Sensor Placement Optimization using Random Sample Consensus for Best Views Estimation

Conference Paper · April 2023

DOI: 10.1109/ICARSC58346.2023.10129597

5 authors, including:

Carlos Miguel Correia da Costa
Institute for Systems and Computer Engineering, Technology and Science (INESC TEC)
54 PUBLICATIONS **293** CITATIONS
SEE PROFILE

Germano Veiga
Institute for Systems and Computer Engineering, Technology and Science (INESC TEC)
101 PUBLICATIONS **888** CITATIONS
SEE PROFILE

Armando Jorge Sousa
University of Porto
138 PUBLICATIONS **859** CITATIONS
SEE PROFILE

Ulrike Thomas
Technische Universität Chemnitz
61 PUBLICATIONS **493** CITATIONS
SEE PROFILE

Sensor Placement Optimization using Random Sample Consensus for Best Views Estimation

Carlos M. Costa[1,2(✉)], Germano Veiga[1], Armando Sousa[1,2], Ulrike Thomas[3] and Luís Rocha[1]
[1]Centre for Robotics in Industry and Intelligent Systems of INESC TEC, Portugal
{carlos.m.costa, germano.veiga, luis.f.rocha}@inesctec.pt
[2]Faculty of Engineering of the University of Porto, Portugal
asousa@fe.up.pt
[3]Robotics and Human Machine Interaction Laboratory at the Technical University of Chemnitz, Germany
ulrike.thomas@etit.tu-chemnitz.de

***Abstract*—The estimation of a 3D sensor constellation for maximizing the observable surface area percentage of a given set of target objects is a challenging and combinatorial explosive problem that has a wide range of applications for perception tasks that may require gathering sensor information from multiple views due to environment occlusions. To tackle this problem, the Gazebo simulator was configured for accurately modeling 8 types of depth cameras with different hardware characteristics, such as image resolution, field of view, range of measurements and acquisition rate. Later on, several populations of depth sensors were deployed within 4 different testing environments targeting object recognition and bin picking applications with increasing level of occlusions and geometry complexity. The sensor populations were either uniformly or randomly inserted on a set of regions of interest in which useful sensor data could be retrieved and in which the real sensors could be installed or moved by a robotic arm. The proposed approach of using fusion of 3D point clouds from multiple sensors using color segmentation and voxel grid merging for fast surface area coverage computation, coupled with a random sample consensus algorithm for best views estimation, managed to quickly estimate useful sensor constellations for maximizing the observable surface area of a set of target objects, making it suitable to be used for deciding the type and spatial disposition of sensors and also guide movable 3D cameras for avoiding environment occlusions.**

***Index Terms*—Best views estimation, sensor placement optimization, random sample consensus, bin picking**

I. Introduction

Object recognition within environments with large and dynamic occlusions is a challenging task that can be tackled by either deploying an extensive and expensive sensor constellation or by actively moving a set of sensors within the environment in order to maximize the observable surface area of the target objects. This is a variant of the View Planning Problem (VPP) [1], which has a wide range of applications within the active perception domain, such as the estimation of the next best view for 3D scanning [2], object recognition with occlusions, exploration of unknown environments, deployment of sensor networks to monitor targets, among many others.

Within the active perception domain, several approaches have been proposed depending on the particular use cases. In [3] it was presented a probabilistic active planner using a partially observable decision process model for improving the perception of house hold objects that were going to be manipulated by a dual arm robot. For active perception of objects with similar 3D geometry but with unique 2D features, [4] introduced an active perception system for actively looking for bar code regions and unique text on the surface of the objects after having a preliminary estimation of the target object pose. To compute the next best sensor observation pose, it was uniformly generated a set of possible viewpoints on sections of the surface of a sphere and then it was selected the one which achieved the best observation utility that incorporates the quality of the pre-compute features along with the distance that is required to move the sensor from its current pose to the new sensor view observation pose. On the other hand, [5] also selected the best next view by balancing the expected information gain with the required sensor movement while using a VP-tree for performing object recognition and pose estimation. Another approach introduced in [6] targeted bin picking operations and relied on randomly deploying a set of possible views over the target objects and then selecting the one with the best trade off between information gain and sensor traveling cost. Unlike previous approaches that used image or geometric features, this system modeled each target object as a set of primitive shapes (such as planes, cylinders and spheres) that were assembled on a graph and recognized on the sensor data using a RANdom SAmple Consensus (RANSAC) algorithm. Another approach presented in [7] also starts with a randomly generated set of viewpoints for the estimation of the first best view, but them reduces the regions in which views are generated and favors frontiers between known and unknown environment sections that are stored in a voxel grid. The observation probability was modeled as a hidden Markov model and the posterior probability relied on a Bayes filter. In the end, the sensor view that achieves higher information gain is selected and the world model is updated to reflect the new observed space. In [8] it is presented another system for active bin picking that takes advantage of the accurate modeling of range sensors that was presented in [9], while [10] introduces strategies to generate sensor views targeted for object pose estimation.

Besides active perception, the best view estimation algorithms can also be used to actively explore the environment

979-8-3503-0121-2/23/$31.00

for mapping purposes using aerial vehicles in which the goal is to estimate the minimum set of sensor views that maximize the observation of the unknown space. The system proposed in [11] achieves this by using a receding horizon path planning algorithm while the approach presented in [12] expands the observation goals further and tries to find a given target object in a continuously updated environment. The planning of the set of views necessary to explore the environment relied on an octree for space modeling and randomly deployed observation views along frontier regions that were later on analyzed and selected based on the expected information gain and traveling distance of the mobile robot. These exploration goals were also taken to underwater environments by the work presented in [13], in which special care was taken to model the sensor data degradation over distance and the necessity of artificial light for deep sea exploration and mapping.

Another research domain that uses best view estimation algorithms is 3D scanning and reverse engineering [14]. In [15] it is introduced a volumetric 3D modeling scanning system that deploys a set of possible viewpoints on a tessellated sphere or a cube and then based on the expected information gain, the overlap of known and unknown regions, distance to previous selected view and orientation of the view to the observed surface, it selects the set of sensor views required to perform 3D scanning and surface reconstruction of the object. In [16], besides actively moving the sensor, it is used a robotic manipulator for grasping an object and then estimate the constellation of sensor views that maximize the amount of unknown object cells that can be observed while keeping a reasonable overlap with know regions. After finishing a set of observations, the system chooses another grasp configuration and tries to observe the remaining regions that were previously occluded by the robotic arm.

A related area to best view estimation is sensor deployment for monitoring extensive areas in order to track a set of interest objects or providing a communication infrastructure. In [17] it is introduced an adaptive 2D sensor placement and boundary estimation system for monitoring and tracking objects. The disposition of the sensors is based on signal propagation and area coverage and aims to track (with the minimum number of sensors) a given set of objects modeled as Gaussian mixture of models that are updated using a recursive distributive expectation maximization algorithm. Extending the sensor deployment to 3D, [18] provides an optimal range sensor placement approach for minimizing the target localization uncertainty using the Fisher information matrix.

With these goals and possible applications in mind, it was developed a system[1] for the estimation of the sensor constellation that maximizes the observable surface area (cost function) of a given set of target objects within a simulated scene with occluding geometry. By relying on simulated sensors and environments, the system allows quick evaluation of which types of sensors and which environment regions maximize the capture of 3D data for achieving better surface coverage of the target objects, making it suitable as a decision support system for helping the deployment of sensor constellations.

The development of the proposed system was split into 4 main stages. In the first step it was modeled the 3D scene geometry of both the target and occluding objects. For testing the capabilities of the system, 4 simulation environments were created, targeting active perception and bin picking applications. Then, a set of sensor populations was deployed in each environment within regions of interest in which useful sensor data could be retrieve (given the sensors characteristics and physical constraints of the real sensors). Each population was of a specific sensor type that simulated the main hardware characteristics of commercially available sensors, such as the depth camera resolution, its field of view, range of valid measurements and acquisition rate. The third step included the generation and analysis of the sensor data for each sensor. This included the extraction of the target objects point clouds using color segmentation (the target objects had a unique color material that was not affected by lighting effects), followed by the 3D projection of the 2D depth pixels using the pinhole camera model, which were later on transformed into the world coordinate system (for fast merging of data from different sensors) and filtered with a voxel grid. This filtering step was critical to ensure consistent surface area evaluation even when sensors with different image resolution where observing the same surface area at varying distances. This regular space partition assumes that too many points within a small region do not contribute to better 3D perception, and as such, a given surface cell can be considered as observed if it has at least one sensor measurement. This approach also allows to very efficiently compute the surface area coverage (by simply dividing the number of observed voxels by the number of expected surface voxels). Finally, in the forth stage, the best sensor constellation for each testing environment was estimated. When the goal was the selection of a single sensor, then the simulated sensor with the best surface coverage was selected. On the other hand, if several sensors could be used, a RANSAC approach was employed to estimate the N sensors that when merging and filtering their measurements managed to achieve the best surface coverage of the target objects.

The main contributions of the paper are the proposal of an efficient sensor fusion method that relies on color segmentation of the target objects followed by voxel grid merging, along with a RANSAC approach that computes a constellation of sensors that maximizes the surface area coverage of a set of target objects deployed in simulated environments. Moreover, for allowing future benchmarking of systems with similar goals, the testing environments and the implementation were made publicly available.

In the following section it will be presented how the 3D testing environments were created, including the sensors modeling and deployment. Then in Section III it will be introduced the algorithms used to process the sensor data and estimate the best sensor constellation, which will be supported by an experimental evaluation that will be discussed in Section IV. Finally, Section V will present the conclusions.

[1] https://github.com/carlosmccosta/sensor_placement_optimization

II. 3D SCENE MODELING

For being able to compute the surface coverage of a given set of simulated target objects that a given constellation of sensors can observe, it is necessary to model the 3D geometry of the scene objects. Moreover, the depth sensors must have a realistic data acquisition formulation that is representative of the real sensors. As such, the development of the proposed system started with the 3D modeling of the scene geometry, namely the environment objects and sensors 3D meshes using Computer Aided Design (CAD) systems. Later on, several types of depth sensors were modeled within the Gazebo simulator[2] in order to perform accurate 3D rendering of the scene and generate representative sensor data taking into consideration the specific characteristics of each type of sensor (such as image resolution, field of view and depth range) while also accounting for the occlusions that other objects in the environment might cause in relation to the target models that each sensor is trying to observe from its given view point. The next sections will present the modeling of the simulation worlds along with how the depth sensors were deployed within plausible regions of interest that take into account where the sensors can be placed in the real environment.

A. Environment modeling

For testing active perception and bin picking operations it was modeled 4 different simulation worlds. Within these environments it was deployed one or several target objects, which were instances of a starter motor CAD model with a unique green surface material that had no light effects, such as shading and shadows.

The first environment (shown in Figure 1), focused on an active perception task in which a starter motor was placed on top of a trolley and was being occluded by a human hand starting to grasp it. The goal of this environment was to simulate an active perception task, in which we may need to actively move a sensor within the environment to be able to keep tracking the pose of a given object (such is the case of objects that are being manipulated by humans in which the hands are creating significant occlusions and a static sensor constellation may not be able to observe enough surface geometry to be able to perform pose tracking with accuracy).

On the other 3 worlds, the main goal was similar but applied to bin picking operations. In this use case a static overhead camera can provide a rough estimation of the target objects and then based on the level of object recognition confidence and how significant are the occlusions, we may need to move a sensor attached to a robotic arm to several poses in order to gather further sensor data to increase the object recognition confidence and its pose estimation accuracy. In the first bin picking world, the starter motor was inside a large staking box together with an alternator and a differential gearbox (displayed in Figure 2). The second bin picking environment is a variation of the first in which it was added 3 more differential gearboxes into the stacking box in order to significantly increase the occlusion of the target object (seen in Figure 3). Finally, the last bin picking environment (seen in Figure 4) is another variation of the first environment in which it was added 3 more target objects (one on top of the trolley and two on the middle shelves).

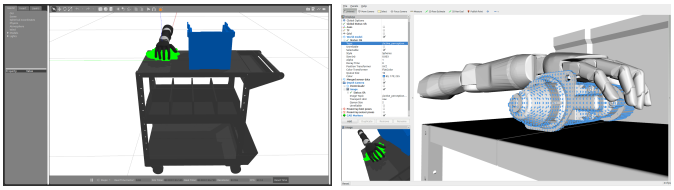

Figure 1. Environment for active perception of a starter motor being grasped by a human hand. The left image is showing a rendering from the Gazebo simulator with the target object in green color while the right image is displaying with blue spheres in Rviz the associated reference point cloud.

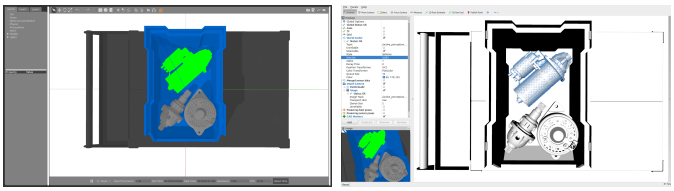

Figure 2. Environment for bin picking of one starter motor that is inside a large staking box together with an alternator and a differential gearbox.

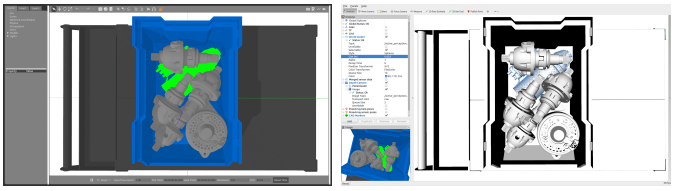

Figure 3. Environment for bin picking of one starter motor with 3 differential gearboxes causing occlusions.

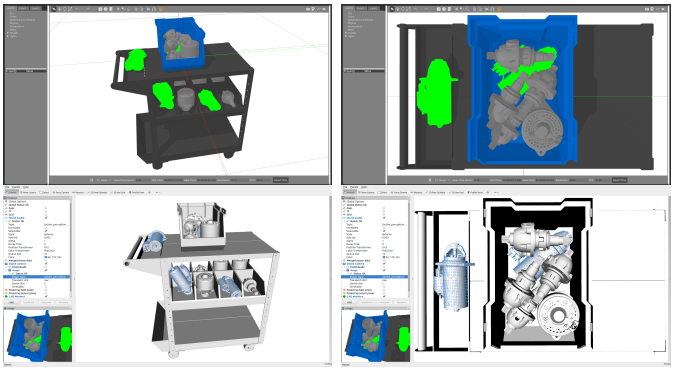

Figure 4. Environment for picking 4 starter motors with multiple occlusions.

B. Sensors modeling

Over the years it was developed a wide range of technologies for performing environment sensing. From the passive

[2]http://gazebosim.org

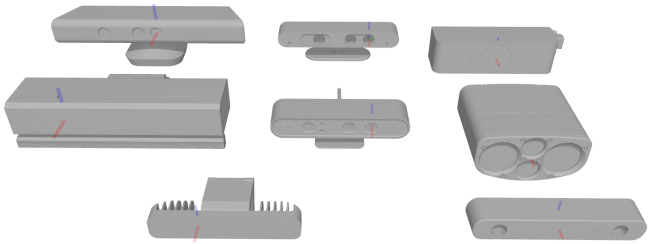

Figure 5. CAD models of the 3D sensors with the display of the depth image coordinate frames using the ROS convention of x-y-z -> forward-left-up. Name of the 3D sensors from top left to bottom right: Kinect XBox 360, Asus Xtion Pro Live, Ensenso N35, Kinect XBox One, Orbbec Astra, MultiSense S7, Intel RealSense SR300, ZED stereo camera.

image sensors to the active systems that probe the environment using projected patterns, lasers or Time of Flight (ToF) devices. Given that the goal of the proposed system was to perform active perception or environment monitoring, it was modeled 8 different types of depth sensors (shown in Figure 5) which relied on 3 types of environment sensing technologies. One of them was the Kinect XBox One ToF device, 5 were structured light sensors (such as the Asus Xtion Pro Live, the Ensenso N35, the Intel RealSense SR300, the Kinect XBox 360 and the Orbbec Astra) and 2 were stereo vision systems (namely the MultiSense S7 and the ZED stereo camera). Each of these depth sensors can be modeled using the pinhole camera model, which allows to specify the main unique characteristics of each sensor, such as the depth image resolution (width and height in pixels), its Field of View (FoV) (horizontal and vertical in radians) and the range in which the sensor can retrieve valid measurements (minimum and maximum in meters). Moreover, since this camera model is implemented in most 3D rendering engines and optimized in todays powerful Graphics Processing Units (GPUs), the sensor data generation can be performed very fast and efficiently using 3D rendering Application Programming Interfaces (APIs) such as the Open Graphics Library (OpenGL). This is the case of the Gazebo simulator, that uses the Ogre3D[3] rendering engine which in turn relies on OpenGL. Besides camera modeling, Gazebo also allows to simulate the sensor acquisition rate (specified as the number of depth images generated per second), which is usually higher on structured light sensors and lower on stereo vision systems.

C. Sensors deployment

Finding the optimal sensor constellation that maximizes the observed surface area of a given set of target objects is a challenging combinatorial explosive problem when considering the presence of occlusions within the environment. As such, for making the estimation of the sensor disposition computational feasible, the 3D continuous space was populated with a given set of sensors within regions of interest while looking at a specified observation point (with the sensor roll either 0º or random). This approach allows to reduce the sensor pose estimation from a 6 Degree of Freedom (DoF) to a 4 DoF problem (x, y, z position plus the sensor rotation along the observation axis). Moreover, the continuous solution space with an infinite amount of observation view points is reduced to a bounded number in which the sensors can either be deployed uniformly or randomly inside regions of interest. This allows to sample the solution space with a reasonable and representative amount of simulated sensor data for computing a good enough sensor disposition for the problem at hand.

The proposed system allows the deployment of several populations of sensors within a simulated world. Each population contains a given number of sensors of the same type that can be deployed uniformly / randomly within a box / cylinder or in a grid / linear disposition. This allows to deploy the sensors in the simulated environment spaces that represent valid positions for the real sensors. For example, limiting the possible sensor view points to the walls and ceiling of a room, avoiding deploying sensors in areas in which they could not provide any valuable sensor data or in which they could not be physically placed due to spatial restrictions or safety reasons.

The populations of sensors that were deployed on the 4 simulation worlds were fine tuned to the particular goals of each test. For the active perception environment, 450 sensors were deployed close to the target object, on the top, right and back side of the trolley (shown in Figure 6). This was done to simulate the closest range in which a dynamically moving sensor attached to a robotic arm could move (taking into consideration the human safety and the sensor minimum measurement distance, that was 0.2 meters).

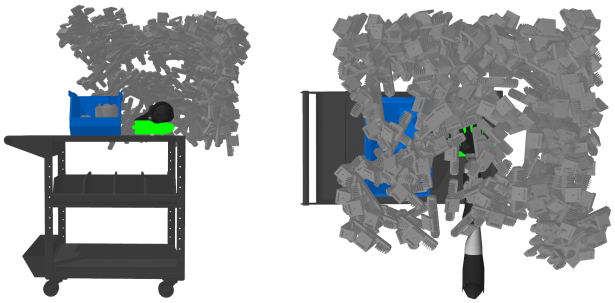

Figure 6. Sensors deployment for the active perception environment.

For the single object bin picking environments, given that the target object was inside the stacking box, the sensors were deployed close to the target object (displayed in Figure 7), but only on top of the trolley, on 3 layers (each with a different type of sensor). In the world with minimal occlusions it were deployed 100 sensors while in the world with significant occlusions it were deployed 300 sensors. The sensor density was increased because when there is a high amount of occlusions, the best views have tighter observation regions, which could be missed with a sparse sensor deployment.

For the multiple object bin picking environment, given that there were several target objects (1 inside the stacking box, 1 on top of the trolley and 2 on the shelves of the trolley), it were deployed 450 sensors across 7 populations (visualized in

[3]http://www.ogre3d.org

Figure 8), 5 of them simulating fixed sensors on the walls and ceiling and 2 of them simulating dynamic sensors attached to a robotic arm above the trolley.

It should be noted that the visual sensor disposition shown in Figures 6 to 8 was for human visual inspection only. During the sensor data generation, the 3D models of the sensors are hidden to avoid occlusion of the scene objects.

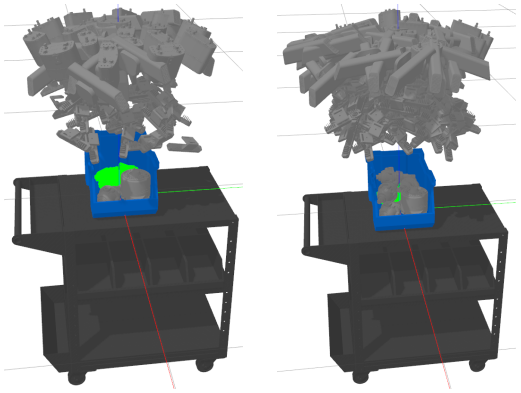

Figure 7. 2 sensors deployments for the 1 object bin picking environments.

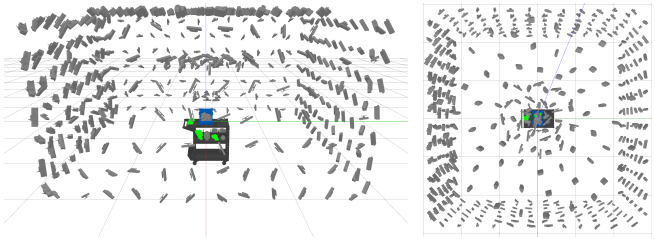

Figure 8. Sensors deployment for the multiple object bin picking environment.

III. Best views estimation

The estimation of the best views for a constellation of sensors requires the ability to generate accurate sensor data for each type of sensor and also an efficient approach to compute the surface coverage area (cost function) that we are trying to maximize. The next sections explain how the sensor data is analyzed and also present the approaches used to estimate the best constellation of sensors for a given simulation world.

A. Reference surface point cloud

The first step in the processing pipeline includes the generation of the multiple object reference point cloud that is built by transforming the point cloud associated with the target CAD model into each target object instance within the simulation world (example shown in Figure 9), followed by the merge of each object instance point cloud into a single point cloud in the world coordinate system, which later on is filtered with a voxel grid algorithm in order to perform a regular space partition and extract the surface voxels centroids that contain points. This approach allows to generate a reference point cloud with a constant surface point density, which will be critical later on when computing the surface coverage area percentage achieved with a given sensor constellation.

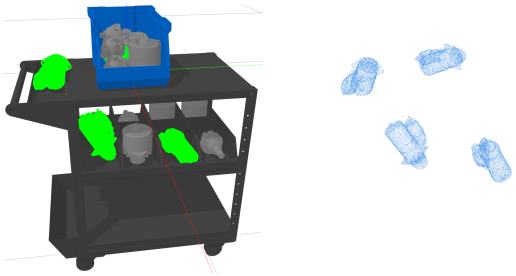

Figure 9. Scene rendering from the Gazebo simulator with the 4 target objects in green color (left image) along with the associated reference point cloud that was generated for the 4 target objects (right image).

B. Sensors data analysis

After loading the simulation world 3D models, deploying the sensors populations on the environment and building the filtered reference point cloud, the proposed system generates a color and depth image for every sensor. Then, for each pixel in the color images that have the target objects unique color (green), the corresponding pixel in the depth image is retrieved and using the pinhole model equations shown in Equation (1), the 3D point is computed from the 2D pixel coordinates and the depth value (retrieved form the OpenGL Z buffer). Later on, the 3D point is transformed from the sensor coordinate system into the world coordinate frame (having all sensor data in the same coordinate system allows fast merging of point clouds from several sensors).

After processing all pixels of a given color image, the associated point cloud in the world coordinate frame is filtered with a voxel grid algorithm with a cell size tuned for the objects geometry we are trying to observe (given that too many points on a small area of a large object do not provide a significant advantage for 3D perception and require more processing time). This allows to perform a regular space partition for extracting the centroid of each voxel containing sensor points. This step is critical for allowing consistent evaluation of the object(s) observed surface area percentage, given that sensors with different resolution or at varying distances may generate point clouds with different point density even when observing the same surface area. Moreover, given that both the reference point cloud and the sensor data point clouds were filtered in the same coordinate frame and with the same voxel grid cell size, the surface coverage percentage can be computed very efficiently by simply dividing the number of surface points in the filtered sensor data point cloud by the number of surface points in the filtered reference point cloud.

In the end of the sensor analysis stage (presented in Algorithm 1), each sensor is associated with a filtered point cloud in the world coordinate frame containing only points belonging to the target objects surface (example in Figure 10).

$$
\begin{aligned}
X &= \frac{(PixelColumn - XPrincipalPoint) \times PixelDepth}{XFocalLenght} \\
Y &= \frac{(PixelRow - YPrincipalPoint) \times PixelDepth}{YFocalLenght} \\
Z &= PixelDepth
\end{aligned}
\tag{1}
$$

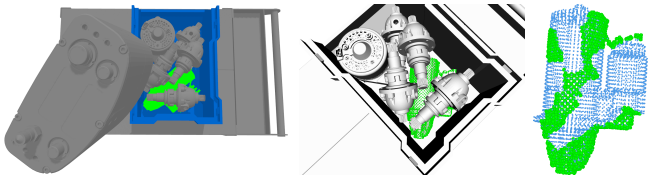

Figure 10. Environment for bin picking of 1 object with occlusions along with the selected 3D sensor (left image) and with the generated point cloud for the target object taking into consideration the environment occlusions (center and right images, in which the green spheres are the points observed by the 3D sensor and the blue spheres are from the filtered reference point cloud).

Algorithm 1 Sensor data analysis

Input:
$S \leftarrow$ deployed sensors
$C \leftarrow$ target objects unique color
$F \leftarrow$ cell size for the voxel grid filter
procedure SENSORANALYSIS(S, C, F)
  $q \leftarrow Empty$ ▷ sensors filtered point clouds
  for all s sensors in S **do**
    $c \leftarrow RenderColorImage(s)$
    $d \leftarrow RenderDepthImage(s)$
    $w \leftarrow GetSensorWorldPose(s)$
    $u \leftarrow Empty$ ▷ target objects points
    for all y image rows in c **do**
      for all x image columns in c **do**
        $p \leftarrow GetPixel(c, x, y)$
        if $p = C$ **then**
          $k \leftarrow GetDepth(d, x, y)$
          if $InValidRange(k, s)$ **then**
            $j \leftarrow 3DPoint(k, x, y, s)$
            $m \leftarrow TransformPt(j, w)$
            $AppendPoint(u, m)$
    $z \leftarrow FilterPointCloud(u, F)$
    $AppendPointCloud(q, z)$
  return (q)

C. Estimation of the best sensors views

When only one sensor is needed for the task at hand (for example when we are trying to perform 3D perception of the environment in which the sensor is attached to a robotic arm), the estimation of the best sensor can be performed by simply selecting the one that achieved the best surface coverage area percentage. On the other hand, if several sensors are available or we want a single sensor to observe the target objects from a set of N best views, then it is used a RANSAC approach to estimate the constellation of sensors that can achieve the best surface coverage. This approach allows to mitigate and bound the combinatorial explosion that happens when we need to estimate a high number of best views from a large population of sensors. As can be seen in Algorithm 2, this approach runs at most a fixed number of iterations. In each iteration, a set of N sensors are chosen randomly, their sensor data is merged and filtered, and if the surface coverage percentage achieved by this set of views is higher than a given threshold, then the search is stopped. In the end, it is returned the best sensor constellation found along with its associated point could (with the merged sensor data) and the best surface coverage percentage that was achieved.

Algorithm 2 Estimation of the best N sensors views

Input:
$N \leftarrow$ number of desired sensors
$P \leftarrow$ point clouds from each deployed sensor
$F \leftarrow$ cell size for the voxel grid filter
$C \leftarrow$ minimum surface coverage percentage
$I \leftarrow$ maximum number of iterations
procedure BESTSENSORSVIEWS(N, P, F, C, I)
  $s \leftarrow Empty$ ▷ best coverage sensors
  $p \leftarrow Empty$ ▷ best merged point cloud
  $c \leftarrow 0$ ▷ best coverage percentage
  $i \leftarrow 0$ ▷ current iteration
  while $i < I$ and $c < C$ **do**
    $x \leftarrow SelectSensorsRandomly(P, N)$
    $m \leftarrow MergePointClouds(P, x)$
    $f \leftarrow FilterPointCloud(m, F)$
    $k \leftarrow ComputeSurfaceCoverage(f)$
    $i \leftarrow i + 1$
    if $k > c$ **then**
      $s \leftarrow x$
      $p \leftarrow f$
      $c \leftarrow k$
  return (s, p, c)

IV. EXPERIMENTAL EVALUATION

Several tests were conducted in the simulated environments presented earlier for evaluating the ability of the proposed system to find suitable constellations of sensors for maximizing the observable surface area of a given set of target objects.

In the active perception environment introduced in Figure 1, it was performed two tests with the sensor deployment shown in Figure 6. In the first test it was estimated the best sensor pose for observing a single target object (green starter motor) being occluded by a human hand grasping it. By visually inspecting the scene in Figure 11, it can be seen that the system chose a very reasonable sensor pose, achieving a surface coverage of 27.73%, despite the heavy object occlusions introduced by the human hand. Moreover, when expanding the number of sensors to 3 (in the second test), the system managed to select a sensor constellation with a good spatial distribution (shown in Figure 12) that managed to improve the sensor coverage to 61.91%.

Moving to the single object bin picking environments, presented in Figures 2 and 3, it was made four more tests using the deployed sensors seen in Figure 7. In the first test it was estimated the best pose for a single sensor to observe the target object that was inside the stacking box, which had large occlusions on its surroundings, but could be clearly observed from above. As can be seen in Figure 13, the system choose a suitable observation sensor that managed to achieve

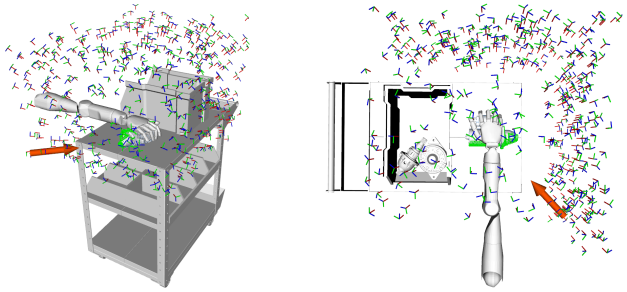

Figure 11. Estimation of the best sensor pose for the active perception environment with a 27.73% of surface area coverage (best sensor displayed as a large red arrow, while the deployed sensors are shown as small coordinate frames and the observed sensor data is represented with green spheres).

a surface coverage of 45.10%. When increasing the number of sensors to 5 (in the second test), the system relied on more sensor data and improved the surface coverage to 64.63% (as seen in Figure 14). To make the active perception for this bin picking use case more challenging, it was added three occluding differential gearboxes on top of the target object (scene shown in Figure 3) in order to create large occlusions that significantly reduced the number of useful sensors in the deployed populations (presented in the right image of Figure 7). Analyzing the best sensor pose estimated by the system (shown in Figures 10 and 15), it can be seen that the pose chosen was very reasonable, achieving a surface coverage of 19.27%. When increasing the number of sensors to 3, the system deployed a constellation with good spatial distribution and managed to improved the surface coverage percentage of the target object to 31.19% (as can be seen in Figure 16).

Increasing the level of complexity even further, in the final test it was added three more target objects to the simulation environment (as presented in Figure 4) and the number of populations with different sensor types was increased to 7 (shown in Figure 8). Analyzing Figure 17, in which the system estimated a constellation of 10 sensors to observe the 4 target objects, it can be seen that the system chose 4 sensors on the front wall (which had a better observation area for the target objects in the trolley shelves), 3 on the ceiling (for retrieving sensor data for the target objects on top of the trolley) and then for observing the remaining surface areas of the target objects, it chose one sensor on the left wall, another on the right wall and finally another one on the back wall, reaching 10 sensors in total and achieving a surface coverage of the target objects of 43.93%.

These 7 constellations of sensors computed using Algorithm 2 (which relied on a RANSAC approach), show that the proposed system can estimate a suitable sensor configuration for maximizing the observable surface area of several target objects even on complex environments with significant occlusions. Moreover, the system managed to compute useful solutions in bounded and reasonable time (from less than a second to a few minutes depending on the number and characteristics of the deployed sensors) for a problem that is combinatorial explosive in terms of processing time complexity.

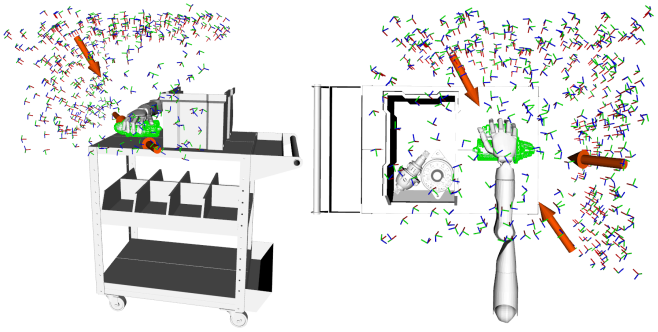

Figure 12. Estimation of the 3 best sensors disposition for the active perception environment with a 61.91% of surface area coverage.

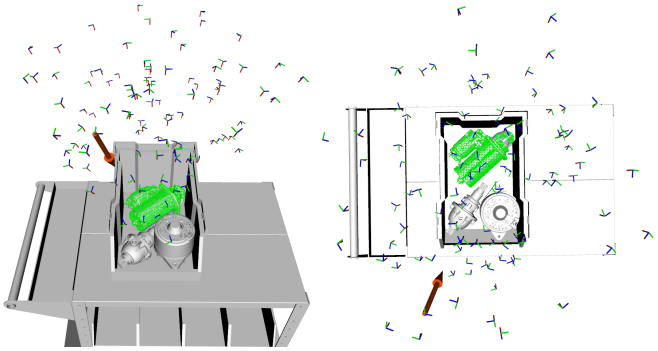

Figure 13. Estimation of the best sensor pose for the bin picking environment with a 45.10% of surface area coverage.

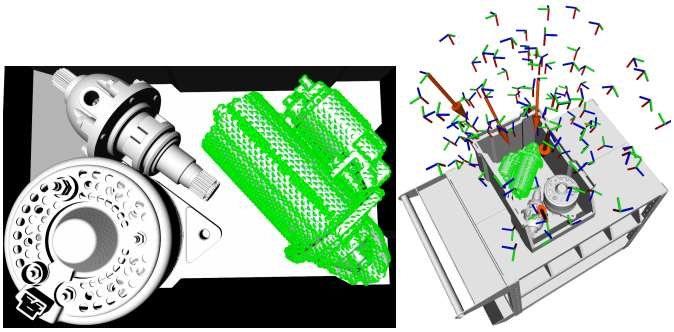

Figure 14. Estimation of the 5 best sensors disposition for the bin picking environment with a 64.63% of surface area coverage.

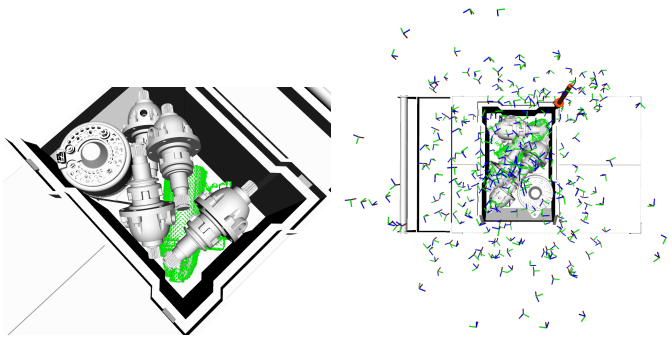

Figure 15. Estimation of the best sensor pose for the bin picking with occlusions environment with a 19.27% of surface area coverage.

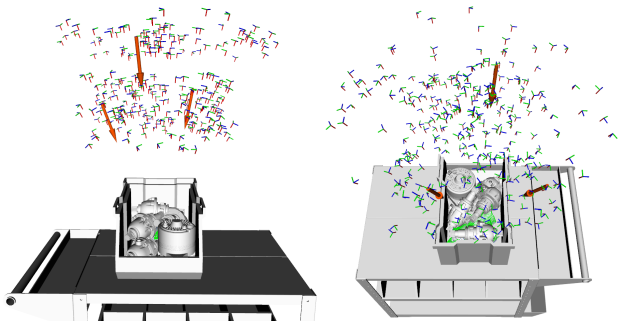

Figure 16. Estimation of the 3 best sensors disposition for the bin picking with occlusions environment with a 31.19% of surface area coverage.

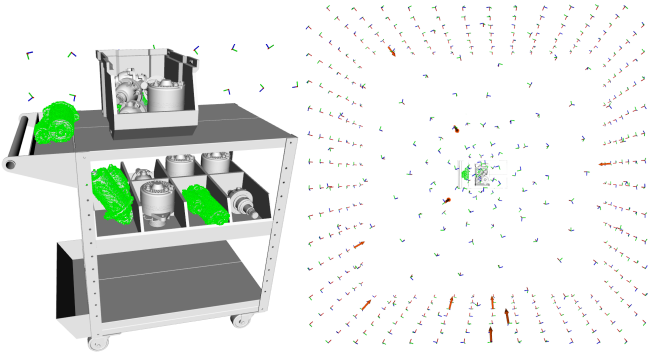

Figure 17. Estimation of the 10 best sensors disposition for the 4 object picking with occlusions environment with a 43.93% of surface area coverage.

V. Conclusions

The proposed sensor placement system was able to generate sensor constellations for maximizing the observable surface area percentage of a given set of target objects (starter motors) that were deployed on a trolley in several testing environments with increasing perception complexity and varying degree of occlusions. Each constellation had several types of depth cameras, that were modeling the main characteristics of eight 3D sensors, such as depth image resolution, field of view, range of valid measurements and data acquisition rate. For making this combinatorial explosive problem computational tractable, a random sample consensus algorithm was employed for determining which sensors should be selected from the populations of 4 DoF poses associated with each sensor type, that was either randomly or uniformly deployed over regions in which the real sensors could be placed and provide useful perception data. With a reasonable sensor count, the proposed sensor placement system managed to compute a good sensor pose in a few seconds and suitable sensor constellations in a few minutes, which makes it suitable for active 3D perception operations and sensor layout optimization tasks.

Future work could include the testing of the proposed approach in conjunction with a object recognition system in order to reliably perform object tracking when an operator is manipulating a target object (by moving the sensor within the environment using a robotic arm).

Acknowledgments

This work has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement number 101006798.

References

[1] R. Zeng, Y. Wen, W. Zhao, and Y.-J. Liu, "View planning in robot active vision: A survey of systems, algorithms, and applications," *Computational Visual Media*, vol. 6, no. 3, pp. 225–245, 2020.

[2] M. Mendoza, J. I. Vasquez-Gomez, H. Taud, L. E. Sucar, and C. Reta, "Supervised learning of the next-best-view for 3d object reconstruction," *Pattern Recognition Letters*, vol. 133, pp. 224–231, 2020.

[3] R. Eidenberger and J. Scharinger, "Active perception and scene modeling by planning with probabilistic 6d object poses," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2010, pp. 1036–1043.

[4] D. Stampfer, M. Lutz, and C. Schlegel, "Information driven sensor placement for robust active object recognition based on multiple views," in *IEEE International Conference on Technologies for Practical Robot Applications (TePRA)*, April 2012, pp. 133–138.

[5] N. Atanasov, B. Sankaran, J. L. Ny, G. J. Pappas, and K. Daniilidis, "Nonmyopic view planning for active object classification and pose estimation," *IEEE Transactions on Robotics*, vol. 30, no. 5, pp. 1078–1090, Oct 2014.

[6] D. Holz, M. Nieuwenhuisen, D. Droeschel, J. Stückler, A. Berner, J. Li, R. Klein, and S. Behnke, *Active Recognition and Manipulation for Mobile Robot Bin Picking*. Springer International Publishing, 2014, pp. 133–153.

[7] C. Potthast and G. S. Sukhatme, "A probabilistic framework for next best view estimation in a cluttered environment," *Journal of Visual Communication and Image Representation*, vol. 25, no. 1, pp. 148 – 164, 2014.

[8] A. D. Mezei and L. Tamas, "Active perception for object manipulation," in *IEEE 12th International Conference on Intelligent Computer Communication and Processing (ICCP)*, Sept 2016, pp. 269–274.

[9] M. Gschwandtner, R. Kwitt, A. Uhl, and W. Pree, "Blensor: Blender sensor simulation toolbox," in *Proceedings of the 7th International Conference on Advances in Visual Computing - Volume Part II*, ser. ISVC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 199–208.

[10] J. Hu and P. R. Pagilla, "View planning for object pose estimation using point clouds: An active robot perception approach," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9248–9255, 2022.

[11] A. Bircher, M. Kamel, K. Alexis, H. Oleynikova, and R. Siegwart, "Receding horizon "next-best-view" planner for 3d exploration," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2016, pp. 1462–1468.

[12] T. Gedicke, M. Günther, and J. Hertzberg, "FLAP for CAOS: Forward-looking active perception for clutter-aware object search," in *Proc. 9th IFAC Symposium on Intelligent Autonomous Vehicles (IAV)*. Leipzig, Germany: IFAC, Jun. 2016, pp. 114–119.

[13] M. Sheinin and Y. Y. Schechner, "The next best underwater view," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 3764–3773.

[14] M. Peuzin-Jubert, A. Polette, D. Nozais, J.-L. Mari, and J.-P. Pernot, "Survey on the view planning problem for reverse engineering and automated control applications," *Computer-Aided Design*, vol. 141, p. 103094, 2021.

[15] J. I. Vasquez-Gomez, L. E. Sucar, R. Murrieta-Cid, and E. Lopez-Damian, "Volumetric next-best-view planning for 3d object reconstruction with positioning error," *International Journal of Advanced Robotic Systems*, vol. 11, no. 10, p. 159, 2014.

[16] M. Krainin, B. Curless, and D. Fox, "Autonomous generation of complete 3d object models using next best view manipulation planning," in *IEEE International Conference on Robotics and Automation*, May 2011, pp. 5031–5037.

[17] Z. Guo, M. Zhou, and G. Jiang, "Adaptive sensor placement and boundary estimation for monitoring mass objects," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 38, no. 1, pp. 222–232, Feb 2008.

[18] S. Zhao, B. M. Chen, and T. H. Lee, "Optimal sensor placement for target localisation and tracking in 2d and 3d," *International Journal of Control*, vol. 86, no. 10, pp. 1687–1704, 2013.