2016 IEEE-RAS 16th International Conference on
Humanoid Robots (Humanoids)
Cancun, Mexico, Nov 15-17, 2016

Physically Plausible Scene Estimation for Manipulation in Clutter

Karthik Desingh[1] Odest Chadwicke Jenkins[1] Lionel Reveret[2] Zhiqiang Sui[1]

***Abstract*— Perceiving object poses in a cluttered scene is a challenging problem because of the partial observations available to an embodied robot, where cluttered scenes are especially problematic. In addition to occlusions, cluttered scenes have various cases of uncertainty due to physical object interactions, such as touching, stacking and partial support. In this paper, we discuss these cases of physics-based uncertainty one by one and propose methods for physically-viable scene estimation. Specifically, we use Newtonian physical simulation to validate the plausibility of hypotheses within a generative probabilistic inference framework for: particle filtering, MCMC and an MCMC variant on particle filtering. Assuming that object geometries are known, we estimate the scene as a collection of object poses, and infer a distribution over the state space of scenes as well as the maximum likelihood estimate. We compare with ICP based approaches and present our results for scene estimation in isolated cases of physical object interaction as well as multi-object scenes such that manipulation of graspable objects can be performed with a PR2 robot.**

I. INTRODUCTION

In order to perform purposeful manipulation, robots must have estimates of the pose and geometry of the objects in their environment, which collectively describes the robot's scene. This level of scene understanding is crucial for robots to perform the basic aspects of manipulation, including grasping of objects and planning collision-free movement. However, the perception of a robot's scene is dependent on what can be inferred from sensory information observable to the robot. Such inference is fraught with challenges, such as occlusions and physical contacts, which prevent acceptable levels of scene perception and, consequently, manipulation. Even when object geometries are known, the estimation of even a single object is a challenge, addressed by recent research [5]. The challenge for scene perception becomes much greater as the scene becomes more cluttered with an increasing number of objects. A common approach for tabletop scenes is to assume objects are physically separated [3], essentially removing the challenge of clutter.

Addressing this challenge for cluttered environments, we posit that physical plausibility is a necessary component in the estimation of scenes for robot manipulation. The challenges of perception in cluttered scenes is caused directly by the physical configuration and interactions between objects, as well as partial observability from the robot's viewpoint. As with similar analogous approaches to human tracking [22], [23], respecting physical viability often provides improved accuracy in the presence of uncertainty and efficiency in disregarding implausible scene configurations. For example, consider a case of a robot looking down at a large object stacked on top of a (completely occluded) small object. Current methods often misinterpret this scene as a single large box floating above the support surface. In addition to floating objects, physically implausible scene estimates can also occur due to inter-penetrating objects, unsupported objects, and unstable structures.

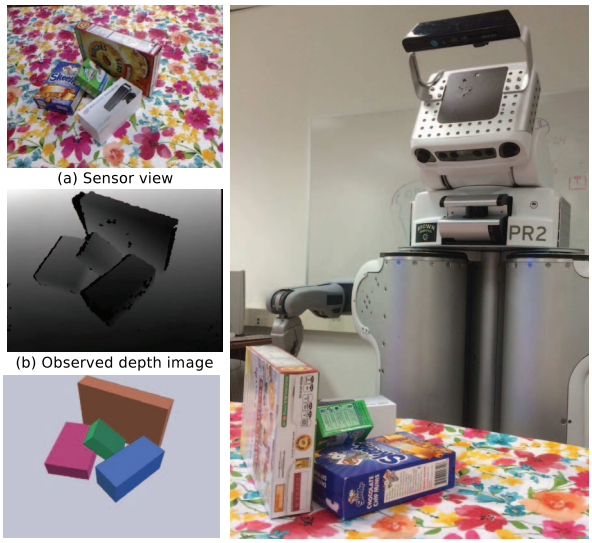


Fig. 1: A physically plausible scene estimate of a cluttered environment (a) viewed by a PR2 robot for manipulation. Using a Bayesian particle filter, hypotheses of possible states are generated through sampling, projected into physical plausible states and evaluated against robot's observation (b) to infer the most likely scene state (c) for manipulation.

In this paper, we propose a means for incorporating physical plausibility into generative probabilistic scene estimation using Newtonian physical simulation. Assuming geometry (dimensions), friction, and mass properties of N unique objects in 3D as known parameters, we explore three approaches to inference as a form of physics-informed scene estimation for static environments. In each of these methods, we use a physical simulation engine to constrain inference to the set of physically plausible scene states, which we treat as a *physical plausibility projection*. In terms of Bayesian filtering, we describe a physics-informed particle filter (PI-PF) that uses physical plausibility projection to correct implausibility that can occur due to additive diffusion. Based on the idea of [12], we bring PI-PF and MCMC sampling technique together as a physics-informed Markov Chain Particle Filter (PI-MCPF), where MCMC is performed

[1]K. Desingh, C. Jenkins and Z. Sui are with the Department of Computer Science and Engineering, University of Michigan, Ann Arbor, MI, USA, 48109-2121 [kdesingh|ocj|zsui]@umich.edu

[2]L. Reveret is with INRIA Rhône-Alpes, Saint Ismier, France, 38334 lionel.reveret@inria.fr

within the resampling stage of particle filter.

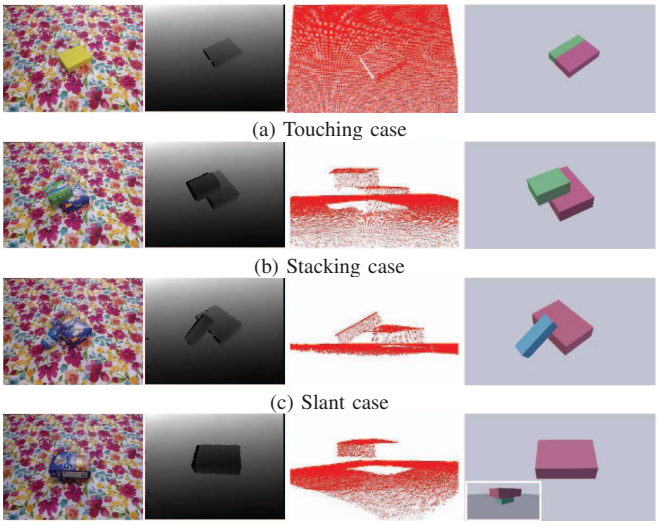
We provide ICP based approaches as the baseline, to discuss the limitations of data driven approaches and the advantage of the proposed methods. We present results for inference with the three physics-informed state estimators in primitive cases of cluttered scenes with two objects and more complex scenes with three and four object cases. While our results suggest that the PI-PF and PI-MCPF produce comparable estimation results, we observed the PI-MCPF converges in fewer iterations, albeit with more computational cost per iteration. Using our physics-informed estimators, we demonstrate manipulation of cluttered scenes with a PR2 robot.

II. RELATED WORK

The problem addressed by our physics-informed particle filter is to infer object-level manipulation semantics from 3D point clouds, or 3D maps more generally. Based on the semantic mapping work of Rusu et al. [18], the PR2 Interactive Manipulation pipeline [6] is able to perform relatively reliable pick-and-place manipulation for tabletop settings without object occlusions. This approach to localize objects relies upon estimation of the largest flat surface, where any contiguous mass extruding from this surface is considered a single object.

A number of discriminative methods have been proposed for estimating objects in point clouds and/or grasping in clutter scenes using depth images as their sensory input. ten Pas et al. [20] have shown impressive results for grasping in clutter scenes through matching graspable end-effector volumes against observable point clouds, as a complement to distinguishing individual objects. Rosman and Ramamoorthy [16] are able to estimate a relational scene graph for objects in contact as a collection of axioms. Papazov et al. [15] perform rigid registration of known object geometries to point cloud data, using methods based on the Iterative Closest Point (ICP) algorithm. The approaches mentioned above are quite useful for manipulation, but require discriminable features that can be directly observed. In terms of utilizing physics, Dogar et al. [8] have incorporated quasi-physical prediction for grasping heavily-occluded non-touching objects cluttered on flat surfaces.

In terms of generative inference, there has been considerable work in using physics within Bayesian filtering models for tracking of people [4], [23] often for locomotion-related activities. Such physics-informed tracking applied to manipulation scenes presents new challenges as the complexity of several interacting objects introduces more complex contact and occlusion dynamics. Outside of robotics and manipulation, recent work by Wu et al. [24] estimated the physical properties of an object using physics engine with deep learning techniques over an input video. Work by Jia et al. [10] used physics stability to improve the RGBD-segmentation of objects in clutter that could eventually be used to estimate 3D geometry for manipulation. Liu et al. [13] used knowledge-supervised MCMC to generate abstract scene graphs of the scene from 6D pose estimates from



(a) Touching case

(b) Stacking case

(c) Slant case

(d) Occlusion case

Fig. 2: Motivational cases for the primitive object interactions, commonly seen in cluttered scenes: Left to right, Real world scene, depth observation, point cloud view and estimated scene (using our approach) in blender view

uncertain low level measurements. Joho et al. [11] used Dirichlet process to reason about object constellations in a scene, helping unsupervised scene segmentation and completion of a partial scene. Zhang et al. [25] formulated a physics-informed particle filter, G-SLAM, for grasp acquisition in occluded planar scenes. Sui et al. [19] proposed a similar model for estimating the entire relational scene graph and object pose demonstrated relatively small scenes with simple geometries. Narayanan et al. [14] have similar assumptions as ours and formulated the object localization task under occlusions as a multi-hueristic search problem to search over the space of hypothesized scenes. Collet et al. [7] proposed MOPED framework that uses iterative feature clustering for object recognition and pose estimation, and heavily relies on visual features. The methods above are often restricted to quite simplistic scenes and do not consider physical interaction between objects like we do.

In this work, we address these challenges by focusing on specific cases of inter-object interaction for estimating the object pose across all six degrees-of-freedom for each object. Distinguishing our work from above methods, we substantiate the accuracy of the object pose estimation by performing robotic manipulation task on the estimated scenes.

III. MOTIVATION

A cluttered scene can be defined as a scene where objects are not segregated from each other and, as a result, not optimally visible to a sensor. Because robotic applications demand reasonable precision in perception to perform even a simple pickup task, the complexity multiplies as the number of objects grow, leading to an increasingly cluttered scene. There are a vast number of object interactions that can cause a scene to be cluttered with this growth in objects. For now, we consider the form of the uncertainty caused by object interactions, and not issues of clutter that might arise with number of objects. As such, we review here the primitive

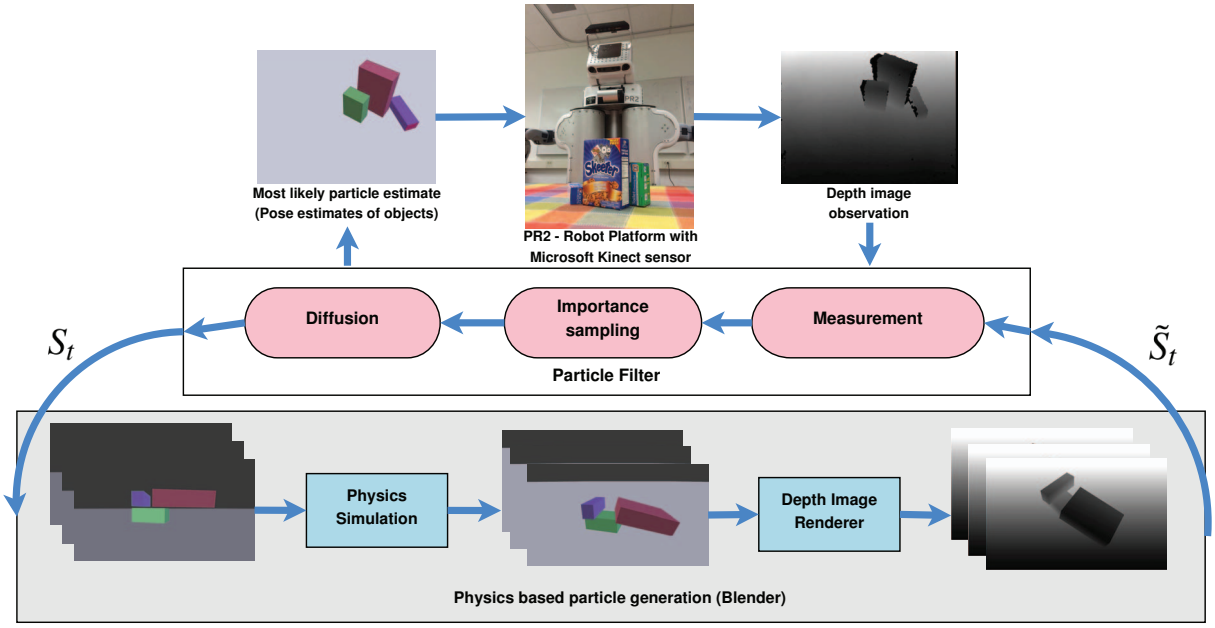


Fig. 3: System architecture for physics-informed particle filter (PI-PF)) for viable pose estimation of objects: Robot observes the scene as a depth image and infers the state by a particle filter approach, where each particle is a hypothesized scene rendered by a graphics engine followed by a physics projection to ensure its plausibility in the real world. After iterating for a set of particles with measurement update and diffusion, the most likely particle is estimated to be the state of the scene.

cases of cluttered from physical object interaction: a) objects touching each other, b) objects stacked on top of each other, c) slant objects supported by either their edge or face and, d) objects completely occluded from view by other objects. General clutter scenes are some combination of these four cases.

A. *Object touching*

Consider a case where two objects touch, as shown in Fig. 2 (a), with similar texture and appearance. From the depth sensing, these objects could be segmented as a single cluster of objects from the tabletop. However, there is no discriminable depth discontinuity between the objects. Under-constrained and discriminative methods that depend on features, such as corners or pre-segmentation, often fail to estimate the touching cases reliably. Our proposal to use a generative approach can be advantageous in these scenarios as shown in Fig. 2 (a).

B. *Object stacking*

Another frequent interaction between objects is stacking. Consider a two object stacking case as shown in Fig. 2(b), where the top object is close to the edge of the bottom object. The depth data as seen in the point cloud view of Fig. 2(b) is very sparse. RGBD feature extraction and/or discrimination might be able to detect the objects in the scene but precise pose estimation would still be a problem as it will depend on the sparse depth data observed. Further, an ambiguous pose estimation might lead to states that are not physically plausible. For example, an estimate could have poses with the center of the mass of the top object away from the edge of the bottom object, towards unsupported space. This results in a state estimate that is not plausible with the physics of the environment. Therefore, we claim that integrating physics as a part of the estimation process is essential to reject such implausible hypotheses and converge to the ground truth scene as shown in the Fig. 2(b).

C. *Object slant*

Cluttered scenes may also include piles of objects, which produces cases where objects are not just supported by one of their faces, but by their edges and corners. Consider two objects slant case as shown in Fig. 2(c), where one object is oriented such that its mass is supported both by the table and the other object. With the sparse depth data as shown in the Fig. 2(c), pose estimation of the slant objects is challenging. In addition, a wrong estimation of the pose of the slant object might lead to objects inter-penetrating. Our proposed method is able to handle the slant object cases which requires consideration of an object‘s possible inter-penetration and its physically plausible constraints.

D. *Object occlusion*

Object occlusion is another common problem in cluttered scenes; it ranges from partial occlusion to complete occlusion of objects. Consider two objects as shown in Fig. 2(d), where one object is on top of a second object that is not visible to the sensor. This configuration results in the data driven approaches being unaware of the bottom object, unless a prior informs of the bottom object being at a known location. A generative approach, such as ours, hypothesizes object poses that produce scenes matching to the observation shown in Fig. 2(d). Occluded objects will have multiple pose hypotheses that generate scenes to best match the observation. Our Bayesian filter approach maintains a distribution over these possible poses and estimates the likely pose of the

occluded object in the next time frame when the scene is acted upon by a robot.

IV. PHYSICS-INFORMED PARTICLE FILTER

We denote our physics-informed particle filter as PI-PF. We model this problem of pose estimation as a recursive Bayesian filter, a common model used for state estimation in robotics [21]. The Bayesian filter is described by the following equation, with X_t being the state of the scene X at time t, sensory observations Z_t, control actions U_t taken by the robot:

$$p(X_t|Z_{1:t}) \propto p(Z_t|X_t) \int p(X_t|X_{t-1},U_t)p(X_{t-1}|Z_{1:t-1})\mathrm{d}X_{t-1}. \quad (1)$$

Scene state X_t is a set of object poses in the scene, represented as $X_t = \{p_1, p_2, p_3, ...p_m\}$. Pose of an i^{th} object in a scene state is $p_i = \{x_i, y_i, z_i, \varphi_i, \theta_i, \psi_i\}$ where x_i, y_i, z_i are the 3D position of the center of mass and φ_i, θ_i, ψ_i are three Euler angles parameterizing the rotation in space. $S_t = \{X_t^1, X_t^2, X_t^3, ...X_t^N\}$ represents a set of scenes or particles before physics plausibility projection. $\tilde{S}_t = \{\tilde{X}_t^1, \tilde{X}_t^2, \tilde{X}_t^3, ...\tilde{X}_t^N\}$ represents a set of scenes or particles after physics plausibility projection. U_t is the sum of the user forces applied to the set of objects, which will be zero for this current work.

Our proposed framework consists of two major components: a particle filter and the physics based particle generator (Fig. 3). Initially, a set of n particles is generated randomly (uninformed by the observation) to form S_t states. Each particle X_t^j is physically projected to a state $\tilde{X}_t^j$ and thus forms $\tilde{S}_t$ set of states. The particle filter consists of *measurement*, *importance sampling* and *diffusion* submodules. The *measurement module* takes in the observation Z_t in the form of depth image given by the robot's depth sensor and physically viable particles $\tilde{S}_t$ generated by the physics based particle generator (a set of depth images rendered by a 3D z-buffer renderer). The *measurement module* compares each of the particle $\tilde{X}_t^j$ represented as depth image with the observation Z_t using sum squared distance function over every pixel. This comparison gives the likelihood of each particle being close to the observation. The *importance sampling* module takes the likelihood of all the particles to perform resampling of states, based on their likelihood. This process generates more particles created with the states that were plausible. These states are diffused by the *diffusion* submodule to provide the states for the next iteration S_t.

It should be noted here that the states S_t generated by the *diffusion module* are not guaranteed to be physically viable. Therefore, physics based particle generator takes the states produced after the diffusion from the filter and projects it to $\tilde{S}_t$. These projected states are then rendered out as depth images and the process continues till the convergence is reached.

As alluded to above, the sequential Bayesian filter in Eq. 1 is commonly approximated by a collection of N weighted particles, $\{X_t^{(j)}, w_t^{(j)}\}_{j=1}^N$, with weight $w_t^{(j)}$ for particle $X_t^{(j)}$, expressed as:

$$p(X_t|Z_{1:t}) \propto p(Z_t|X_t) \sum_j w_{t-1}^{(j)} p(X_t|X_{t-1}^{(j)}, U_{t-1}) \quad (2)$$

From this approximation, we will still resample as in standard particle filtering by drawing N updated samples:

$$X_t^{(j)} \sim \pi(X_t|X_{t-1}^{(j)}, U_{t-1}). \quad (3)$$

Because $X_t^{(j)}$ are potentially physically implausible, we will apply a function f to each of these drawn samples to produce a new set of physically-plausible particle hypotheses:

$$\tilde{X}_t^{(j)} = f(X_t^{(j)}, V_t^{(j)}, h). \quad (4)$$

where $f(X_t^{(j)}, V_t^{(j)}, h)$ is the function integrating a model of Newtonian physics forward in time by h seconds from the positions $X_t^{(j)}$ and velocities $V_t^{(j)}$ of objects in a scene. Because we are considering static scenes, it should be noted that both the object velocities $V_t^{(j)}$ and control forces U_t are assumed to be zero in magnitude. The resulting set of physically-viable particles are used to form an approximation of the posterior at time t by computing the new weights $\tilde{w}_t^{(j)}$ through evaluating their likelihood:

$$\tilde{w}_t^{(j)} = p(Z_t|\tilde{X}_t^{(j)}), \quad (5)$$

and normalizing to sum to one:

$$w_t^{(j)} = \frac{\tilde{w}_t^{(j)}}{\sum_k \tilde{w}_t^{(j)}}. \quad (6)$$

Although we are considering static scenes, it should also be noted that the particle filter is able to perform tracking over time for moving objects as well with non-null object velocities and control forces.

With regard to function f, given the geometry of a rigid object and its physical properties (mass, inertia and friction), a stable position and orientation of this object can be computed with gravitational and contact forces using a physics simulator. We cast *physical plausibility projection*, as the process of submitting a state X_t^j of the scene, which might not be physically plausible or stable, as an initial condition of the physics simulator in order to generate a guaranteed physically plausible and stable state $\tilde{X}_t^j$ at the end of the simulation.

An example of physics projection is shown in the Fig. 3. The scene state from the diffusion module is not guaranteed to be physically stable. As shown in Fig. 3, the green object is stable on the surface, whereas the other two objects are floating in the air. When a scene goes through the physical simulation, it is projected to a state that is physically stable as shown in the Fig. 3. This projection could lead to stacking and slant cases as in this example where the blue object is stacked on top of green and the red object rests in a slant position supported by the green object. There are many other physically implausible cases such as object inter-penetrations and center of mass not fully supported by other objects in the scene, that can be projected to a stable scene with this physics projection. These examples show how physics brings realism to the estimation process, making it a plausible perception.

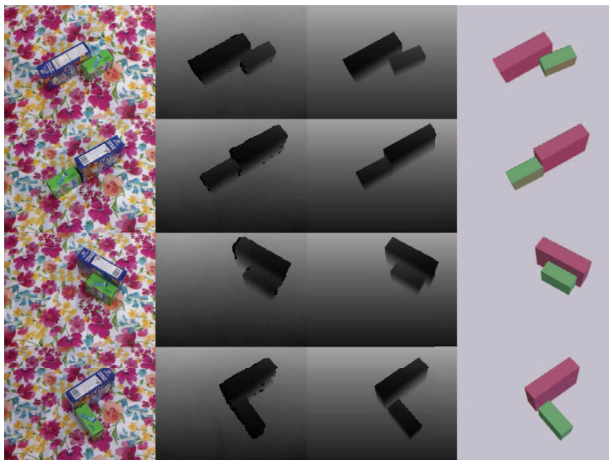


Fig. 4: Objects touching experiment results: From left Original Scene, Observed depth image, Estimated most likely scene as a depth image, Blender camera view of the estimated scene

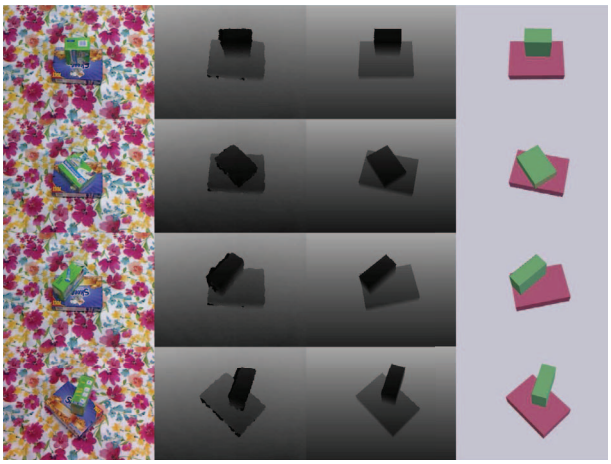


Fig. 5: Objects stacking experiment results: From left Original Scene, Observed depth image, Estimated most likely scene, Blender camera view of the estimated scene

A. Physics-informed Markov Chain Particle Filter

We explored Markov Chain Monte Carlo (MCMC) [9], a popular method employed for inference in scene estimation problems. To integrate physically stable sampling strategy into the single-site Metropolis Hastings algorithm [9], every new sample X^* generated from proposal distribution $q(X_t^*|\tilde{X}_{t-1})$ has to be physically projected, where $\tilde{X}_{t-1}$ is the previous sample. The proposal distribution $q(X_t^*|\tilde{X}_{t-1})$ is defined as a $\mathcal{N}(\tilde{X}_{t-1},\Sigma)$, where Σ is the same as used in the diffusion of PI-PF. It should be noted that the generated sample X_t^* is not guaranteed to be a physically plausible state. Hence, we project the X_t^* to $\tilde{X}^*$ using function f as shown in Eq 4.

The physics projection of the new sample makes the random walk in the neighborhood no more a useful sampling technique. Hence, we discarded the direct application of MCMC method with physics plausibility check and instead integrate MCMC in our PI-PF method to improve the posterior distribution represented by the collection of the particles. This method of inference is inspired by Khan et al. [12] for MCMC in particle filter for tracking. Once we have $\tilde{S}_t$, a set of physically viable particles in PI-PF at iteration t, we sample a different particle as proposed by $q(X^{*(j)}|\tilde{X}_t^{(j)})$ to get $S_t^* = \{X_t^{*1}, X_t^{*2}, X_t^{*3}, ...X_t^{*N}\}$. S_t^* is then physically projected to get $\tilde{S}_t^* = \{\tilde{X}_t^{*1}, \tilde{X}_t^{*2}, \tilde{X}_t^{*3}, ...\tilde{X}_t^{*N}\}$. Now, an acceptance probability check is performed on each particle $\tilde{X}_t^{*(j)}$, to either accept or reject each of these new samples to get a new set $\tilde{S}_t$ for the iteration t. The acceptance probability check is defined as below.

$$A(\tilde{X}_{t-1}^{(j)}, \tilde{X}_t^{*(j)}) = min\Big\{1, \frac{L(\tilde{X}_t^{*(j)})}{L(\tilde{X}_{t-1}^{(j)})}\Big\}. \quad (7)$$

where $L(X_t)$ is the likelihood of a state X_t given by the below equation.

$$L(X_t) = p(Z_t|X_t) \quad (8)$$

When $A(\tilde{X}_{t-1}^{(j)}, \tilde{X}_t^{*(j)})$ is 1, then the new sample $\tilde{X}_t^{*(j)}$ is accepted to be $\tilde{X}_t^j$, else a random number α from $\mathcal{U}(0,1)$ is used to reject the new sample if $\alpha > A(\tilde{X}_{t-1}^{(j)}, \tilde{X}_t^{*(j)})$ and retain the previous sample ($\tilde{X}_t^{(j)} = \tilde{X}_{t-1}^{(j)}$). Now, the particles $\tilde{S}_t$ goes through the *importance sampling* module and then *diffusion* module to follow the particle filter approach. We denote this method as PI-MCPF for the rest of the paper.

V. EXPERIMENTAL DETAILS AND RESULTS

In this section, we give details about our implementation. We compare the proposed methods (PI-PF and PI-MCPF) with a baseline ICP based method on the primitive object interaction cases. We report our observations and demonstrate the methods on complex scenes. We use Blender v2.74 [1] binaries, along with its Python support and built-in implementation of Bullet [2] physics simulator. Prior to the experiment, a template scene is created in Blender with a camera, 3D object meshes and a supporting surface that acts as the table. We used real world objects with cuboid geometry for our experiments, whose object meshes are trivial to create in Blender using their real dimensions. For every experiment, the system is provided with the number of objects in the scene and their geometries in the form of meshes. We assume that an ideal recognition system provides this information without localizing the geometries in the scene. We used the default density value (1.0) in Blender for our experiments, which makes the object mass equal to its volume. All the object meshes in the scene are set as *active rigid bodies*, which means they react to collision and are subjected to gravitational forces. The supporting surface created is set to behave as a *passive rigid body*, which means it reacts to collisions but is not subjected to gravity (i.e. it interacts with objects but stays fixed in the scene). A Microsoft Kinect depth sensor mounted on top of the PR2 is externally calibrated with respect to the table using AR_Marker package *ar_track_alvar* from ROS providing extrinsic parameters. This calibration helps in creating a virtual supporting surface in Blender. After the template scene's blend file is created, at every iteration of the particle filter, the S_t set of scenes are loaded in parallel on multiple instances of Blender. In each

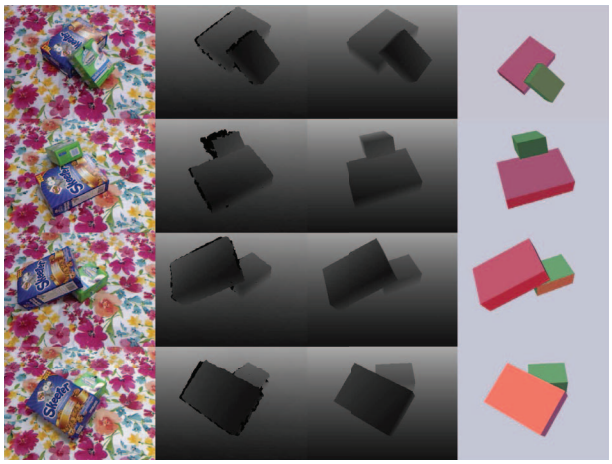


Fig. 6: Objects slanted experiment results: From left Original Scene, Observed depth image, Estimated most likely scene, Blender camera view of the estimated scene

instance, a particle X_t^j is loaded to set the pose of the object meshes and then physics rigid body simulation is triggered to project each of the states from X_t^j to $\tilde{X}_t^j$. Blender rigid body simulation requires few critical parameters: we set up the friction coefficient to 0.75, rigid body sim frame_end at 500 (threshold to end the simulation), solver iterations at 60 and steps per second at 750. We found these parameters to be optimal for realistic physics simulation of the cuboid geometries considering its computation time. Depth images are rendered in HDR format to extract the exact metric information from the OpenGL renderer of Blender. We used 1444 particles for all our experiments. For primitive cases, PI-PF method was run for 150 iterations and PI-MCPF method was run for 70 iterations. For complex scenes, PI-PF method was run for 250 iterations and PI-MCPF method was run for 150 iterations.

In the below subsections, we discuss the implementation of baseline ICP method and compare its results with our proposed methods on the primitive cases considered in Section III. For the base clutter scenes, we created scenes which are difficult with insufficient depth data for traditional discriminative methods of object segmentation, object detection or pose estimation to perform robustly. The base clutter experiments involves two objects in touching, stacking and slant positions and also in complete occlusion. We experiment on 7 touching scenes, 7 stacking scenes, 7 slant cases and 7 complete occlusion scenes.

A. Iterative Closest Point method

To the best of our knowledge, we have not come across a state-of-the-art method that works on the depth data (with no visual features) and deal with occlusions due to physical object interactions. Hence, we created a baseline with Iterative Closest Point (ICP) to estimate object poses in a scene. ICP takes in two point clouds namely the source cloud and the target cloud, and finds the transformation between them by iteratively by minimizing their point-to-point distance. This procedure requires the source and target to contain the same object to perform optimally. To provide

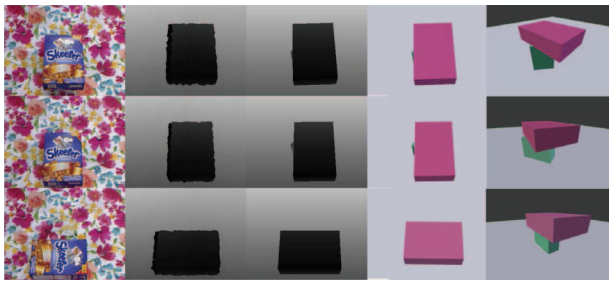


Fig. 7: Objects occluded experiment results: From left Original Scene, Observed depth image, Estimated most likely scene, Blender camera view of the estimated scene with an additional view to show how the occluded object's pose is estimated by our method

this advantage to ICP based method, the 3D point cloud of each scene in the base clutter cases is processed in two stages: 1) the table background is subtracted by removing the largest plane in the scene using plane segmentation from PCL (Point Cloud Library) [17] resulting in a foreground point cloud of interest 2) each of the two objects are manually segmented from the foreground cloud resulting in two object point clouds (as the base clutter scene experiments contain only two objects). Point cloud of each object geometry is synthetically generated based on their dimensions and considered as source clouds for ICP matching. Each of these source clouds are matched with their respective target clouds segmented from the the scene. ICP matching is prone to be sensitive to the initialization of the source point cloud. Initial position (x, y, z) of the source clouds are generated randomly above the table level. The orientation (φ, θ, ψ) of these source clouds are set to the 3 principle components of their respective target clouds. For each scene, 50 randomly initialized source clouds of the objects are used to perform the ICP matching.

B. Base Clutter Scene Results

In the touching cases, two objects are placed in different orientations on the table, touching each other as shown in Fig. 4. We show the cases where objects are in contact on their edges or their faces. It is observed that the estimates of these cases using PI-PF and PI-MCPF methods are close to the ground truth with average errors in position and angles as shown in Table I. ICP on the object segments fail with large pose errors as they are not physically informed about their object boundaries leading to inter-penetrations.

In the stacking cases, two objects are placed in different orientations on table, with one object placed on top of the other object. This other object is supported by the table as shown in Fig. 5. Note, that we used only small objects to be on top of the larger object, because the converse structure creates complete occlusion, which is discussed in the following set of experiments. It is observed that, in order to generate stacking scenes using physics projection, the diffusion of the resampled $\tilde{S}_t$ states should accommodate elevation of objects randomly. This diffusion creates S_t. We observed that the estimated scenes using PI-PF and PI-MCPF methods are close to the ground truth with average errors in position and angles as shown in Table I. ICP based approach

Category	Error	ICP on Object Segments				PI-PF				PI-MCPF			
		Large Obj		Small Obj		Large Obj		Small Obj		Large Obj		Small Obj	
		(mean)	(var)	(mean)	(var)	(mean)	(var)	(mean)	(var)	(mean)	(var)	(mean)	(var)
Touching	Position (cm)	9.58	4.99	10.7	3.68	**1.83**	0.18	**1.75**	0.11	2.10	0.15	2.10	0.50
	Roll (deg)	25.9	4.51	62.0	0.14	0.19	0.05	0.30	0.20	**0.17**	0.05	**0.23**	0.19
	Pitch (deg)	34.0	2.71	38.8	2.61	0.05	0.00	**0.05**	0.01	**0.03**	0.00	**0.05**	0.00
	Yaw (deg)	28.8	2.03	33.1	8.45	**1.86**	3.06	**1.10**	0.58	2.30	3.23	8.70	3.06
Stacked	Position (cm)	11.3	1.83	13.0	0.94	2.19	0.60	**2.23**	0.20	**1.84**	0.99	2.67	0.85
	Roll (deg)	32.2	0.13	37.6	0.54	0.53	0.37	0.77	1.13	**0.48**	0.57	**0.79**	1.77
	Pitch (deg)	37.1	0.63	26.2	2.20	**1.09**	3.81	1.54	2.59	1.35	5.45	**1.18**	3.19
	Yaw (deg)	57.5	3.04	38.4	2.59	**4.71**	6.74	**6.05**	5.86	5.50	8.43	6.63	8.32
Slant	Position (cm)	10.4	5.23	14.3	0.72	**3.09**	5.51	4.38	11.4	4.42	5.90	**4.33**	6.82
	Roll (deg)	36.9	8.97	39.3	2.50	14.5	86.5	0.38	0.10	**0.54**	1.02	**0.33**	0.10
	Pitch (deg)	38.8	0.29	33.4	2.94	**1.58**	2.97	31.5	23.3	5.96	69.3	**19.4**	74.3
	Yaw (deg)	19.9	2.78	27.6	1.93	10.5	84.3	**30.7**	42.4	**10.3**	19.9	36.5	31.6
Occluded	Position (cm)	26.7	2.33	NA	NA	**2.83**	1.47	**4.23**	5.65	3.23	2.38	4.28	5.63
	Roll (deg)	13.8	3.37	NA	NA	**20.0**	71.1	**29.9**	43.6	**20.0**	72.8	44.9	44.8
	Pitch (deg)	8.47	1.10	NA	NA	**0.05**	0.00	**30.0**	85.3	**0.05**	0.00	**30.0**	87.5
	Yaw (deg)	27.5	3.40	NA	NA	**15.0**	53.6	40.0	40.0	16.1	18.1	**33.9**	49.8

TABLE I: Object pose estimation errors are reported here with respect to the ground truth poses. Ground truth is generated by manually matching the object geometries to the observed point cloud using the Blender user interface.

fails to perform as the objects are not enforced to stack based on their poses and hence could result in floating objects.

In the slant cases, two objects are placed in different orientations on table such that one object is on the table, supporting the other object, which is in a slant pose as shown in Fig. 6. To generate slant scenarios, the rigid body simulation in Blender requires care in setting up the parameters as mentioned above. If physics projection cannot produce these slant cases, the experiments will not converge to the observed scene. As it can be seen in Fig. 6, even in the cases where the bottom object is occluded by the top slant object, its pose in the estimated depth image matches the observation. More importantly, we find that estimated state is physically plausible. We observed that slant cases are difficult, and estimates from both PI-PF and PI-MCPF methods are not as close to the ground truth as in touching and stacked cases. The average angular error is high for the small object, which is occluded in most of the cases and very hard to be estimated. On the other hand, the larger object which, even on having an advantage of being highly visible requires a trade off in matching the observation and also maintaining physical plausibility. ICP fails in slant cases too as it is not informed about both the object boundaries as well as gravitational force to support itself in a slant position.

In the occlusion cases, as shown in Fig. 7, the small object is completely occluded by the larger object in the observation. Our proposed methods PI-PF and PI-MCPF robustly handles these cases and estimates the pose of the larger object with average position errors shown in Table I. However these methods have higher position errors for the smaller object that is not visible to the sensor. It should be noted that the ground truth for all these scenes were generated using visual inspection and matching of the object geometries to the observed point cloud. Because the small object was not seen in the point cloud, the ground truth was generated to just make sure physical plausibility of the scene. The last column in Fig. 7 shows the view of the estimated scene from a different viewpoint, to see the estimated pose of the occluded small object. In complete occlusion, we also had cases where the larger object was slanted on the small object, occluding the small object. Hence there is a high error in the *Roll* of the larger object similar to that of the slant cases in both PI-PF and PI-MCPF methods. ICP based method does not have the target cloud for the small object, and, thus there is no way to estimate the pose of that object.

The ICP based method purely relies on 3D data association. It is observed to fail consistently on all categories. It should also be noted that ICP will perform much worse if the 3D scene is not preprocessed. Overall the PI-PF and PI-MCPF methods perform comprehensively on these difficult primitive setups and help us develop an understanding of using physical plausibility in the estimation process of more complex scenes discussed in next section.

C. Cluttered Scene Results

We have performed experiments on three and four objects cases, that combined the base cases discussed earlier. With inclusion of additional objects, the state space for search explodes and it takes lot of iterations to converge to the ground truth. For experimental purpose the time complexity is avoided with constrains on the object poses. Poses of the objects are limited to $\{x_i, y_i, z_i, \varphi_i\}$ (i.e. φ_i is the yaw angle of an object to determine its rotation on the surface plane which is aligned to XY plane) dimensions in the initialization and updates. However physics projections at each iteration results in real valued numbers on all the $6xN$ dimensions of the scene.

In Fig. 8 we show experiments with four objects in the scene with results from PI-PF. It can be seen that the experimental set up has the combinations of the primitive cases discussed earlier. These scenes have a lot of occlusions with respect to the sensor viewpoint. The scenes are estimated using PI-PF and PI-MCPF, and are close to the ground truth poses, except for the objects that are occluded. However if

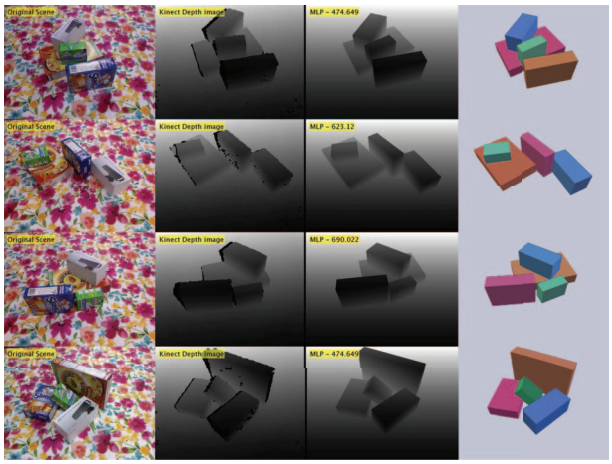


Fig. 8: Complex experiment results with 4 objects: From left to right, Original Scene, Observed depth image, Estimated most likely scene, Blender camera view of the estimated scene

Conditions	Maximum iterations	PI-PF converges	Maximum iterations	PI-MCPF converges
Touching	150	85.26	70	30.42
Stacking	150	90.84	70	53.55
Slant	150	143.7	70	70.00
Occlusion	150	70.50	70	46.98
3 Objects	250	188.2	150	113.4
4 Objects	250	224.6	150	142.1

TABLE II: Shows the average number of iterations each of methods, took to converge. Maximum iterations are the number of iterations each method is allowed to run. We consider the experiment to have converged if the change in the pose estimate of the most likely particle is less than 1 cm in position and less than 3 degrees in the angles.

a continuous perception is performed, our estimation along with the distribution over the state space will act as a prior knowledge over time. We performed sequence of object manipulations on the estimated scenes using PR2 robot, whose gripper has a small tolerance to the error in estimation. Precision to which the pose estimation is performed in PI-PF and PI-MCPF methods are good enough to let the robot perform successful manipulation. A couple of scenes are shown in the video submission with robotic manipulation on the estimated poses. We observed that the accuracy of the PI-PF and PI-MCPF are close to each other in all the experiments performed, but the number of iterations taken by PI-PF is higher compared to PI-MCPF as shown in Table II.

VI. CONCLUSION AND FUTURE WORK

In this paper, we propose a generative, probabilistic scene estimation using Newtonian physical simulation for physically plausible scene estimation to enable robotic manipulation in clutter. Our method estimates cluttered scenes as a collection of object poses to generate and match observation. We discuss primitive cases causing observation uncertainty due to object interactions like touching, stacking and slant support poses. We present cases where physical plausibility is at most essential in robotic perception and show results using our framework on some difficult cases of clutter settings. We explored variants of our approach and report the results with observations on each case. The current framework is limited by its computational demands in estimating cluttered scenes with large number of objects. As a next step, we would like to scale it to different object geometries as well as work on GPU implementation for real time robotic manipulation tasks.

VII. ACKNOWLEDGMENTS

This work was supported in part by grants from the Office of Naval Research (award N00014-08-1-0910) and NASA (award NNX13AN07A).

REFERENCES

[1] Blender, an open-source 3d computer graphics software. `www.blender.org`.
[2] Bullet physics library. `www.bulletphysics.org`.
[3] PR2 interactive manipulation. `http://wiki.ros.org/pr2_interactive_manipulation`.
[4] M. A. Brubaker, D. J. Fleet, and A. Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 2010.
[5] C. Choi and H. I. Christensen. Rgb-d object tracking: A particle filter approach on gpu. In *IROS, 2013*.
[6] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*. 2014.
[7] A. Collet, M. Martinez, and S. S. Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *IJRR*, 2011.
[8] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa. Physics-based grasp planning through clutter. In *RSS*, 2012.
[9] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.
[10] Z. Jia, A. C. Gallagher, A. Saxena, and T. Chen. 3D reasoning from blocks to stability. *PAMI*, 2015.
[11] D. Joho, G. D. Tipaldi, N. Engelhard, C. Stachniss, and W. Burgard. Nonparametric bayesian models for unsupervised scene analysis and reconstruction. *Robotics*, page 161, 2013.
[12] Z. Khan, T. Balch, and F. Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *ECCV 2004*.
[13] Z. Liu, D. Chen, K. M. Wurm, and G. von Wichert. Table-top scene analysis using knowledge-supervised MCMC. *Robotics and Computer-Integrated Manufacturing*, 33:110–123, 2015.
[14] V. Narayanan and M. Likhachev. Perch: Perception via search for multi-object recognition and localization. *ICRA*, 2016.
[15] C. Papazov, S. Haddadin, S. Parusel, K. Krieger, and D. Burschka. Rigid 3D geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, 2012.
[16] B. Rosman and S. Ramamoorthy. Learning spatial relationships between objects. *Int. J. Rob. Res.*, 30(11):1328–1342, Sept. 2011.
[17] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, 2011.
[18] R. B. Rusu, Z. C. Marton, N. Blodow, M. Dolha, and M. Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 2008.
[19] Z. Sui, O. C. Jenkins, and K. Desingh. Axiomatic particle filtering for goal-directed robotic manipulation. In *IROS, 2015*.
[20] A. ten Pas and R. Platt. Localizing handle-like grasp affordances in 3d point clouds. In *International Symposium on Experimental Robotics*. Citeseer, 2014.
[21] S. Thrun, W. Burgard, and D. Fox. *Probabilistic robotics*. MIT press, 2005.
[22] M. Vondrak, L. Sigal, and O. Jenkins. Physical simulation for probabilistic motion tracking. In *CVPR*, 2008.
[23] M. Vondrak, L. Sigal, and O. C. Jenkins. Dynamical simulation priors for human motion tracking. *PAMI*, 2013.
[24] J. Wu, I. Yildirim, J. J. Lim, B. Freeman, and J. Tenenbaum. Galileo: Perceiving physical object properties by integrating a physics engine with deep learning. In *NIPS*, 2015.
[25] L. E. Zhang and J. C. Trinkle. The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing. In *ICRA, 2012*.