RBE 595 Course Project Final Report:

Atlas' Escape

Tsung-Chi Lin, Shang Gao, Chaitanya Perugu, Sanjuksha Nirgude

Introduction

Humanoid robots are designed to work in environments suitable for humans, which include indoor environments. Doors are a very common part of most human environments. In this project we are performing a task in which the humanoid robot Atlas finds a door in his environment, approaches the door and opens it using his arms. This task consists of sub-tasks like door-detection, localization, navigation towards the door and pushing open the door. The three fields of robotics covered in this project include loco-motion as we perform walking with the Atlas robot, perception for the tasks of door detection and localization and manipulation as we are using Atlas's arms to open the door. The robot performs all these tasks fully autonomously. The main motivation behind this project was to perform tasks in human environment as it the ultimate goal of humanoid robotics. Also, performing an in-door searching task for the door and also navigation of the humanoid robot in a human environment is another motivation.

System Framework

In order to achieve escape purpose, we decompose the task into three parts including searching the door, walking to the desired position where in front of the door, and open the door then go through it. Each of them stands for particular motion which are perception, loco-motion, and manipulation. For perception section, we extract the information from the camera to find the door in 3D space and combining the LiDAR feedback to calculate the depth then generate the desired pose for Atlas robot including the goal position and an appropriate orientation. After we defined the objective for Atlas, the walking motion executed by developed walk controller and walk to goal function. Finally, when Atlas already stand in front of the door, the arm controller involved performing the pre-defined arm motion with certain arm pose combination. The building blocks described above are represented graphically in Figure 1.

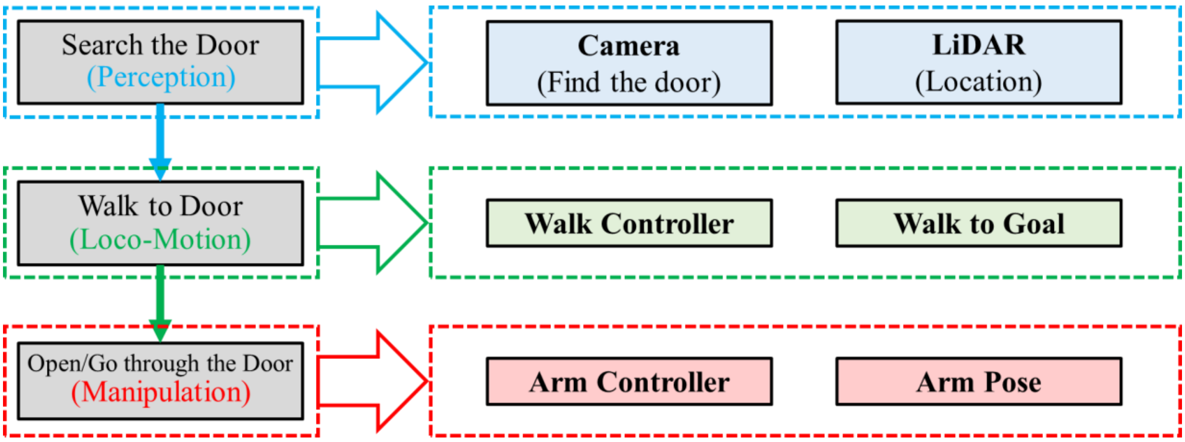

Figure 1. Building Blocks for Escaping.

Perception

The computer vision techniques have been known as an intuitive and efficient approach for sensing the environment. Therefore, we combined the information from the camera and LiDAR to generate the desired position which as an input to navigate the robot to the desired position.

Camera Information

The process for using camera information including two steps, find the door and view adjustment. For the door searching part, if there was no door in the view, we let the Atlas turn the whole-body by 45 degrees in the clockwise direction (Figure 2) for each search using the function of RobotWalker::walkRotate(). The searching section stops until the door in the view but not strict to where is it. Then we extracted the BGR image from camera and converted it into HSV format in order to get the red pixels and calculate the centroid of the door.

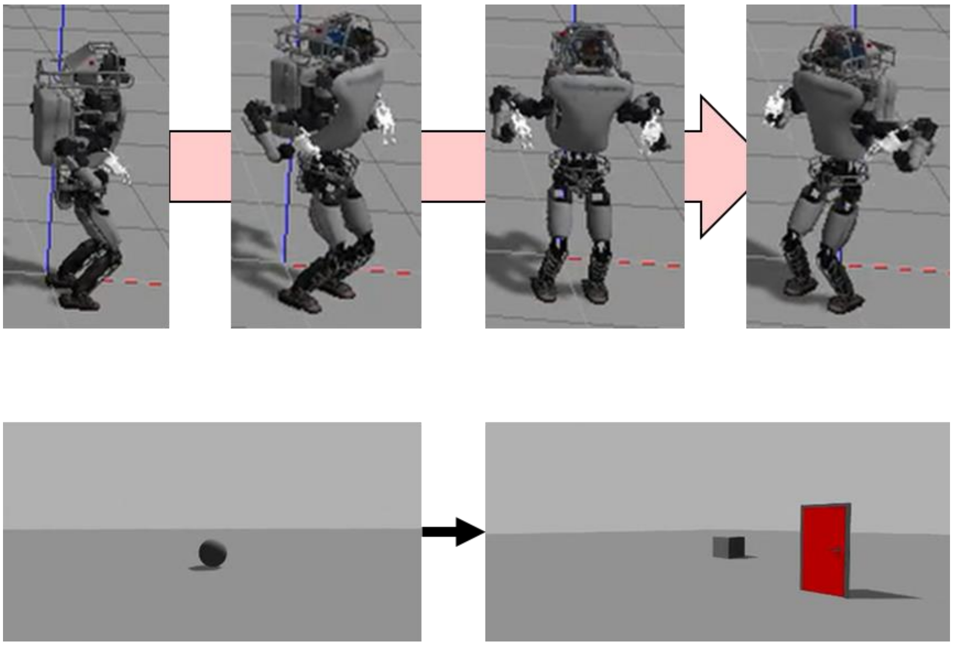

Figure 2. Turn the whole-body to find the red door.

After the door located in the view, we further move the chest to let the calculated centroid of the door in the range of pre-defined centroid width which is eight percent of the entire width of the view (Figure 3). The sector of the chest turning keeps the constant of five degrees. The end of view adjustment process will let the Atlas face directly to the door.

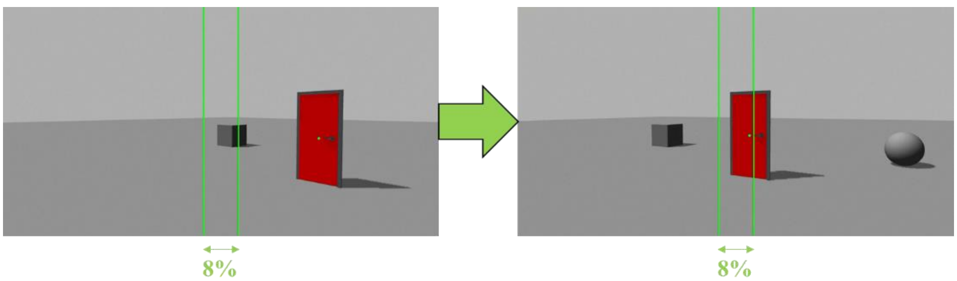

Figure 3. Turn the chest to let the door in the centroid region.

LiDAR Point Cloud

Locating the door in 3D space calls for the use of some sort of a depth measurement. We chose the LiDAR point cloud generated by a rotating laser sensor atop the head. Our other option was the stereo depth estimation, but we rejected it because we were not getting good depth estimates due to the door and most of the environment being largely featureless. To get the point cloud data, we subscribe to the /atlas/assembled_cloud2 topic though the MultisensePointCloud class.

The point clouds generated by ATLAS while mapping the world were over 2 million points in size. That is a huge amount of data to reliably process. Also, it is extremely difficult to differentiate between a door and a plain wall in point clouds. This is where the aforementioned camera exercise comes into play. It ensures that the head of ATLAS, which houses all the sensor equipment, is facing straight at the door. This helps us narrow our point cloud to only the relevant door region, eliminating walls and other obstacles from the point cloud.

To achieve this, we use a technique called Frustum Culling which essentially only retains points that lie in the field of view of an arbitrarily defined camera. The PCL library provides the class FrustumCulling that implements this and returns the filtered cloud. We define a hypothetical camera with a very narrow horizontal FOV located at the same position and orientation as the head of ATLAS and perform frustum culling. The result is a smaller point cloud that only contains a sliver of the original point cloud, free of all objects but the door and some ground points. We can filter the ground points using a z-axis threshold of 0.1 m to leave only the door point cloud behind (Figure 4).

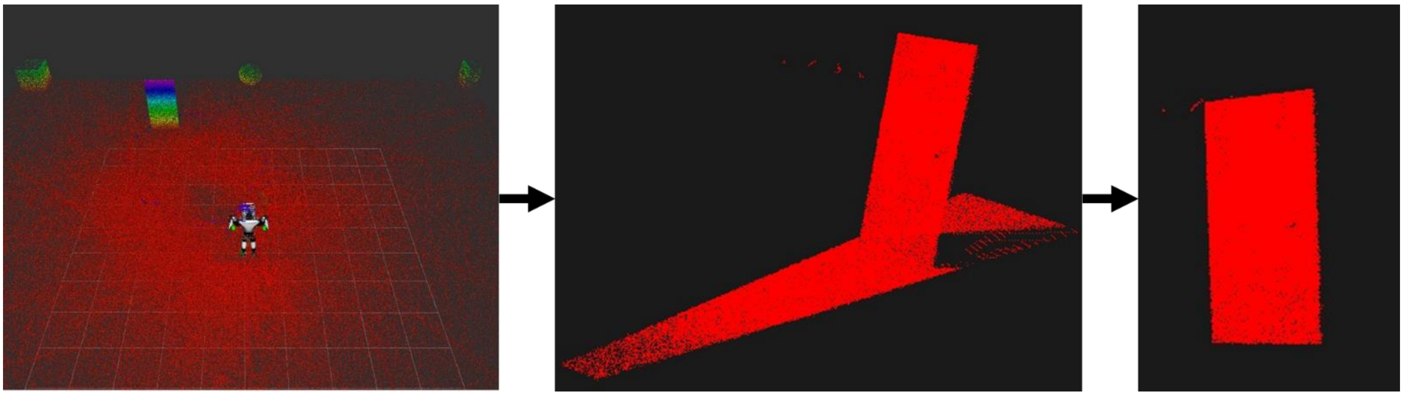

Figure 4. Process to get the door point cloud.

Once we have the filtered door point cloud, we can compute the centroid of the point cloud to locate the door in 3D space. The centroid tells us more than we need though, because the height of the centroid of the ground is irrelevant to navigating to the door. The x and y components of the centroid locate the door in the environment. But this does not entirely describe the door, because it does not tell us anything about how the door is oriented. To properly interact with the door, we need to approach it in a perpendicular direction.

To find the orientation of the door, we use the NormalEstimation class from the PCL library to compute the normal direction of the point cloud at each point. The way this works is by finding the best-fit plane to a few nearest-neighbors and finding the normal. We can aggregate this point cloud of normals by calculating the centroid of these normals. This gives the overall mean normal of the door, the perpendicular direction towards the world origin. We now have the position and orientation of the door in the world frame.

Loco-Motion

From previous color segmentation and point cloud extraction, the XY coordinates of the centroid on the door is acquired with respect to robot's local frame. To let the robot opening the door, Altas must be standing in front the door first, which position and orientation is shown in Figure 5. The robot needs to be standing at the point 1.25 away from the center of the door, with the opposite orientation from the door.

After the goal position and orientation is found, RobotWalker::WalkToGoal function is called, with world frame X, Y coordinates and pose input. The Atlas foot step planner will be called and a smooth and feasible trajectory is planned (Figure 6).

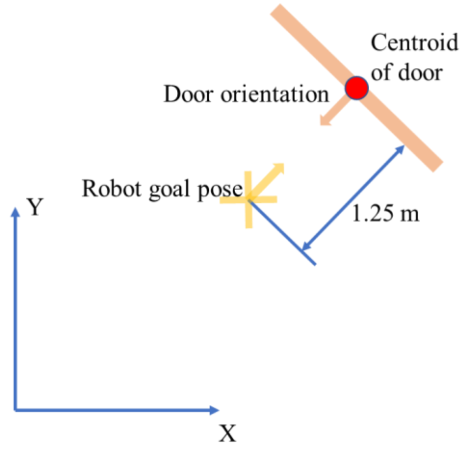

Figure 5. Calculate Goal Point

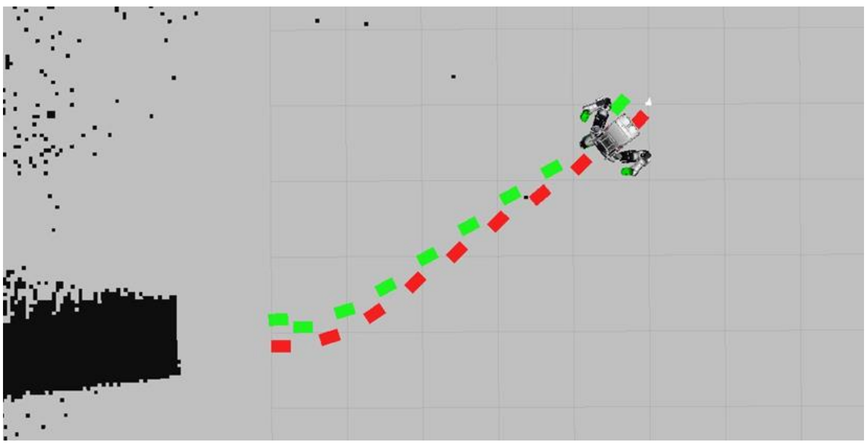

Figure 6. Calculated Foot Steps

Manipulation

When atlas is standing in front the door, certain arm motion is required so that it could walk pass the door. During the walking, the robot arm is in default configuration. Figure 7 shows below, the robot is wider than the door at this point. Also, it is needed to pose the arm toward front so that the arm could contact the door when pushing, instead of robot body. The arm motion is achieved by calling ArmControlInterface::moveArmJoints(). However, an intermediate arm position is needed between arm's default configuration and the final forward posture, which is lifted the arm upward. By doing this, it could be prevented that robot hugs the door and hits the door frame, further stacking and falling over. Finally, the arm is set as the "zombie" configuration, and RobotWalker::walkNStepsWrtPelvis() function is called to push Atlas walk 1 step forward (Figure 8) and the door is opened now.

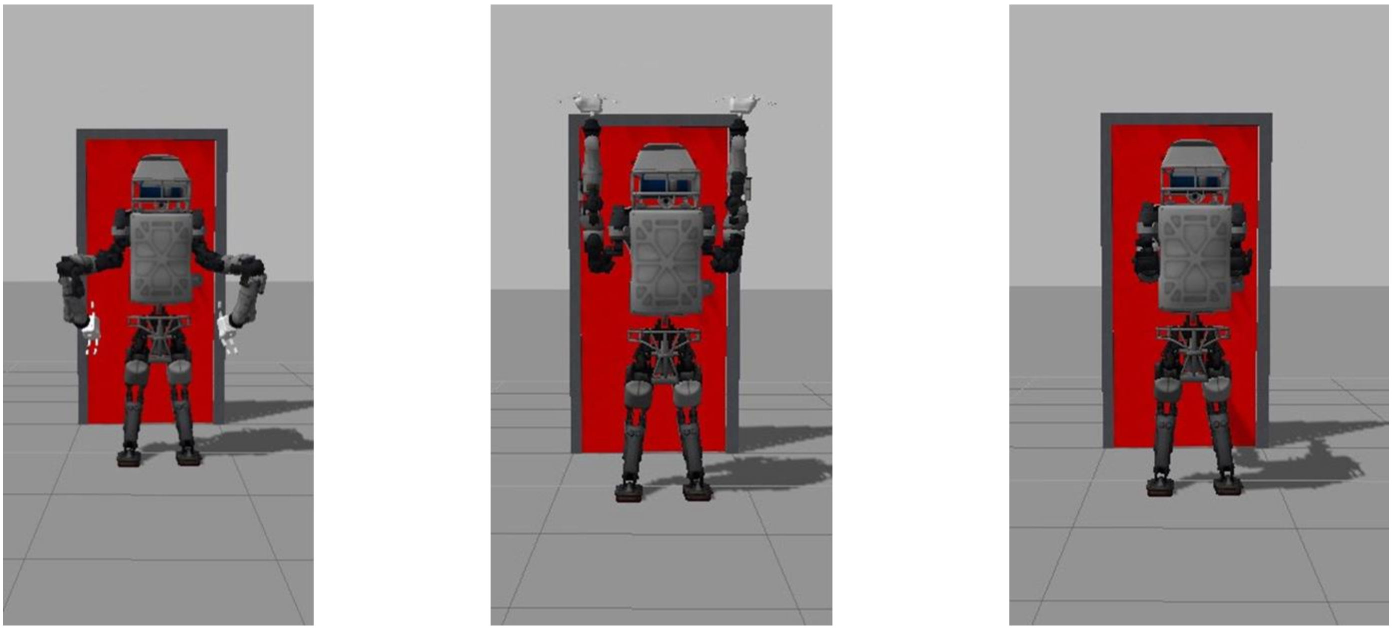

Figure 7. Three sequence of arm pose.

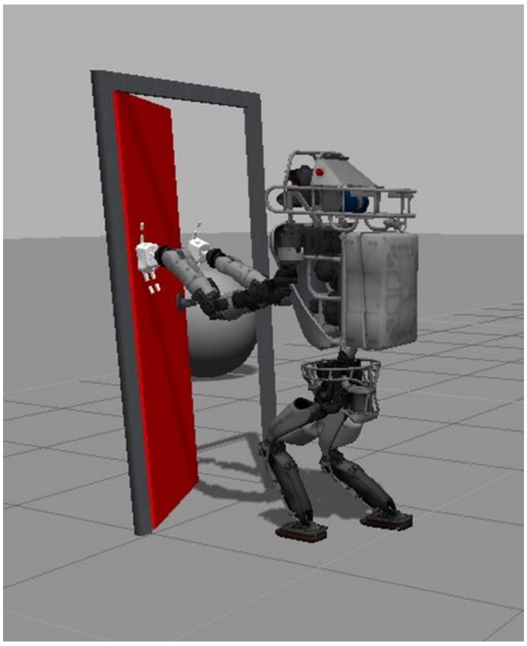

Figure 8. Open the door.

Conclusion

In this project we fulfilled our ultimate goal for performing a task in which the humanoid robot Atlas finds a door in his environment, approaches the door and opens it using his arms by combining perception, loco-motion and manipulation. Also, the robot performs all these tasks fully autonomously and the total lines of the code goes to ###.