Robotics Project: Estimating the Kinematic State of a Lockbox Puzzle

Yuchen Liu
Supervisor: Manuel Baum

***Abstract*— This project aims at tracking a lockbox with movable blocks and doors on its surface based on DART, which is a general framework for tracking articulated objects. However, there are some problems with DART, such as it is not able to converge correctly when objects are close to each other, or the initial guess is too far away from the ground truth. In this project, we evaluate the performance of the DART framework on lockbox tracking, we also propose an approach to improve the intersection term in the loss function so that it will perform better on lockbox tracking.**

***Index Terms*— Articulated lockbox tracking, DART, Signed distance function**

I. Introduction

Object pose estimation based on depth maps or point clouds is important for a variety of computer vision and robotics applications, such as indoor 3D environment understanding and robotic manipulation. Researchers have been working among research including robot navigation, planning, manipulation, and human-robot interaction on state-space and model-based algorithms in computer vision and robotics. In real life, the category of objects which can be described as several rigid body parts articulated together is quite broad. Such articulated objects include tools, furniture, animal bodies and human bodies. While manipulating such articulated objects, the known information of the configuration of the object makes the result more accurate and reasonable, which will give the robots a better understanding of the environment and avoid collapse due to unreasonable estimation. [1].

This project aims at tracking the state of an articulated lockbox by RGB-D data. The image of the lockbox is shown in Figure 1. The key of this project lies in how to design an algorithm specifically for tracking lockbox as well as to improve the robustness of this framework.

The inputs of this system are depth images captured by depth cameras, the 3D model, defined as obj files, and the initial poses. Then the system should optimize the prediction of new poses by finding the local optima of the signed distance function which is the distance between the surface point clouds of previous poses and the point clouds of depth images.

However, this system can only be applied to rigid bodies connected by joints which are either prismatic or rotational. Also, since the system applies a iterative method, the settings of initial poses have a great impact on the tracking results. Therefore, the convergence abilities of the system is tested and analysed. Meanwhile, some improvements for lockbox specified tracking system are proposed and evaluated.

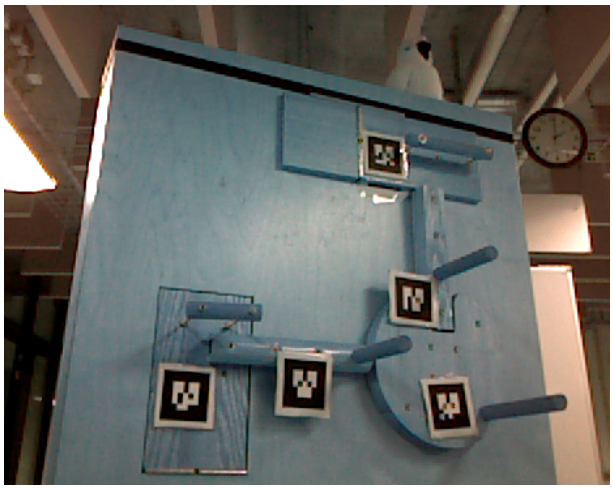


Fig. 1: The image of lockbox for tracking

The main contributions of this project are: (1) make use of the general articulated object tracking framework DART to track lockbox, (2) improve its performance on rigid objects with large contact area by modifying its intersection punishment calculation.

II. Related Work

There are mainly two different approaches to estimate object poses, one is based on machine learning, the other one is directly based on extracting features from RGB-D frames. Recent works such as [2] focuses on the 6-D object tracking based on deep neural network, while [3] utilizes the YOLO (You Only Look Once) [4] to detect 3-D bounding box corners of objects in the images and then recover the 6-D poses. On the other side, in [5], the pose of object is updated by optimizing the projected contour from the 3-D model. This work [6] utilizes both the color data and the depth data, and optimize it on a single CPU. Meanwhile, [7] uses only depth data for tracking articulated objects and optimizes it on GPU.

This project is mainly based on DART: Dense Articulated Real-Time Tracking [7]. DART is a general framework for tracking articulated models with defined kinematic and geometric structure using depth data. DART uses signed distance function (SDF) to represent each rigid part, then generates the signed distance function for the whole articulated objects based on the kinematic tree and the pose estimation. The signed distance function is used to achieve robust and efficient results for online 3D mapping [8, 9] and for tracking rigid objects in six degrees of freedom

[10]. Then, based on traditional Iterative Closest Point (ICP) algorithm, DART directly extends the loss function with free space information and the intersection punishment. The loss derived from free space information, which is called observation SDF, increases the loss when the surface of the model appears where there is no point cloud between the model and the camera. The intersection punishment increases the loss when rigid parts cross each other in 3D space. The diagram of DART is shown in Figure 2.

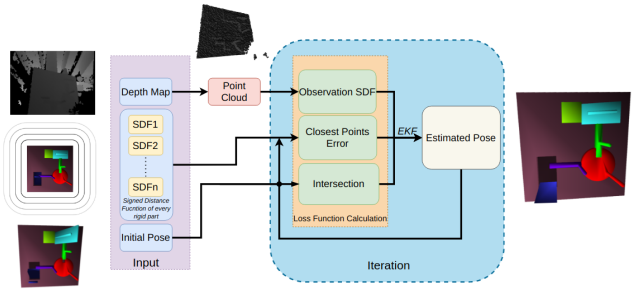


Fig. 2: The DART diagram

However, DART simplifies the intersection term by replacing the triple integral over a discrete representation of a SDF with two surface integrals, which means, instead of taking interior parts of both models into account, DART only checks a finite set of points on the surface of all rigid bodies. It performs well for hand tracking [11], but for rigid objects that are in close contact over large areas, such as blocks on the surface of the lockbox shown in 1. To deal with this problem, we propose an approach to shrink the surface of double integral, so the intersection term in loss function will be negligible when objects are just in close contact, but do not cross much in 3D space.

In the next section, we will discuss the technical details of DART system, explain the problems regarding lockbox tracking, and illustrate our improvement implementations.

III. Technical Section

Since our project is mainly based on DART, and the improvement is deep into this algorithm, so we will briefly introduce the ideas of DART first.

A. DART: Dense Articulated Real-time Tracking

In general, DART stores the models as a collection of rigid parts connected by a kinematic tree based on estimated poses. The rigid parts are described as signed distance functions (SDF), which denotes the signed distance of every point in 3D space to the surface of the model. When combining SDF with ICP (Iterative Closest Point) algorithm, the advantage is obvious: we don't have to repeatedly calculate the distance between every point in the point cloud and the surface of the model. We only have to find out the closest rigid part of the point cloud, then transfer the point to the frame of the closest rigid part. Finally, the signed distance could be easily acquired from previously calculated SDF.

However, the original DART framework doesn't consider any physical constraints, which means the framework will consider intersection between several rigid parts as reasonable results. To extend this framework, [11] proposes a new intersection term to the full error function to be minimized. The new error function becomes:

$$E(\theta;D) = E_{mod}(\theta;D) + \lambda_{obs}E_{obs}(\theta;D) + \lambda_{int}E_{int}(\theta) \quad (1)$$

where E_{mod} and E_{obs} are the errors between observed points and the estimated model points in the model SDF and the observation SDF. The last term, intersection term, however, is the most important to focus on in this project.

B. Intersection Term

The intersection term is established based on an obvious fact in reality: rigid bodies cannot intersect with each other in space, i.e., for any point in three-dimensional space, it cannot be contained inside by more than one rigid body at the same time. Intentionally, the error function to penalize such intersection is:

$$\iiint \min(0, f_a(x,y,z))\min(0, f_b(x,y,z))\mathrm{d}x\mathrm{d}y\mathrm{d}z \quad (2)$$

In order to accelerate the calculation of the algorithm, DART calculates the intersection term on double surface intergral instead, since points on the surface of one rigid body shouldn't be also on the surface of another rigid body. The intersection term then becomes:

$$\oiint \min(0, f_a^2(x,y,z))\mathrm{d}S_B + \oiint \min(0, f_b^2(x,y,z))\mathrm{d}S_A \quad (3)$$

Such penalization works well in some scenarios such as robot hand tracking and human body tracking [12]. However, in our case, where most of the rigid objects are attached to others, those closely-contacted surfaces will cause significant incorrect penalization on error function, since every point on the attached surface will be taken as intersected points and will be added to the intersection term. The framework will be completely not working if we take the intersection into account.

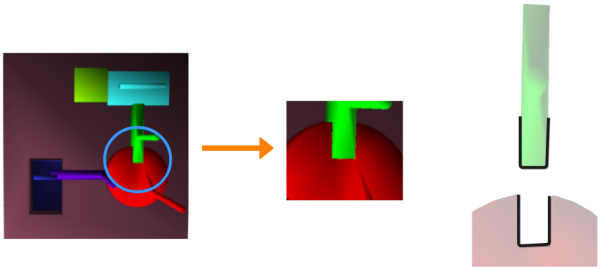
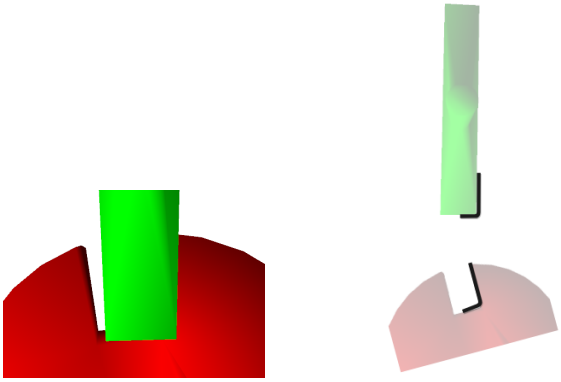


Fig. 3: Explanation of bad intersection term

To illustrate this problem more explicitly, we will take the connection area between the disk and the cuboid as an example. As shown in Figure 3, when DART looks into the surface where the the cuboid model and the disk model



(a) The wrong convergence due to the unsuitable calculation method of the intersection term

(b) The punished area that is calculated by Equation 3 in wrong convergence situation shown on the left

Fig. 4: An example of bad convergence result

touch, the system will consider the whole contacted surface area as penalized area, which is shown as black area on the right. Such penalization will cause the estimation poses converging into unexpected results, as the ground truth will be penalized in the algorithm. In Figure 4, an example of bad convergence result is shown.

One thing to clarify is that, ideally, the intersection term based on Equation (3) is not supposed to punish the loss function for ground truth scenarios shown in Figure 3. However, the surface of the lockbox model is first established by SDF, estimated poses and kinematic tree, then rendered by direct tri-linear interpolation of the SDF. Each calculation process discussed above causes deviation to the final estimated model. Therefore, for articulated objects which barely contact such as robot hands, such deviation can be ignored. But for closed contacted articulated objects, such as lockbox, the deviation cannot be neglected.

To avoid incorrect punishment to intersection term, we therefore propose an idea to improve the system by shrinking the model when calculating the intersection term. The distance between the surfaces in ground truth scenarios will thereby be larger, which gives some space for deviation generated in the graphics rendering and interpolation process.

To evaluate the DART tracking performance and test our improvement on the algorithm, we will show the experimental evaluation of DART, and the effects of our improvement in the next section.

IV. Experiments

In this section, we first evaluate the tracking performance in rigid tracking, by analyzing the tracking errors and the convergence ability beginning from different initial guesses, in order to analyze the importance of the initial guess in a quantitative way.

A. Rigid tracking

1) Rigid errors: First, we fix all the articulated parts, and track the lockbox by a collection of depth images in which the lockbox and the camera hold the same place. We then increase the position error between the ground truth and the initial guesses and run the whole tracking algorithm for each initial guess. Each tracking is continuously performed throughout 93 depth frames, and the number of the iteration per frame is 3, to achieve real time tracking. The tracking errors versus frame are shown in Figure 5, in which red means higher initial errors, and blue indicates lower initial errors. The screenshots on the left show different initial guesses along the Y-axis. The dark gray areas are the point cloud derived from observed depth images, and the model is at the position where the initial guess is.

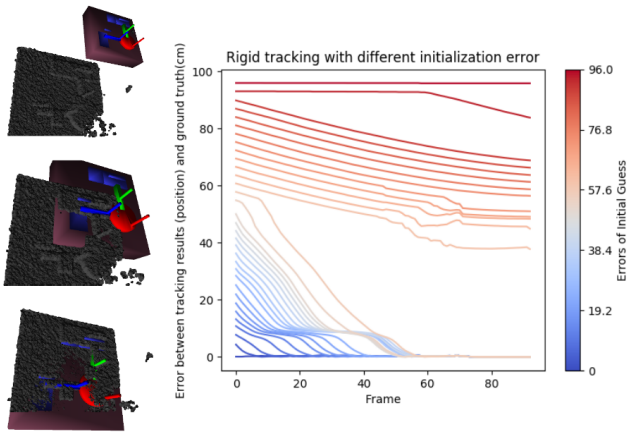
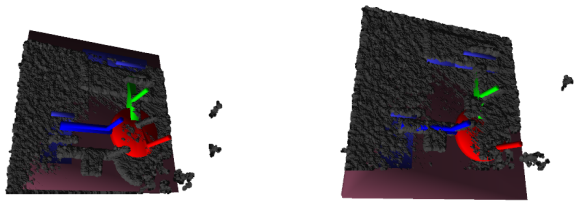


Fig. 5: Rigid tracking errors

As we can see, the initial guess is crucial to tracking performance. If the error of initial guess is higher than a threshold, in our case it is around 60cm (noting that the size of the lockbox backboard is 95cm×95cm).

One interesting phenomenon showing in the result is that, if the initial error is between 15cm to 60cm, the tracking will first converge to a same state, whose error is approximately 10 cm. This happens because the depth images we acquire do not cover the whole lockbox, and the lower part is missing. This specific local optimal is shown on the left in Figure 6.



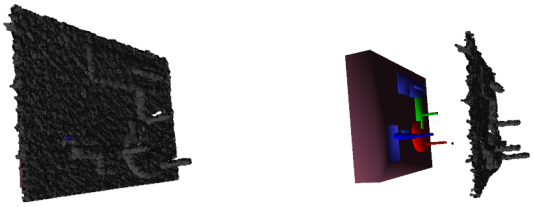
(a) The local optimal our tracking system falls into due to the in-completion of depth images

(b) The ground truth

Fig. 6: The local optimal in rigid tracking

Besides, if the initial error is between 60cm to 90cm, the tracking seems to converge slowly to another state. Seen from the camera frame, this state is shown in Figure 7. On the left

is the view from the camera frame, on the right is the view from side. We can see that the model is completely occluded by the point cloud. The reason of this is the first two terms of Equation 1. The first one E_{mod} calculates the point error between the model points and the observed depth points, which will become ineffective if the model and the point cloud is too far away. The second term E_{obs} punishes the loss function if the model is shown in free space. The definition of free space is the collection of all 3d points that are not occluded by the point cloud in the camera frame. Therefore, if the initial guess is too far away from the ground truth, the estimated pose will probably fall into a local optimal as long as it is behind the point cloud.



(a) Wrong convergence: front view

(b) Wrong convergence: side view

Fig. 7: The wrong convergence in rigid tracking

2) Convergence ability: Now we add the rotation error to the initial guess, and test the convergence ability. We tested all the combination of initial translation error from 0cm to 45cm, 5cm step, and initial rotation error from 0 degree to 55 degrees, 5 degrees step. For each error, we randomly select 20 samples, and run the whole tracking process for each sample continuously with 93 depth frames, and 3 iterations per frame. For each sample, if the final translation error and the rotation error are both lower than a threshold, then we consider it as a successful convergence. The probability of convergence for an error is calculated by the number of successful convergence among all the samples, divided by the number of samples, which is 20. The result is shown in Figure 8, in which more purple means higher convergence probabilities, and more blue indicates lower convergence probabilities.

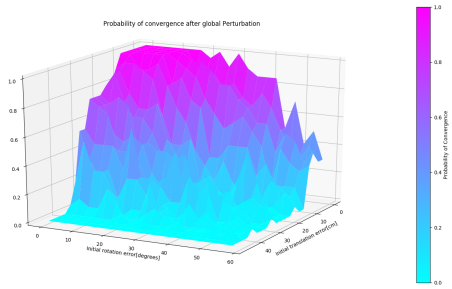
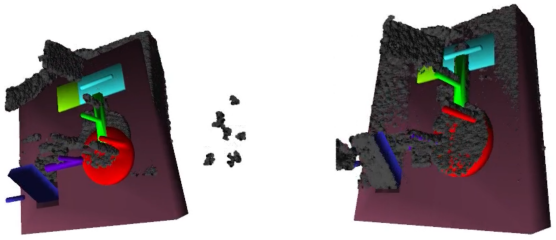


Fig. 8: Probability of converge

As we can see from the figure above, A guaranteed success of convergence requests the initial guess be very precise. In our case, it will likely converge if the initial translation error is no more than 10cm, and the initial rotation error is no more than 10 degrees.

B. Improvement on Intersection terms

When tracking the articulated lockbox, the DART system is not able to work before we improve the intersection term. A screenshot of the tracking result is shown on the left in Figure 9. It is completely not working. After the improvement we mentioned in Section III-B, now the system works much better. For most of the articulated parts, such as the disk, the cylinders, and the door in the bottom-left corner, they are able to respond to the change in the point clouds.



(a) Failed tracking

(b) Successful tracking

Fig. 9: The comparison before and after the improvement on intersection term

However, as we can see, the door whose color is light blue is not responding at all. This is probably because the dart system doesn't equally treat every articulated object. If we only change the order of the articulated objects defined in the model file without changing any parameters, the tracking results will be completely different. However, the actual reason of this problem remains ambiguous to us now, which requires future research.

V. Conclusion

To conclude, this project tracks the articulated lockbox using depth data and predefined models based on a general tracking framework DART. After the evaluation of the tracking performance on lockbox, we found that the initial guess is crucial to this system, which is reasonable since it is a iterative tracking algorithm based on gradient.

Besides, since the lockbox is a closely-attached articulated objects, the original intersection term defined in the loss function is not effective any more. Large surfaces which are closely touched can cause significant punishment on the loss function. Therefore, we propose an approach to solve this problem, by shrinking the size of the model when calculating the intersection term. Such approach will leave more distance between articulated objects, which successfully avoids incorrect punishment to the loss function.

However, there are also unsolved problems. The order of articulated objects defined in the model seems to be crucial to the performance, but it should be equal since the point error

based on SDF is calculated separately. Besides, sometimes the model will fall into the local optima behind the point clouds. Such kidnap situation is a common problem of all iterative methods, which requires future work and research.

REFERENCES

[1] A. Paolillo, K. Chappellet, A. Bolotnikova, and A. Kheddar, "Interlinked visual tracking and robotic manipulation of articulated objects," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, pp. 2746–2753, 2018.

[2] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao–blackwellized particle filter for 6-d object pose tracking," *IEEE Transactions on Robotics*, vol. 37, no. 5, pp. 1328–1342, 2021.

[3] B. Tekin, S. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," 06 2018, pp. 292–301.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," 06 2016, pp. 779–788.

[5] V. Prisacariu and I. Reid, "Pwp3d: Real-time segmentation and tracking of 3d objects." vol. 98, 01 2009.

[6] W. Kehl, F. Tombari, S. Ilic, and N. Navab, "Real-time 3d model tracking in color and depth on a single cpu core," 11 2019.

[7] T. Schmidt, R. A. Newcombe, and D. Fox, "Dart: Dense articulated real-time tracking," in *Robotics: Science and Systems*, 2014.

[8] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, 2011, pp. 127–136.

[9] E. Zobeidi and N. Atanasov, "A deep signed directional distance function for object shape representation," *CoRR*, vol. abs/2107.11024, 2021. [Online]. Available: https://arxiv.org/abs/2107.11024

[10] P. Jatesiktat, M. Foo, G. Lim, and W. Ang, *SDF-Net: Real-Time Rigid Object Tracking Using a Deep Signed Distance Network*, 06 2018, pp. 28–42.

[11] T. Schmidt, K. Hertkorn, R. Newcombe, Z. Marton, M. Suppa, and D. Fox, "Depth-based tracking with physical constraints for robot manipulation," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 119–126.

[12] T. Schmidt, R. Newcombe, and D. Fox, "Dart: dense articulated real-time tracking with consumer depth cameras," *Autonomous Robots*, vol. 39, 07 2015.