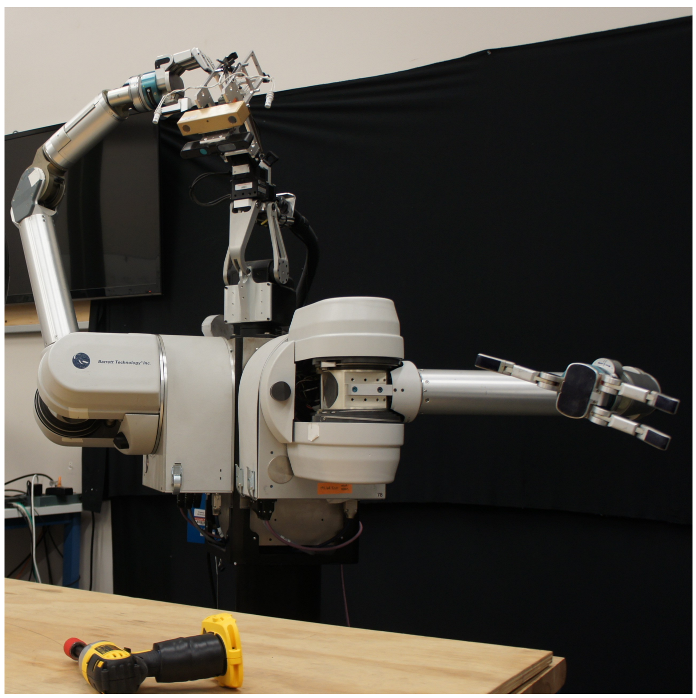
Andy Project

Building Robots that work with people

Shiyu Dong, Bikramjot Hanzra, William Seto, Tushar Chugh, Tae-Hyung Kim
Mentors – Jean Oh, Katharina Muelling
The Robotics Institute
Carnegie Mellon University



Even ANDY gets confused !!

Abstract

Robots are increasingly becoming key players in human-robot teams. To become effective teammates, robots must possess profound understanding of an environment, be able to reason about the desired commands and goals within a specific context, and be able to communicate with human teammates in a clear and natural way. To address these challenges, NREC is developing an intelligence architecture that combines cognitive components to carry out high-level cognitive tasks, semantic perception to label regions in the world, and a natural language component to reason about the command and its relationship to the objects in the world. As the part of this project, our team had to extend this existing frame to robots having capabilities of manipulation.

Contents

1 Requirements and Results

In the table below, the requirements that were submitted at the start of the semester and our performance on each are discussed below –

Mandatory Functional Requirements

Requirement 1:	The robot is able to detect certain kinds of blocks on the table surface. During this project, the robot shall be able to distinguish between blocks of at least four different colors namely pink, blue, purple and green.
Result:	SUCCESS – All 4 colors were getting detected with an accuracy with 100%.
Requirement 2:	Extend the communication between the robot operating control software (OCS) and the language grounding unit to include the color information of the block.
Result:	SUCCESS – We also included the orientation of the blocks which was helpful in spatial reasoning of the language part.
Requirement 3:	Identify manipulation actions needed to build geometrical shapes like a triangle and a square using the blocks provided.
Result:	SUCCESS – Rotate by direction, rotate by degrees, move by distance in centimeters, place on orientation, put down.
Requirement 4:	Implement the missing manipulation actions identified that are needed to build the above mentioned geometrical shapes.
Result:	SUCCESS – Implemented the actions mentioned in requirement 3.

Mandatory Non Functional Requirements

Requirement 1:	Extend the current vocabulary of verbal commands given by the user.
Result:	SUCCESS – Extended the vocabulary for actions mentioned in requirement 3.

Desirable Functional Requirements

Requirement 1:	The robot shall be able to distinguish between blocks of more than the above mentioned four colors.
Result:	SUCCESS – Extended color detection to Red, Blue, Green, Pink, Purple, Yellow, Black, White, Gray and Orange color.

Desirable Non Functional Requirements

Requirement 1:	Build a mobile application to send text commands to the robot.
Result:	SUCCESS – We used Amazon Echo to control the robot from mobile locations.

2 Setup

The robotic platform (i.e. Andy) used in this project consists of a RGBD camera mounted on a two stage, four axis neck and two seven degrees of freedom Barrett WAM arms. The neck and both arms are mounted to a base giving Andy a humanoid robot look. Both arms are equipped with 6 axis force/torque sensors and a four degree of freedom Barrett Hand. A table is located right in front of the robot. The task of this robot is to manipulate objects on top of this table surface.

3 Perception Subsystem

In the perception subsystem, we used the Asus Xtion Pro Live to capture 3D data and used Point Cloud Library(PCL) [1] for 3D point cloud processing. Figure 1 shows the captured images.

(a) RGB Camera Image (b) Kinect Depth Map (c) Detected Contours (d) Labelled Examples

Figure 1: Detecting Blocks

3.1 Detecting blocks

In order to detect the blocks, the first task is estimating the position of the table. After this, 3D template matching is done to get the cluster of the blocks. The 3D points in the world are then mapped to 2D points in the camera plane.

3.2 Color Detection

As shown in Figure 2, once we have the 2D points in camera plane, we calculate the enclosing contour for each block and then we calculate the average of the RGB values across the 3 channels.

Figure 2: Average Color Detection

3.3 Color Reasoning

Although RGB values are a convenient way to represent colors in computers, we humans perceive colors in a different way from how colors are represented in the RGB color space. So, the average color is converted from the RGB color space to the CIE Lab color space and ranking is done based on Euclidean Distance from the desired color.

For example, if the user said **Pick up the red block** when there are only pink and purple blocks on the table, Andy should rank pink and purple according to its RGB/LAB values and pick the one that is most similar to red as perceived by humans.

4 Language System

Our goal in this part was to take voice from the user as input, process it as per the constraints specified by the language part (see next section) and finally publish it to the language part as a ROS message.

4.1 Speech Recognition

Figure 4 shows the architecture of what we implemented. The first part was creating the intent schema using the Alexa Skills SDK. This defines the type of interaction the user can have with the robot. The examples of intent are put, move, place and the attributes of these intents can be object (block, canteen), color, distance, alignment, orientation and more. The performance of the recognition is determined by the quality of our intent schema. The schema is written in the JSON format and example sentences are also provided to create the language model.

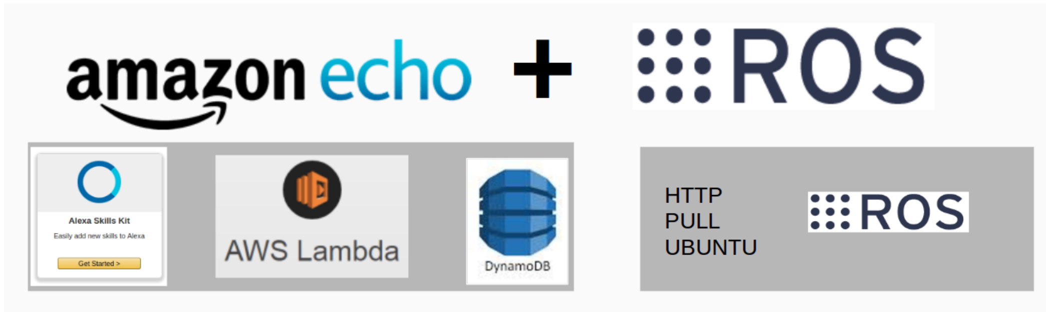


Figure 3: ROS and Amazon Echo Integration

The next task is to process the output given by the Alexa SDK which is an HTTP PUT request in the JSON format. Amazon Lambda function, the latest feature added by AWS, is used for the part. The JSON is processed to generate strings which can be given as input to the language part. Lambda function is written as a Python script and is scalable to handle any number of requests (provided we have the right tier subscription from Amazon). The last part on AWS is to insert the response from the Lambda function to the database. We select Dynamo DB as it is a noSQL database and it is quite fast. We also stored the timestamp of when the data was added to the database. The next part in the integration was to pull this data on the PC. For this, we wrote the Python script to execute an HTTP get request. Finally, the data was published on the same ROS topic which Sphinx was also publishing to. Hence, we can now switch between Alexa and Sphinx without changing any code.

4.2 TBS (Tactical Behavior Specification) Parser

Presently, most robots take commands from users which are mathematical and are expressed in terms of coordinates. For example, we can direct the robot to reach at a particular location with some coordinates (x, y, z) and avoiding obstacles along the way or to pick up an object kept at (x_1, y_1, z_1). The ability to learn new skills is missing. This part aims to introduce robots with cognitive abilities. A robot for example should be able to understand and execute a command like "Pick up the pink block that is next to the blue block".

The attributes that are associated to add these cognitive abilities to the robot include Natural Communication, Semantic Perception, Spatial Reasoning and Semantic understanding of tasks and the environment.

These attributes have been implemented in the project using the VOMA framework [2] which stands for (Verb, Object, Mode, Adverbial) Here, the mode can be something like "quickly", "a little bit", "more" and adverbial can be something like "to the left", "up", down etc.

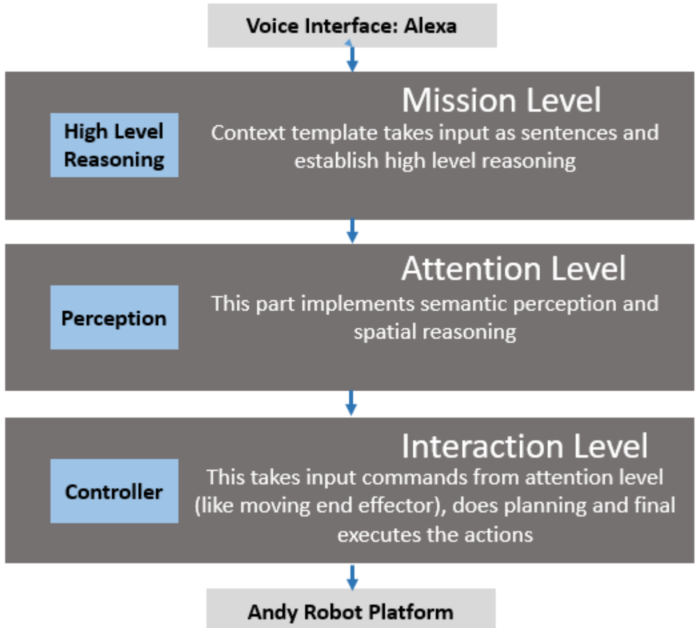


Figure 4: ROS and Amazon Echo Integration

5 Manipulation Subsystem

The main tasks in the manipulation subsystem were implementing new actions that were not implemented before.

5.1 Implementing missing actions

We implemented the following actions –

- *Rotations* – Rotate Left, Rotate Right, Rotate Up and Rotate Down.
- *Relative Placement* – Place on top, Place on left, Place on right.
- *Slight Movement* – Move little in left, Move little in right, Move little up or down.
- Put down

5.2 CHOMP

For motion planning, we used Covariant Hamiltonian Optimization for Motion Planning(CHOMP) [3]. The advantage of CHOMP over other sampling based planners is that it generates very smooth trajectories. Smooth motion makes the robot's actions look more natural to the people around it.

6 User Study

We performed a user study with 35 CMU Robotics Institute students.

Figure 5: Students participating the survey

6.1 Inference

- There was a lot of intuition involved which is hard for robots to decipher.
- Synonyms:
  - Stack: Put on top of
  - shift: Push
  - Take: Pick up
- Repeat actions:
  - Do the same thing
  - Undo the previous
  - Do it again
  - Repeat the same thing
  - once more

6.2 Summary of structures

Place <object> [to <geometric constraint>] [<object>] <geometric constraint>
Put <object> <geometric constraint>
Move <object> <geometric constraint>
Rotate <object> [by <degrees>]
Slide <object> <geometric constraint>
Repeat with <color> object [_except_ [<action>|<geometric constraint>]]
Keep <color> <object> <geometric constraint> <object>
Shift <color> <object> <geometric constraint>
Undo <action>
Align <geometric constraint> <color><object> <geometric constraint> <color> <object>

6.3 Summary of Constraints

- on the colored block,

- On the side of colored block
- Next to the colored block
- On the top of
- In the middle
- Facing yourself
- Opposite to ..
- On the left edge
- In front of you
- Center
- On ground
- perpendicular to
- at the end of
- adjacent to

7 Challenges

The key challenges we faced during the project –

- Integration of Amazon Echo and ROS was time consuming as the devices is relatively new and there are not much online resources available.
- The code base of the project is gigantic and it took us considerable amount of time to get familiarized with it.
- Katharina, who is the go-to person for help with the OCS code base went on vacation in May. So debugging errors without her help was a challenge.

8 Future Work

Adding the functionality which enables the user to build shapes and structures without having visual feedback of what Andy is doing. To do this, we need to increase the accuracy of the manipulation actions. Commands like "Put it on the top of green block" and "Align the block with the blue block" need to be executed with high accuracy on the manipulation side.

References

[1] Radu Bogdan Rusu and Steve Cousins. 3d is here: Point cloud library (pcl). In *International Conference on Robotics and Automation*, Shanghai, China, 2011 2011.

[2] Jean H Oh, Arne Suppé, Felix Duvallet, Abdeslam Boularias, Luis E Navarro-Serment, Martial Hebert, Anthony Stentz, Jerry Vinokurov, Oscar J Romero, Christian Lebiere, et al. Toward mobile robots reasoning like humans. In *AAAI*, pages 1371–1379, 2015.

[3] Nathan Ratliff, Matthew Zucker, J. Andrew (Drew) Bagnell, and Siddhartha Srinivasa. Chomp: Gradient optimization techniques for efficient motion planning. In *IEEE International Conference on Robotics and Automation (ICRA)*, May 2009.