Experimental multi-camera setup for perception of dynamic objects

Rafael Herguedas, Gonzalo López-Nicolás and Carlos Sagüés

***Abstract*— Currently, perception and manipulation of dynamic objects represent an open research problem. In this paper, we show a proof of concept of a multi-camera robotic setup which is intended to perform coverage of dynamic objects. The system includes a set of RGB-D cameras, which are positioned and oriented to cover the object's contour as required in terms of visibility. An algorithm of a previous study allows us to minimize and configure the cameras so that collisions and occlusions are avoided. We test the validity of the platform with the Robot Operating System (ROS) in simulations with the software Gazebo and in real experiments with Intel RealSense modules.**

I. Introduction

Perception of dynamic objects is a challenging process due to their complex and highly variable behavior. A continuous and complete measure of the object can be required in robotic applications as transport or manipulation of deformable objects [1]. In particular, when dealing with large size or multiple dynamic objects, cooperative manipulation and perception units seem a necessary requirement [2], [3]. These necessities motivated our previous work [4], in which we proposed an algorithm for computing the position and orientation of a minimal set of cameras so that the contour of a deformable object is detected as required in terms of visibility. In this document, we describe and test a multi-camera setup in order to illustrate the performance of the former algorithm. Potential applications of this setup include, but are not limited to, coverage of deformable parts for manipulation, active visual objects inspection and surface treatment if some of the cameras are substituted by other tools as spray guns.

Relevant works related to the one we describe in this document are reported next. Based on a new visual distance, a multi-camera system for coverage of 3D scenes is proposed in [5]. In [6] an algorithm for nondecreasing collective coverage of 3D structures is developed and tested in real experiments. A multi-camera architecture based in the Robot Operating System (ROS) can be found in [7], for shape reconstruction of 3D rigid dynamic objects. Shape recovery is also tackled in [8] for unknown deformable objects by means of a multi-camera system, which is reconfigured in order to maximize the visible area. In [9] the authors propose an optimal perception and tracking system with an RGB-D module focused on deformable objects.

Fig. 1. Prototype of a mobile robotic collaborative manipulator named *Campero*. This platfform is now commercially available as the model *RB-EKEN BASE*, by Robotnik Automation [10]. A UR10 collaborative arm with an Intel RealSense D435 camera is installed on top of the mobile platform, which also carries two laser scanners Sick S300 and a Axis M5525-E PTZ camera, among other equipment. The setup we envision consists of multiple *Campero* manipulators with cameras in hand.

The main contribution of [4] is a new perception system of deformable objects with a minimal set of mobile manipulators and sensors. The system adapts its configuration to achieve the perception objectives avoiding collisions and occlusions. The interest of this document is the description of a multi-camera setup that illustrates and tests the feasibility of the proposed scheme [4] in motivating applications.

II. Background

In this section we provide a high level explanation of the algorithm that computes the positions and orientations of the cameras in the setup [4]. The dynamic object to be detected is at the center of the scene, and an approximate sampled contour of it is known. Then, the number of sensors that must detect each contour section (*target visibility*) is set depending on the necessities of the task. A safety distance

The authors are with Instituto de Investigación en Ingeniería de Aragón, Universidad de Zaragoza, Spain. {rherguedas, gonlopez, csagues} @unizar.es

This work was supported by projects COMMANDIA SOE2/P1/F0638 (Interreg Sudoe Programme, ERDF), PGC2018-098719-B-I00 (MCIU/AEI/FEDER, UE) and DGA_T45-17R (Gobierno de Aragón). The first author was partially supported by the EU through the European Social Fund (ESF) "Construyendo Europa desde Aragón".

from the centroid of the object, which guarantees clear perception and maneuverability, and the cameras' field of view (FOV) constraints are given to the algorithm. After this, the algorithm is given a certain number of cameras and it minimizes a cost function with three different terms, which incorporates the *target visibility*, the collision avoidance and the robust perception aspects into the problem. While the *target visibility* is not achieved, the number of cameras is increased and the optimization is repeated. At the end of this process, the minimum number of sensors that achieve the *target visibility* and their near-optimal positions and orientations around the object are provided. This process must be repeated for each deformation instant. In summary, the output of the described algorithm is the configuration of each camera for each time instant to provide perception of the dynamic object as required.

In this approach, we can guarantee the performance of the system with the following assumptions:

1) The 3D positions and orientations of the cameras and the object are known at any time in a global reference frame.
2) The object to be detected is slowly and smoothly deformed.
3) The effects of external factors affecting visibility, as shadows or vibrations, are considered negligible under normal working conditions.

III. Setup Description

A. Software architecture

The core of the setup's architecture is the Robotic Operating System (ROS), a high level task supervisor for scheduling the connections and the information exchange between the different elements of the system. ROS connects in a common framework the low level software layer with the hardware layer. The algorithm for computing the minimal number of necessary sensors and their near-optimal configurations is implemented in Matlab with the *pattern search* optimizer. Matlab is connected to ROS by means of the *ROS Toolbox*, which allows not only to send the positions of the cameras to the manipulators, but also to receive an updated point cloud of the object. This requires some preprocessing for extracting the object points from the point clouds of the environment, obtained with the RGB-D cameras. Point cloud aggregation and object segmentation are performed by means of the OpenCV software library, which is also connected to ROS. Although the Matlab software includes algorithms for point cloud segmentation, a mixed strategy with both Matlab and OpenCV is convenient for two main reasons: (i) the processor running Matlab can be dedicated to the algorithm, while secondary processing is performed in another processor with OpenCV, and (ii) the modular software architecture allows to easily substitute some of the blocks in the future with more advanced algorithms.

The hardware layer, consisting in the mobile manipulators and the cameras, can be substituted by virtual components in a simulation environment. Gazebo is a robotic simulator with connection to the ROS system that incorporates different physics engines, and allows to test and display in a graphic interface multiple sensors and actuators. By combining Gazebo with the ROS package Rviz, the sensor readings can also be displayed and monitored.

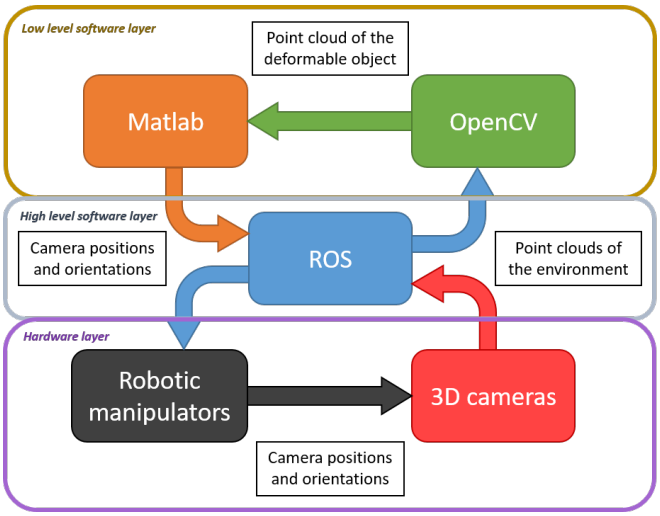


Fig. 2. Main software and hardware elements of the coverage system, where the different layers and information flows between them are depicted.

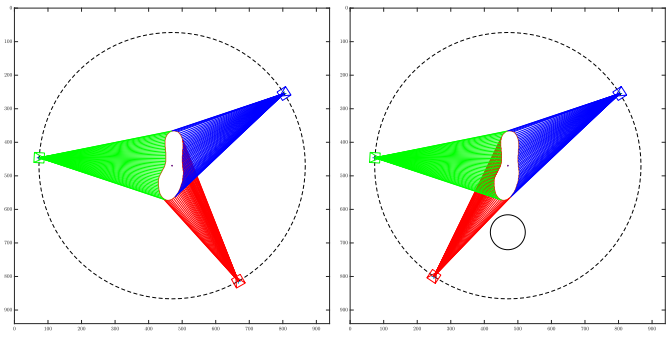
Figure 2 shows the main elements of the proposed setup with the information flow in the network.

B. Experimental setup

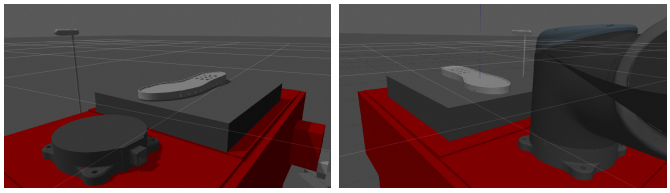
The platform we envision consists of a set of RGB-D cameras and robotic mobile manipulators. The cameras are mounted on the frame and the arm of the manipulators, so that they can reach different positions and orientations at the same time that the robots hold the object and perform different operations on it (see Fig. 1 of the *Campero* manipulator).

The algorithm provides the minimimum number of cameras and their near-optimal configurations in the 2D space. Thus, displacements and rotations of the cameras can be given by the displacements and rotations of the arms' end effectors where they are installed or, alternatively, by the position and orientation of the mobile platform. Depending on the characteristics of the manipulators, the task or the environmental constraints, whether the controllers of the arms, the controllers of the mobile platforms or both will receive the Matlab configuration commands through ROS. In the current development of the platform no specific workcells are considered, and all solutions are equally accepted.

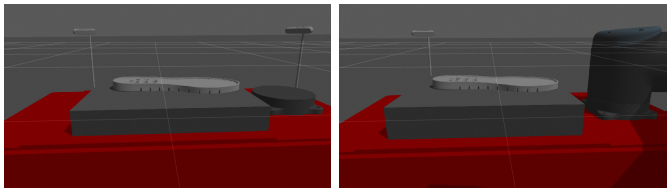
As reported in Assumption 1, the position and orientation of the cameras and the object must be known in a global reference frame. An accurate map of the environment can be a useful tool to locate the cameras with respect to it. In case the global map of the environment is not available, positioning systems based on visual markers, which are versatile and easy to implement, can also be installed.



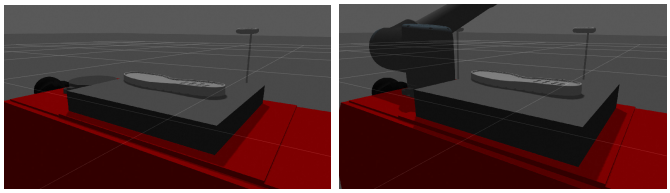
(a) Cameras' configurations without obstacles and with the occluding arm



(b) Bottom camera views (red)



(c) Left camera views (green)

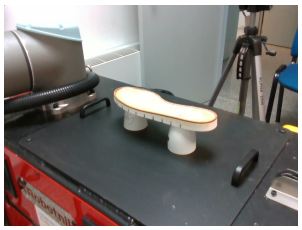


(d) Right camera views (blue)

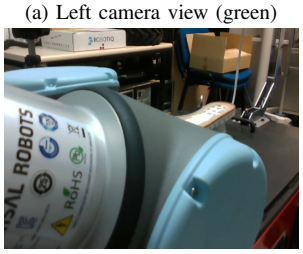
Fig. 3. The solutions from the algorithm in Matlab are shown in (a), for the setup with no occluders at the left side and the setup with the UR10 arm as an occluder at the right side. From (b) to (d), the color images from the RealSense cameras in the occluder-free Gazebo setup are depicted at the left side. At the right side, we provide the same outputs in the Gazebo setup with the occluding UR10 arm. It can be seen that the virtual platform is able to detect the complete contour of the sole in both scenarios (best seen in color).

For the proof-of-concept experiments, a simplified version of this setup is considered. In both simulations and real experiments the cameras are not installed in robotic arms, but they rest at the top of static camera holders, and only one mobile manipulator *Campero* is included to support the objects. The camera model we utilize in simulation is the Intel RealSense D415 module, and for the real experiments we utilize the Intel RealSense D435 module. These RGB-D modules open numerous possibilities for perception due to the rich information they provide, but for the current tests only the color images of the RealSense stereo left cameras are analyzed. In this case, color images are explanatory enough for the purpose of showing the overall performance of the coverage system. An external PC controls the image

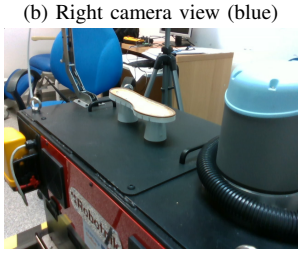
(a) Left camera view (green)



(b) Right camera view (blue)



(c) First bottom camera view (red)



(d) Second bottom camera view (red)

Fig. 4. A simplified version of the real platform is tested with the configurations in Fig. 3. We can see that the configuration of the bottom camera is occluded in (c) when the arm is not included in the algorithm. However, the full contour of the sole is detected when the configuration of this camera is updated including the arm as an obstacle (d).

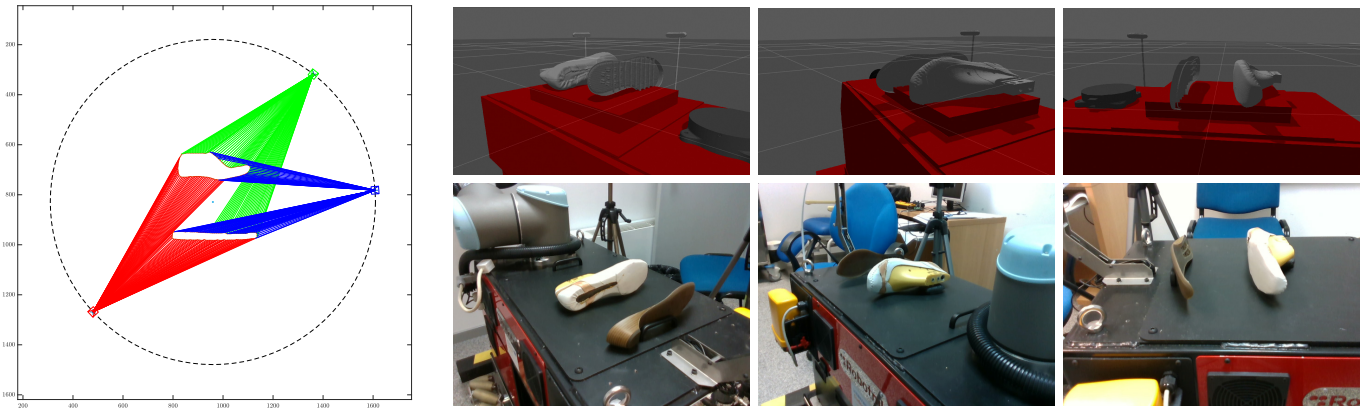
acquisition process as well as other input-output flows of the platform. In the future, we plan to include the OpenCV module to the workflow for extracting the contour of the object from the set of images.

C. Simulation and experimental results

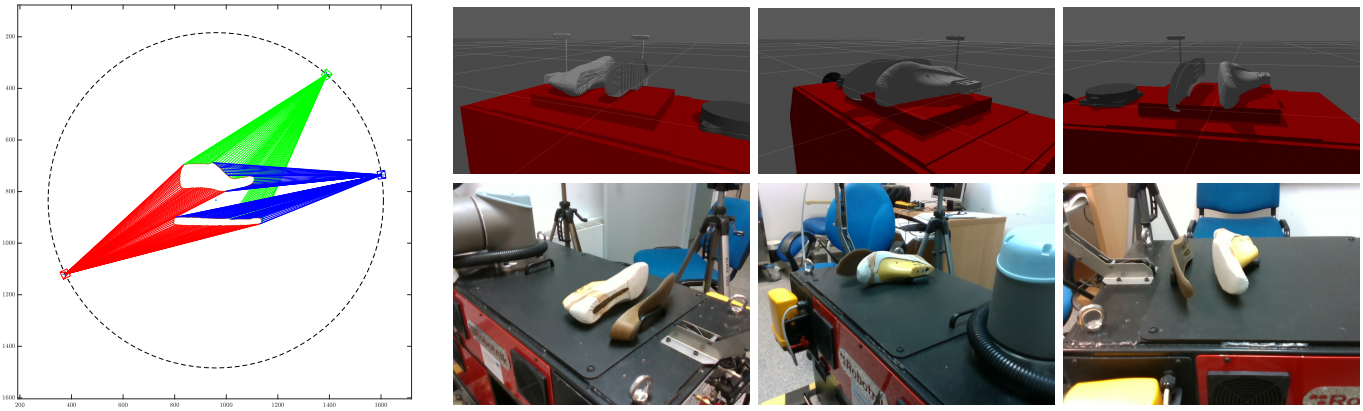
In this section we report the results from different proofs of concept we have performed with the simplified version of the proposed setup. Firstly, we study the setup in simulations with ROS 1.12.15, Gazebo v7.0.0 and Matlab R2020a, which allow us to quickly test our system's performance with ideal conditions in virtual environments. The main steps involved in these simulations are summarized in Algorithm 1. After that, we reproduce these simulations in an approximate way in real experiments, in order to evaluate how the real world disturbances (shadows, vibrations, spatial constraints...) affect the system.

Algorithm 1 Simulate the experimental setup in Gazebo.

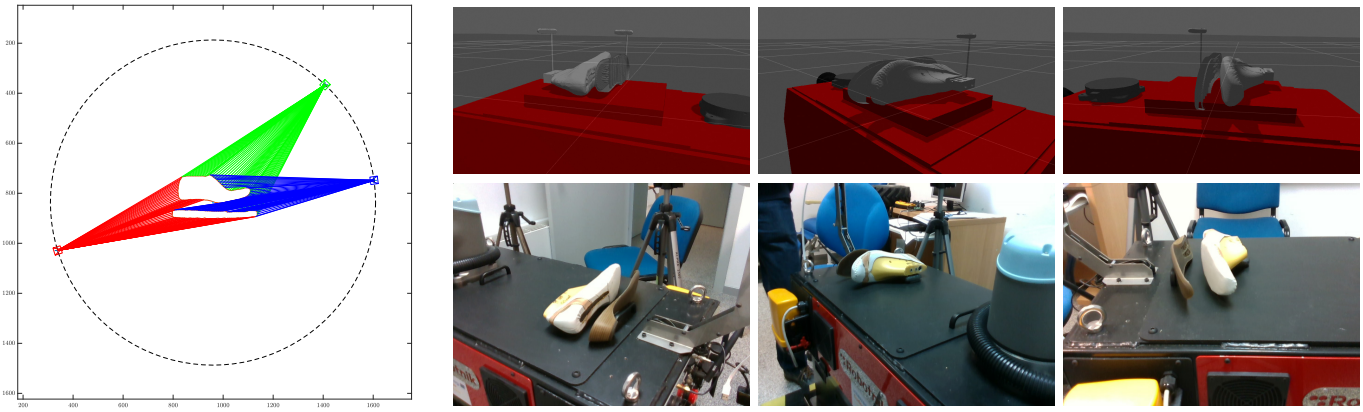
1: **while** $t_{sim} < t_{final}$ **do**
2: Send the estimated contour of the object and the problem constraints to Matlab.
3: Compute the minimum number of cameras and their configurations.
4: Send the number of cameras and the configurations to Gazebo.
5: Spawn the cameras in the virtual environment with the computed configurations.
6: Image acquisition from the cameras.
7: **end while**



(a) Cameras' configurations for the maximum sole-last separation



(b) Cameras' configurations for the intermediate sole-last separation



(c) Cameras' configurations for the minimum sole-last separation

Fig. 5. The solutions from the algorithm in Matlab are shown at the left side for three steps of the shoe assembly task. At the right side of each one of the previous images, two rows of images are depicted. The first rows show the color images from the RealSense D415 modules in the Gazebo simulator, and the second rows show the color images from the RealSense D435 modules in the real scenario, where the setup parameters are approximately the same as in simulation. It can be seen that the virtual platform tries to cover as much contour as possible of both parts with the three cameras in the three steps (best seen in color).

The behavior of the system against occlusions is tested with a static thin sole. This object is positioned over the surface of the *Campero*, which mounts an UR10 arm that may produce occlusions during the operations, and the objective is detecting the complete contour of the sole with a minimum number of cameras. Figure 3 shows the coverage results with and without the occluding arm. In the first row, (a) contains the results of the algorithm in Matlab, where the 2D sampled contour of the object appears at the center of the image. The dashed-line circumference indicates the perimeter around the objects where the cameras, represented as oriented boxes in different colors, are allowed to move. The beams connecting the cameras' centers with the detected vertexes of the contour are also depicted. In the following rows, (b), (c) and (d) show the color images from the cameras in the 3D virtual environment of Gazebo. We can see that in order to detect the full contour, the red camera at the bottom of the first diagram moves to the left side, and avoids occlusions. Figure 4 depicts the view of the cameras in the real setup: the left and right cameras with the configurations from Fig. 3 and the view of the bottom camera with the first configuration, in the scenario without the UR10, and the second configuration where the UR10 robot is included. As expected, the contour of the sole is occluded if the configuration of the bottom camera is not updated.

Simulating deformation in Gazebo is a complex process in which the object must be converted to an articulated structure. Thus, instead of testing the platform with a deformable object we have simulated a shoe assembly process, where the sole and the last are joined together with glue. Three steps of this task are considered, in which the two parts progressively approach to each other. We can see in Fig. 5 the solutions of the algorithm in Matlab, the color images of the RealSense D415 modules in Gazebo and the color images of the RealSense D435 in the real experiment, from the configurations given by the algorithm. The detected contour of both parts is maximized while obeying the collision constraints and the limited FOV of the cameras.

IV. Conclusions and future work

In this paper, we describe an experimental setup for perception of dynamic objects. The number of necessary cameras and their configurations are given by an algorithm of a previous study, which includes formulation for collision and occlusion avoidance and robust perception. We have successfully tested a proof-of-concept setup in simulation and real experiments, for providing perception support in different scenarios and tasks. The future improvements of the proposed system include testing the setup with more mobile manipulators *Campero* with active roles. Our final objective is to develop a multi-robot setup for performing operations as manipulation, visual evaluation, quality control and surface treatment of dynamic objects of medium-large sizes.

References

[1] J. Sanchez, J. A. Corrales, B. C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[2] A. Khan, B. Rinner, and A. Cavallaro, "Cooperative Robots to Observe Moving Targets: Review," *IEEE Transactions on Cybernetics*, vol. 48, no. 1, pp. 187–198, 2018.

[3] C. Robin and S. Lacroix, "Multi-robot target detection and tracking: taxonomy and survey," *Autonomous Robots*, vol. 40, no. 4, pp. 729–760, 2016.

[4] R. Herguedas, G. Lopez-Nicolas, and C. Sagues, "Multi-camera coverage of deformable contour shapes," in *IEEE International Conference on Automation Science and Engineering (CASE)*, Aug. 2019, pp. 1597–1602.

[5] X. Zhang, X. Chen, F. Farzadpour, and Y. Fang, "A Visual Distance Approach for Multicamera Deployment With Coverage Optimization," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 3, pp. 1007–1018, jun 2018.

[6] A. Adaldo, S. S. Mansouri, C. Kanellakis, D. V. Dimarogonas, K. H. Johansson, and G. Nikolakopoulos, "Cooperative coverage for surveillance of 3d structures," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 1838–1845.

[7] E. Hernandez-Murillo, R. Aragues, and G. Lopez-Nicolas, "Multi-camera architecture for perception strategies," in *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, sep 2019, pp. 1799–1804.

[8] E. Nuger and B. Benhabib, "Multi-Camera Active-Vision for Markerless Shape Recovery of Unknown Deforming Objects," *Journal of Intelligent and Robotic Systems: Theory and Applications*, vol. 92, no. 2, pp. 223–264, 2018.

[9] I. Cuiral-Zueco and G. Lopez-Nicolas, "RGB-D Tracking and Optimal Perception of Deformable Objects," *IEEE Access*, vol. 8, pp. 136 884–136 897, 2020.

[10] R. Automation, "RB-EKEN BASE," https://www.robotnik.es/robots-moviles/rb-eken-base-2/, 2020.