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]

23th IEEE International Conference on
Autonomous Robot Systems and Competitions
(ICARSC 2023)

[1] Centre for Robotics in Industry and Intelligent Systems of INESC TEC, Portugal
[2] Faculty of Engineering of the University of Porto, Portugal
[3] Robotics and Human Machine Interaction Laboratory at the Technical University of Chemnitz, Germany

Presentation Outline

Context

- Perception of objects is a challenging task, especially when they have large sensing occlusions, because they make ineffective the traditional approach of capturing data in several views uniformly generated in a semi-sphere around the target objects due to the fact that the occlusions may result in views that gather very little surface data from the target objects.
- Moreover, the constellation of sensor views that maximizes the observable surface area of a set of target objects cannot be created by simply selecting the best N views, because they will likely have overlap in the surface areas that they observed from the target objects. As such, a smarter approach is required that takes into account the combined surface area coverage of the constellation.
- To further complicate the problem, the constellation of views will likely need to have sensors with different characteristics, because we may want to mount on the robotic arm compact sensors with low resolution and short range and attach outside the robot large sensors with high resolution and long range.

MAIN CONTRIBUTIONS

- Reliable segmentation of the target objects surface 3D points from the simulated worlds modeled in the Gazebo simulator using a special surface material that is not influenced by vertex shading and shadows and has a unique color to identify the target objects surface.
- Multisensor fusion using a voxel grid, which solves the problem of correctly computing the observed surface area coverage of the target objects when:
  - There is overlap between the point clouds captured from different views.
  - The sensors have different resolutions and are placed at varying distances.
- RANSAC approach for tackling the combinatorial explosive problem of estimating the type, number and disposition of 3D sensors that maximize the target objects observable surface area.
- Public release[1] of the Gazebo plugin code and dataset with 4 simulation worlds developed for bin picking and active perception use cases.

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

Depth sensors modeling

- Modeling in the Gazebo simulator of 3D sensors with different technical characteristics:
  - Resolution
  - Field of view
  - Minimum and maximum measurement range
  - Sensor acquisition rate
- Creation of simulation models for 3D sensors widely used from several manufacturers:
  - Ensenso
  - Kinect
  - Realsense
  - Orbbec
  - ZED

Environments modeling

- Modeling of 4 different test environments:
  - 1 for active perception with hand occlusions.
  - 1 for bin picking with minimal occlusions.
  - 1 for bin picking with large occlusions.
  - 1 for multiple object bin picking with large occlusions.
- Target object:
  - Starter motor
- Support and occluding objects:
  - Large stacking box
  - Trolley with shelves
  - Differential gearbox
  - Alternator
- The target object model uses a special surface material that ignores light effects (such as shading) and has a unique color (green) that will be used for sensor data segmentation.

ENVIRONMENT FOR ACTIVE PERCEPTION

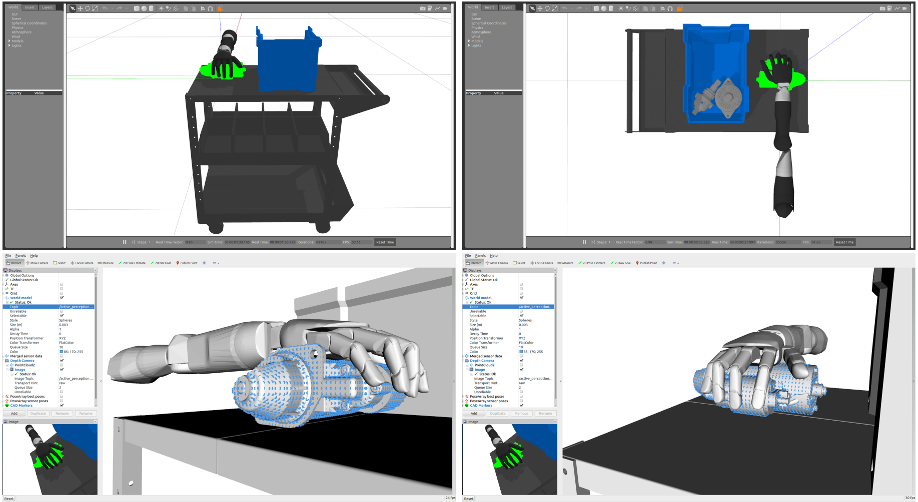

Fig. 1: Environment with a hand occluding a starter motor.

ENVIRONMENT FOR BIN PICKING

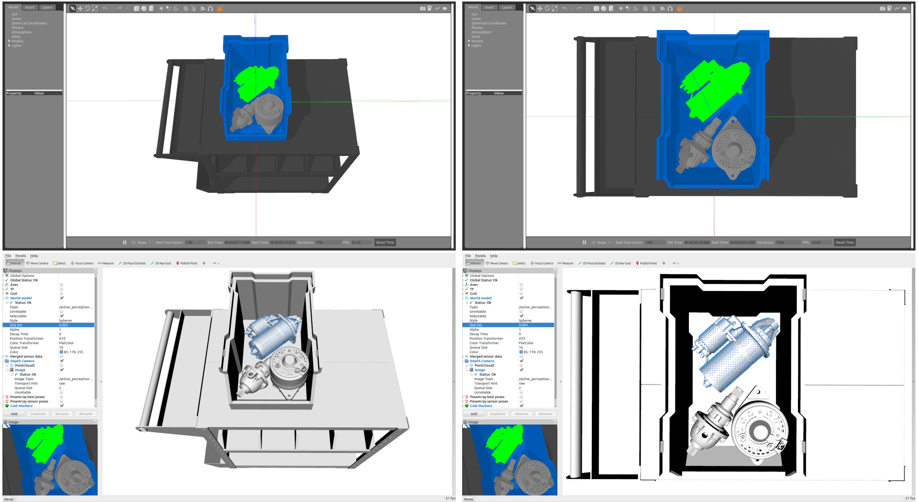

Fig. 2: Environment with a gearbox and an alternator occluding a starter motor.

Environment for bin picking with occlusions

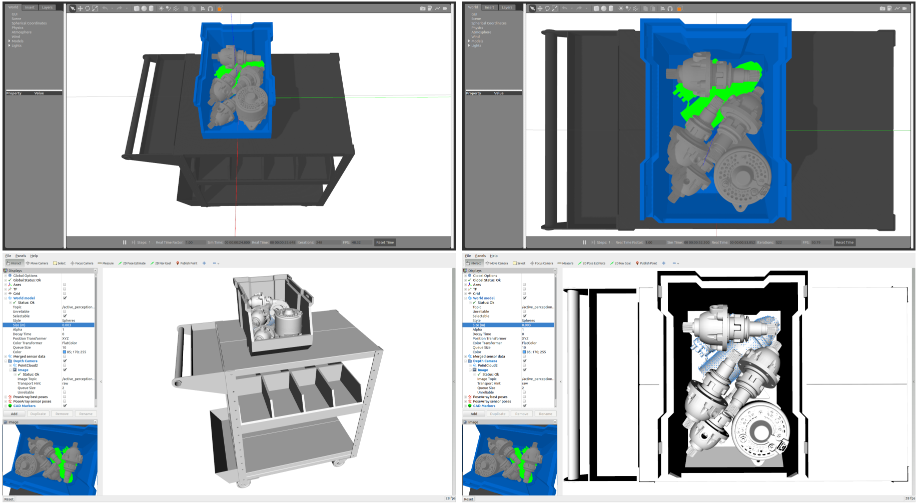

Fig. 3: Environment with gearboxes and an alternator occluding a starter motor.

ENVIRONMENT FOR MULTIPLE BIN PICKING WITH OCCLUSIONS

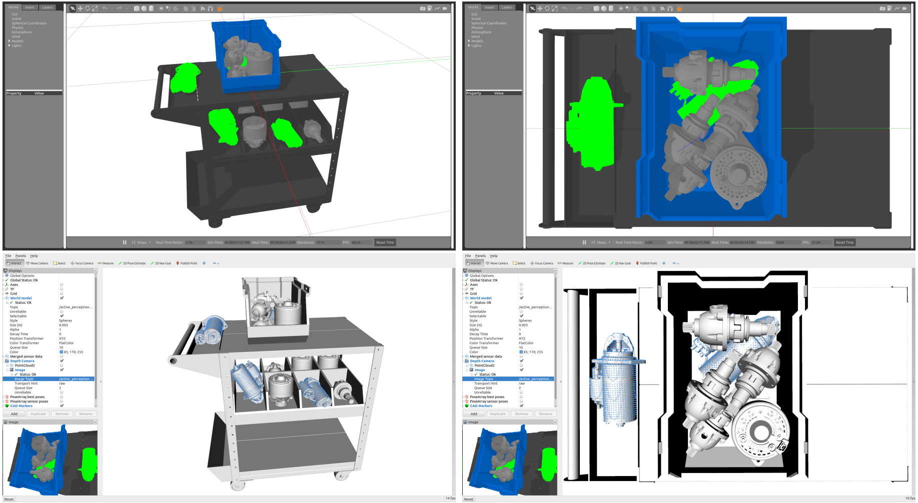

Fig. 4: Environment with gearboxes, an alternator and shelves occluding several starter motors.

Sensors deployment

- Several populations of sensors were added to each simulated world.
- Each population is of a given sensor type and is deployed within a given region of interest.
- Currently supported deployment configurations:
  - Uniform along a line.
  - Uniform within a 2D grid.
  - Uniform or random deployment within a box.
  - Uniform or random deployment within a cylinder.

SENSORS DEPLOYMENT FOR ACTIVE PERCEPTION

- For the active perception environment, 450 sensors were deployed close to the target object, on the top, right and back side of the trolley.

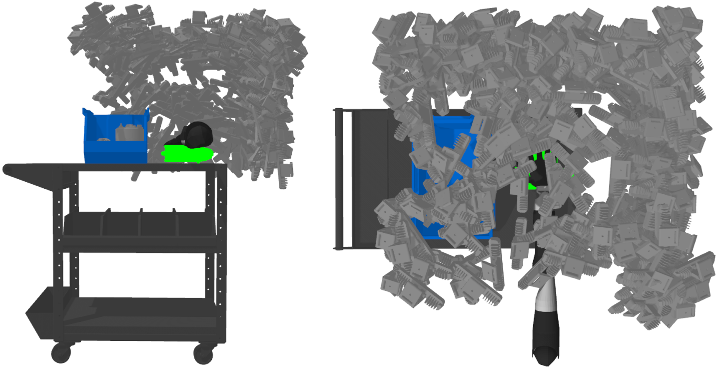

Fig. 5: Sensors deployment for the active perception environment (the CAD models of the sensors are hidden during the generation of the depth image).

SENSORS DEPLOYMENT FOR SINGLE OBJECT BIN PICKING

- In the world with minimal occlusions it was deployed 100 sensors while in the world with significant occlusions it was deployed 300 sensors.

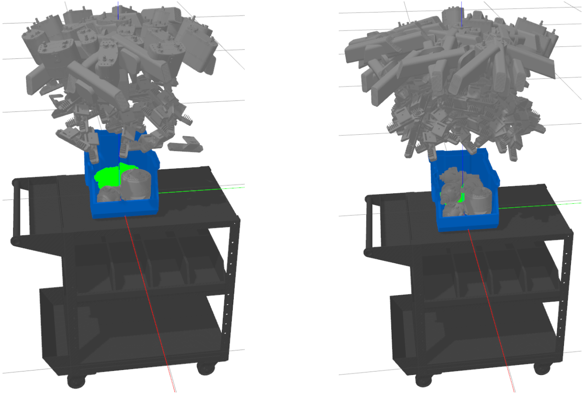

Fig. 6: Sensors deployment for the 2 bin picking environments that had a single target object.

Sensors deployment for multiple object bin picking

- For the multiple bin picking environment, given that there were multiple target objects it was deployed 450 sensors across 7 populations:
  - 5 populations simulating fixed sensors on the walls and ceiling.
  - 2 populations above the trolley, simulating dynamic sensors attached to a robotic arm.

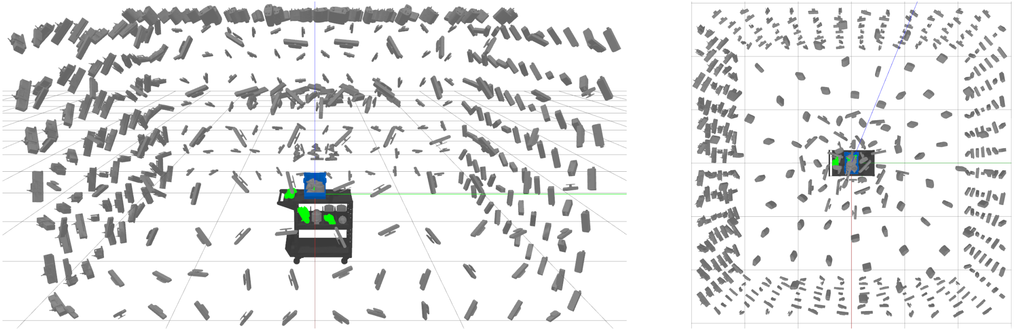

Fig. 7: Sensors deployment for the bin picking environment that had multiple target objects.

REFERENCE POINT CLOUD

- The first step in the processing pipeline includes the generation of the multi-object reference point cloud that is assembled using the CAD data and the objects poses given by the simulator, which is later on filtered with a voxel grid algorithm to perform a regular spatial partition and extract the points that are in the surface voxels centroids.

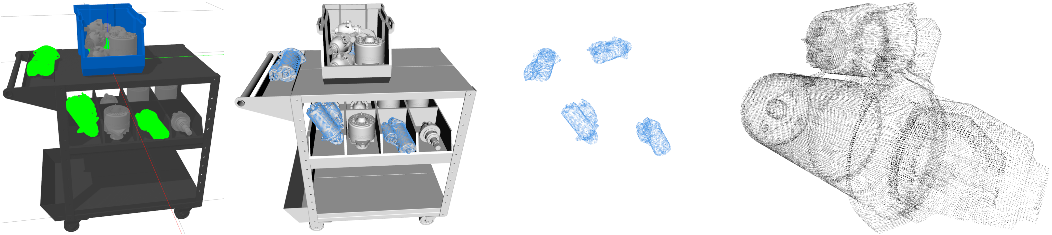

Fig. 8: The first image illustrates the color scene rendering in Gazebo with the target objects in green while the second and third images display the reference point cloud that was generated from the CAD points data shown on the last image.

Data analysis for each sensor

- Color segmentation is performed to identify the sensor image pixels that belong to the target objects (which have a unique green material that does not render shadows).
- For each image pixel associated with a target object, the 3D depth point is computed from the z-buffer depth image using the pinhole camera model.
- The generated point cloud is transformed from the sensor into the world coordinate system.
- A voxel grid filtering algorithm is applied to perform a regular space partition in which the centroid is computed for each voxel.
  - Critical for allowing consistent evaluation of the object(s) observed surface area coverage percentage, even when the sensors have overlap between their observed surfaces and also solves the problem of merging data from sensors with different resolutions that are at different distances from the target object(s).

Data analysis for each sensor

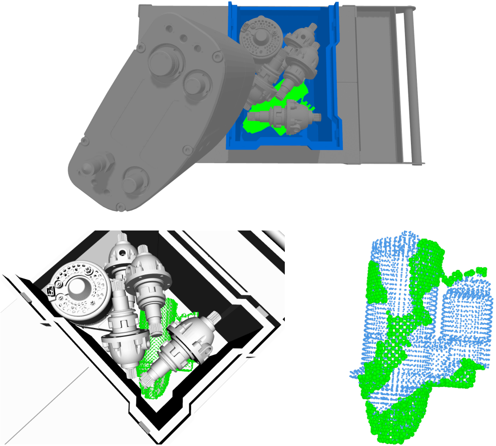

Fig. 9: Color image rendered with the Gazebo simulator along with the generated point cloud for the target object taking into consideration the environment occlusions.

Estimation of the best sensor

If only one sensor is enough (decision made by the system user):

- The surface area percentage for each sensor is analyzed.
- The sensor with the highest surface area percentage coverage of the target object(s) is chosen.

Estimation of the sensor constellation

- Using a Random Sample Consensus (RANSAC) approach, a set of N sensors is selected randomly.
- The sensor data from the selected sensors is merged using a voxel grid algorithm to ensure that there is only one point per voxel.
- The observable surface area percentage for the selected sensors is computed.
- If the current subset of sensors achieved better observable surface area percentage than the current best, then it becomes the current best views estimation for the sensor disposition.
- At the end of a given number of iterations or if the observable surface area percentage reaches a given threshold, the search is terminated, returning the best sensor constellation found.

Active perception environment - 1 sensor

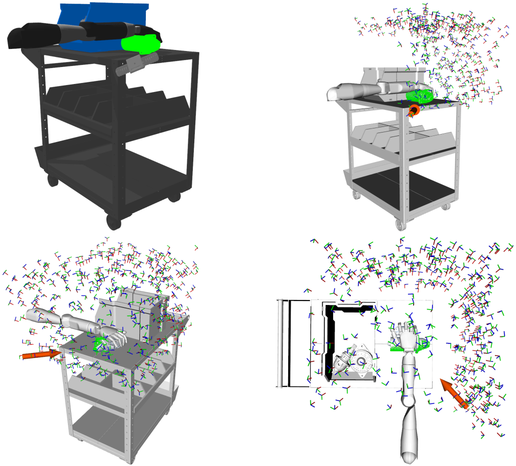

Fig. 10: Estimation of the best sensor for the active perception environment with a 27% of surface area coverage.

ACTIVE PERCEPTION ENVIRONMENT - 3 SENSORS

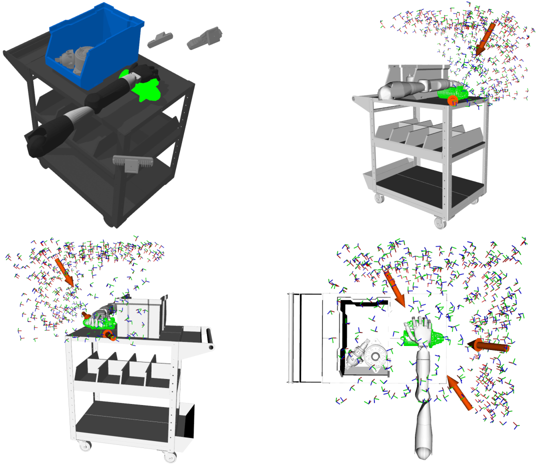

Fig. 11: Estimation of the 3 best sensors for the active perception environment with a 62% of surface area coverage.

Bin picking environment - 1 sensor

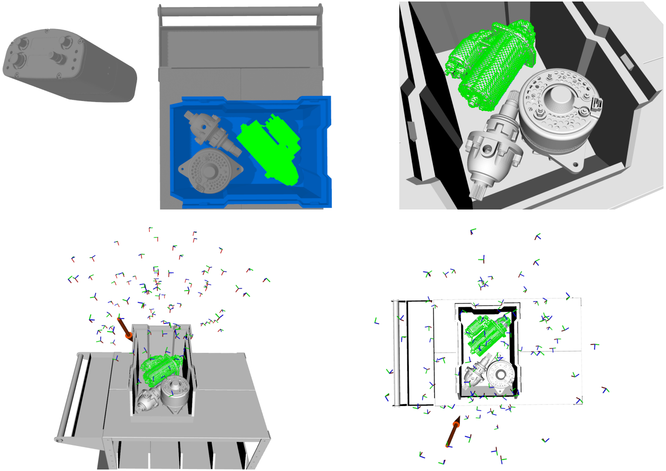

Fig. 12: Estimation of the best sensor for the bin picking environment with a 45% of surface area coverage.

Bin picking environment - 5 sensors

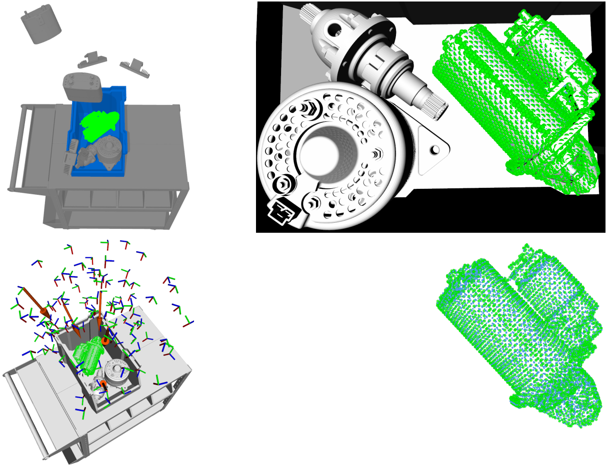

Fig. 13: Estimation of the 5 best sensors for the bin picking environment with a 65% of surface area coverage.

Bin picking with occlusions environment - 1 sensor

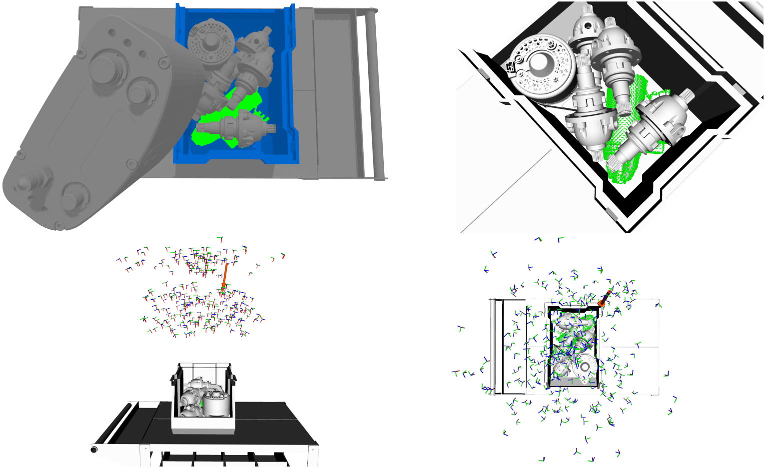

Fig. 14: Estimation of the best sensor for the bin picking with occlusions environment with a 19% of surface area coverage.

Bin picking with occlusions environment - 3 sensors

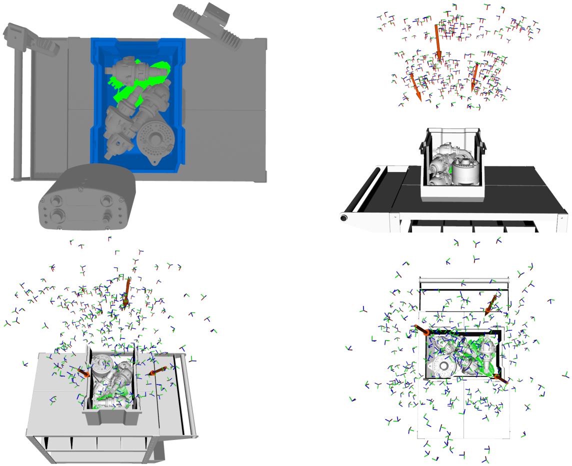

Fig. 15: Estimation of the 3 best sensors for the bin picking with occlusions environment with a 31% of surface area coverage.

Multiple bin picking with occlusions environment - 10 sensors

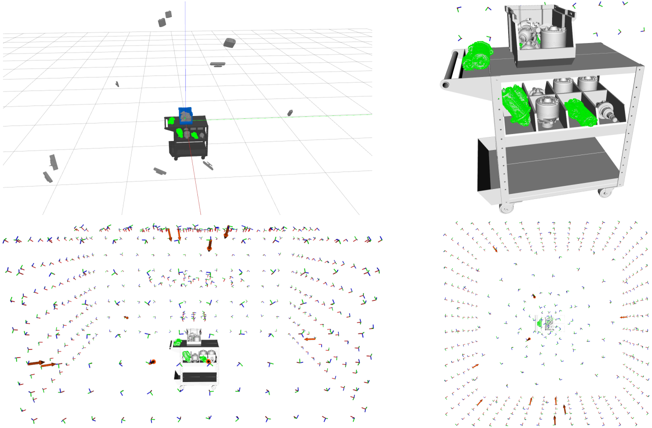

Fig. 16: Estimation of the 10 best sensors for the multiple bin picking with occlusions environment with a 44% of surface area coverage.

CONCLUSIONS

- The proposed system is able to estimate the N best sensors constellation for maximizing the observable surface area coverage for a given set of target objects.
- With a low sensor count the system can compute the best sensor constellation in less than a second, which makes it suitable for active perception tasks.
- Future work may 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).

Thank you!
Questions?