Flexible Visually-Driven Object Classification using the Baxter Robot

Jose Avalos*† and Oscar E. Ramos*

*Department of Electrical Engineering, Universidad de Ingenieria y Tecnologia - UTEC, Lima, Peru.

† Department of Electrical and Computer Engineering, University of New Mexico, Albuquerque, NM, USA.

***Abstract*—One of the main applications for the robotics industry is the classification and manipulation of manufactured objects to increase productivity. Classical open loop robotic manipulation does not allow for changes in the environment without prior re-programming. To make the system more flexible, sensors are used to reduce the error and improve the efficiency for repeatable tasks. An important improvement consists in using visual feedback to avoid mechanical errors and chaining according to real-time circumstances. This work presents the classification of a group of objects based on their color and shape. The process includes image processing, inverse kinematics, and an automation algorithm which allows the task to be defined by the user or by a specific goal. This approach is validated using the Baxter robot and its internal cameras.**

Keywords - Baxter robot, robotic classification, inverse kinematics, image processing.

I. Introduction

Robust perception and manipulation of objects in varying environments is one of the biggest challenges currently faced in robotics. During the last decades several manufacturing processes consisting of repetitive and hazardous tasks have switched to using robotic technologies and automated systems. The field of robotics thus advanced in this direction, and very powerful robotic arms were developed to improve manufacturing efficiency [1]. However, these robots are typically unaware of their environment and, thus, metallic cages are used to keep them away from human workers. This unawareness is a main constraint for flexible manipulation and even for human safety. Moreover, perception problems also arise in state-of-the-art robotic systems and are currently a crucial restriction for robot autonomy in unknown environments. The utter goal in this direction is to develop robots as coworkers able to autonomously help human operators achieve specific tasks.

Machine vision constitutes one step towards perception autonomy. In industrial environments, it can also improve the sensitivity and the resolution, and it is being increasingly used as a guidance for robot controllers in small part identification and inspection [2]. However, a main challenge in autonomous manipulation consists in developing the capacity of visually recognizing specific objects and modifying the motion process accordingly [3], which is called flexible object classification and manipulation. Color segmentation is used in several robotic platforms as the basis for visual recognition of specific patterns. In the case of outdoor mobile platforms with on-board cameras, it is almost certain that lighting conditions will be time and space dependent, and color-based approaches will present a poor performance. Similar effects will be encountered when the lighting environment for which the robot was calibrated changes [4]. In these cases invariant features such as corners or edges can be more useful. However, in closed and well illuminated environments, as is the case of most automated factories, a constant object color is a good approximation that is satisfied in practice.

978-1-5090-6363-5/17/$31.00 ©2017 IEEE

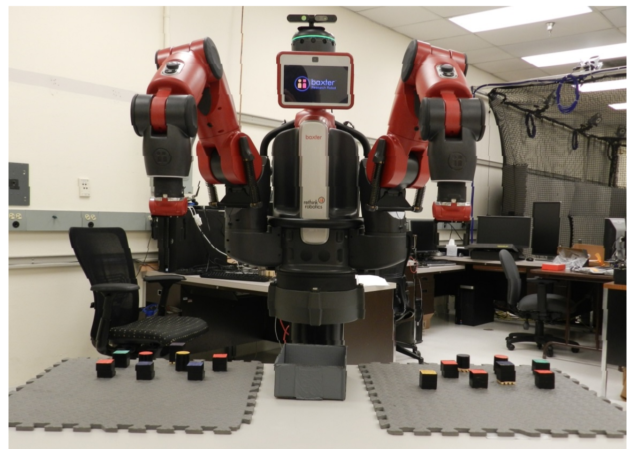


Fig. 1: Baxter workstation. The robot employs its end-effector camera to identify cubes of a specific color and put them inside a box using both arms simultaneously.

This work is a step towards autonomous object classification and manipulation. The paper presents a vision-based system able to identify objects on the fly based on their color and regardless of their position, which can be arbitrarily changed. After identification, an inverse kinematics based approach is used to manipulate the objects and classify them according to a given criterion. These tasks are achieved without any human intervention in the process. The proposed system was tested using a Baxter robot in the Multi-Agent, Robotics, Hybrid and Embedded Systems (MARHES) Laboratory at University of New Mexico. In Fig. 1, workspace is presented.

II. Methodology

The proposed system, summarized in Fig. 2, consists of three main stages: visual recognition trough arm-cameras for object identification, inverse kinematics for object manipulation, and high-level task and goal specifications. This section presents the details of these steps.

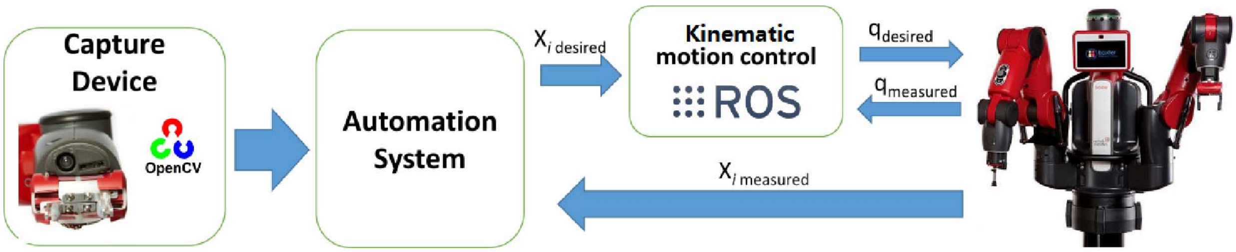


Fig. 2: Scheme showing the parts of the system from the Baxter camera to the motion generation to select the objects.

Fig. 3: Baxter camera and electric parallel gripper. The type of gripper was chosen according to the object dimensions. However, in other cases, the gripper can obstruct the camera's view.

A. Object Position based on Visual Inspection

The system uses the camera located at the end effector of each arm, as shown in Fig. 3. All the movements are performed with the arm pointing vertically down at the table. To this end, the orientation is kept constant throughout the arm motion. Since the objects will be located on a predefined plane, this arm constraint simplifies the conversion from image pixel coordinates to the robot workspace coordinates. In more general scenarios, a plane transformation can be used. For our case, both the camera calibration factor (c_c) and the distance from the table (d) can be used to convert pixel coordinates of the object $\mathbf{p}_p$ to robot coordinates $\mathbf{p}_r$ (relative to the gripper pinch point) as:

$$\mathbf{p}_r = d\ c_c(\mathbf{p}_p - \mathbf{o}_p) + \mathbf{p}_d + \mathbf{p}_g. \tag{1}$$

where $\mathbf{o}_p$ is the center pixel coordinate, $\mathbf{p}_d$ is the robot default position, and $\mathbf{p}_g$ is the distance variation to reach the desired object. We assume that d is a constant if the initial point for identification is always the same $\mathbf{p}_d$. In this case, d is measured between the camera and the table. The robot default position is obtained using forward kinematics. The camera calibration factor c_c is obtained by analyzing the camera conditions; however, these values are different even for both Baxter cameras.

B. Robot Motion Generation

The motion of the robot arms is based on inverse differential kinematics. Let the robot joint configuration vector be $\mathbf{q} = (q_1, q_2, \cdots, q_n)$, where n is the number of degrees of freedom. Let task i be represented by a Cartesian position $\mathbf{x}_i$ and its reference by $\mathbf{x}_i^*$, so that the task error can be defined as

$$\mathbf{e}_i = \mathbf{x}_i - \mathbf{x}_i^*. \tag{2}$$

Since the Cartesian position is a function of the joint space $\mathbf{x}_i = f_i(\mathbf{q})$, given by the forward kinematics function f_i, the task itself is also joint dependent. The derivative of the task error is related to the joint angle velocities by

$$\dot{\mathbf{e}}_i = J_i\dot{\mathbf{q}} - \dot{\mathbf{x}}_i^* \tag{3}$$

where $J_i = \frac{\partial f_i}{\partial \mathbf{q}}$ is referred to as the task Jacobian.

To solve for the joint angular velocities $\dot{\mathbf{q}}$ in (3), a task weighted approach based on [5] was used. Considering N tasks, the problem can be expressed as

$$\min_{\dot{\mathbf{q}}} \left\{ \sum_{i=1}^{N} w_i \|\dot{\mathbf{e}}_i - J_i\dot{\mathbf{q}} + \dot{\mathbf{x}}_i^*\|^2 \right\} \tag{4}$$

where the i^{th} task has an associated weight w_i which defines its relative importance. Considering a small reference variation between successive control times, the term $\dot{\mathbf{x}}_i^*$ can be neglected, and it can be shown that this optimization problem is equivalent to the following quadratic program (QP):

$$\min_{\dot{\mathbf{q}}} \left\{ \dot{\mathbf{q}}^T W \dot{\mathbf{q}} + \dot{\mathbf{q}}^T \mathbf{p} \right\} \tag{5}$$

where

$$W = \sum_{i=1}^{N} w_i J_i^T J_i, \quad \mathbf{p} = -2\sum_{i=1}^{N} w_i J_i^T \dot{\mathbf{e}}_\mathbf{i}.$$

In order to generate feasible motion, avoiding mechanical damage to the robot, joint limits need to be taken into account. Provided that the kinematic control is expressed as a QP, the joint limits can easily be added as the following constraint:

$$\max\left\{ \frac{\underline{\mathbf{q}} - \mathbf{q}}{\Delta t}, \underline{\dot{\mathbf{q}}} \right\} \leq \dot{\mathbf{q}} \leq \min\left\{ \frac{\overline{\mathbf{q}} - \mathbf{q}}{\Delta t}, \overline{\dot{\mathbf{q}}} \right\} \tag{6}$$

where $\underline{\mathbf{q}}$, $\overline{\mathbf{q}}$ are the lower and upper joint angular limits, $\underline{\dot{\mathbf{q}}}$, $\overline{\dot{\mathbf{q}}}$ are the lower and upper joint velocity limits, and Δt is the control time between each control signal. The relation between joint velocities and joint positions can be found using a first order Taylor expansion.

The previous scheme provides the desired joint velocity, but position controlled robots such as Baxter rather require

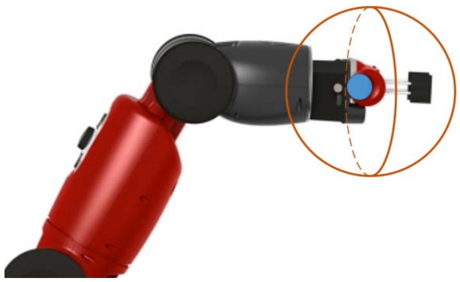


Fig. 4: Baxter marge position. It defines the parameter space to catch/drop the object. The action arm will wait a defined period time after detect stay in this space.

joint positions. This can be obtained from (5) using Euler integration as

$$\mathbf{q}_{k+1} = \mathbf{q}_k + \dot{\mathbf{q}}_k \Delta T \tag{7}$$

where $\mathbf{q}_k$ represents the joint configuration at time k. However, not all Cartesian points have an inverse kinematics solution. Moreover, since we are considering a redundant robot, there will typically be an infinite number of solutions for a single desired pose. Multiple solutions are reduced due to the constant orientation defined in II-A, and the existence of solutions is guaranteed as the robot does not exceed its workspace.

C. Automation System

To conclude the automation system, a cyclic algorithm needs to be developed. This section presents the proposed system, which is divided in three parts: initial position setting, object localization, and objective space shifting.

1) Initial position: The robot has the initial position described in II-A that includes a constant point of reference for a fast camera calibration. This position is arbitrarily determined by the user and the only constraint is its orientation, as previously discussed. The robot stays in this position to obtain the mean of 10 or more samples and to reduce the error probability for the desired position. Because of this, it is necessary to set some parameters which can optimize the time without losing precision.

2) Object localization: The robot moves its arms using the inverse kinematics based motion generation scheme previously discussed. To properly approach the object, the orientation is kept constant at the beginning of the motion and when the arm is over the object. After that, in order to grasp the objective, the final position is sent using a 10 hz frequency control loop compared to the normal frequency of 100 hz. The effect is the reduction of the motion speed and the main advantage is error reduction. Also, in extreme cases, the grippers can help to catch the object.

3) Shift to objective space: The last part of the system includes the the definition of a space for principal grasping actions. To this end, a three-dimensional space is defined as shown in Fig.4. This space considers a ratio of acceptance to open the gripper and place the object inside the box.

III. Experimental Setup

The experimental setup was shown in Fig. 1. It consists of the robot and a horizontal table which contains the objects to be moved. There is also a bin at the middle where the objects will be finally located.

A. Hardware for the experimental environment

The robot used to demonstrate the application was the Baxter robot, which is a compliant dual-arm humanoid torso developed by Rethink Robotics. Each arm has seven degrees of freedom and an installed gripper at its end. This allows for the manipulation of objects.

B. Software Framework

1) Robot Operating System (ROS): Our approach uses ROS, which is becoming a *de facto* standard in robotics, as a middleware for interconnecting all the elements in the proposed system. ROS works under Linux and is the default platform for the Baxter robot.

The communication between all the nodes is done through dedicated ROS topics. The node running the inverse kinematics algorithm reads the message that specifies the desired end-effector pose and computes the desired joint angular positions. These control elements are then published in separate topics that are read by the low-level controllers of each arm of the robot. The whole process runs at 100 Hz, which can be considered real time for practical situations for robot motion.

2) OpenCV (Open Source Computer Vision): It is one of the most popular libraries for real-time computer vision [6]. As the robot has to identify the position of the object and process the estimated position, OpenCV was used. The Baxter robot system publishes the camera information as a topic in ROS. This feature helps to build ROS nodes without delay.

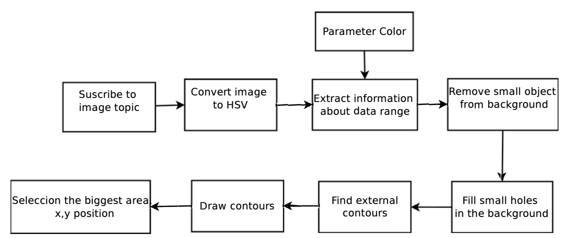
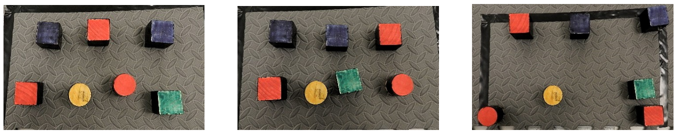


Fig. 5: Flow diagram of *OpenCV* for image classification. The algorithm using color parameters to find the centroid position of the object.

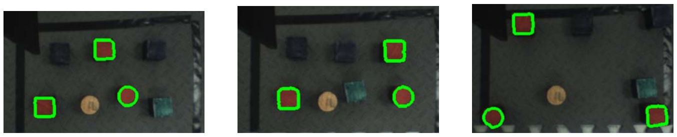
Using OpenCV, the parameters of (1) are computed with the transformation information stored in the image topic. This process is shown in Fig. 5 and is based on the HSV color space with the information in Table I. This values are optimized for workspace and results could change for drastic brightness variations.

TABLE I: HSV parameters for colors.

HSV	High_H	Low_H	High_S	Low_S	High_V	Low_V
Green	90	50	147	74	255	160
Yellow	50	10	175	85	255	70
Red	9	0	255	50	255	50



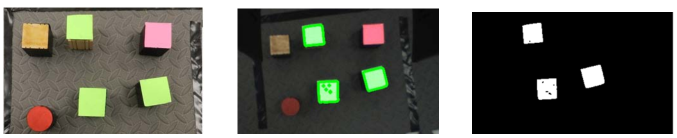
(a)



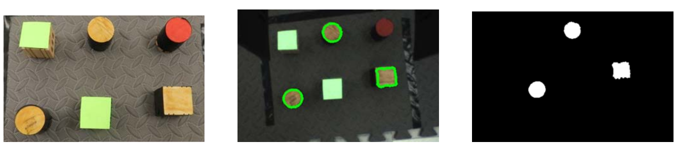
(b)

(c)

Fig. 6: Red objects identification. (a) High resolution camera output of the environment. (b) Identification of the red objects using the robot camera. (c) Segmentation of the red objects.



(a)



(b)

Fig. 7: (a) Green and (b) yellow objects identification. Both cases show the versatility for identifying different colors.

IV. Results

Using the previously experimental setup, several runs of the system were performed with different positions of the objects. Fig 6 shows the camera output of the objects on the table for red objects. After identification, a segmentation is performed to then compute the centroid of the object, which is used as an indicator for the grasping position. The color objective can be changed according to user. In fig.7, the parameters of the algorithm are adjusted to identify the green and yellow colors.

An example of joint space motion showing the desired and the achieved joint angle (for a typical joint) is shown in Fig. 8. A delay can be observed between both signals and is mainly due to the inertia of the robot. However, this delay did not cause any loss of information from the user. The error for the tests is presented in Fig. 9. Due to the dimension of the grasps and the delimitation of the work space, the error is not reduced to zero. However, this does not affect the autonomous selection of the objects made by the robot, which was the main purpose of the work. For all the considered cases, the robot show a correctness with an error of 1 case per 20 repetitions.

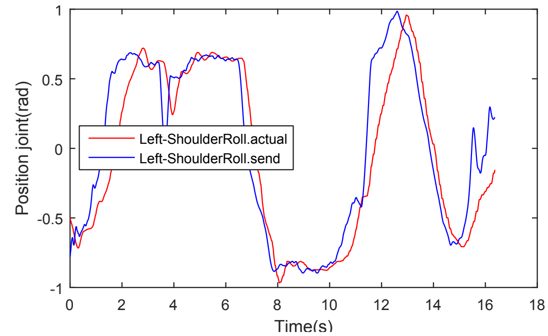


Fig. 8: Comparison between the position sent y the actual position of *Shoulder Roll*.

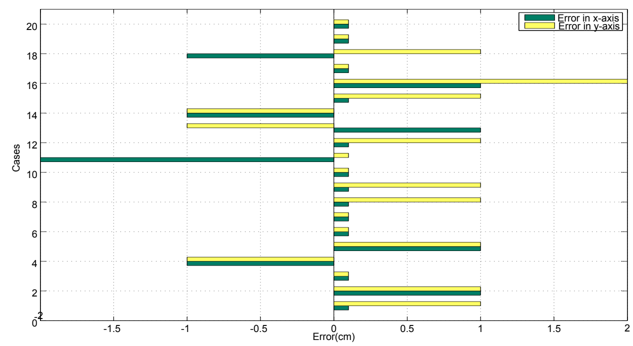


Fig. 9: Result for 20 consecutive test using the Baxter robot. Represents the distance error in axes x and the error distance in y axes.

V. Conclusion

Our proposed framework allows for the autonomous and flexible classification and manipulation of objects based on visual features. An example of the methodology showed a robot using sensed color cues for the position feedback. Moreover, this experiment can be further scaled as a major solution for industrial applications where autonomy is needed. Although the classification algorithm was able to properly grasp objects with simple shapes, further processing would be needed for handling more generic objects. Future work includes more robust object detection approaches as well as a force-based compliant motion control scheme to grasp more complex objects under different lightning conditions.

References

[1] J. F. Engelberger, *Robotics in practice: management and applications of industrial robots*. Springer Science & Business Media, 2012.

[2] P. Andhare and S. Rawat, "Pick and place industrial robot controller with computer vision," in *Computing Communication Control and automation (ICCUBEA), 2016 International Conference on*. IEEE, 2016, pp. 1–4.

[3] C.-h. Chen, *Emerging topics in computer vision and its applications*. World Scientific, 2012, vol. 1.

[4] H. Okuda, R. Haraguchi, Y. Domae, and K. Shiratsuchi, "Novel intelligent technologies for industrial robot in manufacturing - architectures and applications," in *Proceedings of ISR 2016: 47st International Symposium on Robotics*, June 2016, pp. 1–6.

[5] O. E. Ramos, "A kinematic whole-body control system for highly redundant robots," in *IEEE Andean Council International Conference (Andescon)*, 2016.

[6] J. Howse, *OpenCV Computer Vision with Python*. Packt Publishing Ltd, 2013.