Towards Autonomous Apple Fruitlet Sizing with Next Best View Planning

Harry Freeman, George Kantor

Carnegie Mellon University, Robotics Institute
5000 Forbes Ave, Pittsburgh, PA 15213

Abstract

As a result of the expected effects of global population growth and the anticipated need to increase food production, agricultural robotics has become a popular research area. Robots are able to automate laborious and time-consuming tasks which allows farmers to make faster decisions and ultimately improves yield. In this paper, we build towards designing a robotic system that autonomously sizes apple fruitlets. Our proposed system adopts a viewpoint planning approach targeted towards sizing smaller fruit. We utilize a coarse and fine planning tree along with a region of interest utility-gain mechanism to generate next-best view candidates to capture images of fruitlets. A truncated signed distance function is used to build a dense surface point cloud and fruits are sized using a combination of 3D and 2D techniques. We provide preliminary simulated results demonstrating that our system can effectively size fruitlets in occluded environments.

Introduction

Apple fruitlet sizing is important because farmers use sizes to determine when to apply chemical thinners to their crops to optimize yield. The current sizing method used in practice involves using calipers to manually record the diameters of hundreds of fruitlets. This process is quite labor intensive and farmers are actively seeking alternative solutions.

There has been significant effort dedicated towards deploying robotic systems in the field in order to automate agricultural tasks. Robotic systems have been designed to assist with mapping (Marangoz et al. 2022), phenotyping (Shafiekhani et al. 2017), pruning (Silwal et al. 2021), and harvesting (Mangaonkar et al. 2022). These systems are able to produce high-throughput results that enable farmers to make real-time decisions to better manage their crops. However, it is challenging for a robot to non-destructively size apple fruitlets. This is because fruitlets are small - around 6mm when the sizing process starts - and grow in close proximity to one another. This makes them hard to detect, track, and uniquely identify. As well, they grow in very occluded environments, making it difficult to capture high-quality images and generate complete and accurate 3D models.

In this paper, we present an system for sizing apple fruitlets with a 7 DoF robotic arm. To determine where images should be captured, we develop a Next-Best View (NBV) planning approach that makes use of both coarse and fine planning trees and a region of interest (ROI) utility gain mechanism that can accommodate the fruitlets' small size. Once all images are captured, a surface point cloud is built and the diameters of fruitlets are measured using a combination of 3D and 2D sizing techniques. We provide preliminary simulated results demonstrating that our approach can effectively size fruitlets in an occluded environment and outperforms the current state-of-the-art method.

Copyright © 2023, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Related Work

Several approaches have been introduced to size fruits in agriculture. Popular methods include photogrammetric techniques where fruits are sized from single 2D images. One such method is capturing images with calibration spheres placed behind fruits of interest (Cheng et al. 2017; Wang et al. 2018). This does not adapt well to apple fruitlets as it is time-consuming to place reference objects behind every fruitlet to be sized. Approaches have also been developed to directly measure the widths of fruits using simple 2D geometric approaches (Gongal, Karkee, and Amatya 2018; Wang, Walsh, and Verma 2017; Stein, Bargoti, and Underwood 2016; Ponce et al. 2019). However, they require the entirety of the fruit to be visible in the image and fail in the presence of occlusions.

Approaches have also been developed to perform sizing using 3D information. 3D models are reconstructed from multiple sensor measurements in the works of (Wang and Chen 2020; Jadhav, Singh, and Abhyankar 2018). However, they inconsistently perform in occluded environments where reconstructions are often incomplete. Automated shape completion methods have been proposed by (Lehnert et al. 2016; Marangoz et al. 2022) which fit superellipsoids to accumulated point clouds. This does not extend well to apple fruitlets due to their small size and the inability to capture enough of the fruit's surface.

Recently there has been work dedicated towards NBV planning in agriculture. NBV planning approaches based off ROI exploration are used by (Zaenker et al. 2020; Menon, Zaenker, and Bennewitz 2022; Zeng, Zaenker, and Bennewitz 2022). However, they perform poorly when mapping at the resolution required by fruitlets as a result of the slow and computationally expensive raycasting operations.

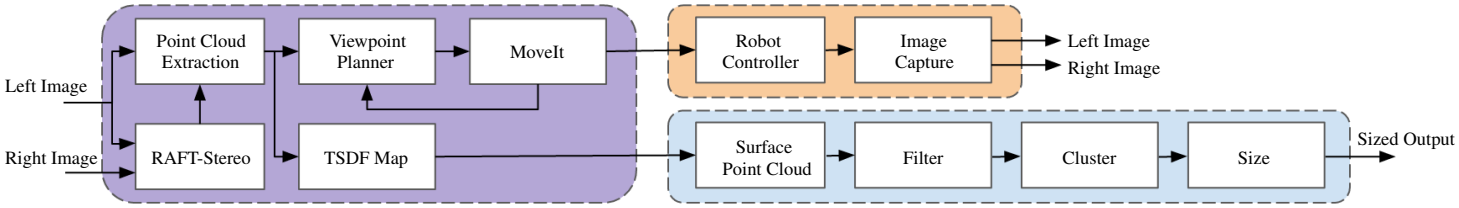

Figure 1: Overview of our fruitlet sizing system

Methodology

System Overview

An overview of our system can be seen in Figure 1. Images are captured using an in-hand flash stereo camera. The camera is attached to the end of a 7 DoF robotic arm (Figure 2) consisting of a UR5 and linear slider (Silwal et al. 2021). Fruitlets are segmented and points are projected onto 3D to build a point cloud with known ROI regions. The point cloud is passed to 1) a planner which maintains both a coarse and fine 3D map of the environment stored as an octree; and 2) a voxblox (Oleynikova et al. 2017) mapping system which maintains a Truncated Signed Distance Function (TSDF). The planner then samples and evaluates viewpoints and determines the next best pose for the end-effector based on expected utility. A path is planned to the pose using the MoveIt framework (Coleman et al. 2014) which is executed by the robot controller. The process then repeats until the specified planning duration is exceeded.

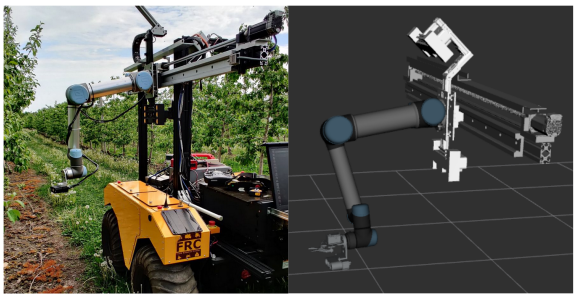

Figure 2: Inhand camera attached to 7DoF robotic arm. Left: real robotic system. Right: simulated model.

Point Cloud Extraction

In our simulated environment, fruitlet segmentation is performed by applying a threshold in the HSV color space while RAFT-Stereo (Lipson, Teed, and Deng 2021) extracts disparities. The segmentations and disparities are used to project points onto a 3D point cloud with ROI information.

Viewpoint Planning

Our viewpoint planner is an extension to the work of (Zaenker et al. 2020), adapted to work in the fruitlet domain. Viewpoint candidates are sampled from the sampling tree and workspace tree using a combination of ROI-targeted and exploration sampling. Estimated information gain is calculated for each sampled viewpoint, and the viewpoint with the maximum utility is selected as the next best pose.

Workspace and Sampling Tree Generation Our viewpoint planner utilizes both a workspace tree and a sampling tree to sample valid viewpoints. The workspace tree defines the valid end-effector poses the robot is able to reach, while the sampling tree identifies the areas of interest that viewpoint targets should be sampled from. To generate the workspace tree, ten million randomly generated joint configurations were sampled in simulation.

Because apple fruitlets are typically sized in clusters, our sampling tree was created by defining a region of space around the cluster of interest. The cluster is identified by an AprilTag (Olson 2011) and a sphere of fixed sized is positioned around the cluster which we will refer to as the Cluster Region (Figure 3). The Cluster Region is then discretized into an octree to build the sampling tree.

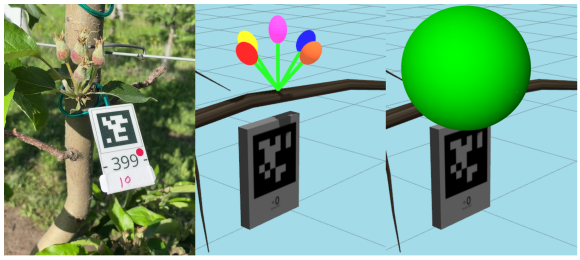

Figure 3: Left: example fruitlet cluster. Middle: simulated fruitlet cluster. Right: Cluster Region positioned around cluster approximated by detected AprilTag pose.

Coarse and Fine Octree The planner maintains two maps of the environment: a coarse octree that stores occupancy information at low resolution, and a fine octree that stores both occupancy and ROI information at higher resolution. The coarse map spans the entire observation space and is used to approximate the occupancy of voxels outside the Cluster Region, whereas the fine map is restricted to within the Cluster Region and is used to identify which voxels are occupied by fruitlets. To build the coarse and fine octrees, we use the OctoMap framework (Hornung et al. 2013) and an ROI extended implementation (Zaenker et al. 2020) respectively.

Two maps are used because of the expensive raycasting operations required to update octrees. Using only one map at finer resolution is slow and inefficient for real-world applications as too few images can be captured within a reasonable planning time. On the other hand, a single coarse map is unable to accurately represent ROIs and size fruitlets as a result of insufficient voxel resolution.

Viewpoint Sampling Viewpoint sampling consists of ROI-targeted and exploration sampling. ROI-targeted sampling is used to propose viewpoint candidates around previously explored ROIs. ROI frontier voxels are identified and used as viewpoint targets. For each target, a viewpoint is selected by sampling a random direction and sensor distance from the target. If the viewpoint does not lie within the workspace tree it is discarded.

Exploration sampling is used to find candidate viewpoints when the space around known ROIs has been sufficiently covered. Frontier voxels are used as targets, and a viewpoint is selected by sampling a random sensor distance along the direction of the estimated surface normal of the target. As in ROI-targeted sampling, the viewpoint is discarded if it does not lie within the workspace tree.

Viewpoint Utility Evaluation

Once all candidate viewpoints are sampled, their estimated information gain is calculated. The information gain metric we use is a modified version of the Proximity Count metric presented in (Zaenker et al. 2020). For each viewpoint, a single ray is cast from the viewpoint towards the target. Each unknown voxel along the ray that lies within the Cluster Region is assigned a weight w_i based on its distance d_i to the nearest known ROI

$$w_i = \begin{cases} 0.5 + 0.5 \cdot \frac{\max_d - d_i}{\max_d}, & \text{if } d_i < \max_d \\ 0.5, & \text{otherwise} \end{cases} \tag{1}$$

where $\max_d$ is a specified maximum distance. Known voxels and voxels that do not lie within the Cluster Region are assigned a weight of 0. This ensures that only uknown voxels within our sizing area of interest contribute the information gain.

Raycasting terminates once an occupied voxel is encountered or the target voxel is reached. The number of voxels along the ray that lie within the Cluster Region N_c are counted and the information gain (IG) is calculated as

$$IG = \frac{1}{N_c} \sum_{i=0}^{N_c - 1} w_i \tag{2}$$

We also compute a cost C to move to the viewpoint which is the sum of the movements of all the joints of the robot and is scaled by α. The final utility of the viewpoint is

$$U = IG - \alpha \cdot C \tag{3}$$

Viewpoint Selection Our viewpoint planning algorithm can be seen in Algorithm 1 (appendix). Viewpoints are sampled using both ROI-targeted sampling and exploration sampling. The viewpoint candidates from both sampling methods are added to a single heap with order determined by their utility. The planner then iterates through the heap until it finds a viewpoint with an expected utility greater than a specified threshold, and that the motion planner can find a successful path to. If no viewpoints are left in the heap, new viewpoints are sampled and the utility threshold is reduced.

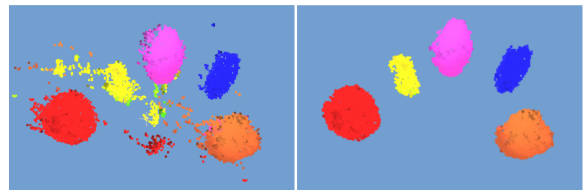

Figure 4: Left: example dense surface point cloud generated by the TSDF. Right: surface point cloud after filtering. Individual fruitlets are clustered using DBSCAN.

Fruitlet Sizing During planning, the generated ROI point clouds are passed to the voxblox mapping system to maintain a TSDF of ROIs. Once planning is complete, the TSDF is used to generate a dense surface point cloud of the fruitlets. The surface point cloud is filtered by removing points that do not fall within an ROI occupied voxel of the fine octree and by applying statistical outlier removal (Figure 4). DBSCAN (Ester et al. 1996) is then used to cluster the individual fruitlets.

Sizing fruitlets in 3D is challenging as a result of their small size and limited surface visibility. To overcome this, we utilize a combination of 2D sizing techniques and our 3D model. To size an individual fruitlet point cloud, we reproject the point cloud onto each image that was taken during planning. For each image, a convex hull is fit around the 2D segmentation and an ellipse is fit using a least squares formulation. The size of the fruitlet is calculated as

$$\text{size} = \frac{ma \times b}{d} \tag{4}$$

where ma is the minor axis of the fit ellipse, b is the baseline of the stereo camera, and d is the max disparity value found in a square region around the center of the reprojected segmented fruitlet. The derivation of equation 4 can be found in (Qadri 2021). After a size is calculated for each image, the largest size is used and estimated to be the fruitlet's diameter. An overview of the ellipse sizing process can be seen in Figure5.

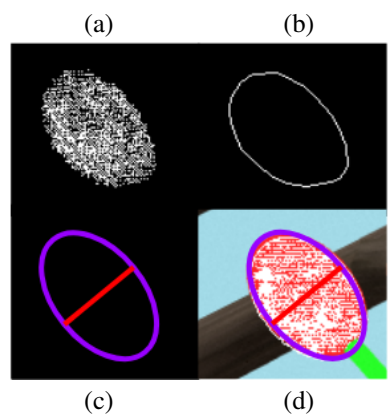

Figure 5: Ellipse sizing process. (a) Segmented fruitlet is reprojected back onto 2D image. (b) Convex hull is fit. (c) Ellipse is fit and minor axis used to estimate size (d) Example segmentation and ellipse on top of the original fruitlet.

Preliminary Experiments and Results

We run preliminary experiments of our robotic system in a Gazebo (Koenig and Howard 2004) simulated environment. For each simulated world, a tree is placed within the workspace of the robotic arm. A fruitlet cluster with a random number of fruitlets in the range of 3 - 6 is randomly placed on the tree. Each fruitlet is modelled as an ellipsoid and is randomly sized. The different coloring of the fruitlets is for visualization purposes and has no effect on segmentation or clustering. To represent the occluded environment, leaves are randomly placed on the tree. An example of a simulated world can be seen in Figure 6.

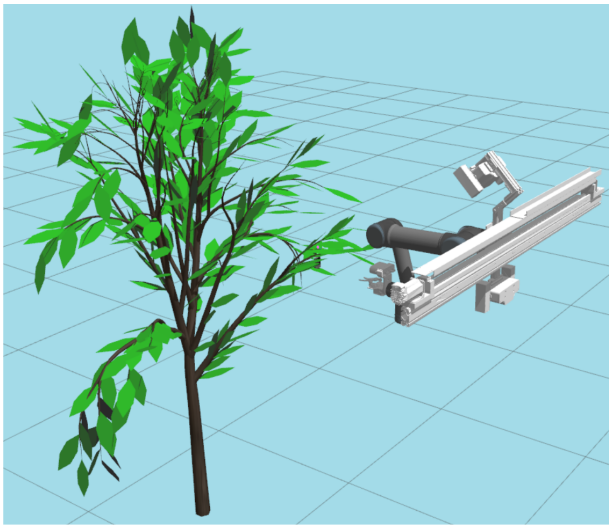

Figure 6: Example simulated world with a randomly generated cluter and leaves.

We evaluate our fruitlet viewpoint planner (FVP) against the state-of-the-art ROI viewpoint planner (RVP) presented in (Zaenker et al. 2020). Both planners were given a three minute planning time to move around the world and capture images. For each planning iteration, 100 ROI-targeted and 100 exploration sampled viewpoints were evaluated to determine the next best view. For the fruitlet viewpoint planner, a coarse octree resolution and fine octree resolution of 0.01m and 0.001m were used. For the ROI viewpoint planner, we used a single ROI octree map of resolution 0.001m. The TSDF voxel size for both planners was 0.5mm. Both planners were run for 35 trials in different simulated worlds.

The Match Percent (MP), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) of both planners averaged over all trials is presented in Table 1. MP is the percentage of clustered fruitlets whose center distance is less than 5mm apart from a ground-truth cluster center. MAE and MAPE are calculated using the error in planner-measured and ground-truth sizes of the matched pairs.

In these preliminary experiments, our viewpoint planner does a notably better job at localizing and measuring the sizes of fruitlets in simulation. Our planner is able to match 8% more fruitlets compared to the start-of-the-art planner. Our MAE is just over 1mm and MAPE is under 10%, which is a 37% improvement over the state-of-the-art method.

	RVP	***FVP***
MP (%)	82.2	**90.1**
MAE (mm)	1.65	**1.04**
MAPE (%)	14.9	**9.35**

Table 1: Match Percent, Mean Absolute Error, and Mean Absolute Percentage Error of our viewpoint planner (FVP) compared to state-of-the-art (RVP)

The distribution of sizes can also be seen in Figure 7. It is clear from the results of both planners that our sizing method produces slightly larger sizes on average compared to ground-truth. However, our planner produces more accurate and consistent results with a much narrower distribution.

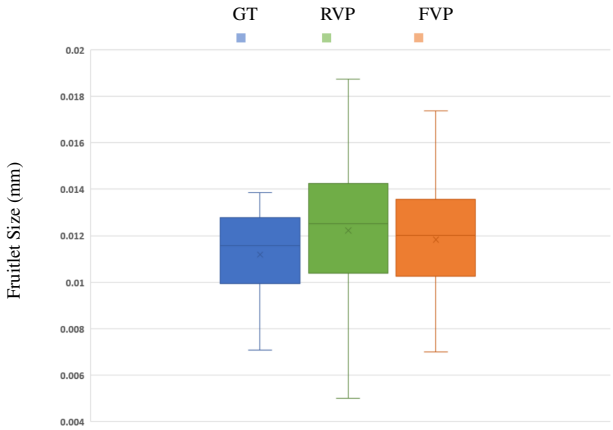

Figure 7: Distribution of ground-truth (GT), RVP, and FVP measured fruitlet sizes. The ”x” symbol indicates the mean and the horizontal line indicates the median.

Future Work

The initial sizing results in simulation show promise, but there is still work needed in order to produce similar results on a real robotic system. For one, our HSV segmentation approach in simulation will not work on fruitlets in the real-world. While we are able to detect and segment fruitlets using the work of (Qadri 2021), real-world segmentation may produce variations as a result of varying illumination conditions that affect the downstream tasks of NBV planning and sizing. In addition, the octree and TSDF maps will suffer from measurement error in the robotic arm forward kinematics. This may have significant effect on the sizing results of fruitlets due to their small size. We look forward to identifying approaches to solve these issues as we build towards developing a fully autonomous fruitlet sizing system.

Acknowledgments

This work was supported by NSF / USDA NIFA 2020-01469-1022394 and NSF Robust Intelligence 1956163.

References

Cheng, H.; Damerow, L.; Sun, Y.; and Blanke, M. 2017. Early Yield Prediction Using Image Analysis of Apple Fruit and Tree Canopy Features with Neural Networks. *Journal of Imaging*, 3(1).

Coleman, D.; Sucan, I. A.; Chitta, S.; and Correll, N. 2014. Reducing the Barrier to Entry of Complex Robotic Software: a MoveIt! Case Study. *CoRR*, abs/1404.3785.

Ester, M.; Kriegel, H.-P.; Sander, J.; and Xu, X. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. of 2nd International Conference on Knowledge Discovery and*, 226–231.

Gongal, A.; Karkee, M.; and Amatya, S. 2018. Apple fruit size estimation using a 3D machine vision system. *Information Processing in Agriculture*, 5(4): 498–503.

Hornung, A.; Wurm, K. M.; Bennewitz, M.; Stachniss, C.; and Burgard, W. 2013. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots*. Software available at https://octomap.github.io.

Jadhav, T.; Singh, K.; and Abhyankar, A. 2018. Volumetric estimation using 3D reconstruction method for grading of fruits. *Multimedia Tools and Applications*, 78: 1613–1634.

Koenig, N.; and Howard, A. 2004. Design and use paradigms for Gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, 2149–2154 vol.3.

Lehnert, C.; Sa, I.; McCool, C.; Upcroft, B.; and Perez, T. 2016. Sweet pepper pose detection and grasping for automated crop harvesting. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2428–2434.

Lipson, L.; Teed, Z.; and Deng, J. 2021. RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching. *arXiv preprint arXiv:2109.07547*.

Mangaonkar, S. M.; Khandelwal, R.; Shaikh, S.; Chandaliya, S.; and Ganguli, S. 2022. Fruit Harvesting Robot Using Computer Vision. In *2022 International Conference for Advancement in Technology (ICONAT)*, 1–6.

Marangoz, S.; Zaenker, T.; Menon, R.; and Bennewitz, M. 2022. Fruit Mapping with Shape Completion for Autonomous Crop Monitoring.

Menon, R.; Zaenker, T.; and Bennewitz, M. 2022. Viewpoint Planning based on Shape Completion for Fruit Mapping and Reconstruction.

Oleynikova, H.; Taylor, Z.; Fehr, M.; Siegwart, R.; and Nieto, J. 2017. Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.

Olson, E. 2011. AprilTag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, 3400–3407.

Ponce, J. M.; Aquino, A.; Millan, B.; and Andújar, J. M. 2019. Automatic Counting and Individual Size and Mass Estimation of Olive-Fruits Through Computer Vision Techniques. *IEEE Access*, 7: 59451–59465.

Qadri, M. 2021. *Robotic Vision for 3D Modeling and Sizing in Agriculture*. Master's thesis, Carnegie Mellon University, Pittsburgh, PA.

Shafiekhani, A.; Kadam, S.; Fritschi, F. B.; and DeSouza, G. N. 2017. Vinobot and Vinoculer: Two Robotic Platforms for High-Throughput Field Phenotyping. *Sensors*, 17(1).

Silwal, A.; Yandún, F.; Nellithimaru, A. K.; Bates, T.; and Kantor, G. 2021. Bumblebee: A Path Towards Fully Autonomous Robotic Vine Pruning. *CoRR*, abs/2112.00291.

Stein, M.; Bargoti, S.; and Underwood, J. 2016. Image Based Mango Fruit Detection, Localisation and Yield Estimation Using Multiple View Geometry. *Sensors*, 16(11).

Wang, Y.; and Chen, Y. 2020. Fruit Morphological Measurement Based on Three-Dimensional Reconstruction. *Agronomy*, 10(4).

Wang, Z.; Koirala, A.; Walsh, K.; Anderson, N.; and Verma, B. 2018. In Field Fruit Sizing Using A Smart Phone Application. *Sensors*, 18(10).

Wang, Z.; Walsh, K. B.; and Verma, B. 2017. On-Tree Mango Fruit Size Estimation Using RGB-D Images. *Sensors*, 17(12).

Zaenker, T.; Smitt, C.; McCool, C.; and Bennewitz, M. 2020. Viewpoint Planning for Fruit Size and Position Estimation. *CoRR*, abs/2011.00275.

Zeng, X.; Zaenker, T.; and Bennewitz, M. 2022. Deep Reinforcement Learning for Next-Best-View Planning in Agricultural Applications. In *2022 International Conference on Robotics and Automation (ICRA)*, 2323–2329.

Appendix

Algorithm 1: Viewpoint Planning

Parameter: u_{t0}
Parameter: $b < 1$

```
u_t = u_t0;
for Planning Duration do
    r_s = roiTargetSample();
    e_s = explorationSample();
    v_s = {r_s ∩ e_s};
    h_s = makeHeap(v_s);
    moveSuccessful = False;
    while h_s ≠ ∅ and peek(h_s) > u_t do
        vp = pop(h_s);
        if moveToPose(vp) then
            moveSuccessful = True;
            break;
        end if
    end while
    if moveSuccessful then
        u_t = u_t0;
    else
        u_t = b · u_t;
    end if
end for
```