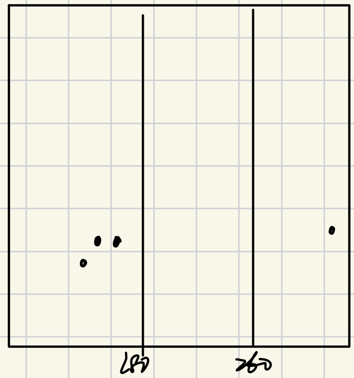
Pb 1

1.1

1.1.1

$N_{near} = N_{start} \langle 90, 120 \rangle$

N_{new} is on the direction from N_{near} to N_{rand} with L1 dist 10



$\Rightarrow N_{new} = \langle 90-5, 120-5 \rangle = \langle 85, 115 \rangle$

N_{new} is valid according to the C-space graph (its θ_1 is not between 180 and 360)

1.1.2

Calculate the L_1 distance between $N_{rand} \langle 200, 120 \rangle$ and existing two point

$$\begin{cases} d_1 = \| \langle 200, 120 \rangle - \langle 90, 120 \rangle \|_1 = 110 \\ d_2 = \| \langle 200, 120 \rangle - \langle 85, 115 \rangle \|_1 = 120 \end{cases}$$

$\Rightarrow N_{near} = \langle 90, 120 \rangle$

$\Rightarrow N_{new} = \langle 90+10, 120 \rangle = \langle 100, 120 \rangle$, it is valid according to the graph (it θ_1 is not btn 180 and 360)

1.1.3 Since the path from $360 \to 0$ is blocked, we can only find a path from $360 \to 0$ (due to C-space is conti)

$N_{goal} = \langle 0, 115 \rangle$

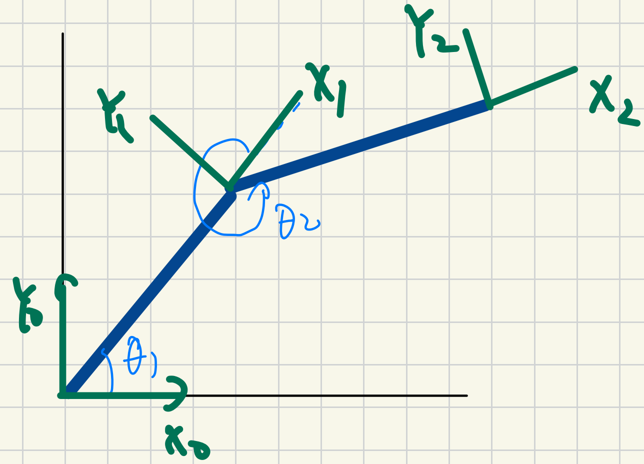
$$\left. \begin{cases} d_1 = \| \langle 0, 115 \rangle - \langle 90, 120 \rangle \| = 95 \\ d_2 = \| \langle 0, 115 \rangle - \langle 85, 115 \rangle \| = 85 \\ d_3 = \| \langle 0, 115 \rangle - \langle 100, 120 \rangle \| = 105 \end{cases} \right\} \quad N_{near} = \langle 85, 115 \rangle$$

$N_{new} = \langle 85-10, 115 \rangle = \langle 75, 115 \rangle$, it is valid since θ_1 is outside of $180 \sim 360$

1.2

1.2-1

$${}^{c}T_w = {}^{0}T_1 \, {}^{1}T_2$$



$${}^{0}T_1 = \begin{bmatrix} \cos\theta_1 & -\sin\theta_1 & 0 & a_1\cos\theta_1 \\ \sin\theta_1 & \cos\theta_1 & 0 & a_1\sin\theta_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{1}T_2 = \begin{bmatrix} \cos\theta_2 & -\sin\theta_2 & 0 & a_2\cos\theta_2 \\ \sin\theta_2 & \cos\theta_2 & 0 & a_2\sin\theta_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \#$$

1-2-2

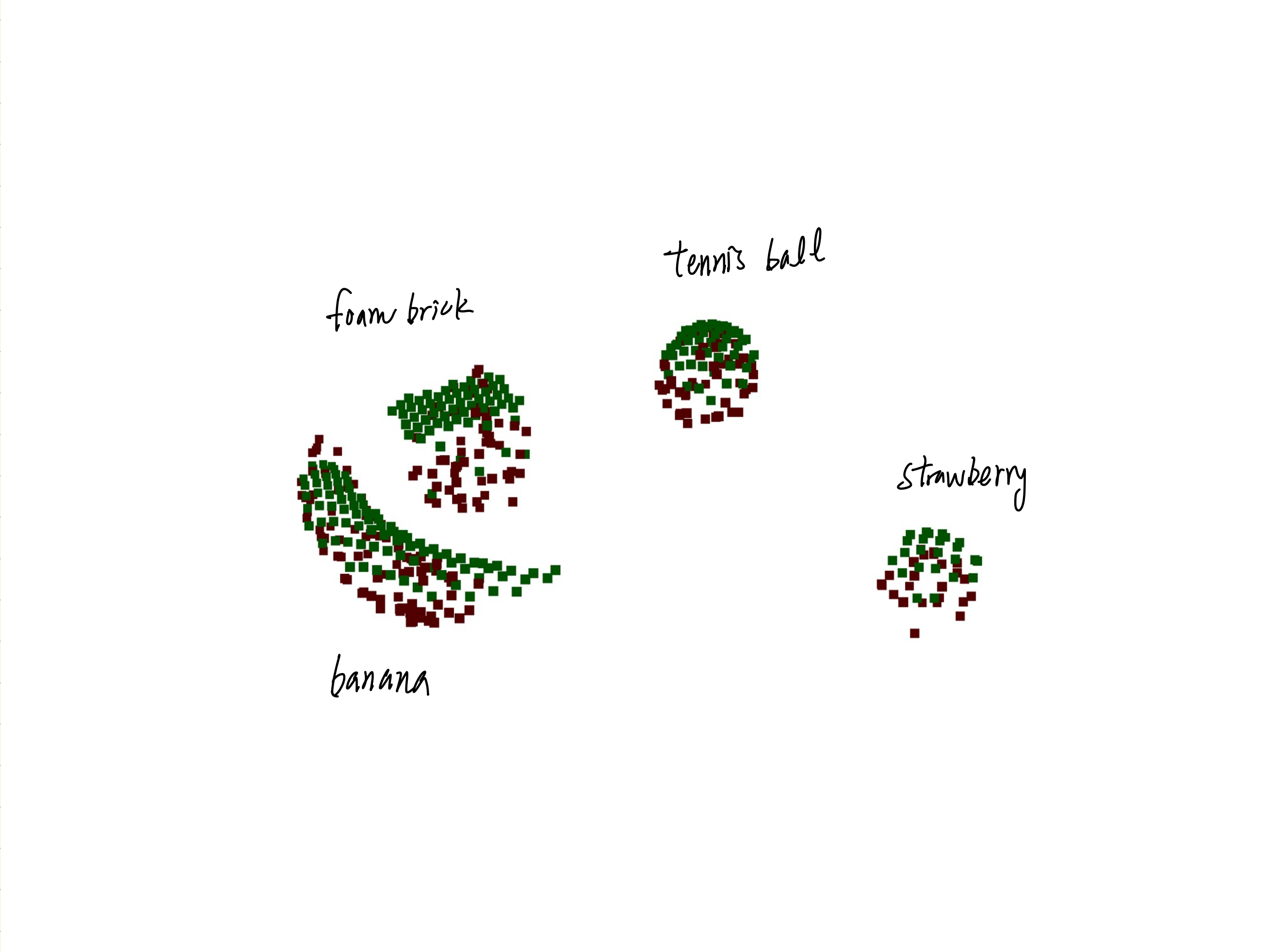
$${}^{w}P = {}^{w}T_c \, {}^{c}P, \quad \theta_1 = \frac{\pi}{2}, \; \theta_2 = -\frac{\pi}{2}, \; a_1 = 1, \; a_2 = 1$$

$$= \begin{bmatrix} 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 1 \\ 0 \\ 1 \end{bmatrix} \Rightarrow {}^{w}P = \langle 2, 1, 0 \rangle$$

The object cannot be reached by the arm since the distance between P and origin is $\sqrt{5} > 2$ (the maximum length of the two-link arm, $a_1 + a_2$)

3.1.1



3.1.2

The common failure modes for this system is

(1) The ICP gives pose estimation which is not accurate enough so that robot cannot correctly grasp the object based on the wrong grasping orientation

(2) The wrong pose estimation may also provide wrong 3D position estimation of the target, which might also cause grasp failure. For example, the gripper might start grasping a bit distance away from the target.

The predicted and the observed point clouds are not exactly aligned since the icp algorithm do not provide perfect transformation matrix.

Also, the inverse kinematic result is not exact solution, where there is error in it.

3.1.3

(1) Improve the ICP result by either increase iteration or decrease threshold. Or increase the observed point clouds number to increase ICP performance (by taking more picture or adding additional camera)

(2) Optimize the IK solver by increasing the iterations or decreasing the threshold.

3.2

3.2.1

We can incorporate RRT, RRT*, A* path planning algorithm in move_joints( ) function

By using inverse kinematics with collision constraints, we can implement collision detection while moving joint.

3.2.2

1. We assume object pose is the grasp pose, which is incorrect in many cases
2. No obstacle in the path
3. We use a large force of 10000 on grasping while also consider banana and strawberry are rigid
4. We assume camera have no distortion, which might not be the case in reality
5. We assume ground truth segmentation mask
6. Assume same friction coefficient among different surface