Spatial target estimation and robot arm workspace control implemented on Baxter robot

Yuan Zhou
Boston University College of Engineering
110 Cummington Mall, Boston, MA
zhouyuan@bu.edu

Abstract

In this term project, a visual servoing control method is introduced. The project can be divided into two sub contents which are static spatial target position estimation using two views geometry and robot arm end pose workspace control using position-velocity Jacobian method to approach the target. The relationship is that after gotten the estimated target, we use this information as the goal point of followed arm control algorithm. The features of this project are local precise position measurement and a workspace arm control algorithm created by the auther. The approach is validated by implementation on the Baxter robot.

1. Introduction

Visual servoing (VS) is defined as the use of visual feedback mechanisms for the kinematic control of a robot. In visual servoing, we can divide the algorithms into two parts: the first apart is object position based on visual inspection and the second part is related arm control strategy.

For the first part, we can further divide the algorithms into two parts: vision-based sensor arrangement choice and detection method choice. There are two choices for vision-based sensor arrangement: eye-to-hand arrangement and eye-in-hand arrangement. For the first one, the sensor is arranged outside the working space and sense the whole working space, which usually three-party sensor is being used like Kinect sensor. Several relative works have done based on this kind of sensor arrangement. In [11] Xinyu arranges Kinect sensor on the front upper side of the Baxter robot to get the 3D information of whole robot front view and surrounding environment to do further object self-identification. The camera in [6] is set on the top of the robot to get the 2D top view of the industrial robot arm and a potential function of the robot contour is gotten through the image. For [3] images taken from several stationary cameras in the work cell is presented and general a three views like image system to represent the robot arm and surrounding environment. Eye-to-hand arrangement can provide a large enough view angle and suitable for whole robot though there will have some issues of accuracy due to the resolution of the sensor. Another important issue for the eye-to-hand method is that because of three-party sensors the relative position measurement between the sensor frames and robot frame may become a problem.

For eye-in-hand arrangement, the sensors are mounted on the end-effector of the robot. Specified in Baxter robot, embedding RGB-camera in the robot arm end pose can be used. Sten [7] uses Baxter arm camera to tracking object in horizontal version. Not only in Baxter robot, in [9] a Kinect camera is installed on the six-legged robot which is similar as eye in hand sensor arrangement. The advantages of eye-in-hand arrangement are providing higher resolution information of the object and coordinate systems are much easier to translate between each other. Eye-in-hand method is chosen in this paper because of local precise position estimation reuquirement which means the whole robot work space is not we are really looking for but the accuracy of the particular position measurement is really important.

For detection method choice, most of the relative works using color-based detected method to segment the robot or object from the environment [6, 3, 9]. The reliability of color-based detected method is highly depended on the surrounding environment, the object to be detected and sensor property. Particular image processing methods may need to used depend on parameters presented above. In [7] the author does not choose detect object directly but use attached ARtag on the object to get the relative position of the tag with respect to the camera and finish the tracking movement. However, ARtag tracking method is basically tracking the ARtag not the object and the object needs to have a flat surface to place the tag. In this paper, color-based detected method is selected.

The other part is robot arm workspace movement generation. Unlike the vision part, this part is better to have a real system to fulfill the algorithm and in this term project the Baxter robot is chosen. The goal of this particular movement is to control the Baxter right arm gripper to approach the measurement target based on the position estimation. This movement can be further model as a workspace(w-space) control which means find the relationship between the designed w-space trajectory and the configuration space(c-space) joint angles and this relationship can be realized from the level of position, first order derivative(velocity) and second order derivative(acceleration). In detail, for position level, inverse kinematic process is needed to get the related joint angles and input them to the motors to move each joints. For velocity level, It is based on the Jacobian matrix. Notice that the Baxter robot arm itself has 7 degree of freedom(DOF), so a redundant manipulator is its kinematic model. The most direct method is using Moore-Penrose pseudo-inverse of the Jacobian matrix. Whitney[10], in his pioneering work on resolved-rate control, proposed this technique. Although it still has drawback of avoiding kinematic singularities which pointed by Baillieul[1] and lack of repeatability which pointed by Klein and Huang[4], this solution is quite attractive because the pseudo-inverse has a least squares property that generates the minimum norm joint angles. Generated from the Moore-Penrose pseudo-inverse, gradient projection method is another approach to solve redundancy. It adds the particular solution which is the result of Moore-Penrose pseudo-inverse with a homogeneous solution which can projects an arbitrary vector into the null-space of the Jacobian matrix. The vector usually be choosing by a gradient function to optimize different properties but does not affect the end pose result. The third technique is called task space augmentation which conceptually different from the above two methods. It adds a constraint task on the joint variables. However, to find the constraint task is not an easy job, Baillieul[2] suggested to use the extended Jacobian technique. In this term project direct Moore-Penrose pseudo-inverse method is chosen and a method between the level of position and velocity control is introduced mainly due to the base application programming interface (API) limitation of the Baxter robot.

2. Methodology

2.1. Spatial target estimation

Problem formulation In this term project, the target object is a tennis ball which has a radius of 0.03 meter. The reason is that the object color is easy to separate from the surrounding captured environment and has a circle contour. For Baxter arms, the right one is the main moving arm and left one the the wingman to finish the vision part. In this project we have the D-H table of the right arm and have the access API for control both joint position and velocity.

Mass point spatial position based on visual inspection General methods used in this part are color-based pattern recognition and classic two-view geometry using linear triangulation and single value decomposition technique.

All following vision algorithms are based on the information which be gotten from the camera image. The technique of pattern recognition is introduced in the experimental setup section. The basic idea is that we want to recover or estimate the coordinate information of a spatial mass-point from related planar camera pixel image frames. Due to the fact that we only use the RGB-camera to detect the object, we can not recovery all the coordinates information of the mass-point using only its own information from one camera because of Single-view ambiguity showed in Figure 1. We can fix this problem by either known part of the mass-point information in advance or use multi-view geometry showed in Figure 2.

The first part of two-view geometry is based on the classic pinhole camera model which in detail is the camera forward projection mapping shown in the following equation

$$x = KR[I| - C]X$$

In which K is the camera intrinsic matrix which is related to the camera CMOS sensor property. It can be viewed as the transform between the CMOS plane and the image plane. R and C are the rotation and transformation part of the extrinsic matrix which is just a different name of transform matrix in computer vision area and it contains the relationship between the base frame and the camera image frame. x is the image pixel coordinates gotten from the pattern recognition and finally the X is the spatial mass point. What we interested is the inverse process of the inverse process and it has been shown that it can not be done with only one camera because of projection mapping. Therefore, the other arm camera comes to the party to generate another forward projection mapping and merges these two mappings to finish the inverse process. However, in this case, the system becomes overestimates which means it uses four inputs to solve three outputs. Mathematically, an overestimated homogeneous equation sets need to be solved. In this project, single value decomposition (SVD) is used to solve this equation set. The performance of the vision part is crucial for the whole process because it directly decides the goal position of the following motion part and also in this project, there is not feedback control when do the motion part, so a check strategy is needed to assess the estimation performance. The back projection checking precess is introduced as follow: used the estimation coordinates and do the forward projection mapping in each arm camera to generate two parts(four datas) of the back projection image pixel coordinates and compare them with the pattern recognition results. It can be checked both directly from the image or from the data. The following pictures in the experimental

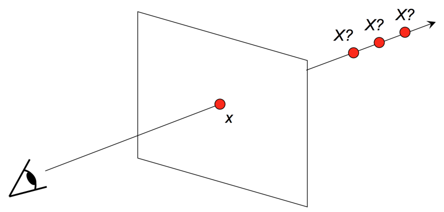

Figure 1. Single-view ambiguity

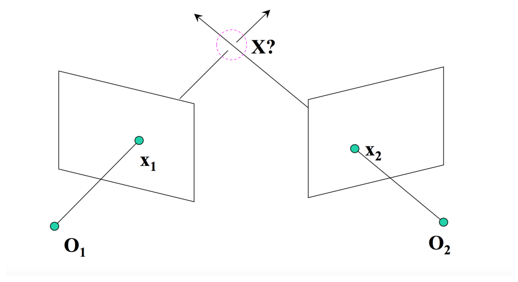

Figure 2. Two-views geometry

setup show the back projection performance. We claim that the result is a local precise estimation which can be use as the goal of the following motion part.

2.2. Robot arm workspace control

Problem formulation If uses plain language, the motion part is like the following lines: Now our Baxter opens its 'eyes' and can 'see' something. We want to use its arm to move to it. Maybe readers will confuse that is it a real difficult action or not? Actually it is a good question because in our human life this is kind like a 'every second' action which means we master it and do it unconsciously. However, for robot arms whether it is easy or not. A math model has to generate to describe this action. The math model of the motion process is that Known the end pose start and target position also D-H table of the robot arm generates the trajectory relate to the end pose in the work space and maps back to the configuration space to control the movement.

Methods detail and choice There are three methods to do this action position inverse kinematic control, position jacobian velocity control and pure jacobian velocity control. The basic information of the first and the third one are given in the introduction and detail can be seen from the references. For this project, I generate a control strategy between this two levels of control which is the second one(position jacobian velocity control). The position jacobian velocity control is actually still a position control because the output data still go through the joint angle API of the Baxter, but with a particular trajectory. The following one line algorithm show the strategy.

$$\theta_{nexttimestep} = \theta_{now} + J^{+}\dot{x}$$

in which the whole movement process is divided into many tiny time steps in each time step the difference of the joint angles between now and next time step is given by the $J^{+}\dot{x}$ in which the J^{+} is Moore-Penrose pseudo-inverse and the $\dot{x}$ is a given w-space trajectory velocity.

Compared to the position inverse kinematic control, the advantage of this strategy is that direct inverse kinematic solving precess is ignored and only differential forward kinematic (Jacobian) and inverse of Jacobian need to be solved.

Compared to the pure velocity control, the advantage in terms of this Baxter case is that more stable position API can be used. The detailed reason can be shown in the experimental setup.

However, a pre-designed w-space trajectory is needed to generated and it depends on the motion users provided. For this particular motion the trajectory is given in the following experimental setup.

3. Experimental setup

3.1. General idea

Experiment and simulation are necessary ways to test algorithms. For this term project it is not a theoretic paper, the experiment and simulation should play a more important role. All the following contents are performed in real Baxter and the Baxter gazebo& Rviz simulator based on robot operation system(ROS) in linux ubuntu 14.04 main environment. The whole process is followed the order of methodology and several key points are shown.

3.2. ROS

Our approach uses ROS, which works under Linux and is the default platform for the Baxter robot to work as a middleware for interconnecting all the elements in the proposed system. The communication element in ROS is called "node" which is the ROS term for an executable that is connected to the ROS network. In each node there could have "publishers" and "subscribers" to output and input the information over ROS "topic" element.

In this term project, Four Python files are created related to the process, two for arms image processes, one for position estimation and one for motion.

3.3. Vision

OpenCV OpenCV(Open Source Computer Vision Library) is introduced to use in video image process. The color-based pattern recognition is very popular so it does

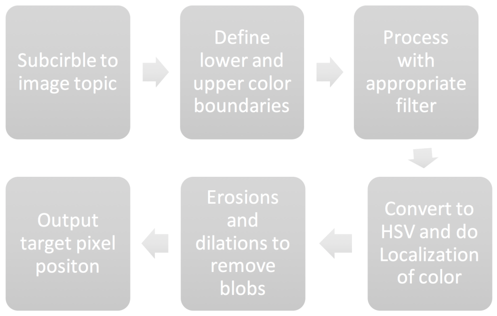
Subcirble to image topic
Define lower and upper color boundaries
Process with appropriate filter
Convert to HSV and do Localization of color
Erosions and dilations to remove blobs
Output target pixel positon

Figure 3. General color-based image process

not appear in the methodology. General color-based image process is showed in Figure 3.

Generally, color-based detection is sensitive with the surrounding light environment. Through several experiments, to get a stable output, a bright and well-lit environment is necessary.

Had appropriate color boundaries is very crucial for getting a right object contour. However, due to the different experimental environment and the particularity of Baxter hand camera , it is hard for us to find a 'constant' color boundary range like pre-experiment using laptop webcam. One feasible method is redefine the range at the start of each experiment. Color boundary used in this experiment is showed in following table.

	$Lower Boundary$	$Upper Boundary$
R	29	84
G	65	175
B	47	255

Baxter hand camera generates relative large noise compared with the laptop webcam, so it is necessary to add filter to have a better output which is sometimes no need for webcam. In general sense of the term "filtering", the image goes through a low pass filter and gets blur to remove the high frequency information which is the detail information of the image. To remove the noise, the value of the filtered image at a given location is a function of the values of the input image in a small neighborhood of the same location, so after filtering, the image resolution will decrease. Generally, gaussian blur is the most popular image blur method. In this case, we would like to keep the edge information when we removed the noise, however, gaussian blur does not preserve edges in the input image. Bilateral filter which is an edge-preserving smoothing filter is be chose. Bilateral filter which first introduced by Tomasi [8] removes fine texture of the image but the overall shading is preserved. Compared with the Gaussian filter, one more (multiplicative) Gaussian filter component which is a function of pixel intensity differences is merged with the original filter.

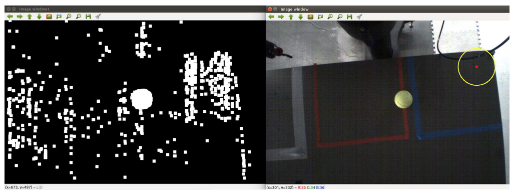

Figure 4. Output of the general process

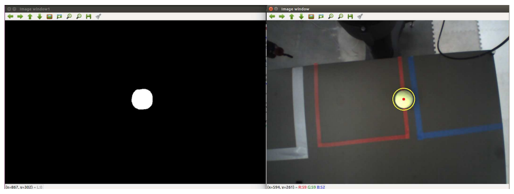

Figure 5. Output of the customized process

The build-in function from OpenCV called cv2.bilateralFilter is used in our experiment.

$$cv2.bilateralFilter(src, d, sigmaColor, sigmaSpace)$$

There are four parameters that we can adjust. src: the source image. d: the Diameter of each pixel neighborhood that is used during filtering. sigmaColor: the filter sigma in the color space. sigmaSpace: the filter sigma in the coordinate space. Based on OpenCV docs, it is recommended to use $d = 5$ for real-time applications. For sigma value, 2 sigmas values can be the same. If they are small (less than10), the filter will not have much effect, whereas if they are large (larger then 150), they will have a very strong effect, making the image look cartoonish.

For step five, in Baxter camera environment, contour of the ball cannot be detected fully sometimes mainly because of the color range and this phenomenon will cause the joggle of the target point which is the center of the tennis ball, so we remove the erosion function in order to enlarge the detected pixels to cover the undetected pixels which inside the contour.

After above processes, the performance of the detection is much better than the general one. Figure 4 and Figure 5 show the difference. For each figure, the left one is the HSV image which contains the information (white part) to output the contour of the object and the right one is raw image with marked output center coordinates and radius. Figure 4 shows the output used general image process and Figure 5 shows the output with customized image process based on above analysis.

Performance of the position estimation Figure 6 and 7 show the back projection test for both left and right arm. The red points are the pattern recognition results and the

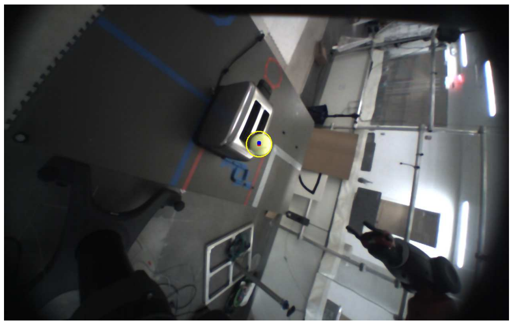

Figure 6. Left arm back projection test

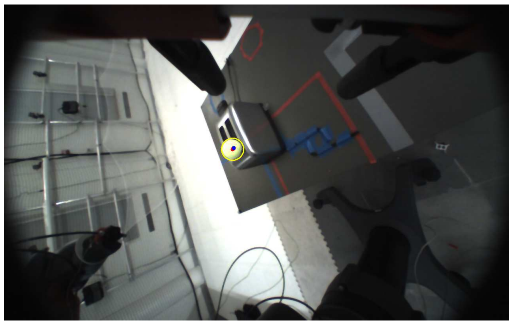

Figure 7. Right arm back projection test

blue one is the back projection results. It can be show that under this circumstance the result is precise enough to continue to motion part.

However, this is a 'local' precise estimation which means if the target appears in the corner of any image frame, the performance becomes very unstable and can not be used. Although, for this term project goal, this issue can be 'solved' if we adjust the camera views before the vision process, it blocks the dynamic tracking road which real-time robot arms tracking performance can not be done in this way.

3.4. Motion

Followed the methodology part. Actually for the first method(position-IK control), Baxter provided API $.move\_to\_position()$ and its own numerical IK solver do a great job. No better improvement can be done in terms of the time of the term project. However, this is still a good choice to test the vision result and compared with our own control law.

For the pure velocity control, actually it was first be chosen because of its relationship between the class material. Unfortunately, there is no zero-gravity mode in the velocity control mode.

Zero-graivty mode is a Baxter embedded underlying control strategy implemented on the hardware level in order to 'erase' the self arms' gravity effect which will affect the movement design. In detail, it uses springs and damping in motors to 'pull' the configuration. It can be seen from[5].

When switch to the velocity mode, this zero-gravity mode deactivates at the same time, so self gravity effect needs to be considered to merge it to the main trajectory control. It is believed that this relate to the dynamic of the robot manipulators. Consider the time limitation of this term project, this candidate does not be chosen.

For our own control algorithm, due to the mass point property of target tennis ball, there is no information about the orientation, so for the trajectory design, the orientation of gripper movement is given for free. In order to make the precess stable, A constant gripper orientation and straight line trajectory is been chosen. In detail, lock the orientation of the gripper related to the base frame in the whole process and the workspace trajectories for each X, Y, Z direction is a straight line with constant velocity. Therefore, an adjustment of the right arm pose between the vision and motion is needed, because it will be strange to use IK to set orientation of the end pose of the robot arm but we do like to set the orientation vertical a particular pose in order to design and understand the w-space trajectory easily. We approximately set one joint angle (w0) to zero based on the D-H table and Baxter arm configuration (seen from figure 8) though there will have some bias, the tolerance is still acceptable. For the Jacobian calculation, in this term project Baxter provided method Pykdl.Jacobian is used.

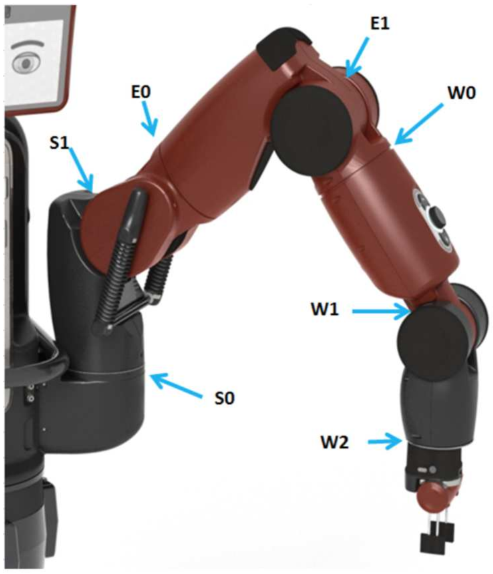

Figure 8. baxter joints

4. Results

In this section, we validate our method through simulation on Baxter robot and Baxter simulator, through several

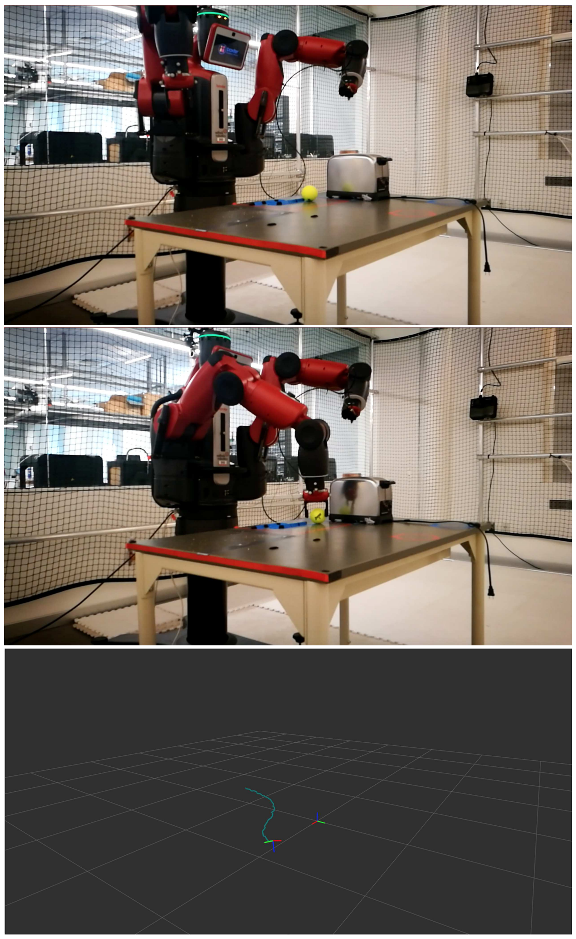

Figure 9. Experiment with target on a table

experiments Figure 9 and Figure 10. In each figure the first and second image are the start and the end configuration of Baxter end pose. The third image is the simulation from Rviz to show the particular trajectory and two coordinates are base frame and right gripper frame. The whole process successful shown. Baxter robot estimate a spatial target position and use our own control strategy to approach the target.

5. Conclusions

In this term project, a vision servoing approach is given with local precise mass position estimation and own position-velocity control law is created to approach the target. A real Baxter experiment is shown to validate the whole process and also a simulation is given to show the trajectory.

However, this is not a dynamic tracking performance. Along with the limitation of the position estimation mentioned before, the arm camera refresh rate also has limitation to perform real time tracking. A third party high speed camera could be a solution if the extrinsic matrix can be measure precisely.

All in all, this term project provided a deep understanding of the robot manipulation kinematic and an invaluable working experience with a real robot system not just on a theoretical or simulation level.

References

[1] Baillieul. Programming and control of kinematically redundant manipulators. *Decision and Control, 1984. The 23rd IEEE Conference*, pages 768–774, 1984.

[2] Baillieul. Kinematic programming alternatives for redundant manipulators. *IEEE Robotics and Automation, 1985.*, pages 722–728, 1985.

[3] D. M. Ebert and D. D. Henrich. Safe human-robot-cooperation: Image-based collision detection for industrial robots. *IEEE/RSJ International Conference On Intelligent Robots and Systems*, pages 1826–1831, 2002.

[4] H. Klein. Review of pseudoinverse control for use with kinematically redundant manipulators. *IEEE Transactions on Systems, 1983.*, pages 245–250, 1983.

[5] R. Robotics. Worked example visual servoing. `http://sdk.rethinkrobotics.com`, 2015.

[6] K. Stefan and H. Dominik. Fast vision-based minimum distance determination between known and unkown objects. *Intelligent Robots and Systems*, pages 2186–2191, 2007.

[7] M. Stein. Visual servo control of a dual-armed baxter robot. `http://cs.nyu.edu/totok/tpcw.html`, 2014.

[8] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Computer Vision, 1998. Sixth International Conference on*, pages 839–846, 1998.

[9] G. A. Vinod, G. Hoang, P. S. Pratama, H. K. Kim, S. B. Kim, and B. H. Jun. Object following control of six-legged robot using kinect camera. *Advances in Computing, Communications and Informatics*, pages 758–764, 2014.

[10] whitney. Resolved motion rate control of manipulators and human prostheses. *IEEE Transactions on man-machine systems, 1969. IEEE*, pages 47–53, 1969.

[11] W. Xinyu, Y. Chenguang, J. Zhaojie, H. M, and F. Mengyin. Robot manipulator self-identification for surrounding obstacle detection. *Multimedia Tools and Applications*, 76(5):6495–6520, 2017.

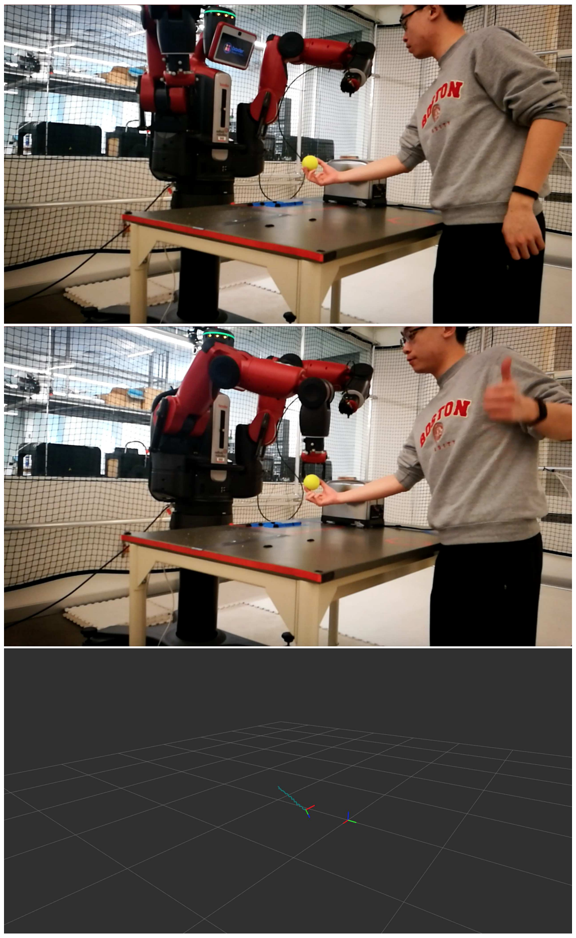

Figure 10. Experiment with target on hand