Using Human Actors in Robotic Manipulation Study: An Initial Attempt with Peeling Banana Task

○Fucai Zhu, Akihiko Yamaguchi, Koichi Hashimoto

Abstract : Our goal is automating complicated manipulations, especially manipulating deformable objects. This paper explores a strategy to simplify this challenge by introducing a human actor obeying the system instead of using robots. We report the initial attempt with a task of peeling banana.

1. Introduction

Manipulating deformable objects is a common activity in human daily life. They are difficult tasks for robots to perform. The bottlenecks are hardware (robot hands are less functional than human hands), control strategy (lack of robust AI) and perception. Deformable objects in general have huge state variables, and their dynamics are hard to model; i.e. their movement are unpredictable. Since the dynamics are unmodeled, their control problem is considered as reinforcement learning problem in general [10]. For solving such difficult tasks, we need to combine different methods including machine learning, control theory, motion planning, and so on. However there is no established theory unifying these methods. We emphasize that the current capability of robots in manipulating deformable objects is far from that of humans. There are many problems to solve; even if we have a good AI, it cannot perform tasks without good robotic hands.

Learning from human demonstrations is an approach to tackle to such difficult tasks [2]. However implementing a learned behavior on robotic systems is complicated due to the difference of the embodiment between humans and robots. Even implementing a skill takes over a year (e.g. [14]). Since simulating deformable objects is another challenge, practicing on simulation is also difficult. In order to boost the research of automating deformable object manipulation, we need a framework to practice control strategies and/or perception methods for such manipulations without suffering from the hardware bottlenecks. This paper explores a framework where we introduce a human actor obeying the system. The human actor is expected to perform control instructions generated by the system, but not to do things that are not indicated by the system. Such an actor is considered as an obedient robot with a good hardware. With this framework, we can practice a control strategy and/or a perception system in manipulating deformable objects without being frustrated by the hardware difficulties. As the case study, this paper reports the implementation of a peeling banana task and the results of experiments with subjects.

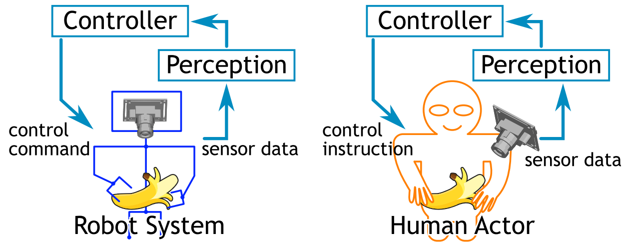

Fig. 1: Left: Typical robotic system for manipulation. Right: Conceptual illustration of the proposed framework.

2. Overview

Figure 1 shows the conceptual illustration of the proposed framework compared with a typical robotic system for manipulation. In a robot manipulation, sensor data is processed by the perception system, and then the controller decides control command to actuate the robot. In the proposed framework, the robot is replaced by a human actor who obeys control instructions generated by the system. We introduce sensors similar to those in the robotic system, whose data is processed by the perception system. The controller decides a control instruction according to the control strategy. The human actor executes the control instruction. Similar to the robotic system, this sensing-control loop is repeated until a terminal condition is satisfied.

The design of the control instructions is important. Our guideline to design a set of control instructions is as follows:

- Human understandable.
- Ambiguity should be reduced as much as possible so that different human actors perform the instructions similarly.

A text instruction is sometimes difficult to remove the ambiguity. For example “grasping the body of banana with the thumb and index fingers of the left hand” is ambiguous in terms of the position and orientation to grasp. We choose to combine a text instruction and a graphical user interface (GUI) to represent the control. For example the grasping position and orientation are indicated through the GUI.

2.1 Case Study of Peeling Banana Task

In order to investigate if the proposed framework works, we apply it to a peeling banana task. Peeling banana is difficult for robots to perform. In our preliminary experiments [11], we operated a dual arm robot Baxter with a joy-stick controller to peel a banana. Although the control strategy was decided by the human operator, it was difficult to peel since the robot had two finger parallel grippers, and the robot could not peel it with a usual peeling motion. Since the physical properties of bananas vary, such as shape and hardness, sometimes bending the tip of banana breaks the skin, but sometimes does not. Peeling banana is a good example to explore the proposed framework.

Since the most challenging part of the peeling banana task is the design of control strategy, we focus on that component. For the perception, we take a color-based approach with artificial marking on banana.

From the observation of [11], we found that the human operator used various skills, such as grasping the banana body or tip, twisting, pulling the grasped point, and so on. We take an approach of behavior generation with a library of skills. Each skill is designed as a parameterized policy. The library consists of four skills: grasp, twist, pull, and release. The combination of skills is predesigned according to human demonstrations. It is referred to as a behavior graph since the control strategy forms a graph structure of skills. The behavior graph of peeling banana is manually optimized to adapt to a variety of bananas through the experiments where the proposed framework is used. In order to demonstrate that the optimized behavior graph generalizes over different human actors, we conduct experiments with subjects.

2.2 Related Work

There is a study of learning behavior models with the "game with a purpose" framework researched by the group of Beetz [7]. They use a virtual environment where a human operator manipulates the objects with a motion capture glove to achieve a task (game). Since it is performed under a virtual environment, all low-level data is recorded and analyzed to create high-level behavior models. The behavior models can be tested in the virtual environment as well as the actual robotic system. They explored pouring liquids and making pancakes [4, 5]. A similar approach is taken in [9] to demonstrate complex manipulations. A drawback of such an approach is that it is limited to the capability of simulation. They simulated liquids such as pancake mix with many spheres, but simulating other deformable objects such as bananas is much complicated. We can use any objects in our approach.

There is another approach of teaching motions to robots through teleoperation [15]. They used a dual arm robot in reality and operated it to demonstrate manipulating cloths through a teleoperation system with a head mount display. The robot learned a behavior model from the demonstrations and could perform it. Building a robot system that performs complicated manipulations is our final goal. Such a teaching by teleoperation seems to be ideal for that purpose. Our framework is better in two cases: one is avoiding the construction of expensive robotic system, and the other is quickly testing a control strategy that cannot be explained by a policy learning like [15].

The difference from the learning from demonstration (LfD) approach [2] is that the behavior of the human actor is (ideally completely) decided by the system. While the objective of LfD is learning behavior models from humans, the objective of our framework is testing and improving the behavior models with human actors. Actually we take the LFD approach to construct the behavior graph of peeling banana.

Regarding the control strategy, the skill-based approach has decades of history. For example Hasegawa et al. developed a robot system with skills for robust manipulation [6]. Yamaguchi et al. studied robotic manipulation of liquids and powders in various situations where a behavior generation with a skill library is used with reasoning and learning methods [14]. They developed machine learning and reasoning tools for this approach [13, 12]. We think this approach is promising for complicated robotic manipulations not only peeling banana.

3. Control Strategy

By observing and analyzing how human peels banana, we manually extract skills and construct a behavior graph represented by a state machine. According to our observation and experiments, breaking upper tip of banana is always necessary to peel off banana skin without damaging the flesh. Thus, in the behavior graph, the skill performed at the first time is designed to be breaking the upper tip of banana. For the simplicity, we preset only two peeling directions and make our system peel the banana from one end to another along these two directions. The achievement of the peeling banana task is decided by the ratio of the flesh area over the initial skin area. If this ratio exceeds a certain threshold, it is considered as the task completion.

3.1 Skill Library

We represent each skill as a function taking a parameter vector as input. The number of parameters has been empirically decided. The parameterized skills we designed for peeling banana task are listed below. For the simplicity, we only consider the skills in 2 dimensional space.

1) Grasp

$$G(h, g_x, g_y, \sigma) \tag{1}$$

$h \in \{R, L\}$ indicates the hand that the operator is performing the skill with. (g_x, g_y, σ) is grasp pose defined by two coordinates $(g_x, g_y) \in \mathbb{R}^2$ and a rotation angle from X-axis. When performing the Grasp skill, the hand grasps on pose (g_x, g_y, σ).

2) Twist

$$T(h, a_x, a_y, \xi) \tag{2}$$

(a_x, a_y) is the position of the anchor point and ξ is the twisting angle. When performing the Twist skill, the hand rotates around the anchor point for an angle ξ in the coordinate plane.

3) Pull

$$P(h, d, \theta) \tag{3}$$

d is the pulling distance and θ is rotation angle from X-axis that indicates the pulling direction. When performing the Pull skill, the hand moves straightly in direction θ for a distance of d.

4) Release

$$R(h) \tag{4}$$

When performing the Release skill, the hand moves back to its initial pose.

3.2 Behavior Graph

The behavior design in this work is described in a graphical model shown in Figure 2. The behavior graph has been optimized by human experience. The circle vertices represent the states of the procedure. The square vertices represent basic skill from skill library. An edge between two vertices indicates the their direct interaction. (B_x^1, B_y^1) is the centroid position of No.1 segment of the object main body. (Z_x, Z_y) is the position of upper tip. (Q_x, Q_y) is the point coordinates of banana skin that is closest to line $y = -x$ in LCS. S_{flh} is detected area of flesh. S_{trg} is a threshold proportional to the measured area of the object. The proportion is set to be 0.7. σ_2 chosen randomly from $\{-45°, 45°\}$. Loop counter is a memorizer for counting the number of times that state 1 has been reached. D is proportional to measured object length. The object length is measured as the distance between two banana end points. The proportion is set to be 0.6. $\theta_0 \in \{0°, 45°\}$. θ_0 is equal to 0° when σ_2 is 45°. θ_0 is equal to 45° when σ_2 is −45°.

4. Perception System

Perception system of this work takes the video frame captured by a RGB camera as input and outputs semantic segmentation of the frame and the location of the object parts according to a designed local coordinate system.

4.1 Color-based Semantic Segmentation

We use super-pixel segmentation method SLIC [1] with color histogram [8] and support vector machine (SVM) [3] classification. The perception algorithm is illustrated in Figure 3. By applying SLIC on current video frame t , we will get N super-pixel partitions. Let us define n_i^t as the associated cluster index assigned to pixel i

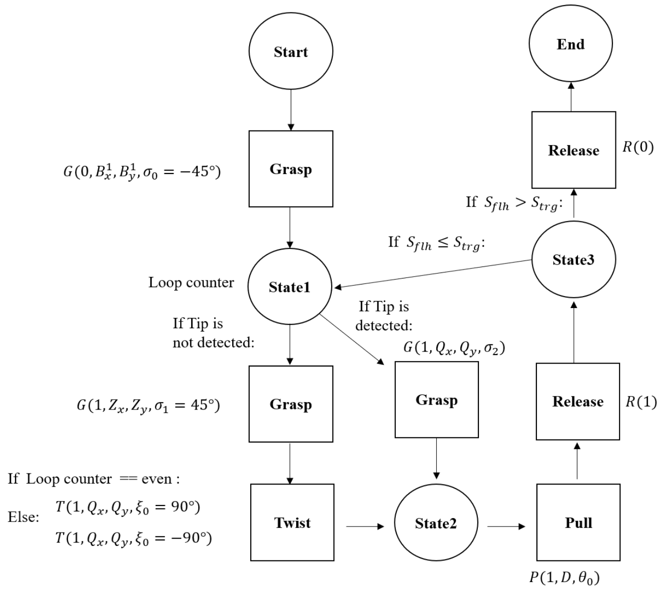

Fig. 2: Behavior graph with empirically optimized parameters.

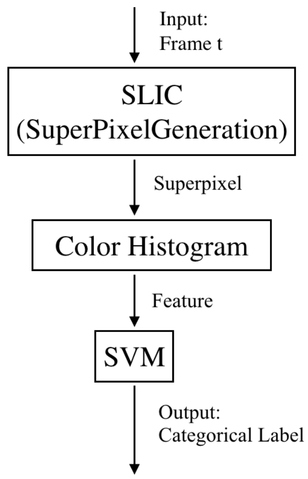

Fig. 3: Perception algorithm based on super-pixel segmentation, color histogram and SVM.

in frame t. ($n_i^t \in \{1, ..., N\}$.) Then by calculating the color histogram of the group of pixels that share the same index, we will get a feature matrix $C^{N\times L}$. J is the number of bins in color histogram. Each row of $C^{N\times J}$ is the vector of color histogram corresponding to the superpixel with certain index. Then we input the matrix into a pre-trained SVM and output a categorical label for each component in $\{0, 1, ..., K\}$. K is the number of the scene components. The SVM is trained with 150 sample histograms for each component class. In Figure 4, we can see the result of semantic segmentation based on the prediction from SVM.

4.2 Manipulation Features

For object part localization, we defined a local coordinate system (LCS), in which the center of upper tip

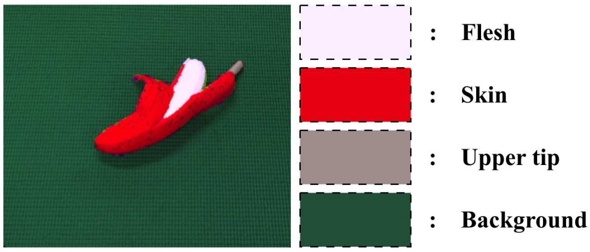

Fig. 4: Semantic segmentation result.

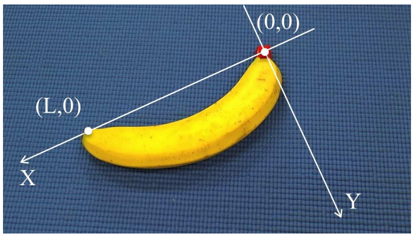

Fig. 5: Local Coordinate System. L is the measured length of the object.

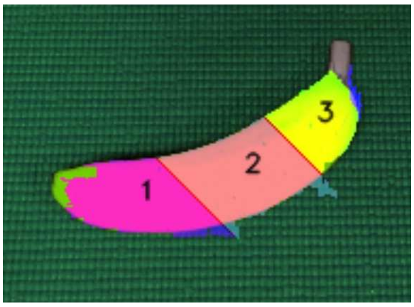

Fig. 6: Main body is segmented into 3 parts.

of banana is considered to be original and downer tip of banana is on X-axis and main body of banana is in first quadrant. The LCS is illustrated in Figure 5. With the help of color markers, we detect 2 end points of the banana in a video frame and get transformation matrix W, which is from locate coordinate system to global coordinate system. Besides, for developing the decision making algorithm in controller, we segment the main body of object banana into three parts equally along X-axis, which is shown in Figure 6. Then we consider the centroid of each part as its position, denoted as $(B_x^p, B_y^p); p \in \{1, 2, 3\}$.

5. Experiments

During the system operation, a series of instructions are sent to human actor through a GUI. To constrain human actor's dexterity and also simulate the manipulator with two-finger gripper, we require human actor to use only thumb and index finger at each hand to execute the instructions. To validate the performance of replacing robot arm with a human actor, we conducted a subject experiment. During the experiment, a subject is asked

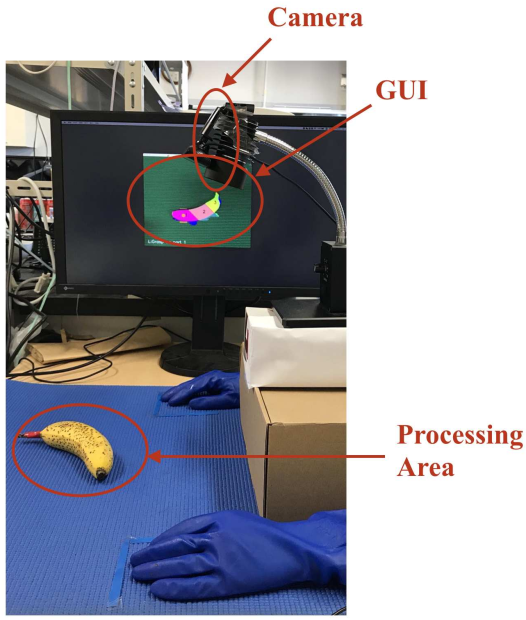

Fig. 7: Experimental setup.

to act as the human operator in our system and operate in the task of peeling banana. We have conducted the experiment for 5 times with 5 different subjects. We are trying to demonstrate with the experiment result that the human actor can act as robot arm in the procedure of creating the theory of behavior generation, the method of making use of human actor can be applied universally and the skills are enough systematic (the variance among subjects are small) as if we are using robots.

5.1 Experimental Setup

The experimental setup is shown in Figure 7. The object processing area is placed on a table and human actor is asked to sit in front of the table. The RBG camera is hanging above the object processing area. The GUI is shown on a screen placed at the side of the table.

5.2 GUI

We have designed a GUI to translate the policies to human operator. The semantic segmentation result is shown in GUI window with the instruction texts appearing in the left-down corner. During executing the instructions, human operator is only allowed to look at the GUI instead of real scene in order to constrain the effect of human's perception in the system. An example of the GUI is illustrated in Figure 8.

5.3 Instruction

The instruction given in GUI is designed to inform human actor the necessary information for them to execute the policy generated from the system. The instruction

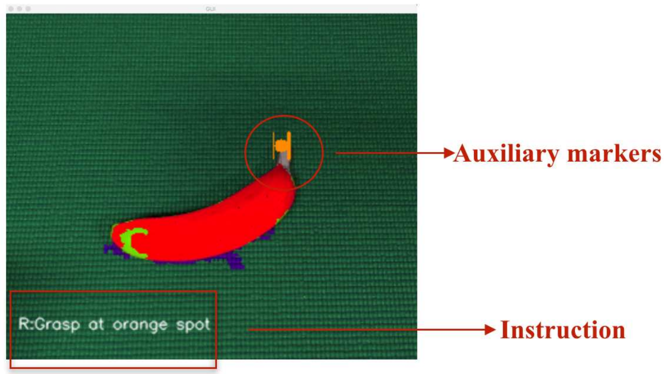

Fig. 8: GUI for human operator. Auxiliary markers are designed to help human operator to execute the instructions.

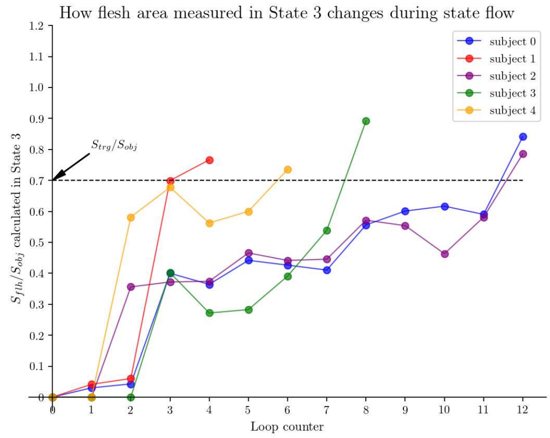

Fig. 9: Change of peeled area during state flow in each subject experiment.

is in the format of "hand indication + fingers motion description". All of the instructions are listed in the below Table 1. With the auxiliary markers of two lines shown in Figure 8, in which the thick one indicating the 2D pose of thumb and thin one indicating 2D pose of index finger, human actor is able to follow the instruction and execute the policy systematically.

Skill	Instruction Text
Grasp	R: Grasp at the orange spot.
	L: Grasp at the orange spot.
Twist	R: twist around blue spot to white spot.
	L: twist around blue spot to white spot.
Pull	R: Pull to green spot.
	L: Pull to green spot.
Release	R: Release.
	L: Release.

Table. 1: Instruction detail.

5.4 Result Analysis

In all of the experiments, our system has achieved to peel the object banana successfully within a certain period of time. Figure 9 shows how the peeled area is changed while the system is running. We can find that the area is growing to the maximum gradually with empirically optimized parameters during the execution of the policy with human operator in all of the experiments. This result shows that the performance of the generated policy can be evaluated objectively in our system with human operator. And it also demonstrates that our proposed policy generation framework based on skill library is capable of generating a efficient policy for manipulating the complicated manipulation task such as peeling banana. We recorded human actor's operation in videos with a side video recorder and a GUI screen capture software. From the video we found that the banana and subject variation is the main reason of the performance difference between subjects.

6. Conclusion

The result of subject experiment have demonstrated that introducing human operator in manipulation system is a promising approach to simplify the difficulties in deformable object manipulation. With human operator design, robotic researchers would be able to empirically decompose a complicated manipulation task into simple skills execution and verify the robustness of their designed behavior graph in an easier way. Besides, we have built up a system that can achieve the complicated task of peeling banana. It is an important step to fully automate the complicated deformable object manipulation. In the future work, there are several directions we can focus on based on our established framework such as improving perception system, extending the policy into three dimensional space, automating the policy generation and including other complicated manipulation tasks.

7. Acknowledgement

This work was supported by JSPS KAKENHI Grant Number JP16H06536.

References

[1] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(11):2274–2282, Nov 2012.

[2] Aude Billard and Daniel Grollman. Robot learning by demonstration. *Scholarpedia*, 8(12):3824, 2013.

[3] Corinna Cortes and Vladimir Vapnik. Support-vector networks. In *Machine Learning*, pages 273–297, 1995.

[4] A. Haidu, D. Kohlsdorf, and M. Beetz. Learning task outcome prediction for robot control from interactive environments. In *2014 IEEE/RSJ Inter-*

national Conference on Intelligent Robots and Systems, pages 4389–4395, 2014.

[5] A. Haidu, D. Kohlsdorf, and M. Beetz. Learning action failure models from interactive physics-based simulations. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5370–5375, 2015.

[6] T. Hasegawa, T. Suehiro, and K. Takase. A model-based manipulation system with skill-based execution in unstructured environment. In *Fifth International Conference on Advanced Robotics*, pages 970–975, 1991.

[7] L. Kunze, A. Haidu, and M. Beetz. Acquiring task models for imitation learning through games with a purpose. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 102–107, 2013.

[8] Carol L. Novak and Steven A. Shafer. Anatomy of a color histogram. In *CVPR*, 1992.

[9] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, John Schulman, Emanuel Todorov, and Sergey Levine. Learning complex dexterous manipulation with deep reinforcement learning and demonstrations. *ArXiv e-prints*, (arXiv:1709.10087), 2017.

[10] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

[11] Akihiko Yamaguchi. Baxter peels banana. `https://youtu.be/rEeixPBd3hc`, 2016. [Online; accessed Oct-28-2016].

[12] Akihiko Yamaguchi and Christopher G. Atkeson. Differential dynamic programming for graph-structured dynamical systems: Generalization of pouring behavior with different skills. In *the 16th IEEE-RAS International Conference on Humanoid Robots (Humanoids'16)*, 2016.

[13] Akihiko Yamaguchi and Christopher G. Atkeson. Neural networks and differential dynamic programming for reinforcement learning problems. In *the IEEE International Conference on Robotics and Automation (ICRA'16)*, 2016.

[14] Akihiko Yamaguchi, Christopher G. Atkeson, and Tsukasa Ogasawara. Pouring skills with planning and learning modeled from human demonstrations. *International Journal of Humanoid Robotics*, 12(3):1550030, 2015.

[15] P. C. Yang, K. Sasaki, K. Suzuki, K. Kase, S. Sugano, and T. Ogata. Repeatable folding task by humanoid robot worker using deep learning. *IEEE Robotics and Automation Letters*, 2(2):397–403, 2017.