Rotating Cloned Task-Space Trajectories for Efficient Robotic Manipulation

Travers Rhodes and **Manuela M. Veloso**
Robotics Institute, Carnegie Mellon University, USA
{traversr, mmv}@cs.cmu.edu,

Abstract

Robots with manipulation skills acquired through trajectory cloning, a type of learning from demonstration, are able to accomplish complicated manipulation tasks. However, if the skill is demonstrated on one arm and applied to a different arm with different kinematics, the cloned trajectory may not be well matched to the new robot's kinematics and may not even be feasible on the new robot. Additionally, even if the new skill is demonstrated and applied on the same robot, the demonstrated trajectory might be feasible in the training configuration, but may not be feasible under desired translations of the trajectory, since the translated trajectory might go outside the robot's workspace. For some tasks like picking up a tomato slice, rotations of the trained trajectory would be successful in accomplishing the task. In this paper, we address how a robotic system can use knowledge about its own kinematic structure to rotate trained trajectories so that the new trajectories are feasible and lower cost on the robotic system, while still mimicking the trajectories defined for the robot by the human demonstrator.

1 Introduction

One of the current difficulties in manipulating food is the lack of an accurate model for deformable food. For example, consider the task of plating a caprese salad by picking and placing alternating slices of tomatoes and cheese using a single fork. One strategy for performing the acquisition of slices is to pierce each slice using a vertical approach, tilt the fork to a horizontal position (bending the tomato in the process), and lift the fork with the prongs held horizontally. The middle step of this complicated robotic manipulation is shown in Figure 1, where the fork is being tilted to a horizontal position before being lifted off the blue cutting board. In the example shown, the tomato slice remains on the fork when lifted vertically. We note that in this example, holding the fork prongs vertically (instead of horizontally) releases the tomato from the fork. This has two ramifications. First, we can use this behavior to deposit the tomato slice at the desired location by moving the fork to the desired location before tilting the fork prongs to vertical. Second, this manipulation problem with a ripe tomato slice is complicated enough that vertical skewering and vertical carrying is not sufficient.

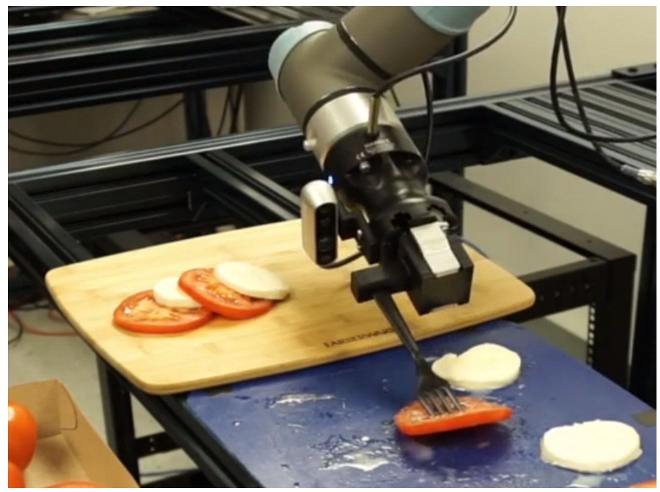

Figure 1: UR5 Robot performing a complicated tomato slice acquisition

While it would be possible to use a more complicated end-effector design to simplify the manipulation process, we are interested in using a multi-purpose tool like a fork that can be also used for other tasks as well. Additionally, we are specifically interested in complicated manipulation strategies for deformable, pierceable, and bendable objects because these manipulation strategies are not as well understood.

Without a deformable, pierceable, bendable physics model of a tomato slice, it is not possible for an learning algorithm to develop this type of pickup strategy in simulation. Instead of having the robot learn this pickup strategy from scratch, we give the robot a single demonstration trajectory and have the robot mimic the training trajectory as closely as possible. This is a type of behavioral cloning, in which a robot learns a policy that imitates training trajectories without learning a reward function [Osa *et al.*, 2018]. We find that behavioral cloning is able to learn to accomplish the desired task using only a single demonstration in the real world. This suggests the power of behavior cloning for manipulation tasks of hard-to-model objects like food.

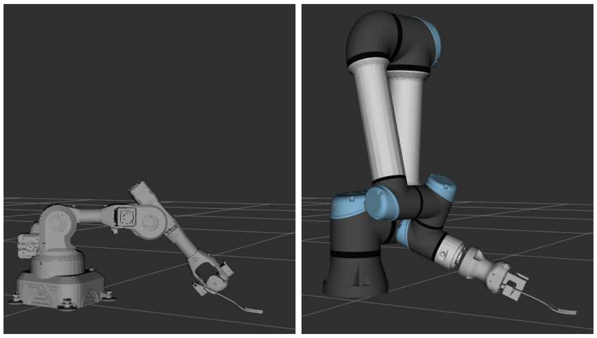

Figure 2: The fork can be translated directly toward the base of the Niryo One robot pictured on the left, but the resulting fork positions would lead to self collision on the much larger UR5 robot pictured on the right.

We implement our behavior cloning of a single demonstration in the following way. First, a training trajectory is recorded on a small Niryo One robot [Niryo, 2019] with a fork, and the trajectory of the fork tip is computed in task space based on joint angle recordings. Second, a vision system identifies the location of a tomato slice relative to the base of a UR5 robot. Finally, the task-space trajectory is translated to align with the tomato slice, and we use a continuous inverse kinematics solution to give target joint positions for the UR5 robot to follow. We find that the recorded trajectory is often able to successfully pick up the tomato slice.

However, one downside of this implementation of direct behavioral cloning is the lack of modification to account for the kinematics of the robot. In particular, exactly following the demonstration trajectory may not be feasible for certain tomato slice locations with the given robot kinematics, or may lead the robot to travel through awkward joint trajectories including possibly near singularities with high joint velocities. Figure 2 shows how some reasonable fork poses on a Niryo One robot would lead to self-collisions on the larger UR5 robot.

In order to make the power of behavioral cloning more flexible, we present a search-based method to consider rotations of the original trajectory in order to minimize cost of the trajectory for the particular robot kinematics.

2 Related Work

There are multiple current commercial approaches to robotic picking and placing of food that do not require complicated manipulation strategies. Most of these approaches use custom grippers designed for specific types of food. For food slices, the robot can slide support surfaces underneath the food from both sides, and stabilize the slices from above. This principle is used in the commercially available "Meat Gripper" built by Applied Robotics [AppliedRobotics, 2019], which can pick up and deposit sliced meats and cheeses. A surface attractive gripper for food slices, in particular for tomato and cucumber slices, is developed in [Davis *et al.*, 2008]. It is based on the Bernoulli principle, and shows great promise for food picking, similar to the success seen by suction grasping for packaged items.

Finally, forks have previously been used successfully by [Gallenberger *et al.*, 2019; Herlant, 2018] to pick up food and bring it to a user's mouth for autonomous feeding tasks. In those works, a linear skewering motion (either vertical or angled) was used to successfully pick up the food and bring it to the user's mouth. In this work, we consider behavioral cloning to define the pickup trajectory, and consider the associated question of how to fit a cloned trajectory best to the robot arm kinematics.

3 Approach

We want to modify the demonstrated trajectory to better fit the kinematics of the robot. For this work, we allow the robot to modify the demonstrated trajectory by rotating the trajectory around a vertical axis centered at the tomato slice. We allow this type of rotation as we assume that those rotated trajectories will have an equal success rate in picking up rotationally symmetric objects. Our goal is to find rotated trajectories that have the lowest cost to perform.

Our algorithm works by taking the desired trajectory to replicate and rotating it around a vertical line passing through the center of the tomato slice. We perform a grid search over N different rotations, evenly dividing the possible rotation angles in $[0, 2\pi)$. For each of these trajectories, we use an analytic inverse kinematics solver [Diankov, 2010] to provide possible starting joint configurations at the start of the rotated trajectory. We then plan through the trajectory and check if that trajectory is collision free. We use MoveIt! [Sucan and Chitta, 2013] to easily handle collision checking, including checking whether the trajectory would collide with the table that the robot arm is mounted on.

We performed our experiments on a UR5 robot which has 6 degrees of freedom. Therefore, given a desired trajectory for the fork tip pose, and an initial joint configuration at the start, the joint space trajectory of the UR5 robot is uniquely defined. The above only holds so long as the robot does not hit a joint singularity, but we note that the set of exact joint singularities within the workspace has probability zero for feasible trajectories in task space.

If a trajectory is collision-free, we compute for that trajectory the cost of the trajectory. For this work, we used the Euclidean length of the joint-space path as the cost of the trajectory. This encouraged the robot to choose paths that reduced the overall distance traveled by the joints, which in turn reduces the average speed that the joints would need to travel. We consider our cost function to be a measure of the "comfort" of the robot in performing the trajectory. Other more complicated cost functions could include penalty terms for the static torque on the robot joints due to gravity throughout the trajectory or dynamic terms that penalize joint acceleration during the trajectory.

4 Results

We find that this approach was able to perform the expected optimizations. We consider a recorded trajectory for moving a fork down, rotating it to pick up a tomato slice (Figure 1), and lifting it. If we do not rotate the recorded trajectory, the fork prongs point in the negative X direction when lifting the

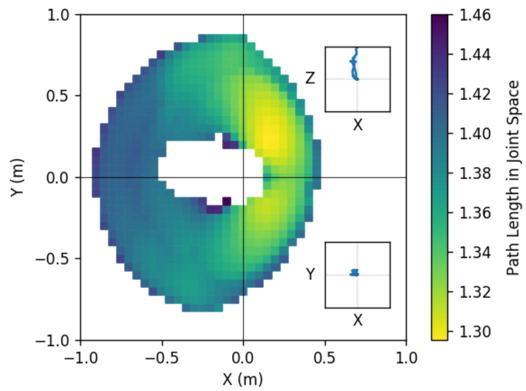

Figure 3: For the given X-Y location on the table, we plot the minimal joint space path length required to perform the recorded trajectory at that location. The recorded trajectory points the fork prongs in the negative X direction. The insets show to scale the fork tip path during the pickup trajectory. The base of the robot is at (0,0). No color means that the resulting trajectory is not feasible (either extends outside the reach of the robot, or results in collision).

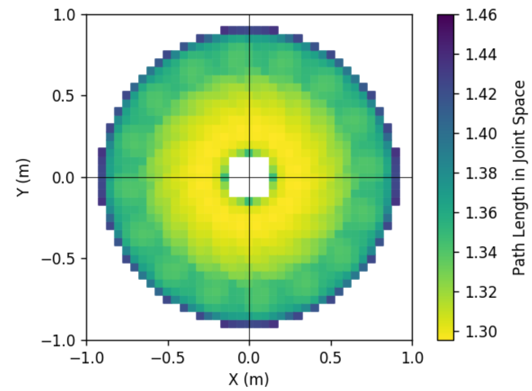

Figure 4: For the given X-Y location on the table, we plot the minimal joint space path length required to perform the recorded trajectory (rotated by some multiple of $\pi/8$ radians) at that location. The base of the robot is at (0,0). No color means that the resulting trajectory is not feasible (either extends outside the reach of the robot, or results in collision).

tomato. The cost of that unrotated trajectory, for different locations of the tomato slice, relative to the base of the robot at (0,0), is shown in Figure 3.

When we do not allow for rotations, we find many locations for the tomato slice where the robot is not able to perform the desired trajectory because the resulting trajectory would head outside of the robot's workspace, or would lead to a self-collision. For example, in Figure 3, the infeasible area directly to the left of the origin is due to self collisions for poses similar to that displayed on the UR5 in Figure 2.

However, when we do allow for the robot to change the behavioral cloned trajectory by rotating the trajectory, we expand the effective workspace of the robot and reduce the cost of performing the trajectory. Additionally, since the UR5 robot's first joint allows for rotations around the vertical axis, we expect that if we allow vertical rotations of the target trajectory as well, then the resulting feasible workspace should also be rotationally symmetric around the vertical axis centered at the robot center. In particular, if we moved the target tomato-slice location in a circle centered at the robot base, we expect the robot to be able to simply rotate the solution trajectory a corresponding amount and find a solution to the new target location with the same cost. For computational efficiency, we discretize rotations of the original trajectory by a multiple of $\pi/8$ (dividing the circle into 16 components). Accounting for this approximation, we find that the resulting costs do indeed appear to be rotationally symmetric.

We find that the robot can rotate a recorded motion to effectively increase its workspace in order to perform the given task. We also note that in locations that were part of the workspace without rotation, the robot can still choose to rotate the trajectory in order to reduce the cost of performing the trajectory. Using our approach, a new robot with new kinematic properties can efficiently figure out a good rotation of the original trajectory to use to complete the assigned task.

5 Conclusion and Future Work

In this work, we leverage the effectiveness of behavioral cloning for food acquisition tasks. We show how the robot is able to apply rotations to the demonstrated trajectory to reduce the cost of performing that trajectory and to expand the set of feasible trajectories.

In future work, we hope to address additional, more complicated perturbations of the demonstration trajectory, including adding an allowable margin around the demonstration trajectory. That is, if we give the robot some epsilon bound where it only needs to stay within some distance ϵ of the trajectory in task space (instead of matching the trajectory points exactly), we hope to see how much that constraint relaxation will increase the workspace of the robot and the efficiency of the robot performing the new trajectory.

References

[AppliedRobotics, 2019] AppliedRobotics. Applied robotics. https://www.appliedrobotics.com, 2019. Accessed: 2019-05-07.

[Davis *et al.*, 2008] S Davis, JO Gray, and Darwin G Caldwell. An end effector based on the bernoulli principle for handling sliced fruit and vegetables. *Robotics and Computer-Integrated Manufacturing*, 24(2):249–257, 2008.

[Diankov, 2010] Rosen Diankov. *Automated Construction of Robotic Manipulation Programs*. PhD thesis, Carnegie Mellon University, Robotics Institute, August 2010.

[Gallenberger *et al.*, 2019] Daniel Gallenberger, Tapomayukh Bhattacharjee, Youngsun Kim, and Siddhartha S Srinivasa. Transfer depends on acquisition: Analyzing manipulation strategies for robotic feeding. In *2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, pages 267–276. IEEE, 2019.

[Herlant, 2018] Laura V Herlant. *Algorithms, implementation, and studies on eating with a shared control robot arm*. PhD thesis, Carnegie Mellon University, 2018.

[Niryo, 2019] Niryo. Niryo one. https://niryo.com/niryo-one, 2019. Accessed: 2019-05-07.

[Osa *et al.*, 2018] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel, Jan Peters, et al. An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics*, 7(1-2):1–179, 2018.

[Sucan and Chitta, 2013] Ioan A Sucan and Sachin Chitta. Moveit! *Online at http://moveit. ros. org*, 2013.