Multi-camera architecture for perception strategies

Enrique Hernández-Murillo Rosario Aragüés Gonzalo López-Nicolás

***Abstract*—Building the 3D model of an object is a complex problem that involves aspects such as modeling, control, perception or planning. Performing this task requires a set of different views to cover the entire surface of the object. Since a single camera takes too long to travel through all these positions, we consider a multi-camera scenario. Due to the camera constraints such as the limited field of view or self occlusions, it is essential to use an effective configuration strategy to select the appropriate views that provide more information of the model. In this paper, we develop a multi-camera architecture built on the Robot Operating System. The advantages of the proposed architecture are illustrated with a formation-based algorithm to compute the view that satisfies these constraints for each robot of the formation to obtain the volumetric reconstruction of the target object.**

***Index Terms*—Multi-Robot, 3D Object Reconstruction, Robot Operating System (ROS), Gazebo, Visual Perception.**

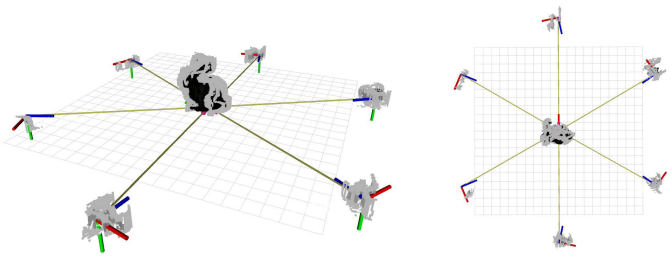
I. Introduction

Nowadays in manufacturing industry, perception is one of the most basic tasks related to manipulation of objects. Here, the main goal is the reconstruction and perception along time of an, a priori, unknown object. The process of building a 3D model of a real object, which is known as volumetric object reconstruction is essential in robotic manipulation.

On the one hand, active vision could be taken as a viable alternative [3]. This technique may require using a vision-based sensor, mounted on a mobile robot, providing dense 3D input data. Active vision methods [1], [2], address the problem of generating a complete volumetric model of the object, as fast as possible, by solving the next best view problem. Thus, they compute a sequence of camera locations (position and orientation) around the object. In order to reduce the number of candidate views, they may be restricted to a sphere or a cylinder around the object. However, one of the main challenges involved in active vision are deformable or mobile objects, which requires instantaneous and full perception of the object. The main drawback of active vision in this framework is the time required to follow the sequence of camera locations. So, how can be something perceived from multiple views at once? The natural answer could be to fuse the information from several cameras in a volumetric map of the entire object at each moment (Fig. 1). Besides, the multi-camera system could be a good choice if the model is deformable or the object moves. We propose to place several cameras in a geometric formation around the moving object to enclose it and track it. This strategy [6] ensures that the multi-robot system can perform full perception of the target along its motion.

The authors are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. ehernandez@unizar.es, gonlopez@unizar.es, raragues@unizar.es

This work was supported by projects COMMANDIA SOE2/P1/F0638 (Interreg Sudoe Programme, ERDF) and PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE).



(a) Perspective View (b) Top View

Fig. 1. *Illustrations from two diferent views of the reconstructed Bunny object and 6 cameras. Final representations (point cloud) of the scene and the multi-camera formation enclosing the object are visualized. The figure depicts the output of the system as the data structure visualized by Rviz.*

In this work, we consider the problem of the reconstruction of an a priori unknown object that deforms and moves (its boundaries are not limited). We approach the problem of the instantaneous 3D reconstruction using a probabilistic volumetric map built in real time. Besides, we extend the problem to a multi-robot team scenario. We propose and evaluate a multi-camera generic system architecture for the camera positioning mechanism to model 3D arbitrary objects. The multi-robot scenario allows us to elaborate a strategy which improves the process of tracking a moving and deforming object. The proposed approach presented in this paper is a modular tool for the specific task of multi-sensor based reconstruction of an object of interest. Nevertheless, numerous problems of a technological nature can benefit of our proposed architecture.

The main goal of this work is to design a modular software framework to perform the instantaneous reconstruction of an object, with a multi-robot strategy. Versatility and modularity are the main advantages of the proposed technological tool. Within this framework, the architecture of our approach uses both a *ROS-based* generic system and a *Matlab-based* front-end strategy interface and builds on [5] and [6]. We evaluate the complete volumetric map of the object in simulations performed in *Gazebo*.

The paper is organized as follows: we introduce our background in Sect. II. In Sect. III we address the problem, then give an overview of our software framework in Sect. IV. Simulations testing the performance of the multi-robot strategy and the camera positioning mechanism in a *Gazebo-based* enviroment are shown in Sect. V.

978-1-7281-0303-7/19/$31.00 ©2019 IEEE

II. Background

There are several robotic proccesses, such as tracking, robust perception and manipulation tasks where three-dimensional reconstruction of real objects is an important step. The model of the object could be obtained using an a priori and accurate representation of the object, for instance, a CAD model; this way is the least efficient due to the necessity of a professional human modeller. Moreover, in case the object deforms, the selection and reconstruction of an appropiate model for the object is an open issue nowdays.

Because exhaustive observation is time-consuming, if we want to perform the task efficiently, it is essential to choose the points of view that provides more information. One possible solution for robotic applications, is a view planning algorithm which can automatically plan the location of the sensor [7]. To obtain a complete reconstruction of the object, the sensors must be placed on different points of view that cover the entire surface of the object.

A. *Vision-based perception*

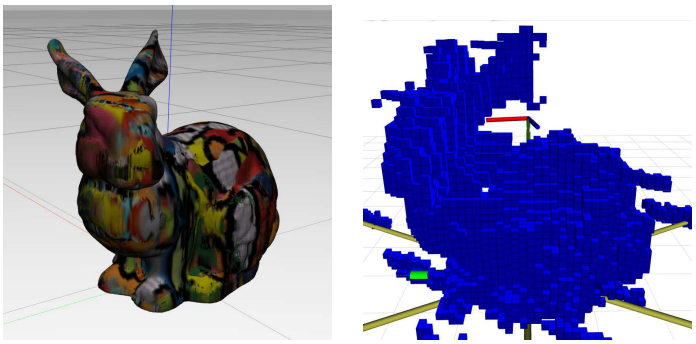
In robotic applications, the visual perception of the object to be manipulated is fundamental. Here, it is useful to have 3D information from the scene, and this data can be obtained from stereo cameras. The stereo vision technique obtains information about objects in the environment from images in three dimensions. To obtain this 3D information, point clouds are used, as shown in Fig. 1. A point cloud is a three-dimensional representation of points based on a data structure. Conceptually, point clouds are used to model both shape and location of an object digitally on a three-dimensional axis.

3D enviroment processing: As a result of all the information of the enviroment observed by 3D sensors, a point cloud with the information is generated. To process this information, several algorithms are developed to build a complete 3D model of the scene. One basic element of these algorithms is Octree.

B. *OctoMap: A 3D mapping framework based on octrees*

OctoMap is an open source framework for three-dimensional mapping. This approach is based on the implementation of both [4] and [5] that use an efficient data structure with Octrees. Using probabilistic occupancy estimation, the approach is able to represent volumetric 3D models that include free and unknown areas.

Probabilistic representation: The sensors of the robots carries out the construction of 3D maps by measuring 3D distances in the environment with respect to the object to be reconstructed. These measurements are presented with uncertainty, in addition, there may be random measurements caused by dynamic objects or reflections. In order to generate an accurate model from these measurements, the uncertainty is approached in a probabilistic manner. In this way, by merging all the measurements taken, a robust estimate of the object is obtained. Note that the probabilistic fusion of sensory information allows the implementation of multiple sensors and consequently, a multi-robot scenario.



(a) Bunny Stanford model (b) Voxels ocuppied cells

Fig. 2. *Output model of our generic system for the bunny object. Image a) shows the synthetic model of the bunny, while image b) shows the reconstructed 3D object with the occupied voxels.*

Full 3D Modelling: Octomap builds the volumetric map of an object from a point cloud as a 3D grid based on an octree structure based in voxels. A voxel is a cube containing points clouds. These cubes are placed as a grid along the entire point cloud to get a homogeneous and lower resolution than the starting one. As discussed in the section II-A, to generate a complete model of the object, Octomap receives as input data point clouds, and provides as output the information for the 3D model to be visualized (Fig. 2). Octomap is able to represent both occupied areas and free space in a three-dimensional environment.

C. *Next Best View Problem*

The planning of sensor views based on 3D sensor data is often referred to as planning the next-best-view (NBV) [7]. The key point of NBV is the decision the robot must make to locate a vision-based sensor. Despite all the research on this problem, as mentioned in [3] and [5], a predefined model of the target object is usually required.

Generally speaking, when it comes to the NBV problem, there are always two assumptions that are common to all approaches. One of them is that the NBV problem is located within a workspace with a set of views sampled on a model of a geometric figure (cylinder, sphere), which is predefined. The other one is that these approximations do not take into account that the performance of the sensor depends on the distance to the surface of the target.

III. Problem Addressed

We first consider that the object from which we generate a 3D model is rigid and is located at the center of an empty simulation environment. The object is a generical representation of a static solid on the ground plane. For the volumetric reconstruction task, we define that each robot mounts a pair of vision-based sensors. These sensors provide the sensory information necessary to carry out this task. Mostly, sensory information is translated as images of the target object. In order to build the volumetric map of the object it is necessary to acquire several views around the target object and then generate the images of its entire surface. Therefore, it is

necessary to have a positioning mechanism for the cameras that places them in these views. Within this framework, the pairs of stereo cameras are placed around the target object according to a strategy to be chosen. The strategy we consider consists of a circular formation that encloses the target object. Thus, we obtain a set of images that covers the entire surface of the object and gives *Octomap* a general view of it.

Mainly, the framework of our problem is an object that presents a dynamic behavior, of which we intend to construct a 3D model instantly, in order to observe the changes produced in its surface. Then, one of the objectives is to design a multi-camera architecture, in which we have multiple points of view around the object. On the other hand, another objective is to find a strategy for the cameras, which allows us to move the perception system tracking the target without losing sight of the object. In the following subsection we explain an example strategy on how we have tackled this problem.

A. *Multirobot Strategy*

In a first approximation, we distribute a group of stereo cameras according to a circular formation that encloses the object [6]. This means that the cameras keep their relative positions in formation with respect to the target object, at the same time that they fulfill the objective within everyone's field of vision. We choose to distribute the robots evenly along the circumference forming a regular polygon. The shape of the polygon depends on the number of robots you wish to have.

Given a moving target that follows a previously defined path in $\mathbb{R}^2$, we consider its positions $q_t(t) = (x_t(t), y_t(t))^T$ and its orientation $\phi_t(t) \in \mathbb{R}$. These coordinates are found in a global frame of reference, and we also assume that the target moves according to the unicycle kinematics:

$$\dot{x}_t = v_t\, cos\phi_t, \quad \dot{y}_t = v_t\, sin\phi_t, \quad \dot{\phi}_t = w_t, \tag{1}$$

where $v_t(t) \in \mathbb{R}$ and $\omega_t(t) \in \mathbb{R}$ are the linear and angular velocity of the target.

Consequently, in order to enclose and track the object, we also consider robots in $\mathbb{R}^2$. Its position and orientation are $q_i(t) = (x_i(t), y_i(t))^T$ and $\phi_i(t) \in \mathbb{R}$, with $i = 1, ..., N$. These robots follow the the unicycle kinematics:

$$\dot{x}_i = v_i\, cos\phi_i, \quad \dot{y}_i = v_i\, sin\phi_i, \quad \dot{\phi}_i = w_i, \tag{2}$$

where $v_i(t) \in \mathbb{R}$ and $\omega_i(t) \in \mathbb{R}$ are the linear and angular velocities of the robots. The scale of the formation is defined by its radius $d_i = d(t)$. We also consider that all robots are pointing at the object at all times and that the object is in the center of the circle. Therefore, the value d is constant for all the robots in the formation, equivalent to the distance of each robot from the target. To illustrate this, we expose the coordinates of each robot with respect to the reference of the moving target:

$$x_{0i}(t) = d\, cos\phi_i, \quad and \quad y_{0i} = d\, sin\phi_i. \tag{3}$$

In the strategy proposed in [6], we design the possible trajectories q_{ri} that each robot needs to accomplish the task. Figure 3 shows an example of a well-defined multi-camera strategy in case of six cameras ($N = 6$). See details in [6].

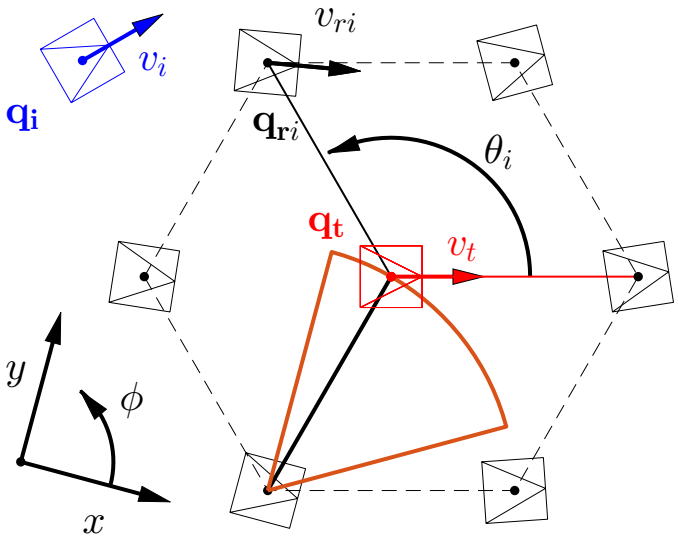


Fig. 3. *Constant scale formation strategy: The parameters involved the multi-robot strategy are depicted. We need to define the reference trajectories* $\mathbf{q_{ri}}$ *for the robots* $\mathbf{q_i}$ *to maintain a circular formation enclosing the target* $\mathbf{q_t}$ *and maintaining FOV constraints. Each robot* $\mathbf{q_i}$ *will track its reference trajectory* $\mathbf{q_i}$*. The FOV of the onboard fixed camera of one robot is also shown.*

IV. System Architecture

Our work consists of two well-defined modules, depending on the task they carry out. The objective of the implementation of these programs is to perceive at each instant of time, variations in the surface of the target object, either because it is deformed, or because it is moving. To address this issue, we split the problem into two separate processes and use a different system architecture than the reconstruction task. Our systems are built on the framework presented in [5] and in [6]; as described in the sections II and III-A. Both systems can be replaced by any other without affecting each other, only by adapting the architecture of the system. Conceptually, it is a modular system architecture consisting of different independent modules that interact through communication interfaces, as shown in Fig. 4. The method is summarized in algorithm 1.

A. *Instantaneous Perception and 3D Reconstruction*

We approach the reconstruction task as an iterative process at every instant; in addition, the architecture of the module that implements this task is a generic system based on ROS. We divided the multi-robot reconstruction task into three parts (i) *reconstruction of instantaneous volumetric models*, (ii) *multi-camera view planner* (Strategy) and (iii) *multi-camera positioning mechanism*. In this section we present an overview of our system, which is built on the frame observed in [5].

- The **Perception System** is responsible for data acquisition and its processing. The components which composes this system are the *sensor module* and the *3D perception Module*. In the multi-camera scenario the output are as many point-cloud of observed 3D points as the number of cameras we have.

Algorithm 1 Instantaneous Reconstruction with Pose Planning

Input: Number of Robots (N) .
Output: Volumetric map of the object.

Multi-robot Planner computes next position of the robots.
ROS Interface establishes communication with the robots.
Multi-robot Planner publishes topic position i to robot i.
ROS Interface subscribes and receives the position i of the robot i.
while $t < t_{final}$ **do**
  for $i = 1$ **to** N **do**
    ROS Interface commands the *Robot Driver* to move the robot i to the position i in *Gazebo*.
    Gazebo gets the images from the sensor of the robot.
    World Representation Module collects the images from *Gazebo* and updates the *Octomap*.
    Multi-robot Planner publishes position i to robot i.
    ROS Interface subscribes and receives the position i of the robot i.
  end for
end while

- The **World Representation Module** registers all the data perceived within the map, given access to the current map. The images taken from previous iterations need to be registered and merged in order to build a common model. This module also carries out the reconstruction task with *OctoMap* [4].
- The **Motion Planning and Control** implements the camera positioning mechanism that locates the robot, and the *Robot Interface* that provides the interface between the *ROS Interface Planner* and the robot.
- The **ROS Interface Module** carries out path planning and sends the new viewpoint to the robot, which receives it using the *Robot Interface*. This new view is where the robot must move the corresponding sensor.

B. Tracking and Enclosing of the Target

In order to provide the **ROS Interface Module** the new location of each camera and also perceive at every moment of time the changes experienced by the object in the environment, we need a program that follows a multi-robot strategy that calculates a geometric formation of the robots around the object and their respective trajectories. Here we implement a program that presents a *Matlab-based* module. As mentioned in the section III-A, the objective is to enclose and track the object in order to build the volumetric map.

- The **Multi-robot Planner Module** computes the relative coordinates of the robots (2)-(3) with respect to the target (1). As a result, we generate a circular formation around the object with the N cameras. Thus, the output is the N views at each instant. We go through each of the generated viewpoint data and send them to the *ROS Interface Module*. Note that each viewpoint data has as many registers as robots times the number of time instants.

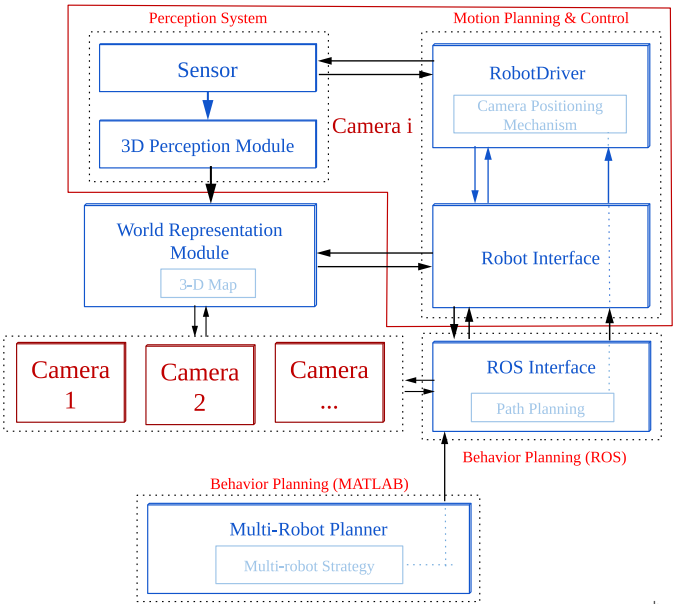


Fig. 4. *Conceptual system overview: Main modules, and their communication interfaces (arrows) are shown. At the bottom of the image is the Matlab-based system, which sends the calculated locations to the ROS-based system. The ROS Interface Module receives the positions and establishes communication with each camera. The Robot Interface Module receives its respective location and the Camera Positioning Mechanism is responsible for placing it in that location. At the center of the image is the World Representation module, which generates the 3D model using Octomap. This module builds the volumetric map from the sensor data of each camera.*

C. Communication

In order to communicate both modules, we developed a system architecture based on ROS. Thus, the *ROS Interface Module* is able to receive from the *Multi-robot Planner Module*, the id of the camera model to be moved according to the defined strategy, and its position. We use topic-based communication, with their respective publishers and subscribers.

We also consider that the perception of the object is instantaneous, although our approximation takes N instants (as many as there are N robots) to form a complete 3D map. Notice that our system only moves one robot per instant in a loop, so to finish the geometric formation around the object with N robots, it takes N instants to build the complete perception.

V. Simulations

In this section we will describe two simulated examples. We have decided to separate it into two parts, according to the methodology carried out. The first simulation combines both tasks, the volumetric reconstruction and the positioning mechanism of the camera, to illustrate an scenario where only one robot operates. In the second simulation we extend the method to a multi-robot environment and implement the strategy that allows us to reduce the computational cost and time of generating the 3D model.

A. *Simulations setup*

Our architecture is designed in such a way that each module works independently of the others, so if any problem arises it can be easily debugged. Moreover, it is easy to include additional requirements for the simulations or integrate new design changes.

The simulation of the reconstruction scene includes an object placed on an empty world (Fig. 5.) and an depth sensor that provides information of each view. The model used for the object is available on-line: the Stanford bunny (Fig. 2). Around the object we generate the robot system. Each robot, is a free-flying stereo camera with 6 DoF. The flying-stereo-cam is an RGB-D sensor that obtains an image of part of the scenario. All the simulations are carried out in Gazebo, in this environment the stereo processing can be carried out using ROS (Robot Operating System). In both simulations we start by positioning the sensors in selected pre-defined views. The output of the sensor is a point cloud that is rerouted by the *Robot Interface Module* across several services that make use of it, to the *Octomap Representation Module*. The probabilistic volumetric map is based in the approach presented in [3], where every pair of stereo-cameras share their computed point-cloud message to be integrated on it. The camera positioning planner is based in [6], which locates the cameras forming a hexagonal formation (Fig. 3).

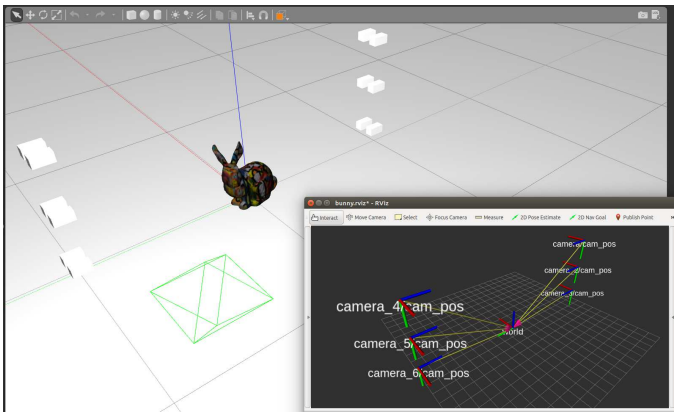


Fig. 5. *Simulation reconstruction scenario: Both bunny object and stereo camera pairs are simulated in Gazebo and shown by Rviz.*

B. *Example of volumetric reconstruction*

First of all, we explain how the concepts presented in sections II and IV are implemented in the present application. Our goal is to model a 3D object using a robot (a stereo camera). For the reconstruction process, we use the volumetric map of probabilities based on a mesh of voxels, implemented as an approximation of [4], as well as, the positioning mechanism of the camera, developed from the *ROS Interface Module* of [5]. We have approximated this mechanism by means of a *ROS-based* implementation, in communication with the N robots.

Within this framework, the object is a priori unknown and spatially limited, as it does not deform or move. Starting from a system that iterated along a set of views to obtain the next best position in which to place the camera, we have eliminated the sequential loop of the algorithm. Our goal is to receive views of an external algorithm, which does not have to select the best view, the strategy is illustrated later. Once we have received the location (position and orientation) our method places the sensor in it, taking an image for the volumetric map. To build the complete volumetric map of the target object, *Octomap* needs point clouds as input data. These point clouds are obtained by processing a set of images taken by several cameras. As output, Octomap returns a data structure of the target object that we visualize in *Rviz*. Note that in order not to lose sight of the object at any time, we impose a geometrical constraint so that the orientation of the camera is always pointing to the center of the object.

C. *Example of a multi-camera scenario*

Secondly, we describe the proposed simulation for a multi-robot scenario, using the planning algorithm based on [6], which we discussed in section III-A. Although the number of cameras used in each execution of the program can be arbitrarialy chosen, in this simulation, we consider a team of $N = 6$ agents, forming a hexagon. The circular formation was observed in Fig. 3. In order to simulate changes in the rabbit model, we make it move following an elliptical trajectory predefined by us. As discussed in III-A, we calculate the trajectories of the target object and the multi-camera team, as well as the values of the formation radius d to accomplish the task. Once the trajectories have been calculated, we carry out a follow-up control. To calculate the positions at which to move, each pair of cameras needs to estimate their relative location with respect to the target object (see Fig. 6). In Fig. 7 we present an example of the commented strategy, with a target object that follows an ellipsoidal path and the formation of cameras tracking it.

Note that the evaluation of the simulations is based on a visual criterion. However, an interesting analysis for future work could be the numerical evaluation of the error of *surface coverage* over time, from which the effect of the occlusions on the generated 3D model would be studied.

VI. Conclusions

In this work, we have developed an instantaneous reconstruction framework based on [5] which is based on a multi-robot approach. We have divided our approach into two independent modules: the multi-camera volumetric reconstruction and multi-camera strategy. We also proposed the connection procedure between them, which ensures the adaptability of the architecture of our modular system to other robot platforms. Given an object with a dynamic behavior that follows a path, our framework of instantaneous reconstruction considers the problem of full perception and reconstruction of a probabilistic 3D model of that object in every instant of time, by means of N stereo cameras placed around it.

We also consider the type of the trajectories that enclose and track a moving target with a multi-robot system while

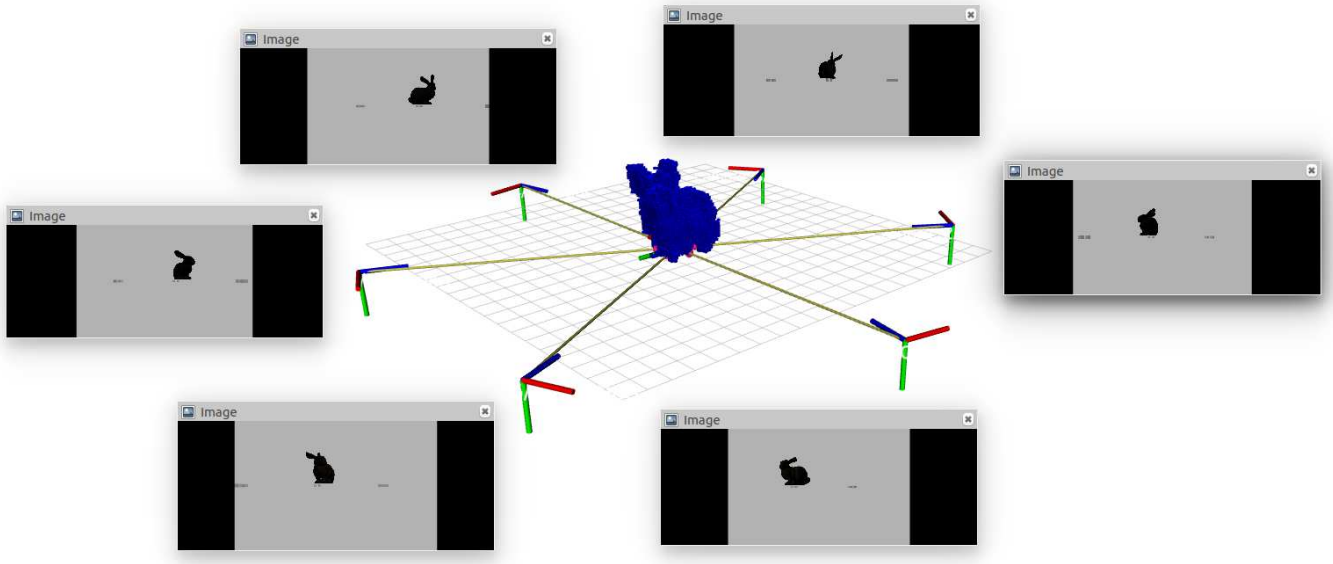
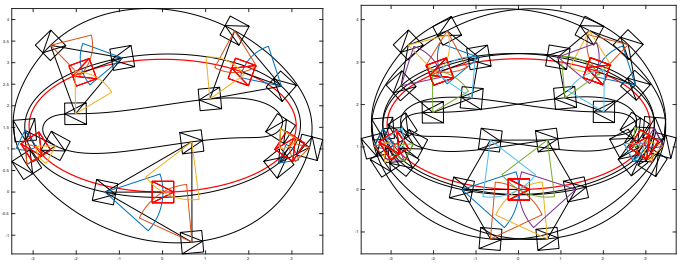


Fig. 6. *Reconstruction in simulation: The full model for the Stanford bunny dataset is observed, where the occupied voxels are painted in blue. The multi-camera hexagonal-shape formation is also visualized. In the task of instantaneous perception of an object, the field of view of each camera covers a portion of the surface of the object at each instant of time. By merging these observations, the object is fully perceived.*



(a) $N = 3$ (b) $N = 6$

Fig. 7. *The image shows a simulation that includes and tracks a target with ellipsoidal motion. Two cases are shown from left to right: the number of robots chosen is $N = 3$ and $N = 6$, respectively. The simulation depicts the movement of the target and the robots surrounding it. In addition, the polygon of the formation and the wedges (FOV) of each robot are also represented during some instants of time.*

obeying the restrictions of movement and FOV. We choose a circular formation that encloses the target object due to a criterion of simplicity and absence of occlusions between cameras. The proposed architecture is general and any other formation strategy can be easily tested by taking advantage of the modular design on the *ROS-based* implementation.

In this work, we visually evaluate the results, focusing our attention on whether the strategy adopted for the arrangement of the cameras around the object produces the object model, without taking into account the accuracy of the observation. The performance of the implemented strategy can be visually evaluated through the *Rviz* program, where the tracking of the dynamic object carried out by the multi-camera system and the corresponding volumetric reconstruction of the model is shown.

Future directions in the field of instantaneous volumetric reconstruction of dynamic objects using multiple robots include many different topics. Some of these aspects are shared among the above strategies, and could be summarized as follows: improvements to the system framework efficiency; evaluation of different multi-robot strategies; and optimization of the communication interface between the components.

REFERENCES

[1] Aloimonos, J., Weiss, I. and Bandyopadhyay, A. “Active Vision,” in International Journal Computer Vision (1988), Volume 1, Issue 4, pp 333-356.

[2] Chen, S., Li, Y., and Kwok, N. M. “Active vision in robotic systems: A survey of recent developments,” in Int J Robotics Research (2011), Volume 30, Issue 11, pp 1343-1377.

[3] Delmerico, J., Isler, S., Sabzevari, R. and Scaramuzza, D. “A comparison of volumetric information gain metrics for active 3D object reconstruction,” in Autonomous Robots (2018), Volume 42, Issue 2, pp 197-208.

[4] Hornung, A., Wurm, K.M., Bennewitz, M., Stachniss, C., and Burgard, W. “OctoMap: An efficient probabilistic 3D mapping framework based on octrees,” in Autonomous Robots (2013), Volume 34, Issue 3, pp 189-206.

[5] Isler, S., Sabzevari, R., Delmerico, J., and Scaramuzza, D. “An information gain formulation for active volumetric 3D reconstruction,” in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, 2016, pp. 3477-3484.

[6] López-Nicolás, G., Aranda, and M., Mezouar, Y. “Formation of differential-drive vehicles with field-of-view constraints for enclosing a moving target,” in IEEE International Conference on Robotics and Automation (ICRA), pp. 261-266, Singapore, 2017.

[7] Scott, W., Roth, G., and Rivest, J. F. “View planning for automated three-dimensional object reconstruction and inspection,” in ACM Comput. Surv. 35, 1 (March 2003), 64-96.