Project Proposal

Bartłomiej Cieślar, Michael Zeng

November 2, 2023

Motivation

Humans manage to throw and catch objects with relative ease; it's often the fastest and most energy-efficient way to move an object from point A to point B. However, catching flying and free-falling objects is a pretty big challenge. The objects may only be in the air for a few hundred milliseconds, leaving very little time to correct the robot's motion if the prior estimate of the object's trajectory was noisy. The shape and size of the object may also necessitate intercepting it from a certain direction. Therefore, the motivation for this project is to tackle the challenge of dynamic object handling for robotic manipulators using mainly just the techniques we have learned in class.

This project is also in part inspired by Shane Wighton's Moving Basketball Hoop on YouTube [5] and Mark Rober's Viral Moving Dartboard [3].

Project

Baseline Deliverable

We propose a project of a robot catching objects in flight based on feedback from RGB-D cameras observing the scene. Our initial goal will be to catch spherical objects thrown at the robot with random trajectories from any direction. To achieve this, we devised a method, which consists of the following two parts (which will run continuously in parallel):

1. **Trajectory prediction** - the first part of the project predicts the trajectory of the flying object. In its simplest form, it will consist of a 2 stage pipeline. First, our algorithm will use the Iterative Closest Point method we learned in class to find the object's position at certain key-frames from point clouds. Then, the algorithm will perform some sort of heuristic (most likely the RANSAC algorithm learned in class and extended with weighted averages in [4]) to extract the ballistic trajectory of the moving object. The output of this part of the project are parameters of the ballistic trajectory (in the form of spatial pose and velocity at time $t = 0$). Bartek is in charge of this half of the project.

2. **Catching** - the second part of the project uses the output of the first part to find the best way to catch the object. More specifically, we will first use the Darboux Frame sampling method touched on in class to compute a few potential grasp poses in object frame, which we will transform into world frame according to the object's trajectory. Then, we will plan an optimal trajectory for the arm's motion, with constraints that the arm must arrive at a grasping pose at the precise time the object will arrive there, with velocity equal to the object's velocity, such that the orientation of the gripper in the grasping pose puts the object in between the fingers of the gripper, and such that the arm doesn't collide with the object while its flying. Finally, we will append a second trajectory onto the optimized trajectory that will close the grippers and slowly decelerate the object along the object's ballistic path. Michael is in charge of this half of the project.

Extensions

It is likely that we will achieve the base deliverable before the deadline for the project. For this reason, we define some extensions to the project:

- Improving the catching optimization algorithm [6] (see the Related Work section for details) by doing a lower-dimensional optimization or performing a hybrid sampling and optimization technique.
- Modelling camera noise [1] and performing probabilistic trajectory prediction [4] to make the simulation model the real world better.
- Placing/dropping objects in a bin after they are caught.
- Catching objects that have a more complex geometry (e.g. YCB objects).
- Catching objects that are rotating (potentially very quickly) mid-air.
- Capturing the information about the object purely from observations without prior knowledge of the geometry of the object thrown (or which specific geometry that is).

Related Work

There have been previous works on dynamical object handling. The method in [2] utilizes a 7 DoF arm and Allegro gripper also to catch objects, and uses learning-based methods and probabilistic models to simultaneously predict object trajectory and control the robot's position, which presents a fairly different approach than our primarily optimization-focused method. In [7], a reinforcement learning with residual physics is used to achieve dynamic tossing movements, which presents yet another different approach.

We also investigated a number of related works that present methods that are similar or build upon the concepts we learned in class. Previous work discusses techniques for improving both speed and performance of trajectory-planning [6]. Firstly, they sample multiple initial predictions for the optimizer to better performance on local minimums. They also use a repeated method of lower-dimensional optimization and interpolation into higher dimensions to decrease the time taken by the optimization while improving the size of the search space (according to the paper). Both are possible extensions for our project, although we hypothesize that the random sampling may not help much because the constraints on the trajectory optimization are relatively simple to begin with.

There has also been previous work related to modelling noise in certain kinds of depth cameras. This is necessary for us to model, since the default depth camera output in Drake is perfect and does not allow us to simulate uncertainty. In [1], the authors consider modelling the noise in structured light depth cameras. They identify that the most notable source of noise in such cameras is quantization noise, the ways to model that noise, as well as methods of dealing with this issue by de-noising.

Some work also considers dealing with uncertain object positions and constructing trajectories around that, which is also necessary for the probabilistic trajectory prediction extension. In [4] the authors consider a modification to the classical RANSAC algorithm that is able to deal with pose uncertainties. However, the method only considers the first-order Taylor Series approximation of object's movement, so some work will be required to extrapolate that into ballistic trajectories.

Timeline

Deadline	**Minimum deliverables**
First check-in (15th Nov)	Environment ready and running with cameras generating point clouds
Second check-in (29th Nov)	The base method implemented and successfully catches simple objects/throws.
Final deliverable (12th Dec)	One or two of the extensions implemented.

References

[1] Avishek Chatterjee and Venu Madhav Govindu. Noise in Structured-Light Stereo Depth Cameras: Modeling and its Applications, May 2015. arXiv:1505.01936 [cs].

[2] Seungsu Kim, Ashwini Shukla, and Aude Billard. Catching Objects in Flight. *IEEE Transactions on Robotics*, 30(5):1049–1065, October 2014. Conference Name: IEEE Transactions on Robotics.

[3] Mark Rober. Automatic Bullseye, MOVING Dartboard, March 2017.

[4] Tal Nir and Orit Eden. Video motion detection algorithm using probabilistic time integrated RANSAC. In *2011 18th IEEE International Conference on Image Processing*, pages 2349–2352, September 2011. ISSN: 2381-8549.

[5] Stuff Made Here. Moving hoop won't let you miss, 2020.

[6] Alessandro Tringali and Silvio Cocuzza. Globally Optimal Inverse Kinematics Method for a Redundant Robot Manipulator with Linear and Nonlinear Constraints. *Robotics*, 9(3):61, September 2020. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.

[7] Andy Zeng, Shuran Song, Johnny Lee, Alberto Rodriguez, and Thomas Funkhouser. TossingBot: Learning to Throw Arbitrary Objects With Residual Physics. *IEEE Transactions on Robotics*, 36(4):1307–1319, August 2020. Conference Name: IEEE Transactions on Robotics.