operating Characteristic (ROC) curves in Fig. 3a depict the performance of the classifier as its discrimination threshold is varied. To summarize, the recognition algorithm has good performance with true positive rate above 80% provided that sufficient viewpoints models are available.

We have then evaluated the recognition algorithm with partial and occluded objects. In order to have comparable results, occlusions have been artificially generated with a random procedure. The *occlusion generator* processes the original test set and for each view chooses a random point in the cloud and removes all points within a random radius. In this way it generates a new *synthetically occluded* test set with occlusions measured as percentage of removed points. Six different tests have

**Table 3** Experimental results for test set with occlusions and recognition parameters as test05 [Table 2]

Occlusions (%)	10–20	20–30	30–40	40–50	50–60	60–70
True Pos. (%)	76	70	60	47	37	23
False Pos. (%)	24	30	40	53	63	77

been performed with increasing occlusions from 10 to 70%. Recognition results are shown in Table 3. Recognition algorithm preserves good performance with occlusions up to 30% with true positive rates above 70%. Performance rapidly decreases with occlusions up to 40% and then collapses with increasing percentage of occluded points. Figure 3b shows Precision-Recall curves for all tests with occlusions and a reference test without them. Performance with occlusions till 30% is consistent with the reference test.

## 5 Conclusion

In this paper, we have presented a complete system that performs unsupervised range image segmentation and object recognition by exploiting joint feature similarity and proximity. The acquired points are partitioned into voxels and a feature vector is extracted from each voxel. The initialization of the classifier is achieved by grouping similar features during a breadth-first expansion over neighbor voxels and then by classifying the voxels according to a  $k$ -means algorithm. The resulting segmentation is refined using a Markov Random Field method in order to detect contiguous and similar voxels. Then, the candidate objects are selected looking for foreground segments likely to represent objects and are compared to the models in an a priori dataset. Object recognition is performed by aligning the two point clouds. Experiments have been performed to assess both the consistency of the segments and the effectiveness of object recognition even with occlusions or partial views.

**Acknowledgments** This research is partially supported by MARIS project.

## References

1. Aleotti, J., Lodi Rizzini, D., Caselli, S.: Object Categorization and Grasping by Parts from Range Scan Data. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (2012) 4190–4196
2. Burel, G., H enocq, H.: Three-dimensional invariants and their application to object recognition. *Signal Process.* 45(1) (1995) 1–22
3. Johnson, A.: Spin-Images: A Representation for 3-D Surface Matching. PhD thesis, Robotics Institute, Carnegie Mellon University (August 1997)

4. Gatzke, T., Grimm, C., Garland, M., Zelinka, S.: Curvature Maps for Local Shape Comparison. In: Proc. of Int. Conf. on Shape Modeling and Applications (SMI). (2005) 246–255
5. Rusu, R., Blodow, N., Beetz, M.: Fast Point Feature Histograms (FPFH) for 3D registration. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (2009) 3212–3217
6. Aldoma, A., Marton, Z., Tombari, F., Wohlkinger, W., Potthast, C., Zeisl, B., Rusu, R., Gedikli, S., Vincze, M.: Tutorial: Point cloud library: Three-dimensional object recognition and 6 DOF pose estimation. *IEEE Robotics & Automation Magazine* 19(3) (Sept. 2012) 80–91
7. Kanezaki, A., Marton, Z.C., Pangercic, D., Harada, T., Kuniyoshi, Y., Beetz, M.: Voxelized Shape and Color Histograms for RGB-D. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Workshop on Active Semantic Perception and Object Search in the Real World.* (September, 25–30 2011)
8. Triebel, R., Schmidt, R., Martinez Mozos, O., Burgard, W.: Instance-based AMN Classification for Improved Object Recognition in 2D and 3D Laser Range Data. In: Proc. of the Int. Conf. on Artificial Intelligence (IJCAI). (2007) 2225–2230
9. Rusu, R., Holzbach, A., Beetz, M., Bradski, G.: Detecting and segmenting objects for mobile manipulation. In: *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on.* (27 2009–oct. 4 2009) 47–54
10. Tombari, F., Di Stefano, L.: 3D data segmentation by local classification and Markov Random Fields. In: Proc. of the Intl. Conf. on 3D Imaging, Modeling, Processing, Visualization and Transmission (TDIMPVT). (2011) 212–219
11. Lai, K., Bo, L., Ren, X., Fox, D.: Detection-based Object Labeling in 3D Scenes. In: Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA). (2012) 1330–1337
12. Douillard, B., Underwood, J., Vlaskine, V., Quadros, A., Singh, S.: A pipeline for the segmentation and classification of 3D point clouds. In: Proc. of the International Symposium on Experimental Robotics (ISER). (2010)
13. Pauling, F., Bosse, M., Zlot, R.: Automatic segmentation of 3D laser point clouds by ellipsoidal region growing. In: Proc. of Australasian Conference on Robotics and Automation (ACRA). (Dec 2009)
14. Uckermann, A., Haschke, R., Ritter, H.: Real-Time 3D Segmentation of Cluttered Scenes for Robot Grasping. In: Proc. of Int. Conf. on Humanoid Robotics (HUMANOID). (2012)
15. Song, R., Liu, Y., Martin, R., Rosin, P.: Markov random field-based clustering for the integration of multi-view range images. In *Bebis, G., Boyle, R., Parvin, B., Koracin, D., Chung, R., Hammoud, R., Hussain, M., Kar-Han, T., Crawfis, R., Thalmann, D., Kao, D., Avila, L., eds.: Advances in Visual Computing. Volume 6453 of Lecture Notes in Computer Science. Springer, Berlin Heidelberg (2010) 644–653*
16. Herbst, E., Ren, X., Fox, D.: RGB-D object discovery via multi-scene analysis. In: Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS). (2011) 4850–4856
17. Burnham, K., Anderson, D.: Kullback-Leibler information as a basis for strong inference in ecological studies. *Wildlife Research* 28(2) (May 2001) 111–119
18. Davies, D., Bouldin, D.: A cluster separation measure. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 1(2) (April 1979) 224–227
19. Wu, F.: The potts model. *Reviews of Modern Physics* 54(1) (Jan 1982) 235–268
20. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence* 23(11) (2001) 1222–1239
21. Oleari, F., Lodi Rizzini, D., Caselli, S.: A Low-Cost Stereo System for 3D Object Recognition. In: Proc. of the Int. Conf. on Intelligent Computer Communication and Processing (ICCP), Cluj-Napoca, Romania (Sept 2013) 127–132
22. Rosenberg, A., Hirschberg, J.: V-measure: A conditional entropy-based external cluster evaluation measure. In: J. Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). (2007)