Object Manipulation in Cluttered Scenes Informed by Physics and Sketching

Karthik Desingh*, Mehran Maghoumi†, Odest Chadwicke Jenkins*, Joseph J. LaViola† and Lionel Reveret‡
*Department of Computer Science and Engineering, Unversity of Michigan, Ann Arbor, USA
†Department of Computer Science, University of Central Florida, Orlando, USA
‡INRIA Rhône-Alpes, Saint Ismier, France

***Abstract*—In this paper, we propose a framework to enable an autonomous robot to manipulate objects in cluttered scenes. Manipulation of objects in a complex cluttered scene demands an extremely precise pose estimation system. In order to precisely estimate object poses, a database of the objects should be acquired from earlier encounters. Hence, in addition to the pose estimation, a system to aid object data collection for building the database is necessary. We consider the estimation and data collection as two modules of our framework: (1) a physics informed pose estimation system and (2) object geometry extraction using sketch-based interface. In this paper, the current state of these two modules are presented with results, and benefits of their combination is discussed.**

I. Introduction

Our goal is to enable a manipulation platform, such as the Fetch robot, to pick-and-place objects in cluttered indoor scenes. Grasping an object and performing manipulation requires that the 6DOF pose of the object be known. An object's pose in a scene is generally estimated using an RGB image and/or a depth image from the robot's sensor. The difficulty in pose estimation is directly proportional to the complexity of a cluttered scene, which in turn is related to the number of objects, their physical interactions with each other and their geometrical shapes. The physical interactions between objects, such as touching, stacking and supporting, results in object occlusions at various levels. When a depth sensor senses object occlusions, the data acquired is not sufficiently complete for performing robust object pose estimation. Even if the object geometries are assumed to be simple cuboid shapes, the pose estimation under object occlusions is still challenging.

One way to generalise the pose estimation to a wide range of object geometries is to create a database of objects. Because existing object datasets [19] fail to represent objects in every environment, a data collection phase is required to add novel objects regularly to the database when encountered. Developing these two modules — one for object pose estimation and one for object geometry collection — should provide the data needed for a robot to precisely manipulate objects in cluttered indoor scenes.

In the object pose estimation module, we assume there exists an ideal object database, which retrieves a list of objects in the scene. Each object in the list is associated with a geometry in the form of a mesh. Based on this assumption, we explore ways to estimate the pose of each object in a scene. However, such an inference is fraught with challenges (e.g., occlusions and physical contacts) that prevent acceptable levels of perceiving the scene and, consequently, manipulating the objects. Even when object geometries are assumed to be known, the estimation of even a single object is a challenge as has been shown in recent research [5]. The challenge for scene perception becomes even 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 [1], essentially removing the challenge of clutter. Addressing this challenge for cluttered environments, we posit that physical plausibility is a necessary component in object pose estimation. For example, consider the case where the robot is 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. Other physically implausible scene estimates can also occur due to inter-penetrating objects, unsupported objects, and unstable structures.

In our work, object pose estimation is formulated as a scene estimation problem, where each scene is a collection of object poses representing a state of the scene. We introduce a means for incorporating physical plausibility into generative probabilistic scene estimation using Newtonian physics simulation. Assuming geometry, friction, and mass properties of objects, we formulate the inference as a physics-informed scene estimation for static environments. In each of these methods, we use a physics simulation engine to constrain inference to the set of physically plausible scene states, which we treat as a *physical plausibility projection*. Following the tenets of Bayesian filtering, we describe a physics-informed particle filter (PI-PF) that uses physical plausibility projection to correct any implausibility that may occur due to additive diffusion. The performance of our pose estimation module is discussed with results from primitive cases of cluttered scenes with two objects and more complex scenes.

For the data collection module, we developed a sketch-based interface to extract objects, as and when encountered by the robot. The object data extraction is performed on RGBD data as seen by the robot. This sensor data is fed to a sketch-based system that enables a human operator to see the target scene. The operator then performs a sequence of strokes to aid the system in extracting the objects in the scene. This interaction to aid the object data collection is known as *shared autonomy*. The data extraction must include the object's physical

and geometrical properties, which are essential inputs to the physics informed pose estimation module. We first focus on the geometrical properties of the object in the data extraction. We develop a system that lets the user sketch directly on the point cloud data (RGBD data) generated by single view depth and color images. The use of various sketching strokes allows the extraction of the complete object geometry of an arbitrary object. The robot can then use the geometries for grasping and manipulation; that is, the user can instantly instruct the robot to perform a grasp action on the object using the interface. In addition to point cloud data, our system can also work with meshes of fully reconstructed scenes to produce better object geometries. We discuss the current status of the sketch-based system in this paper and tabulate results both on the quality of the geometry extracted on Bigbird [19] dataset objects and also on direct robot manipulation given these geometries.

Although two modules are discussed individually in this paper, our eventual goal is to combine them to create an end-to-end perception-to-manipulation pipeline to enable a robot to manipulate objects in cluttered scenes.

II. Related Work

A. Inference methods for object manipulation

The problem addressed by our physics-informed particle filter (PI-PF) 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. [17], the PR2 Interactive Manipulation pipeline [6] is able to perform relatively reliable pick-and-place manipulation for tabletop settings. This pipeline does not account for physical interactions between objects.

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 and Platt [23] have shown impressive results for grasping in cluttered scenes through matching graspable end effector volumes against observable point clouds, as a complement to distinguishing individual objects. Papasov et al. [15] perform rigid registration of known object geometries to point cloud data, using methods based on the Iterative Closest Point algorithm. The approaches mentioned above 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 generative inference, there has been considerable work in using physics within Bayesian filtering models for tracking of people [4, 26], 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. Work by Zhaoyin et al. [11] 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. [14] used knowledge-supervised MCMC to generate abstract scene graphs of the scene from 6D pose estimates from uncertain low level measurements. Joho et al. [12] used the Dirichlet process to reason about object constellations in a scene, helping unsupervised scene segmentation and completion of a partial scene. Zhang et al. [27] formulated a physics-informed particle filter, G-SLAM, for grasp acquisition in occluded planar scenes. Sui et al. [22] proposed a similar model for estimating the entire relational scene graph and object pose demonstrated relatively small scenes with simple geometries. The methods above are often restricted to quite simplistic scenes due to the computational issues of generative inference as the state space grows. In this work, we address these challenges by focusing on specific cases of inter-object interaction and perform robotic manipulation task on the estimated poses.

B. Object geometry extraction and sketch interfaces

Determining the geometrical properties of the objects in a scene is closely related to the task of 3D segmentation. Although there have been many attempts at tackling this segmentation problem (such as [16] or [21]), there is still a considerable difference between the performance of an automatic approach and that of a shared autonomy approach. For instance, a human can guess some geometric properties of objects such as symmetry just by glancing at a vague picture of that object. Although the gap between the performance of the human and the machine is closing, we believe that a human-in-the-loop system is beneficial in extracting information about the robot's working environment.

Our goal is to incorporate human knowledge into the task of robotic manipulation using sketch-based interfaces. Our intuition is that sketching using pen and paper is a natural and expressive means of communication between humans and computers [9]. With the rise of pen-equipped tablets, the task of entering and collecting expressive information in the form of 2D drawings (such as lines, arcs and shapes) has become increasingly simple.

Recently, Valentin et al. [25] presented an object recognition system that was trained using sparse user input in real time and was capable of recognizing similar object instances in other scenes. The user was also able to provide feedback to the system to alter its behavior. They showed that such user input was beneficial to the performance of the system. The key difference of our work from [25] is the ability to do geometry extraction to support robot manipulation.

Leveraging sketch-based interfaces for human-robot interaction, has been explored in the literature. Skubic et al. [20] used a sketch-based interface to control a team of robots while Shah et al. [18] created a sketching interface for natural robot control. Their system would recognize the user's commands and relay them to the robot. Correa et al. [7] created a multimodal system to interact with an autonomous forklift. The key difference between these works and the current work is that we use human-in-the-loop interactions to augment the robot's understanding of a scene rather than issuing direct commands to the robot.

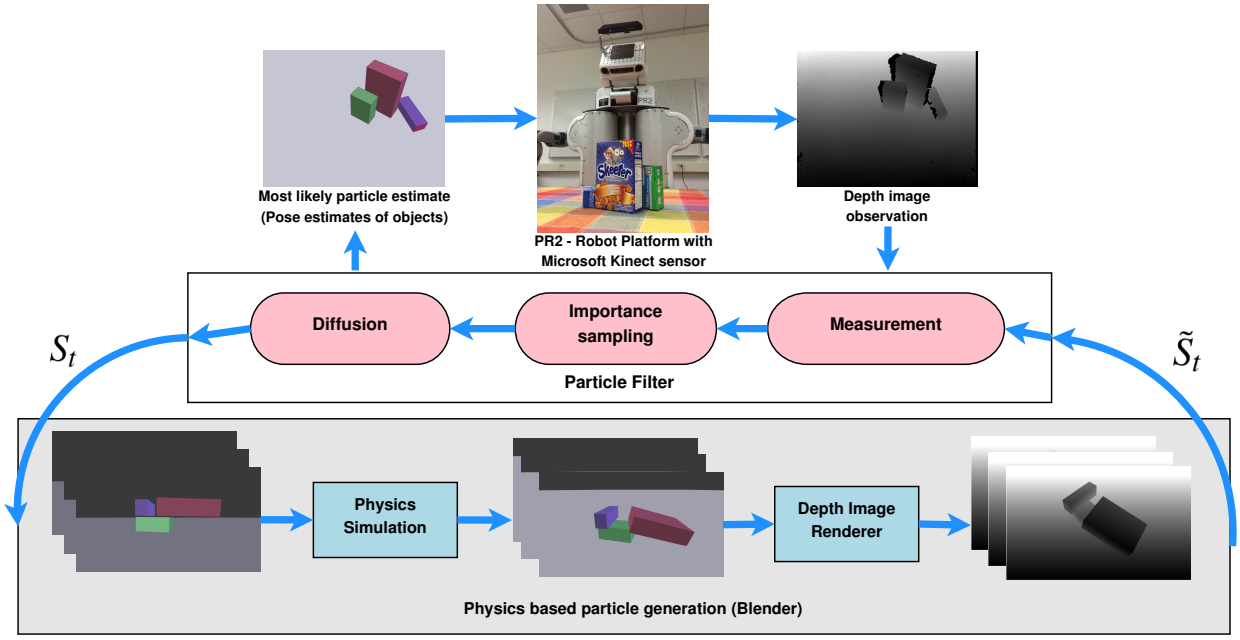


Fig. 1: System Architecture for physics-informed particle filter (PI-PF)) for viable pose estimation of objects: Robot observes the scene as a depth image and infers the state by a particle filter approach, where each particle is a hypothesized scene rendered by a graphics engine followed by a physics 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.

III. Physics Informed Scene Estimation

We denote our physics-informed particle filter as PI-PF to compare with other variants in the paper. We model this problem of pose estimation as a recursive Bayesian filter, a common model used for state estimation in robotics [24]. 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 work.

Our proposed framework consists of two major components: a particle filter and the physics based particle generator (Fig. 1). Initially, a set of n particles is generated randomly 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 module*, *importance sampling* and *diffusion* as submodules. The *measurement module* takes in the observation Z_t in the form of depth image given by the Kinect sensor from a PR2 robot and physically viable particles $\tilde{S}_t$ generated by the physics based particle generator (a set of depth images rendered by the renderer). The *measurement module* compares each of the particle $\tilde{X}_t^j$ represented as depth image with the observation Z_t using a sum squared distance function and outputs the likelihood of each particle. 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 highly likely and physically 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, the physics based particle generator takes the states produced after the diffusion from the filter and projects it to $\tilde{S}_t$ states. 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:

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

However, 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:

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

where $f(X_t^{(j)}, V_t^{(j)}, h)$ is a 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:

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

and normalizing the sum to one:

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

Although we are considering static scenes, it should also be noted that the particle filter described will be 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 Fig. 1. The scene state from the diffusion module is not guaranteed to be physically stable. As shown in Fig. 1, the green object is stable on the surface, whereas the other two objects are floating in the air. When this scene goes through the physical simulation of the blender, they get projected to a state that is physically stable as shown in Fig. 1. This 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 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.

A. Variants of PI-PF

1) Physics-informed MCMC: We explored Markov Chain Monte Carlo (MCMC)[10], a popular method employed to solve the scene estimation problem. We integrated physics projection into the single-site Metropolis Hastings algorithm to ensure that a new sample X^* generated from proposal distribution $q(X_t^* | \tilde{X}_{t-1})$, is physically plausible, where $\tilde{X}_{t-1}$ is the previous sample. We refer to this method as physics-informed MCMC (PI-MCMC) in this paper. 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 which is similar to Eq 4. Followed by the projection, acceptance probability check is performed, which

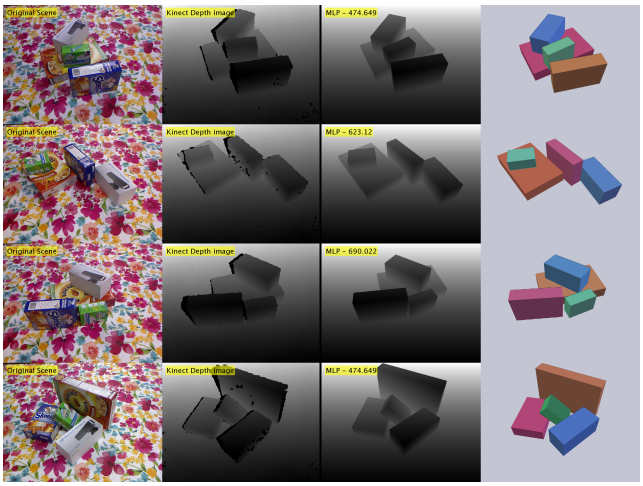


Fig. 2: Complex experiment results with four objects: From left to right: Original Scene, observed depth image, estimated most likely scene, Blender camera view of the estimated scene using PI-PF

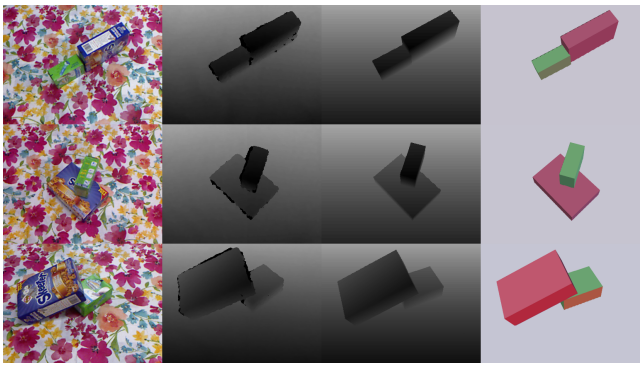


Fig. 3: One example from each of the primitive cases with objects in (from top) touching, stacking and slant poses. From left: Original scene, observed depth image, estimated most likely scene, Blender camera view of the estimated scene.

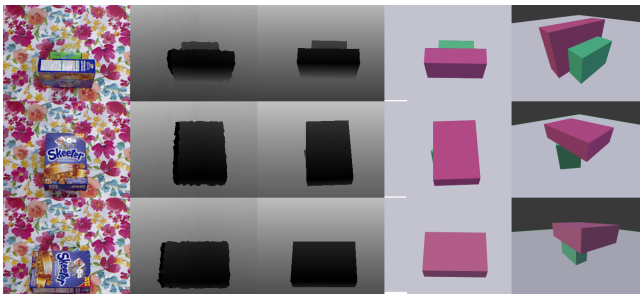


Fig. 4: Objects occluded experiment results. From left: Original scene, observed depth image, estimated most likely scene, Blender camera view of the estimated scene with an additional view to show how the occluded object's pose is estimated by our method

is defined as below:

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

where $L(X_t)$ is the likelihood of a state X_t in the measurement update. When $A(\tilde{X}_{t-1}, \tilde{X}^*)$ is 1, then the new sample $\tilde{X}_t^*$ is accepted to be $\tilde{X}_t$), else a random number α from $\mathscr{U}(0,1)$ to reject the new sample (if $\alpha > A(\tilde{X}_{t-1}, \tilde{X}^*)$ and retain the previous sample ($\tilde{X}_t = \tilde{X}_{t-1}$).

2) Physics-informed Markov Chain Particle Filter: Inspired by MCMC in particle filter for tracking [13], we integrated MCMC in our PI-PF method to improve the posterior distribution represented by the collection of the particles. Once we have $\tilde{S}_t$, a set of physically viable particles in PI-PF at iteration t, we let each of them move to a different state 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 the 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. Now the particles $\tilde{S}_t$ go 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.

B. Inference Results using Physics informed Methods

We worked on basic primitives of cluttered scenes, such as touching, stacking, slant (Fig. 3) along with complete occlusions (Fig. 4), before moving to cluttered scenes with a greater number of objects. We are motivated to use the generative approach based on the results from primitive clutter scenes particularly the complete occlusion cases. When the support object is completely occluded for a given number of objects in a scene, our approach is able to estimate the support object's approximate pose to generate a physically plausible scene. In Fig. 2, we show the scene estimation performed on four object scenes. As can be seen, the pose estimation of each of the objects is precise enough to be used for robot grasping and manipulation (Table. I). Scene estimates using PI-PF and PI-MCPF are comparable to each other. However, we noticed that the convergence using PI-MCPF was significantly less compared to that of PI-PF. PI-MCMC failed in all our scenes, as the random walk step followed by the physics projection fails to guarantee a small step in the jump. The robot manipulating the estimated scene is available on video [1].

Category	Error	Large Obj (mean)	Large Obj (var)	Small Obj (mean)	Small Obj (var)
Touching	Position (cm)	1.83	0.18	1.75	0.11
	Roll (deg)	0.19	0.05	0.30	0.20
	Pitch (deg)	0.05	0.00	0.05	0.01
	Yaw (deg)	1.86	3.06	1.10	0.58
Stacked	Position (cm)	2.19	0.60	2.23	0.20
	Roll (deg)	0.53	0.37	0.77	1.13
	Pitch (deg)	1.09	3.81	1.54	2.59
	Yaw (deg)	4.71	6.74	6.05	5.86
Slant	Position (cm)	3.09	5.51	4.38	11.4
	Roll (deg)	14.5	86.5	0.38	0.10
	Pitch (deg)	1.58	2.97	31.5	23.3
	Yaw (deg)	10.5	84.3	30.7	42.4
Occluded	Position (cm)	2.83	1.47	4.23	5.65
	Roll (deg)	20.0	71.1	29.9	43.6
	Pitch (deg)	0.05	0.00	30.0	85.3
	Yaw (deg)	15.0	53.6	40.0	40.0

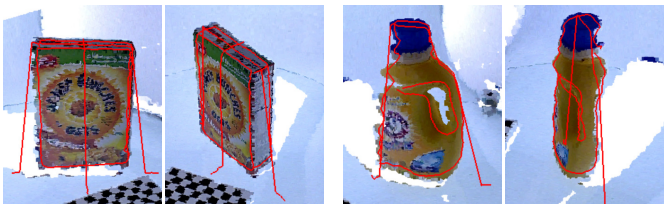
TABLE I: Object pose estimation errors are reported here with respect to the ground truth poses (generated by matching the object geometries to the observed point cloud using the Blender user interface)

IV. Sketch based Geometry Extraction

To support our PI-PF approach, it is important to have extracted geometries from a scene to seed our algorithm. A sketch-based interface provides an intuitive user driven method for gathering this geometry and ultimately supporting automatic learning of similar geometries in different scenes or repetitive object geometries with similar structure in the same scene. The sketching interface makes use of the users' physical and cognitive intuition about the scene. Our implementation works directly with *single-view* RGBD point clouds the from ASUS Xtion depth sensor. The point cloud from the sensor is rendered in such a way that it gives the illusion of looking at a regular RGB image. The model can still be rotated, scaled or translated if needed; see Fig. 5. These various view options are used by a user to sketch over the object of interest in the point cloud. The difficulty of this task is directly related to the complexity of the object of interest as well as the viewpoint of the depth sensor. In our experience, a scene captured with objects at an oblique angle is the best configuration for sketching (as seen in Fig 5).

At first the user begins to sketch by drawing some lines that define the symmetry of the object, which is referred to as *symmetrical hints*. These hints are necessary as the depth sensor is capturing only partial object in the point cloud data when viewed from a single view. The sketching system takes the user knowledge about the object in the form of *symmetrical hints* to initialize the extraction task. The user then proceeds with sketching the object in a way that would denote geometric properties of the object. We refer to these strokes as *geometrical hints*. This could be in the form of tracing the outlines of the object or coloring some regions of the image. Afterwards, the system proceeds by projecting the 2D strokes with *symmetrical and geometrical hints* on the point cloud to obtain 3D strokes. This projection step is depicted in Fig. 5. Common stroke preprocessing steps (such as smoothing or resampling) are performed on the resulting 3D strokes to compute the convex hull of the stroke points. If the strokes are representative of the approximate geometry of the object, the segmentation problem then reduces to a partitioning problem. In this partitioning problem, the goal is to determine and extract the points of the point cloud that fall inside, on the surface, or within a short distance of the convex hull of the strokes. Partitioning can be accelerated using bounding volume hierarchies (such as AABB trees [3]). The parts of the

[1] https://youtu.be/aTD5Nd-ykD4



(a) cereal box (b) detergent bottle

Fig. 5: Some examples of sketched geometries. Front view and side view of the objects viewed by the sketch interface. The protruded lines that extend on either sides of the object are the projected symmetry hints. Other lines are the rough outlines of the object being extracted.

object that are not visible to the camera are approximated using the *symmetrical hints*. To achieve that, using singular value decomposition, a 3D plane is fit to each of these strokes. After plane fitting, the reflection of the extracted points with respect to these planes is determined. Unwanted points and outliers are eliminated using the moving least squares algorithm [2]. The resulting object point cloud represents the geometry extracted using a single-view point cloud data. The pseudo-code of our implementation is given in Algorithm 1.

Input: **SP** `// 2D stroke points`
HSP `// Symmetry hint strokes`
PC `// Scene point cloud`
Output: **Segment** `// The segmented object`
PP = {} `// Projected point`
PPH = {} `// Projected hint point`
foreach *point (x, y)* ***in*** *SP and HSP* **do**
  point' = Backproject(*point*)
  if *point* $\in$ *SP* **then**
    PP = *PP* $\cup$ *point'*
  else
    PH = *PH* $\cup$ *point'*
  end
end
Smooth and filter *PP*
hull = ConvexHull(*PP*)
tree = Partition *PP* using an AABB tree
Segment = {}
foreach *point* ***in*** *3D point cloud* **do**
  distance = distance(*point*, *tree*)
  if *distance* $< \varepsilon$ ***or***
    point inside hull ***or*** *point on hull* **then**
    Segment = *Segment* $\cup$ *point*
  end
end
foreach *stroke* ***in*** *HSP* **do**
  plane = fit-plate(*stroke*)
  Segment = *Segment* $\cup$ Reflect(*Segment*, *plane*)
end

Algorithm 1. Pseudo-code for geometry extraction

To evaluate the performance of the sketch-based system, we chose 50 objects from Bigbird dataset with varying difficulty and performed geometry extraction. Hausdorff distance is used to measure the similarity of the aligned point clouds. On average, with 6 strokes on an object, the extracted object geometry is 79.91% similar to the respective mesh in the Bigbird dataset. Some of the extracted objects, along with their ground truths are shown in Fig. 6. It is important to note that the meshes in the Bigbird dataset are generated with objects viewed from various angles and different viewpoints. Our meshes are generated using user sketches on a single view point cloud. Direct manipulation on the object geometries is shown in video [2].

(a) honey_bunches_of_oats (95.64%) (b) hunts_paste (90.74%) (c) detergent (88.13%)

Fig. 6: Some examples of extracted geometries. Top row is the extracted objects while the bottom row is the ground truths. For each object, similarity of the extracted shape with the ground truth are reported.

V. Discussion

In this paper we describe two modules that will eventually work in conjunction to enable a robot to manipulate objects in cluttered scenes. They are: (1) a physics informed inference system that estimates the scene as a collection of object poses to obtain a precise pose of each object, and (2) a sketch-based geometry extraction system that extracts geometries of objects viewed by the robot. The current state of development of these modules are presented in this paper, substantiating the choice of algorithms and their results. The focus of our research is to combine these two modules in an optimal way, such that the object database built using the sketch-based system will in turn be used by the inference system to perform scene estimation. It should be noted that in our current implementation, we restricted our inference system to handle only cuboid objects. However, to generalize the inference system to handle complex shaped objects, we require an object extraction system such as the sketch-based system described in this paper. We believe that a sketch interface such as ours is the right way to extract object-related information such as geometry (focused on in this paper), visual description, affordance, physical properties

[2]https://youtu.be/aTD5Nd-ykD4

such as mass, centroid and graspable poses. This approach of extraction also enables us to generate rich data online when the robot is in live action rather than using offline object scanning systems. As a next step towards our research goal, we would like to enhance the physics informed object pose estimation for complex object geometries and much more complex cluttered scenes. On the object data extraction front, we would like to extract additional cognitive information associated with each object using annotation. In conclusion, these two essential modules show promise for handling complex scenes and geometries, respectively, and their combination is essential for robot manipulation tasks in cluttered indoor scenes.

REFERENCES

[1] PR2 interactive manipulation. http://wiki.ros.org/pr2_interactive_manipulation.

[2] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva. Computing and rendering point set surfaces. *IEEE Transactions on Visualization and Computer Graphics*, 2003.

[3] Pierre Alliez, Stephane Tayeb, and Camille Wormser. 3D fast intersection and distance computation. In *CGAL User and Reference Manual*. CGAL Editorial Board, 4.7 edition, 2015.

[4] Marcus A Brubaker, David J Fleet, and Aaron Hertzmann. Physics-based person tracking using the anthropomorphic walker. *International Journal of Computer Vision*, 2010.

[5] Changhyun Choi and Henrik I Christensen. Rgb-d object tracking: A particle filter approach on gpu. In *IROS, 2013*.

[6] Matei Ciocarlie, Kaijen Hsiao, Edward Gil Jones, Sachin Chitta, Radu Bogdan Rusu, and Ioan A Şucan. Towards reliable grasping and manipulation in household environments. In *Experimental Robotics*, pages 241–252. Springer Berlin Heidelberg, 2014.

[7] A. Correa, M.R. Walter, L. Fletcher, J. Glass, S. Teller, and R. Davis. Multimodal interaction with an autonomous forklift. In *HRI, 2010*.

[8] Mehmet R Dogar, Kaijen Hsiao, Matei Ciocarlie, and Siddhartha Srinivasa. Physics-based grasp planning through clutter. 2012.

[9] William I. Grosky, Robert Zeleznik, Timothy Miller, Andries van Dam, Chuanjun Li, Dana Tenneson, Christopher Maloney, and Joseph J. LaViola. Applications and issues in pen-centric computing. *IEEE MultiMedia*, 15 (4):14–21, October 2008.

[10] W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1): 97–109, 1970.

[11] Zhaoyin Jia, Andrew C Gallagher, Ashutosh Saxena, and Tsuhan Chen. 3D reasoning from blocks to stability. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 2015.

[12] Dominik Joho, Gian Diego Tipaldi, Nikolas Engelhard, Cyrill Stachniss, and Wolfram Burgard. Nonparametric bayesian models for unsupervised scene analysis and reconstruction. *Robotics*, page 161, 2013.

[13] Zia Khan, Tucker Balch, and Frank Dellaert. An MCMC-based particle filter for tracking multiple interacting targets. In *ECCV 2004*.

[14] Ziyuan Liu, Dong Chen, Kai M Wurm, and Georg von Wichert. Table-top scene analysis using knowledge-supervised MCMC. *Robotics and Computer-Integrated Manufacturing*, 33:110–123, 2015.

[15] Chavdar Papazov, Sami Haddadin, Sven Parusel, Kai Krieger, and Darius Burschka. Rigid 3D geometry matching for grasping of known objects in cluttered scenes. *The International Journal of Robotics Research*, page 0278364911436019, 2012.

[16] R. B. Rusu, G. Bradski, R. Thibaux, and J. Hsu. Fast 3D recognition and pose using the viewpoint feature histogram. In *IROS, 2010*.

[17] Radu Bogdan Rusu, Zoltan Csaba Marton, Nico Blodow, Mihai Dolha, and Michael Beetz. Towards 3D point cloud based object maps for household environments. *Robotics and Autonomous Systems*, 56(11):927–941, 2008.

[18] D. Shah, J. Schneider, and M. Campbell. A robust sketch interface for natural robot control. In *IROS, 2010*, Oct .

[19] Ashutosh Singh, Jin Sha, Karthik S Narayan, Tudor Achim, and Pieter Abbeel. Bigbird: A large-scale 3D database of object instances. In *ICRA, 2014*.

[20] Marjorie Skubic, Derek Anderson, Samuel Blisard, Dennis Perzanowski, and Alan Schultz. Using a hand-drawn sketch to control a team of robots. *Autonomous Robots*, 2007.

[21] Shuran Song and Jianxiong Xiao. *Computer Vision – ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI*, chapter Sliding Shapes for 3D Object Detection in Depth Images. Springer International Publishing, 2014.

[22] Zhiqiang Sui, Odest Chadwicke Jenkins, and Karthik Desingh. Axiomatic particle filtering for goal-directed robotic manipulation. In *IROS, 2015*.

[23] Andreas ten Pas and Robert Platt. Localizing handle-like grasp affordances in 3d point clouds. In *International Symposium on Experimental Robotics*. Citeseer, 2014.

[24] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT press, 2005.

[25] Julien Valentin, Vibhav Vineet, Ming-Ming Cheng, David Kim, Jamie Shotton, Pushmeet Kohli, Matthias Niessner, Antonio Criminisi, Shahram Izadi, and Philip H.S. Torr. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics*, 2015.

[26] Marek Vondrak, Leonid Sigal, and Odest Chadwicke Jenkins. Dynamical simulation priors for human motion tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(1):52–65, 2013.

[27] Li Emma Zhang and Jeffrey C Trinkle. The application of particle filtering to grasping acquisition with visual occlusion and tactile sensing. In *ICRA, 2012*.