Vocal Interaction with a 7-DOF Robotic Arm for Object Detection, Learning and Grasping

S. Rosa, A. Russo, A. Saglimbeni, G. Toscana
DAUIN, Politecnico di Torino
Corso Duca degli Abruzzi 24, 10129, Turin, Italy
Email: {name.surname}@polito.it

***Abstract*—This work presents preliminary results on the development of a system for multimodal interaction between a user and a robotic arm for human-robot cooperation tasks. In particular the system allows the user to ask the robot to grasp objects lying on a tabletop in the robot's working space using natural language. The objects are detected using a low-cost RGB-D camera. The robot is able to deal with fundamental issues such as multiple object instances and unknown objects. In the first case, the robot asks the user for further information on which particular object instance to pick. In the latter case, the robot asks the user to point to the unknown object to be learned. The robot communicates with the user using speech synthesis with natural sentences. The object detection pipeline is robust to partial occlusions. The system has been developed using ROS and was evaluated on a small 7-DOF anthropomorphic arm with a set of household items.**

I. Introduction

Object manipulation in domestic environments is a growing area of interest in robotics research. Interaction with common household objects is a crucial task for human-robot collaboration. Conversational robots are an important area of research because of their potential to provide a natural language interface with the user. Robots should be able to refine their already learned skills over time and/or acquire new skills by (verbally) interacting with its users and its spatial environment an [1].

The aim of our joint research laboratory between industry and academia is to investigate service robotics technologies where the focus is mainly placed on human-robot interaction and the relationship between robots and cloud computing, addressing technological issues as well as aspects related to ergonomics, cognitive perception, and relational experience.

The main idea, here, is to implement an autonomous system in which the robot can recognize different household objects for pick and place operations, while being able to address special cases such as unknown objects and duplicate objects. While the majority of works on object grasping are based on graphical user interfaces, (e.g., MoveIt! for ROS), keyboard or joystick, the aim of this work is to bring the user closer to the robot manipulator via more natural, multimodal interaction. For this purpose, there is no graphical interface between the robot and the user.

The system was implemented using the Robot Operating System (ROS) [2]. ROS is an open-source, meta-operating system for robot software development and it is nowadays becoming the de facto standard for robotic software development.

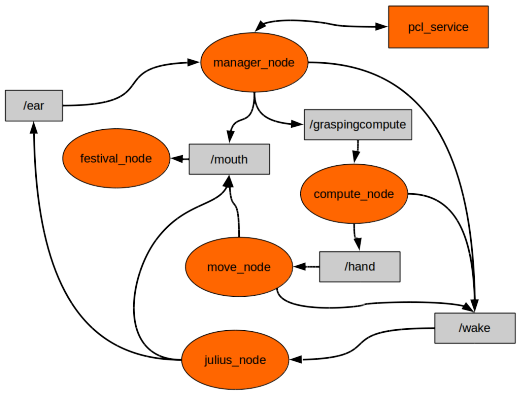

Fig. 1. Architecture of the system.

II. Architecture

The *Manager* node coordinates the other nodes and the information stream, starting from the *Julius* node and ending with the *Move* node. The *Julius* node interprets user commands using speech recognition. The *PCL* node implements object recognition. The *Festival* interacts with the user using voice commands. The *Compute* node computes the 3D pose of the object for grasping. The *Move* node implements path planning for the Cyton Gamma 300 arm for reaching and grasping the objects. An overview of the system architecture is shown in Figure 1.

A. Speech interaction

For interpreting user commands we use Julius, an open source library for speech recognition based on Hidden Markov Models, since it is able to interpret words using a large vocabulary with limited computational resources (Large Vocabulary Continuous Speech Recognition). We use the *Acoustic Models database* from www.voxforge.org. For vocal synthesis we use the *Festival* engine.

The interaction scheme goes as follows. The user can say:

- "ROBOT, TAKE THE [OBJECT TYPE]": if only one object of the requested type is available, the robot confirms the request, picks it up and notify the user when it is releasing it for the user to take. If more than one

instance of the type is present, the robot asks the user which one to pick; the user can say something like:
  - "TAKE THE [FIRST,SECOND,etc.] ONE FROM [LEFT,RIGHT]"

  If the robot does not know how to recognize the requested object type, it tells the user and asks to repeat the name. It then tries to fetch the training data for the new object in a remote database (we are using a custom ROS-based cloud robotics platform with a large database of trained objects) and, upon success, the object is picked up and delivered to the user.
- "ROBOT SLEEP": the robot answers "OK, BYE", terminates all running processes and quits.

While the possible interactions implemented so far are limited, both in the syntax and in the number of commands, the aim is to develop a more general interaction scheme based on natural language.

B. Object recognition framework

1) Learning objects: The dataset is composed of Point clouds representing objects were captured using the RGB-Demo app. Point clouds of each object taken from different points of view were aligned using 2D ARTag markers on the tabletop plane. ... Then the point clouds were post-processed using MeshLab, and triangulated using Poisson Surface Reconstruction.

Different 2D images of each point cloud were created from many points of view. Spherical tessellation was used to generate virtual views of the point cloud with fine angle granularity. For each view, global and local feature descriptors are computed. We used *Ensemble of Shape Functions* as global descriptors [3] and *Clustered Viewpoint Feature Histogram* (CVFH) as local descriptors [4], with the addition of *Camera Roll Histogram* (CRH) for encoding 6DOF poses. ESF is a combination of three different shape functions describing distance, angle and area distributions. The object is divided in stable, smooth regions using region-growing. Then, a VFH descriptor is computed for each region, allowing regions of the object to be occluded. A K-Nearest Neighbor (k-NN) classifier is then trained on the training set.

2) Detection: Because global features require the notion of object instance, the first step in the recognition pipeline involves a segmentation of the scene in order to extract the objects in it. Under the assumption that objects are lying on a relatively flat surface, we first estimate and remove the dominant scene plane using RANSAC, then we extract the objects using Euclidean clustering step (similar to flood filling) on the remaining points. For each object, feature descriptors are computed and the object is then classified using the k-NN classifier. An example is shown in Figure 2.

C. Path planning and grasping

We modeled the kinematic chain for the arm using the *Unified Robot Description Format* (URDF) for ROS compatibility. A path planning algorithm together with an inverse kinematic solver (IKFast) has been used for the generation of suitable trajectories to move the robot arm from its starting pose to the target pose. We detect grasping points at run-time, since the system must be able to cope with previously unknown objects. For each new object, we first cluster the corresponding point cloud into different surfaces, based on the direction of its normals. Then, the cluster that best fits the dimensions of the gripper (we use a simple 2-finger gripper) is selected. We extract the centroid of the cluster, its orientation with respect to the base of the robot, and the opening angle of the gripper.

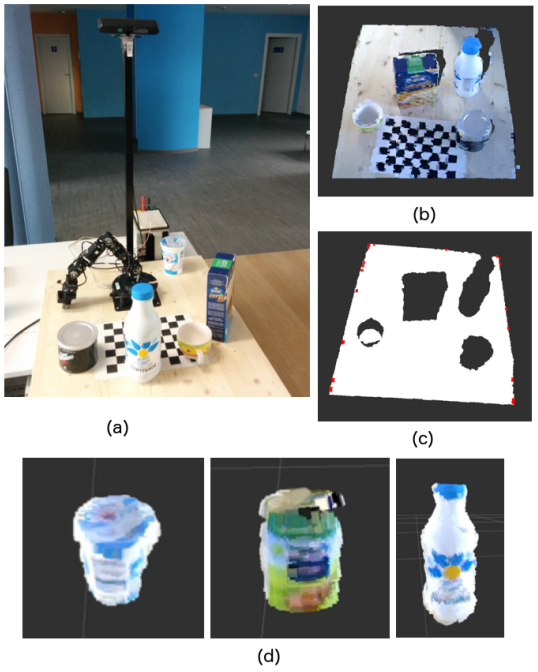

Fig. 2. Object detection pipeline. (a)The setup; (b) depth map of the table; (c) dominant plane; (d) segmented objects.

III. Robot setup and experimental tests

The system is implemented using ROS and the PCL library in C++ under Linux. The full source code will be available online. The anthropomorphic robotic arm used in experimental test is a Cyton Gamma 300 robotic arm. For the tests, a set of 10 household objects has been used. The RGB-D camera is an Asus Xtion Pro sensor mounted over a pole in order to have a full view of the table top. The software runs on a laptop which serves also as microphone and loudspeaker for interacting with the user. A video of the system is available at [1]

References

[1] H. Cuayáhuitl, "Robot learning from verbal interaction: A brief survey."
[2] "Ros (robot operating system)," Website, http://www.ros.org.
[3] W. Wohlkinger and M. Vincze, "Ensemble of shape functions for 3d object classification," in *Robotics and Biomimetics (ROBIO), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2987–2992.
[4] A. Aldoma, M. Vincze, N. Blodow, D. Gossow, S. Gedikli, R. B. Rusu, and G. Bradski, "Cad-model recognition and 6dof pose estimation using 3d cues," in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*. IEEE, 2011, pp. 585–592.

[1] https://www.dropbox.com/s/k7yj4juetcaltdg/cyton-vocal-interaction.mp4