Global Registration of Mid-Range 3D Observations and Short Range Next Best Views

Jacopo Aleotti[1] and Dario Lodi Rizzini[1] and Riccardo Monica[1] and Stefano Caselli[1]

***Abstract*— This work proposes a method for autonomous robot exploration of unknown objects by sensor fusion of 3D range data. The approach aims at overcoming the physical limitation of the minimum sensing distance of range sensors. Two range sensors are used with complementary characteristics mounted in eye-in-hand configuration on a robot arm. The first sensor operates at mid-range and is used in the initial phase of exploration when the environment is unknown. The second sensor, which provides short-range data, is used in the following phase where the objects are explored at close distance through next best view planning. Next best view planning is performed using a volumetric representation of the environment.**

A complete point cloud model of each object is finally computed by global registration of all object observations including mid-range and short range views. The global registration algorithm has also been compared to a standard sequential registration approach, which is more subject to error propagation. Experiments have been performed in environments with multiple rigid objects.

I. INTRODUCTION

A valuable application of 3D range sensing is active vision, a strategy where the sensor is mounted on the end effector of a robot arm to change the viewpoint. A large number of studies have investigated the problem of sensor based planning for eye-in-hand sensors. However, previous works did not consider that 3D sensors have a minimum sensing distance at which range data can be acquired. Thus, the available configuration space of the robot is reduced to those configurations where the sensor stays beyond the minimum distance from the object. Of course, this physical limitation is a serious drawback, especially for small robot manipulators. This paper proposes a method for robot exploration of unknown objects using a hybrid 3D sensing approach. The system is fully autonomous. The robot setup includes a small robot manipulator and two range sensors in eye-in-hand configuration. The two range sensors have complementary characteristics: one sensor provides mid-range data, while the second sensor provides short-range data. In the context of this work, mid-range is defined as being larger than about 0.7 m and shorter than 3 m, while short-range is defined as being shorter than 0.7 m. Figure 1 shows the end effector used for the experimental evaluation equipped with the two range sensors.

The task of the robot is to explore the observable objects in the environment and to generate a complete point cloud model of each object. Robot exploration consists of two phases. In the first phase, the robot arm is moved in the region in front of the scene so that an initial and naive observation of the environment is taken using the mid-range sensor alone (i.e. varying the viewpoint of the sensor). The first phase results in a frontal view of the scene. The mid-range observation is then clustered to extract an initial point cloud set of each object. In the second phase, a next best view planner (NBV) is run to perform a close range exploration of each object using the short-range sensor. The NBV planner relies on a volumetric representation of the environment, which is updated after each observation, and it generates a collision-free path for the robot. Finally, a complete point cloud model is generated for each object by global registration of all object observations including the mid-range observation and the short range views. In this work it is assumed that the objects are rigid, that they lie on a plane and that they do not touch each other. The approach has been evaluated in environments where additional obstacles make it impossible for the robot to explore the objects using only a single mid-range sensor. Likewise, environment exploration using only a short range sensor would be ineffective due to limited visibility.

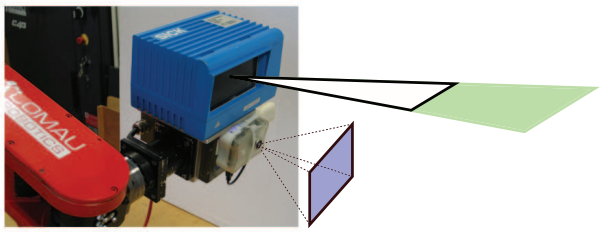

Fig. 1. Robot end effector equipped with middle range sensor (2D planar laser scanner, top) and short range camera (bottom).

The proposed approach is general as it does not depend on the choice of the eye-in-hand sensors (e.g. 3D stereo vision could be adopted). Additional contributions of the paper are as follows. The NBV planner uses a modification of the objective function presented in [2], [7] for viewpoint selection. The modified objective function balances the number of known and unknown voxels to ensure an overlapping area among the short-range views, that facilitates registration. Moreover, a parallel and efficient global registration algorithm (named p-ICP-GPA) has been developed for point cloud registration of the object views. Global registration has not been investigated together with NBV planning in previous works. The p-ICP-GPA algorithm improves the efficiency of the registration approach of Toldo et al. [24], based on Generalized Procrustes Analysis, while retaining

[1]Authors are with RIMLab - Robotics and Intelligent Machines Laboratory, Dipartimento di Ingegneria dell'Informazione, University of Parma, Italy, {aleotti,dlr,rmonica,caselli}@ce.unipr.it

its ability to perform simultaneous (i.e. global) registration of all partial views of an object.

Furthermore, the global registration approach has been compared to a sequential registration strategy of the object views based on the iterative closest point algorithm (ICP) [4]. Results confirm that the global registration approach is more accurate and more robust than the sequential registration strategy, which is more subject to error propagation. The paper is organized as follows. Section II describes related works about next best view planning for object exploration and methods for global registration of range data. Section III presents the proposed approach. Section IV illustrates the robot setup and reports the experimental results. Section V concludes the paper with some directions for future research.

II. RELATED WORK

Early works about NBV planning [2], [7], which were evaluated in simulation, introduced the concept of a viewing sphere for sensor placement around the object and defined an objective function for viewpoint selection that maximizes the unknown volume of the object. Improvements have been proposed for the objective function to facilitate registration that look for smooth regions of the object [18], or that enforce an overlap scanning constraint [19], as in this paper. However, in [19] an artificial approach is used for moving the object based on a turntable. In [28] the objective function minimizes the navigation distance of the robot. Reed et al. [21] proposed a NBV system based on a turntable that includes multiple constraints like sensor imaging, scene occlusion and sensor placement. Other approaches have been proposed using full 6D planning around the object [6], [14], [25], [32]. In particular, Chen et al. [6] used trend surface, but the method worked with simple objects. In [32] an information theoretic approach was proposed that explores the whole C-space of the robot rather than focusing explicitly on the objects. Torabi et al. [25] considered a single object in the environment. The assumption of having a single object in the environment appears recurrently also in [8], [16], [29]. In [8] a probabilistic approach was used that does not require accurate hand-eye calibration. Whaite et al. [31] adopted a parametric model of the objects based on superquadrics. Other works are more focused on object exploration using robot manipulation [12], [11], [26], [30]. Kriegel et al. [13] proposed a hybrid eye-in-hand system, based on NBV planning with multi-view recognition, which is the closest approach to the method presented in this paper. However, there are substantial differences since in [13] objects are modeled by registration of 3D measurements from one sensor only, without data fusion. Moreover, registration is performed using standard sequential ICP. Hence, none of the previous works adopted a hybrid sensory system like the one proposed in this work.

Most previous works on global registration of multiple views perform an initial pairwise alignment followed by a global registration phase [5], [10], [17], [20], [23], [27]. In particular, in [27] global registration is based on the vector field representation and it does not require a priori knowledge about relative transformation of views. In [5] a method is presented that optimizes the alignment of range scans and recovers the structure of articulated objects. Schmedding et al. [23] proposed a method that finds feature points of matching objects based on surface-approximating polynomials. In [15] a global multiview registration algorithm is proposed that does not require initial pairwise alignment, based on manifold optimization. However, correspondences are assumed to be known a priori. Finally, Toldo et al. [24] proposed an automatic global registration approach, based on the Generalized Procrustes Analysis, that does not require prior knowledge of the correspondences nor initial pairwise alignment. The algorithm has been evaluated only on synthetic data. In this work a parallel version, p-ICP-GPA, of the algorithm has been developed and evaluated in a physical robotic setup against a standard sequential registration algorithm.

III. METHOD

This section presents the method for object exploration using eye-in-hand sensors with different visibility ranges. The algorithm operates under some general assumptions on the experimental setup and the scene. The setup consists of a manipulator equipped with a mid-range sensor and a short-range sensor. The two sensors are placed on the robotic arm in eye-in-hand configuration. The objects to be observed lie on a plane inside a delimited region of the space with a minimum distance between each pair of objects. It is also assumed that there are no objects totally occluded by other objects. The manipulator may not be able to reach all the required viewpoints to achieve a complete observation with a single sensor since there are obstacles in the scene.

The exploration is organized into two main phases. The first phase is the mid-range observation and aims at detecting objects in the scene and at building the initial volumetric representation required for the subsequent close-range exploration. During the first phase, the robot takes an initial observation of the region where the objects are placed, while staying far from potential obstacle, resulting in a possibly incomplete representation of the environment. Obstacles and free space are represented using an occupancy grid map (described in the following). The second phase corresponds to the short-range observation, which is performed iteratively for each of the previously detected objects. Next best views are computed for efficient exploration. The robot plans and executes the motion to reach the corresponding desired configuration. The short-range observations are used to update the occupancy grid map and are collected together in order to be registered. Figure 2 provides an outline of the complete procedure, while each step is illustrated in the remaining of the section.

The initial observation is performed using the mid-range sensor (line 2 in Figure 2). The sensor measurements from the mid-range sensor are accumulated into a single point cloud $\mathcal{P}$. Such point cloud is used to extract clusters of points $\mathcal{P}_j^0 \in \mathcal{P}$ representing objects for $j = 1, \ldots, n_c$ (line 3 in Figure 2). Cluster extraction is performed by applying a

Algorithm 1: Multiple Range Object Observation

1: {Phase 1: mid-range observation}
2: $\mathcal{P} \leftarrow$ MoveAndAcquireMidRangeCloud();
3: $\{\mathcal{P}_j^0\}_{j=1,\dots,n_c} \leftarrow$ ClusterPointCloud($\mathcal{P}$);
4: initialize $\mathcal{M}$ with unknown voxels;
5: $\mathcal{M} \leftarrow$ ComputeOccupancyGridMap($\mathcal{P}$,$\mathcal{M}$);
6: $(\bar{\mathcal{M}}) \leftarrow$ ComputeReducedMap($\mathcal{M}$);
7: {Phase 2: short-range observation of each object $j-th$}
8: **for** each cluster $\mathcal{P}_j^0$ $(j = 1, \dots, n_c)$ **do**
9: $c \leftarrow 0$; $gain \leftarrow \infty$;
10: **while** $gain > gain_{thr}$ **do**
11: $\mathcal{V} \leftarrow$ ComputeViewpointList($\bar{\mathcal{M}}$,$\cup_{h=0}^{c}\mathcal{P}_j^h$);
12: $done \leftarrow false$;
13: **while** not $done$ and $\mathcal{V} \neq \emptyset$ **do**
14: $v \leftarrow$ pop_first($\mathcal{V}$);
15: $done \leftarrow$ PlanAndExecute(v,$\bar{\mathcal{M}}$);
16: **end while**
17: $c \leftarrow c + 1$;
18: $\mathcal{P}_j^c \leftarrow$ AcquireShortRangeCloud();
19: $\mathcal{M} \leftarrow$ ComputeOccupancyGridMap($\mathcal{P}_j^c$,$\mathcal{M}$);
20: $\bar{\mathcal{M}} \leftarrow$ ComputeReducedMap($\mathcal{M}$);
21: $gain \leftarrow$ComputeInformationGain(v); {eq. (5)}
22: **end while**
23: p-ICP-GPA($\{\mathcal{P}_j^h\}_{h=0,\dots,c}$); {global registration}
24: **end for**

Fig. 2. Outline of the algorithm.

flood fill algorithm after outlier removal and dominant plane extraction, as illustrated in [1]. The presence of obstacles and the limitation on the minimum and maximum range of the sensor may reduce the available configuration space of the robot and the available viewpoints. Hence, the clusters $\mathcal{P}_j^0$ tipically provide a partial representation of objects.

Next, an *occupancy grid map* is computed (line 5 in Figure 2), i.e. a data structure that represents the occupied, empty and unknown regions. In particular, an occupancy grid map can be represented as a set of *voxels* (or cells) $\mathcal{M} = \{m_1, m_2, \dots, m_{|\mathcal{M}|}\}$. In the following, the symbol m_i will refer also to the center of the corresponding voxel m_i when its meaning is clear from the context. Each voxel m_i is associated to its occupancy probability $p(m_i)$, i.e. the probability that voxel m_i is occupied. However, it is more convenient to operate with the *log-odds* function that is related to $p(m_i)$ by the equation $o(m_i) = \log \frac{p(m_i)}{1-p(m_i)}$.

The log-odds formulation is commonly used to update occupancy grid maps [9]. All the voxels in the initial occupancy grid map are set as unknown ($o(m_i) = 0$). Each point p of the point cloud $\mathcal{P}$ represents an observation z_p in the form of a ray $\overrightarrow{\text{origin}(p)p}$ from the sensor origin $\text{origin}(p)$ to p. The ray sensor model used in this work is defined by the conditional probability

$$p(m_i|z_p) = \begin{cases} \bar{p} & \text{if } m_i = \text{voxel}(p) \\ 1-\bar{p} & \text{if } m_i \text{ on ray } \overrightarrow{\text{origin}(p)p} \\ 0.5 & \text{otherwise} \end{cases} \quad (1)$$

where $0.5 \leq \bar{p} \leq 1$ and the log-odds is

$$o(m_i|z_p) = \begin{cases} \log \frac{\bar{p}}{1-\bar{p}} = u & \text{if } m_i = \text{voxel}(p) \\ \log \frac{1-\bar{p}}{\bar{p}} = -u & \text{if } m_i \text{ on ray } \overrightarrow{\text{origin}(p)p} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Thus, the updated log-odds $o'(m_i)$ of a voxel m_i after an observation z_p is $o'(m_i) = o(m_i) + o(m_i|z_p)$ (see e.g. [9]). In order to find the occupied voxel ($m_i = \text{voxel}(p)$) and the free voxels (m_i on ray) detected by observation z_p, a ray tracing procedure is performed on the occupancy grid map (lines 5 and 19 in Figure 2).

A reduced occupancy grid map $\bar{\mathcal{M}}$ is then derived from $\mathcal{M}$. The *reduced occupancy grid map* $\bar{\mathcal{M}}$ is the subset of the voxels m of the occupancy grid map $\mathcal{M}$ such that either m state is occupied ($o(m) > \epsilon_o$) or m state is unknown ($|o(m)| \leq \epsilon_o$) and its 6-neighbors are empty. The reduced occupancy grid map represents the voxels on the crust of the occupied and unknown region. Since range sensors observe the crust of objects, the reduced map provides the same information about the unobserved regions, but operating on a smaller data structure. The occupied and unknown voxels of the reduced occupancy grid map represent obstacles during motion planning. The computation of $\bar{\mathcal{M}}$ is shown at lines 6 and 20 in Figure 2.

Each cluster $\mathcal{P}_j^0$ represents an object to be explored. The aim of the proposed method is to collect and register several partial observations of each object. The next best view algorithm illustrated in the section III-A uses the reduced occupancy grid map $\bar{\mathcal{M}}$ to compute a list $\mathcal{V}$ of candidate viewpoints. The first element v of $\mathcal{V}$ is the candidate viewpoint that maximizes the objective function defined in eq. (4). Therefore, the planner computes a collision free path to v. If the planner fails or the path cannot be executed, the procedure is repeated extracting another viewpoint from $\mathcal{V}$ (lines 11-15 in Figure 2). After a new point cloud $\mathcal{P}_j^c$ is acquired from the viewpoint (line 18 in Figure 2), the occupancy grid map $\mathcal{M}$ is updated using again equation (2). The procedure stops when the information gain associated to the latest observation is less then the threshold $gain_{thr}$, as discussed in the next subsection. Finally, a global representation of the object is computed (line 23 in Figure 2) by registering $\mathcal{P}_j^0, \dots, \mathcal{P}_j^c$ using the p-ICP-GPA algorithm discussed in section III-B.

A. Planning short range next best views

The short-range observation of each object in the scene requires to compute the viewpoints for the sensor fixed on the manipulator end-effector. The candidate viewpoints are sampled on a sphere of constant radius r centered on the centroid $p_c = [x_c, y_c, z_c]^T$ of the cluster corresponding to the object to be observed. Centroid p_c is updated after each observation (Algorithm 2 in Figure 3, line 1). The viewpoints are obtained by discretizing the inclination θ and the azimuth φ. In particular, the samples are taken at $\theta = \Delta\theta\ i_\theta$ with $i_\theta = 0, \dots, \lfloor \theta_{max}/\Delta\theta \rfloor$ and $\varphi = -\pi + \Delta\varphi\ i_\varphi$ with $i_\varphi = 0, \dots, \lfloor 2\pi/\Delta\varphi \rfloor$. The uniform discretization

of inclination and azimuth corresponds to a non-uniform sampling of the sphere. However, the viewpoints that are close to already visited viewpoints are discarded during the execution of the task. Given a choice of r, θ and φ the resulting transformation of sensor frame S from base frame B is

$$ {}^{B}_{S}T_{p_c,r,\theta,\varphi} = \begin{bmatrix} \mathrm{c}\theta\mathrm{c}\varphi & \mathrm{s}\varphi & -\mathrm{s}\theta\mathrm{c}\varphi & x_c + r\mathrm{s}\theta\mathrm{c}\varphi \\ \mathrm{c}\theta\mathrm{s}\varphi & -\mathrm{c}\varphi & -\mathrm{s}\theta\mathrm{s}\varphi & y_c + r\mathrm{s}\theta\mathrm{s}\varphi \\ -\mathrm{s}\theta & 0 & -\mathrm{c}\theta & z_c + r\mathrm{c}\theta \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

where $\mathrm{c}\cdot$ and $\mathrm{s}\cdot$ are abbreviations of sine and cosine functions. The corresponding configuration of the manipulator is ${}^{B}_{W}T = {}^{B}_{S}T\ {}^{S}_{W}T$ and can be computed as the pose ${}^{W}_{S}T$ of the sensor frame S from the wrist frame W is known. Since the robot inverse kinematics is available, the next best view planner is able to filter all the unreachable robot configurations ${}^{B}_{W}T$ without performing a computationally expensive motion planning query. The apex and subscript ${}^{B}_{S}\cdot$ will be omitted in the following when no ambiguity arises. The generation of the candidate viewpoints is shown in Algorithm 2 (Figure 3) at lines 1-10. The sensor model that predicts the scene region observed from a viewpoint is a ray-box intersection model. A range sensor can be described by a set of ray directions $\mathcal{D}=\{\overrightarrow{d}_j\}_j$ and the values of minimum and maximum ranges. A voxel m_i is observable from a given viewpoint v if m_i is the closest voxel intersected by a ray $\overrightarrow{d}_j \in \mathcal{D}$ and the distance between v and m_i lies in the range interval of the sensor. Thus, each viewpoint v is defined by the corresponding transformation matrix T_v and the set of voxels $\mathcal{O}_v \subset \bar{\mathcal{M}}$ that are visible from v. The algorithm could choose the viewpoint v_{best} that allows the observation of the largest unexplored region of the object, i.e. $v_{best} = \mathrm{argmax}|\mathcal{O}^u_v|$ where $\mathcal{O}^u_v = \{m_i \in \mathcal{O}_v | |o(m_i)| \leq \epsilon_o\}$ is the set of unknown voxels observable from v. Such criterion reduces the number of observations required to achieve a complete model of the object, but does not take into account that the acquired point clouds $\mathcal{P}^0_j, \ldots, \mathcal{P}^c_j$ must be registered. Indeed, registration requires that each view is partially overlapped with some other views. The overlap region of a view v is estimated by the set of occupied voxels $\mathcal{O}^o_v=\{m_i \in \mathcal{O}_v | o(m_i) > \epsilon_o\}$. Thus, the optimal view can be selected as

$$ v_{best} = \mathrm{argmax}_v \left(\alpha_u\ |\mathcal{O}^u_v| + \alpha_o\ |\mathcal{O}^o_v|\right) \quad (4) $$

The parameters α_u and α_o weight the importance of occupied cells w.r.t. the unknown ones.

The observation of a single object is performed until there is a view that can improve the representation of the object. The *information gain* measures the information acquired in the estimation of occupancy grid map $\mathcal{M}$ after the new observation z_v from viewpoint v and is used as a criterion to stop the close range exploration of an object (line 10 in Figure 2). The value of such parameter depends on the occupancy probability of the voxels before and after the observation z_v, i.e. respectively from $p(m_i)$ and from

Algorithm 2: ComputeViewpointList($\bar{\mathcal{M}}$,$\mathcal{P}$)

Input: $\bar{\mathcal{M}}$: reduced occupancy grid map;
$\mathcal{P}$: the point cloud related to the object;
Input: parameters: viewing sphere radius r, inclination step $\Delta\theta$, maximum inclination θ_{max}, azimuth step $\Delta\varphi$, sensor rays $\mathcal{D}$, minimum and maximum sensor ranges r_{min} and r_{max};
Output: $\mathcal{V}$: sorted list of candidate viewpoints (best view first);

1: $p_c \leftarrow$ centroid($\mathcal{P}$); $\mathcal{V} \leftarrow \emptyset$;
2: **for** $i_\theta \leftarrow 0$ to $\lfloor \frac{\theta_{max}}{\Delta\theta} \rfloor$ **do**
3: **for** $i_\varphi \leftarrow 0$ to $\lfloor \frac{2\pi}{\Delta\varphi} \rfloor$ **do**
4: $\theta \leftarrow \Delta\theta i_\theta$; $\varphi \leftarrow -\pi + \Delta\varphi i_\varphi$;
5: $v \doteq (T_v, \mathcal{O}_v) \leftarrow (T_{p_c,r,\theta,\varphi}, \emptyset)$; {eq. (3) for T_v}
6: **if** is_reachable(T_v) **then**
7: $\mathcal{V} \leftarrow \mathcal{V} \cup \{v\}$;
8: **end if**
9: **end for**
10: **end for**
11: **for** $v \in \mathcal{V}$ **do**
12: {estimate the number of occupied and unknown voxel}
13: **for** each ray $\overrightarrow{d}_j \in \mathcal{D}$ **do**
14: $d_{min} \leftarrow \infty$;
15: **for** $m_i \in \bar{\mathcal{M}}$ **do**
16: $(inters, p_{inters}) \leftarrow$ intersect_ray(T_v, $\overrightarrow{d}_j$,m_i);
17: **if** $inters$ and $|p_{inters} - pos(T_v)| < d_{min}$ **then**
18: $d_{min} \leftarrow |p_{inters} - pos(T_v)|$; $m_{min} \leftarrow m_i$;
19: **end if**
20: **end for**
21: **if** $m_{min} \notin \mathcal{O}_v$ and $r_{min} \leq d_{min} \leq r_{max}$ **then**
22: $\mathcal{O}_v \leftarrow \mathcal{O}_v \cup \{m_{min}\}$;
23: **end if**
24: **end for**
25: **end for**
26: sort $\mathcal{V}$ according to eq.(4);

Fig. 3. Procedure to compute the next best view.

$p(m_i|z_v)$. The information gain is defined as

$$ G(m_1, \ldots, m_{|\mathrm{M}|}|z_v) = \sum_{m_i \in \mathcal{O}_v} -p(m_i)\ \log\left(\frac{p(m_i|z_v)}{p(m_i)}\right) \quad (5) $$

Although $G(m_1, \ldots, m_{|\mathrm{M}|}|z_v)$ is a function of all voxel states, its value depends only on the voxels whose occupancy probability is affected by the observation z_v, i.e. m_i s.t. $p(m_i) \neq p(m_i|z_v)$.

B. Global registration of range data

A parallel version of the registration algorithm [24], called p-ICP-GPA, has been developed for global point cloud alignment. The p-ICP-GPA algorithm, shown in Figure 4, performs a simultaneous registration of all the partial views of an object based on the Generalized Procrustes Analysis. The approach detects independent sets $\mathcal{A}_h$ of points belonging to the different views $\{\mathcal{P}^i\}^c_{i=1}$. An independent set $\mathcal{A}_h$ is such that each point $p \in \mathcal{A}_h$ is mutually nearest neighbor of all the other points in $\mathcal{A}_h$. Given two (not necessarily consecutive) views of an object with indices i and j, a point $p \in \mathcal{P}^i$ is mutually nearest neighbor to another point $p_{nearest} \in \mathcal{P}^j$ if the closest neighbor $\in \mathcal{P}^j$ of p is $p_{nearest}$ and the closest neighbor $\in \mathcal{P}^i$ of $p_{nearest}$ is p.

Algorithm 3: p-ICP-GPA($\{\mathcal{P}^i\}_{i=1}^c$)

Input: $\{\mathcal{P}^i\}_{i=1}^c$: input cloud clusters;
Input: parameters: maximum point association distance d_{max};
Output: $\{{}_i^R T\}_{i=1}^c$: the transformation matrices to common reference frame R;

```
1: while not done do
2:    G ← ∅; C ← ∅;
3:    for each cluster P^i, i = 1,...,c do
4:       for each point p ∈ P^i do
5:          {lines 7-13 executed by parallel threads}
6:          for each cluster P^j, j = i+1,...,c do
7:             p_nearest ← FindNearestInCluster(p,P^j);
8:             if |p − p_nearest| < d_max then
9:                p' ← FindNearestInCluster(p_nearest,P^i);
10:               if p' = p then
11:                  G ← G ∪ {(p, p_nearest), (p_nearest, p)};
12:               end if
13:            end if
14:         end for
15:      end for
16:   end for
17:   A ← FindIndependentSets(G); {find complete graphs in G}
18:   for each A_h ∈ A do
19:      C ← C ∪ { ComputeCentroid(A_h) };
20:   end for
21:   {line 22 computes transformations according to [24]}
22:   {}_i^R T}_{i=1}^c ← ComputeTranformations(C,A);
23:   for each cluster P^i, i = 1,...,c do
24:      P^i ← TransformCloud(P^i, {}_i^R T);
25:   end for
26: end while
```

Fig. 4. Parallel ICP-GPA.

The p-ICP-GPA parallelizes the detection of the independent sets by distributing the points of each view among multiple threads (lines 7-13). Every point of an independent set $\mathcal{A}_h$ is driven towards the centroid of the points of the set by solving a least square optimization problem that is run iteratively until global convergence (line 22). The outcomes are the rigid transformations ${}_i^R T$ (line 24) to be applied to each view $\mathcal{P}^i$ of an object.

Fig. 5. Objects used in the experiments.

IV. EXPERIMENTAL EVALUATION

A. Experimental Setup

The robotic setup consists of a six degrees of freedom robot arm (Comau SMART SiX) with a horizontal reach of 1.4 m and two range sensors mounted in eye-in-hand configuration. The mid-range sensor is a planar (2D) laser scanner (SICK LMS 400) with a minimum and maximum sensing distance of 0.7 m and 3 m respectively. The laser scanner has a high accuracy of ± 4 mm and it is used with a field of view of 70° and a scanning frequency of 190 Hz. The second sensor is the Dinast IPA-1110 short range camera which has a sensing range of 0.1 m to 0.8 m, a field of view of 80° and a resolution of 320×240 pixels. The short range sensor is one order of magnitude less accurate than the mid-range sensor. Drawing a comparison to a well known sensor, the Dinast camera is lighter and smaller than the Kinect sensor but less accurate (the minimum sensing distance of the Kinect is 0.8 m).

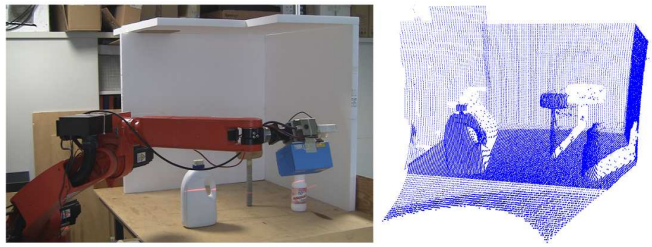

Fig. 6. Mid-range observation of the environment (left) and the acquired point cloud (right).

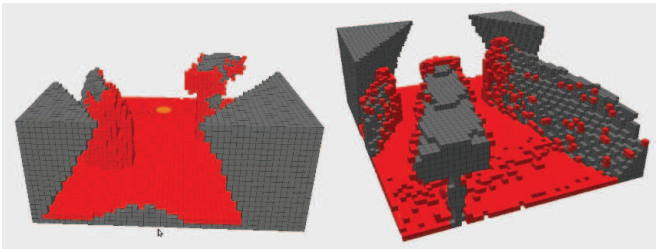

Fig. 7. Front (left image) and back (right image) views of the occupancy grid map computed from the mid-range observation in a first environment. Unknown voxels are colored in grey, occupied voxels are colored in red.

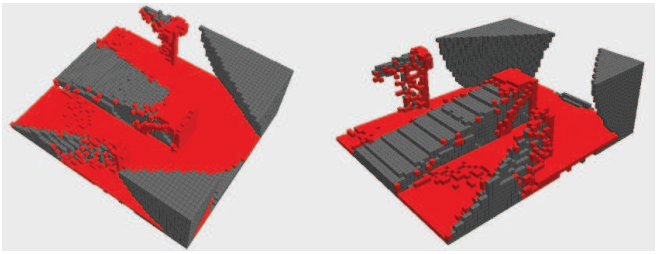

Fig. 8. Front (left image) and back (right image) views of the occupancy grid map computed from the mid-range observation in a second environment.

B. Experiments

Experiments of autonomous object exploration have been performed in environments with multiple rigid objects. Figure 5 shows the objects used in the experiments: a jug, two plastic bottles, a hammer, a toy horse and a box. Obstacles have been added at the back of the scene and their 3D models have been included in the simulation environment

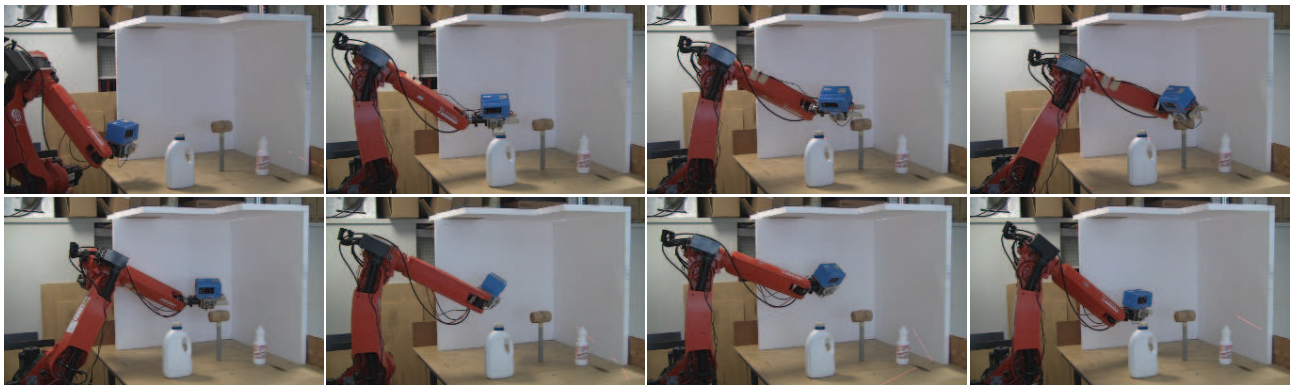

Fig. 9. Exploration of the objects in the first environment using the NBV planner and the short range sensor. Exploration of the jug (top row) and exploration of the hammer (bottom row). A complete experiment is also reported in the accompanying video.

TABLE I

PROCESSING TIMES (SECONDS) FOR OBJECT EXPLORATION.

Step	Phase	Time
Mid-range observation	1	44 s
Computation of occupancy grid map	1	37 s
Computation of NBV (short range)	2	13 s
Motion planning and execution	2	25 s
Occupancy grid map update	2	4 s

TABLE II

PROCESSING TIMES (SECONDS) FOR P-ICP-GPA AND SEQUENTIAL ICP.

algorithm	1 thread	4 thread	8 thread
ICP	15.4 s	-	-
p-ICP-GPA	27.7 s	9.6 s	8.8 s

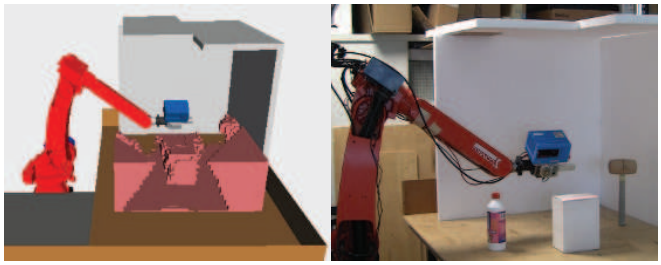

Fig. 10. Simulation environment used for planning short-range next best views (left column) and real planned configurations (right column) in the second environment.

used for NBV planning, which is based on the OpenRAVE engine [3]. Since the robot can not explore the whole environment using only the mid-range sensor, the only way to observe the back part of the objects is to take short-range observations. Moreover, it must be remarked that exploring the environment by using the short-range sensor alone would be a rather weak solution. Indeed, if only the short-range sensor was available the robot would not have any initial clue about the location of the objects and, therefore, the robot would need to perform a time consuming brute-force exploration of the whole environment. Figure 6 shows an image of the first phase of a robot exploration task, i.e. the mid-range observation of the environment using the planar laser scanner. In this phase the robot is programmed to take an initial frontal observation of the region where the objects are located. Thanks to the large field of view of the mid-range laser scanner there is no need, in this phase, to perform NBV planning. Figure 6 also shows the raw point cloud obtained from the mid-range observation. The raw point cloud contains on average 55K points. Point cloud processing is based on the Point Cloud Library [22].

Figures 7 and 8 illustrate two examples of the reduced occupancy grid map computed after the initial mid-range observation of the environment. The unknown voxels are colored in grey, while the occupied voxels are colored in red. The reduced occupancy grid map has the same visual appearance of the occupancy grid map. In the current setup, the size in meters of the occupancy grid map is $1.40 \times 1.20 \times 0.60$ and the resolution is 0.02 m (voxel size). Hence, the occupancy grid map contains a total of about $130K$ voxels. After the initial mid-range observation, the occupancy grid map contains on average 5% of occupied voxels, 50% of empty voxels and 45% of unknown voxels. In the reduced occupancy grid map the number of unknown voxels is reduced, on average, by 67%. Figures 9 and 10 show examples of the second phase of the robot exploration task, i.e. the close-range observation of the objects using the short-range sensor. In particular, Figure 9 shows a sequence of planned short-range next best views of two objects in the first environment. Figure 10 shows the 3D simulation environment used for planning the short-range next best views. The reduced occupancy grid map is imported in the motion planning environment and all the occupied and unknown voxels (displayed in reddish color) are considered as obstacles for the robot. The radius of the viewing sphere for short range exploration is $r = 0.3$ m in all the experiments. Parameters of equation (4) have been set to $(\alpha_u, \alpha_o) = (4, 1)$.

The average processing times of the steps for object exploration are reported in Table I. Timings for the short-range exploration (phase two) are for a single iteration. Experiments have been performed on an Intel Core2 Quad CPU at 2.83 GHz. The robot is moved at relatively low speed during both the first phase (to increase accuracy of mid-range laser scans) and second phase (due to motion in

TABLE III

AVERAGE ERROR AND STANDARD DEVIATION FOR REGISTRATION EXPERIMENTS (OBJECTS IN FIGURE 11).

object	p-ICP-GPA avg. error (std) (cm)	ICP avg. error (std) (cm)
jug	1.7(1.4)	3.1(3.2)
mallet	1.6(1.2)	2.9(3.0)
box	1.8(1.5)	3.0(4.0)
horse	1.9(1.7)	3.4(3.0)
bottle	1.6(0.8)	2.8(3.1)

narrow regions). The time required to update the occupancy map after each short range observation is significantly lower than the time required for the computation of the initial occupancy grid map from mid-range observation. The total time for exploration of a single object is about five minutes, which has been estimated by considering an average of five short-range views. The total time for the computation of a complete point cloud model of an object includes the time for object exploration and the time required for global point cloud registration, which is discussed at the end of this section.

The mid-range views of each object, extracted from the raw point cloud, and the short-range views are undersampled and contain on average $2K$ points. Figure 11 shows examples of the point cloud model of some objects before alignment and after the application of global registration algorithm. Even though the benefit of the p-ICP-GPA algorithm is noticeable, the quality of the registered point cloud is not uniform due to the different accuracy of the two range sensors.

The p-ICP-GPA global registration algorithm has been compared to a standard sequential algorithm based on ICP [22], that registers two consecutive views at a time. The sequential strategy is not optimal due to error propagation. The complexity of the p-ICP-GPA algorithm is $O(w^2 \cdot n \cdot log(n))$, n being the number of points and w the number of views, while the complexity of sequential ICP is $O(w \cdot n \cdot log(n))$. Table II reports the processing time of the registration algorithms in a point cloud registration experiment with 5 views with a total of $15K$ points. These experiments have been performed on an Intel Core i7 CPU at 2.2 GHz to test the performance up to 8 threads. Although there is no control on thread scheduling and the effectiveness of hyperthreading, the experiment suggests that the parallel execution speeds up registration thus making p-ICP-GPA faster than sequential ICP.

The p-ICP-GPA algorithm is faster than the sequential ICP algorithm when using multiple threads. Moreover, the speedup of p-ICP-GPA is 3.14 using 8 concurrent threads with respect to a single threaded execution. The p-ICP-GPA and the sequential ICP algorithms have also been compared in terms of robustness and accuracy in both real and synthetic experiments. A total of 30 registration experiments were generated for the objects in Figure 5 (containing an average of 6 views). Robustness was measured by the convergence rate, accuracy was measured manually by the average error to the ground truth of the distance of pairs of keypoints for each object. Convergence was assessed with a supervised process, by checking whether the correct alignment was achieved. Results confirm that registration with global p-ICP-GPA algorithm (3 registration failures) is more robust than registration with sequential ICP (8 registration failures) and more accurate as shown in Table III.

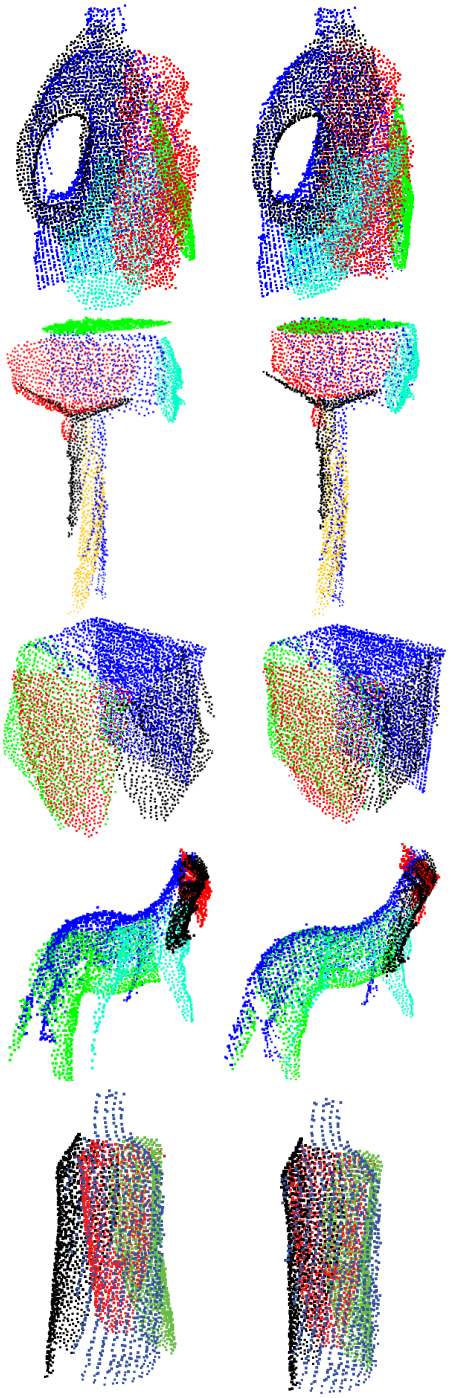

Fig. 11. Examples of point cloud registration before (left) and after (right) p-ICP-GPA global registration. Object views have different colors. Mid-range observation is in blue.

V. CONCLUSIONS

In this paper a method for object exploration has been presented based on hybrid sensor fusion. Two complementary sensors have been used to overcome the physical limitation of the minimum sensing distance typical of range sensors. The proposed approach is fully autonomous and it is general in the sense that it does not depend on the choice of the sensors. Close range exploration is guided by a next best view planner based on volumetric data. Moreover, a parallel algorithm for global registration of multiple views, based on Generalized Procrustes Analysis, has also been developed and evaluated. In the current setup a highly accurate 3D range scanner is used for mid-range observation and a less accurate short-range camera for close observations. The benefits of using such a hybrid method for object exploration will certainly increase with the advent of more accurate and established short-range sensors. Future work will investigate strategies to cope with more cluttered environments and strong occlusions. Object grasping and manipulation will also be included as they would be helpful to improve the autonomous exploration capabilities of the robot.

REFERENCES

[1] J. Aleotti, D. Lodi Rizzini, and S. Caselli. Object Categorization and Grasping by Parts from Range Scan Data. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2012.

[2] J.E. Banta, L. R. Wong, C. Dumont, and M.A. Abidi. 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, 2000.

[3] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner. Grasp planning in complex scenes. In *7th IEEE-RAS Int'l Conference on Humanoid Robots*, pages 42–48, nov. 2007.

[4] P.J. Besl and Neil D. McKay. A Method for Registration of 3-D Shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[5] W. Chang and M. Zwicker. Global Registration of Dynamic Range Scans for Articulated Model Reconstruction. *ACM Trans. Graph.*, 30(3):26:1–26:15, May 2011.

[6] S.Y. Chen and Y.F. Li. Vision Sensor Planning for 3-D Model Acquisition. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 35(5):894–904, 2005.

[7] C. Connolly. The Determination of Next Best Views. In *IEEE Intl Conference on Robotics and Automation (ICRA)*, volume 2, pages 432–435, 1985.

[8] S. Foix, G. Alenyà, J. Andrade-Cetto, and C. Torras. Object Modeling Using a ToF Camera under an Uncertainty Reduction Approach. In *IEEE Intl Conference on Robotics and Automation (ICRA)*, pages 1306–1312, 2010.

[9] A. Hornung, K.M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013.

[10] D.F. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. *Image and Vision Computing*, 21(7):637–650, 2003.

[11] D. Kraft, R. Detry, N. Pugeault, E. Başeski, F. Guerin, J.H. Piater, and Norbert Kruger. Development of Object and Grasping Knowledge by Robot Exploration. *IEEE Transactions on Autonomous Mental Development*, 2(4):368–383, 2010.

[12] M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and Object Tracking for In-Hand 3D Object Modeling. *The Intl Journal of Robotics Research*, 30(11):1311–1327, September 2011.

[13] S. Kriegel, M. Brucker, Z. C. Marton, T. Bodenmuller, and M. Suppa. Combining Object Modeling and Recognition for Active Scene Exploration. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, pages 2384–2391, 2013.

[14] S. Kriegel, C. Rink, T. Bodenmuller, A. Narr, M. Suppa, and G. Hirzinger. Next-Best-Scan Planning for Autonomous 3D Modeling. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, pages 2850–2856, 2012.

[15] S. Krishnan, P.Y. Lee, J.B. Moore, and S. Venkatasubramanian. Global Registration of Multiple 3D Point Sets via Optimization-on-a-Manifold. In *Proceedings of the third Eurographics symposium on Geometry processing*, SGP '05, 2005.

[16] Y.F. Li and Z.G. Liu. Information Entropy-Based Viewpoint Planning for 3-D Object Reconstruction. *IEEE Transactions on Robotics*, 21(3):324–337, 2005.

[17] A. Makadia, A. Patterson, and K. Daniilidis. Fully Automatic Registration of 3D Point Clouds. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 1297–1304, 2006.

[18] K. Morooka, Hongbin Zha, and T. Hasegawa. Next Best Viewpoint (NBV) Planning for Active Object Modeling Based on a Learning-by-Showing Approach. In *Fourteenth Intl Conference on Pattern Recognition*, volume 1, pages 677–681 vol.1, 1998.

[19] R. Pito. A Solution to the Next Best View Problem for Automated Surface Acquisition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(10):1016–1030, 1999.

[20] K. Pulli. Multiview registration for large data sets. In *Second Intl Conference on 3-D Digital Imaging and Modeling*, pages 160–168, 1999.

[21] M.K. Reed and P.K. Allen. Constraint-Based Sensor Planning for Scene Modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1460–1467, 2000.

[22] R.B. Rusu and S. Cousins. 3D is Here: Point Cloud Library (PCL). In *IEEE Intl Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 9-13 2011.

[23] R. Schmedding, B. Frank, W. Burgard, and M. Teschner. Transformed Polynomials for Global Registration of Point Clouds. In *Proceedings of the 27th Spring Conference on Computer Graphics*, SCCG, pages 129–136, New York, NY, USA, 2011. ACM.

[24] R. Toldo, A. Beinat, and F. Crosilla. Global registration of multiple point clouds embedding the Generalized Procrustes Analysis into an ICP framework. In *5th Intl Symposium 3D Data Processing, Visualization and Transmission (3DPVT2010)*, 2010.

[25] L. Torabi and K. Gupta. Integrated View and Path Planning for an Autonomous six-DOF Eye-in-hand Object Modeling System. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, pages 4516–4521, 2010.

[26] A. Tsuda, Y. Kakiuchi, S. Nozawa, R. Ueda, K. Okada, and M. Inaba. On-line Next Best Grasp Selection for In-Hand Object 3D Modeling with Dual-Arm Coordination. In *IEEE Intl Conference on Robotics and Automation (ICRA)*, pages 1799–1804, 2012.

[27] N. Van-Tung and D. Laurendeau. A Global Registration Method Based on the Vector Field Representation. In *Canadian Conference on Computer and Robot Vision (CRV)*, pages 132–139, 2011.

[28] J.I. Vasquez-Gomez, E. Lopez-Damian, and L.E. Sucar. View Planning for 3D Object Reconstruction. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, pages 4015–4020, 2009.

[29] G. Walck and M. Drouin. Automatic Observation for 3D Reconstruction of Unknown Objects Using Visual Servoing. In *IEEE/RSJ Intl Conference on Intelligent Robots and Systems (IROS)*, pages 2727–2732, 2010.

[30] K. Welke, J. Issac, D. Schiebener, T. Asfour, and R. Dillmann. Autonomous Acquisition of Visual Multi-View Object Representations for Object Recognition on a Humanoid Robot. In *IEEE Intl Conference on Robotics and Automation (ICRA)*, pages 2012–2019, 2010.

[31] P. Whaite and F.P. Ferrie. Autonomous Exploration: Driven by Uncertainty. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(3):193–205, 1997.

[32] Yong Yu and Kamal K. Gupta. C-space Entropy: A Measure for View Planning and Exploration for General Robot-Sensor Systems in Unknown Environments. *The Intl Journal of Robotics Research*, 23(12):1197–1223, 2004.