A research of autonomous loading / unloading of consumer products using a dual-arm robot

Hiroaki Yaguchi, Shintaro Hori, Kazuhiro Sasabuchi, Kotaro Nagahama and Masayuki Inaba

Abstract—In this paper, we describe the problem definition and system construction for automatic loading / unloading of consumer products by dual-arm robot. We define the Loading / unloading task as complex of 3 elements; installation localization, object recognition, and robot planning. We also propose a hardware configuration and software components to manipulate a wide variety of items. The proposed system was evaluated in the warehouse environment with variety items and the store environment with multiple items, and the result shows ability of handling of a wide variety of items in the warehouse and reliability of a long-term continuous working in the store environment.

I. INTRODUCTION

Recently autonomous loading / unloading in the imitation warehouse environment was held. Configuration of the contest is almost equivalent to actual working sets, the robot requires to handle wide variety of items. According to a report by Correll et. al. [1], many teams focused into recognition process and used suction manipulator to handle many kind of objects. However, we consider that the loading / unloading problem includes not only recognition but integration of planning and action, especially robustness is required to work in actual environment.

The loading / unloading task in the warehouse environment requires that the robot does not break items, on the other hand, in the store environment the task requires that all items are sorted in addition. Also in the warehouse contest there is wide variety and few (1 or 2) items for each categories, however in the store, many same items are on the same place. We consider to solve these tasks with similar definition but different configuration with same robot.

In this paper we propose an autonomous solution for loading / unloading of variety objects in the warehouse environment [2] and for sorting multiple objects in the store environment [3] using a dual-arm robot.

II. HARDWARE CONSTRUCTION AND TASK CONFIGURATION

A. Semi-humanoid robot AERO

We use semi-humanoid robot AERO [4] (Fig. 1), uses smart actuator for whole body joints, it is constructed by motor, motor driver, and communication board. All actuators are connected via CAN as daisy chain including communication and power supply, also upper body and lower body has independent network individually as shown in Fig. 2. For this structure AERO can easily replace hand module and lower body depending on the application.

B. Configuration in the warehouse environment

In the warehouse environment, robot picks and places object between a “pod” having 12 bins (Fig. 3a) and 1 “tote”. Following 2 tasks are held individually; “stow” = loading all objects from tote into shelf, and “pick” = unloading designated objects from shelf into tote. There are known 38 categories shown in fig. 4 and each categories have 1-2 objects. For all objects, what is where bin / tote are known as symbolic, but unknown about its geometric poses. Finally all objects must be unbroken but must not be sorted.

For hardware construction, we employ large size lifter as
lower body because shelf is high and work space is wide vertically, also 3D camera is fixed on the right shoulder to observe upper bins. Robot has 3 types of end effectors as shown in left side of fig. 1; pinching hand for small object, center grasping hand to handle by straight arm motion, and vacuum attachment held on pinching hand for thin object.

C. Configuration in the store environment

In the store environment, the robot pick and place objects between display shelf having 2 boards (Fig. 3b) and working table fixed on robot base. There are 2 kinds of known categories and at most 4 objects for both categories. Symbolic information of object position is unknown. Finally all objects must be unbroken and sorted by category.

Shelf is lower than warehouse and workspace is narrow vertically, so robot has small lifter as lower body and 3D camera is fixed on the circular rail to rotate camera around yaw axis at waist for both recognition object at low position and navigation.

III. SOLUTION OF LOADING / UNLOADING TASK

Fig. 5 shows the solution of loading / unloading task. In this paper we define the problem as to pick and place objects between a shelf fixed on the environment and a cargo fixed on the robot or near place on the environment. We focus following 3 functions as requirement of loading / unloading automation; A) Shelf Localization, B) Object Recognition, and C) Robot Planning. To apply 2 problem setting having different configuration, hardware structure and software components are changed for each environment.

A. Shelf localization

1) Model-based bin detection: In the warehouse environment, both cart and shelf detection is based on geometrical model, for robot’s travel distance is too short. Target bin region of image is obtained from reprojection from 3D position of bin corners into camera coordinate.

2) Navigation using 3D camera: In the store environment we consider navigation using 3D camera to detect shelf directly because robot travels long distance (> 5[m]). Shelf position is estimated following 2 processes. Firstly distance and angle of wall fixed shelf are obtained by plane detection (Fig. 6). Secondly shelf position is obtained from 3D edge detection (Fig. 7). Note that height of shelf board is calculated from geometrical model.

B. Object recognition

1) Superpixel segmentation for clutter scene: In the warehouse environment, objects are segmented using super pixel based segmentation. In the stow (loading) task, firstly segmentation is applied to normal image of depth, secondly all segments having upper direction normal are detected as top face of object. Finally highest top face is selected as grasp target unless object classification. Fig. 9 shows a process of segmentation.

Fig. 9 shows an algorithm of recognition process in the pick (unloading) task. At first Region of Interest is cropped from bin region obtained in previous section, then object
classification is applied to object candidate segments from ROI. For object classification, we use Microsoft Azure cloud service learned from COCO[5]'s 80 object categories, object dataset itself is not used for learning, so classification result indicates not object class itself but multiple annotations of objects from Azure. Classification of designated object is applied associating object and annotations of classification result of cloud. Note that search range is easily narrow down less than 10 categories because objects included for each bin is known.

2) Saliency regioning for sorting: For target objects have only 2 classes, classification use object’s height in store environment. On the other hand, comparing to warehouse, this task has additional requirement that all objects are sorted finally. Object arrangement estimation on the shelf bin is necessary to place and sort another object. So at first occupied region by placed objects is detected using saliency map[6], then free space that can be place object is estimated from obtained region. Fig. 11 shows the diagram of shelf occupancy recognition.

C. Robot motion and grasp planning

1) Geometrical Model for Warehouse task: We construct geometrical shape model of 38 objects and shelf from problem definition for grasp and motion planning. All objects are classified into 5 categories based on geometrical shape as follows; a) Box shape, b) Cylinder shape, c) Flat shape, d) Complex shape, and e) Deformable shape. Grasp configuration is defined for each object with gripper type, multiple grasp poses and free rotation axis. Geometrical models are described using EusLISP [7]. Robot can handle designated object using a model because recognition process estimates object ID and its position. The robot and the pod also have geometrical models respectively, robot motion including waist position is planned considering collision check between robot arm and pod.

2) Accurate grasp planning for Store task: Comparing to warehouse, in the store environment robot handles multiple objects having same shape. We defined 2 types of geometrical shapes and grasp pattern: cylindrical pet bottle and cubic snack box. Final placement of objects is not considered
in the warehouse so in the grasp planning we define only simple grasp direction and via points, however in the store it is necessary to make sorted display, so we define grasp trajectory to keep object pose in the gripper considering placement.

IV. EVALUATION

A. Warehouse environment

Firstly we evaluated loading / unloading task in the warehouse environment (Amazon Picking Challenge 2016, Leipzig). In stow (loading) task robot loaded pets bowl (cylinder · torus) and brush (complex shape) successfully. In pick (unloading) task robot also unloaded knit groves (deformable) successfully. For either cases pinching hand is used to grasp objects. This result indicates that pinching hand is effective to pick up objects from lower position like floor of bin or tote.

B. Store environment

Secondly we also evaluated in the store environment. Environment is set up in conventional hall (Japan Robot Week 2016, Tokyo) and evaluation is held for total 24 hours (3 days × 8 hours). Object grasping is tried 518 and succeeded 461, successful rate was 89%. Object putting is also tried 471 and succeeded 453, successful rate was 96%.

In the environment light condition was so hard for it was dark and large color deviation, however successful rate is high because we did not employ unrobust information for light condition changing like color and considered noise reduction of depth information. Putting failure was avoided by grasp planning considering to keep object pose when grasping.

V. CONCLUSIONS

In this paper we propose a framework of autonomous loading / unloading task for dual arm robot. We also evaluated the system in the warehouse environment and the store environment, the robot loaded / unloaded many categories or multiple objects automatically.

We manually classified shape category and made geometrical model for relatively few objects, but actual store has more object category and some object requires to handle with care when it is easily broken. In the proposed method, models must be added manually to add new objects. It is more useful to obtain object models autonomously.

REFERENCES