

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 17th April 2022

MAJOR UPDATES

1. Made a submission with the following changes:
 1. New heuristic task planner replaced the OR-Tools based planer
 2. Gripper width/effort changes
2. Working on grasp modification to avoid:
 1. Collisions during pick operation
 2. And to adjust or sample new grasp pose, if the previous grasp pose is faulty

EXPECTATIONS AND MOTIVES FROM OUR RECENT SUBMISSION

Expectations:

1. We think that the new update will perform as well as the old one with some small ups and downs. There are some advantages in picking the buffer instead of placing the objects blindly in the target position. There are some disadvantages too in the same. So, until we get the results (from the unknown scenes), we won't be able to make a great decision. So, this submission will help us in deciding which task planner to use for our final submission (and, we may also try out hybrid mode, that was proposed by Bipasha)

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 9th April 2022

MAJOR UPDATES

1. Two different versions of task planner ready
 1. One that uses only stacking information, to reform the OR-Tools generated plan. This is a one-time plan generator (similar to OR-Tools but with scene-stacking taken into consideration)
 2. One that uses stacking and target occupancy information to dynamically select the next best action to take
2. We shall make one submission tonight

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 2nd April 2022

MAJOR UPDATES

1. We have received the **results for the third submission** we had made on Mar 24, 2022.
2. It involved **2 simulation submissions** one each in **Sapien** and **Pybullet**.

SAPIEN AND PYBULLET SUBMISSIONS

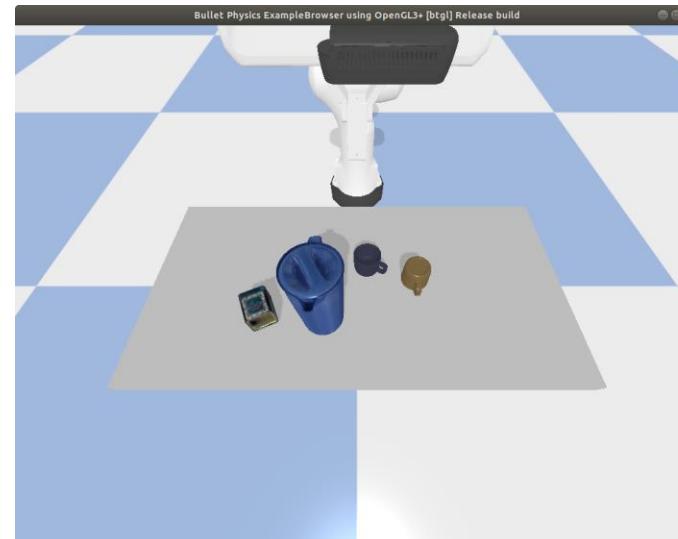
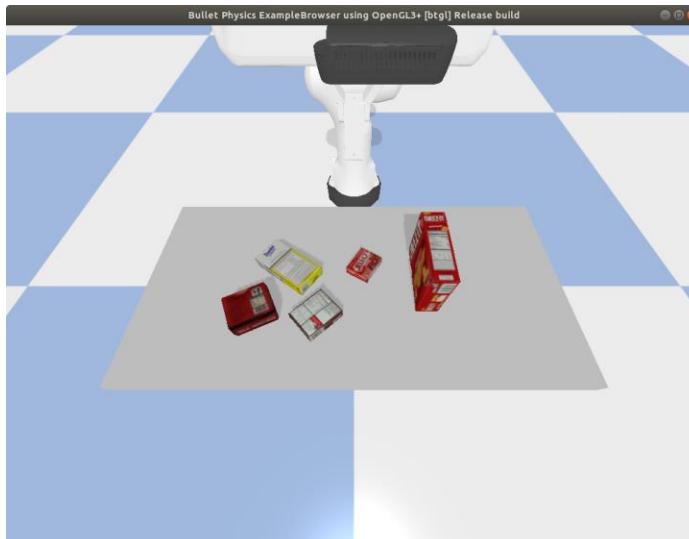
Sapien	PyBullet
Contact Graspnet Fix	Contact Graspnet Fix
Ignoring just the tray like before.	Ignoring certain objects such as plates, books and book holders to reduce the –ve improvement on scenes we can't handle.

ANALYSIS (3RD SUBMISSION)

			Best Scene	
	Major Changes	error per object averaged over all tasks (cm)	number	%improvement
Submission 1: January 1, 2022		34.48	1-5-3-	82.36
Submission 2 : January 30, 2022	Contact Graspnet	33.3	1-1-8	86.58
Submission 3a : Pybullet Mar 24, 2022	<ul style="list-style-type: none"> • Contact Graspnet fix • Ignore certain objects 	28.83	1-1-3-	84.28
Submission 3b : Sapien Mar 24, 2022	<ul style="list-style-type: none"> • Contact Graspnet fix • Test on Sapien 	26.21	1-1-5	90.13

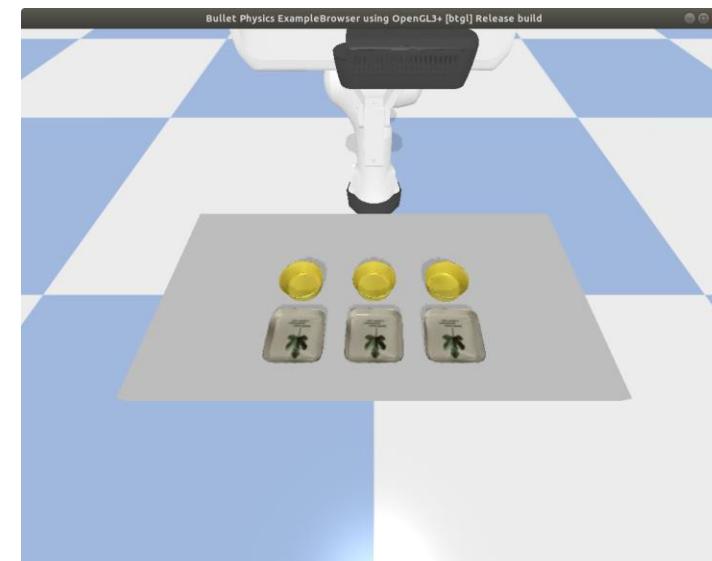
1 SERIES: SCENES THAT NEED IMPROVEMENT

task_index	Scene	Comments:Sapien	Comments: Pybullet
1-2-*	Objects can't be picked [stacking + pushing]	In some scenes we have picked lipton_tea / orion_pie and in some we have a negative improvement over the baseline.	In some scenes we have picked lipton_tea / orion_pie and in some we have a negative improvement over the baseline.
1-4-*	The jug cant be picked	We are picking 3 objects and the scene is not getting destroyed (have +ve improvement on all scenes)	In some scenes, we have picked 2-3 objects and posted an improvement over the baseline, but we also see it failing badly on 50% of the scenes.



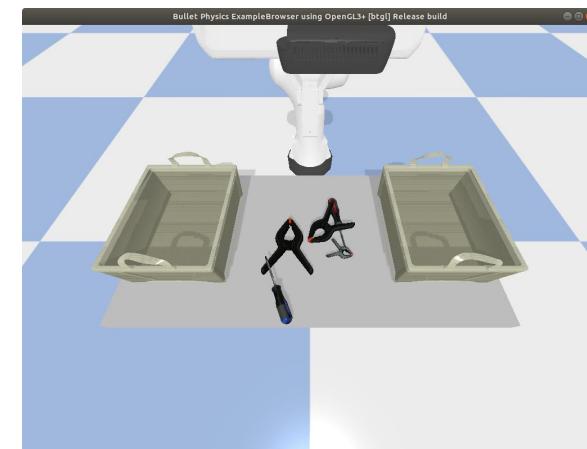
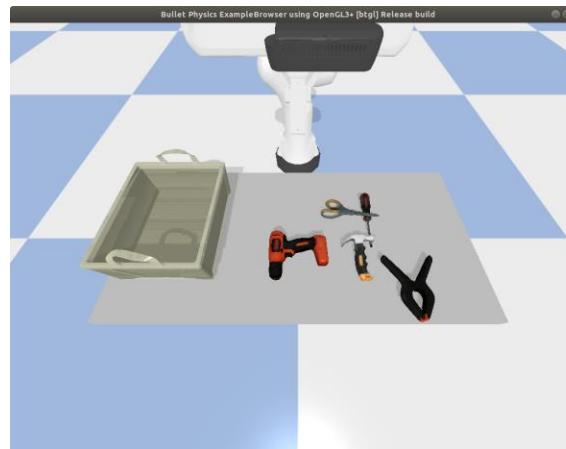
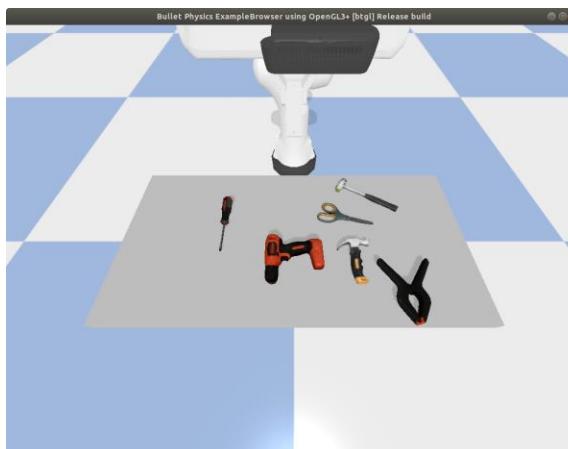
2 SERIES: SCENES THAT NEED IMPROVEMENT

task_index	Scene	Comments:Sapien	Comments: Pybullet
2-1-*	Fruits slip off the gripper and the tray needs to be moved	In most scenes we are picking only 3 or 4 of the 5 fruits. Surprisingly we are seeing a -26% improvement on a known scene 2-1-1. Why in sapien?	In most scenes we are picking only 2 or 3 of the 5 fruits. Detection issue?
2-2-*	Duplicate Objects	Before we were not able to work on the scene. Now we are picking just 1 object . No negative improvement though.	We are picking only 1 object . No negative improvement though.



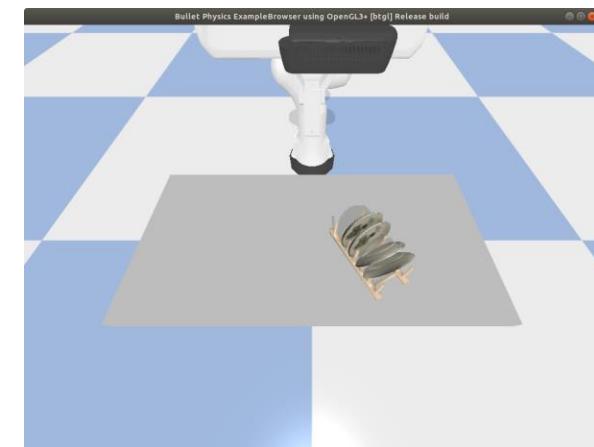
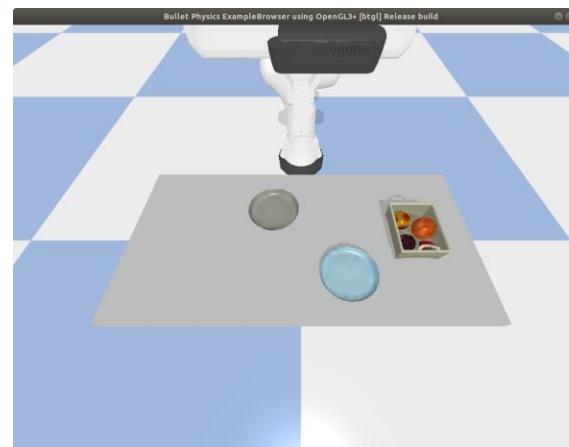
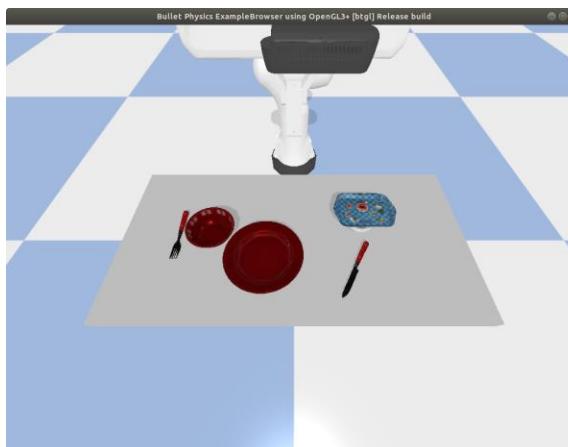
3 SERIES: SCENES THAT NEED IMPROVEMENT

task_index	Scene	Comments:Sapien	Comments: Pybullet
3-1-*	Contact graspnet is imp here	In most scenes we are picking 4/5 objects	In most scenes we are picking 3 / 5 objects.
3-2-*	Contact Grapsnet	In most scenes we are picking 4/5 objects. Some scenes we are picking only 2. Because of clutter?	In most scenes we are picking 3/5 objects. Some scenes we are picking 2 objects. Clutter + detection issue?
3-3-*	Contact Graspnet	We are picking 2 or 3 objects. But in one scene(3-3-1) we have picked 3 and have a – 100 % improvement. Tray issue?	We are picking 1 or 3 objects in most scenes.



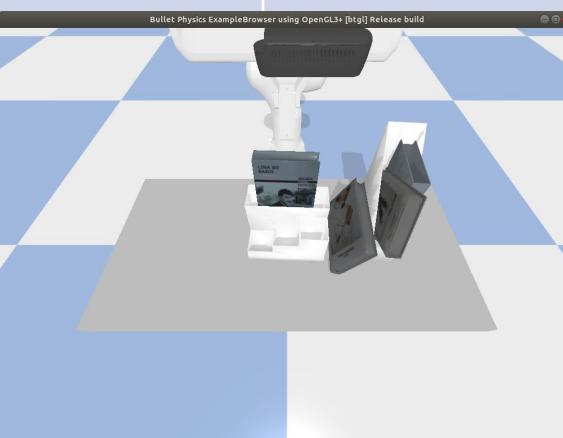
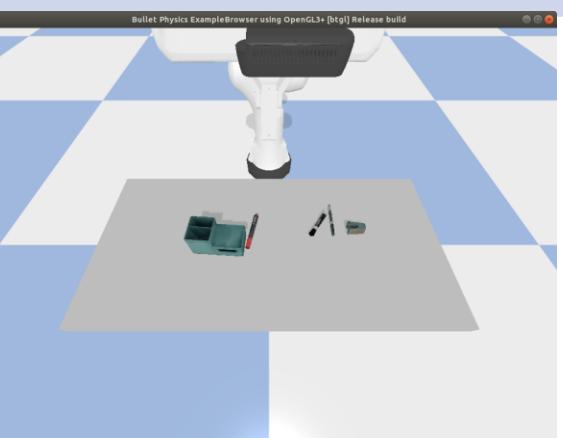
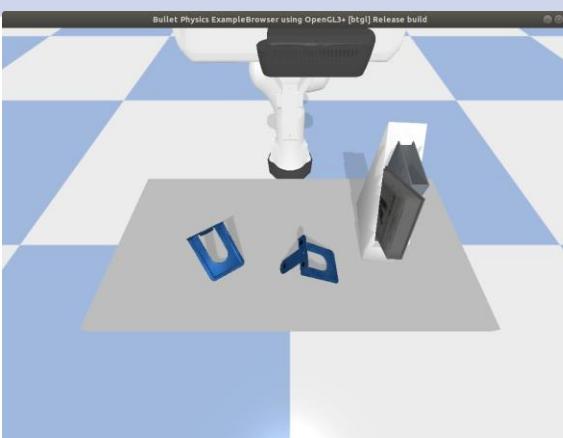
4 SERIES: SCENES THAT NEED IMPROVEMENT

task_index	Scene	Comments:Sapien	Comments: Pybullet
4-1-*	Too many flat + 'difficult to pick' objects	Picking just 1 object. We have a negative improvement of -25 % across most scenes but better than pybullet.	In most scenes we are picking 1 object. And we have a negative improvement across most scenes. Reaches -66 %.
4-2-*	Buffer position + flat objects + pick from inside the tray	We are not ignoring plates, but still we are not able to pick any object.	Fruits not graspable and plates ignored during planning. So, 0% improvement.
4-3-*	Most difficult	We are picking 1 object only. We are posting an improvement of -20% on 50% of the scenes.	Ignoring the plates and tray, so we have 0% improvement. Before we used to have an average of -90% improvment.



5 SERIES: SCENES THAT NEED IMPROVEMENT

task_index	Scene	Comments:Sapien	Comments: Pybullet
5-1-*	Very difficult to pick without hitting the others	<p>Not ignoring scenes. And destruction of scene is happening when it tries to pick. Still reaching -93% improvement. Why in Sapien? But instead of -50% on most scenes, it 0% on most and destruction in one.</p>	Books and holder ignored, so 0% improvement.
5-2-*	Grasp pose issue	Picking 3/4 objects and overall great improvement of >25% improvement across scenes.	Picking 2/3 objects and overall great improvement of >30% improvement across scenes.
5-3-*	Needs orientation on the final object	We are picking 1 object only . And in most scenes >10% improvement. But in one scene, it is -80% after picking.	Books and holder ignored, so 0% improvement.

6 SERIES: SCENES THAT NEED IMPROVEMENT

task_index	Scene	Comments:Sapien	Comments: Pybullet
6-1-*	Needs buffer positions + grasp poses	Picking 2/3 objects on all scenes. Performance similar to pybullet.	Picking 2/3 objects on all scenes.
6-2-*	Grasp pose issue	Picking 2 objects only. But no destruction in the scene, so the improvement > 40 % on all scenes.	Picking 1 or 2 objects only. Why?
6-3-*	Needs orientation on the final object	We are picking 1 object only. We have also posted a -72 % in one of the scenes.	Picking 2 or 3 objects only, but mixed performance. Improvement in some and bad performance in some despite picking.



CURRENT STANDINGS

Rank	Team	error (cm)	1st best task		
			task index	error (cm)	pick-up
1	Lumos	26.21	1-1-5	3.25	4
2	BUAA-GR	27.93	2-2-1	1.52	2
3	NTXZ_robatics	28.82	1-1-4	4.08	3
4	THU-SUN-LAB	34.84	1-1-7	7.81	4

COMMENTS

1. **Sapien is performing extraordinarily well on scenes where pybullet is good.**
2. **Sapien's performance is bad in scenes where pybullet's performance is worse.**
3. We are **not detecting objects** in some scenes.
4. We need to investigate 4 and 5 series majorly as we are not able to pick and thus no improvement.
5. We could integrate pushing. Idea needs reconsideration.

UPCOMING

1. Submissions

- 4th Submission with Vishal's Planner on 7th April
- 5th Submission on 18th April
- Final Submission 30th April.

2. Ideas

- Push Manipulation

PUSH MANIPULATION

Idea:

1. Construct bounding boxes/occupancy grid for the obstacles
2. Fit a bounding box for the target object. Use this bounding box to build a graph of nodes (where each node represents a location on the table that is free)
3. Free nodes can be identified using Convolution operation or any similar methods
4. Use BFS to find the shortest route to the target pose from the object's initial pose. The route can be seen as a set of straight lines. Moving the manipulator's end-effector along a straight line can be achieved using Cartesian planner in move-it
5. Regenerate/update the nodes and the route after every push operation (as object's movements can be fairly uncertain) and go back to step 4. Continue this until the target position is reached

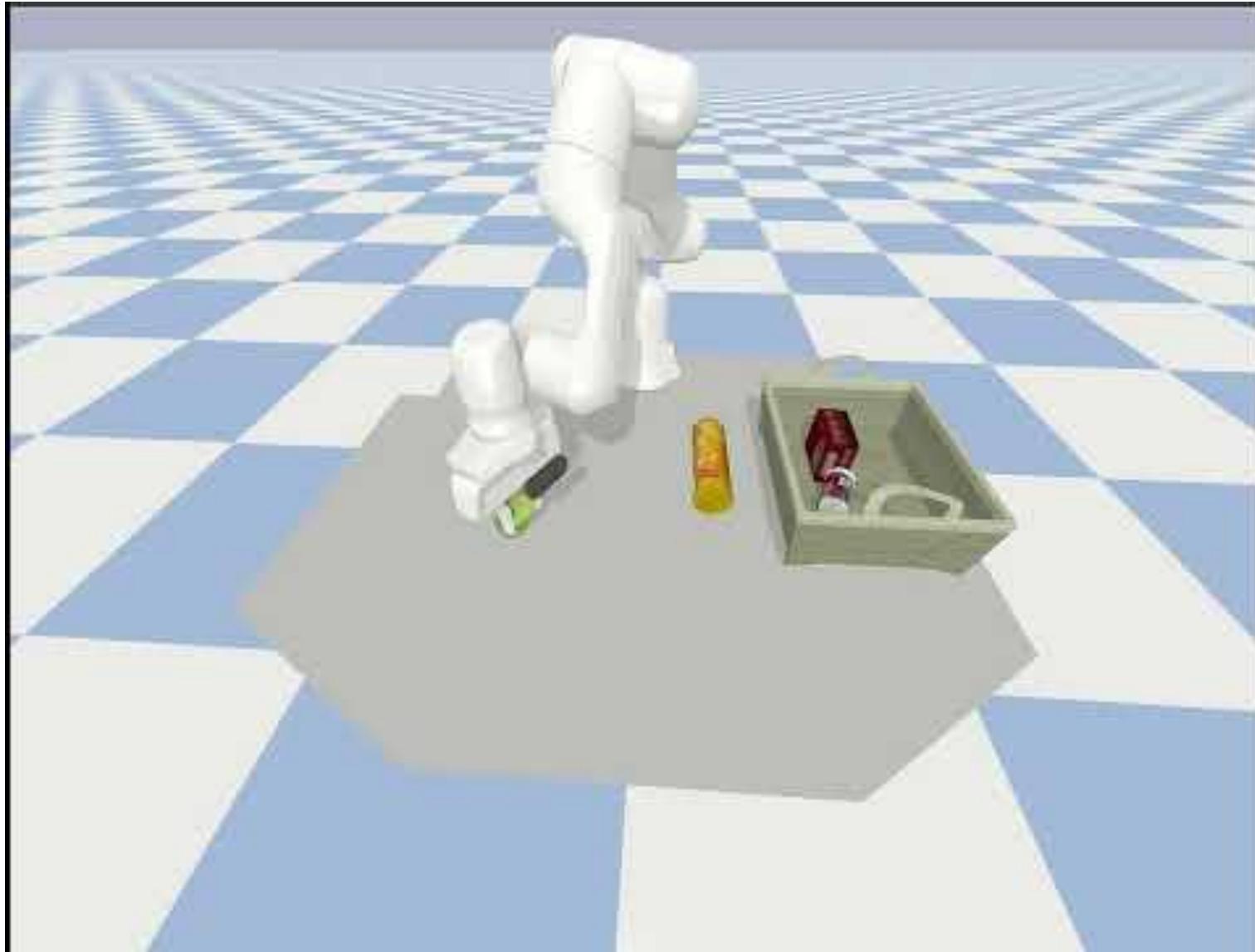
Challenge

1. Identifying appropriate push points before each push operation
2. Since the nodes are at pixel level, we get a smooth curve as a trajectory. But we need a trajectory that looks like a set of line segments. For this, we could try line-fitting using RANSAC for every x-points. There are a lot of potential strategies, we may need to fine-tune by testing selective number of them.

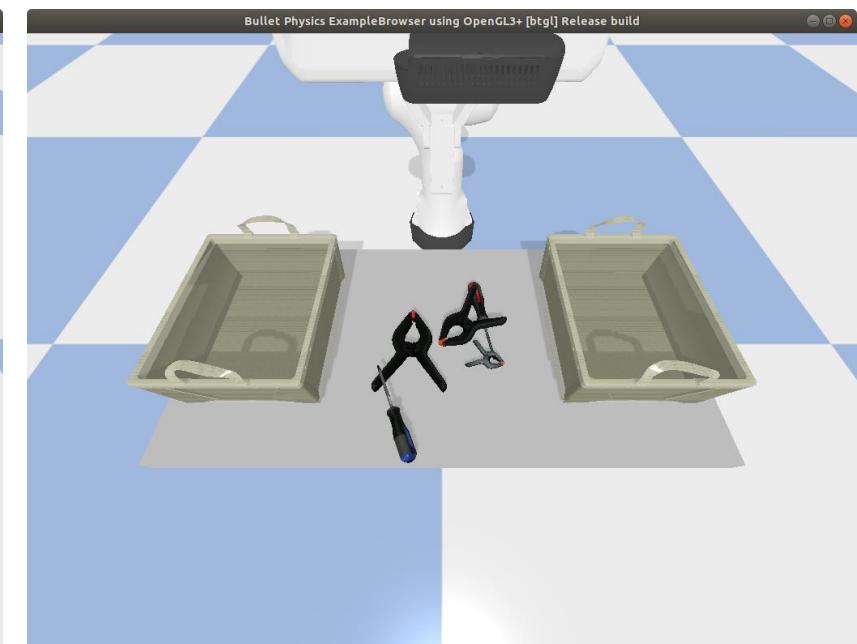
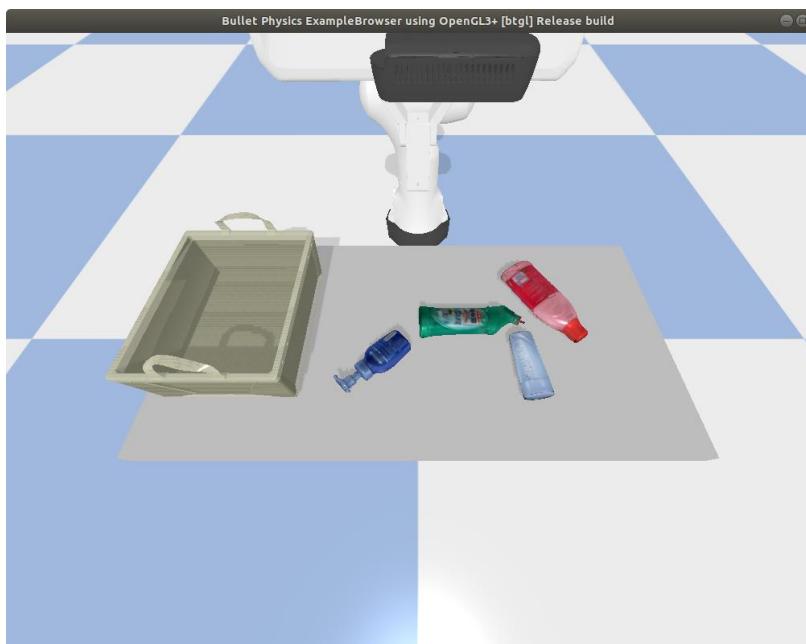
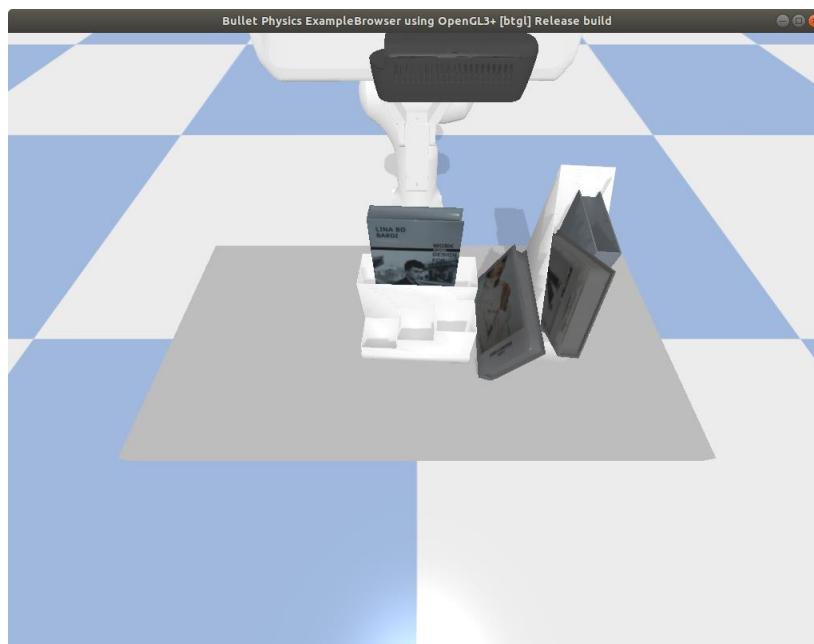
OCRTOC Challenge 2021

Robotics Research Center Submission

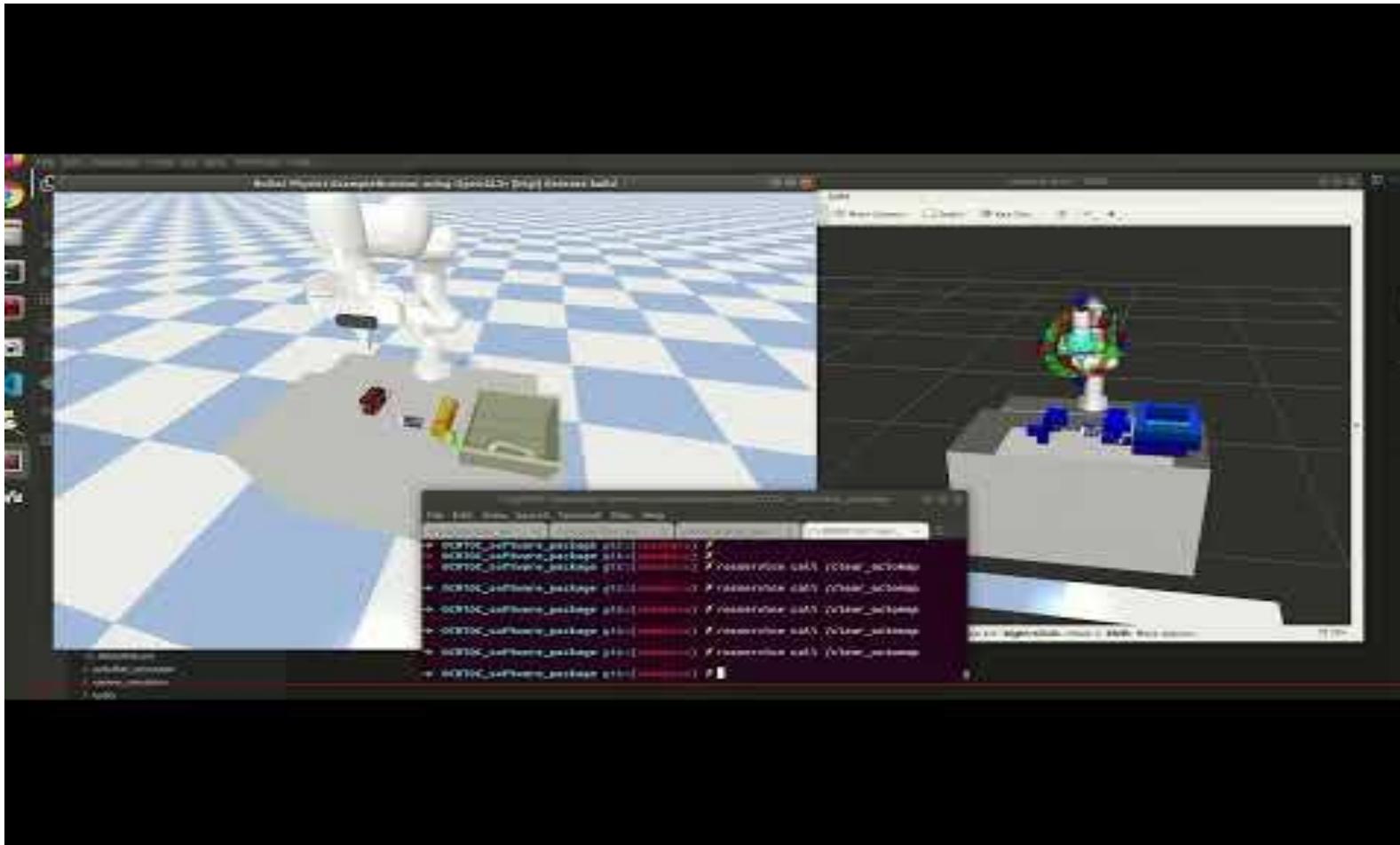
CURRENT WORKING DEMONSTRATION



WE ALSO HAVE SCENES LIKE THESE



CLEAR OCTOMAP FOR A FASTER FREE-SPACE UPDATE - DEMO



Collision Avoidance?

We would be able to consider orientation instead of just a top-down pick place.



OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 5th March 2022

MAJOR UPDATES

1. OCRTOC real robot logs
2. We have received the submission results of the second submission (only contact graspnet) we made on Jan 30, 2022. We have asked them to consider our second simulation submission too.
3. Scene graph generation is now done using an alternate method (ray casting) - Works with 100% accuracy

REAL ROBOT PERFORMANCE FEEDBACK

- Real robot test summary:
- According to our test, the joint limits problem **has been well solved**.

Your algorithm would have a better performance if the following problems can be solved:

- In motion_planning.py: the current gripper control interface is “/franka_gripper/gripper_action”, this interface is not proper for grasping objects with unknown size. We recommend the interface **“/franka_gripper/grasp”**. Of course, the module “franka_msg.msg” should be imported first and the message type would be “franka_gripper.msg.GraspAction”. For the “pick” action, we recommend the following parameters: goal.with = 0.0, goal.epsilon.inner = 0, goal.epsilon.outer = 0.075, goal.speed = 5, goal.force = 20. For the “place” action, we recommend the following parameters: goal.with = 0.078, goal.epsilon.inner = 0.001, goal.epsilon.outer = 0.001, goal.speed = 5, goal.force = 5.
- One of the problems in our last feedback is that the robot often pick an object and place it immediately at the same position. This problem is caused by the “gripper_width_test()” function in task_planning.py. We know you are trying to get the current gripper width with the code “gripper_dist = [joint_state.position[0], joint_state.position[1]]”, but you will get the joint positions of the first two joints of the manipulator, instead of gripper width. The right way to get the gripper width is **“gripper_dist = [joint_state.position[7], joint_state.position[8]]”**. In addition, it would be better if the parameters applied to judge whether an object has been grasped could be changed to **0.0005 instead of 0.005**, because sometimes the object is about 1 or 2 millimeters, such as the wall of a plastic bowl.
- If the performance of the perception algorithm could be improved, the whole performance of your algorithm would be largely improved.
- The joint limits problem has been well solved, **but the robot still reached its joint limits in some cases**.

ANALYSIS (2ND SUBMISSION)

Total number of scenes	100
Public Tasks	30
Hidden (Unknown) Tasks	70

In the Hidden Tasks:

- At each level of hidden tasks, the objects known to us have been used but the arrangement is different.
- The objects are the same as those in the corresponding known public scenes.
- They have used the same scenes as in submission 1

ANALYSIS (2ND SUBMISSION)

			Best Scene	
	Major Changes	error per object averaged over all tasks (cm)	number	%improvement
Submission 1: January 1, 2022		34.48	1-5-3-	82.36
Submission 2 : January 30, 2022	Contact Grasnet	33.3	1-1-8-	86.58



BEST SCENE

task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups	objects picked up
1-1-8	4.48	33.34	86.58	4	<code>['orion_pie_v1', 'suger_2_v1', 'potato_chip_1_v1', 'suger_3_v1']</code>

RED: PUBLIC TASKS

Total number of scenes	30
------------------------	----

SUCCESS:

	Submission 1	Submission 2
Number of Scenes where we have succeeded (i.e. > 10% improvement)	10	9
Number of Scenes with > 1% improvement	16	15

RED: PUBLIC TASKS

Total number of scenes	30	
	Submission 1	Submission 2
Number of Scenes where we have succeeded (i.e. > 10% improvement)	10	9

SUCCESS:

We have 4 new scenes where we are performing really well and
 5 old scenes where our performance has worsened - 3 and 6 series scenes

task_index	Submission 1	Submission 2
1-4-2	37.87	-0.22
1-5-2	24.73	21.97
1-5-3	82.36	39.87
2-1-1	20.46	23.77
3-2-2	34.33	28.43
3-3-1	33.45	-17.82
6-1-1	19.8	-32.73
6-2-3	37.72	25.02
6-3-1	18.29	7.48
6-3-2	24.02	-19.72

task_index	Submission 1	Submission 2
1-4-1	-19.46	32.46
1-5-2	24.73	21.97
1-5-3	82.36	39.87
2-1-1	20.46	23.77
3-1-1	8.27	12.02
3-2-2	34.33	28.43
4-1-2	-19.9	27.76
4-2-2	0.01	12.68
6-2-3	37.72	25.02

BLUE: HIDDEN TASKS

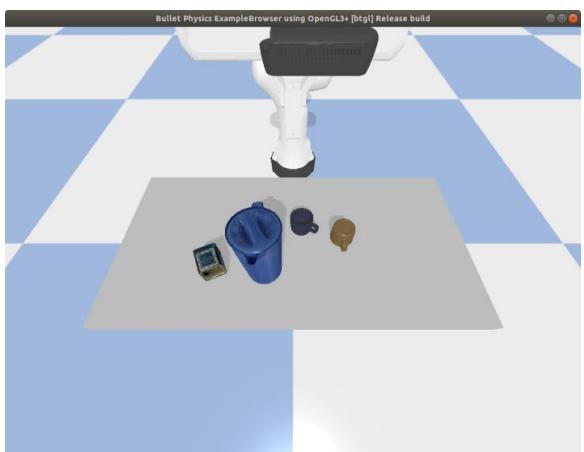
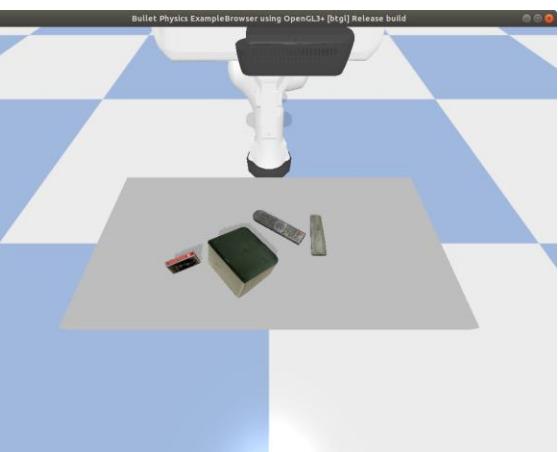
Total number of scenes	70
------------------------	----

SUCCESS:

Number of Scenes where we have succeeded (i.e. > 10% improvement)	27	31
Number of Scenes with > 1% improvement	37	38

1-* SERIES

task_index	Scene description	Success over sub 1	Success over baseline	Comments
1-1-*	Important baseline	6/7 scenes	6/7	In the failed scene, we had picked ¾ objects properly [suger_2 was not picked]
1-2-*	Objects can't be picked [stacking + pushing]	-	-	In some scenes we have picked lipton_tea / orion_pie but we have a negative improvement over the baseline
1-3*	No scene	-	-	-
1-4-*	The jug cant be picked	-	-	In some scenes, we have picked 2-3 objects and posted an improvement over the baseline, but in the failures, its might be because of the jug [rolls off the table when picked incorrectly].



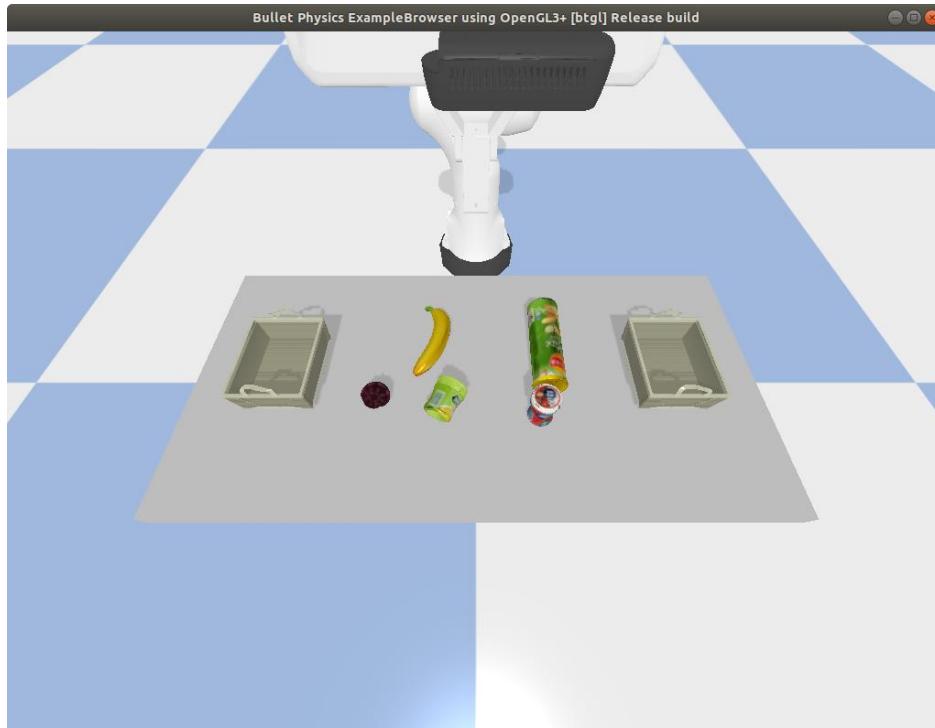
1-1-* SERIES: CONTINUED SUCCESS!!

- 5 out of 7 tasks improved! (out of which 1 is a known task, 5 are unknown!)

		improvement(%)		final error per object (cm)				successful pick-ups
task_index	Submission 1	Submission 2	default error per object (cm)	Submission 1	Submission 2	Submission 1	Submission 2	
1-1-3	4.6	8.2	33.69	32.14	30.9	3	3	
1-1-4	63.28	86	33.25	12.21	4.66	4	3	
1-1-5	66.23	85.84	32.92	11.12	4.66	3	3	
1-1-6	18.88	24.98	32.11	26.05	24.07	2	3	
1-1-7	61.61	-40.69	31.99	12.28	44.92	4	3	
1-1-8	-11.27	86.58	33.3	37.05	4.48	2	4	
1-1-9	78.01	78.65	31.35	6.89	6.69	4	4	

1-5-* SERIES

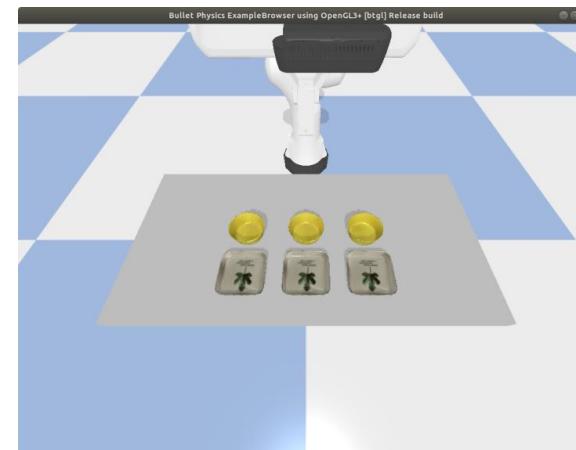
- We have had a reduction in improvement% w.r.t submission1.
- 1-5-3 had been the best scene in Submission 1.



task_index	Submission 1	Submission 2
1-5-2	24.73	21.97
1-5-3	82.36	39.87
1-5-4	-31.56	40.84
1-5-5	63.67	40.89
1-5-7	4.15	-0.74

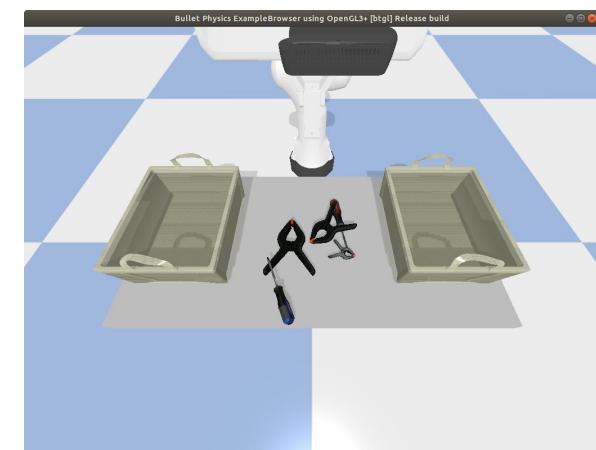
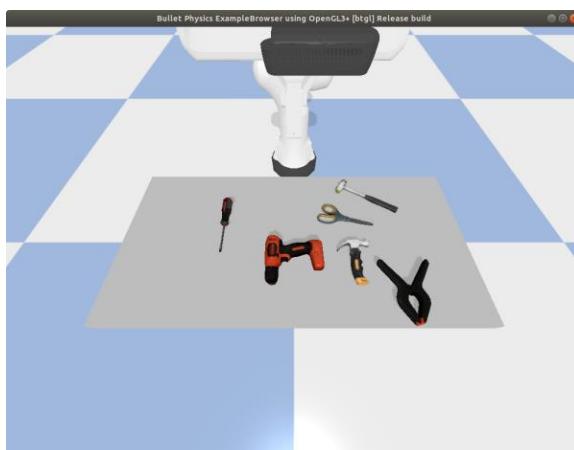
2-* SERIES

task_index	Scene	Success over sub 1	Success over baseline	Comments
2-1-*	Change in tray position needed	3/6 scenes	3/6 scenes	Needs checking since we have mixed results. Especially in one scene where it has picked 3 objects but has a -48% improvement.
2-2-*	Duplicate objects	0/6	-	Task planner had been failing. Vishal's fix will solve this.



3-* SERIES

task_index	Scene description	Success over sub 1	Success over baseline	Comments
3-1-*	Grasp poses from contact graspnet	4/5 scenes	5/5	The success is because of contact graspnet. We are able to pick 3-4 objects compared to 0-1 objects from normal graspnet.
3-2-*	Grasp poses from contact graspnet	1/5 scenes	4/5	It has scored lower than the prev submission but the reduction in improvement is less than 10%
3-3-*	Grasp poses from contact graspnet	3/5 scenes	3/5 scenes	Despite 2 scenes scoring less due to it not picking any object, it has shown a great improvement in terms of the number of objects picked

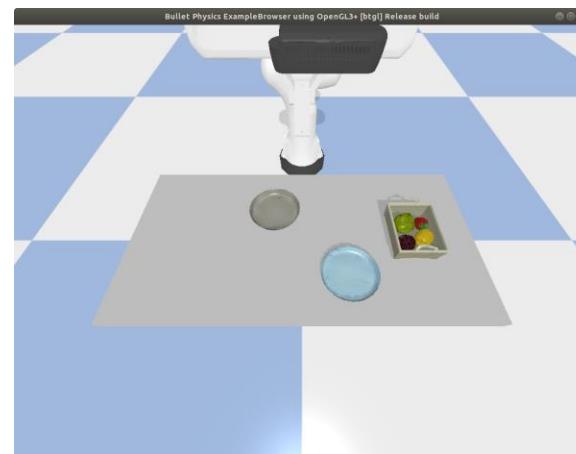
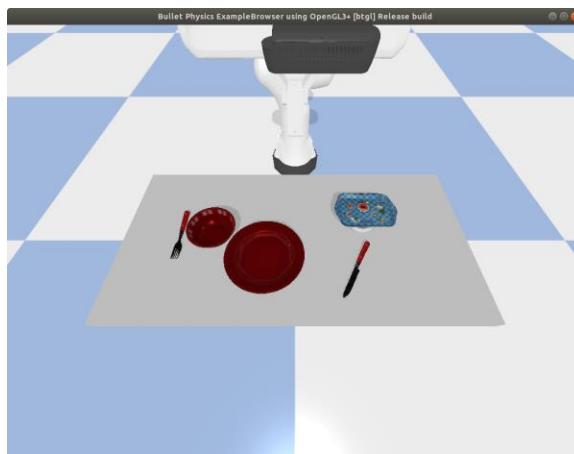


3-* SERIES

task_index	Submission 1	Submission 2
3-1-1	8.27	12.02
3-1-3	8.97	33.22
3-1-5	27.3	23.77
3-1-6	23.45	29.45
3-1-7	23.49	51.81
3-2-2	34.33	28.43
3-2-3	45.01	34.03
3-2-4	6.08	41.91
3-2-7	24.75	19.9
3-2-9	14.56	-10.3
3-3-1	33.45	-17.82
3-3-3	-9.2	20.38
3-3-5	-0.59	53.86
3-3-6	-79.3	53.64
3-3-8	40.01	-12.85

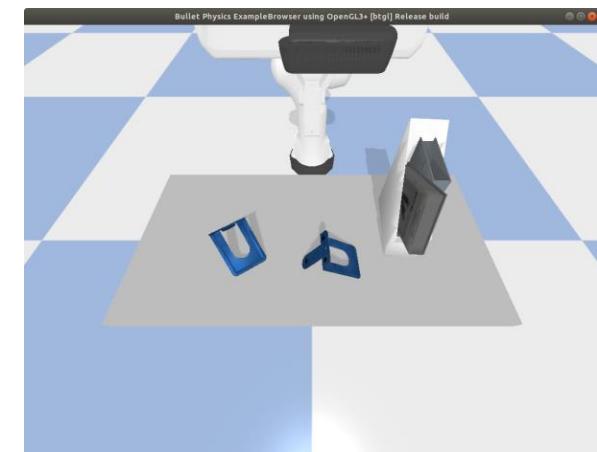
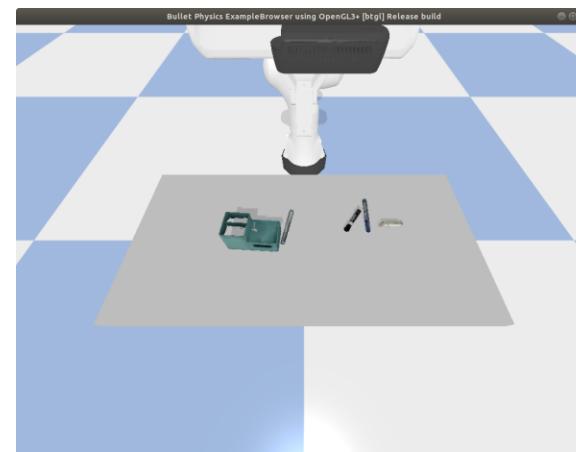
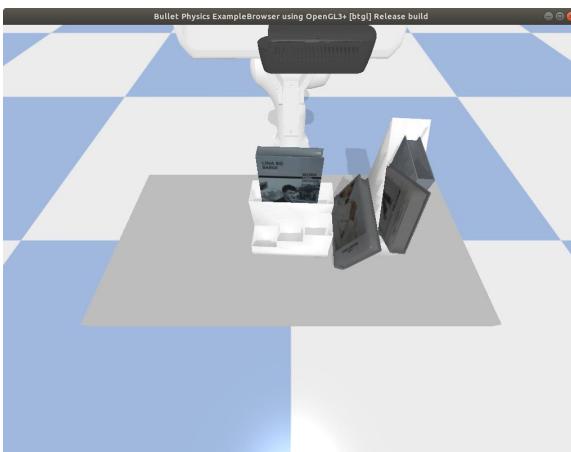
4-* SERIES

task_index	Scene	Success over sub 1	Success over baseline	Comments
4-1-*	Too many flat + 'difficult to pick' objects	4/7 scenes	4/7 scenes	Needs checking since we haven't been able to pick more than 1 object and in some scenes no object has been picked.
4-2-*	Buffer position + flat objects + pick from inside the tray	-	-	Hasn't picked any object in any of the scenes
4-3-*	Most difficult	-	-	Worst performance as it has scored lesser than the baseline while picking 1 object. [reaches -105% improvement]



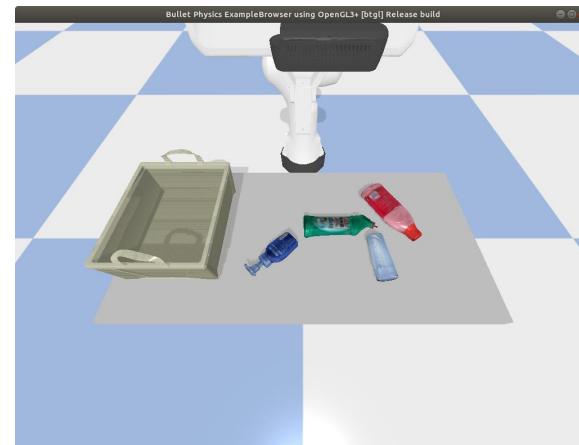
5-* SERIES

task_index	Scene	Success over sub 1	Success over baseline	Comments
5-1-*	Very difficult to pick without hitting the others	-	-	Has negative scores on all the scenes [-16.09 to -138%]
5-2-*	Grasp pose issue	4/5	4/5	Slight improvement because it has been able to 1 object on all the scenes [marker]
5-3-*	Needs orientation on the final object	-	-	Has negative scores on all the scenes. Has picked only one object in just one of the scenes.



6-* SERIES

task_index	Scene	Success over sub 1	Success over baseline	Comments
6-1-*	Needs buffer positions + grasp poses	3/4	3/4	The buffer position issue still exists but it is now able to pick objects. In one scene it has picked all the objects.
6-2-*	Grasp pose issue	4/4	4/4	Improvement because it is now able to pick 2/4 objects in the scenes
6-3-*	Needs orientation on the final object	1/5	1/5	Has negative scores on most scenes. It has picked 0 or 2 objects but has recorded a negative score. Needs investigation.



COMMENTS

1. Contact Graspnet has improved our scores [3-series]!
We might be able to score more with the tiny transformation change reqd in the predicted grasp poses.
2. We need to concentrate on 4 and 5 series majorly as they have recorded a huge –ve improvement and thus hindering our progress.
3. There are some scenes that need investigations.
4. The inclusion of buffer + stacking + orientation would improvise our scores.
5. Some scenes require pushing. Needs reconsideration..
6. Our avg error across the top 5 best scenes are better than the others. [but they might have been consistent across scenes]

Rank	Team	error (cm)	1st best task			2nd best task			3rd best task			4th best task			5th best task		
			task index	error (cm)	pick-up												
1	BUAA-GR	31.62	6-2-8	7.45	4	3-1-3	16.52		6-1-9	12.77	3	6-3-1	22.21	3	1-4-6	19.29	2
2	RRC	33.3	1-1-8	4.48	4	1-1-4	4.66		1-1-5	4.66	3	1-1-9	6.69	4	6-1-7	10.5	5
3	THU-SUN-LAB	34.84	1-1-7	7.81	4	1-1-5	9.65		3-3-3	8.47	0	1-5-2	13.53	4	2-1-1	18.38	4

Scene graph generation using Ray casting

Method is quite simple:

1. Create rays with their starting points lying on the given object's mesh
2. Let these rays run vertically downwards into the plane of the table-top
3. Detect the objects hit by these rays using Open3D's ray-casting functionality
4. All these objects will now be considered as the ones lying under the current object in the given scene

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 14th February 2022

Modified Perception Pipeline

Things to add

1. Add identity rotation to the object point cloud –
currently the object rotation was given according to the target pose.
2. Add projection loss –
currently the PointNet network has no strong feedback about the pcd points that need to light up.
3. Add background to the image –
currently the images were taken without a background, we have randomized the background for the images.
4. Add translation to the objects on the scene –
currently only rotation is added to the objects, add translation to the objects.
5. Add clutter – add more objects to the scene.
6. Detect exact correspondences –
we want to find the correspondences between the points and pixels in the point cloud and image, respectively.
7. Duplicate objects on the table

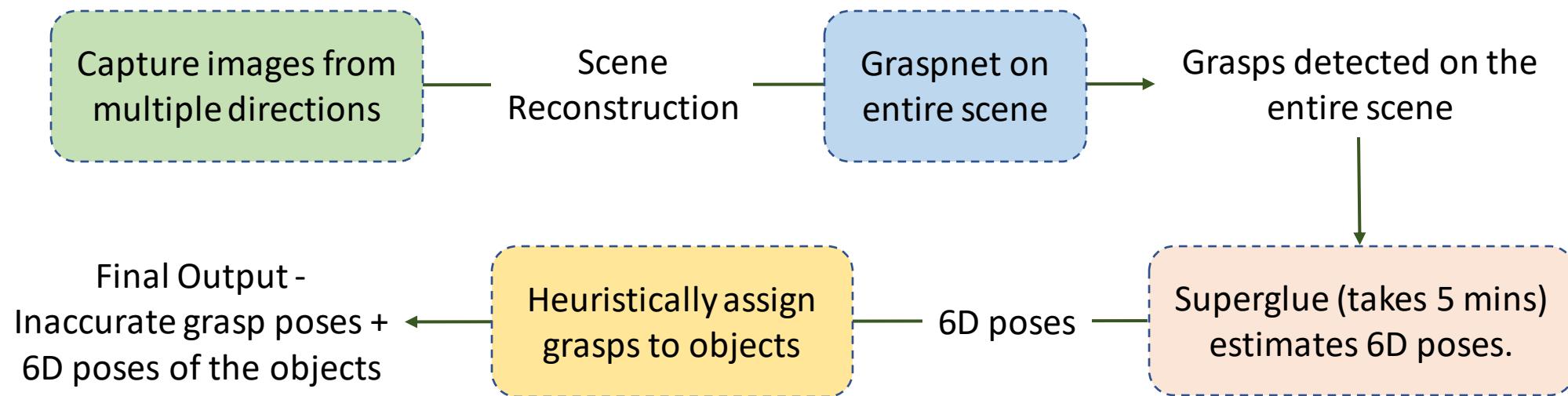
OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 5th February 2022

Modified Perception Pipeline

The current pipeline takes the following steps -

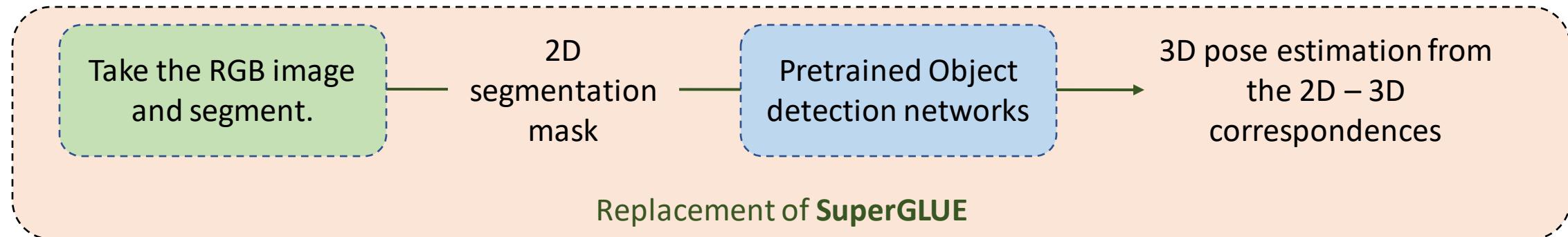


The current pipeline is takes longer (super glue 5 mins + reconstruction 2 mins) and is inaccurate under clutter and occlusion (for assigning grasps to the object as that is heuristics based on distance from centroid)

How can we make the pipeline faster and more accurate?

Modified Perception Pipeline

First alternate approach -



Advantage –

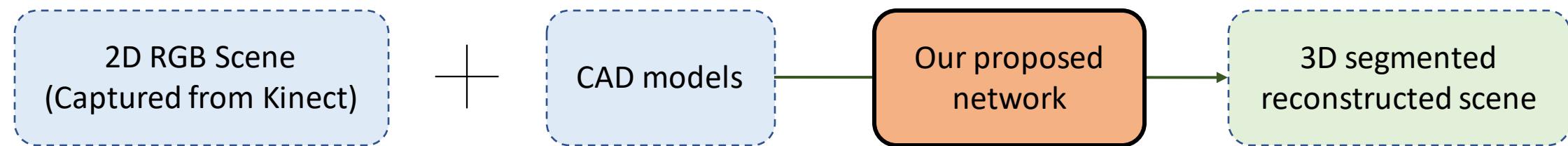
Faster than SuperGLUE

Problems –

1. This approach replaces the SuperGLUE part that takes about 3-5 mins to complete. However, this is more **inaccurate** as it can only give us the 3D pose.
2. If we don't have the 6D pose, we can't orient the object properly in the final pose!
3. We still need to reconstruct to get grasp poses from Graspnet and inaccurately assign grasp poses to objects.

Modified Perception Pipeline

Our proposed approach -



Advantage –

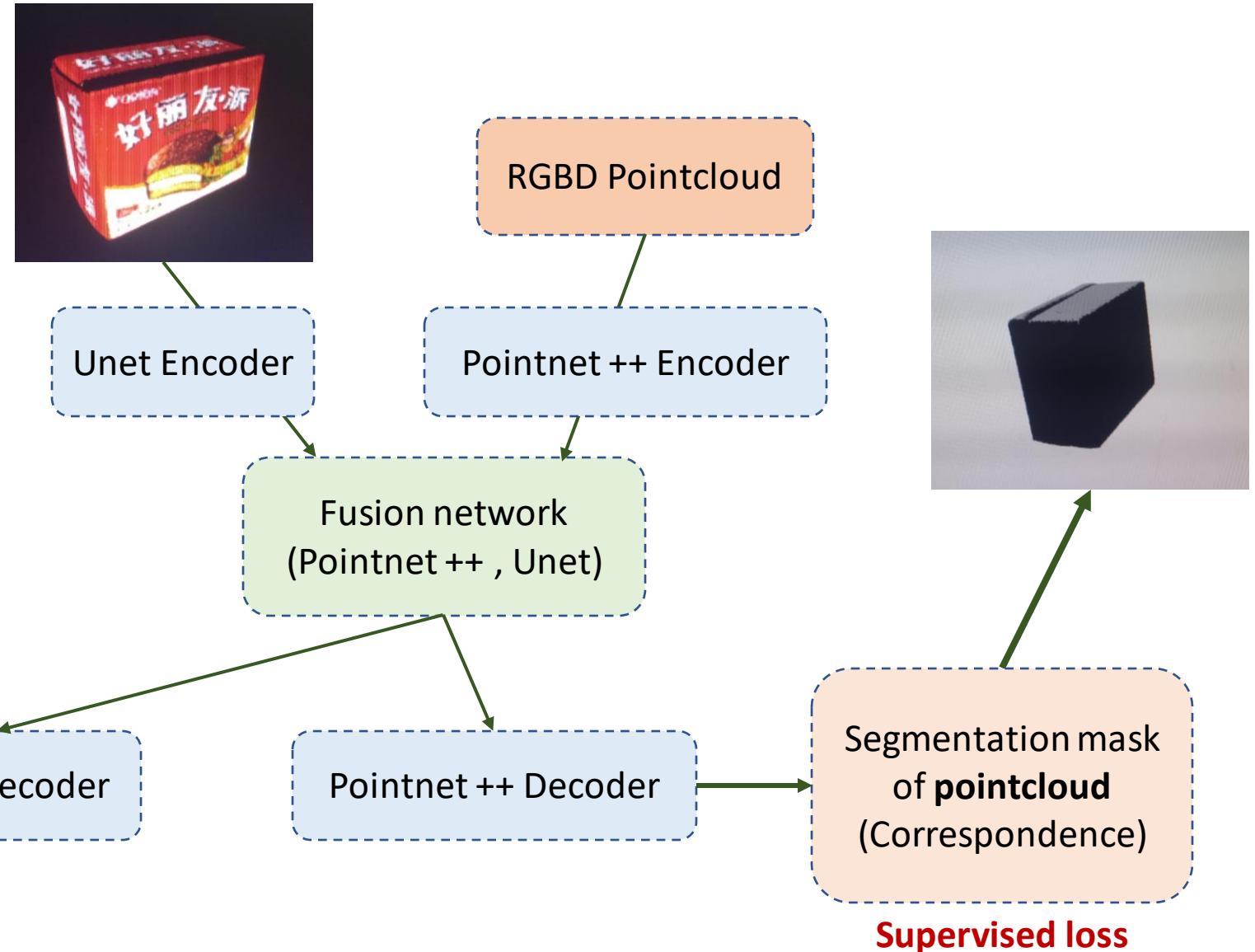
1. No need for capturing pointcloud from multiple directions to register the dense scene.
2. Single forward pass – faster than the current pipeline.
3. 3D segmented reconstructed scene –
 1. Constructed by estimating the 6D poses of the CAD models from the 2D image and placing them back to the scene.
 2. Grasp poses are thus detected for **individual objects** instead of the scene and thus is accurate.
4. Full 6D pose estimation instead of 3D (as the previous approach) – we can place the object in the final pose!

Modified Perception Pipeline – Proposed network

Supervised training!

All we need is –

1. Segmentation mask of the localized projection of the CAD model on the scene.
2. Masked pointcloud – points that were projected = 1 and rest = 0.



Task and motion planning

Limitations of hard-coded task planner:

1. Using occupancy maps to predict buffer spots takes considerable amounts of time (accurate measurements will be reported soon). We could make it more optimal.
2. Coding a planner to handle all possible scenarios is difficult – even, if possible, time complexity will be high – we need a spontaneous decision-making system based on previous experiences

Objective: Try to optimize the hard-coded rule-based planner as much as possible to test the limits and gaps in such planners in real-time applications. We shall then start filling the gaps and breaking the limits using a learning-based planner (Inspired by learning MPC paper which solves the time and limited predictive (myopic and hyperopic vs time tradeoff) power of MPC by simultaneously learning a value-based RL network for autonomous driving – apparently reached the friction limits of the car - [link to paper](#))

First steps:

1. Explore scene graphs to predict stacking
2. Implement scene-graph in the rule-based planner

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 22nd January 2022

MAJOR UPDATES

1. Submission plan for January – 25th January [Tuesday]
2. Leaderboard [minor analysis]
3. Vishal has new Task Planner – solves stacking and buffer places [Level 5]
4. Segmentation maps – redefining the perception stack.
5. [hack] Exploration of Sapien simulator using the rest pose + 6 DOF graspnet code.
6. Real Robot access?

SUBMISSION PLAN FOR 25 JAN

We will make **3 submissions** on **25th Jan [Tuesday]**:

2 simulation + 1 real robot

3 variants of submission:

1. Simulation Phase

1. Rest Pose (removing weird movements) + Contact Graspnet
2. Rest Pose (removing weird movements) + 6 DOF graspnet

Contact graspnet has a minor error in transformation that changes task-to-task. We need to debug this case more. However, with two different submissions, we want to make sure if we good gains on switching to Contact Graspnet. An overall comparison will help us chose.

2. Real Robot

Rest Pose + 6 DOF graspnet + (changes made specifically for sim)

Since we have removed weird movements due to the rest pose, we are expecting the real robot to also work with our submissions as the planner shouldn't fail.

DECEMBER LEADERBOARD

Simulation Result

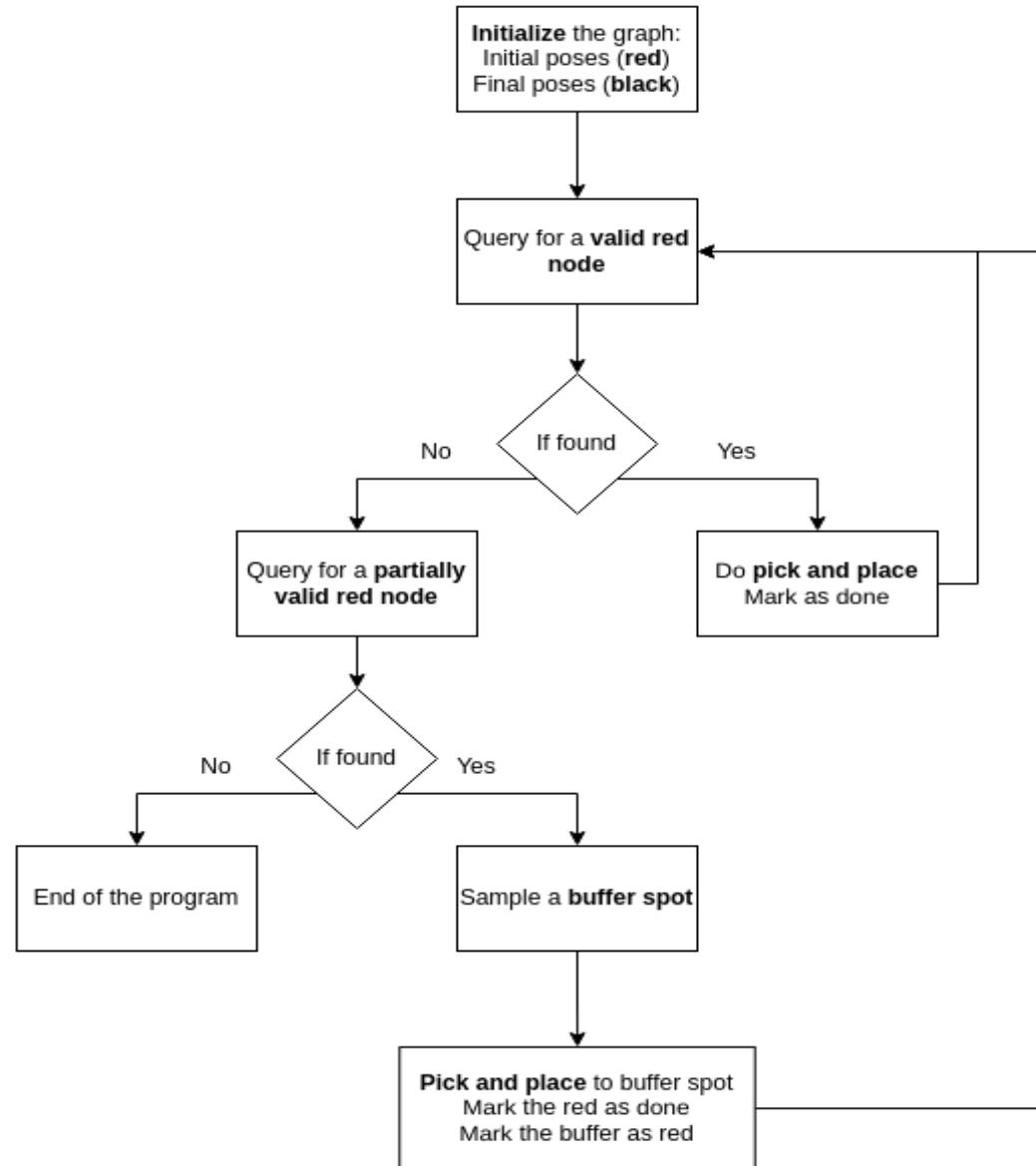
In the Simulation branch, each team is allowed to submit twice per month. Evaluation will be carried out within one week after the submission. Result updated on 2022/1/11.

Main Track				Perception Track				Planning Track						
Rank	Team	error (cm)	1st best task			2nd best task			3rd best task			4th best task		
			task index	error (cm)	pick-up	task index	error (cm)	pick-up	task index	error (cm)	pick-up	task index	error (cm)	pic
1	BUAA-GR	31.64	3-1-6	9.41	1	2-1-9	14.35	5	6-1-1	8.77	3	1-4-2	13.98	3
2	RRC	34.48	1-5-3	6.26	4	1-1-9	6.89	4	1-1-5	11.12	3	1-5-5	13.47	4
3	Wuhan & Chalmers	35.13	1-1-5	7.47	3	1-1-4	13.15	3	1-5-2	19.38	3	1-1-8	19.9	2

Rank	Team	error (cm)	task index			error (cm)		pick-up	
			task index	error (cm)	pick-up	task index	error (cm)	pick-up	task index
1	BUAA-GR	31.64	3-1-6	9.41	1				

1. The other teams have good 1-series results like us.
2. BUAA-GR has one exception – one good result from the 3-series. But they have **picked only 1 without disturbing the others.**
3. The other teams have used the Sapien simulator. So, we gave that a try.

Heuristic Graph-based Task Planner



[Link](#)

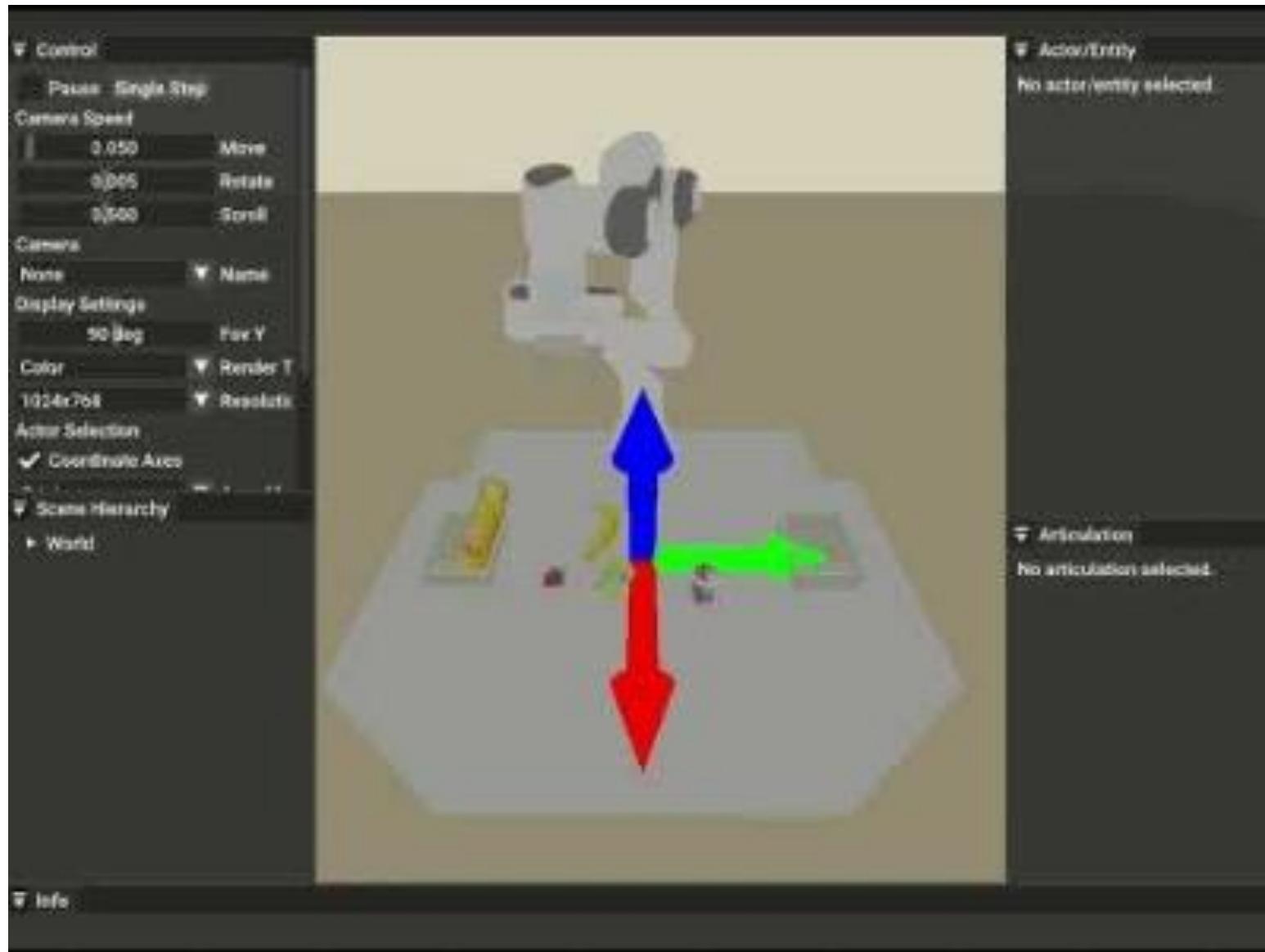
SAPIEN SIMULATOR

The other 2 teams who had submitted had used the Sapien simulator. So, we had run our codes on it.

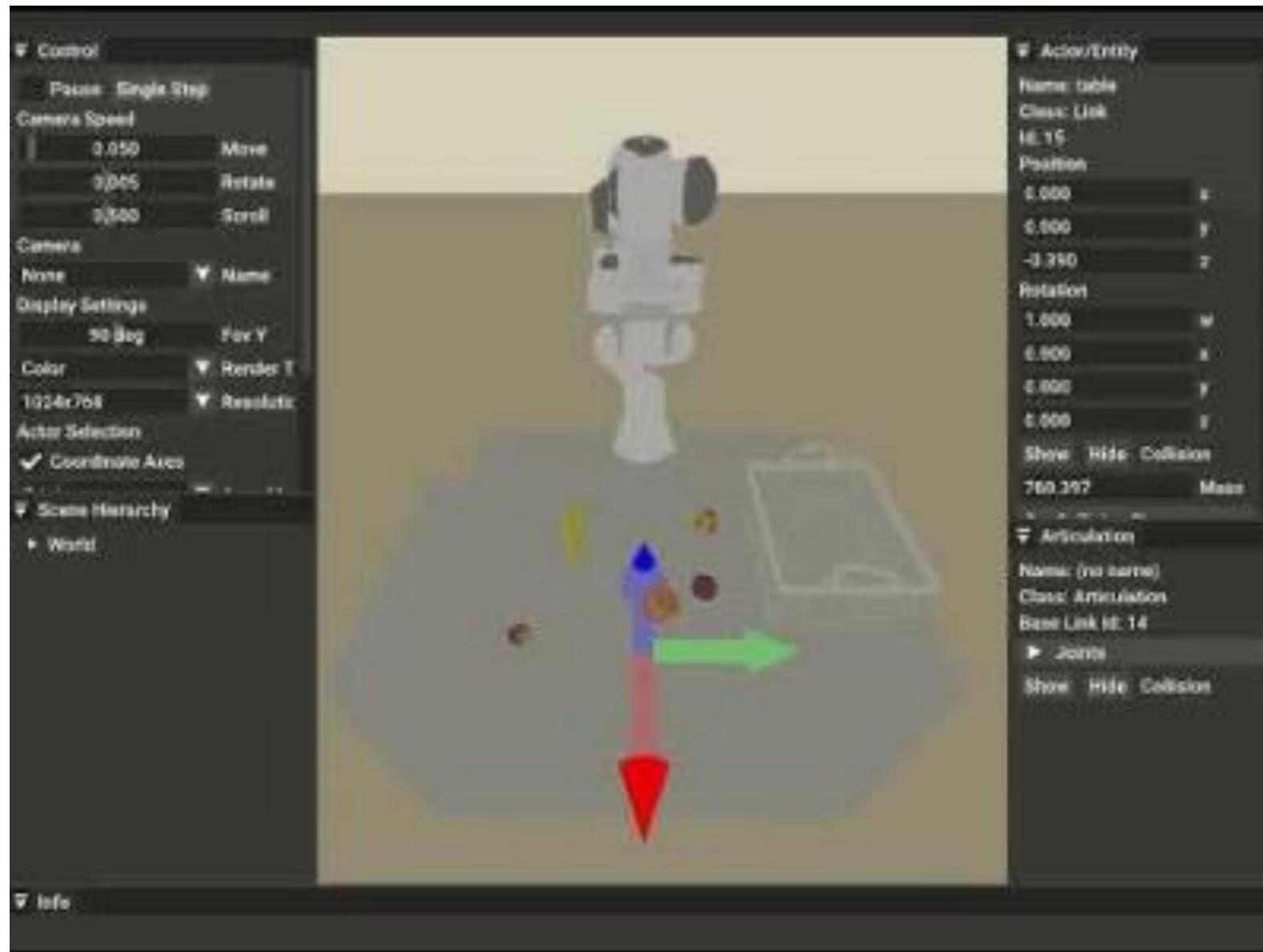
Major observations:

1. Sapien might be more useful with RL (based on its website)
2. The physics doesn't seem to be as good as Pybullet [advantage?]
 - a. The objects seem to stick to the table or the gripper during pick and place, thus reducing the chances of the irregular objects slipping away.
3. Video demos in the next slide!
4. Hack?
 1. Towards the end, after solving major issues, we could run ours on Sapien. The objects won't roll off or bounce off the table or slip from the gripper.
 2. Contact graspnet would need a frame change if we use it with Sapien
 3. We need to run more tests and confirm point (a)

SAPIEN SIMULATOR – DEMO 1

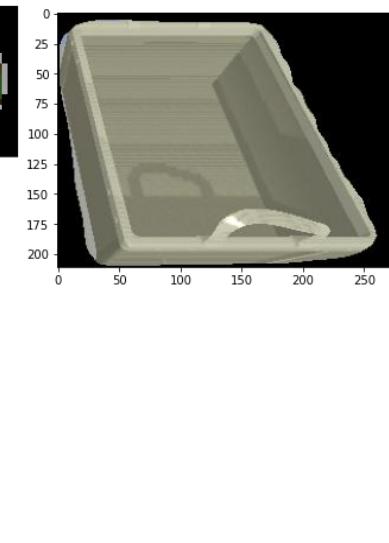
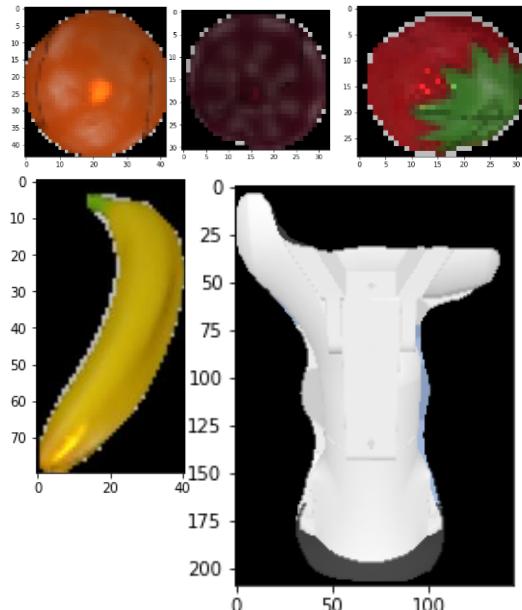


SAPIEN SIMULATOR – DEMO 2



SEGMENTATION MAPS

- We are able to perform object matching using object instance segmentation and 5 different pretrained deep feature extractors using most matches.
- We will integrate the network for object grasping and object feature matching this week.



NEXT WEEK

1. Real Robot access? – Arun sir replied.
2. Submissions on 25th Jan.
3. Integrating and testing new task planner.
4. Integration segmentation mapping technique
 1. Grasp verification
 2. Replacing SuperGLUE
5. Explore scene graphs and other methods that help us better understand the target scenes
(mainly stacking)

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 15th January 2022

MAJOR UPDATES

This week, we have fixed the following -

1. Obtained logs for the real robot submission.

2. Contact Graspnet Transformation – Contact Graspnet is integrated now.

With this fix, we are now able to pick objects in 3-3 series which wasn't possible in the previous versions

3. Time delay in joint movements is gone.

This gives us advantage in terms of time.

4. Robot is not making weird movements anymore.

This avoid wrecking the current set of objects and avoids unnecessary collisions with the environment

Ongoing fixes -

1. Grasp verification using segmentation map + feature similarity obtained through a neural network.

2. Better motion planners and optimizers.

3. Fixing minor bugs.

REAL ROBOT LOGS

Following are the comments from the OCRTOC team -

We are glad to tell you that your solution currently works in simulation tests. However, it faced some problems when tested with a real robot. There are mainly the following problems:

1. **The robot driver throws out an error when trying to pick an object, since the planning solution send a gripper command less than 0 (-0.1).** This can be solved by changing the command value to 0 or a number close to 0. We fix the problem for you, so the test can be continued.
2. **The robot often reaches the joint limits when try to catch an object.** There is no such problem in simulation test, because joint limits are not set in simulation. However, the **real robot has certain joint limits** for each joint.
3. **The robot reaches the grasp pose, close the gripper and open the gripper a few seconds later without moving the arm** (In the simulation environment, it would pick the object and place it in a box). Then it moves to the grasp pose of next object and repeat the motion.
4. Sometimes, the robot reaches a position and stops. This may be caused by the robot controller. We meet this problem only for a few times.

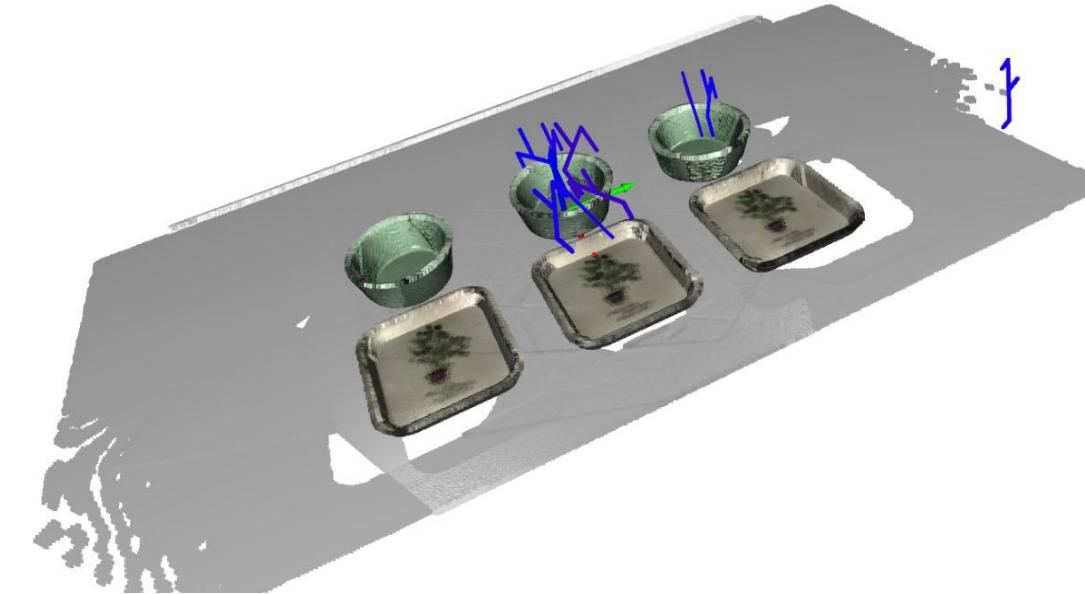
The robot meets the problem (3) and then meets the problem (2), then the solution stops in this test.

CONTACT GRASPNET – COMPARISONS (NO SEGMENTATION MAPS USED)

Contact
Graspnet

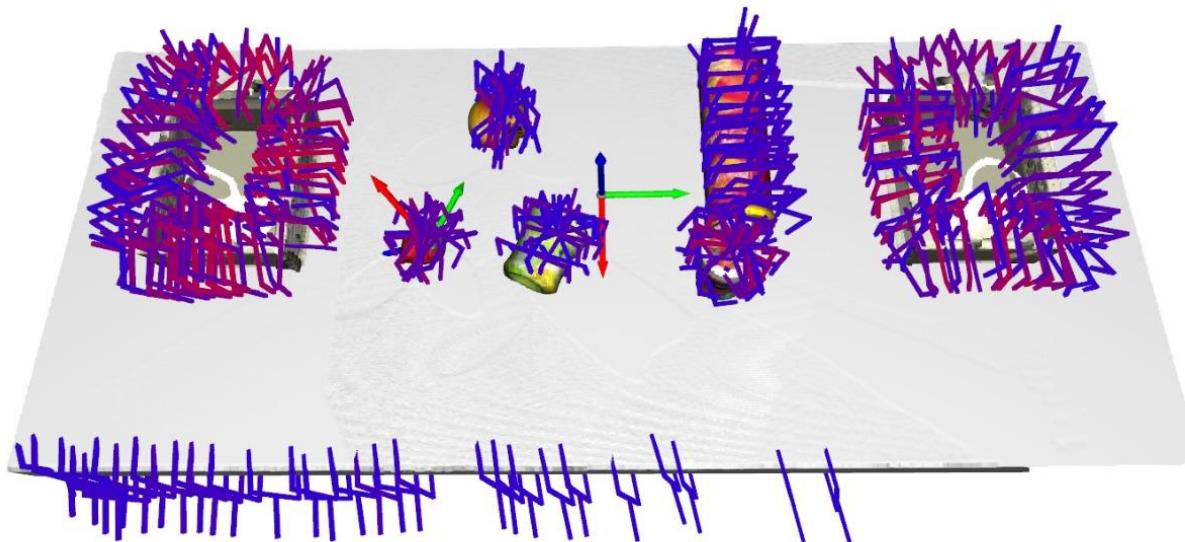


Baseline

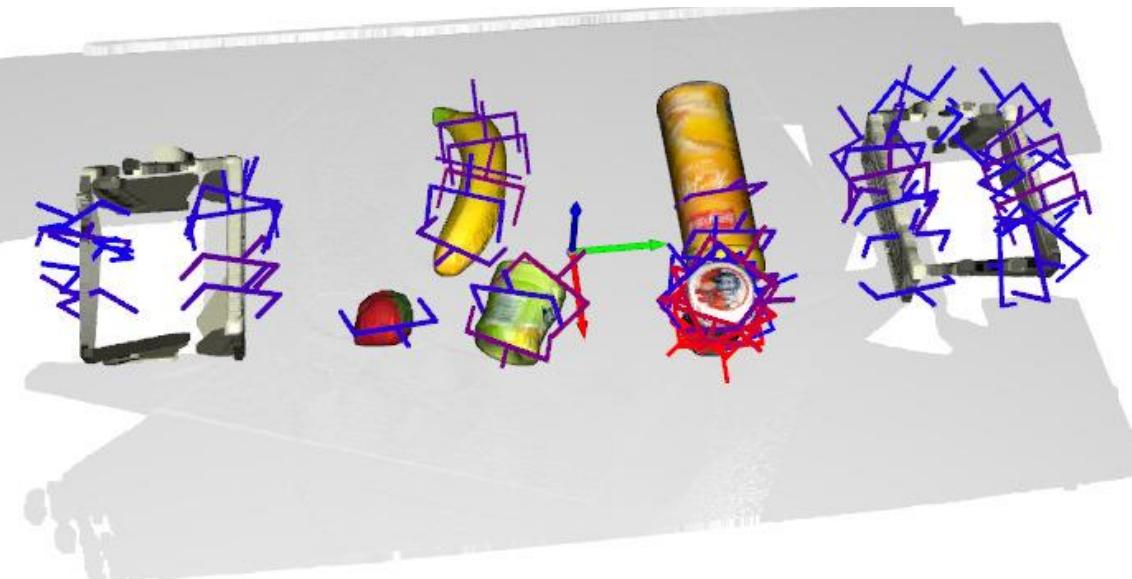


CONTACT GRASPNET – COMPARISONS (NO SEGMENTATION MAPS USED)

Contact
Graspnet

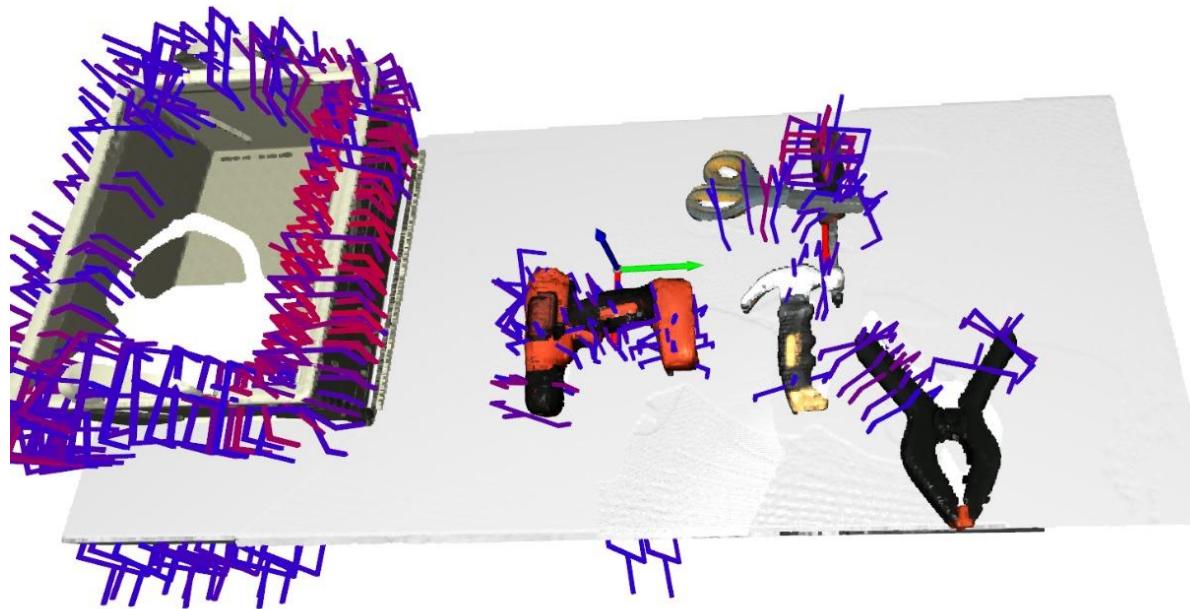


Baseline

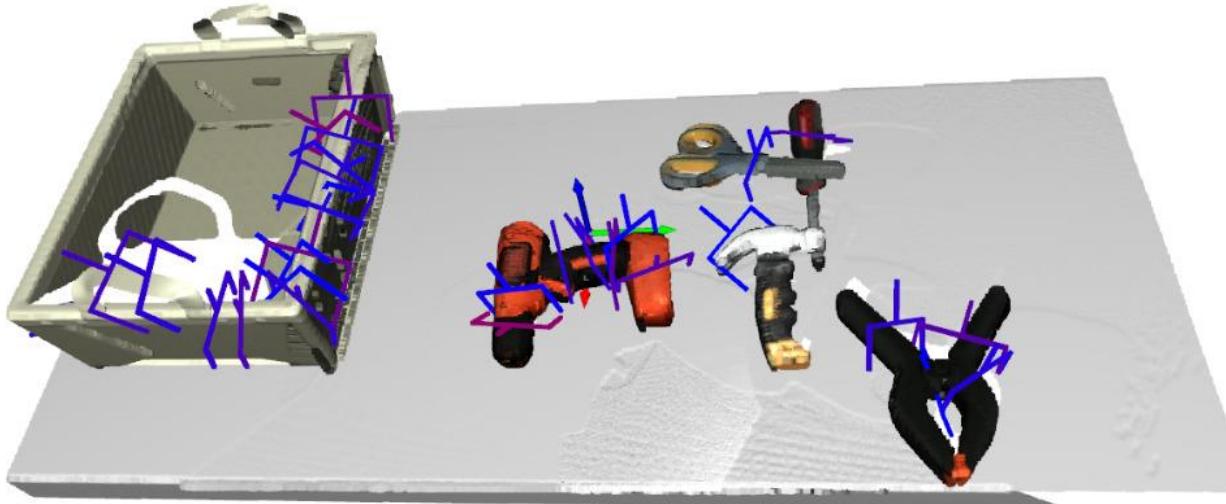


CONTACT GRASPNET – COMPARISONS (NO SEGMENTATION MAPS USED)

Contact
Graspnet

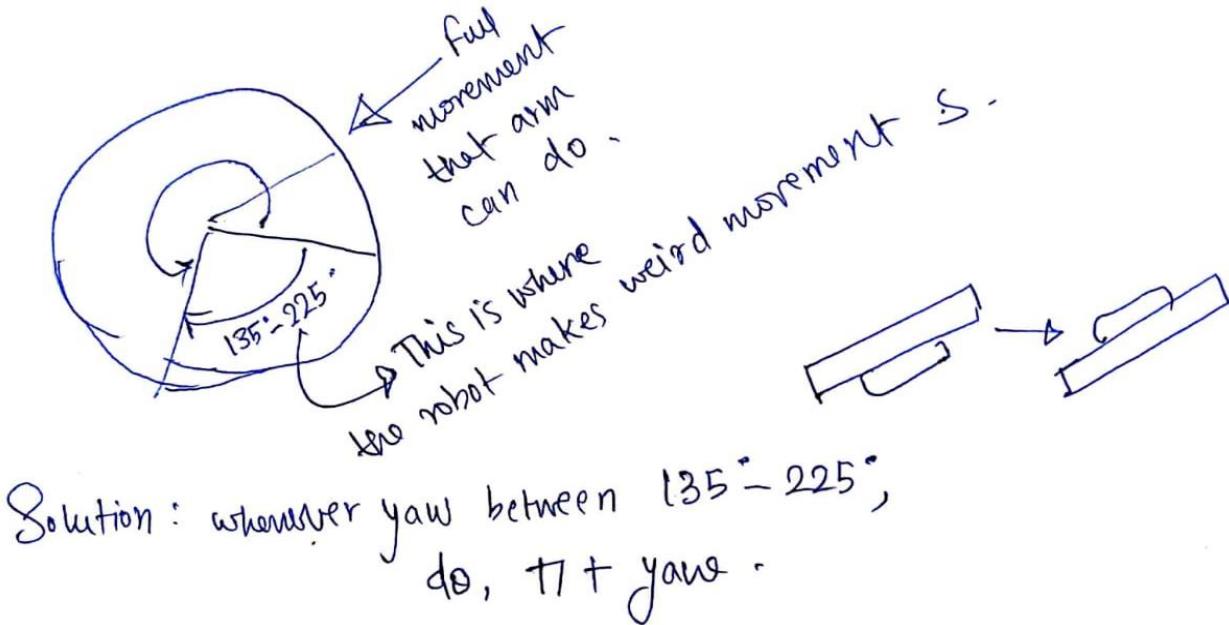


Baseline



ROBOT IS NOT MAKING WEIRD MOVEMENTS ANYMORE.

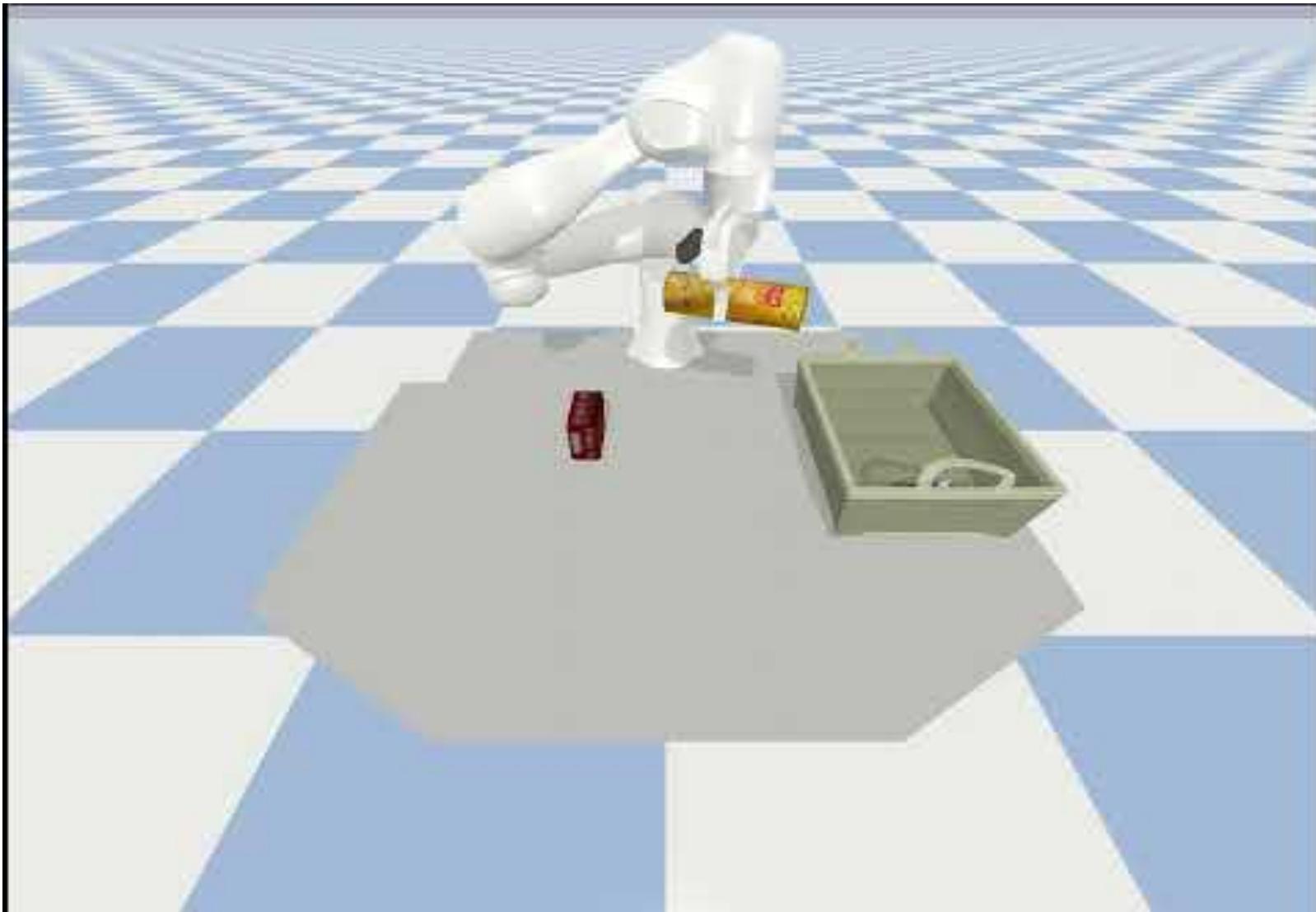
1. Since the entire plan is planned in cartesian space, to continuously change the joint movements, the robot is getting tangled up and making weird movements that is wrecking the entire scene.
2. Current Strategy we are now following:
Rest Pose -> Exit Pose -> Grasp -> Pick -> Exit Pose -> Rest pose -> Rest pose -> Entrance Pose -> Place
3. We are allowing the robot to disentangle in the joint space by coming back to a rest pose. This allows the robot to start from scratch.
4. Another change is have done
in the movements is made
sure that robot doesn't have
to turn too much in the yaw
direction.



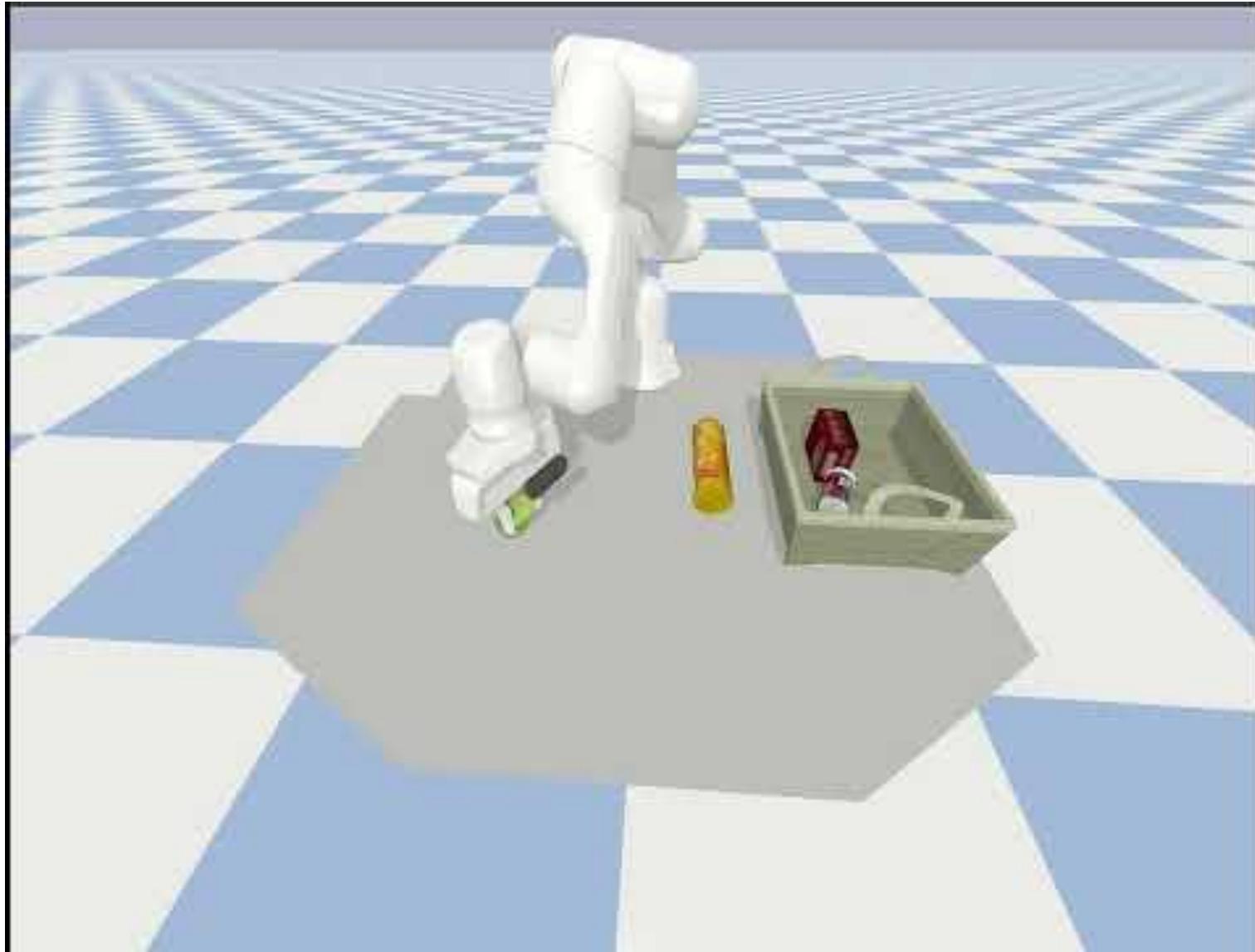
TIME DELAY IN JOINT MOVEMENTS IS GONE

1. To avoid weird movements, we are going to a rest pose after every pick and place. However, this is causing some delay as the movement in joint space is very slow and the movement to home-pose is in joint space.
2. Our solution – go to x-y-z of the home pose in cartesian space and then realign in the joint space.
3. In the next page, we have the before – after demonstration of the following changes -
 1. If the robot didn't go to the home pose, then the robot is making very weird movements (this include the fix of making weird movements without the home pose + without the yaw rotation fix).
 2. If we don't fix the time delay, then the joint movement is very weird.

PREVIOUS WEIRD MOVEMENTS



CURRENT WORKING DEMONSTRATION



WEIRD MOVEMENTS FIX WITH TIME DELAY



TO-DO: THIS - WEEK

IMPROVING MOTION PLANNING AND OPTIMIZERS

To plan smooth trajectories for the robot arm, we are exploring optimization-based motion planners.

Moveit mentions the below combinations (we are planning to explore)

1. **OMPL as a pre-processor for CHOMP:** OMPL as a base planner to produce an initial motion plan which can act as an initial guess for CHOMP. **CHOMP can then produce optimized paths.** In most cases, the quality of such a path produced should be **better than** that produced by **OMPL alone or CHOMP alone.**
2. **OMPL as a pre-processor for STOMP:** As **stomp can used as a smoothing algorithm, it can be used to smoothen the plans produced by other motion planners.** OMPL first produces a path, STOMP can then generate a smoothed version of that path. Such a path in most cases should be better than a path produced by either just OMPL or STOMP alone.
3. **STOMP as a pre-processor for CHOMP:** For this case, a path can be initially produced by STOMP, CHOMP can then take this as an initial guess and produce an optimized version of the smoothed path produced by STOMP.
4. **CHOMP as a pre-processor for STOMP:** CHOMP can be used to produce a path and then **STOMP can be used to smoothen the path.** This helps in getting rid of the jerky motion of the trajectories produced by CHOMP alone in the presence of obstacles.

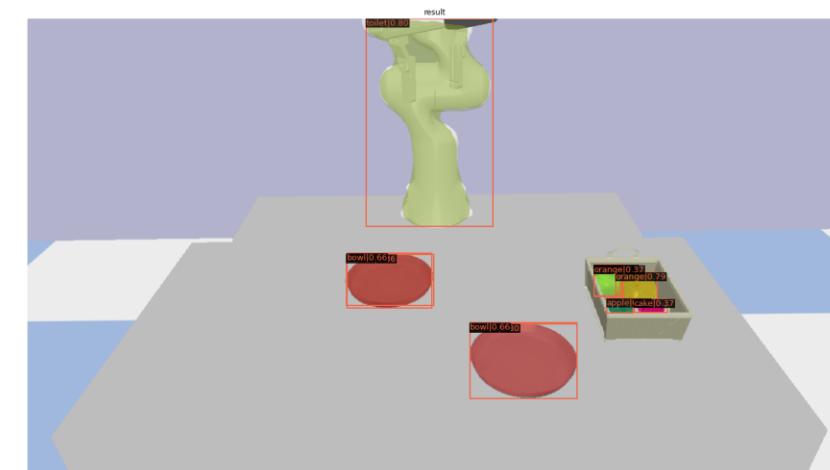
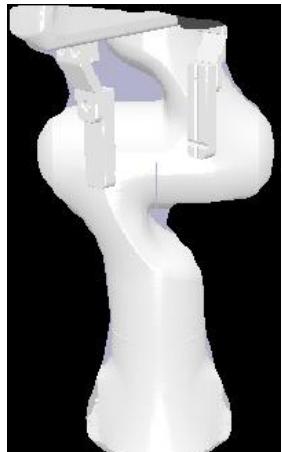
SEGMENTATION MAPS FOR FEATURE MATCHING AND GRASP VERIFICATION

We think that the segmentation map can be used for two things:-

Object grasp verification –

- Most observed case** – We observe that the gripper picks the incorrect object in environments with clutter. Although the grasp poses have been accurately assigned and generated, the gripper goes to the object pose of the correct object and inadvertently picks other objects in the clutter which are closer to the desired object.
- Other cases** – When we generate grasp poses over point clouds, due to the grasp pose assignment logic (assign grasp pose to object with the closest centroidal distance), the gripper goes to pick another object which was incorrectly assigned the grasp/object pose.

Solution - We can pick the objects to a certain height and take a picture using the Kinect. The segmentation map of the image taken by the Kinect can be used to verify if the correct object has been picked up.



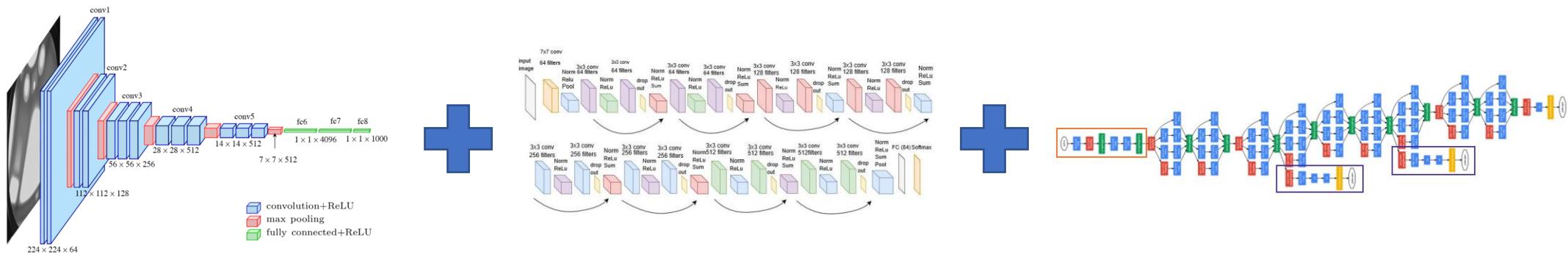
SEGMENTATION MAPS FOR FEATURE MATCHING AND GRASP VERIFICATION

Feature Matching (Replacing SuperGLUE):

Currently, feature matching is done to assign the object to the desired target using SuperGLUE. SuperGLUE takes a considerable amount of time to match the features between the objects in the scene and the desired target pose.

Solution:- To obtain accurate feature matches, we use an Ensemble network comprising of multiple pretrained deep feature extractors, to assign the correct pose to the segmented object.

We can use multiple views of the target object and generate a confidence score over all objects in the scene.



Scores generated from multiple networks:- Orion box: 0.75, Doraemon plate: 0.63, Clear tray: 0.33
Object assigned: *Orion box*

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 8th January 2022

MAJOR UPDATES

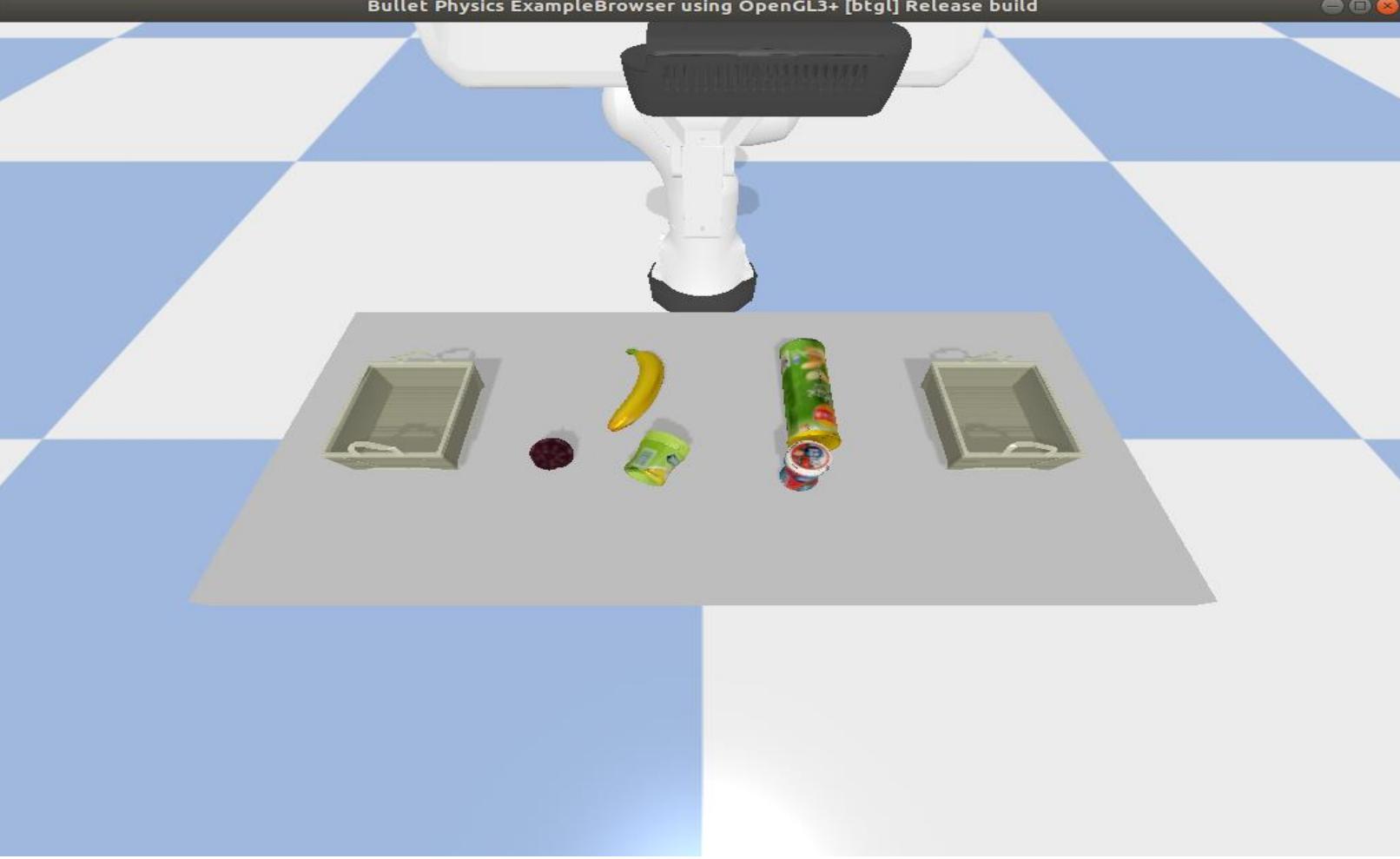
1. OCRTOC real robot logs
2. We have received the submission results of the second submission (only simulation) we made on Jan 1, 2022. The submission was considered in the January window – we have mailed them regarding this.

ANALYSIS (2ND SUBMISSION)

Total number of scenes	100
Public Tasks	30
Hidden (Unknown) Tasks	70

In the Hidden Tasks:

- At each level of hidden tasks, the objects known to us have been used but the arrangement is different.
- The objects are the same as those in the corresponding known public scenes.



BEST SCENE

task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups	objects picked up
1-5-3	6.26	35.5	82.36	4	<code>['suger_1_v1', 'suger_2_v1', 'plastic_plum_v1', 'plastic_banana_v1']</code>

RED: PUBLIC TASKS

Total number of scenes	30
------------------------	----

SUCCESS:

Number of Scenes where we have succeeded (i.e. > 10% improvement)	10
Number of Scenes with > 1% improvement	16

BLUE: HIDDEN TASKS

Total number of scenes	70
------------------------	----

SUCCESS:

Number of Scenes where we have succeeded (i.e. > 10% improvement)	27
Number of Scenes with > 1% improvement	37

1-1-* SERIES: A BIG SUCCESS (PROBABLY OVERFIT :))

- 6 out of 7 tasks improved! (out of which 1 is a known task, 6 are unknown!)

task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups
1-1-9	6.89	31.35	78.01	4
1-1-5	11.12	32.92	66.23	3
1-1-4	12.21	33.25	63.28	4
1-1-7	12.28	31.99	61.61	4
1-1-6	26.05	32.11	18.88	2
1-1-3	32.14	33.69	4.6	3
1-1-8	37.05	33.3	-11.27	2

1-5-* SERIES: A BIG SUCCESS

- 4 out of 5 tasks improved! (2 are known and 3 are unknown)

task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups
1-5-3	6.26	35.5	82.36	4
1-5-5	13.47	37.08	63.67	4
1-5-2	26.61	35.36	24.73	3
1-5-7	19.93	20.79	4.15	4
1-5-4	34.38	26.13	-31.56	4

2-1-* SERIES: SUCCESS IN PICKING OBJECTS

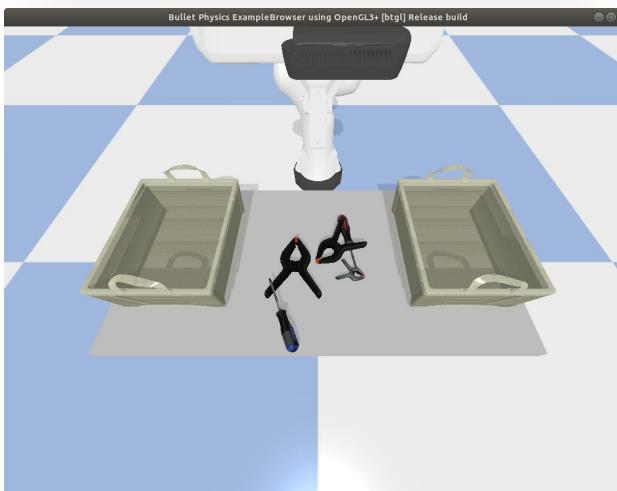
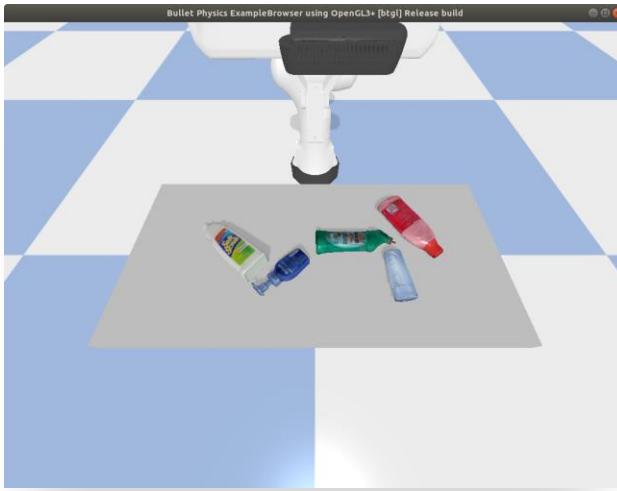
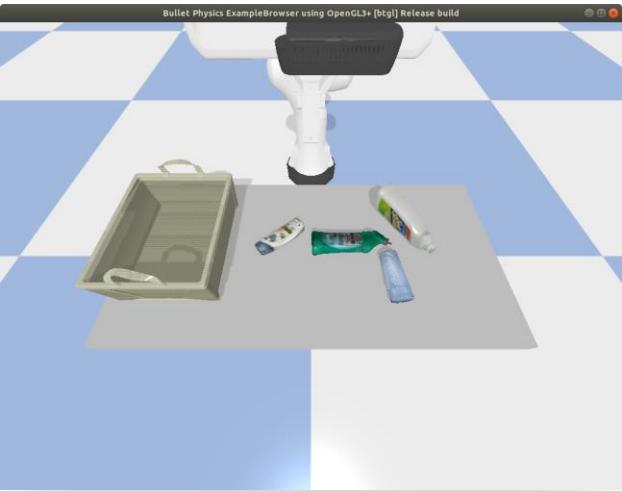
- In all the tasks, we were able to pick most of the objects successfully
- 3 out of 6 tasks had big success
- Failure cases suspect: Tray pushed off the table
- Only 1 known, Others unknown!

task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups
2-1-7	26.74	39.12	31.64	4
2-1-1	36.05	45.32	20.46	2
2-1-9	32.57	35.82	9.06	2
2-1-4	35.9	35.38	-1.47	2
2-1-3	41.08	35.66	-15.21	2
2-1-5	53.16	31.2	-70.38	4

RED: SURPRISES

- Surprise: These objects were not picked properly during our tests in the lab

task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups	objects picked up
6-2-3	26.96	43.29	37.72	2	['conditioner_v1', 'magic_clean_v1']
3-2-2	32.34	49.24	34.33	1	['power_drill_v1']
3-3-1	18.17	27.31	33.45	1	['phillips_screwdriver_v1']
6-3-2	32.9	43.3	24.02	1	['magic_clean_v1']
6-1-1	17.31	21.59	19.8	1	['cleanser_v1']
6-3-1	35.44	43.37	18.29	2	['cleanser_v1', 'magic_clean_v1']
3-1-1	28.39	30.95	8.27	1	['flat_screwdriver_v1']



Success rate was low in our tests

task_index	objects picked up
LU: 6-2-3	['conditioner_v1', 'magic_clean_v1']
LD: 3-2-2	['power_drill_v1']
RD: 3-3-1	['phillips_screwdriver_v1']
RU: 6-3-1	['cleanser_v1', 'magic_clean_v1']

BLUE: SURPRISES

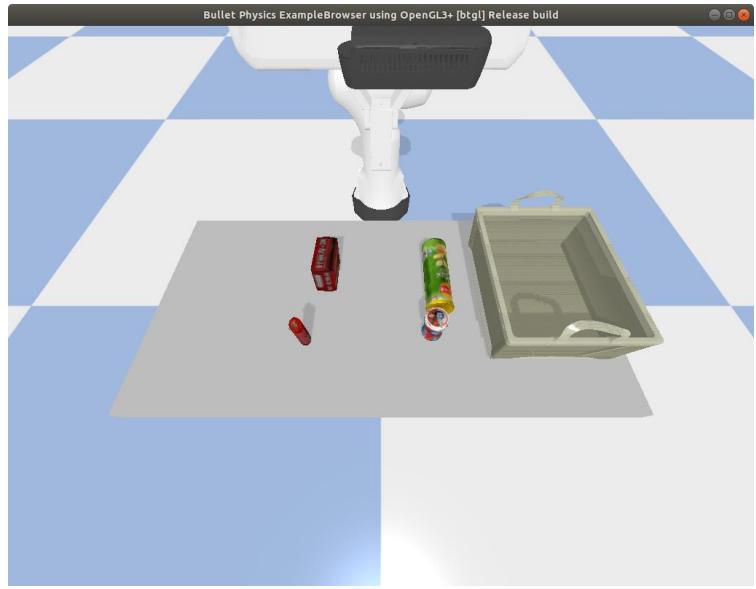
6-1-9	18.27	34.25	46.66	2		['soap_v1', 'soap_dish_v1']
3-2-3	20.18	36.71	45.01	1		['power_drill_v1']
3-3-8	17.84	29.74	40.01	2		['phillips_screwdriver_v1', 'flat_screwdriver_v1']
6-1-3	20.37	31.28	34.88	2		['conditioner_v1', 'soap_v1']
6-1-7	25.93	35.77	27.52	1		['toothpaste_1_v1']
3-1-5	25.41	34.96	27.3	1		['power_drill_v1']
4-2-7	22.18	29.28	24.25	2		['plastic_lemon_v1', 'plastic_plum_v1']
3-1-7	25.24	32.98	23.49	1		['power_drill_v1']
3-1-6	20.6	26.91	23.45	1		['phillips_screwdriver_v1']
1-2-9	32.05	39.17	18.18	2		['lipton_tea_v1', 'pudding_box_v1']
1-2-4	36.71	41.63	11.83	1		['lipton_tea_v1']
4-3-9	31.39	33.73	6.94	2		['round_plate_4_v1', 'round_plate_3_v2']
6-3-9	24.08	24.77	2.75	2		['conditioner_v1', 'repellent_v1']

RED: FAILURE DATA

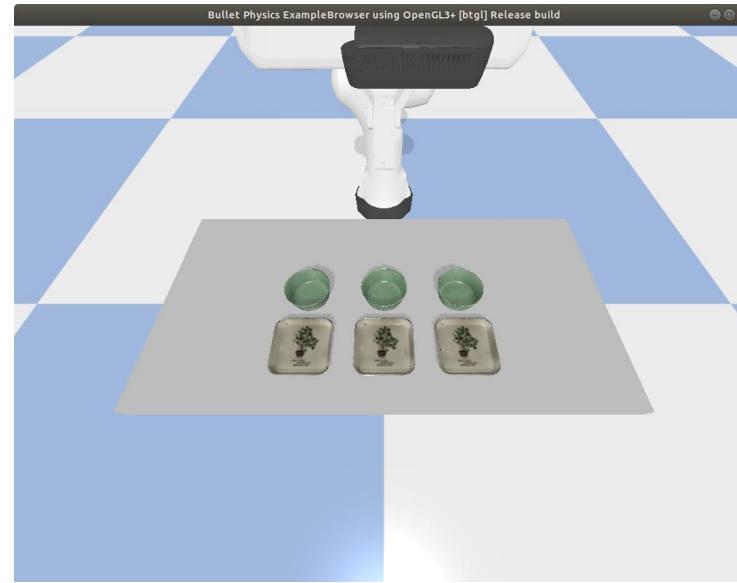
Number of Scenes where we have done really bad (< 1% improvement)	16
--	-----------

UNEXPECTED FAILURE:

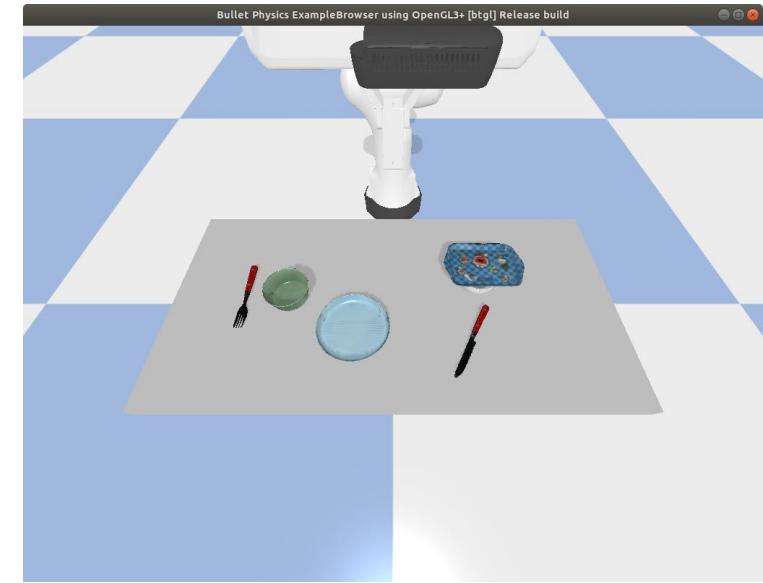
task index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups	objects picked up
1-1-3	32.14	33.69	4.6	3	['orion_pie_v1', 'suger_2_v1', 'suger_3_v1']
2-2-1	36.33	37.89	4.12	1	['green_bowl_v3']
4-1-2	21.08	17.58	-19.9	0	[]



1-1-3



2-2-1



4-1-2

- These scenes were found to be partially successful when executed on our server, but they failed on the competition platform
- 1-1-3 possible reason for failure: Tray falling off the table
- 2-2-1 – It used to pick the bowls before, but on the competition platform, nothing happened
- 4-1-2 – It never used to pick the doraemon bowl on our server. It used to pick other objects though. But on the competition platform, it only picked doraemon in each of the scenes tested

BLUE: FAILURE DATA

UNEXPECTED FAILURES:

2-2-9	36.12	35.95	-0.48	0	[]
2-1-4	35.9	35.38	-1.47	2	['plastic_peach_v1', 'plastic_orange_v1']
1-1-8	37.05	33.3	-11.27	2	['orion_pie_v1', 'suger_2_v1']
2-2-7	55.15	47.97	-14.97	0	[]
2-1-3	41.08	35.66	-15.21	2	['plastic_plum_v1', 'plastic_pear_v1']
6-3-4	37.88	29.74	-27.38	0	[]
1-5-4	34.38	26.13	-31.56	4	['suger_2_v1', 'plastic_apple_v1', 'potato_chip_1_v1', 'plastic_strawberry_v1']
1-4-5	51.88	37.58	-38.05	2	['pink_tea_box_v1', 'green_cup_v1']
6-2-6	56.97	39.24	-45.2	3	['bleach_cleanser_v1', 'cleanser_v1', 'shampoo_v1']
6-2-8	56.15	36.89	-52.2	1	['magic_clean_v1']
2-1-5	53.16	31.2	-70.38	4	['plastic_plum_v1', 'plastic_strawberry_v1', 'plastic_apple_v1', 'plastic_peach_v1']
6-3-7	56.6	29.41	-92.45	1	['cleanser_v1']
4-1-9	54.2	27.19	-99.32	1	['doraemon_bowl_v1']

POSSIBLE REASONS FOR FAILURE

In 1-1-, 1-5-, 2-1-, 6-2- series:

The results are really bad despite picking most of the objects.

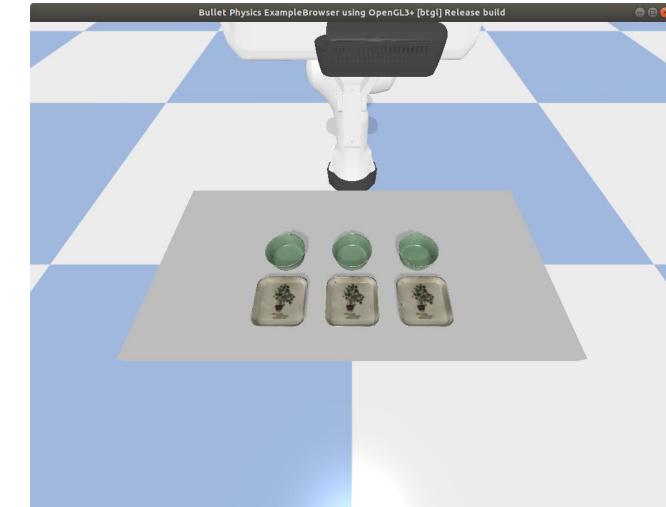
- **Common point :** Aim is to place it inside the tray
- **Reason (might be):** We are placing the objects properly but while picking some of them, we are hitting the tray and the tray, along with the objects inside, is falling outside the table, onto the ground.



task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups	objects picked up
1-1-3	32.14	33.69	4.6	3	['orion_pie_v1', 'suger_2_v1', 'suger_3_v1']
1-1-8	37.05	33.3	-11.27	2	['orion_pie_v1', 'suger_2_v1']
1-5-4	34.38	26.13	-31.56	4	['suger_2_v1', 'plastic_apple_v1', 'potato_chip_1_v1', 'plastic_strawberry_v1']
2-1-3	41.08	35.66	-15.21	2	['plastic_plum_v1', 'plastic_pear_v1']
2-1-4	35.9	35.38	-1.47	2	['plastic_peach_v1', 'plastic_orange_v1']
2-1-5	53.16	31.2	-70.38	4	['plastic_plum_v1', 'plastic_strawberry_v1', 'plastic_apple_v1', 'plastic_peach_v1']
6-2-6	56.97	39.24	-45.2	3	['bleach_cleanser_v1', 'cleanser_v1', 'shampoo_v1']
6-2-8	56.15	36.89	-52.2	1	['magic_clean_v1']

POSSIBLE REASONS FOR FAILURE

- In 2-2- series
- **Common point** : Duplicate objects
- **Reason (might be)**: It's trying to go to ghost places i.e. to places where no object exists.
- We are investigating why this is happening



task_index	final error per object (cm)	default error per object (cm)	improvement(%)	successful pick-ups	objects picked up
2-2-2	38.14	40.57	6	0	[]
2-2-1	36.33	37.89	4.12	1	['green_bowl_v3']
2-2-9	36.12	35.95	-0.48	0	[]
2-2-7	55.15	47.97	-14.97	0	[]

NEXT TO-DOS

1. **Time Scaling** – The panda arm takes a considerable amount of time to return to the home space before pick and after place. *There are time scaling approaches that we can use to increase the speed of the robot arm in the joint space – time scaling, time scaling approach*
2. **PUSH** – There are objects that can't be picked up because the object width is greater than the maximum width of the gripper. *We can push the objects to the edge of the table and grasp them.*
3. **Object-wise-grasp-pose-generation** - The contact graspnet has transformation issues which we were not able to solve analytically in the last few days. In case contact can't be integrated properly, we propose to use GraspNet with segmented pointclouds in order to assure that we have grasp poses for each object
4. **Object grasp verification** – In cluttered scenes, the gripper can inadvertently pick some other object due to clutter even though it estimates the object pose accurately. *After pick, we can use the Kinect to verify the object picked using feature matching on pretrained VGG features.*

Object-wise-grasp-pose generation

- Approach 1: Convert the standard mesh files into point cloud and transform them to the actual pose of the object (detected via superglue) and then find grasp poses for this using GraspNet
- Approach 2: 2D segmentation of PCD

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 1st January 2022

WEEKLY UPDATES

Made two submission (2 submissions are allowed each month) -

1. 27th December

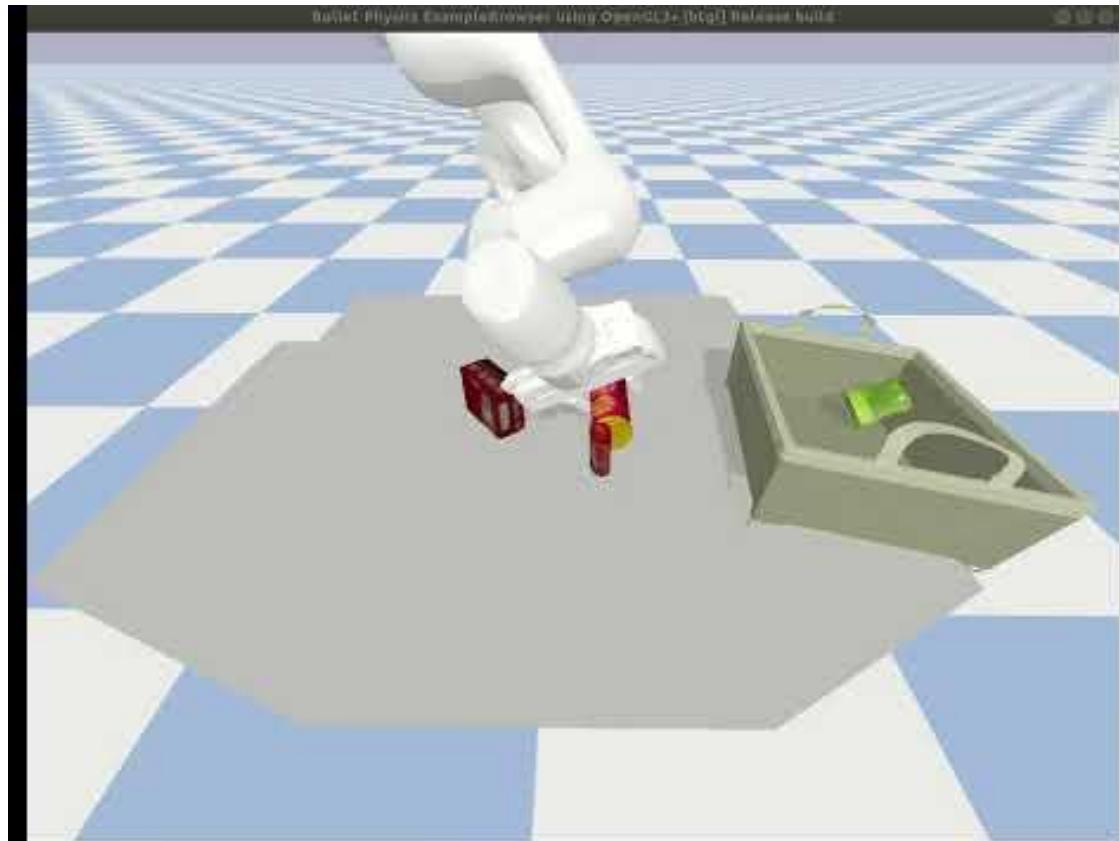
1. Duplicate Object Fix – Able to identify all the duplicate objects + pick and place the objects.
2. Better arm poses – Kinect + 2 back arm poses for faster and denser reconstruction.
3. Testing gripper to see if something is being held by the gripper.
4. Superglue features are cached to make the execution faster.

2. 31st December

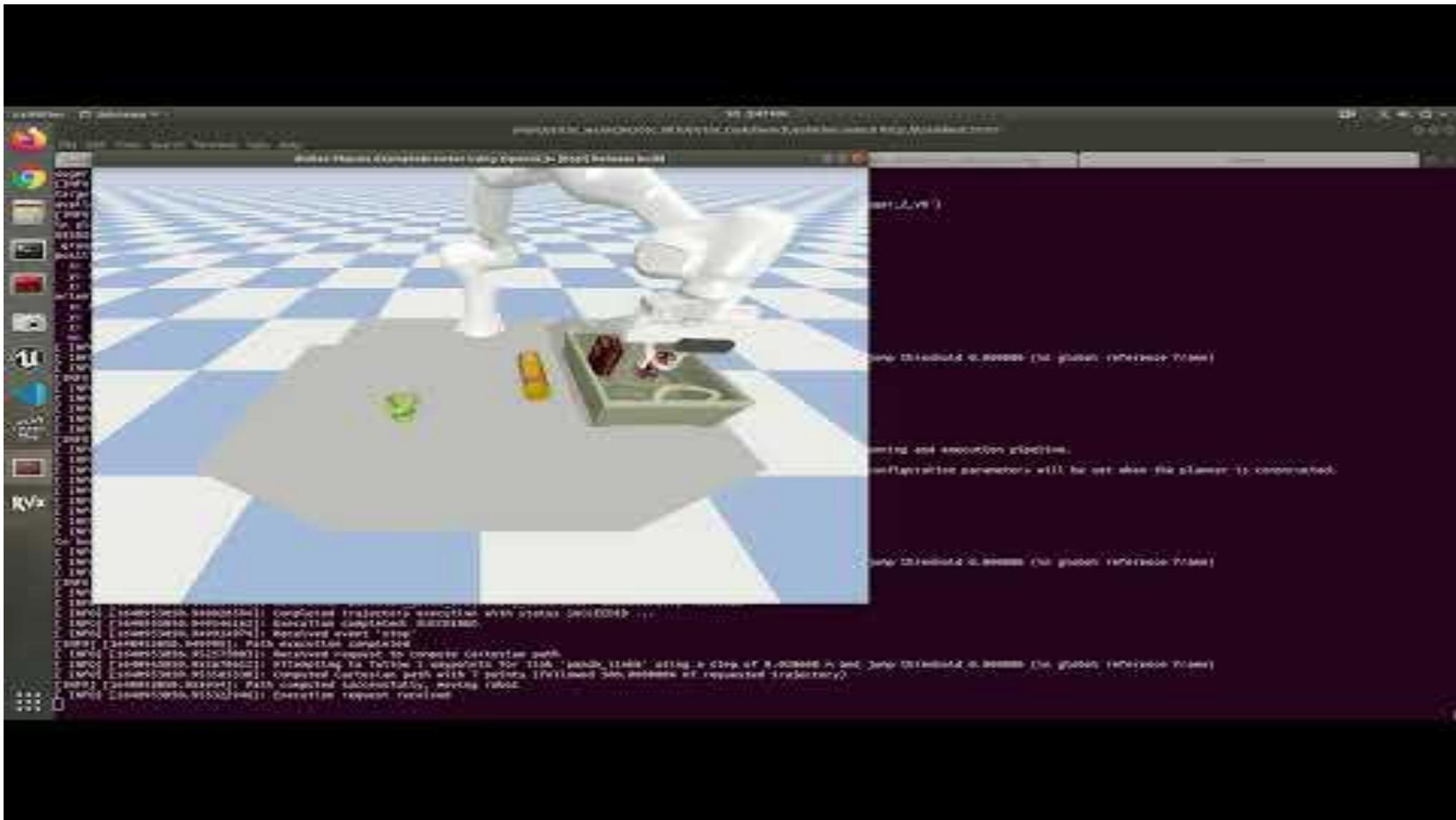
1. Adding a home-pose to the pick and place trajectory. This is removing weird movements.
2. Better grasp assignment logic such that the grasp is always assigned from the top.
3. Grasp orientation fix such that there is no angle between the gripper and the table (smoother).
4. We can now pick plates!

1. HOME POSE DURING PLACE AND PICK

1. Moving between different objects, the grasper made weird movements (very complicated movements) to reach certain position. We are now adding home pose between every pose.
2. This is allowing the robot to re-align itself and not get entangled in the weird joint movements.



1. HOME POSE DURING PLACE AND PICK



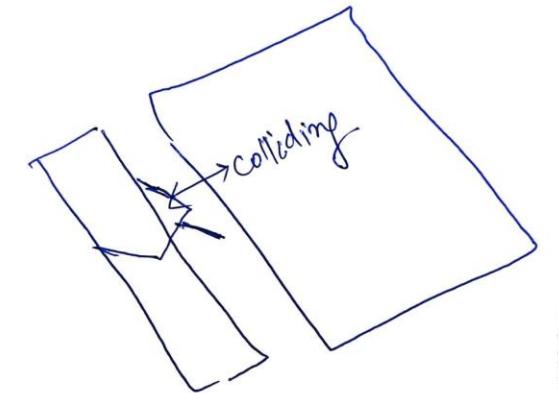
2. A BETTER GRASP ASSIGNMENT LOGIC

Old Algorithm –

1. Distance threshold to assign grasps to a given object – here, grasps closest to an object centroid were assigned to the object.
2. Sorting based on 'confidence' scores of all the grasps assigned to the given object.

Problems -

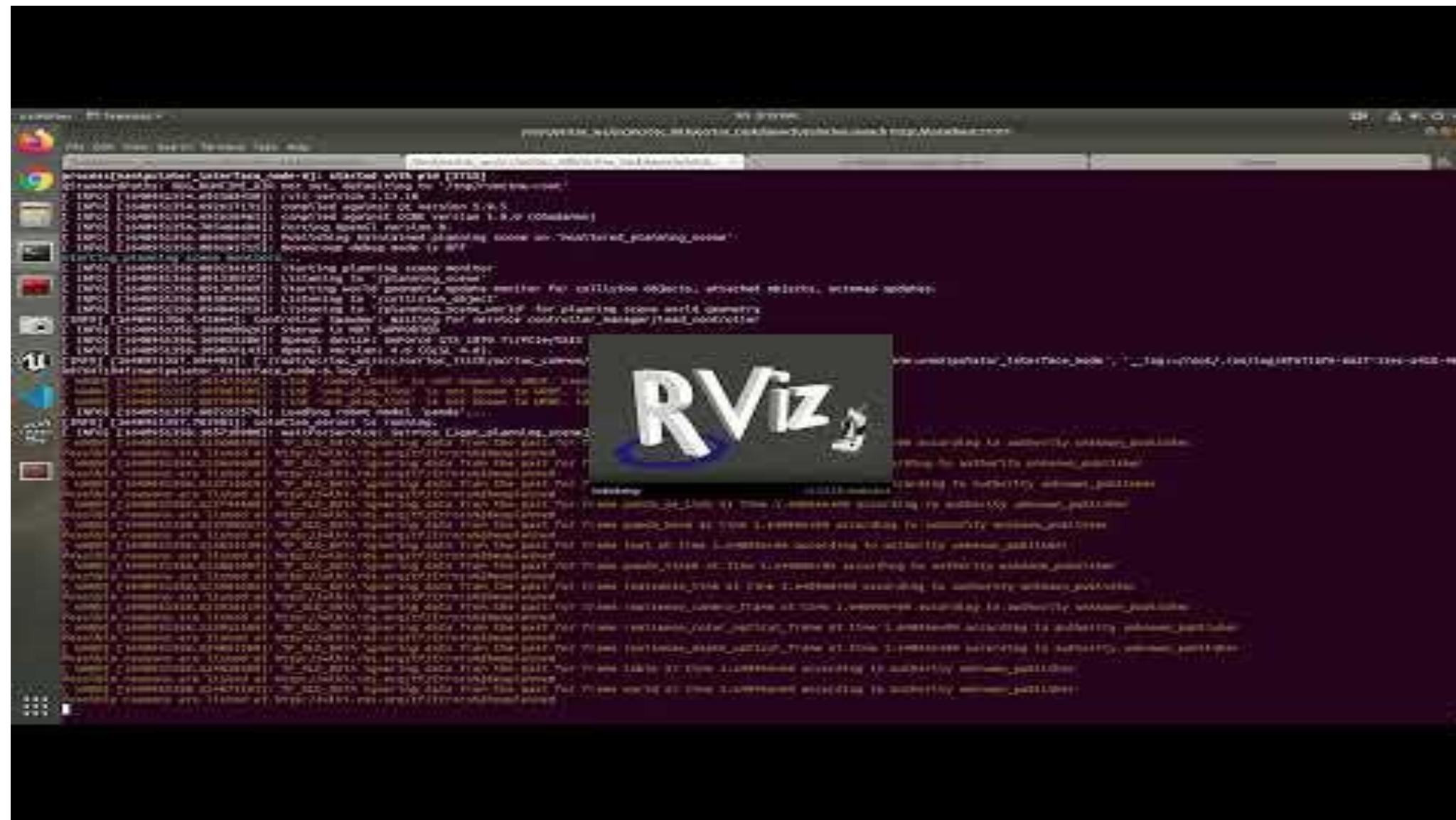
1. Weird angles to pick an object is colliding with objects beside the object to be picked.
2. The gripper isn't going down fully (task planning fails) as it collides with the table.



Our algorithm –

1. Distance threshold to assign grasps to a given object – here, grasps closest to an object centroid were assigned to the object.
2. Obtaining the Euler-angles of the orientation of the grasp pose. Pick the grasp pose with the lowest roll and pitch angle so that the gripper picks from as top as possible (choose pose with the lowest angle)

2. A BETTER GRASP ASSIGNMENT LOGIC – PREVIOUS LOGIC DEMO

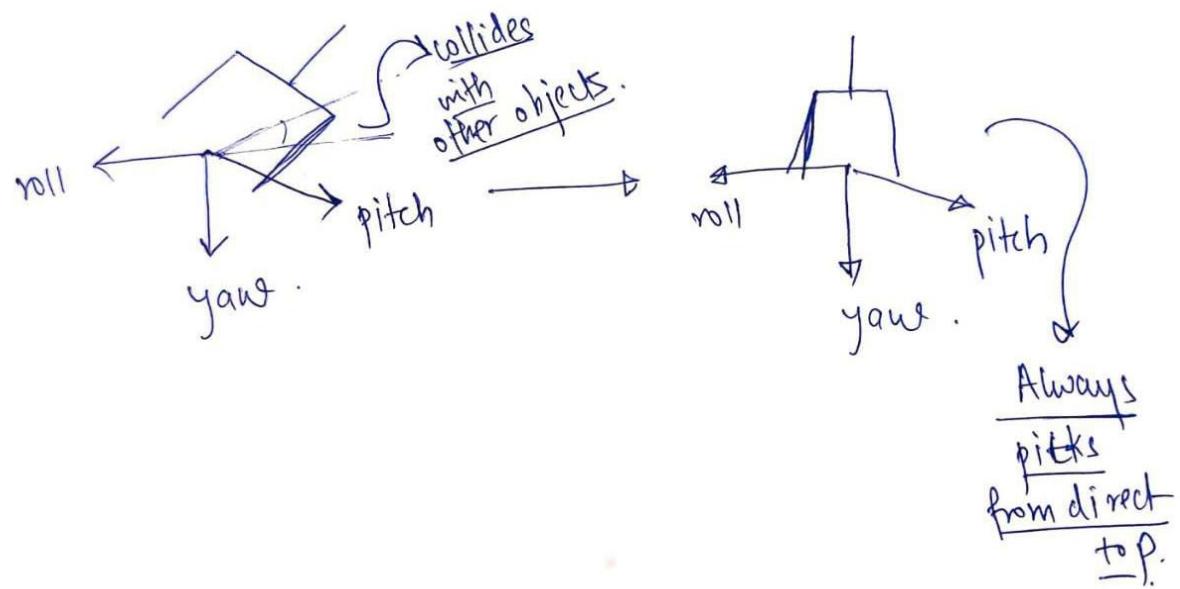


3. FIXING THE ORIENTATION FOR THE GRASP POSE

Along with the previous fix, we added two more changes -

We always pick from the top now – this works for 99% of the objects.

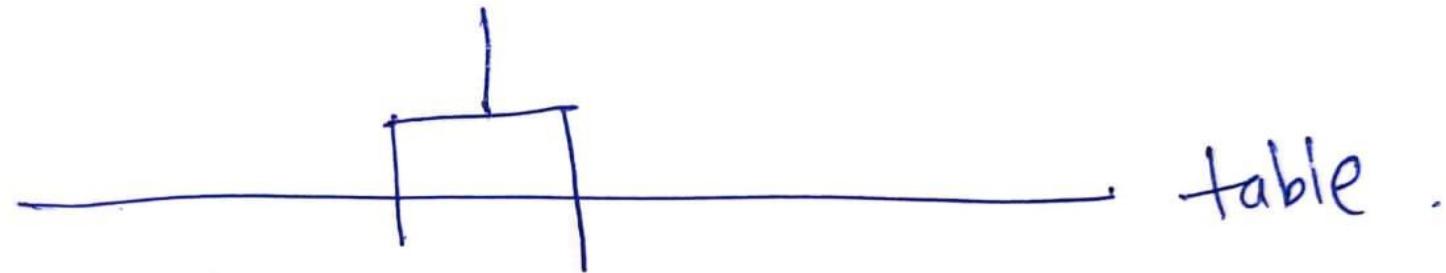
1. Here, we first determine the best grasp pose from the previous logic.
2. Next, we rotate the grasp pose about its own axis such that the x-y-z of the grasp pose is not changed.
3. We set the roll and pitch to 0 and but keep the yaw angle unchanged.



4. WE CAN NOW PICK PLATES!

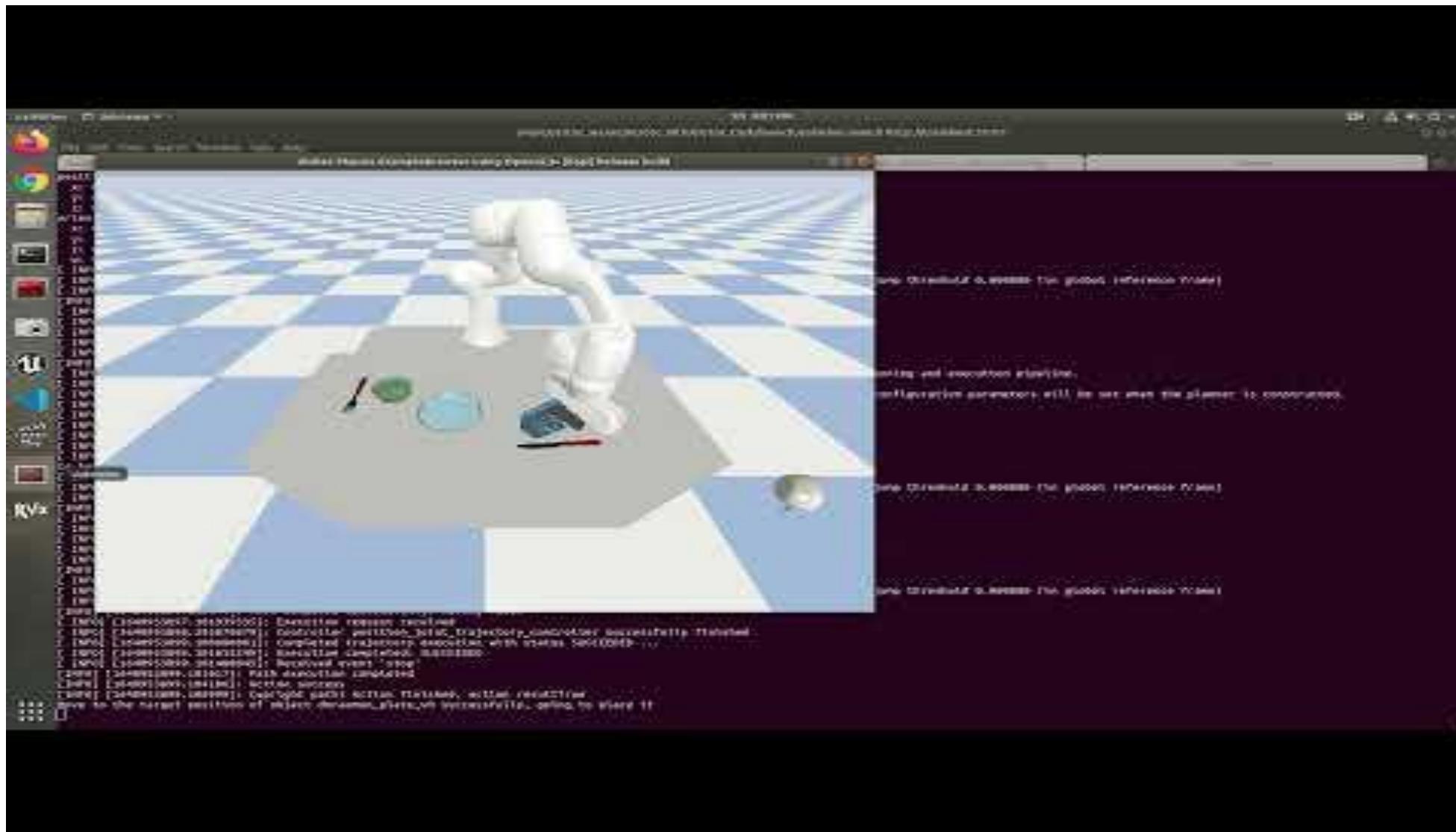
We add one more fix --

The 'z' value of the grasp pose is sometimes 'below' the table. This causes a collision and then the task planner fails in such cases. Our simple fix -- **Set z to zero whenever $z < 0$.**

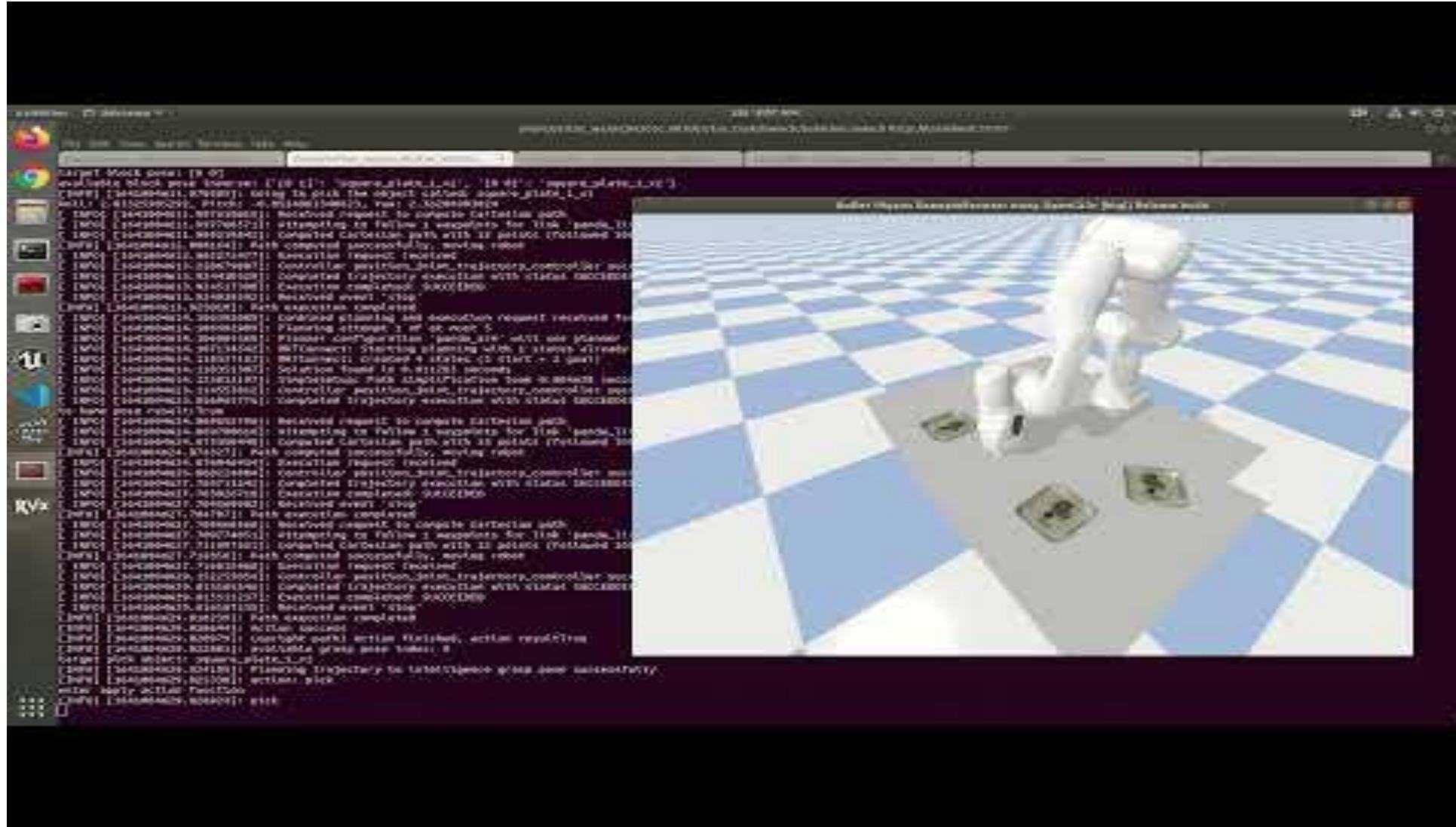


Now, we can pick plates (and flat objects) too! Demo in the next slide.

WE CAN NOW PICK PLATES – DEMO!



WE CAN NOW PICK PLATES – DEMO!



NEXT TO-DOS

Plan	Submission	Date
I		15th January
II		28th January

There is significant amount of stochasticity, need to make sure our current fixes always work!

Immediate TO-DOs.

1. Perception Stack --
 1. Contact Graspnet – Have integrated already, need to fix transformation between the poses.
 2. Grasp Verification – Was the correct object picked?
2. Planning Stack --
 1. Fix Task Planner in cases like Stacking (tasks like 4-1-2)
 2. Faster joint space trajectory movements.
3. Need to make more observations and prioritize the task items.

OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 25th December

PLAN FOR FIRST SUBMISSION – 26TH DECEMBER

1. First submission is planned on 26th December (Sunday). Second submission on 30th December (Thursday)
2. Following are the improvements planned for submission:

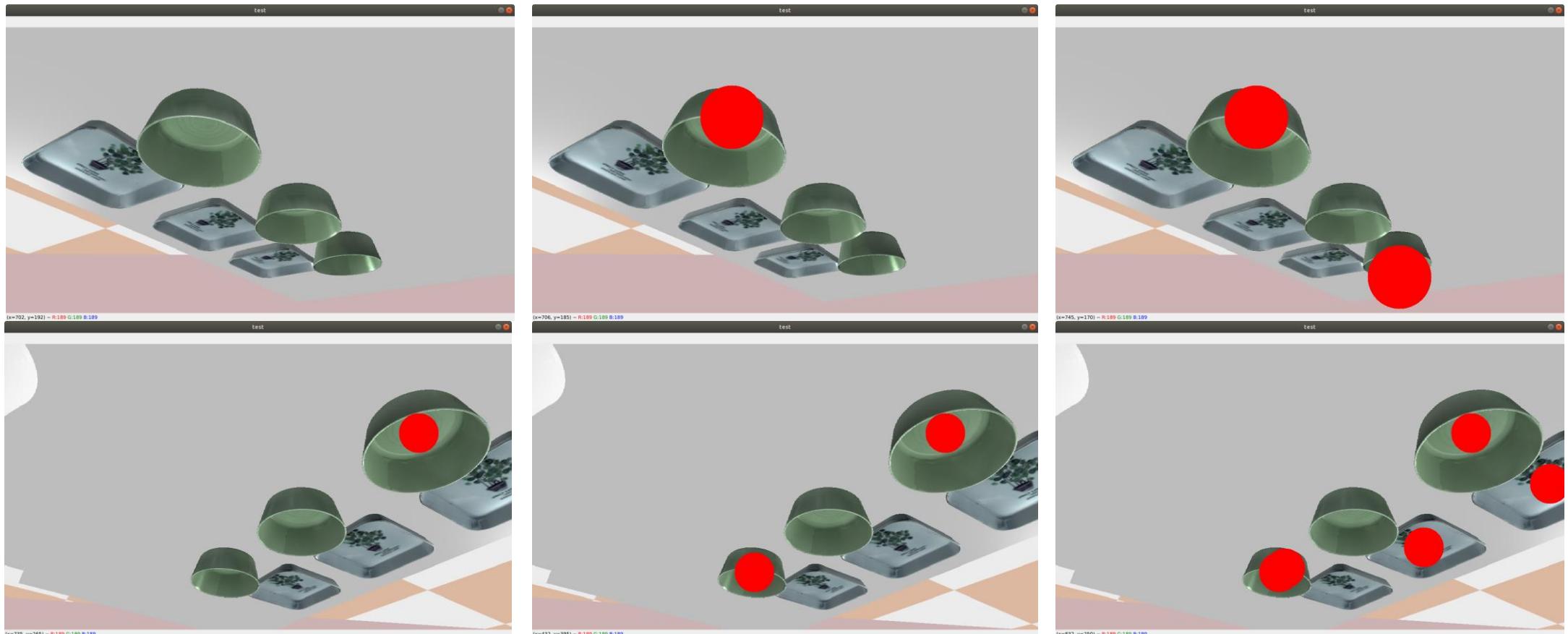
	Fix	Advantage
1.	Duplicate Object Fix	Able to detect multiple poses for duplicate objects (better than baseline)
2.	Contact Grasnet	Able to detect better grasps for each object (better than baseline)
3.	Gripper Width	Removing redundant movements – saving on time.
4.	Caching Superglue	Removing redundant computations (+ skip views) – saving on time.
5.	Kinect (better poses)	Denser and faster pointcloud reconstruction – saving on time + better performance.
	Ad-hoc task	Integrate from multiple branch + proper file that can be submitted (1 day)

3. For the second submission, make the above fixes perfect and check for any bugs on all tasks.

DUPLICATE OBJECT FIX

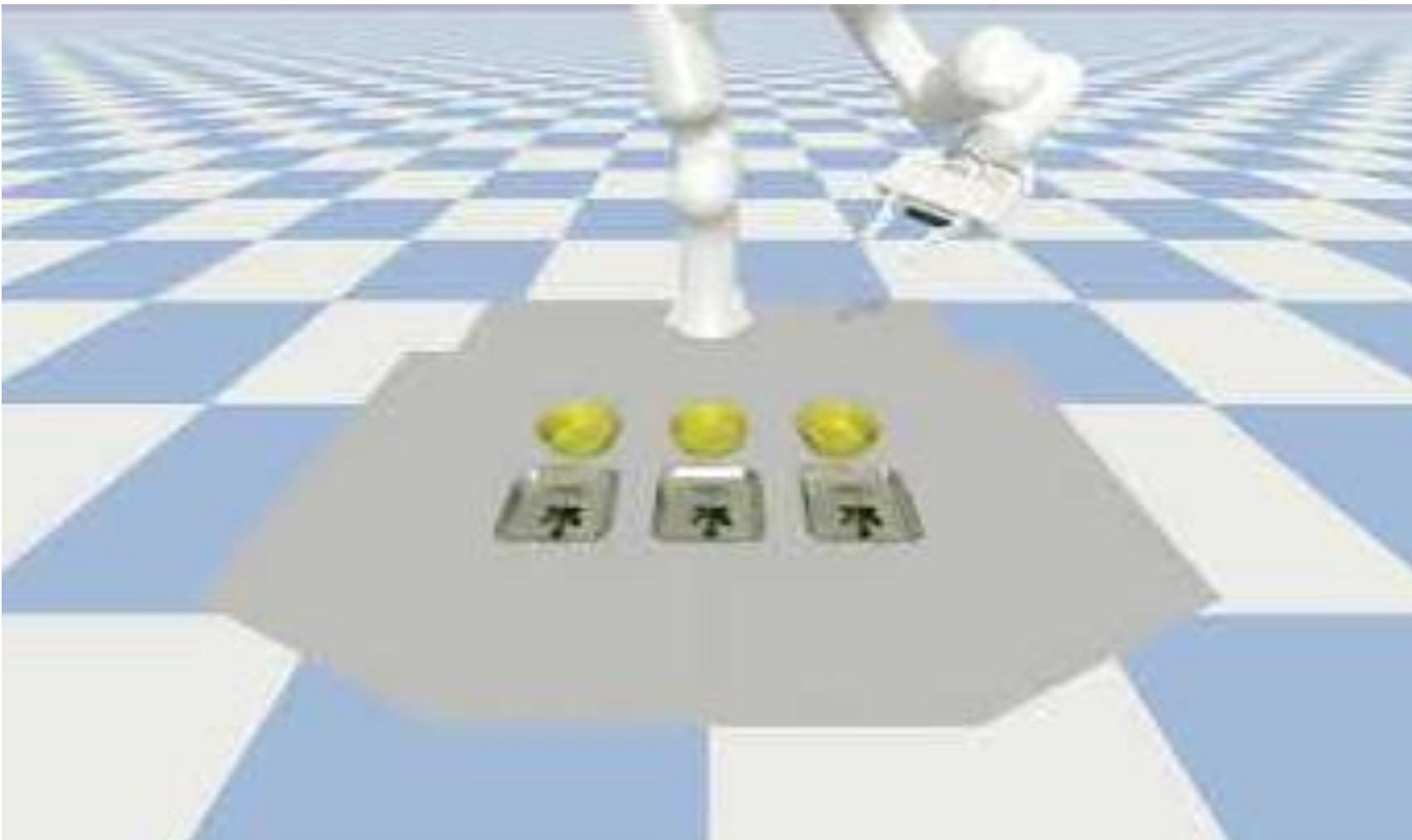
Problem: Superglue assigning the same pose for duplicate objects.

Approach – After the detection of each object – mark that object with a color to reduce the matches.



Verdict – It worked!! 😊 Unique pose is getting assigned for each object.

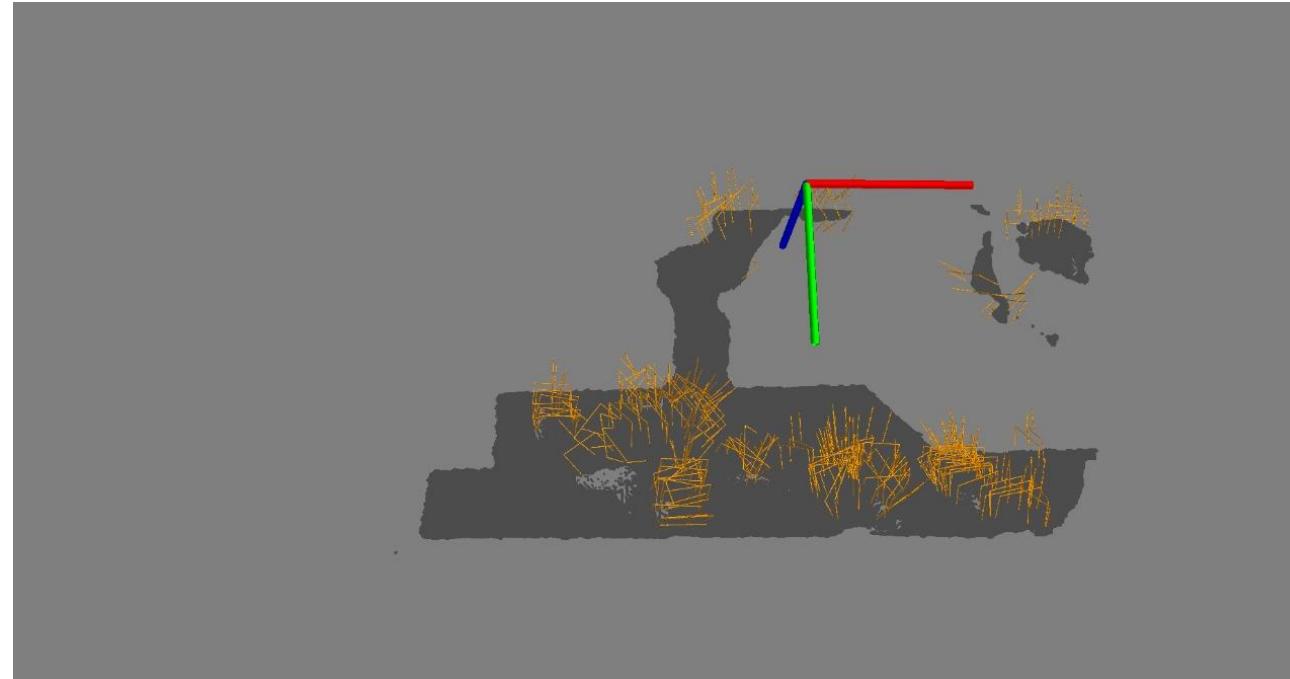
DUPLICATE OBJECT FIX - DEMO



CONTACT GRASPNET

Most of the objects do not get any grasp pose!

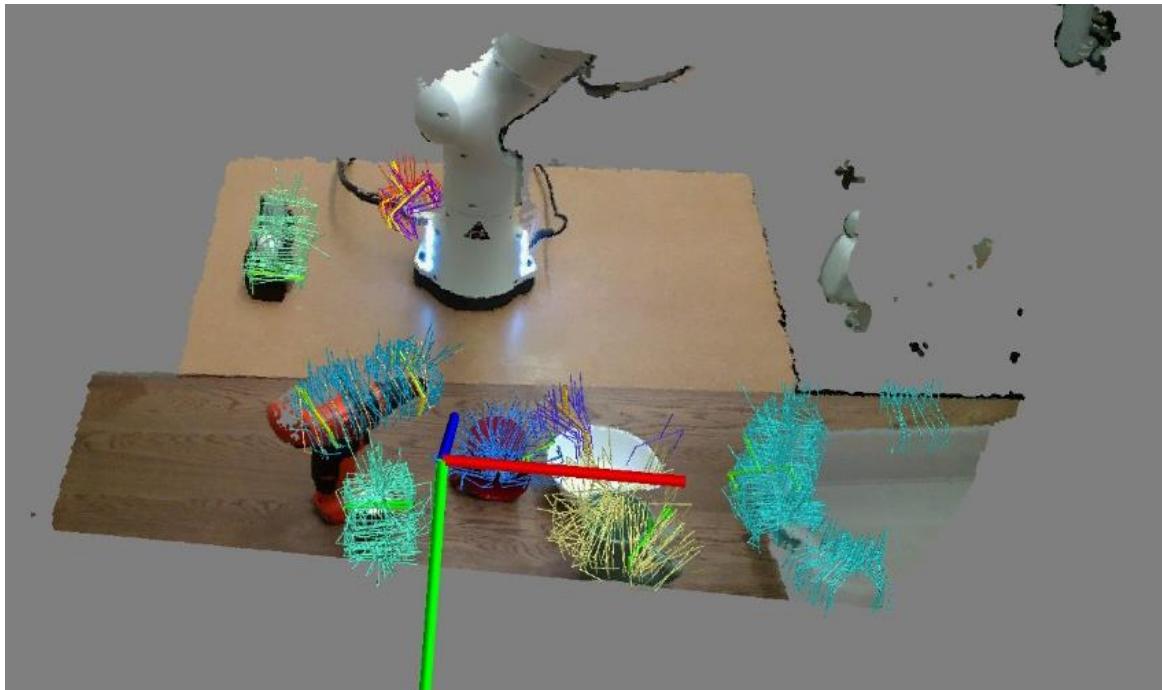
1. Contact graspnet specializes in assigning grasps to a pointcloud made of multiple objects –we need!
2. It can also take segmentation map and generate grasp poses for individual objects.
3. Following is the result of a test pcd taken from their test set – without any segmentation map.



4. We are in the middle of integrating it.

CONTACT GRASPNET

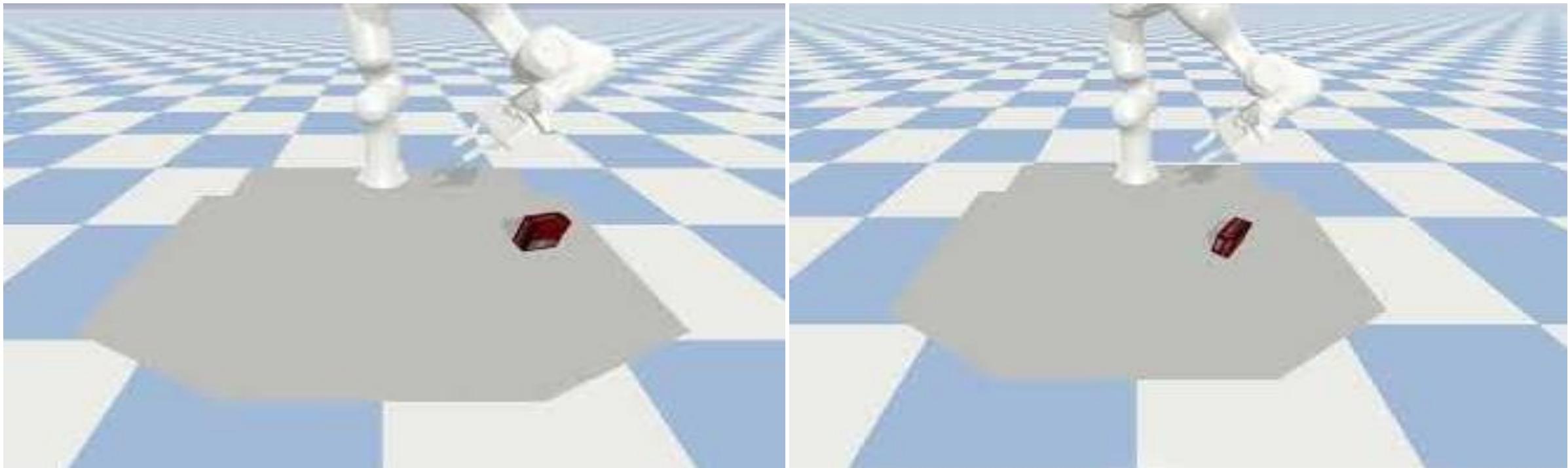
1. Grasp poses are much denser – even without any segmentation information.
2. However, the poses are getting generated well for their test set, on our test set we did not get any pose.
3. We are trying to debug the issue, maybe finetune on our dataset.
4. Following is the result using the segmentation map



FINAL POSE AFFECTED BY THE INITIAL POSE

Motion Planning is not taking the object to the same final pose when the initial pose is changed.

Need to debug.



We are targeting this fix for the 2nd submission of this month.

TO-DO BEYOND SUBMISSION – FOR JANUARY MONTH

Minor targets –

1. With the submitted fixes – make them perfect (perfect pose, no bugs – all in one branch).
2. Clear tray fixes – hacks to not touch clear tray and drop from the top.

Major targets –

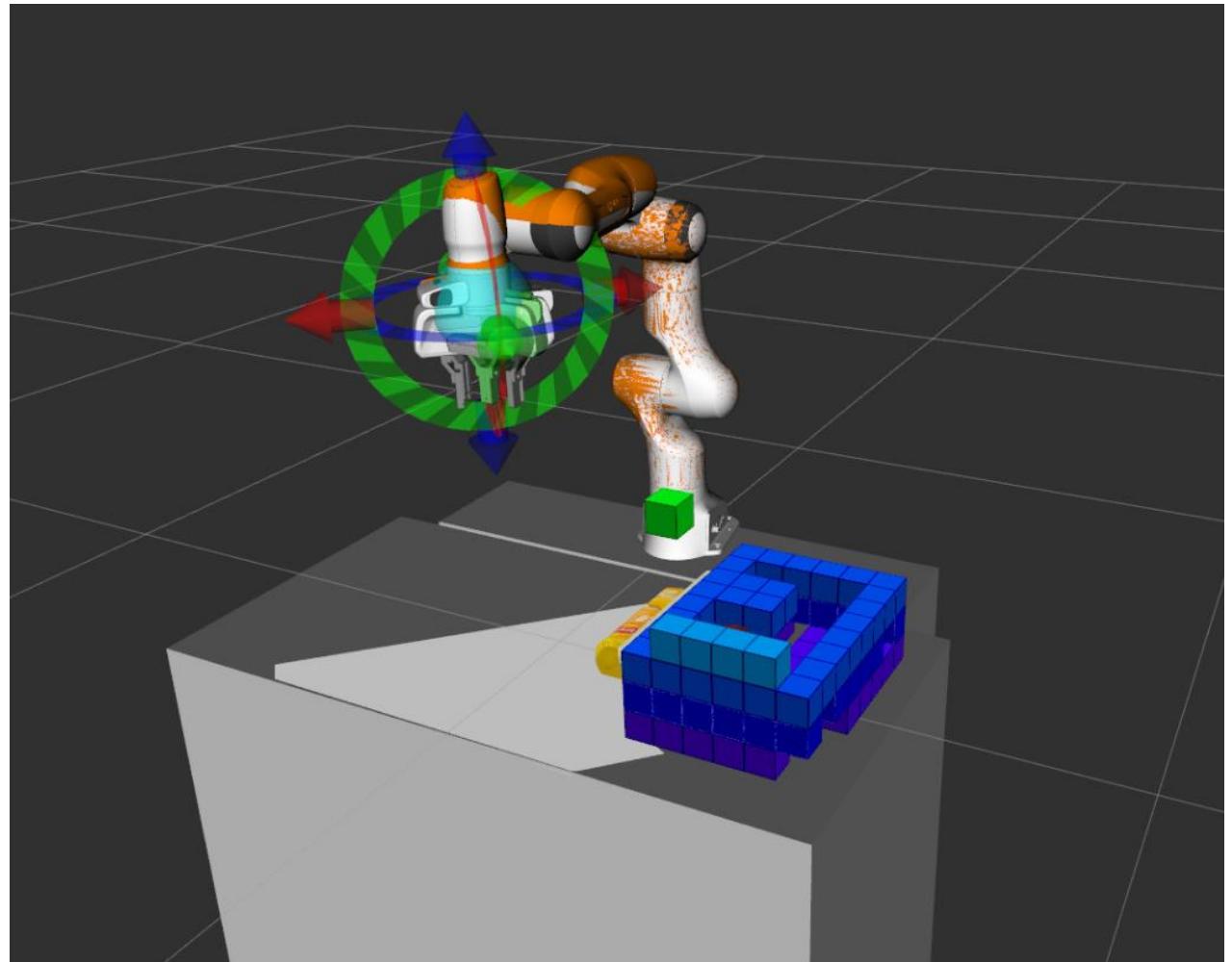
1. Scene collision avoidance – integrate [Scene Collision Net](#) by NVLabs.
2. We are currently verifying if the gripper has picked any object at all. Some mechanism to verify if the ‘correct’ object was picked? Say, it is supposed to pick chips, did it pick chips or something else?
3. Push action? Many of the tasks require the push action – could be one advance thing that we can solve – maybe TCS group can take this up?
4. Any other suggestion?

OCTOMAP

The collision voxels update fast
and hence get added to the scene.

So, before the placements, collision
voxels get added to the object in
hand (like the green shown) and
prevents the arm from placing.

→ Integrate Scene Collision Net?



OCRTOC Challenge 2021

Robotics Research Center Submission

Updates – 18th December

OVERVIEW – 18TH DECEMBER

1. Grasp verification: If an object has been properly grasped by the gripper.
2. Collision avoidance with the clear box: drop the objects from a certain height above the clear box.
3. SuperGlue – Reducing the operational time from 5 minutes to less than 1 minute.
4. Octomap – update rate issue – addition of collision voxels faster than the removal.

Figured a workaround? - clear Octomap service

5. Get denser 6DOF grasp poses from graspnet.
 1. 6DOF parameter to make it denser?
 2. Segment object point clouds and generate grasp poses for individual point clouds instead of full point cloud.
 3. Upgrade to contact graspnet is denser?

GRASP VERIFICATION USING GRIPPER'S WIDTH

18th December

Problem: Even if an object is not picked up by the gripper, the arm is executing the entire plan.

Solution: Checking the joint states data of the finger joints before moving after grasping (pick) the object.

Merits:

- If the object hasn't been held, the grippers would be completely closed, and we need not do the complete pick and place without the object in hand and move on to pick the next object.
- This object would also be removed from the completed tasks so that the perception task will restart after placing the other objects and start searching for its updated current pose and new grasp pose.

Failure:

- If the object was held improperly and if the object falls off during the movement. This will fail.

GRASP VERIFICATION USING GRIPPER'S WIDTH - DEMO

18th December

1. If the gripper fails to grasp, it moves on to the next object without executing the entire plan of placing.
2. The objects are being dropped at a height to prevent gripper's collision with the tray.



CLEAR OCTOMAP FOR A FASTER FREE-SPACE UPDATE

18th December

Problem: Octomap update rate is slow for free space detection

(takes about 2 to 5s for complete update – a huge bottleneck).

- Currently, in our strategy, before picking, we would remove the about-to-be-picked object's point cloud cluster so that it won't be considered as a collision object while planning for the pre-grasp pose. This is where the update is slow (on removing the object, it takes about 2 to 5s to update), and hence the plan fails.
- It adds the collision voxels for new objects instantly, but it doesn't update collision voxels around the removed objects at the same rate.

Potential solution/hack:

Clear/reset the Octomap and pass the new point cloud (without the about-to-be-grasped object's point cloud cluster).

- This is found to be much faster than allowing the Octomap updater to update the free space on the original map (observed from our initial experiments with manual intervention).

CLEAR OCTOMAP FOR A FASTER FREE-SPACE UPDATE

18th December

Plan

1. Implement the proposed hack – clear and reset the Octomap before grasping.
2. If the above doesn't work as expected, try to find a workaround to manipulate sensor_model/[hit/miss] and sensor_model/[min/max] parameters.
3. Look for other strategies if 1 fails.

SENDING OBJECT FEATURES TO REDUCE BANDWIDTH

18th December

Problem: Superglue is being used to obtain the initial 6DOF pose of the objects that is used to map the initial and final position of the objects. The entire process is taking about 5 mins.

How does superglue work?

1. For every object that is present on the table (obtained from the final pose yaml), capture the image of the corresponding mesh from 82 different angles.
2. For every image taken by the gripper, match the 82 views per object to the image and find match.
3. Total feature computation and matching: $82 \times \text{angles (6)} \times \text{number of objects on the table}$.

Solution

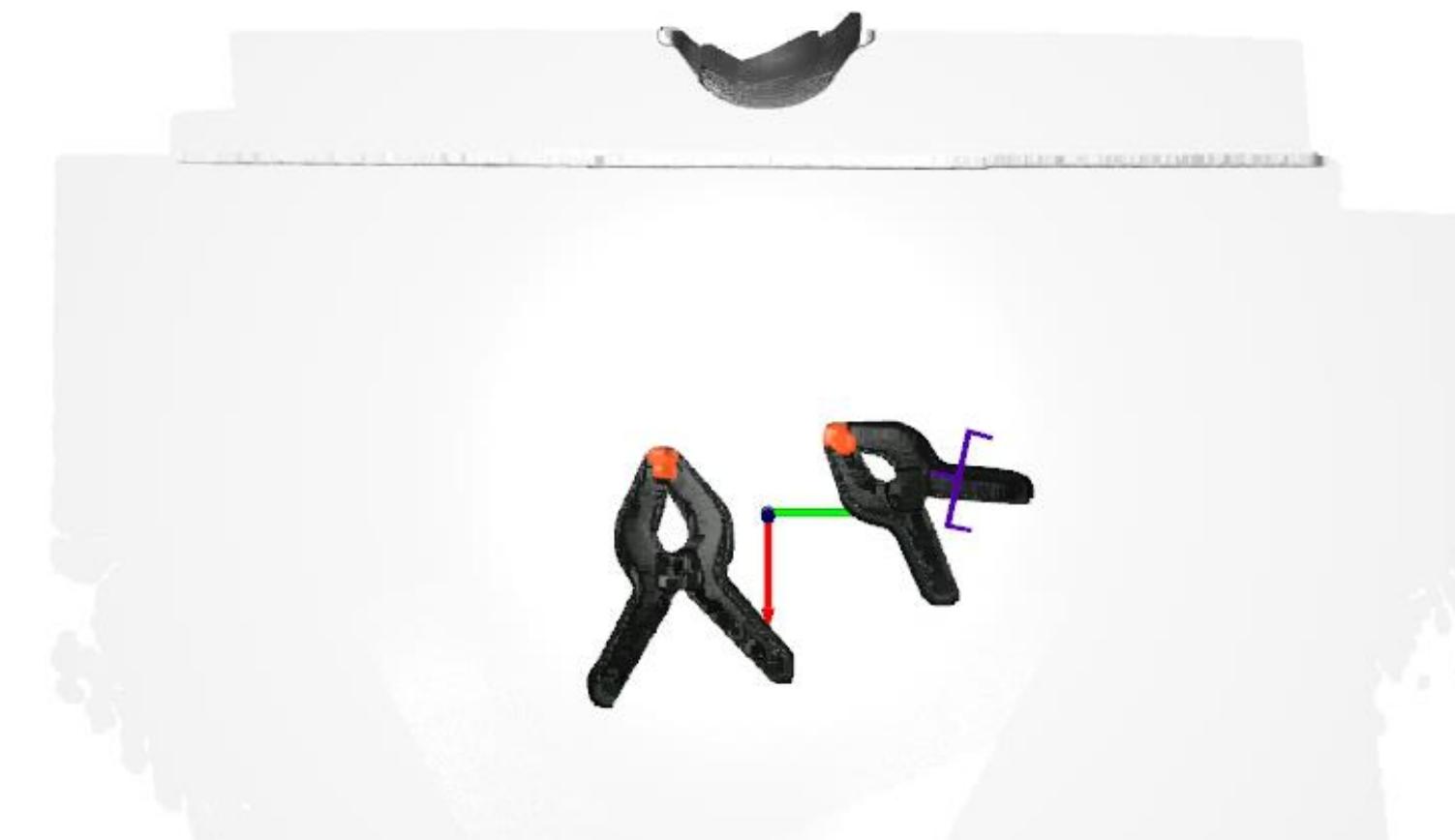
1. Store the features computed for every angle of every objects in the cache. Takes < 30 seconds.
2. Decrease the number of views per object to 20, this will give us less accuracy for the unseen objects

Yet to be verified completely if it causes any drop in performance anywhere else.

GRASP POSES

18th December

Can 6DOF graspsnet be denser?



PLANS FOR NEXT WEEK

18th December

1. Implement the collision feature using the proposed hack – clear and reset the Octomap before grasping.
2. Save SuperGlue features to a pickle file, reuse them for known models and check the performance.
3. Denser grasp poses –
 1. segment object point clouds and generate grasp poses for individual point clouds instead of full pcd.
 2. **Downfall** – We might see object collision as grasp pose generation won't be self-aware.
4. How to approach duplicate objects?
5. Collision Avoidance - Select grasp poses (out of all the poses predicted by graspnet) for individual objects using max L2 distance from other pcds **along with** min L2 distance from object centroid.
6. Initial camera poses with good reconstruction.
7. Combine the pcds of kinect and realsense camera by fixing pointcloud merge issues.

PLANS FOR NEXT WEEK

18th December

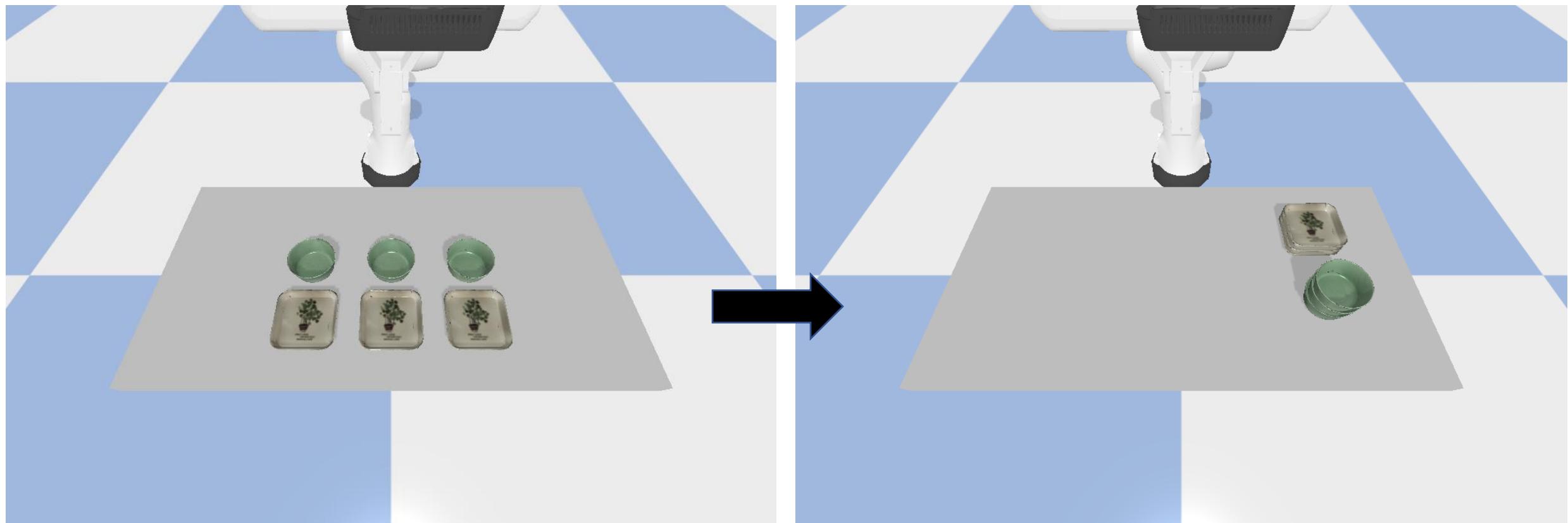
Module	Estimated Date
Collision feature (clear Octomap)	22/12
Superglue features	
Contact GraspNet	
Duplicate objects	
Initial camera poses (good reconstruction)	
Combining PCDs of Kinect and Realsense	
First Submission	25 th December – understand the process to submit.
Second Submission	31 st December – with more improvements + estimate of our position on the leaderboard.

FEEDBACK DURING THE MEETING

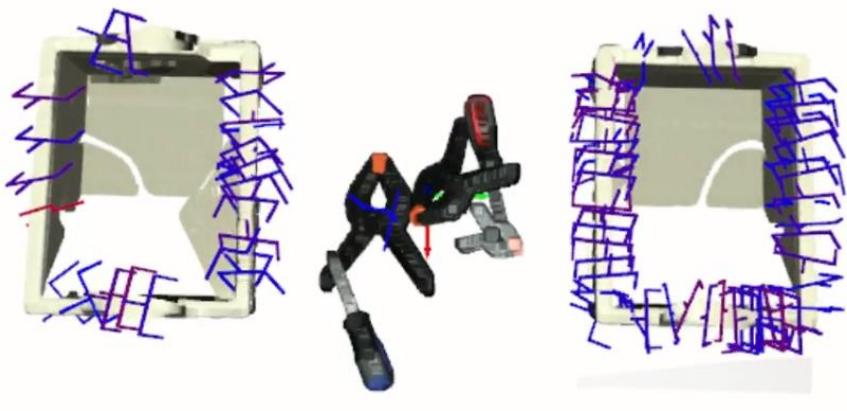
18th December

1. Implementation of Octomap updates – check if the implementation is a problem.
2. Check if the orientation of an object lead to penalization – for larger object, there can be more penalization if their orientation is not proper (this is after an object has been placed to its final location).
3. Hierarchical pose estimation for grasp detection and collision avoidance –
Generate grasp poses for individual objects. If collision detected or invalid grasp, fall back to the grasp poses which were detected for the entire pointcloud. Avoid grasping where there are collisions and then retry logic – Segmentation and collision free scene clearing.
4. Duplicate objects – can change the color for duplicate objects such that SuperGlue is able to handle it.
5. Use ICP for finding the Kinect transformation and extrinsic (calibration of Kinect).

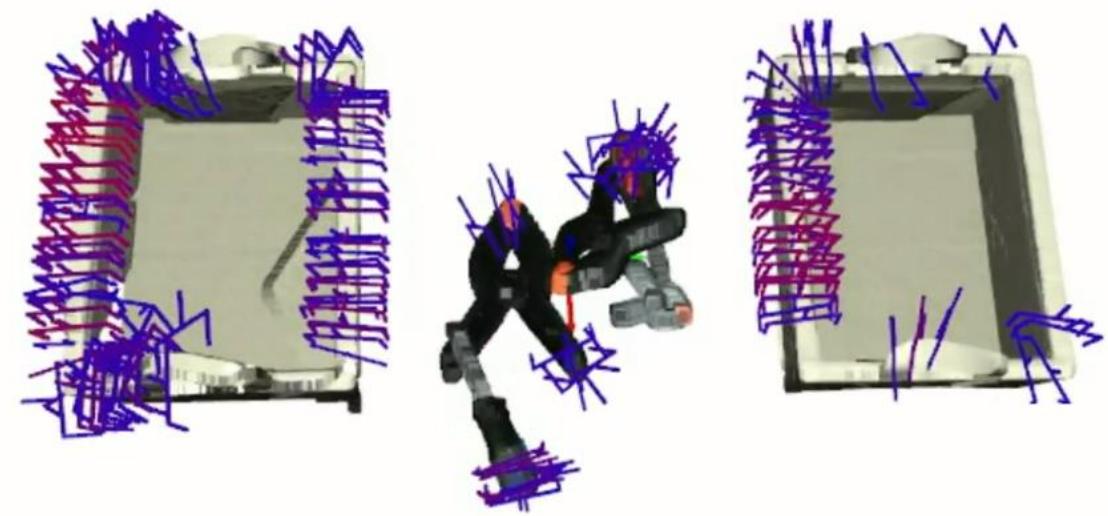
Duplicate Objects



Baseline can't handle this.

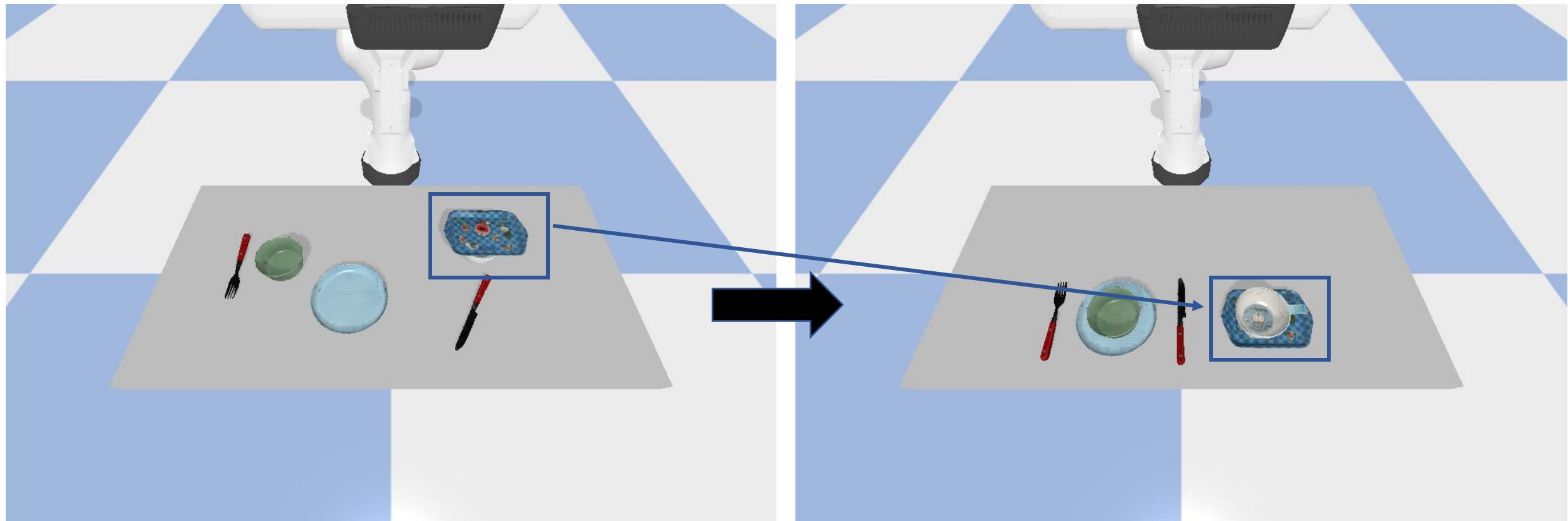


Baseline – 6dof grasynet



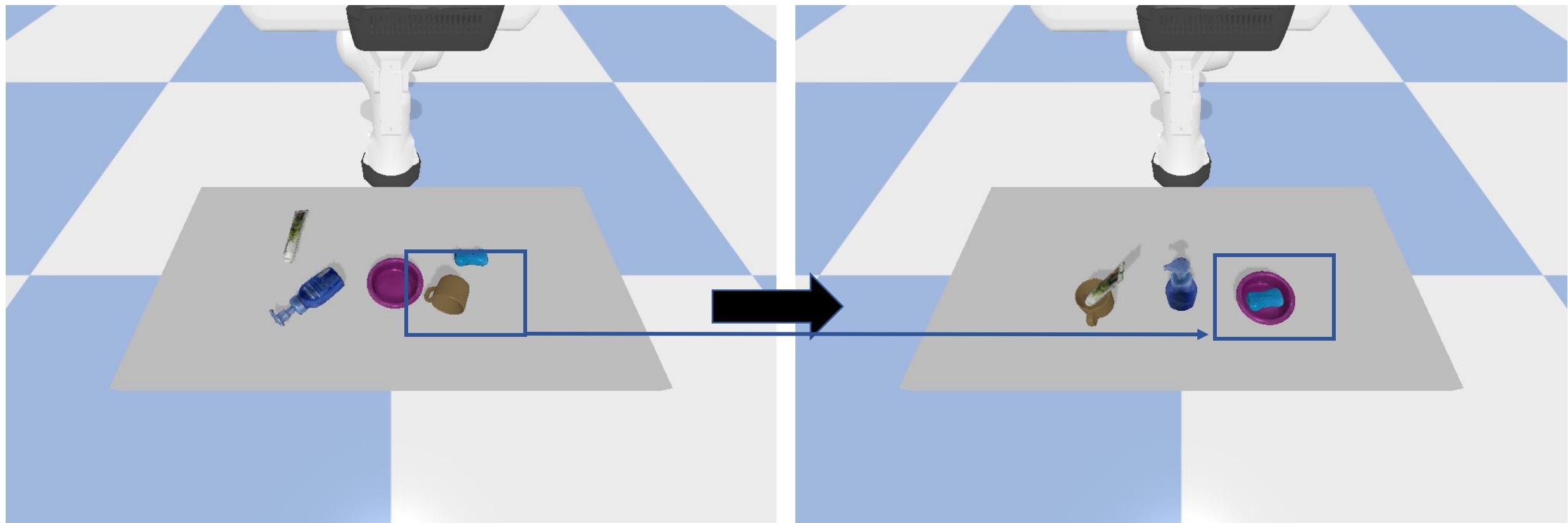
Our framework – contact grasynet

Destacking and Stacking

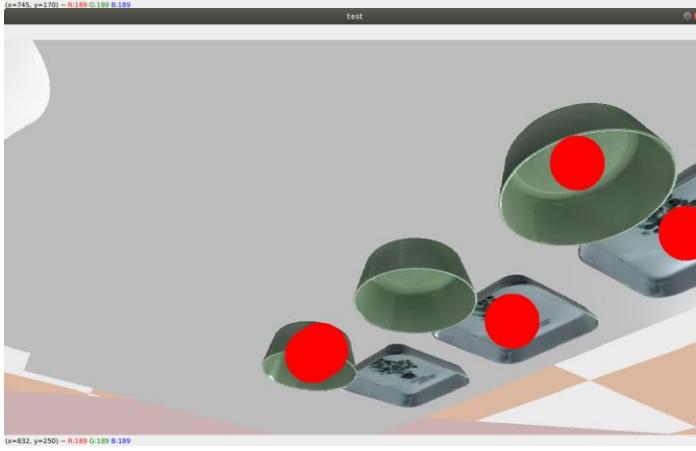
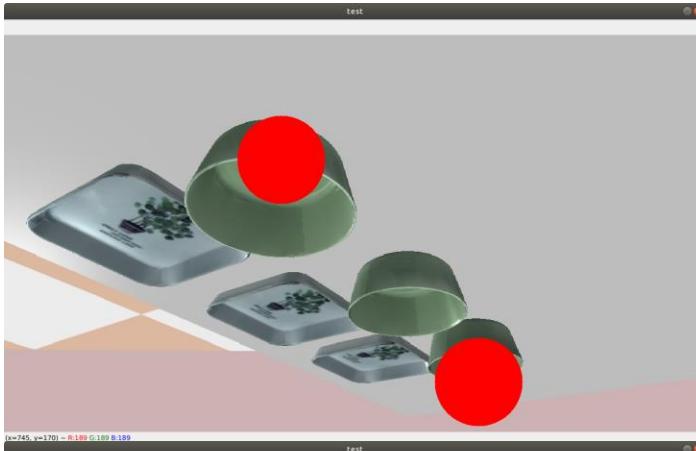
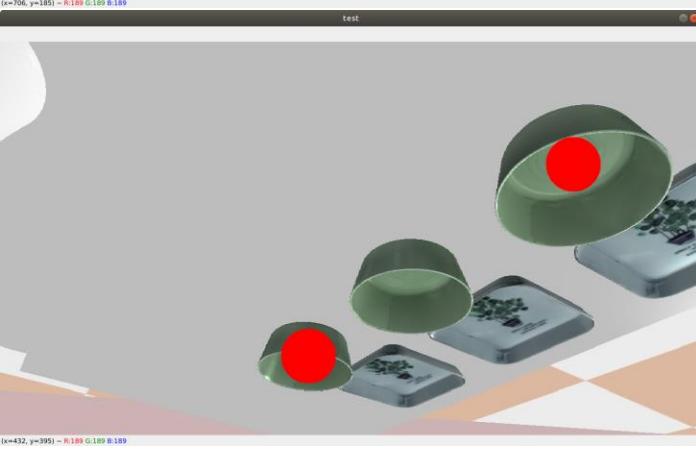
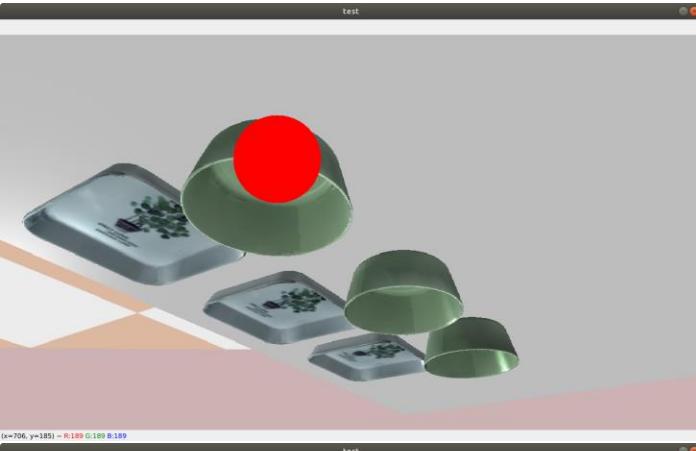
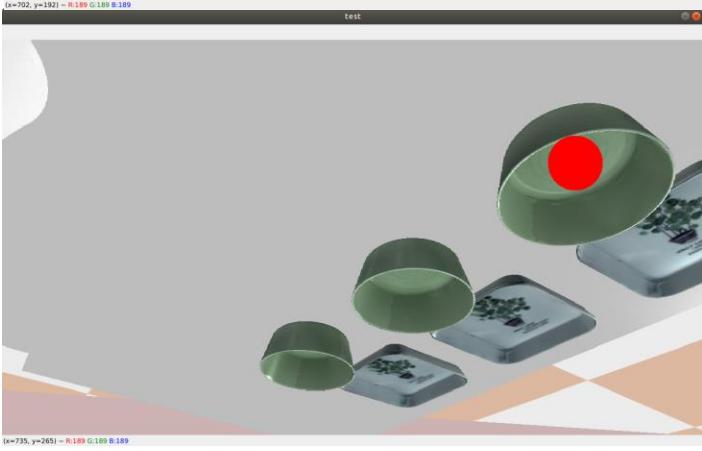
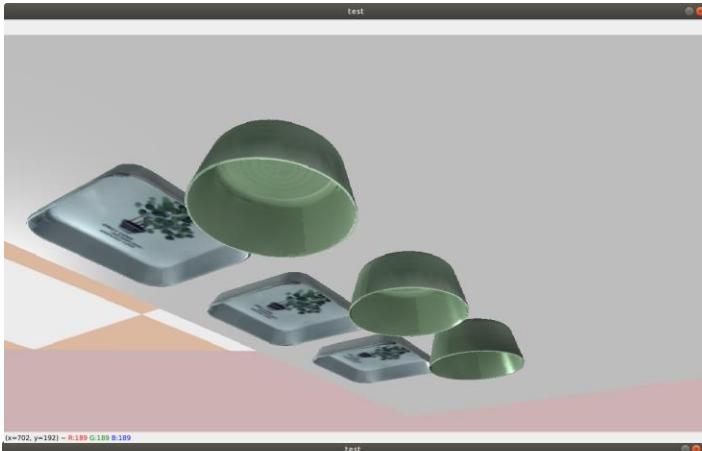


Baseline can't handle this.

Buffer Placement



Baseline can't handle this.



Object Swap

