

# Low Resolution Next Best View for Robot Packing

Giuseppe Fabio Preziosa\* Chiara Castellano\*  
Andrea Maria Zanchettin\* Marco Faroni\* Paolo Rocco\*

\* The Authors are with Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Piazza Leonardo da Vinci 32, 20133, Milano (Italy). (e-mail: giuseppefabio.preziosa, chiara2.castellano, andreamaria.zanchettin, marco.faroni, paolo.rocco)@polimi.it

**Abstract:** Automating the packing of objects with robots is a key challenge in industrial automation, where efficient object perception plays a fundamental role. This paper focuses on scenarios where precise 3D reconstruction is not required, prioritizing cost-effective and scalable solutions. The proposed Low-Resolution Next Best View (LR-NBV) algorithm leverages a utility function that balances pose redundancy and acquisition density, ensuring efficient object reconstruction. Experimental validation demonstrates that LR-NBV consistently outperforms standard NBV approaches, achieving comparable accuracy with significantly fewer poses. This method proves highly suitable for applications requiring efficiency, scalability, and adaptability without relying on high-precision sensing.

Copyright © 2025 The Authors. This is an open access article under the CC BY-NC-ND license (<https://creativecommons.org/licenses/by-nc-nd/4.0/>)

**Keywords:** Low-Resolution Perception, 3D Object Reconstruction, Sensory-Based Robot Control, Collaborative Robots, Computer Vision for Robotics.

## 1. INTRODUCTION

Automation in packaging and warehouse logistics has emerged as a pivotal area of industrial research, driven by the increasing demand for flexible and customizable production. Among the many processes impacted by this shift, the final packaging phase—where objects are identified, their dimensions determined, and placed in appropriately sized containers—represents a shared challenge that spans industries and product types. Currently, these tasks are predominantly performed by human workers, with container selection and object arrangement often relying on intuition and manual effort. This dependency introduces inconsistencies and limits scalability, particularly in dynamic, high-volume environments.

Research in robotic packing has traditionally focused on optimizing spatial efficiency within packages. Wang and Hauser (2019) tackled the challenge of optimizing space while considering package stability, whereas more recently, Wang and Hauser (2021) focused on exploring optimal spatial arrangements to maximize efficiency with both known and unknown objects. These methods often rely on high-resolution sensors and precise planning designed to be implemented offline before the arrival of objects. However, a shift has emerged towards solutions based on low-cost and easily scalable hardware. These approaches explore simpler sensing systems, such as RGB cameras in Hong et al. (2020), and minimalistic grippers with Shome et al. (2019), aiming to provide practical and adaptable solutions for real-world applications. Along this line, the present work addresses the perception challenges associated with the use of *low-resolution sensing* systems in robot packing processes.

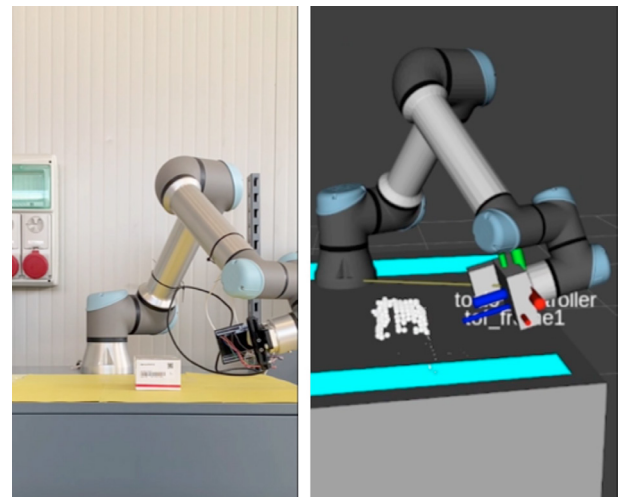


Fig. 1. Last acquisition of the LR-NBV algorithm showing on the right the final result of the reconstruction.

Despite the proven effectiveness of low-resolution sensors in other contexts, such as human-robot safety, as demonstrated by Ceriani et al. (2013), their use for object reconstruction presents significant challenges. Their reduced fidelity can make it difficult to capture sufficient detail with a single acquisition, often requiring additional view-points to build an adequate representation of the object. In packaging tasks, however, detailed 3D reconstructions are not strictly necessary. Instead, efficiently estimating object dimensions and bounding volumes is sufficient for subsequent steps, such as grasp planning, container selection and arrangement of the objects. Studies like Kopicki et al. (2016); Alliegro et al. (2022); Ten Pas et al. (2017)

demonstrate that even partial reconstructions, covering only 25–30% of an object surface, can enable successful robotic manipulation. This highlights how low-resolution sensors can effectively balance efficiency and precision, offering scalability and cost advantages.

However, low-resolution sensing introduces further challenges, such as reducing the number of acquisitions while managing viewpoint diversity evaluation. Traditional *Next Best View (NBV)* algorithms, designed for high-resolution sensors, often fail to adapt to these constraints, necessitating the development of tailored solutions. To address these limitations, this paper proposes a novel Low-Resolution Next Best View (LR-NBV) algorithm, which, to the best of our knowledge, is one of the first works to introduce and explore the use of low-cost and easily integrable sensors for perception tasks like this.

## 2. RELATED WORKS

The problem of observing and evaluating the best next viewpoint to gain information about the environment is a well-known challenge, referred to as the *Next Best View* problem. Initially introduced in the early 1990s by Maver and Bajcsy (1993); Pito (1999); Scott et al. (2003), this concept has been extensively studied and applied to tasks such as exploration, object reconstruction, and inspection, across a wide range of robotic applications, including both mobile robots and fixed manipulators. The characterization of an NBV algorithm involves addressing several core aspects, which provide a clear framework for classifying our proposed approach. These aspects include task definition, spatial representation, observation evaluation, utility functions, constraints, and planning strategies.

NBV algorithms can be broadly categorized into *model-based view planning* and *non-model-based view planning*. As outlined in Scott et al. (2003), model-based approaches aim to verify an existing model by comparing it to a physical object, often for defect detection or conformity checks. In contrast, non-model-based approaches, which are the focus of this work, assume no prior knowledge of the object and aim to incrementally build its representation from scratch. This distinction is particularly relevant for 3D object reconstruction in unstructured environments, where the lack of prior information necessitates flexible and efficient strategies. Non-model-based methods often rely on volumetric representations to discretize the environment into manageable units. These methods segment the space into voxel grids, typically combined with an associated probability and hierarchical structure, as represented by the octomap framework in Hornung et al. (2013). Each voxel is classified as free, occupied, or uncertain, simplifying visibility operations and enabling probabilistic updates of voxel states based on sensor data. Viewpoint evaluation is typically conducted using raycasting, where rays are cast from the sensor into the voxelized space, simulating the sampling process of a camera. As the rays traverse the voxel grid, each voxel state is updated probabilistically, enabling the estimation of information gain.

Schmid et al. (2020) highlighted how the combination of volumetric representations and sampling-based algorithms provides a flexible framework for defining utility functions. These utility functions, which are at the core of NBV al-

gorithms, guide the selection of viewpoints by quantifying their informativeness and can be tailored to specific tasks and hardware constraints, providing the foundation for effective exploration strategies. Classic NBV algorithms typically define utility functions based entirely on exploration gain, which focuses on maximizing the amount of new information acquired. For instance, Vasquez-Gomez et al. (2014) introduce an AreaFactor that balances the observation of different voxel types, while Selin et al. (2019) propose a volumetric gain that is computationally efficient, making it both fast and suitable for a packing task. Additionally, Banta et al. (2000) directly counts the number of unknown voxels along the raycast path to quantify exploration. Finally, Isler et al. (2016) presents multiple formulations for exploration gain, including metrics such as occlusion-aware and rear-side voxel information, optimized for volumetric 3D reconstruction. On the other hand, fewer studies incorporate hardware-specific considerations into utility functions, often referred to as *quality gains*. For example, Vasquez-Gomez et al. (2014) prioritize viewpoints that minimize robot motion by favoring poses requiring less movement. Similarly, in sensor-specific scenarios, Massios et al. (1998) optimize the sensor acquisition angle, with Massios et al. (1998) positively weighting viewpoints that are perpendicular to the surface.

Although effective, the quality gains discussed earlier, when combined with classical exploration gains, face limitations in addressing the challenges posed by low-resolution sensors. A key limitation of these sensors is their inability to differentiate effectively between acquisitions made from slightly different positions. Due to the sparsity of the data describing the object, traditional utility functions assign similar exploration weights to these redundant viewpoints, leading to unnecessary acquisitions. In robotic packing, where speed and efficiency are paramount, this redundancy significantly slows down the process. To address redundancy, Naazare et al. (2022) introduces a *Visited Gain* that penalizes redundant acquisitions by negatively weighting revisited areas. While this gain was designed for mobile manipulators, our work extends its application to fixed manipulators. Furthermore, we introduce additional gains tailored to the unique constraints of low-resolution sensors, aiming to optimize the acquisition process and enhance efficiency in robotic packing tasks.

Additionally, this work differs from many approaches requiring prior knowledge of the object position and size. For instance, Krainin et al. (2011), and Wenhardt et al. (2007) sample viewpoints on a sphere centered around the object, which implicitly requires the knowledge of the object pose and dimension. In contrast, this work structures sampling directly within the 3D space of the sensor pose, using only the robot workspace boundaries as constraints. Similar to Daudelin and Campbell (2017), the algorithm requires only an initial acquisition of the object to start the process, allowing it to adapt seamlessly to objects of varying sizes and positions.

This work also belongs to the family of approaches that expand a tree of viewpoints in the sampling space, drawing inspiration from RRT-based methods. At each iteration, the tree is incrementally built starting from the end effector pose and integrates updated information from the environment. This ensures that the planning process adapts

dynamically as new data are acquired. Unlike classical RRT, where the entire path is planned upfront, this algorithm follows the receding horizon paradigm introduced in Bircher et al. (2016). In this approach, the tree is constructed iteratively, and only the first segment of the most promising branch—evaluated based on utility gain—is executed. The process is then repeated, progressively refining the tree and adapting to the newly acquired map of the environment. To address the risk of getting trapped in local minima, as highlighted by Selin et al. (2019) and Naazare et al. (2022), the algorithm retains the most promising nodes from previous iterations, reevaluating them in subsequent cycles. This strategy reduces computational costs while providing useful indices for monitoring reconstruction progress and defining termination criteria, enhancing the robustness of the exploration process.

### 3. PROPOSED APPROACH

---

#### Algorithm 1 Low Resolution Next Best View Algorithm

---

```

1: Inputs:  $A^0, C^0, n$  given.
2: completed  $\leftarrow$  False
3:  $i \leftarrow 0$ 
4:  $B^{*i} \leftarrow \emptyset$ 
5:  $O \leftarrow \emptyset$ 
6: while  $\neg$ completed do
7:    $(B^{*i}, C_{\mathcal{T}^i}) \leftarrow$  ExpandRRT( $n, A^i$ )
8:    $C^i \leftarrow$  AppendNodes( $C^{i-1}, C_{\mathcal{T}^i}$ )
9:    $x_T^* \leftarrow$  arg max $_{x_T \in B^{*i}} U(x_T)$ 
10:  if  $U(x_T^*) < U(x_T^{\text{bestC}})$  then
11:     $(B^{*i}, C_{\hat{\mathcal{T}}^i}) \leftarrow$  ExpandRRTCached( $C^i, A^i$ )
12:     $C^i \leftarrow$  AppendNodes( $C^i, C_{\hat{\mathcal{T}}^i}$ )
13:     $x_T^* \leftarrow$  arg max $_{x_T \in B^{*i}} U(x_T)$ 
14:  end if
15:  if  $U(x_T^*) > U(x_T^{\text{worstC}})$  then
16:     $D^i \leftarrow$  captureData( $B^{*i}$ )
17:     $A^i \leftarrow$  UpdateOctomap( $D^i, A^{i-1}$ )
18:     $O^i \leftarrow$  ExtractObjectPoints( $A^i$ )
19:     $C^i \leftarrow$  UpdateCachedNodes( $C^{i-1}, A^i$ )
20:     $x_T^{\text{bestC}} \leftarrow$  arg max $_{x_T \in C^i} U(x_T)$ 
21:     $x_T^{\text{worstC}} \leftarrow$  arg min $_{x_T \in C^i} U(x_T)$ 
22:     $i \leftarrow i + 1$ 
23:  else
24:    completed  $\leftarrow$  True
25:  end if
26: end while
27: return  $O^i$ 

```

---

The approach presented in this paper aims to identify an efficient sequence of sensor poses, reducing the makespan of the acquisition while ensuring a sufficient reconstruction of the object surface to enable the robot to correctly grasp the object. Each sensor pose  $x_T \in SE(3)$  generates an observation  $D(x_T)$ , defined as a set of  $M$  points  $\{p_1, p_2, \dots, p_M\} \subseteq \mathbb{R}^3$ , where  $M$  represents the resolution of the sensor and determines the number of points captured during each acquisition. The object  $O$  reconstructed is a function of the sequence of observations collected from the  $k$  poses selected by the algorithm, formally expressed as  $O = f(D(x_{T_1}), D(x_{T_2}), \dots, D(x_{T_k}))$ . To guide the selection of these poses, the algorithm employs a utility function  $U(x_{T_j})$ , which evaluates the expected contribution

of each pose to the reconstruction process. This function balances the need to explore new areas of the object and avoid redundant measurements caused by the sensor limited resolution.

The three main components of this algorithm are presented in the following sections. In particular, Section 3.1 describes the pose-searching algorithm, which serves as the backbone of the entire approach. Section 3.2 explains how the object is identified and characterized based on data extracted from the environment after observations. Finally, Section 3.3 provides a detailed explanation of the utility function used during the evaluation process.

#### 3.1 Sampling-Based RRT

The proposed algorithm follows a cyclic structure inspired by state-of-the-art approaches described by Bircher et al. (2016); Naazare et al. (2022). At each iteration  $i$ , starting from line 7 in Algorithm 1, a tree  $\mathcal{T}^i$  is constructed through the expansion step. Specifically, the tree  $\mathcal{T}^i$  is initialized from the current sensor pose, which is determined by the robot end-effector pose, and its branches are expanded by sampling  $n$  points in the 3D space constrained by the robot workspace. For each sampled point, a corresponding pose  $x'_T \in SE(3)$  is generated, where the orientation is determined to maximize the utility function  $U(x'_T)$ . From this tree, the best branch  $B^{*i}$  is selected by maximizing the cumulative utility of its nodes:

$$B^{*i} = \operatorname{argmax}_{B_k^i \in \mathcal{T}^i} \sum_{x_T \in B_k^i} U(x_T) \quad (1)$$

From the best branch, the best node is then extracted (line 9). Finally, the best branch is used to suggest the next poses for data acquisition (line 16). The algorithm also evaluates the remaining branches of  $\mathcal{T}^i$ , identifying nodes not included in  $B^{*i}$  but with utility  $U(x_T)$  exceeding a threshold  $\tau$ . These nodes, referred to as *cached nodes*, are stored in  $C_{\mathcal{T}^i}$  and added to the global structure  $C^i$  (lines 8 and 12). Since a new tree is created at every iteration, cached nodes play a crucial role in mitigating issues related to local minima. Specifically, the cached nodes from the previous iteration  $i-1$  are analyzed to extract the best and worst nodes according to their utility. The *best cached node* is used to decide whether it is worth expanding the tree toward the cached nodes (lines 10 and 11), where the tree is expanded by sampling from the most promising cached nodes. On the other hand, the *worst cached node* provides a utility gain threshold to determine whether exploration should continue. The exploration is considered complete if the utility of the best node falls below the utility of the worst cached node (line 24). It is also important to note that at the end of each iteration, the cached nodes are updated using new data acquired during the process. This update reorders the cached nodes, keeping only a predefined number of nodes  $m$ , which decreases linearly as iterations progress. This mechanism ensures that only the most relevant nodes are retained, improving the efficiency of subsequent iterations.

#### 3.2 Object Clustering and Dynamic ROI

One of the key strengths of this algorithm is that no prior knowledge of the object pose and dimensions are required.

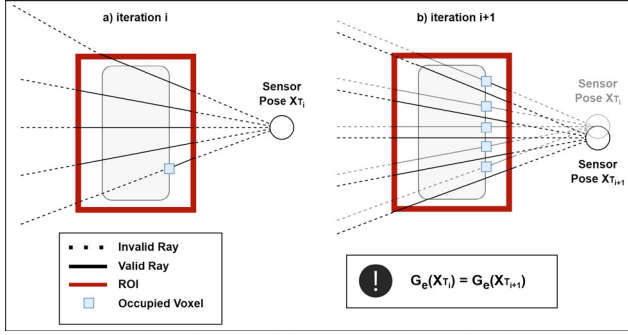


Fig. 2. Validity of rays and exploration gains are shown in two subsequent iterations of the algorithm. Relying solely on exploration gain makes pose selection challenging, as poses slightly different from the previous ones yield the exact same gain.

By integrating sensor measurements into the volumetric octomap  $A$ , the algorithm dynamically updates voxel occupancy probabilities, categorizing them as occupied, uncertain, or empty (line 17 in Algorithm 1). This process enables efficient segmentation of the environment, laying the groundwork for object identification. To achieve this, the algorithm applies the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) from Deng (2020), which identifies clusters based solely on point density. Its adaptability to irregular shapes and independence from prior object information make it computationally efficient for this task (line 18 in Algorithm 1). Depending on the type of sensor used and insights gained through adequate sensor characterization, additional filters may need to be applied to DBSCAN to refine the clustering process and handle sensor-specific noise effectively. Once the object  $O^i$  is identified, it is enclosed within a bounding box that serves two key roles. First, it defines the *Region of Interest (ROI)*, guiding the utility function to prioritize reconstruction efforts in relevant regions while avoiding unnecessary exploration. Second, it acts as a *Collision Checking* element, facilitating safe tree node generation and robot motion planning.

### 3.3 Utility Function

The proposed utility function evaluates the potential of a candidate pose  $x_T$  by combining four main components: Exploration Gain ( $G_e$ ), Density Gain ( $G_d$ ), Quality Gain ( $G_q$ ), and Visited Gain ( $G_v$ ). The utility function, therefore, takes the form of:

$$U(x_n) = w_e G_e(x_T) + w_d G_d(x_T) + w_q G_q(x_T) + w_v G_v(x_T)$$

where  $w_e, w_d, w_q, w_v > 0$  are the weights associated with each gain and can be adjusted to balance the relative importance of each component based on the specific task.

**Exploration Gain ( $G_e$ ) and Density Gain ( $G_d$ )** The first two gains are based on a raycasting process that simulates the sensor behavior at a given pose. This process is defined by the sensor intrinsic properties, such as field of view (FoV) and resolution (res), which determine the spatial distribution of the emitted rays. Each ray  $\mathbf{r} \in \mathcal{R}(x_n, \text{FoV}, \text{res})$  originates from the sensor position  $x_T$  and can be defined as:  $\mathbf{r}(t) = \mathbf{o} + t\mathbf{d}$ , where  $\mathbf{o} \in \mathbb{R}^3$  is the ray origin,  $\mathbf{d} \in \mathbb{R}^3$  is its direction, and  $t$  represents the distance

along the ray. We define  $t^*$  as the first point along the ray  $\mathbf{r}(t)$  that intersects the object  $O^i$ , represented as a set of points grouped as described in Section 3.2. Based on this, we introduce the concept of ray validity as:

$$V(\mathbf{r}, t) = \begin{cases} 1 & \text{if } \mathbf{r}(t) \cap \text{ROI} \neq \emptyset \text{ and } \exists t^* \leq t : \mathbf{r}(t^*) \in O^i, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

where, as visually explained in Fig. 2a, a portion of ray is considered valid only if it lies within the ROI and respects the occlusion constraint. This approach simulates the impossibility of perceiving regions beyond an occlusion represented by the object, ensuring that only the visible portions of the workspace contribute to the reconstruction process. Based on that, the *Exploration Gain*  $G_e(x_T)$  is defined as:

$$G_e(x_T) = \sum_{\mathbf{r} \in \mathcal{R}(x_T, \text{FoV}, \text{res})} \sum_{t_k=0}^1 V(\mathbf{r}, t_k) W(\mathbf{r}, t_k) \quad (3)$$

where  $t_k$  represents discrete samples along the ray, uniformly spaced between  $t = 0$  and  $t = 1$ . In this formulation,  $W(\mathbf{r}, t_k)$  quantifies the contribution of each ray segment to volumetric exploration, accounting for the volume traversed in voxels as described by Selin et al. (2019). The combination of ray validity and volumetric exploration focuses the process on areas of high uncertainty, maximizing the acquisition of new information.

However, the use of a purely exploratory gain does not adequately weigh or prioritize one pose over another, especially in the context of low-resolution sensors. As illustrated in Fig.2b, the same  $G_e$  value can be associated with two slightly different poses, as the rays would be entirely unobstructed and thus considered fully explorable. This limitation arises because  $G_e$  does not account for the density of object points in the explored regions, potentially leading to redundant acquisitions. To address this issue, we introduce the *Density Gain*  $G_d$ , designed to complement  $G_e$  by favoring regions with sparse object points, as:

$$G_d(x_T) = \sum_{\mathbf{r} \in \mathcal{R}(x_T, \text{FoV}, \text{res})} \sum_{t_k=0}^1 V(\mathbf{r}, t_k) \cdot \frac{1}{1 + \rho(\mathbf{r}, t_k)} \quad (4)$$

Here,  $\rho(\mathbf{r}, t_k)$  represents the number of object points within a spherical neighborhood of radius  $r_d$  centered on  $\mathbf{r}(t_k)$ . This approach prioritizes regions with lower point density by penalizing rays in densely populated areas reducing redundancy.

**Quality Gain ( $G_q$ ) and Visited Gain ( $G_v$ )** The *Quality Gain*  $G_q$  is particularly important when working with low-resolution and low-cost sensors, as it prioritizes acquisitions closer to the object to reduce noise. It is defined as:

$$G_q(x_T) = \frac{1}{1 + \|x_T - x_{\text{object}}\|} \quad (5)$$

where  $x_{\text{object}}$  represents the estimated position of the object center. This formulation ensures that closer poses receive higher weights, promoting lower-noise acquisitions.

The *Visited Gain*  $G_v$  prevents the algorithm from re-visiting already explored regions. Following the approach presented by Naazare et al. (2022), a coarser discretization of the workspace is applied compared to the Octomap. Each discrete cell is assigned a value of 1 if already visited or 0 otherwise. The gain is then defined as:

$$G_v(x_T) = -\text{Visited}(x_T) \quad (6)$$

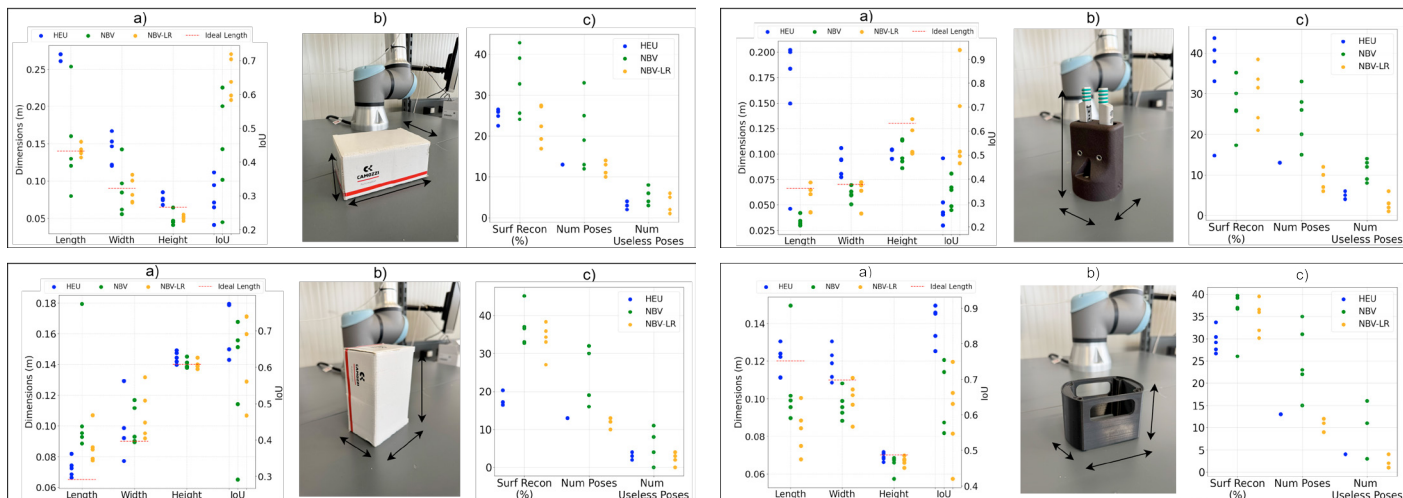


Fig. 3. Acquisition results for four objects: the *horizontal box* (top left), the *vertical box* (bottom left), the *gripper* (top right), and the *flange* (bottom right). For each object: (a) shows the comparison of estimated dimensions (length, width, height) and IoU, (b) displays the object with its canonical dimensions, and (c) presents the surface reconstruction percentage, number of poses, and number of useless poses.

The minus sign is used to reduce the overall gain when a cell has been already visited.

#### 4. EXPERIMENTAL VALIDATION

The validation of this work compares the proposed *LR-NBV* algorithm with two alternatives. The first alternative is a standard *NBV* approach that follows the algorithm described in Algorithm 1, characterized by the same *Exploration Gain* and *Quality Gain* introduced in Section 3.3. Notably, this standard approach does not utilize the *Density Gain* and *Visited Gain*, also described in Section 3.3, which are key contributions of this work. This comparison aims to evaluate the effect of a utility function primarily based on exploration, as is common in most works in the literature. The second alternative is a heuristic method (*HEU*) that requires the object position and generates 13 evenly spaced poses around an ellipse centered on the workspace. The experimental validation was conducted by performing 5 reconstructions for 4 different objects, shown in Fig. 3, for the three proposed algorithms.

##### 4.1 Experimental Setup

The algorithms were tested in a real-world setup using a UR5e robotic arm equipped with a VL53L8CX time-of-flight sensor mounted on the end effector. The sensor features a  $45^\circ \times 45^\circ$  field of view (FoV) and an 8x8 resolution, making it a very low-resolution device. The tests were conducted on a Lenovo Yoga Pro 7i Gen 9 laptop with an Intel<sup>®</sup> Core<sup>™</sup> i7-13700H processor and 32GB of RAM. The implementation was based on Robot Operating System (ROS), utilizing the Octomap Library for dynamic measurement management and spatial representation, and MoveIt for motion planning. The parameters used during the experimental phase are summarized in Table 1.

The evaluation of the algorithms was based on four metrics. 1) *Bounding box dimensions*: These represent the estimated dimensions of the object and are complemented by the intersection over union (IoU) index, which quantifies

the precision of the reconstruction, which is critical for selecting appropriate-sized containers in packaging lines. 2) *Number of poses*: this serves as a proxy for time efficiency in robotic packing applications, as the execution time between iterations for NBV-based algorithms averages under 2 seconds on the experimental setup (excluding robot movement time). 3) *Surface coverage*: represents the percentage of the surface of the object reconstructed during the process. 4) *Number of useless poses*: refers to poses that do not increase the reconstructed surface and instead cause a decrease in the overall reconstruction quality. These metrics were extracted and are represented across five measurements for each object in Fig. 3.

##### 4.2 Results

We conducted the Whitney U-test to compare the methods LR-NBV, NBV, and HEU across all metrics and experiments. The results indicate that LR-NBV achieves statistically significant improvements ( $p < 0.05$ ) in terms of IoU, particularly for the objects *horizontal box* and *gripper*. For the other two objects (*vertical box* and *flange*), differences ( $p < 0.05$ ) are observed for at least one dimension (length, width, or height), highlighting the ability of LR-NBV to provide more accurate bounding box reconstructions in several cases compared to the baselines. In terms of efficiency, LR-NBV consistently requires a significantly lower number of poses ( $p < 0.05$ ) to complete the reconstruction, often less than half the poses required by

Table 1. Parameters used for the experiments.

Parameter	Value
num.iterations	300
sample.per.iteration	50
num.cached.nodes	50
visited.cells.grid.size [m]	0.18
exploration.weight ( $w_e$ )	500.0
density.weight ( $w_d$ )	0.05
quality.weight ( $w_q$ )	0.08
visited.weight ( $w_v$ )	500.0
Octomap resolution [m]	0.01

NBV, while maintaining a reconstruction percentage that is at least comparable. This improved efficiency stems from the ability of LR-NBV to better prioritize informative poses, reducing redundant acquisitions. Conversely, NBV exhibits a higher number of *useless poses*, which temporarily decrease the reconstruction percentage. These fluctuations extend the process as additional poses are required to recover the reconstruction quality lost due to these redundant acquisitions. However, even though we obtained satisfactory results for two of the four objects, the results for the remaining two objects show that there is still room for improvement. Therefore, more extensive evaluations will be conducted.

## 5. CONCLUSIONS

This work has demonstrated the potential of low-resolution sensors, characterized by low cost and ease of integration, for object reconstruction in robotic perception. The proposed algorithm has shown that sufficiently accurate reconstructions can be achieved to estimate object dimensions while significantly reducing the number of acquisitions compared to traditional methods. Future work will focus on validating the adequacy of the reconstructed surfaces for grasp planning and evaluating the algorithm performance on objects with more complex geometries using different sensors.

## ACKNOWLEDGEMENTS

The authors acknowledge *Camozzi Research Center* for providing sensor and robotic equipment, as well as for their support in this work.

## REFERENCES

- Alliegro, A., Rudorfer, M., Frattin, F., Leonardis, A., and Tommasi, T. (2022). End-to-end learning to grasp via sampling from object point clouds. *IEEE Robotics and Automation Letters*, 7(4), 9865–9872.
- Banta, J.E., Wong, L., Dumont, C., and Abidi, M.A. (2000). A next-best-view system for autonomous 3-d object reconstruction. *IEEE Trans. Syst. ManCyb.-Part A: Systems and Humans*, 30, 589–598.
- Bircher, A., Kamel, M., Alexis, K., Oleynikova, H., and Siegwart, R. (2016). Receding horizon "next-best-view" planner for 3d exploration. In *IEEE Int. Conf. on Robotics and Automation*, 1462–1468.
- Ceriani, N.M., Avanzini, G.B., Zanchettin, A.M., Bascetta, L., and Rocco, P. (2013). Optimal placement of spots in distributed proximity sensors for safe human-robot interaction. In *IEEE Int. Conf. on Robotics and Automation*, 5858–5863.
- Daudelin, J. and Campbell, M. (2017). An adaptable, probabilistic, next-best view algorithm for reconstruction of unknown 3-d objects. *IEEE Robotics and Automation Letters*, 2(3), 1540–1547.
- Deng, D. (2020). Dbscan clustering algorithm based on density. In *Int. Forum on Electrical Engineering and Automation*, 949–953.
- Hong, Y.D., Kim, Y.J., and Lee, K.B. (2020). Smart pack: online autonomous object-packing system using rgb-d sensor data. *Sensors*, 20(16), 4448.
- Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., and Burgard, W. (2013). Octomap: an efficient probabilistic 3d mapping framework based on octrees. *Auton. Robots*, 34(3), 189–206.
- Isler, S., Sabzevari, R., Delmerico, J., and Scaramuzza, D. (2016). An information gain formulation for active volumetric 3d reconstruction. In *IEEE Int. Conf. on Robotics and Automation*, 3477–3484.
- Kopicki, M., Detry, R., Adjigble, M., Stolkin, R., Leonardis, A., and Wyatt, J.L. (2016). One-shot learning and generation of dexterous grasps for novel objects. *The Int. Journal of Robotics Research*, 35(8), 959–976.
- Krainin, M., Curless, B., and Fox, D. (2011). Autonomous generation of complete 3d object models using next best view manipulation planning. In *IEEE Int. Conf. on Robotics and Automation*, 5031–5037.
- Massios, N.A., Fisher, R.B., et al. (1998). A best next view selection algorithm incorporating a quality criterion. In *Proc. British Machine Vision Conference*, volume 2.
- Maver, J. and Bajcsy, R. (1993). Occlusions as a guide for planning the next view. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(5), 417–433.
- Naazare, M., Rosas, F.G., and Schulz, D. (2022). Online next-best-view planner for 3d-exploration and inspection with a mobile manipulator robot. *IEEE Robotics and Automation Letters*, 7(2), 3779–3786.
- Pito, R. (1999). A solution to the next best view problem for automated surface acquisition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 21(10), 1016–1030.
- Schmid, L., Pantic, M., Khanna, R., Ott, L., Siegwart, R., and Nieto, J. (2020). An efficient sampling-based method for online informative path planning in unknown environments. *IEEE Robotics and Automation Letters*, 5(2), 1500–1507.
- Scott, W.R., Roth, G., and Rivest, J.F. (2003). View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.*, 35(1), 64–96.
- Selin, M., Tiger, M., Duberg, D., Heintz, F., and Jensfelt, P. (2019). Efficient autonomous exploration planning of large-scale 3-d environments. *IEEE Robotics and Automation Letters*, 4(2), 1699–1706.
- Shome, R., Tang, W.N., Song, C., Mitash, C., Kourtev, H., Yu, J., Boularias, A., and Bekris, K.E. (2019). Towards robust product packing with a minimalistic end-effector. In *Int. Conf. on Robotics and Automation*, 9007–9013.
- Ten Pas, A., Gualtieri, M., Saenko, K., and Platt, R. (2017). Grasp pose detection in point clouds. *The Int. Journal of Robotics Research*, 36(13-14), 1455–1473.
- Vasquez-Gomez, J.I., Sucar, L.E., Murrieta-Cid, R., and Lopez-Damian, E. (2014). Volumetric next-best-view planning for 3d object reconstruction with positioning error. *Int. J. of Advanced Robotic Systems*, 11(10), 159.
- Wang, F. and Hauser, K. (2019). Stable bin packing of non-convex 3d objects with a robot manipulator. In *Int. Conf. on Robotics and Automation*, 8698–8704. IEEE.
- Wang, F. and Hauser, K. (2021). Dense robotic packing of irregular and novel 3d objects. *IEEE Trans. on Robotics*, 38(2), 1160–1173.
- Wenhardt, S., Deutsch, B., Angelopoulou, E., and Niemann, H. (2007). Active visual object reconstruction using d-, e-, and t-optimal next best views. In *IEEE Conf. Computer Vision and Pattern Recognition*, 1–7.