Contour-based next-best view planning from point cloud segmentation of unknown objects

Dear Author,

Here are the proofs of your article.

- You can submit your corrections **online**, via **e-mail** or by **fax**.
- For **online** submission please insert your corrections in the online correction form. Always indicate the line number to which the correction refers.
- You can also insert your corrections in the proof PDF and **email** the annotated PDF.
- For fax submission, please ensure that your corrections are clearly legible. Use a fine black pen and write the correction in the margin, not too close to the edge of the page.
- Remember to note the **journal title**, **article number**, and **your name** when sending your response via e-mail or fax.
- **Check** the metadata sheet to make sure that the header information, especially author names and the corresponding affiliations are correctly shown.
- **Check** the questions that may have arisen during copy editing and insert your answers/ corrections.
- **Check** that the text is complete and that all figures, tables and their legends are included. Also check the accuracy of special characters, equations, and electronic supplementary material if applicable. If necessary refer to the *Edited manuscript*.
- The publication of inaccurate data such as dosages and units can have serious consequences. Please take particular care that all such details are correct.
- Please **do not** make changes that involve only matters of style. We have generally introduced forms that follow the journal's style.
  Substantial changes in content, e.g., new results, corrected values, title and authorship are not allowed without the approval of the responsible editor. In such a case, please contact the Editorial Office and return his/her consent together with the proof.
- If we do not receive your corrections **within 48 hours**, we will send you a reminder.
- Your article will be published **Online First** approximately one week after receipt of your corrected proofs. This is the **official first publication** citable with the DOI. **Further changes are, therefore, not possible.**
- The **printed version** will follow in a forthcoming issue.

Please note

After online publication, subscribers (personal/institutional) to this journal will have access to the complete article via the DOI using the URL: http://dx.doi.org/[DOI].
If you would like to know when your article has been published online, take advantage of our free alert service. For registration and further information go to: http://www.link.springer.com.

Due to the electronic nature of the procedure, the manuscript and the original figures will only be returned to you on special request. When you return your corrections, please inform us if you would like to have these documents returned.

Metadata of the article that will be visualized in OnlineFirst

Please note: Images will appear in color online but will be printed in black and white.

ArticleTitle	Contour-based next-best view planning from point cloud segmentation of unknown objects	
Article Sub-Title		
Article CopyRight	Springer Science+Business Media New York (This will be the copyright line in the final PDF)	
Journal Name	Autonomous Robots	
Corresponding Author	Family Name	**Aleotti**
	Particle	
	Given Name	**Jacopo**
	Suffix	
	Division	Dipartimento di Ingegneria dell'Informazione
	Organization	University of Parma
	Address	Parma, Italy
	Phone	+39-0521-905727
	Fax	
	Email	aleotti@ce.unipr.it
	URL	
	ORCID	http://orcid.org/0000-0003-2498-932X
Author	Family Name	**Monica**
	Particle	
	Given Name	**Riccardo**
	Suffix	
	Division	Dipartimento di Ingegneria dell'Informazione
	Organization	University of Parma
	Address	Parma, Italy
	Phone	+39-0521-905702
	Fax	
	Email	rmonica@ce.unipr.it
	URL	
	ORCID	

	Received	26 February 2016
Schedule	Revised	
	Accepted	11 January 2017

Abstract

A novel strategy is presented to determine the next-best view for a robot arm, equipped with a depth camera in eye-in-hand configuration, which is oriented to autonomous exploration of unknown objects. Instead of maximizing the total size of the expected unknown volume that becomes visible, the next-best view is chosen to observe the border of incomplete objects. Salient regions of space that belong to the objects are detected, without any prior knowledge, by applying a point cloud segmentation algorithm. The system uses a Kinect V2 sensor, which has not been considered in previous works on next-best view planning, and it exploits KinectFusion to maintain a volumetric representation of the environment. A low-level procedure to reduce Kinect V2 invalid points is also presented. The viability of the approach has been

	demonstrated in a real setup where the robot is fully autonomous. Experiments indicate that the proposed method enables the robot to actively explore the objects faster than a standard next-best view algorithm.
Keywords (separated by '-')	Next-best view planning - KinectFusion - Point cloud segmentation
Footnote Information	**Electronic supplementary material** The online version of this article (doi:10.1007/s10514-017-9618-0) contains supplementary material, which is available to authorized users. This is one of several papers published in *Autonomous Robots* comprising the Special Issue on Active Perception.

Auton Robot
DOI 10.1007/s10514-017-9618-0

Contour-based next-best view planning from point cloud segmentation of unknown objects

Riccardo Monica[1] · **Jacopo Aleotti**[1]

Received: 26 February 2016 / Accepted: 11 January 2017

Abstract A novel strategy is presented to determine the next-best view for a robot arm, equipped with a depth camera in eye-in-hand configuration, which is oriented to autonomous exploration of unknown objects. Instead of maximizing the total size of the expected unknown volume that becomes visible, the next-best view is chosen to observe the border of incomplete objects. Salient regions of space that belong to the objects are detected, without any prior knowledge, by applying a point cloud segmentation algorithm. The system uses a Kinect V2 sensor, which has not been considered in previous works on next-best view planning, and it exploits KinectFusion to maintain a volumetric representation of the environment. A low-level procedure to reduce Kinect V2 invalid points is also presented. The viability of the approach has been demonstrated in a real setup where the robot is fully autonomous. Experiments indicate that the proposed method enables the robot to actively explore the objects faster than a standard next-best view algorithm.

Keywords Next-best view planning · KinectFusion · Point cloud segmentation

This is one of several papers published in *Autonomous Robots* comprising the Special Issue on Active Perception.

Electronic supplementary material The online version of this article (doi:10.1007/s10514-017-9618-0) contains supplementary material, which is available to authorized users.

✉ Jacopo Aleotti
aleotti@ce.unipr.it

Riccardo Monica
rmonica@ce.unipr.it

1 Dipartimento di Ingegneria dell'Informazione, University of Parma, Parma, Italy

1 Introduction

Autonomous robot exploration and 3D reconstruction of a scene may be very time consuming if not guided by active perception, even in tabletop scenarios. An active perception behavior usually drives the robot by computing the Next Best View (NBV) to observe the most relevant areas of the environment, given the data acquired so far. Traditional NBV algorithms attempt to maximize the information gain by exploring unknown or incomplete parts of the scene. However, a straightforward maximization of the volume of the unknown space may not be the proper solution as the robot may prioritize large occluded areas that do not contain any interesting object. Moreover, NBV planning is usually performed by constraining the viewpoint to lie on a viewing sphere around the object, but the location of the objects may be unknown in advance.

This paper proposes a novel approach for NBV planning of a robot arm equipped with an eye-in-hand range sensor in a tabletop scenario. The robot gives precedence to the exploration of the objects in the scene without any prior knowledge about their shape and position. Such non-model-based approach is achieved by applying a point cloud segmentation algorithm to the sensor data and then by assigning a saliency value to each segment. The NBV system prioritizes viewpoints that observe the segment with the highest saliency. We show that after point cloud segmentation a simple heuristic can be adopted to identify meaningful segments that belong to the objects. In particular, a method for point cloud segmentation is adopted based on Locally Convex Connected Patches (LCCP) by Stein et al. (2014), which is available within the PCL library. The exploitation of point cloud segmentation for active scene exploration has been considered in few previous works.

Journal: **10514** MS: **9618** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2017/1/23** Pages: **16** Layout: **Large**

A further contribution is the computation of the NBV on the GPU through a modified version of KinFu. The KinFu Large Scale (KinFu LS) project is an open source implementation of KinectFusion (Newcombe et al. 2011) in the PCL library. The system exploits the GPU NBV algorithm developed in Monica et al. (2016) and the environment is modeled as a volumetric 3D voxel grid on the GPU using a Truncated Signed Distance Function (TSDF). It is also shown how viewpoint directions can be extracted directly from the KinFu TSDF volume, using a local contour extraction algorithm. The proposed approach is fully autonomous, it only requires an initial short scan of the environment, from one side, and it does not assume the existence of a dominant plane in the scene. In the experimental setup the robot arm is equipped with a Kinect V2 sensor. To the best of our knowledge, this is the first work that reports the use of Kinect V2 for NBV planning. Kinect V2 has a higher resolution and a higher field of view with respect to Kinect V1. Moreover, Kinect V2 has proven to be two times more accurate in the near range and it presents an increased robustness to artificial illumination. A novel procedure has also been developed for Kinect V2 depth image pre-processing. Experiments in environments with multiple complex objects show that the system is able to reconstruct the scene around the objects faster than a traditional NBV planner which maximizes the volume of the unknown space.

The paper is organized as follows. Section 2 provides an overview of the state of the art. Section 3 describes the proposed active perception system. Section 4 illustrates the experimental results. Section 5 concludes the paper and provides suggestions for possible extensions.

2 Related work

The two closest works to ours that considered NBV planning from point cloud segmentation are Wu et al. (2015) and Kai et al. (2015). Both methods have been evaluated in scenes without or with few stacked objects. In Wu et al. (2015) an active object recognition system was proposed for a mobile robot. A feature-based model was used to compute the NBV in 2D space by predicting both visibility and likelihood of feature matching. Experiments were reported with box-shaped objects where the mobile robot was not autonomous but it was manually placed as dictated by the NBV algorithm. Main differences are that this work focuses on an autonomous robot arm and that objects have more complex shapes. In Kai et al. (2015) a graphcut object segmentation is performed on an initial robot scan, using Kinect V1, through KinectFusion. Then, the PR2 robot performs proactive exploration to validate the object-aware segmentation by combining next-best push planning and NBV planning. NBV planning is performed only on pushed objects for scan refinement. The procedure for robot motion planning is not described. Another difference is that in our work NBV is computed on the GPU.

The most common assumption for NBV planning is to determine the optimal placement of the eye-in-hand sensor on a viewing sphere around a target location. An objective function is usually chosen which maximizes the unknown volume as proposed by Connolly (1985) and Banta et al. (2000). Pito (1999) used a turntable and ensured an overlap among consecutive views. In Reed and Allen (2000) and Vasquez-Gomez et al. (2009) sensor constraints were included to minimize the distance traveled by the robot, but the methods were evaluated in simulation. In Yu and Gupta (2004) NBV was aimed at reducing ignorance of the configuration space of the robot. Potthast and Sukhatme (2014) proposed a customizable framework for NBV planning in cluttered environments where a PR2 robot estimates the visibility of occluded space using a probabilistic approach. Kahn et al. (2015) presented a method to plan the motion of the sensor to enable robot grasping by looking for object handles lying within occluded regions of the environment.

Several NBV approaches assume that the location of the target object is known and do not cope with the problem of detecting the most relevant regions of the environment to be explored. Indeed, in Torabi and Gupta (2010), Kriegel et al. (2012), Foix et al. (2010), Morooka et al. (1998), Li and Liu (2005) and Walck and Drouin (2010) a single object in the environment was considered. Another less sophisticated strategy is to adopt a turntable to rotate the object observed from a fixed sensor. In Kriegel et al. (2012) a next-best scan planner was proposed for a laser stripe profiler aimed at maximizing the quality of the reconstruction. In Kriegel et al. (2011) a non model-based approach was introduced for NBV using the boundaries of the scan and by estimating the surface trend of the unknown area beside the boundaries. Some authors have addressed the NBV problem assuming a simple geometry of the objects to be scanned (Chen and Li 2005), or adopting simple parametric models like superquadrics (Whaite and Ferrie 1997). In Welke et al. (2010) and Tsuda et al. (2012) approaches have been developed for humanoid active perception of a grasped object.

In model based approaches the environment is actively explored to discover the location of objects of interest whose template or class is known in advance (Kriegel et al. 2013; Atanasov et al. 2014; Stampfer et al. 2012; Patten et al. 2016). Kriegel et al. (2013) presented an exploration system, combining different sensors, for tabletop scenes that supports NBV planning and object recognition. In Atanasov et al. (2014) an active hypothesis testing problem was solved using a point-based approximate partially observable Markov decision process algorithm. Stampfer et al. (2012) performed active object recognition enriched with common features like text and barcode labels. In Patten et al. (2016) a viewpoint evaluation method was proposed for online active object clas-

Journal: 10514 MS: 9618 TYPESET DISK LE CP Disp.:2017/1/23 Pages: 16 Layout: Large

sification that predicts which points of an object would be visible from a particular viewpoint given the previous observation of other nearby objects.

As mentioned above active exploration strategies have also been proposed where the robot interacts with the environment by pushing the objects (van Hoof et al. 2014; Kai et al. 2015). In van Hoof et al. (2014) the robot autonomously touched the objects to resolve segmentation ambiguities using a probabilistic model. However, NBV planning was not considered. Beale et al. (2011) exploited the correlation between robot and objects motion data to improve segmentation.

Several works have addressed the problem of change detection for scene reconstruction using attention-based approaches. Most attention based approaches do not consider active exploration using NBV sensor planning. Herbst et al. (2014) presented a method for online 3D object segmentation and mapping from recordings of the same scene at several times. Attention based systems have been proposed to direct gaze of humanoid robots or stereo-heads toward relevant locations. Bottom up saliency maps were used in Orabona et al. (2005) from blobs of uniform color. In mobile robotics attention driven methods have been investigated to maintain a consistent representation of the environment as the robot moves (Finman et al. 2013; Drews et al. 2013). In Finman et al. (2013) segmentations of objects were automatically learned from dense RGBD mapping. A method for novelty detection based on Gaussian mixture models from laser scan data was introduced in Drews et al. (2013).

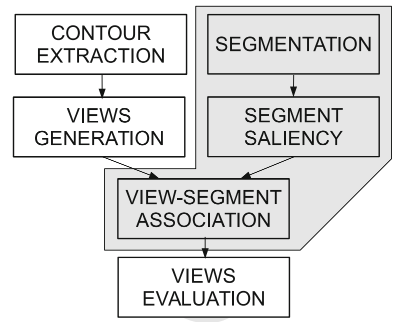


Fig. 1 Pipeline of the view planning algorithm. The *grey* background highlights the intermediate phase

Algorithm 1: View planning

Input: *WS*: 3D volumetric environment representation
Output: Next-best view
1: $Contour \leftarrow$ ContourExtraction(WS)
2: $\forall_{b \in Contour}$ $b.Viewpoints \leftarrow$ GetViewpoints(b)
3: $PointCloud \leftarrow$ ExtractSurfacePts(WS)
4: $Segments \leftarrow$ SegmentPointCloud($PointCloud$)
5: $Saliency \leftarrow$ SegmentSaliency($Segments$)
6: $Segments \leftarrow$ OrderBy($Segments$,$Saliency$)
7: **for** i **from** 1 **to** size($Segments$) **do**
8: $SContour \leftarrow$ FindNear($Contour$,$Segments[i]$)
9: $SViewpoints \leftarrow \bigcup_{b \in SContour} b.Viewpoints$
10: $Scores \leftarrow$ EvaluateViewpoints($SViewpoints$)
11: $SViewpoints \leftarrow$ OrderBy($SViewpoints$,$Scores$)
12: **for** j **from** 1 **to** size($SViewpoints$) **do**
13: **if** $Scores[j] > ScoreTH$ **then**
14: **return** $SViewpoints[j]$
15: **end if**
16: **end for**
17: **end for**
18: **return** {no suitable viewpoint found}

3 Proposed method for next-best view planning

In traditional non-model based approaches next-best view planning is performed in two phases. In the first phase, candidate view poses are generated. In the second phase, all the poses are evaluated according to a score function to find the next-best view pose. The proposed pipeline to compute the NBV, illustrated in Fig. 1, differs from traditional approaches as it introduces an intermediate phase between viewpoint generation and evaluation. In the intermediate phase the input point cloud is segmented into clusters and a saliency value is computed for each point cloud segment. The aim of the point cloud segmentation phase is to automatically detect segments that belong to the objects of the scene. In the evaluation phase potential view poses are associated to point cloud segments and the NBV is searched among view poses in decreasing order of segment saliency.

A more detailed overview of the proposed view planning pipeline is reported in Algorithm 1. The view generation phase is performed by a contour extraction algorithm (line 1), detailed in Sect. 3.1, which extracts contour points, i.e. points at the border of incomplete surfaces. Contour extraction also produces a view direction for each contour point. Then, from each view direction multiple view poses are generated, as shown in Fig. 2, mainly to increase the probability of finding a reachable pose for the robot manipulator. In particular, for each direction four additional view directions towards the same contour point are sampled within a small solid angle (15°). To convert each view direction into a pose for the sensor, a distance from the contour point must be selected, compatible with the sensor minimum and maximum sensing distance. Although view poses may be generated by selecting multiple distances, in this work a fixed distance of 80 cm was adopted, which was empirically determined by evaluating the average maximum distance that the robot is able to reach from the objects in the current experimental setup. In addition, a rotation angle around the view direction must be chosen. Eight samples for each view direction are generated at 45° intervals, starting from an arbitrary initial orientation.

In line 3, a point cloud ($PointCloud$) is extracted from the TSDF volume using the marching cubes algorithm, already available in KinFu. Marching cubes generates a mesh from

Journal: 10514 MS: 9618 TYPESET DISK LE CP Disp.:2017/1/23 Pages: 16 Layout: Large

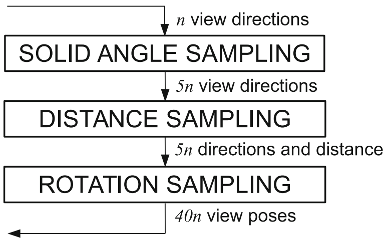


Fig. 2 Flowchart of viewpoint generation (line 2 in Algorithm 1). Each candidate view direction generates 40 view poses for the sensor

the isosurface between positive (empty) and negative (occupied) TSDF voxels. The vertices of the mesh define the point cloud. In the segmentation phase (line 4) the point cloud is segmented using the LCCP algorithm. Then, a saliency value is computed for each segment (line 5), as described in Sect. 3.2. Finally, the segments are ordered by decreasing saliency (line 6).

In the viewpoint evaluation phase (lines 7–17) view poses are associated to segments and are processed by decreasing segment saliency. In particular, all contour points close to the current segment are determined (line 8). Given the set $P_C \equiv PointCloud$ of all points in all segments, a contour point p is close to the current segment S if the nearest point to p in P_C belongs to S. All view poses generated by the contour points of the current segment are then retrieved (line 9). View poses associated to a segment are evaluated by assigning a score proportional to the expected information gain, as in traditional NBV approaches. Indeed, the expected information gain of each view pose is given by the amount of unknown voxels visible from that pose, which is available from the TSDF volume (Monica et al. 2016). A voxel contributes to the score only if it falls inside a sphere with radius 20 cm larger than the bounding sphere of the segment.

View poses associated to the current point cloud segment are then ranked and processed in decreasing order of score. If the expected information gain of a view pose exceeds a threshold value (line 13) that pose is considered the NBV. Otherwise, if the expected information gains of all the view poses of the current segment are below the threshold, the algorithm moves to evaluate the view poses of the next most salient segment. In summary, the proposed procedure is aimed at giving priority to active exploration of salient segments of unknown objects, not fully reconstructed, rather than favoring viewpoints that blindly try to minimize the size of the unknown space.

3.1 Contour extraction from TSDF volume

The TSDF volume is a volumetric representation of the environment used by the KinectFusion algorithm. The space is subdivided into a regular 3D grid of voxels and each voxel holds the sampled value $v(x, y, z)$ of the Truncated Signed Distance Function $R^3 \rightarrow R$, which describes the signed distance from the nearest surface, clamped between a minimum and a maximum value. The TSDF is positive in empty space and negative in occupied space. Each voxel also contains a weight w, initialized to 0, that counts the number of times the voxel has been observed, up to a maximum amount. The TSDF value v and the weight can be used to distinguish between empty, occupied and unknown voxels as follows:

$$\begin{cases} w = 0 & \rightarrow \textit{unknown voxel} \\ w > 0 \begin{cases} v \leq 0 \\ v > 0 \end{cases} & \begin{matrix} \rightarrow \textit{occupied voxel} \\ \rightarrow \textit{empty voxel} \end{matrix} \end{cases} \tag{1}$$

Rarely observed voxels have a low weight, while completely unknown voxels have 0 weight. In unexplored space, or deep inside the surface of objects, voxels are unknown.

In NBV planning a frontier is defined as the region between seen-empty voxels and unknown space. A frontier is a region that can be explored, since the viewing sensor might be placed in the empty space next to the frontier to observe the unknown space. Occupied voxels do not belong to the frontier, since the sensor can not see through them. However, occupied voxels lying next to a frontier have implications for NBV planning. Indeed, observation of the region of space in close proximity to occupied voxels next to a frontier can extend the perception of the surface of the object those occupied voxels belong to.

In the context of this work a contour is defined as the set of empty voxels that are near to occupied voxels next to a frontier, i.e. a contour consists of voxels that are near to both an occupied voxel and an unknown voxel. To exclude false positive known voxels from being processed, due to noise, a voxel is considered known if observed at least 5 times, i.e. $w \geq W_{th}$, where $W_{th} = 5$ is a lower bound threshold. Given the 6-connected neighborhood N_e^6 and the 18-connected neighborhood N_e^{18} of a voxel at position e, the voxel belongs to a contour if the following conditions hold:

$$\begin{cases} w\,(e) \geq W_{th} \ \wedge \ v\,(e) > 0 \\ \exists \ u \in N_e^6 \ \mid \ w\,(u) < W_{th} \\ \exists \ o \in N_e^{18} \ \mid \ w\,(o) \geq W_{th} \ \wedge \ v\,(o) \leq 0 \end{cases} \tag{2}$$

A simplified 2D example is shown in Fig. 3, using the Von Neumann neighborhood (4-connected) and the Moore neighborhood (8-connected) in place of the 6-connected neighborhood N_e^6 and the 18-connected neighborhood N_e^{18} used in the 3D case. In the previous view the sensor observed the object from the right side, thus the view was partially obstructed and the cells in the lower left part of the image are not observed and left unknown. The cross marks a computed contour cell.

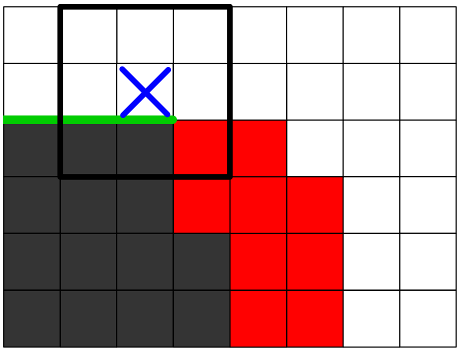


Fig. 3 A simplified 2D example of the contour extraction algorithm using Von Neumann neighborhood (4-connected) and Moore neighborhood (8-connected). In the previous view the sensor observed the object from the *right side*. A computed contour cell is marked with the *cross*. The *thicker square highlights* the Moore neighborhood of the contour cell. The *green segment* represents a frontier. Known and occupied cells are displayed in *red*, known and empty cells are in white, unknown cells are in *dark grey*

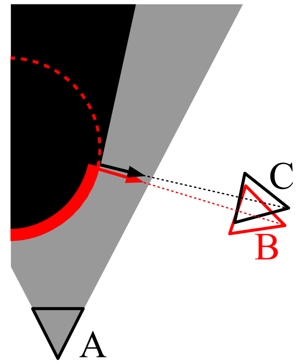


Fig. 4 Generation of the next potential viewpoint for a rounded object. Viewpoints B (computed from the surface normal) and C (computed from the frontier normal) are very similar

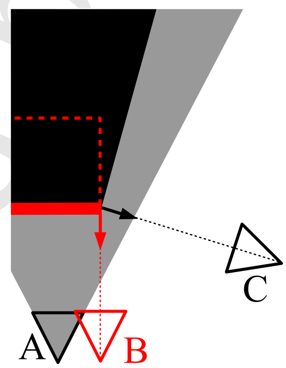


Fig. 5 Generation of the next potential viewpoint for an object with sharp edges. Only viewpoint C computed from the frontier normal allows the observation of the unknown volume behind the sharp edge

Given the previous definitions a method to compute a potential view direction from each contour voxel is described next. For optimal observation, the sensor should observe the object perpendicularly to its surface. Thus, the opposite of the surface normal computed on the occupied voxel next to the contour voxel can be used as potential view direction. The normal to the surface can be computed from the TSDF volume as the gradient $\nabla v\,(x, y, z)$ of v.

Given a neighborhood N_e of a voxel at position e, the normal may be approximated as (normalization omitted):

$$n_e = \sum_{c \in N_e} v\,(c) \cdot \frac{c - e}{\|c - e\|} \tag{3}$$

which, for a 6-connected neighborhood, reduces to

$$n_e = \begin{bmatrix} v\,(x+1, y, z) - v\,(x-1, y, z) \\ v\,(x, y+1, z) - v\,(x, y-1, z) \\ v\,(x, y, z+1) - v\,(x, y, z-1) \end{bmatrix} \tag{4}$$

since $(c - e)/\|c - e\|$ are unary vectors of the coordinate system.

The limitation of this approach is shown in Figs. 4 and 5. In both examples the sensor takes a first observation from the bottom, at position A. The observed volume is displayed in light grey. An object, marked with a dashed line, is partially observed in the red region. The volume behind the object remains unknown (black). The surface normal for the computed contour cell is displayed as a red arrow pointing outside the object. The generated potential viewing pose (B) from the surface normal is shown as the red triangle. In Fig. 4 for a rounded object surface the surface normal provides a good direction for the next view. However, for objects with sharp edges (like boxes), as illustrated in Fig. 5, the normal at the contour cell does not provide a suitable view direction since it does not allow the observation of the region of the object in the unknown space behind the edge. Indeed, in this second example at location B the sensor can not acquire any new information, since the lower plane of the box has already been observed from the initial view.

To overcome this limitation, we propose a method that computes the potential view directions using the normal to the frontier, i.e. the normal to the unknown volume. The normal to the frontier is indicated as view C. While in Fig. 4 for a rounded object the viewing pose is rather similar to the one computed by the surface normal, in Fig. 5 for a sharp edge view C provides a much better view direction to observe the object from the side.

In this work we use a fast local approach to approximate the normal of the frontier using the gradient of the weight function $\nabla w\,(x, y, z)$ which can be computed as

$$n_e = \sum_{c \in N_e^{26}} w'\,(c) \cdot \frac{c - e}{\|c - e\|} \tag{5}$$

Journal: **10514** MS: **9618** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2017/1/23** Pages: **16** Layout: **Large**

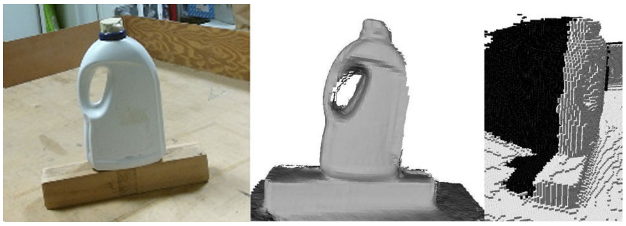


Fig. 6 *Left* a jug observed from the current sensor viewpoint. *Center* the 3D mesh reconstructed by KinFu. *Right* the volumetric representation (rotated view), with occupied (*white*) and unknown (*black*) voxels

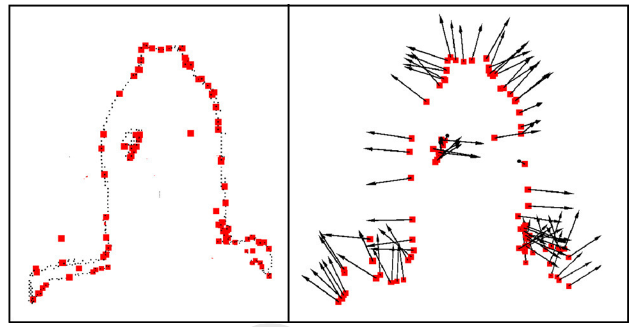


Fig. 7 *Left* contour voxels (*black*) and the contour points (*red*). *Right* contour points with normals. A contour point represent a group of similar contour voxels

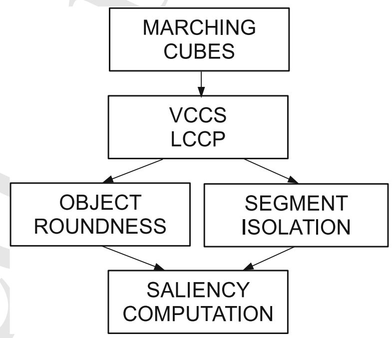


Fig. 8 Proposed procedure for point cloud segmentation and computation of the saliency value of each segment

where the 26-connected neighborhood of a voxel is used to reduce noise and sampling effects.

Since $w(c)$ is a positive integer value, Eq. 5 uses a modified weight function w' defined as

$$w'(c) = \begin{cases} -W_{th} & \text{if } c \text{ occupied} \\ \min(w(c) - W_{th}, W_{th}) & \text{otherwise} \end{cases} \tag{6}$$

For occupied voxels weight w is set to $-W_{th}$, since we want the normal to point away from them. Otherwise, w is first centered around 0 and then truncated to W_{th}.

In practice, after extraction of all the contour voxels with their view directions (line 1 in Algorithm 1), similar contour voxels are reduced into a contour point by a region growing algorithm. Two contour voxels at position e_1 and e_2, with view direction n_1 and n_2 are considered similar if

$$\begin{cases} \|e_1 - e_2\| < D_{th} \\ \|n_1 \cdot n_2\| < A_{th} \end{cases} \tag{7}$$

Each group of similar voxels is reduced to a single contour point with an associated view direction by averaging the positions and the view directions of the voxels.

Figures 6 and 7 show an example of contour extraction and viewpoint computation. In Fig. 6 the sensor observes a jug from the current NBV and a partial 3D representation is produced by KinFu. As shown by the ternary volumetric representation, voxels behind the object remain unknown. Contour voxels are extracted and clustered as illustrated in Fig. 7. The normal vectors point outwards towards the empty space. Thus, from that directions the robot may be able to observe the unknown space behind the object.

3.2 Saliency of point cloud segments

This section illustrates how the segmentation of the point cloud, extracted from the TSDF volume, is performed and how the saliency value of each segment is computed (lines 4–5 in Algorithm 1). The procedure is illustrated in Fig. 8. The point cloud is segmented by the LCCP (Stein et al. 2014) algorithm, available in the PCL library. LCCP partitions the input point cloud into a set of *Segments* (line 4 in Algorithm 1) by merging patches, called supervoxels, of an over-segmented point cloud. Supervoxels are generated by the a Voxel Cloud Connectivity Segmentation algorithm (VCCS) by Papon et al. (2013).

VCCS requires knowledge about the normals to the point cloud, unless points are acquired from the same viewpoint, which is not applicable in our system. Normal vectors could be computed as the normals to the faces of the mesh extracted by the marching cubes algorithm. However, we obtain the vertex normals with minimal overhead by using the gradient of the TSDF volume, as shown by Eq. 3 in Sect. 3.1, using a 6-connected neighborhood which is is already available for marching cubes operations.

The saliency function is a heuristic model that should provide an objectness measure, i.e. it should provide higher values for segments that belong to real objects of the scene. In this work the saliency of each segment is computed as a function of two features: the segment roundness and the degree of isolation.

The roundedness of a segment S is estimated as the ratio of the minimum and maximum sizes of the Oriented Bounding Box (OBB) of S. The sizes (d_1, d_2, d_3) of the OBB are defined in a local reference frame T_{OBB} centered at the mean point of the segment whose axes are given by the eigenvectors of the covariance matrix of the points (principal axes of inertia). More formally,

$$\begin{aligned} d_1 &= \max_{c \in S}\left(c'_x\right) - \min_{c \in S}\left(c'_x\right) \\ d_2 &= \max_{c \in S}\left(c'_y\right) - \min_{c \in S}\left(c'_y\right) \\ d_3 &= \max_{c \in S}\left(c'_z\right) - \min_{c \in S}\left(c'_z\right) \end{aligned} \tag{8}$$

where c is a point of S in the world reference frame and c' is the transformed point in the local reference frame

$$c' = T_{OBB}^{-1} \cdot c \tag{9}$$

The minimum and maximum sizes of the OBB of S are then

$$\begin{aligned} d_{max} &= \max_{i \in \{1,2,3\}} (d_i) \\ d_{min} &= \min_{i \in \{1,2,3\}} (d_i) \end{aligned} \tag{10}$$

We define the degree of isolation of a segment as the fraction of points for which the distance to points belonging to other segments is at least B_{th}. Given a segment $S \in Segments$ and the set $\hat{S}$ of all the points not in S, the degree of isolation of S is given by

$$F(S) = \frac{\left\| \left\{ c \in S \mid \forall o \in \hat{S},\ |c - o| > B_{th} \right\} \right\|}{\|S\|} \tag{11}$$

where $\|S\|$ is the total number of points in S. Equation 11 can be efficiently computed using a KdTree radius search of size B_{th}. Feature F has three benefits. First, it is meant to reward isolated segments belonging to partially observed objects, since a large part of their boundary is not shared with any other segment. Second, this heuristic is helpful for noise rejection as noisy segments, not well separated from other segments, often have a large boundary. Third, Eq. 11 penalizes small segments.

Finally, the saliency value of a segment S is computed as

$$Saliency(S) = F(S) \cdot \frac{d_{min}}{d_{max}} \tag{12}$$

so that saliency is proportional to the degree of segment isolation and it grows the more the maximum and the minimum sizes of the OBB are similar. An example of a segmented point cloud with saliency values is shown in Fig. 9. It can be noted that the segment isolation factor reduces the saliency value of noisy segments (inside the red ellipse).

Figure 10 shows the effect of B_{th} on the saliency. As B_{th} increases, the saliency value of the noisy segments at the front decreases. However, when B_{th} is too high, all the points of the small segments are rejected and, therefore, small objects

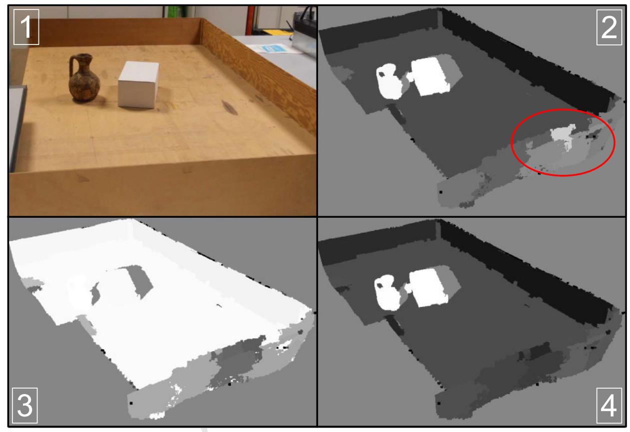


Fig. 9 Example of point cloud segmentation and saliency evaluation. Brighter segments have higher saliency value. (**1**) a picture of the scenario, (**2**) saliency evaluated by segment roundness alone, (**3**) saliency evaluated by segment isolation alone, (**4**) saliency evaluated by both segment roundness and isolation according to Eq. 12. The segment isolation factor reduces the saliency of the noisy segments inside the *red ellipse*

assume a zero saliency value (black color). Hence, for the experimental evaluation reported in Sect. 4.2, the value $B_{th} = 0.02$ m was chosen.

3.3 Kinect V2 depth image pre-processing

This section describes a low-level pre-processing filter to improve the quality of Kinect V2 depth data. The Kinect2 driver (Freenect2) provides two pre-processing filters: a bilateral filter and an edge-aware filter. The proposed filter is executed at the end of the standard filtering pipeline in place of the edge-aware filter, which does not strongly contribute to the removal of invalid points. It is a known issue that Kinect V2 often produces incorrect measurements near the borders of occluded surfaces, as shown in Fig. 11. We are concerned about locating two types of invalid points and removing them from the depth map.

Points visible by the camera but falling in the shadow of an IR emitter have a low accuracy. We call these points shadow points. Shadow points are due to the displacement between the IR emitter and the camera (Fig. 12), which is approximately $\Delta = 8$ cm. In this work we are less concerned about depth image restoration of the regions that are not directly observed by the camera, as in Liu et al. (2013), because the NBV system usually observes the same region of space from multiple viewpoints and the measured data are merged by KinFu.

To detect shadow points we look into the regions of occlusion where a background object is observed only by the camera, but that are not illuminated by the IR emitter (yellow areas). The geometry of the sensor field is illustrated in Fig. 13. Let u and v be the horizontal and vertical image

Journal: 10514 MS: 9618 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2017/1/23 Pages: 16 Layout: Large

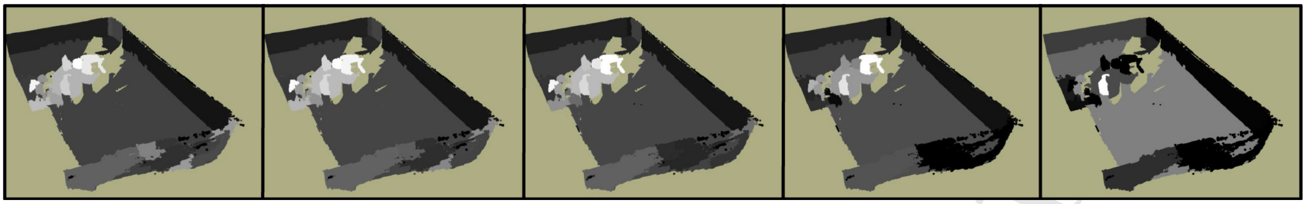


Fig. 10 Saliency computed after the initial scan in experiment 2 described in Sect. 4.2, using $B_{th} \in \{0.005, 0.01, 0.02, 0.05, 0.1(m)\}$ (from *left* to *right*)

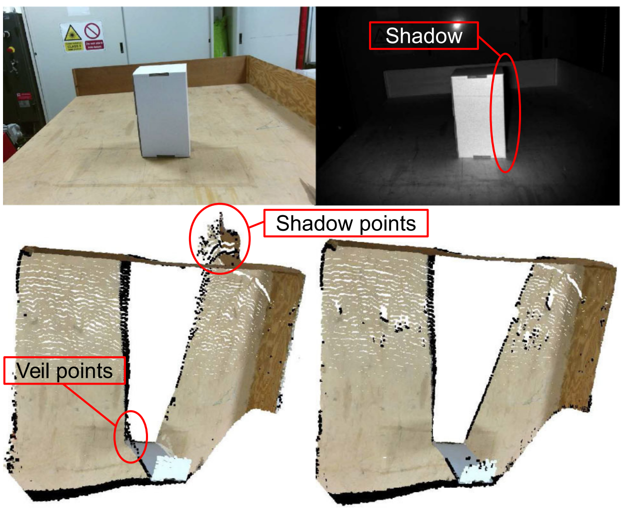


Fig. 11 *Top left* a scene as seen from the sensor. *Top right* the image from the depth camera. *Lower left* the point cloud acquired by the sensor, filtered by the Freenect2 driver. *Lower right* the point cloud filtered by our method; both shadow points and veil points are correctly removed

Fig. 12 The Kinect V2 sensor with IR camera, RGB camera and IR emitters

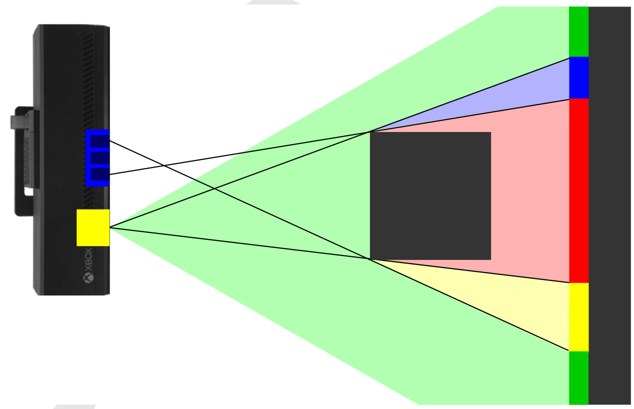


Fig. 13 The Kinect V2 (on the *left*) observes a scene composed by an object (in the *center*) and a background plane (on the *right*). The object partially occludes the background plane. Three kinds of occlusions are possible: camera only (*yellow*), IR emitter only (*blue*), both (*red*)

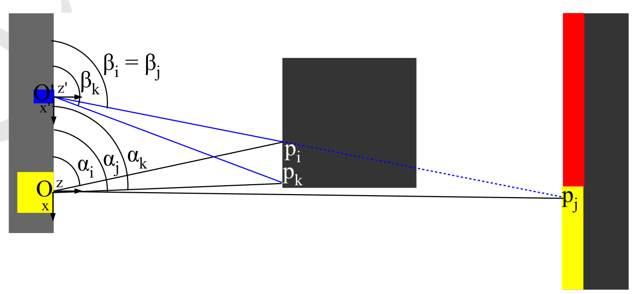


Fig. 14 Illustration of the horizontal angles α and β of the camera and IR emitter with respect to the observed points

coordinates of the sensor, starting from the upper left corner. Let also be the intrinsic parameters of the IR camera defined as follows: $[f_u, f_v]$ the focal lengths, $[m_u, m_v]$ the principal point, $[u_{max}, v_{max}]$ the depth image size and $[\Delta, 0]$ the displacement between the IR emitter from the IR camera, which are aligned horizontally. Given a measured distance z_{uv} along the sensor axis z at image coordinates $[u, v]$, the coordinates of the measured point referred to the IR camera are given by

$$\begin{aligned} x_{uv} &= \tfrac{u-m_u}{f_u} \cdot z_{uv} \\ y_{uv} &= \tfrac{v-m_v}{f_v} \cdot z_{uv} \end{aligned} \tag{13}$$

while the horizontal angle, shown in Fig. 14, referred to the camera is

$$\alpha_{uv} = \text{atan}\left(\frac{x_{uv}}{z_{uv}}\right) + \frac{\pi}{2} = \text{atan}\left(\frac{u - c_u}{f_u}\right) + \frac{\pi}{2} \tag{14}$$

which is monotonically increasing with respect to u. However, when referred to the leftmost IR emitter, the x coordinate becomes

$$x'_{uv} = \frac{u - m_u}{f_u} \cdot z_{uv} + \Delta \tag{15}$$

Journal: 10514 MS: 9618 TYPESET DISK LE CP Disp.:2017/1/23 Pages: 16 Layout: Large

Algorithm 2: Kinect V2 shadow points removal

```
Input: v: vertical coordinate
Input: z_uv: depth image
β'_max ← −∞
for u from 0 to u_max − 1 do
    x' ← (u − c_u)/f_u · z_uv + Δ
    β' ← x'/z_uv
    if β' ≤ β'_max then
        RemovePoint(u,v)
    else
        β'_max ← β'
    end if
end for
```

and the horizontal angle becomes

$$\beta_{uv} = \text{atan}\left(\frac{x'_{uv}}{z_{uv}}\right) + \frac{\pi}{2} \tag{16}$$

Unlike α_{uv}, the value of β_{uv} is not monotonically increasing with respect to u. It can be observed that an increase in u which causes a decrease in β_{uv} means that the depth measurement z_{uv} suddenly increased, i.e. the sensor is no longer observing an occluding object but the object behind it.

Let p_j be an observed point in the shadow of the IR emitter (yellow area in Fig. 14). Let also be α_j the angle from the camera origin, computed using Eq. 14. There exists a point p_i on the object along the illumination ray $\overline{O'p_j}$. There also exists a point p_k inside angle $\widehat{OO'p_j}$ that belongs to the object. At most we can choose $p_k \equiv p_i$. Since p_k belongs to $\widehat{OO'p_j}$, then $\beta_k \geq \beta_j$. Point p_k also belongs to the interior of angle $\widehat{O'Op_j}$ as the object does not intersect segment $\overline{Op_j}$. Then, $\alpha_k < \alpha_j$. Therefore, a necessary condition for a point p_j being in shadow is the existence of a point p_k that satisfies both $\beta_k \geq \beta_j$ and $\alpha_k < \alpha_j$. Thus, the depth measurements are removed if:

$$\beta_j \leq \max_{k \,|\, \alpha_k < \alpha_j} \beta_k \tag{17}$$

i.e., since α is monotonic with respect to u:

$$\text{atan}\left(\frac{x'_{uv}}{z_{uv}}\right) \leq \max_{k \in \{0..(u-1)\}} \text{atan}\left(\frac{x'_{kv}}{z_{kv}}\right) \tag{18}$$

which can be efficiently computed in parallel for each $v \in \{0..(v_{max} - 1)\}$ as shown in Algorithm 2.

Although it is very likely that a point p_k is observed by the sensor, since the object is near the sensor and the resolution is very high, condition 17 is still heuristic. Indeed, in real scenarios an object may be closer to the sensor than the Kinect V2 minimum range, hence p_k may not be really observed. Moreover, only a necessary condition was demonstrated. Indeed, some valid points may be misclassified as shadow points.

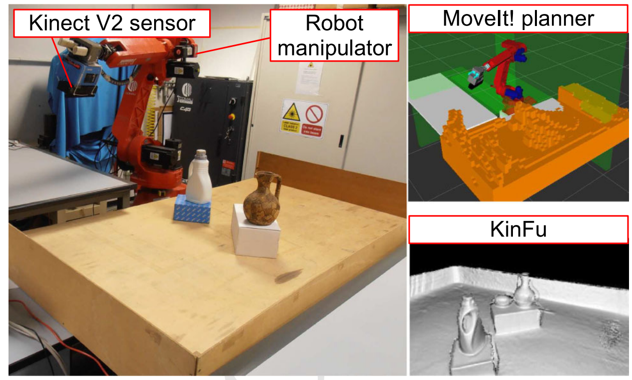


Fig. 15 The experimental setup (*left*). Motion planning environment based on Moveit! (*top right*). Screenshot of KinFu output during the initial scan phase (*bottom right*)

In the pre-processing phase invalid points called veil points are also removed as shown in Fig. 11. Veil points are caused by the lidar technology, which tends to interpolate points near the object border with the background. Veil points are removed if an angle higher than $\Theta_{max} = 10°$ is detected with respect to the observing ray. In particular, given a point p_i on the depth image, the point is removed if there is a point p_k in its Von Neumann neighborhood so that

$$\left|\frac{(p_k - p_i) \cdot p_i}{\|p_k - p_i\| \cdot \|p_i\|}\right| > \cos\left(\Theta_{max}\right) \tag{19}$$

4 Experimental evaluation

4.1 Robot setup and experimental procedure

The experimental setup (Fig. 15) used for the evaluation of the proposed NBV system consists of a robot arm (Comau SMART SiX) with six degrees of freedom. The robot has a maximum horizontal reach of about 1.4 m. A Kinect V2 sensor is mounted on the end-effector and it has been calibrated with respect to the robot wrist. The developed software runs under the ROS framework on an Intel Core i7 4770 at 3.40 GHz, equipped with an NVidia GeForce GTX 670. Collision free robot movements are planned using the MoveIt! ROS stack.

Occupied and unknown voxels are considered as obstacles in the motion planning environment. Experiments have been performed on a workspace of size 2 m×1.32 m. The volumetric representation of the environment within KinFu uses voxels of size 5.8 mm. In the motion planning environment voxels are undersampled to 4 cm. KinFu is fed with the robot forward kinematics as in Monica et al. (2016), Newcombe et al. (2011), Roth and Vona (2012) and Wagner et al. (2013) to improve the accuracy of point cloud registration with respect to the standard sensor ego-motion tracking approach.

The experimental procedure consists of the following steps. At the beginning of each experiment the environment is completely unknown and the robot, starting from a collision-free configuration, takes a short initial scan of the scene, from one side, using KinFu. Then, the system iteratively computes the NBV as described in Algorithm 1. If the motion planner finds a collision-free path the robot is moved to the NBV. Otherwise, the NBV is skipped. KinFu is turned on when the robot reaches each planned next-best view configuration. Since Kinect needs to be moved for KinectFusion to operate properly the sensor is slightly tilted around the NBV by rotating the robot wrist. The volumetric representation of the environment is, therefore, updated by KinFu after each observation. For the evaluation of the proposed approach for active exploration the experiments were concluded after the fifth NBV.

4.2 Experiments

Experiments have been performed in four different scenarios shown in Fig. 16. Each experiment contains multiple objects with complex geometry. In particular, in experiment 1 the environment comprises two stacks of objects, while experiment 2 has been performed in a cluttered scene with eight objects.

The performance of the proposed method was compared to a standard approach where the NBV is chosen at each iteration as the viewpoint that maximizes the size of the expected unknown volume of the whole environment that becomes visible. The standard approach has been developed by skipping the point cloud segmentation phase and by assigning the same saliency value to all points. A video of experiment 4 is available for download (http://rimlab.ce.unipr.it/documents/RMonica-auro-2016.avi).

Quantitative data about the average computational time for each phase are reported in Table 1. The average time for point cloud segmentation and saliency computation is about 23% of the total time. A first advantage of the proposed method is that it completes the five next-best views faster than the standard approach. The average times for motion planning and robot movement are rather similar, since these are fixed costs due to the experimental setup, as well as the running time for updating the collision map of the motion planning environment (planner map update). Also, the time required for viewpoint generation is very short (2.1 seconds for five views), since the computation is performed on the GPU directly on the TSDF volume. The running time required for the computation of the NBV is reported as a subtotal. It can be noted that for the NBV computation phase our method is 3.9 times faster than the standard approach,

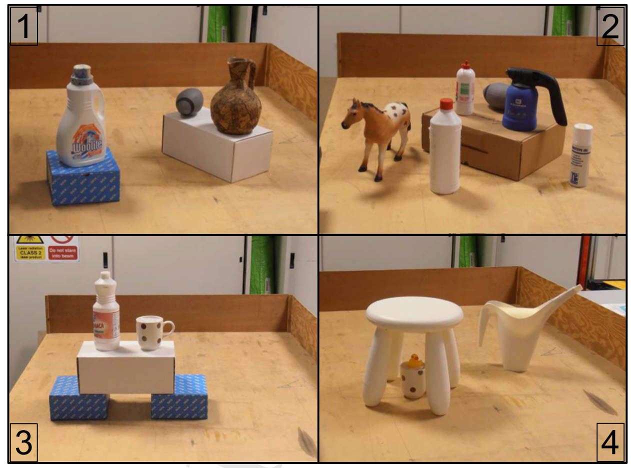


Fig. 16 The experimental scenarios used for the evaluation

Table 1 Average total time (seconds) and standard deviation over the four experiments for each phase

Phase	Method			
	Proposed		Standard	
Segm. + saliency	46.1	±2.5	–	
Views generation	2.1	±0.1	2.1	±0.1
Views evaluation	26.0	±4.1	288.5	±39.0
Subtotal	74.2	±4.8	290.6	±39.0
Planner map update	46.0	±1.6	44.6	±0.4
Motion planning	88.1	±17.9	78.3	±13.8
Robot motion	110.5	±3.9	108.5	±7.7
Total	318.7	±19.1	522.0	±42.0

even though the standard approach does not require point cloud segmentation and saliency evaluation.

The main difference between the two approaches in the time required to compute a NBV lies in the viewpoint evaluation phase. The standard approach evaluates all candidate viewpoints generated in the environment (on the order of thousands), most of which are located on the edges of the supporting table. On the contrary, being able to focus only on the most salient segments, the proposed method rarely evaluates more than two hundreds candidate viewpoints at each iteration. Indeed, the proposed method is strictly focused on the exploration of the salient segments, whose extension is smaller than the size of all the unknown regions of the environment.

In Fig. 17, an example of the generated viewpoints is shown. The total number of candidate viewpoints for all segments is 91960. Using the standard approach all viewpoints would be evaluated to find the optimal NBV. Instead, our method focuses only on the most salient segment and, therefore, only 960 viewpoints are evaluated. In this case, a reachable pose for the robot was found among these view-

Journal: 10514 MS: 9618 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2017/1/23 Pages: 16 Layout: Large

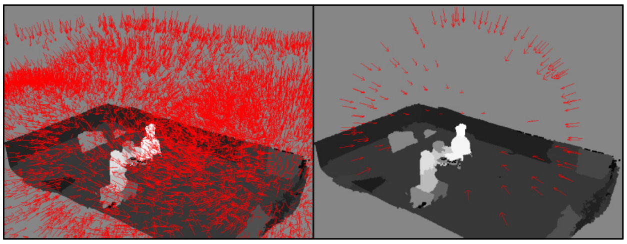


Fig. 17 Candidate viewpoints (represented by *arrows*) for the proposed approach (experiment 1, third NBV). *Left* candidate viewpoints of all segments. *Right* candidate viewpoints for the most salient segment only

Table 2 Saliency values and number of view poses for the point cloud segments in Fig. 17 (in descending order of saliency) up to the first segment not belonging to the objects (part of the supporting table)

Saliency	No. of poses	Description
0.589	960	Cork jug (top part)
0.565	3640	Cork jug (bottom part)
0.510	1680	Plastic jug (top part)
0.459	1560	Box under cork jug
0.424	600	Plastic jug (bottom part)
0.312	1080	Ball
0.300	400	Box under plastic jug
0.212	240	Box under plastic jug
0.177	320	Box under plastic jug
0.172	7720	Part of the table

points. If a reachable pose had not been found the system would have evaluated the second most salient segment, and so on. In Table 2 the saliency values of the point cloud segments are shown as well as the number of associated viewpoints. Had the algorithm tried other segments after the most salient one, the number of evaluated viewpoints would have increased up to 10,480, which is the total number of candidate viewpoints actually pointing towards the objects. The proposed saliency function is working properly even with some degree of over-segmentation by the LCCP algorithm. Indeed, some of the objects are segmented in multiple parts. For example, both jugs are split into two segments and one of the boxes is segmented into three parts. Nonetheless, each of those parts received a high saliency.

In Table 3 marks are reported that indicate whether each NBV points towards the objects or not. In the proposed approach all next-best views pointing towards the objects always occur before any other view, not focused on the objects. In the standard approach next-best views pointing towards the objects occur in an unpredictable order. Therefore, it is possible to conclude that a second and more important advantage of the proposed approach is that it allows a more rapid exploration of the objects thanks to point cloud segmentation and saliency evaluation at the segment level.

Table 3 Marks showing NBVs pointing towards the objects (✓) or not (×), for all the experiments

Method	Exp.	NBV				
		1	2	3	4	5
Proposed	1	✓	✓	✓	✓	×
	2	✓	✓	✓	×	×
	3	✓	✓	✓	×	×
	4	✓	✓	✓	✓	✓
Standard	1	×	✓	×	✓	×
	2	✓	×	×	×	✓
	3	×	×	✓	×	✓
	4	×	×	✓	×	×

This conclusion is also supported by the graphs in Fig. 18, which show the number of unknown residual voxels near the objects over the first five next-best views.

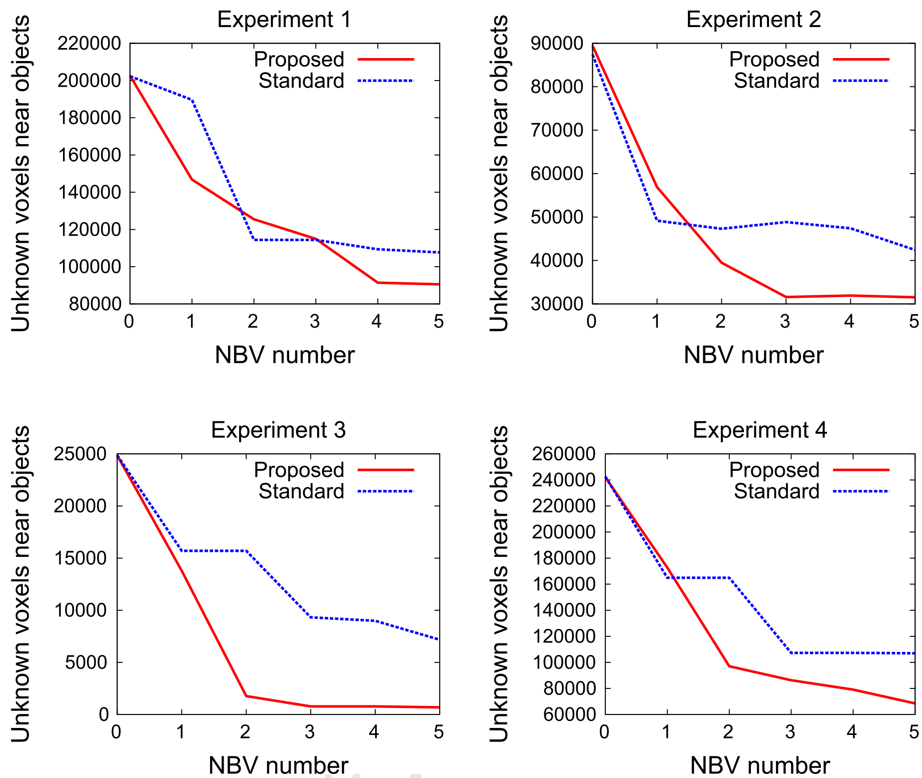
Images of the planned next-best views for experiment 1 are reported in Figs. 19 and 20. Images of the planned next-best views for experiment 3 are reported in Figs. 21 and 22. In experiment 1 the robot focuses on the objects for the first four views. Afterwards, as there are no reachable viewing poses to observe the right side of the objects, due to kinematic constraints, the robot explores a region of space that does not contain any object. In particular, the robot observes the space on the supporting table in the front of the objects, which is incomplete due to noise. A similar behavior is evident, for the proposed approach, in experiment 3. Conversely, it can be noted that the standard approach prioritizes exploration of the unknown voxels occluded by the objects as shown, for example, in the first two views of experiment 3. In the third view of experiment 3 the standard approach takes a frontal observation of the objects, but in the fourth view the robot observes again a region of the supporting plane without any object.

In some cases at the beginning of the exploration, after one or two next-best views, the standard approach achieves a lower number of unknown residual voxels. An example can be seen in Fig. 18 for experiment 1, after the second NBV. This is due to the fact that in the standard approach when the robot observes the unknown voxels occluded by the objects it also partially observes the back of the objects, since the sensor has a large field of view (70° × 60°).

The final voxel-based reconstruction is shown in Fig. 23 for all experiments. The reconstruction of the objects is always more complete for the proposed method. Some unknown voxels are still present, mostly due to unreachable poses aimed at observing the back or below the objects, as stated above. Also, it can be noted that most of the irrelevant voxels around the back panel of the scene remained unknown for the proposed method, while these voxels have been observed by the standard NBV approach.

Journal: 10514 MS: 9618 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2017/1/23 Pages: 16 Layout: Large

Fig. 18 The graphs show the number of unknown voxels near the objects in the scene for the first five next-best views



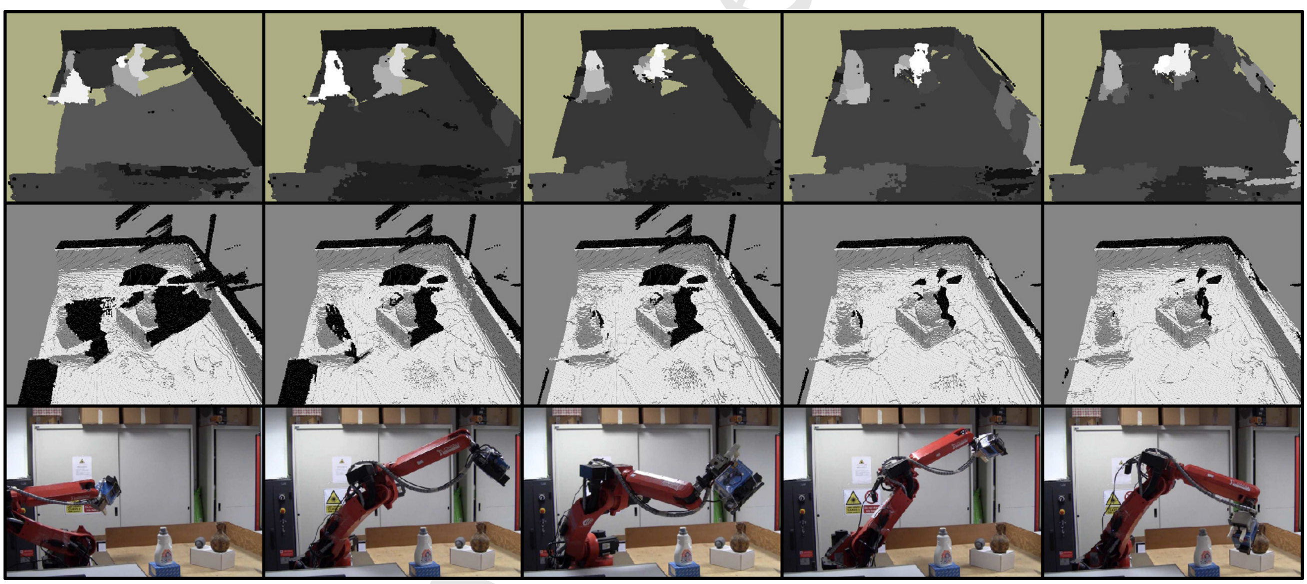


Fig. 19 Images of experiment 1 using the proposed method (*left* to *right*). *Top* saliency map of point cloud segments; *middle* 3D volumetric representation; *bottom* planned robot next-best views

4.3 Evaluation of depth image pre-processing

The proposed Kinect V2 depth image pre-processing filter (Sect. 3.3) has been evaluated in the scenario shown in Fig. 24 (top-left). The environment contains only planar surfaces to facilitate ground truth annotation. A bounding box was defined around the workspace to remove the background of the room. Thus, any point that does not belong to a plane can be considered as an outlier. Depth images were obtained by averaging 30 frames (one second) acquired by the sensor to simulate the noise-reduction effect of the KinFu algorithm. A maximum distance threshold of 3 cm was defined to consider a point as belonging to a plane.

In Fig. 24 it can be noted that our pre-processing method successfully removes the shadow on the left of the box-shaped object. The total number of false negatives, i.e. outlier

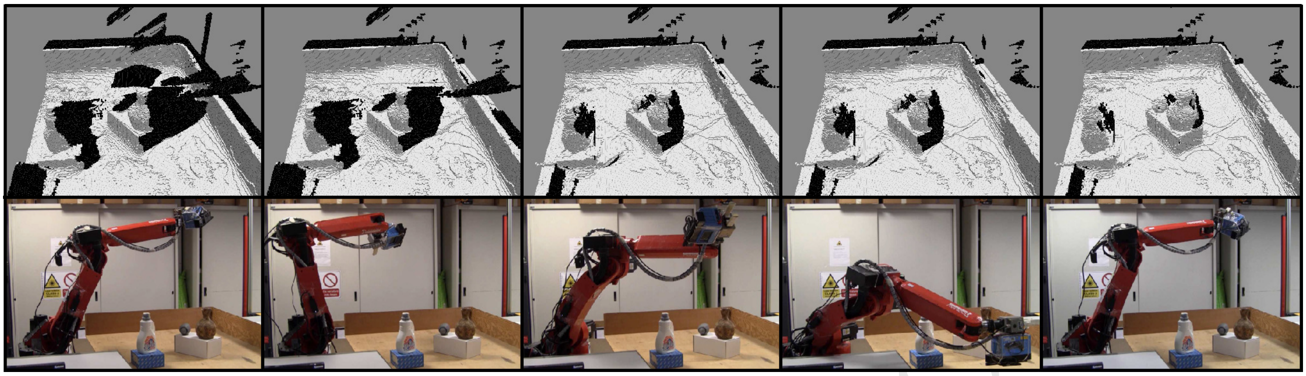


Fig. 20 Images of experiment 1 using the standard NBV approach. *Top* 3D volumetric representation; *bottom* planned robot next-best views

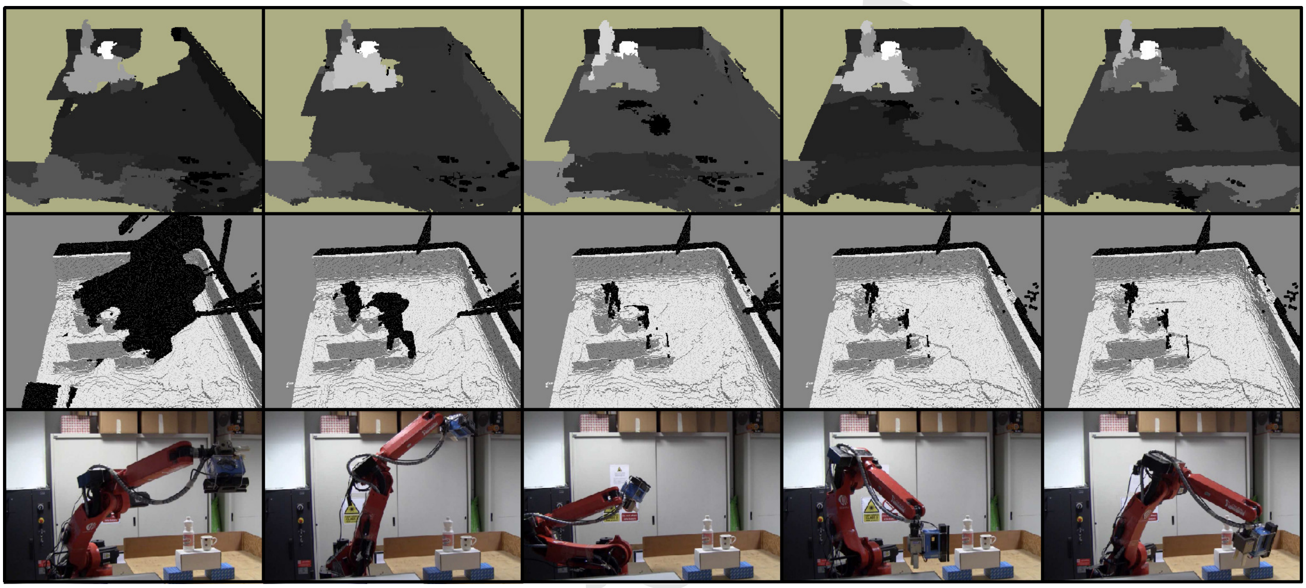


Fig. 21 Images of experiment 3 using the proposed method (*left* to *right*). *Top* saliency map of point cloud segments; *middle* 3D volumetric representation; *bottom* planned robot next-best views

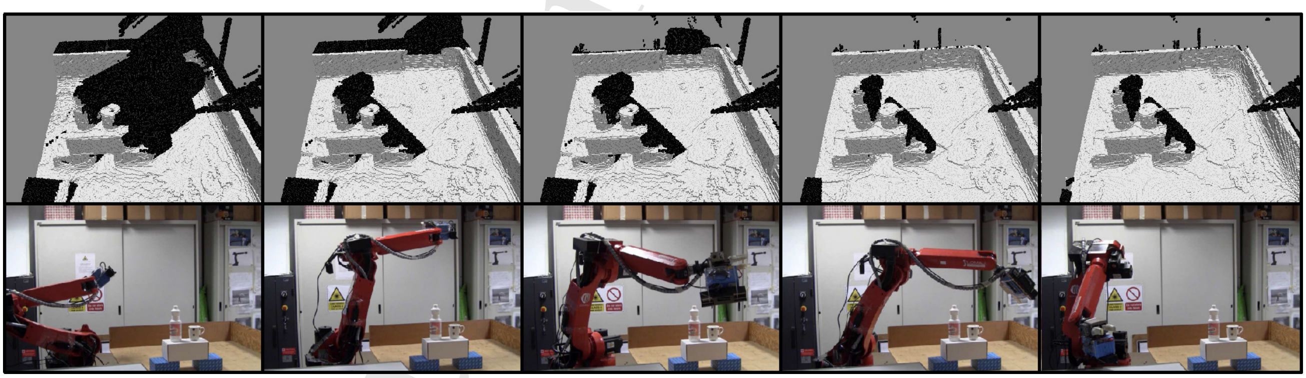


Fig. 22 Images of experiment 3 using the standard NBV approach. *Top* 3D volumetric representation; *bottom* planned robot next-best views

points not belonging to any plane, are reported in Table 4 as well as the number of measurements, i.e. the number of valid points reported by the algorithms. Our algorithm reports a significantly lower number of outliers compared to the standard filtering algorithms already available in the Freenect2 driver (a bilateral and an edge-aware filter). Being conservative, however, it also reports a slightly lower number of valid measurements.

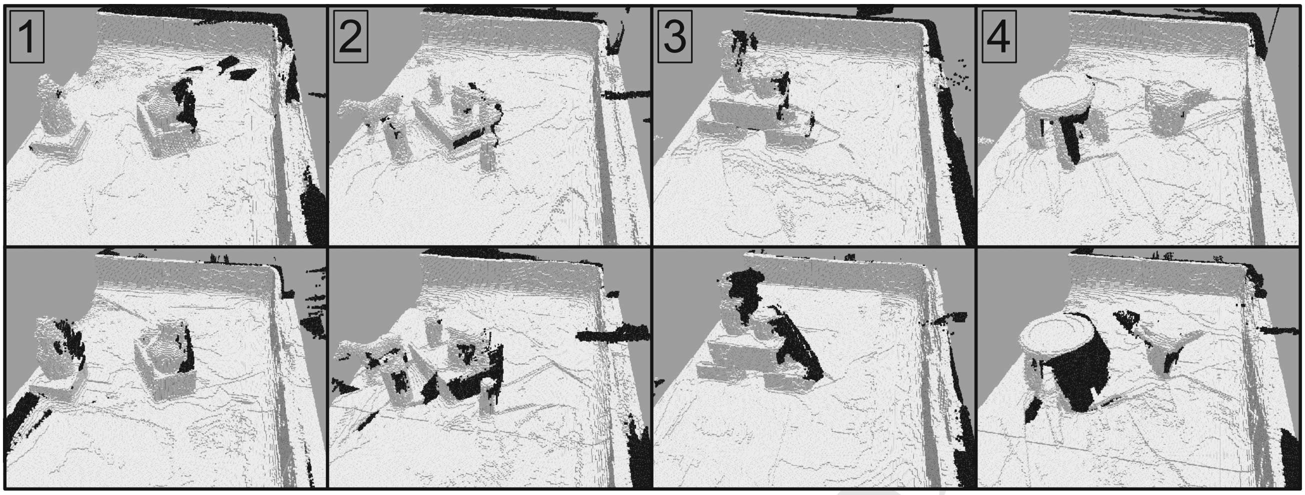


Fig. 23 3D volumetric representation of the environment in the four experiments after five next-best views: proposed method (*top*), standard approach (*bottom*)

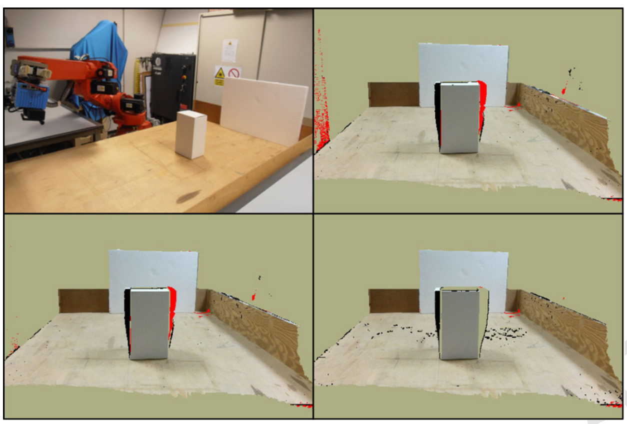


Fig. 24 *Top left* the scenario used for testing the proposed depth image pre-processing filter. *Top right* image preprocessed by the Freenect2 bilateral filter only. *Bottom left* image preprocessed by the Freenect2 bilateral and edge-aware filters. *Bottom right* image preprocessed by the Freenect2 bilateral filter and our filter. The image is displayed in color although the algorithm operates on the depth map only. Outliers points are displayed in *red*

Table 4 Number of measurements and false measurements produced by each algorithm

Method	Measurements	Outliers
Bilateral	83,125	1874
Bilateral + Edge-aware	81,611	849
Bilateral + Proposed filter	79,756	182

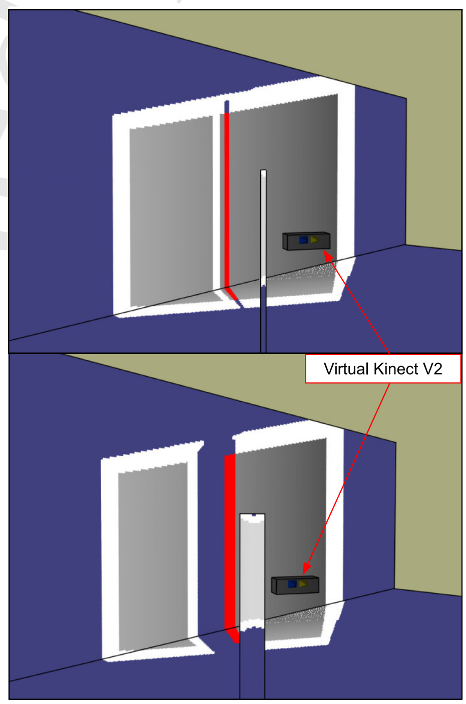


Fig. 25 The simulated environment, with object size 4 cm (*top*) and 16 cm (*bottom*). *White area* illuminated by the emitter only. *Grey area* illuminated and properly acquired by the camera. *Red shadow* visible by the camera. The *vertical blue band* in the bottom image is a region of space that is neither illuminated by the emitter nor observed by the camera

In Sect. 3.3 it was pointed out that the proposed filter for shadow points removal only provides a necessary condition and that false positives may still be present. Evaluation of false positives was carried out in a simulated environment shown in Fig. 25, which contains a ground plane, a wall, and an object (long box). The object is at a distance of 1.5 m from the wall and the Kinect V2 sensor is placed at 1 m from the object. The sensor view of the wall and ground plane is partially occluded by the object. The IR emitter and the camera were simulated as separate entities according to the Kinect V2 technical specifications. The shadow points

Table 5 False discovery rate of shadow points

Sensor angle (°)		−60	−30	0	30	60
Object width (cm)	4	41%	50%	50%	47%	42%
	8	0%	1%	4%	5%	7%
	16	1%	0%	0%	14%	5%

removal filter was tested by varying the width of the object and the observation angles of the sensor around the object. Table 5 reports the ratio between the incorrectly removed points and all the removed points (false discovery rate). For normal-sized object (8–16 cm width) the false discovery rate is low. However, for thin objects (4 cm width) as the one displayed in Fig. 25 (top) the false discovery rate is over 40%. This is due to the fact that light from the emitter can pass behind a thin object and illuminate part of the background which could be correctly perceived by the real sensor, but it is actually removed by the proposed filter. It may be noted, however, that this negative result is quite rare as it happens only if a thin object is in front of a far background; moreover, in these cases only the background region is affected.

5 Conclusions

In this work a novel formulation of the next-best view problem was presented that prioritizes active exploration of the objects without using any prior knowledge about the environment. The next-best view is selected among candidate viewpoints that observe the border of incomplete and salient regions of space. A point cloud segmentation algorithm was adopted to extract salient point cloud segments associated to the objects.

The proposed approach has some limitations and, therefore, a number of directions are open for future research. The heuristic for saliency evaluation has proven robust to detect common objects, however, thin objects or parts usually receive a low score. Hence, computation of segment saliency can be improved by considering more advance features. Following the results achieved in Tateno et al. (2015) and Uckermann et al. (2012, 2014) the quality and robustness of the point cloud segmentation phase can also be improved by performing a real-time segmentation. Indeed, real-time segmentation computed on the TSDF volume is a promising research line. Technical limitations of the current robotic setup are mainly due to the small workspace of the robot arm and to the minimum sensing distance of the Kinect sensor. Finally, a natural extension of this work is the inclusion of object recognition techniques based on point cloud segmentation (Varadarajan and Vincze 2011) and the application of the active perception system for intelligent robot manipulation.

References

Atanasov, N., Sankaran, B., Le Ny, J., Pappas, G. J., & Daniilidis, K. (2014). Nonmyopic view planning for active object classification and pose estimation. *IEEE Transactions on Robotics*, *30*(5), 1078–1090.

Banta, J. E., Wong, L. R., Dumont, C., & Abidi, M. A. (2000). A next-best-view system for autonomous 3-D object reconstruction. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, *30*(5), 589–598.

Beale, D., Iravani, P., & Hall, P. (2011). Probabilistic models for robot-based object segmentation. *Robotics and Autonomous Systems*, *59*(12), 1080–1089.

Chen, S. Y., & Li, Y. F. (2005). Vision sensor planning for 3-D model acquisition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *35*(5), 894–904.

Connolly, C. (1985). The determination of next best views. *IEEE International Conference on Robotics and Automation (ICRA)*, *2*, 432–435.

Drews, P., Núñez, P., Rocha, R., Campos, M., & Dias, J. (2013). Novelty detection and segmentation based on gaussian mixture models: A case study in 3D robotic laser mapping. *Robotics and Autonomous Systems*, *61*(12), 1696–1709.

Finman, R., Whelan, T., Kaess, M., & Leonard, J. J. (2013). Toward lifelong object segmentation from change detection in dense RGB-D maps. In *European conference on mobile robots (ECMR)*, pp. 178–185.

Foix, S.., Alenyà, G., Andrade-Cetto, J., & Torras, C. (2010). Object modeling using a ToF camera under an uncertainty reduction approach. In *IEEE International conference on robotics and automation (ICRA)*, pp. 1306–1312.

Herbst, E., Henry, P., & Fox, D. (2014). Toward online 3-D object segmentation and mapping. In *IEEE International conference on robotics and automation (ICRA)*, pp. 3193–3200.

Kahn, G., Sujan, P., Patil, S., Bopardikar, S., Ryde, J., Goldberg, K., et al. (2015). Active exploration using trajectory optimization for robotic grasping in the presence of occlusions. In IEEE international conference on robotics and automation (ICRA), pp. 4783–4790.

Kai, X., Hui, H., Yifei, S., Hao, L., Pinxin, L., Jianong, C., et al. (2015). Autoscanning for coupled scene reconstruction and proactive object analysis. *ACM Transactions on Graphics*, *34*(6), 177:1–177:14.

Kriegel, S. Bodenmuller, T., Suppa, M., & Hirzinger, G. (2011). A surface-based next-best-view approach for automated 3D model completion of unknown objects. In *IEEE international conference on robotics and automation (ICRA)*, pp. 4869–4874.

Kriegel, S., Rink, C., Bodenmuller, T., Narr, A., Suppa, M., & Hirzinger, G. (2012). Next-best-scan planning for autonomous 3D modeling. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 2850–2856.

Kriegel, S., Brucker, M., Marton, Z. C, Bodenmuller, T., & Suppa, M. (2013). Combining object modeling and recognition for active scene exploration. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 2384–2391.

Li, Y. F., & Liu, Z. G. (2005). Information entropy-based viewpoint planning for 3-D object reconstruction. *IEEE Transactions on Robotics*, *21*(3), 324–337.

Liu, S., Wang, Y., Wang, J., Wang, H., Zhang, J., & Pan, C. (2013). Kinect depth restoration via energy minimization with TV21 regularization. In *IEEE international conference on image processing (ICIP)*, pp. 724–724.

Monica, R., Aleotti, J., & Caselli, S. (2016). A KinFu based approach for robot spatial attention and view planning. *Robotics and Autonomous Systems*, *75*(Part B), 627–640.

Journal: 10514 MS: 9618 ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:2017/1/23 Pages: 16 Layout: Large

Morooka, K., Zha, Hongbin, & Hasegawa, T. (1998). Next best viewpoint (NBV) planning for active object modeling based on a learning-by-showing approach. In *Fourteenth international conference on pattern recognition*, Vol. 1, pp. 677–681.

Newcombe, R. A., Izadi, S., Hilliges, O., Molyneaux, D., Kim, D., Davison, A. J., et al. (2011). KinectFusion: Real-time dense surface mapping and tracking. In *IEEE international symposium on mixed and augmented reality (ISMAR)*, pp. 127–136.

Orabona, F., Metta, G., & Sandini, G. (2005). Object-based visual attention: a model for a behaving robot. In *IEEE computer society conference on computer vision and pattern recognition (CVPR)*, pp. 89–89.

Papon, J., Abramov, A., Schoeler, M., & Worgotter, F. (2013). Voxel cloud connectivity segmentation - supervoxels for point clouds. In *IEEE computer society conference on computer vision and pattern recognition*, pp. 2027–2034. doi:10.1109/CVPR.2013.264.

Patten, T., Zillich, M., Fitch, R., Vincze, M., & Sukkarieh, S. (2016). Viewpoint evaluation for online 3-D active object classification. *IEEE Robotics and Automation Letters*, *1*(1), 73–81.

Pito, R. (1999). A solution to the next best view problem for automated surface acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *21*(10), 1016–1030.

Potthast, C., & Sukhatme, G. S. (2014). A probabilistic framework for next best view estimation in a cluttered environment. *Journal of Visual Communication and Image Representation*, *25*(1), 148–164.

Reed, M. K., & Allen, P. K. (2000). Constraint-based sensor planning for scene modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(12), 1460–1467.

Roth, H., & Vona, M. (2012). Moving Volume KinectFusion. In *Proceedings of the British machine vision conference*. BMVA Press, pp. 112.1–112.11.

Stampfer, D., Lutz, M., & Schlegel, C. (2012). Information driven sensor placement for robust active object recognition based on multiple views. In IEEE international conference on technologies for practical robot applications (TePRA), pp. 133–138.

Stein, S. C., Worgotter, F., Schoeler, M., Papon, J., & Kulvicius, T. (2014). Convexity based object partitioning for robot applications. In IEEE international conference on robotics and automation (ICRA), pp. 3213–3220.

Tateno, K., Tombari, F., & Navab, N. (2015). Real-time and scalable incremental segmentation on dense SLAM. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4465–4472.

Torabi, L., & Gupta, K. (2010). Integrated view and path planning for an autonomous six-DOF eye-in-hand object modeling system. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4516–4521.

Tsuda, A., Kakiuchi, Y., Nozawa, S., Ueda, R., Okada, K., & Inaba, M. (2012). On-line next best grasp selection for in-hand object 3D modeling with dual-arm coordination. In IEEE international conference on robotics and automation (ICRA), pp. 1799–1804.

Uckermann, A., Haschke, R., & Ritter, H. (2012). Real-time 3D segmentation of cluttered scenes for robot grasping. In *12th IEEE-RAS international conference on humanoid robots (humanoids)*, pp. 198–203.

Uckermann, A., Eibrechter, C., Haschke, R., & Ritter, H. (2014). Real-time hierarchical scene segmentation and classification. In 14th IEEE-RAS international conference on humanoid robots (humanoids), pp. 225–231.

van Hoof, H., Kroemer, O., & Peters, J. (2014). Probabilistic segmentation and targeted exploration of objects in cluttered environments. *IEEE Transactions on Robotics*, *30*(5), 1198–1209.

Varadarajan, K. M., & Vincze, M. (2011). Object part segmentation and classification in range images for grasping. In 15th International conference on advanced robotics (ICAR), pp. 21–27.

Vasquez-Gomez, J. I., Lopez-Damian, E., & Sucar, L. E. (2009). View planning for 3D object reconstruction. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 4015–4020.

Wagner, R., Frese, U., & Bauml, B. (2013). Real-time dense multi-scale workspace modeling on a humanoid robot. In *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 5164–5171.

Walck G., & Drouin, M. (2010). Automatic observation for 3D reconstruction of unknown objects using visual servoing. In: *IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 2727–2732.

Welke, K., Issac, J., Schiebener, D., Asfour, T., & Dillmann, R. (2010). Autonomous acquisition of visual multi-view object representations for object recognition on a humanoid robot. In: *IEEE international conference on robotics and automation (ICRA)*, pp. 2012–2019.

Whaite, P., & Ferrie, F. P. (1997). Autonomous exploration: Driven by uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(3), 193–205.

Wu, K., Ranasinghe, R., & Dissanayake, G. (2015). Active recognition and pose estimation of household objects in clutter. In *IEEE international conference on robotics and automation (ICRA)*, pp. 4230–4237.

Yu, Yong, & Kamal, Gupta. (2004). C-space entropy: A measure for view planning and exploration for general robot-sensor systems in unknown environments. *The International Journal of Robotics Research*, *23*(12), 1197–1223.

Riccardo Monica is a Ph.D. student at the Department of Information Engineering of the University of Parma. In 2014 he received the Master degree in computer engineering. His main research interests lie in robot perception of 3D environments.

Jacopo Aleotti received his Laurea degree in Electronic Engineering and Ph.D. degree in Computer Engineering, both from the University of Parma, Italy, in 2002 and 2006, respectively. Since 2014 he is fixed-time assistant professor at the University of Parma. From 2006 to 2013, he conducted postdoctoral research with the Department of Information Engineering, University of Parma. In 2003, he was Marie Curie Fellow at the Learning System Laboratory, Orebro University, Sweden. His current research interests include robot learning from demonstration, robot manipulation and range sensing, human–robot interaction and virtual reality.

Journal: 10514 MS: 9618 TYPESET DISK LE CP Disp.:2017/1/23 Pages: 16 Layout: Large

Journal: 10514
Article: 9618

Author Query Form

Please ensure you fill out your response to the queries raised below and return this form along with your corrections

Dear Author

During the process of typesetting your article, the following queries have arisen. Please check your typeset proof carefully against the queries listed below and mark the necessary changes either directly on the proof/online grid or in the 'Author's response' area provided below

Query	Details required	Author's response
1.	Kindly check and confirm whether the corresponding author is correctly identified and amend if necessary.	
2.	Please confirm the inserted city name 'Parma' is correct and amend if necessary.	
3.	Kindly check and confirm the edit made in the figure captions are correct, amend if necessary.	
4.	As per the information provided by the publisher, Figs. 3, 7, 9, 13, 24 will be black and white in print; hence, please confirm whether we can add 'colour figure online' to the caption.	