Best views estimation for active perception and sensor deployment

ProDEI - Advanced Methods of Modeling and Simulation

Carlos M. Costa
carlos.costa@fe.up.pt

Faculty of Engineering
University of Porto
Portugal

February, 2017

PRESENTATION OUTLINE

CONTEXT

- 3D object recognition is a challenging task that may require active perception of the environment for gathering additional sensor information when the level of confidence in the perception analysis is not enough
- Simulating sensor 3D depth data from a set of representative environments can help decide the type, number and disposition of sensors that maximize the target objects observable surface area

Research Areas

- 3D modeling and simulation of:
  - The 3D environment, including physics
  - The different types of depth sensors
- 3D rendering with occlusions
- Active perception
- Object recognition
- Bin picking

Software dependencies

- Robot Operating System (ROS)
  - For fast integration between simulation and real robots and 3D visual inspection (Rviz)
- Gazebo simulator
  - For world / sensor simulation and 3D rendering
- Point Cloud Library (PCL)
  - For point cloud processing

Fig. 1: Main software dependencies

Depth sensors modeling

- Modeling of 8 different types of depth sensors
  - Structured light sensors
    - Asus Xtion Pro Live
    - Ensenso N35
    - Intel RealSense SR300
    - Kinect XBox 360
    - Kinect XBox One
    - Orbbec Astra
  - Stereo sensors
    - MultiSense S7
    - ZED stereo camera
- Pinhole camera model coupled with OpenGL color rendering and depth buffer allows to specify the sensors:
  - Resolution (width and height in pixels)
  - Field of view (vertical and horizontal FOV in radians)
  - Minimum and maximum measurement range (near and far z-buffer clipping planes in meters)
  - Sensor acquisition rate (number of image acquisitions per second in hertz)

Depth sensors modeling

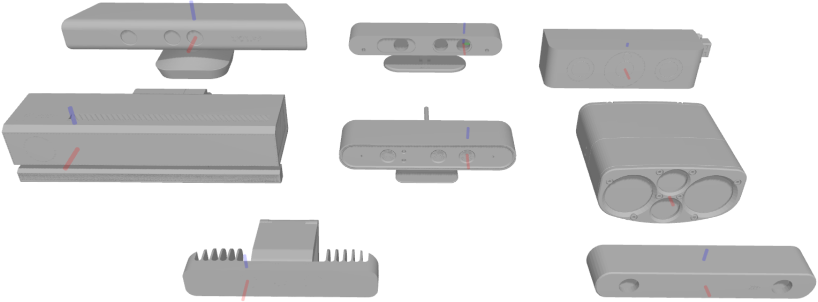

Fig. 2: Sensors 3D CAD models with the display of the depth image coordinate frames using the ROS convention of x-y-z -> forward-left-up (horizontally from top left to bottom right: Kinect XBox 360, Asus Xtion Pro Live, Ensenso N35, Kinect XBox One, Orbbec Astra, MultiSense S7, Intel RealSense SR300, ZED stereo camera)

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 significant occlusions
  - 1 for multiple object bin picking with several occlusions
- Modeling of 3 bin picking objects
  - Starter motor (target object for bin picking)
  - Alternator (for bin picking occlusions)
  - Differential gearbox (for bin picking occlusions)
- Modeling of 2 support objects
  - Trolley with shelves
  - Large stacking box
- Target objects use a special surface material that ignores light effects (such as light shading and shadows) and have a specific color (pure green) that will be used for sensor data segmentation. It was used .stl CAD models for 3D rendering and .ply point clouds for 3D data processing

Active perception environment

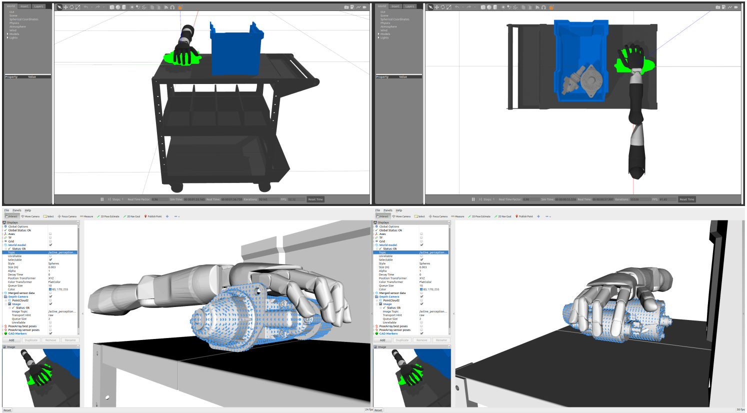

Fig. 3: Active perception environment renderings from Gazebo with target objects in green (top images) and associated CAD model point clouds displayed as blue spheres in Rviz (bottom images)

Bin picking environment

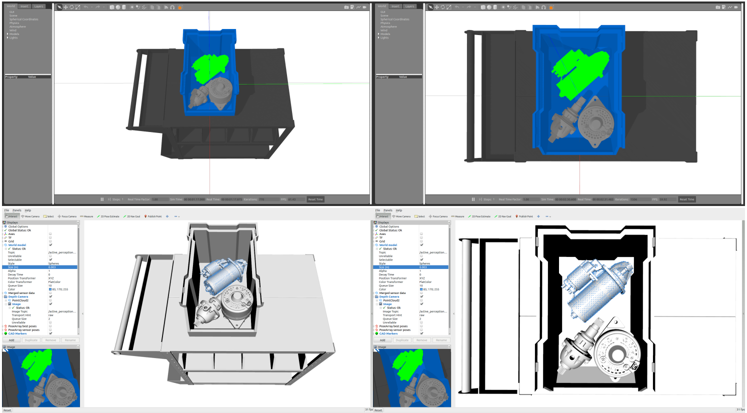

Fig. 4: Bin picking environment renderings from Gazebo with target objects in green (top images) and associated CAD model point clouds displayed as blue spheres in Rviz (bottom images)

Bin picking with occlusions environment

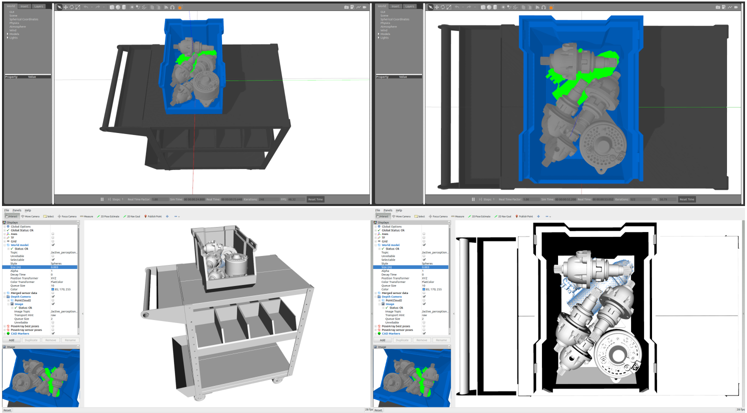

Fig. 5: Bin picking with occlusions environment renderings from Gazebo with target objects in green (top images) and associated CAD model point clouds displayed as blue spheres in Rviz (bottom images)

MULTIPLE BIN PICKING WITH OCCLUSIONS ENVIRONMENT

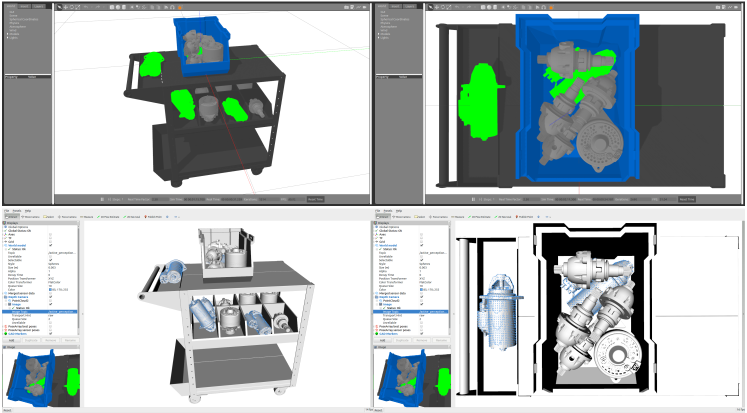

Fig. 6: Multiple bin picking with occlusions environment renderings from Gazebo with target objects in green (top images) and associated CAD model point clouds displayed as blue spheres in Rviz (bottom images)

Sensors deployment

- For making the estimation of the sensor disposition computational feasible, the 3D continuous space was populated with a given set of sensors that were looking at a given point (with the sensor roll either 0º or random)
- Several populations of sensors can be added to the world
  - The 3D sensor models will be hidden at rendering time to avoid occlusion of sensor data
- Each population is of a given sensor type and is deployed within a given region of interest
  - This allows to restrict the spatial distribution of the sensors, for example a given set of sensors should be in the walls or ceiling due to their weight, or they should be close to the target object given their limited depth measurements range
- Currently supported deployment configurations:
  - Uniform or random deployment within a box
  - Uniform or random deployment within a cylinder
  - Uniform within a 2D grid (with a set of rows and columns)
  - Uniform along a line

SENSORS DEPLOYMENT

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

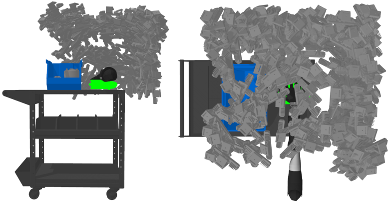

Fig. 7: Sensors deployment on the active perception environment

Sensors deployment

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

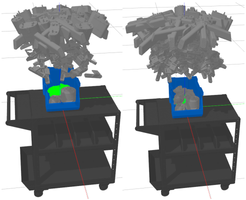

Fig. 8: Sensors deployment on the single bin picking environments

SENSORS DEPLOYMENT

- For the multiple bin picking environment, given that there were multiple target objects (1 inside the stacking box, 1 on top and 2 on the shelves of the trolley), 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

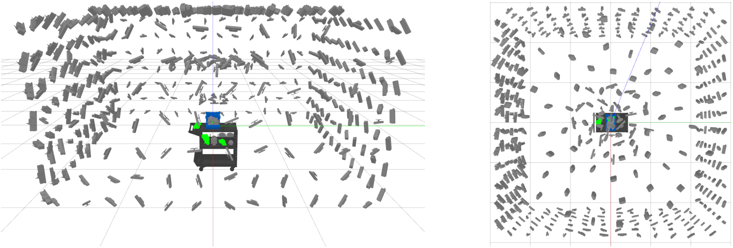

Fig. 9: Sensors deployment on the multiple bin picking environment

Reference surface 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

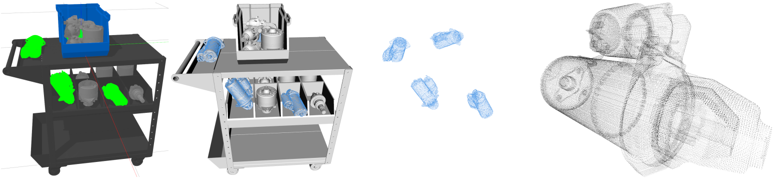

Fig. 10: The first image illustrates the color scene rendering in Gazebo with the target objects in green while the second and third images display the reference point cloud that was generated from the CAD points data shown on the last image

SENSORS DATA ANALYSIS

- Given a set of deployed sensors in the simulation world, for each sensor it is computed the voxelized point cloud of the observed target object(s) points:
  - Color segmentation is performed to identify the sensor image pixels that belong to the target object(s) (which have a unique pure green material)
  - 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 frame
    - Allows fast merging of point clouds generated from different sensors
  - A voxel grid filtering algorithm is applied to perform a regular space partition in which the points centroid are computed for each voxel
    - Critical for allowing consistent evaluation of the object(s) observed surface area coverage percentage, even when the sensors have different resolution and are at different distances from the target object(s)

Sensors data analysis

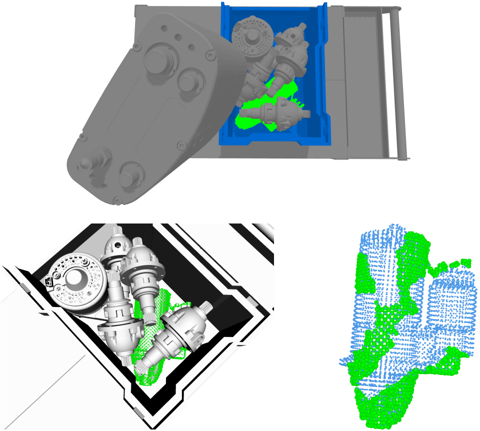

Fig. 11: Color image rendered with the Gazebo simulator (top image containing the scene and sensor) along with the generated point cloud for the target object taking into consideration the environment occlusions (bottom images, in which the green spheres are the observed points and the blue spheres are from the point cloud of the associated CAD model)

Estimation of the best sensor

After having the processed point cloud for each deployed sensor:

- If only one sensor is enough (decision made by the system user):
  - The surface coverage percentage for each sensor is computed
    - Given that both the reference point cloud and the sensor data were filtered with a voxel grid with the same resolution and in the same coordinate system frame, calculating the surface coverage percentage can be efficiently computed by simply dividing the number of surface voxel points in the sensor data by the number of surface voxel points in the reference point cloud
  - The sensor that can observe the most surface area percentage of the target object(s) is chosen

ESTIMATION OF THE BEST N SENSORS DISPOSITION

After having the processed point cloud for each deployed sensor:

- If several sensors can be used (decision made by the system user):
  - Using a Random Sample Consensus (RANSAC) approach, a set of N sensors is chosen randomly
  - The sensor data from the selected sensors is merged
  - The voxel grid filter algorithm is applied 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 disposition found

ACTIVE PERCEPTION ENVIRONMENT - 1 SENSOR

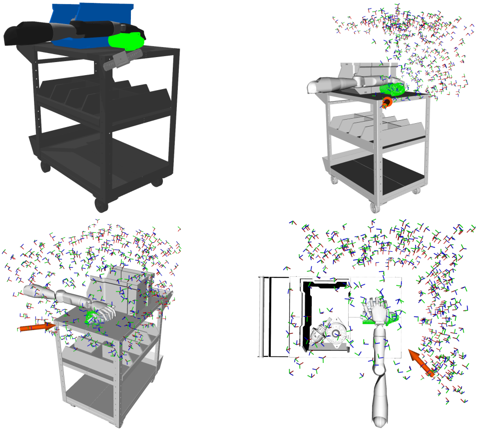

Fig. 12: Estimation of the best sensor position for the active perception environment with a 27.73% of surface area coverage (top left showing the Gazebo color rendering and remaining images displaying the best sensor as a large red arrow, the deployed sensors as small coordinate frames and the observed sensor data as green spheres)

Active perception environment - 3 sensors

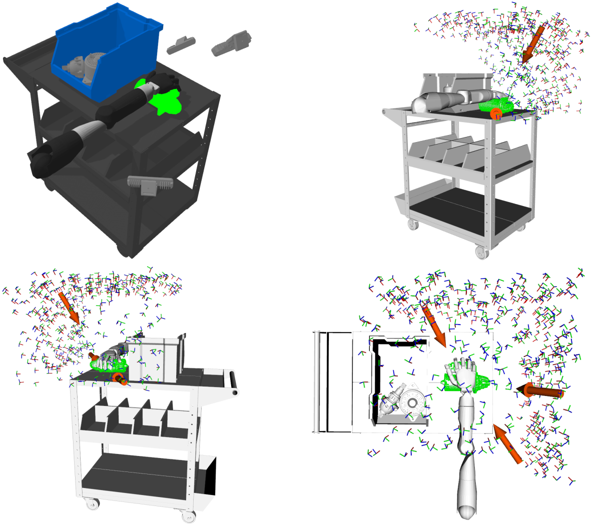

Fig. 13: Estimation of the 3 best sensors disposition for the active perception environment with a 61.91% of surface area coverage

Bin picking environment - 1 sensor

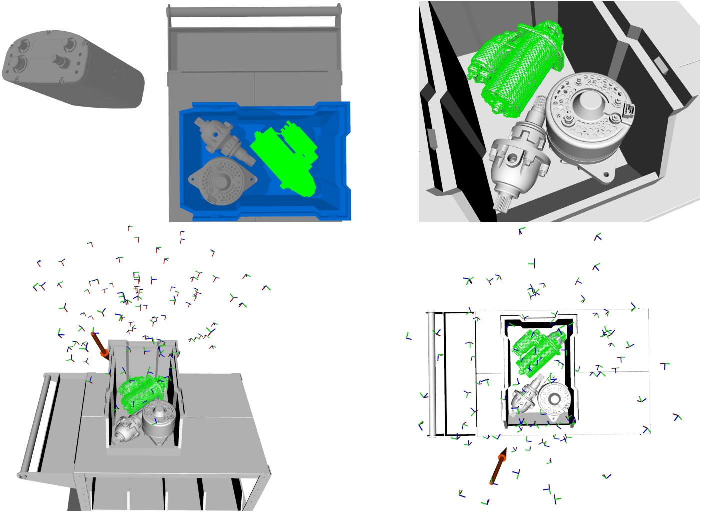

Fig. 14: Estimation of the best sensor position for the bin picking environment with a 45.10% of surface area coverage

BIN PICKING ENVIRONMENT - 5 SENSORS

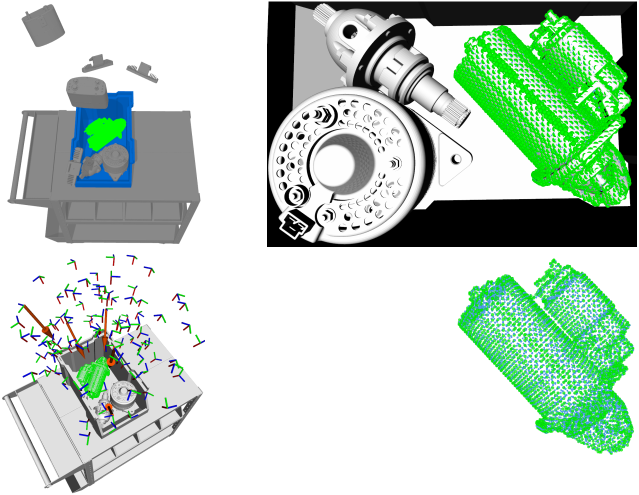

Fig. 15: Estimation of the 5 best sensors disposition for the bin picking environment with a 64.63% of surface area coverage

Bin picking with occlusions environment - 1 sensor

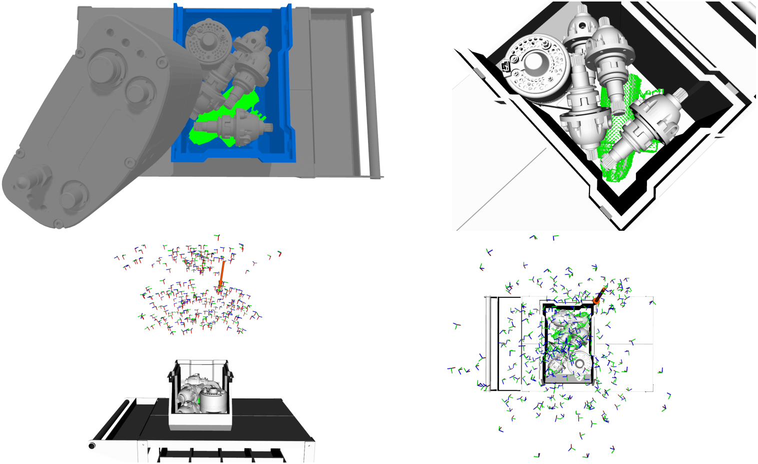

Fig. 16: Estimation of the best sensor position for the bin picking with occlusions environment with a 19.27% of surface area coverage

BIN PICKING WITH OCCLUSIONS ENVIRONMENT - 3 SENSORS

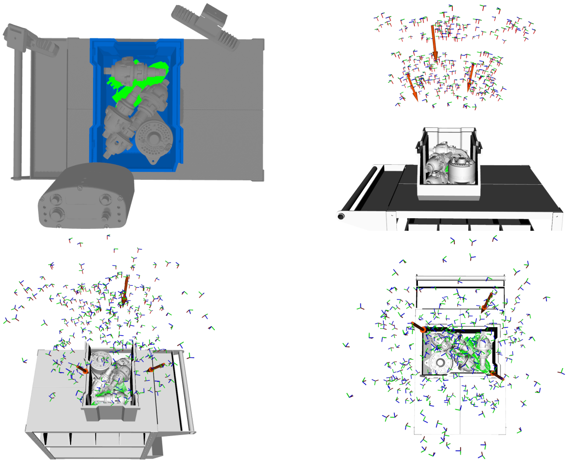

Fig. 17: Estimation of the 3 best sensors disposition for the bin picking with occlusions environment with a 31.19% of surface area coverage

Multiple bin picking with occlusions environment - 10 sensors

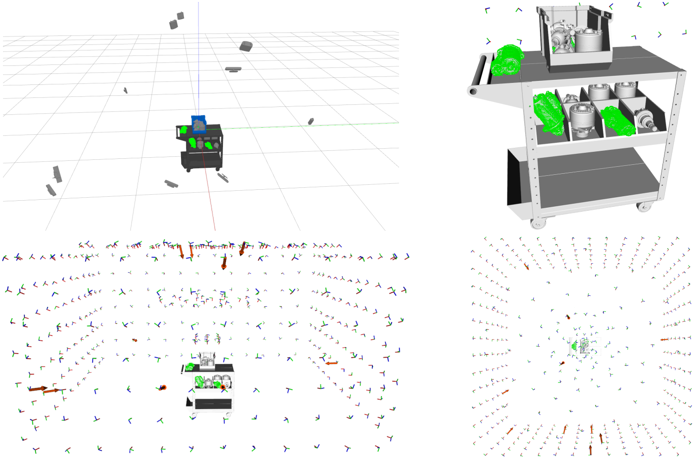

Fig. 18: Estimation of the 10 best sensors disposition for the multiple bin picking with occlusions environment with a 43.93% of surface area coverage

Conclusions

- The proposed system is able to estimate the N best sensors disposition for maximizing the observable surface coverage area percentage for a given set of target objects
- With a low sensor count the system can compute the best sensor disposition in less than a second, which makes it suitable for active perception tasks
- Future work would 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)
- Further research for fast object tracking recovery could include the modeling of the interaction between the target objects and a simulated hand in order to keep an approximate object pose even if it is almost completely occluded

Thank you!
Questions?