Algorithms and Systems for Manipulating Multiple Objects

Zherong Pan[15], Andy Zeng[2], Yunzhu Li[3], Jingjin Yu[4], and Kris Hauser[5]

***Abstract*—Robot manipulation of multiple objects is an important topic for applications including warehouse automation, service robots performing cleaning, and large-scale object sorting. Although problems can range in complexity from a few objects to large disordered piles, autonomy remains a significant technical challenge due to the high-dimensional joint configuration space of the robot and all objects, the complex dynamics of object interaction, and the ambiguity and occlusion caused by clutter. This paper surveys a broad range of classical and state-of-the-art research in multi-object manipulation and categorizes them along the dimensions of tasks, perception, predictive models, and decision-making algorithms. It also covers emerging trends and open problems faced in the ongoing effort to realize robust multi-object manipulation systems in practice.**

***Index Terms*—Manipulators, Manufacturing Automation, Robot Motion, Robot Vision Systems**

I. Introduction

In the last decade, robotic grasping of single objects has rapidly matured from lab demonstrations to industrial deployments [110, 139]. As a result, many robotics researchers have shifted their efforts toward studying how robots can manipulate many objects in coordination. In some scenarios, the objects' poses obscure one another or interfere with subsequent movements, so the robot must reason about the relationships between multiple objects [99, 140]. In others, the robot must reason about the stability of piles of objects [43, 135]. In yet other scenarios, a robot must move many objects at once by pushing or scooping [31, 126]. Such problems arise in a diverse range of application areas at factories, warehouses, stocking, retail, and homes, in which robots are expected to search in piles of objects, assemble or disassemble parts, pack containers, stock shelves, clean clutters, and manipulate granular materials.

Multi-object manipulation poses challenges for every stage of autonomous system development. First, multi-functional robots and end-effectors should be designed to implement various manipulation skills, such as grasping, pushing, and shifting [10, 104]. Second, robots must perceive the state of objects in the world, in which object identity and pose may be ambiguous due to severe occlusions and uncertainties [134]. Third, the dynamic behaviors of multiple interacting objects must be predicted under stochastic contact mechanics [151] and incomplete state information [86]. Fourth, a motion planner must search in the space of manipulation orderings, which has a high combinatorial complexity, as well as the joint state space of the robot and all objects, which has high dimensionality. Theoretical complexity bounds have been established for various tasks [52, 62, 106] and practitioners utilize simplifying assumptions to design tractable decision-making algorithms. Finally, frequent re-planning and closed-loop controllers are used to respond to perceptual uncertainty and to close the gap between predictive models and real-world behaviors [3].

Research on manipulation of multiple objects began decades ago in the motion planning community, and was explored both for theoretical interest as well as applications such as assembly planning. The computational intractability of general multi-object manipulation tasks and some assembly planning tasks were proven during this era [52, 62]. Further progress in motion planning led to practical algorithms for applications such as navigation among moving obstacles [17, 123] and rearrangement in logistics [101]. In the mid-2010s, progress in robot perception and manipulation led to a rapid growth in interest in robot rearrangement [68, 71], singulation [14], and decluttering [61, 109] that has continued to this day. In recent years, learning-based perception, predictive models, and manipulation actions, have fundamentally changed the design of robot systems [81]. Deep learning has made object recognition in clutter far more reliable and accessible, and end-to-end reinforcement learning has the potential to lift restrictive assumptions in classical sense-plan-act architectures. These trends have been adopted with great enthusiasm for multi-object manipulation [10, 54, 126]. It should also be noted that the robotics community has not just been a user of modern AI techniques, as multi-object rearrangement has also been identified as a benchmark for the embodied AI community [5].

It is an opportune time for the development of multi-object manipulation research to address some high-level questions that have been left unanswered despite decades of research. For example, has perception and modeling uncertainty been underappreciated in the multi-object planning community? At what granularity should multiple objects be represented, e.g., can we treat multiple objects as a single entity when they stay in contact? How should we model the complex stochastic contacts, and how much does the accuracy of a contact model affect the accuracy of decision-making? In some tasks, e.g., organizing a cluttered cabinet, the gap between heuristics and optimal solutions is substantial, but other tasks, e.g., singulation, can be solved relatively easily with heuristics. Does the optimality gap relate to the sample complexity of learning-based methods? With this survey, we hope not only to describe the breadth of research in this field, but also to inspire researchers to ponder these grand questions.

This survey focuses on algorithmic research related to multi-object manipulation. It does not cover mechanical design, because the vast majority of past works have focused on moving a single object at a time or pushing multiple objects, with only a few papers covering mechanical devices that move multiple objects simultaneously [98, 108]. The paper does not cover single-object grasping, and refers interested readers to Billard and Kragic [11] for a survey on recent trends in the

[1]Tencent America. zrpan@tencent.com
[2]Google Research. andyzeng@google.com
[3]Department of Computer Science, Massachusetts Institute of Technology. liyunzhu@mit.edu
[4]Department of Computer Science, Rutgers University. jingjin.yu@cs.rutgers.edu
[5]Department of Computer Science, University of Illinois at Urbana-Champaign. kkhauser@illinois.edu

manipulation of single objects, and Stüber *et al.* [125] for a survey on pushing manipulations. The paper is organized by manipulation tasks, perception methods, prediction methods, and decision-making algorithms. Section II describes a taxonomy of manipulation tasks. Section III describes perception components that process raw sensor inputs (e.g. RGBD images [61] or thermal images [111]) into object locations and/or identities. Section IV presents motion prediction methods, which estimate the change of objects' state due to a robot's action e.g., pushing, grasping, or poking. Section V discusses decision-making problems, which involve a motion (re-)planner that calculates sequences of manipulation actions, and a controller that compensates for disturbances. Finally, we discuss open problems in Section VI.

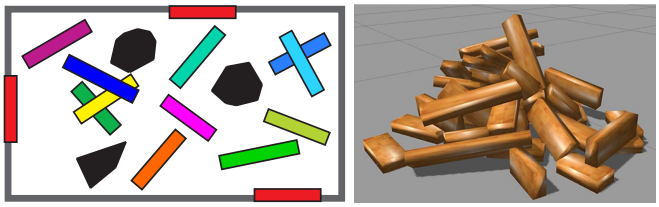
II. Multi-Object Manipulation Task Taxonomy

In general, a multi-object manipulation task is mathematically specified as attempting to bring the state of the *joint state space* of the robot and objects into a state that meets specific goal conditions. Following King and Srinivasa [68], we define the robot state space as $x^R \in \mathcal{C}^R$, the state space of each of m objects as $x^1(t) \in \mathcal{C}^1, \cdots, x^m(t) \in \mathcal{C}^m$, and the joint state space as the Cartesian product $\mathcal{C} = \mathcal{C}^R \times \mathcal{C}^1 \times \cdots \times \mathcal{C}^m$. We further denote the workspace as $\mathcal{X}$, the volume occupied by the robot's geometry as $R(x^R)$, and the volume occupied by an object as $O^i(x^i)$. The objective of a particular manipulation is to reach some goal subset $\mathcal{C}_{goal} \subset \mathcal{C}$ while keeping the state within a free space $\mathcal{C}_{free} \subset \mathcal{C}$. Different manipulation tasks are characterized by their action sets, goal conditions, feasibility conditions, observation spaces, or governing dynamic models of objects. Past literature has categorized tasks primarily in terms of differences in the goal set $\mathcal{C}_{goal} \in \mathcal{C}$, as described below.

- *Singulation:* A small extension of grasping in clutter, where multiple objects may need to be moved in order for a specified object to be grasped. The goal condition is InGrasp(x^R, x^1) where x^1 is, without loss of generality, the target object. As illustrated in Figure 3, singulation can be achieved by moving the target object away from the clutter [63] or moving other objects out of the way [78]. A simplifying assumption taken by several prior works [14, 63] is the target object being a certain distance away from all other objects. The distance-based definition provides a goal condition that is straightforward to test, but it may require moving objects farther than necessary. More recent research tends to consider the problem solved when the final grasp is acquired [99].
- *Navigation:* A mobile robot travels through a cluttered environment to reach a target region, with a set of movable objects blocking its path. The goal is specified only for the robot but not for objects, i.e., $x^R \in \mathcal{C}^R_{goal}$ with $\mathcal{C}^R_{goal}$ the target region for the robot. Navigation has been studied as a discrete search problem [29] or a Sokoban game, where both the robot and the objects move along edges of an axis-aligned grid. It can also be formulated as a continuous motion planning problem [123] as illustrated in Figure 1.
- *Declutter:* Given a region $\mathcal{X}_{clear}$ to clear, a set of objects are moved away from a target region [128], i.e., the goal is $O^i(x^i) \cap \mathcal{X}_{clear} = \varnothing$ for all $i = 1, ..., m$. The robot may have payload limits preventing all objects from being moved together, and/or objects may be entangled, making it difficult to find a feasible order of removal. To avoid disturbing a pile of objects, which would require re-sensing and re-planning, constraints can be specified where objects must remain quasi-statically force-balanced during manipulation [130] (Figure 2). In cases with tightly entangled non-convex objects, finding a feasible object motion may also be a challenge. Decluttering is highly related to the disassembly problem [77], a classical problem in operations research and motion planning.
- *Rearrangement:* A set of objects are moved to take a set of specified goal positions (Figure 4), i.e., $x^i \in C^i_{goal}$, $i = 1, ..., m$. Rearrangement tasks can be *labeled* where each object has a single specified goal configuration, e.g. in [71], or *unlabeled* where object-goal correspondences are arbitrary, in [3, 5, 37, 146] for example. In both cases, the objects' manipulation order and manipulation paths can be considered as two sub-problems, where the decision space of the manipulation order is discrete and that of the manipulation paths is continuous. Such discrete-continuous decomposition allows efficient planning algorithms to be developed [71].
- *Packing:* As a reverse of declutter, a set of objects are moved into a goal region, i.e., $O^i(x^i) \subset X_{pack}$, $i = 1, ..., m$, but the target configuration of each object in the region is unspecified. Essentially, packing is a reverse of declutter. There are two qualitative variants of packing: loose-packing and dense-packing. In loose-packing, the goal region has a much larger volume than the to-be-packed objects and so

(a) (b)

Fig. 1. Stilman and Kuffner [123] considered a continuous navigation task where the robot can move one object at a time in a cluttered environment in order to move from start to goal position. (a): side view; (b): bird view.



(a) (b)

Fig. 2. (a): A 2D illustration of clutter removal problems [128], where objects can overlap to form constraints on the order of removal. (b): Temtsin and Degani [130] compared various heuristic strategies to remove a clutter of 3D bricks without disturbing the pile.

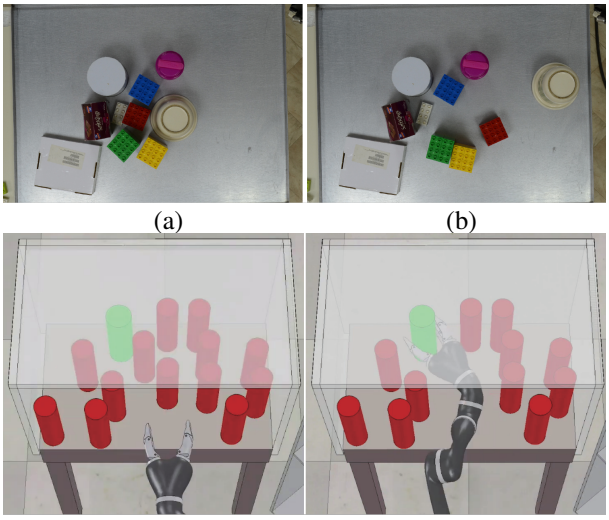


Fig. 3. (a): Kiatos and Malassiotis [63] singulates a cylindrical target object by moving it away from the clutter (b). (c): Lee *et al.* [78] singulates the green target object by moving other objects away from the swept volume of the robot (d).

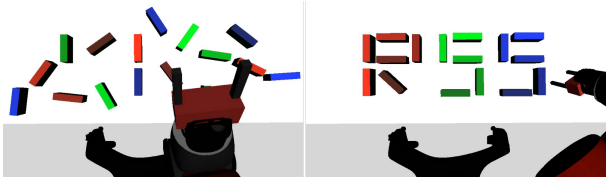


Fig. 4. In the rearrangement task, Krontiris and Bekris [71] moved a set of unorganized bricks (a) to form letters "RSS" (b).

the focus is on transporting objects to the region. Therefore, prior works [90, 147] use sequential single-object manipulation in an arbitrary order with little dependence on already-packed items. In dense-packing, the goal region's volume is approximately the same as the sum of the objects' volumes [135], and hence objects' configurations must be planned to improve the packing density (Figure 5).

- *Placing:* A new object is placed into a goal region, i.e., $x^1 \in \mathcal{C}_{place}$ with x^1 the target object, which may involve rearranging other objects to make room [20] (Figure 6). Placing also has two qualitative variants: loose-placing and dense-placing. In loose-placing [7, 66, 67, 68], the density of objects is low and placing can be achieved by moving other objects out of the target region. In dense-placing [20], the density of objects is high so that the obstructing objects may interfere with other objects, and so the order and trajectories of objects' movements must be planned carefully to increase the success rate. Placing is closely related to packing, which can be achieved by repeatedly placing new objects, as done in all existing works of object packing [90, 135, 147].
- *Sorting:* A set of objects, divided into classes (e.g., color, identity, or type), should be geometrically separated (Figure 7). There are two variants of sorting: sorting-by-clustering and sorting-by-packing. In sorting-by-clustering [120], the goal is to minimize the intra-class distances and maximize the inter-class distances, similar to a multi-class extension of singulation. If we define l_i as the label of the i^{th} object, then we require the intra-class distance $\max_{l_j=l_i} d(x_i, x_j) \leq \min_{l_k \neq l_i} d(x_i, x_k) \ \forall i$. In sorting-by-packing [48], the goal is to move all objects of the same class to a designated goal region, similar to a multi-class version of packing. In other words, the goal set is $O^i(x^i) \subset X_{pack,l}$ for $l_i = l$.

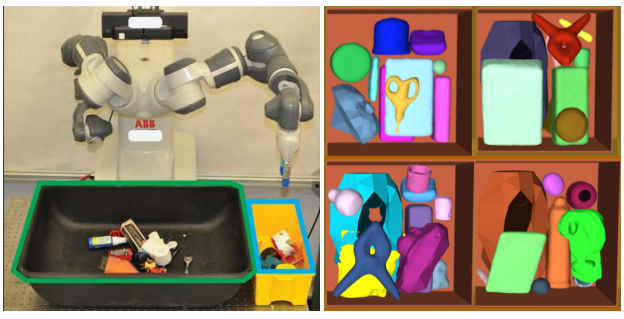


Fig. 5. (a): Mahler and Goldberg [90] considered loose-packing of unsorted objects with a two-armed robot picking from a bin. (b): Wang and Hauser [135] considered dense-packing, placing objects with varied shape tightly into shipping boxes.

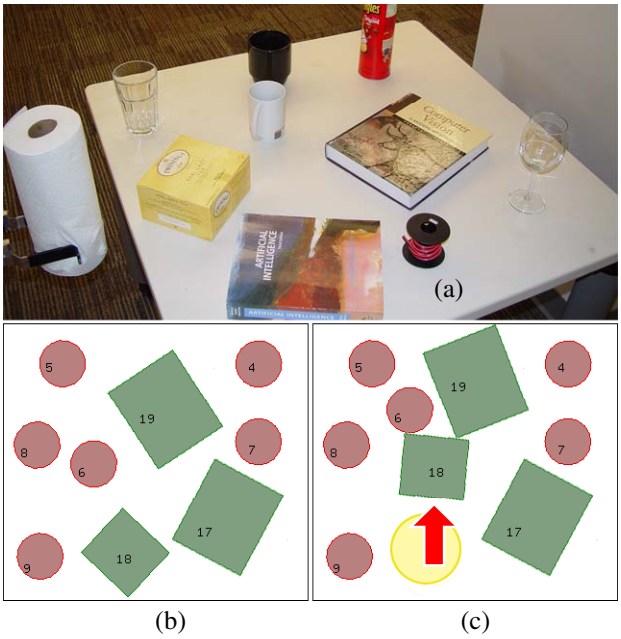


Fig. 6. In order to place the paper towel on the table (a), the yellow box (id=18) needs to be pushed away to make room. Cosgun *et al.* [20] proposed a planner to search for the order of pushes (bc).

Table I lists several prior works categorized by their manipulation tasks. Some works combine multiple tasks to solve problems at a higher-level, so we put them into more than one category. For example, a declutter task is followed by a loose-packing task in [129] to place the dumped objects. In [66, 67, 68], a navigation task is implicitly involved in a placing task so that the robot can clear the path for the to-be-packed object to reach a goal region. In [31, 126], the goal is to move a cluster of small objects to reach a target configuration, which can be used for both declutter and sorting-by-clustering tasks.

There is a general commonality in multi-object manipulation that most tasks can be modeled as a task-and-motion

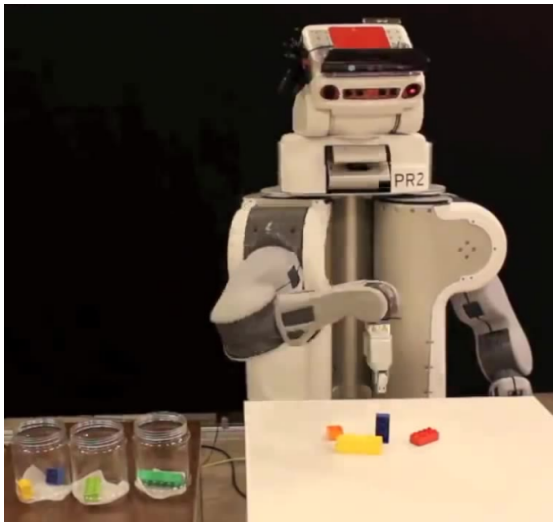


Fig. 7. Gupta and Sukhatme [44] sorted bricks into bottles (at left) by color.

TABLE I
PRIOR WORKS CATEGORIZED ACCORDING TO THEIR TASKS.

Manipulation Task	References
Singulation	[10, 14, 26, 30, 54, 60, 63, 73, 78, 99, 145]
Navigation	[17, 66, 67, 68, 95, 123, 124]
Declutter	[12, 16, 31, 34, 61, 100, 109, 126, 128, 129, 130]
Rearrangement	[3, 70, 71, 101, 114, 115, 117, 144]
Packing	[90, 135, 136]
Placing	[7, 20, 66, 67, 68]
Sorting-by-Packing	[44, 48, 129]
Sorting-by-Clustering	[31, 55, 93, 120, 126, 132, 133]

planning (TMP) problem encompassing discrete tasks, i.e., choices of an object to manipulate, and continuous motions, i.e., choices of a target location and a manipulation movement. However, general models are unlikely to be computationally efficient. Instead, task-specific algorithms and heuristics are preferred for their speed and solution quality. Some of the key aspects used to categorize tasks can be found in Table II.

- *Goal Specification (Robot / One Object / All Objects):* The singulation task requires the robot to acquire a grasp of the target object, while the navigation task only specifies the robot's goal configuration. Placing is only concerned with a single object. The other tasks typically assume to specify all the objects' goal configurations.
- *Object-Object Contact:* Some tasks require the robot to interact with multiple objects in contact simultaneously, e.g., piles of objects in decluttering or packing. Some prior works in singulation have used pushing to sweep multiple objects away from the target object. Without multiple objects in contact, pick and place manipulation is essentially a geometric problem, but otherwise prediction of the physics of object-object contact becomes more challenging.
- *Object Labeling (Labeled / Unlabeled / Classified):* The navigation, declutter, singulation, packing, and placing tasks do not specify goals for individual objects, making object identification less necessary to complete the task. The rearrangement task can have labeled or unlabeled variants. Objects in a sorting task can be labeled, unlabeled, or classified, i.e., partially labeled, where some object or group of objects must be moved to different, specified regions.
- *Prioritized (Yes / No):* In many settings, the robot can focus on the pose of a prioritized object, while other objects are auxiliary and treated as obstacles. For example, the navigation task is divided into several stages, where the robot focuses on the object blocking the way during each stage. Prioritization can be utilized to simplify algorithm design, e.g. using greedy heuristics or by eliminating the need for identification and prediction of auxiliary objects. However, even in such tasks there exist challenging examples in which the robot must reason several steps ahead about the manipulation of auxiliary objects (e.g., non-monotone navigation problems).

TABLE II
DISTINGUISHING TASKS ALONG FOUR CRITERIA AS DESCRIBED IN TEXT. FOR THE GOAL SPECIFICATION COLUMN: R=ROBOT, O=ONE OBJECT, A=ALL OBJECTS. FOR THE LABEL COLUMN: L=LABELED, U=UNLABELED, C=CLASSIFIED.

Manipulation Task	Goal Spec.	O-O Contact	Label	Prioritized
Singulation	RO	Some	U	Yes
Navigation	R	No	U	Yes
Declutter	A	Yes	U	No
Rearrangement	A	No	LU	No
Packing	A	Yes	U	No
Placing	O	No	U	Yes
Sorting	A	No	LUC	No

III. PERCEPTION ALGORITHMS

Perception — estimating the states of objects and their physical attributes from sensor data — is needed before initiating manipulation as well as during manipulation to correct for unexpected events. Accurate perception is especially difficult in cluttered scenes due to occlusions. Moreover, it is difficult to obtain an exact model of object geometries, inertial characteristics, friction characteristics, and interaction behaviors in unstructured settings. As a result, perception for multi-object manipulation remains a significant research challenge.

We note that research on perception for multi-object manipulation has garnered interest much more recently than manipulation planning research. Early work such as [59] in this area assumes occlusion-free, top-down views of known and/or visually-distinctive, isolated objects. Only recently have researchers begun to address perception under occlusion [27, 134], and to handle uncertainty and ambiguity [34, 118]. Sophisticated hypothesis generation algorithms that consider sensor data fit [34], stability [135], and other criteria have been used to determine likely hypothetical object arrangements. There has also been significant recent growth in the use of deep learning techniques that identify objects and their attributes from camera images [129]. A related body of work such as [30, 145] addresses *affordance detection*, which is the problem

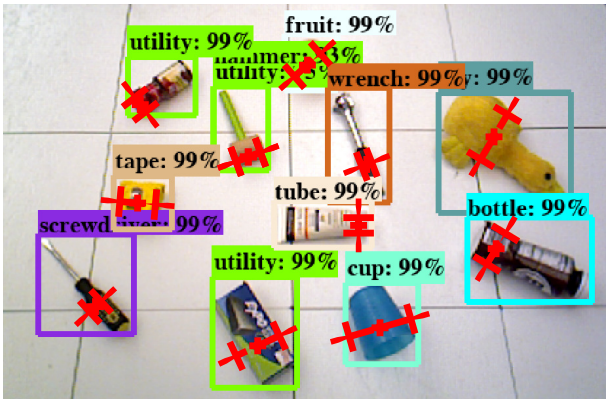


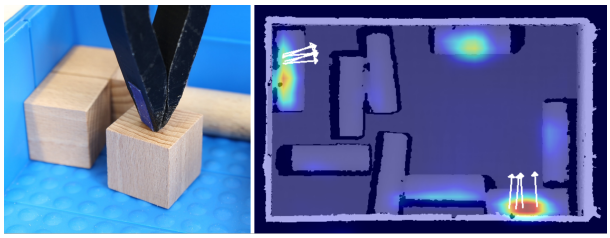
Fig. 8. Tanwani *et al.* [129] used off-the-shelf deep network architecture [88] to recognize and localize well-separated objects, for each of which a grasping action is proposed.

of predicting action choices available for manipulation (e.g., grasp candidates) directly from visual data. This area is gaining attention due to the use of deep learning techniques that predict high-quality actions or policies directly from a camera image.

In Table III, we summarize representative perception approaches according to the problem, algorithm, and scenarios addressed.

- *Problem (Recognition / Segmentation / Affordance Detection / Pose Estimation / Feature Extraction / Tracking):* Object recognition aims at predicting regions of an image (such as a bounding box) in which an object is located (Figure 8). Segmentation is a fine-grained variant of recognition where the pixels of an image corresponding to an object are identified (Figure 9). The goal of pose estimation is to produce 2D/3D object poses for reference models of known objects, often in addition to object recognition information [73]. Affordance prediction provides information in the form of an action selection policy [120] or predictions of success rates of a manipulation operator [30, 60, 144, 145] (Figure 10). Feature extraction summarizes the state of one or more objects into a feature vector, which is used by downstream machine learning algorithms. Unlike all other methods that estimate objects' states at temporally discrete time instances, tracking algorithms maintain object poses across multiple frames [25]. They run at real-time rates during manipulation. Tracking can improve robustness to occlusions and uncertainties by using temporal coherence (Figure 11).
- *Algorithm (Image Processing / (Classical) Machine Learning / Deep Learning):* Low-level image processing operators can detect simply connected components of masks or identify image features such as contours of a sand pile [18]. A widely-used high-level operator extracts keypoints on rigid objects, typically for recognition, estimation, and tracking problems [14]. Keypoints may be passed to RANSAC, ICP, or pose graph optimization methods. Other high-level operators can summarize semantic information, e.g., whether a pixel patch is a single object or not [15]. In the pre-deep-learning era, machine learning relies on manually designed features to solve classification and segmentation problems using algorithms such as graph-cut and clustering.

Fig. 9. Boularias *et al.* [12] performed instance segmentation (segments the pixels belonging to each object). For each instance, a set of push or grasp actions are sampled.



(a) (b)

Fig. 10. Berscheid *et al.* [10] considered the object singulation tasks by overhead shifting (a). They trained a neural network to predict the affordance of shifting actions from all possible positions (b).

For example, Katz *et al.* [61] clustered object patches into flat facets and further classified the affordance of each facet using a support vector machine. By comparison, deep neural networks work end-to-end by learning features from data, whose outputs can represent object positions [129] or affordances [144]. Using convolutional pyramids, deep networks can efficiently process an entire image, which is favorable for multiple objects with severe occlusions [90].

- *Occlusion (None / 3D):* Many works solve tabletop problems assuming objects do not overlap from the perspective of an overhead camera. General 3D occlusions are more challenging and require reasoning about hidden objects. For

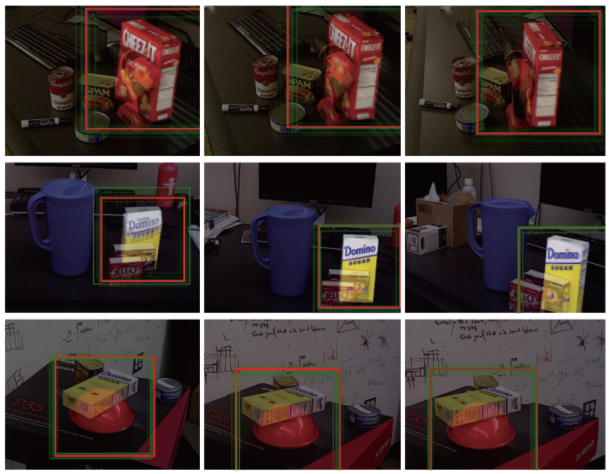


Fig. 11. Combining a particle filter and a deep autoencoder to learn a distribution over 3D rotations, Deng *et al.* [25] proposed a 6D object pose tracking system that is reasonably stable under severe occlusions and changing appearances.

TABLE III
PRIOR WORKS CATEGORIZED ACCORDING TO THEIR PERCEPTION COMPONENT. FOR THE PROB. COLUMN: R=RECOGNITION, S=SEGMENTATION, A=AFFORDANCE DETECTION, P=POSE ESTIMATION, F=FEATURE EXTRACTION, T=TRACKING. ALG. COLUMN: IP=IMAGE PROCESSING, ML=MACHINE LEARNING (CLASSICAL), DL=DEEP LEARNING. OCC. COLUMN: PL=PLANAR OCCLUSION, 3D=3D OCCLUSION. PRI. COLUMN: S=SHAPE PRIOR, A=APPEARANCE PRIOR, P=PHYSICS PRIOR.

Task	Ref	Prob.	Alg.	Occ.	Pri.
Declutter	[61]	S,A	IP,ML	3D	S,A
Declutter	[12]	S	IP,ML	3D	S,A
Singulation	[14]	S,F	IP,ML	3D	S,A,P
Singulation	[73]	P	IP	3D	S,P
Singulation	[145]	A	DL	PL	S,A,P
Singulation	[10]	A	DL	PL	S
Singulation	[63]	F	IP	PL	S
Singulation	[99]	R,P	DL	3D	A
Singulation	[30]	A	DL	3D	S,A,P
Singulation	[54]	A	DL	PL	S,A,P
Rearrangement	[144]	P,A	DL	PL	S,A
Rearrangement	[115, 117]	P	IP	PL	S,A
Sorting-by-Packing/ Declutter	[129]	R,P	DL	3D	A
Sorting-by-Clustering/ Declutter	[31]	S	IP	PL	A
Sorting-by-Clustering	[55]	R,P	IP	PL	S,A
Sorting-by-Clustering	[120]	A	DL	PL	S
Sorting-by-Clustering/ Declutter	[126]	S	DL	PL	A

example, tilted cameras can see a large portion of at least one face of each object [61]. Object search tasks [27, 140] use active perception to location entirely occluded targets.

- *Prior Knowledge (Shape / Appearance / Physics):* Perception problems are oftentimes ill-posed and under-determined due to occlusions and partial observations, so prior knowledge is used to yield better estimates. Shape priors restrict objects' geometries to have certain qualities, e.g., rectangular, planar, convex, or use the knowledge that the objects in the scene come from known 2D or 3D models. As an example, Katz *et al.* [61] used the prior that objects have large planar facets. Appearance priors make assumptions about objects' materials, lighting conditions, or colors. Physics priors model objects' equation of motion to be either rigid, elastic- or plastic-deformable. Training data constructed using physics simulators such as [145] is also considered a form of physics prior.

A. Trends and Open Problems

Overall, perception technology for multi-object manipulation is maturing quickly, but the vast majority of experiments, including all the works listed in Table III, have still taken place only in lab environments. Results from lab setups are likely to generalize fairly well to industrial applications where the robot's environment can be similarly controlled and the perception algorithms specialized to the objects under consideration. In automatic warehouses and fulfillment centers, it is likely for a robot to encounter out-of-sample objects and uncertain environmental conditions [19]. An important direction for future work is to investigate how perception generalizes to uncontrolled lab settings.

Task-dependent perception system design: We observe an imbalanced distribution of manipulation tasks addressed by perception algorithms, with most work in Table III solving singulation tasks. This is presumably because grasping finds a lot of application scenarios and singulation is a way to improve grasping robustness. The packing and placing tasks usually require perception under severe occlusions due to the density of objects in contact in a confined space, which may yield fruitful directions for future perception research. We note that some tasks, such as navigation and decluttering, are even difficult to set up in a lab because they require large spaces. Moreover, due to the size of the area covered, they cannot be addressed using a single camera viewpoint, and instead may require map integration over multiple frames. In addition, we are not aware of any work that uses object tracking in the manipulation pipeline, until very recently Morgan *et al.* [97] showed that tracking the 6D rigid pose can improve the manipulation accuracy for peg-in-hole insertion problems, which is single-object manipulation. All existing methods listed in Table III choose to discretely estimate objects' states before and after actions, which is justified by making quasistatic assumptions that objects move reasonably slowly, and moreover that objects are not entirely hidden after each action. On the other hand, it has been shown in [35, 91] that tracking leads to better robustness in handling partial and presumably full occlusion, although these works are limited to a single object. A better understanding of occluded space in the presence of clutter is another important issue to investigate for future research. Although many tasks, such as planar problems or sorting-by-packing, can be solved purely by addressing the visible objects [134], other tasks such as placing require reasoning about empty space behind occlusions. Tasks, such as object search within piles, or decluttering / packing with stack stability estimation, require reasoning about possible locations of hidden objects. In these cases, perception algorithms can use free-space to ensure that object pose estimates are non-interfering, and they can use physical reasoning to understand how objects support one another under gravity [94, 96]. However, prior work is still at an early stage and this remains an open problem.

System integration for learning-based perception: Recent trends show the popularity of deep neural networks for perception, which aligns with general trends in the computer vision field. In manipulation, deep networks have enabled researchers to investigate learning as a way to combine different system components (e.g., affordance, prediction, and planning) more closely. For example, image prediction methods combine perception and prediction by modeling transitions in the image-feature space [31, 126]. End-to-end reinforcement learning in [145] couples perception, affordance models, and motion planning. With an additional convenience in system design comes increased difficulties in data generation, generalization, system diagnosis, and analysis.

Object tracking: The vast majority of perception models listed in Table III use vision and/or depth as the only sensor modalities. We are aware of one exception [34], where the

forces exerted by the robot is formulated as a part of the state space of the learned predictive model. Recent works have shown that other modalities, such as force and tactile information, can compensate for vision and depth to resolve ambiguities [141] or handle contact-rich manipulation problems [28], but these methods have not been exploited in the multi-object manipulation literature yet.

Uncertainty quantification and modeling: A final note is that planning and prediction modules typically make the assumption that perception is accurate. However, in the real world, perception always suffers from errors. An error-tolerant solution requires a co-design of perception and planning modules. A successful example is [35] where latent state features and feature-space controllers are simultaneously optimized. Downstream effects of these errors are also unclear. Perception models should encode object states for downstream prediction models to reliable estimate future states even under sensing error and uncertainties. An inspiring recent work [38] encodes an object's state as a set of oriented keypoints, which can interface with several downstream control algorithms such as imitation learning and iterative model predictive control.

IV. Predictive Models

Predictive models are indispensable for some decision-making algorithms such as model predictive control to estimate the results of manipulation operations. Although predictive models have garnered some interest in the single-object pushing setting [118, 150], the multi-object manipulation setting is more complicated because objects often interact via contacts. Not only does this make physics modeling more complex, but it also reduces accuracy of predictions due to inherent uncertainties in inter-object contact models as pointed out in [125].

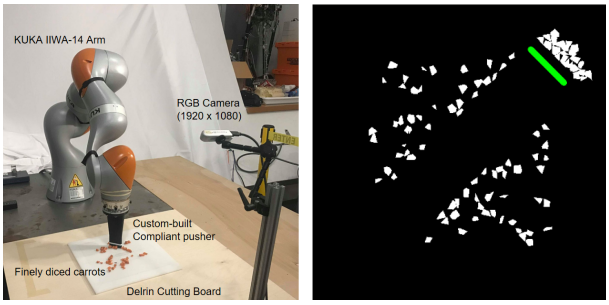
We can denote the robot's predictive model as $\dot{x}^R(t) = f(x^R(t), u(t))$, where $u(t)$ is the control signal, and the corresponding objects' predictive models are $\dot{x}^i(t) = f(x^i(t), u(t))$ [68]. Usually, only a single item is moved at once while other objects are considered as obstacles, so $\dot{x}^i(t) = 0$ for non-manipulated objects. If multiple objects are moved as once, the predictive models for the robot and objects are typically coupled via nonholonomic contact constraints and jointly expressed as $\dot{x}_{R,S}(t) = f(x_{R,S}(t), u(t))$ where $S \subseteq \{1, \ldots, m\}$ denotes the subset of moved objects and $x_{R,S}$ denotes the joint state of the robot and moved objects.

The granularity of prediction is an important characteristic distinguishing past researches. For tasks involving pick-and-place actions on flat surfaces, prior researches [47, 71] ignore the prediction problem entirely by assuming each action will be successfully completed as specified. More sophisticated models can predict grasp and placement stability [100, 135], i.e., the conditions under which high-level actions are successful. For example, when deciding how to extract objects from piles or placing them on piles, the robot must predict the stability of the pile [100]. On the other hand, low-level predictions of object behaviors are necessary for problems involving multi-object contacts, e.g., pushing [55, 120]. Here, predictions can be challenging due to the prevalence of uncertainty in contact states, object inertial parameters, and friction coefficients.

A. Rigid Body Predictive Model

We refer readers to [125] for a detailed discussion and comparison of different predictive models for rigid bodies and we follow their taxonomy as much as possible. We classify predictive models using 4 criteria:

- *Kinematic / Quasistatic / Dynamic:* A kinematic model ignores all forces on objects and only respect geometric constraints such as collision constraints. Kinematic models achieve high-fidelity when robots can immobilize objects by grasping or fixtures and object motions are not affected by contact forces. A quasistatic model considers all forces but ignores inertial and damping forces and assumes that objects always move under external force equilibrium. This model can make accurate predictions when objects are moved slowly and accelerations are very small and quickly damped to zero. A dynamic model considers all forces including inertial forces. If the kinematic information of the object is $q^i(t)$, then we have $x^i(t) \triangleq q^i(t)$ for a kinematic or quasistatic model, and $x(t) \triangleq \left(q^i(t)\, \dot{q}^i(t)\right)$ for a dynamic model.
- *Analytic / Learning-Based:* An analytic model is derived from classical mechanics, although some parameters might be predicted from data [58]. A learning-based model mimics the motion of objects solely from observations (e.g., videos or object features). Analytic models of well-understood phenomena can achieve reasonable overall fidelity and do not require costly data collection and learning steps. On the other hand, learning can achieve higher fidelity, when subtle interactions between multiple objects cannot be captured by analytic methods. On the downside, the fidelity of learning-based models can suffer due to over-fitting or insufficient data coverage.
- *Planar / 3D:* A planar model assumes that movement / forces along gravitational directions and horizontal directions are separable, and so that force balance along the gravitational direction may be ignored. A 3D model considers coupled movements and forces along all directions. The use of planar models requires object relationship assumptions, e.g., objects are not on top of each other and objects are moved slowly enough so they will not topple. With these assumptions, planar models can reduce the dimension of an object's configuration space by half (SE(3) $\rightarrow$ SE(2)) and significantly boost efficiency.
- *Single-Object / Multiple-Object / Eulerian:* A single-object model assumes that objects can be manipulated sequentially and inter-object interference is ignored. This assumption is only valid in less cluttered scenarios, e.g., when objects are well separated. A multi-object model considers possible inter-object interference. Although such models are more general, it is more challenging to handle multi-object contacts, which typically reduce both efficiency and fidelity. In contrast to standard object-based (Lagrangian) discretizations, Eulerian models use a spatial discretization, where the representation of a scene is maintained as an image or volumetric grid. We refer readers to [137] for a more detailed discussion. Lagrangian models are generally more popular due to their high fidelity, because Eulerian models



(a) (b)

Fig. 12. Suh and Tedrake [126] used Eulerian models to predict the motion of small grains of carrots under pushing (a). The grains are discretized in images (b) and motions after the push are predicted via linear models.

lose accuracy by blurring the boundaries between objects. However, Eulerian models are gaining popularity due to the use of deep networks to make predictions directly in image space, and can be computationally more efficient when there are many objects or when object-object interactions are difficult to model. A recent work utilizing these features is illustrated in Figure 12.

In Table IV, we categorize predictive models used in prior works according to granularity of prediction as measured by the four criteria above. Typically, a dynamic model is more fine-grained by predicting the change of velocity as compared with quasistatic and kinematic models. A learning-based model such as [119] can learn subtle details such as anisotropic, material-dependent frictions that are more difficult to model via analytic methods. Some works, e.g., [60, 145], use a learning-based or learning-assisted planner, where different prediction models are used for data generation in offline training and online planning, and we use two different rows to label these two stages.

TABLE IV
PRIOR WORKS CATEGORIZED ACCORDING TO THE TYPE OF PREDICTIVE MODEL: K=KINEMATIC, Q=QUASISTATIC, D=DYNAMIC, A=ANALYTIC, L=LEARNING-BASED, P=PLANAR, S=SINGLE-OBJECT, M=MULTI-OBJECT, E=EULERIAN.

Task	Ref	K/Q/D	A/L	P/3D	SO/MO/E
Navigation	[17]	Q	A	P	MO
Navigation	[123, 124]	D	A	P	SO
Navigation	[95]	K	A	P	SO
Navigation/ Placing	[66, 67, 68]	Q	A	P	M
Declutter	[109]	D	A	3D	M
Declutter	[130]	D	A	3D	M
Declutter	[16]	D	A	3D	M
Declutter	[34] (Train)	Q	L	3D	S
Declutter	[128]	K	A	3D	S
Singulation	[26]	D	A	3D	M
Singulation	[60, 145] (Train)	D	A	3D	M
Singulation	[63] (Train)	D	A	3D	M
Singulation	[78, 99]	K	A	P	S
Singulation	[54] (Train)	D	A	3D	M
Rearrangement	[101]	K	A	P	S
Rearrangement	[70, 71, 114]	K	A	3D	S
Rearrangement	[3] (Train)	D	A	P	M
Rearrangement	[3] (Test)	Q	L,A	P	M
Rearrangement	[117]	D	A	P	M
Rearrangement	[82]	D	L	P	M
Rearrangement	[74]	K	L	P	E
Dense-Packing	[135, 136]	Q	A	3D	M
Placing	[20]	D	A	P	M
Placing	[7]	D	A	P	M
Sorting-by-Packing	[44]	K	A	3D	S
Sorting-by-Packing	[48]	K	A	3D	S
Sorting-by-Packing/ Declutter	[129]	D	A	3D	SO
Sorting-by-Clustering	[93]	K	A	P	S
Sorting-by-Clustering	[132, 133]	D	A	P	M
Sorting-by-Clustering/ Declutter	[31]	K	L	P	E
Sorting-by-Clustering	[55, 120]	D	A	P	M
Sorting-by-Clustering/ Declutter	[126]	K	L	P	E

B. Contact Modeling

The classical contact models [122] use an Lagrangian representation for each rigid body, i.e., the 6D configuration of each rigid body is predicted. Contacts between a pair of rigid bodies are assumed to happen on a discrete set of points. For each contact point, the relative velocity of two bodies and their interaction forces are modeled as two sets of complementary conditions along the normal and tangential directions, respectively. Variants of such contact models have been implemented in various modern rigid body simulation softwares and succeeded in predicting plausible motions involving many objects [33].

Contact-rich physics simulation is extensively used offline for validation and training learned multi-object manipulation systems. Cosgun *et al.* [20] simulate pushes to detect contacts between objects, which then is used to plan an order of pushing actions. They reset the simulation immediately after contacts are detected, so no objects are pushed indirectly in the final motion plan. [117] learn a probabilistic transition model for pushing by generating simulated pushes with a rigid body simulator with mass and contact parameter uncertainty. Reinforcement learning is also performed commonly within simulated environments [12, 16] involving many contacting rigid bodies.

On the other hand, only a few prior works such as [55, 117] choose to perform online physics simulations within the control loop. Simulation-in-the-loop control relies on highly accurate and efficient predictions, which is still difficult to achieve with off-the-shelf simulators. This is partly because the solutions of Coulomb friction problems is non-unique in general [45], and moreover depends on contact points, normals and friction coefficients that are often noisy or unobservable as pointed out in [76, 118]. In particular, the dynamics of pushing on a flat surface is a topic of significant interest because the distribution of contact pressure and friction over the support surface is unknown and indeterminate, and the overall movement of an object when pushed is difficult to predict a priori [41, 118, 150]. Instead, the set of possible resistant planar forces and torques on the object is modeled by a limit surface [41] which can be learned from data [150]. Simplified planar contact models exhibit desirable properties

such as differential flatness [149] and can be exploited to accelerate motion planning [22, 149], although these results are limited to a single object. Probabilistic pushing models may also be learned [118]. Finally, we note that accurate contact modeling can be omitted in some cases. For example, the declutter tasks considered in [31, 126] require predicting the movement of a large cluster of tiny objects (e.g. diced vegetables), which exhibit predictable gross motions even without accurate contact modeling.

The recent research on cluttered objects and pile manipulations [31, 61, 126] pose an ever increasing challenge on the efficacy of simulators. The memory and computational cost of conventional Lagrangian contact models [122] grow superlinearly with the number of rigid bodies, limiting its scalability to hundreds of objects. Therefore, recent works [31, 126] propose to use Eulerian representations. This method models the gross motion of objects as a material flow, with a complexity independent of the number of objects. A large body of research in computational physics such as [9, 80] has formulated contact models under Eulerian representations, but they have not been exploited in manipulation systems yet.

C. Trends and Open Problems

Task-specific simulation benchmarks: The capabilities of rigid body simulators have been exploited to their limits, and the last several years have not witnessed major new features introduced into these simulators. However, we are still lacking effective metrics, datasets, and testbeds for comparing the accuracy of predictive models in the object manipulation context. Our comparison in Table IV is qualitative and based on granularity. The two other crucial and quantitative metrics are *efficiency* (the amount computational cost required to make a prediction) and *fidelity* (the discrepancy between a predicted result and the result in real-world). In a relatively recent work, Erez *et al.* [33] compared several rigid body simulators in terms of efficiency, while their benchmarks are passive simulations of high-speed colliding objects. In a manipulation context, however, objects are undergoing nearly quasistatic manipulation actions such as pushing and grasping. It also helps to benchmark simulators used by different decision-making algorithms, some of which are designed to tolerate high simulation bias and do not require accurate predictions.

Uncertainty quantification and modeling: One major omission in existing methods is to quantify the uncertainty of predictions, which could help planners choose high-confidence actions or feedback strategies. In pushing manipulation, for example, the contact pressure and friction distribution over the support surface is unobservable but affects the overall movement of the object. Representing uncertainty is particularly challenging when contact is involved. Belief propagation with simple (e.g., Gaussian) distributions fails to capture the multi-modal posterior beliefs when contact is made vs. not-made during a motion [46]. Particle-based approaches [56, 79], i.e., Monte-Carlo sampling of simulation initial states and parameters, have been employed for this purpose, but these methods are often too computationally expensive for use in planning. Perhaps surprisingly, uncertainty is not directly correlated with the number of objects in contact. If objects' characteristic scale is much smaller than the end-effector, the gross behavior of dozens or thousands of objects in contact may exhibit strong uniformity, such as following the movement of a broom when cleaning dust or small particles [31, 126].

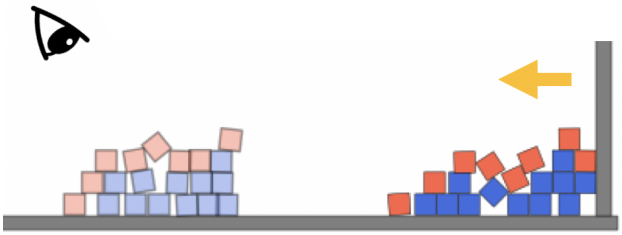


Fig. 13. Li *et al.* [82] proposed a graph neural network with multi-round message passing for learning complex object interactions. This architecture can take partial observations, e.g., when the camera only sees the red boxes on top of the pile. The learned neural dynamics model is inherently differentiable. We can extract the gradient using off-the-shelf deep learning packages [105] and solve the predictive control problems via the shooting method. In this example, the pile of boxes are supposed to be pushed to the target configuration on the left.

Learning-based prediction models under incomplete observations: Recently, semi- and non-parametric, structured, learning-based models [6, 75, 82, 138] have emerged as a promising method to acquire prediction capabilities without painstaking discretization of complex governing equations. These methods do not encode any physical rules, but model physical constraints between objects as edges of a graph and train random forests [75] or neural networks [82] to mimic behaviors of a constraint solver. Complex and realistic behaviors of rigid, fluid, and visco-elastic objects have been reproduced. Although their computational efficacy is not significantly higher than analytical models, learning-based models can make predictions under partial observations. This is achieved by mapping the incomplete observations to a latent state space and training the state-estimator as well as the latent transitional model in a joint manner. Conceptually, an observation function maps the state to a latent space, $h(t) = o(x(t))$, and the latent transition model, $\dot{h}(t) = f(h(t), u(t), \theta)$, is optimized by tuning the learnable parameters θ to mimic the observed dynamics behavior of $x(t)$. As compared with analytical models that require complete state information, learning-based models are favorable in multi-object settings where occlusions are ubiquitous and contact mechanics are stochastic. Some prior works, e.g., [82], have even used learning-based models to perform predictive controls as illustrated in Figure 13. However, these models have not been widely used to solve any high-level planning tasks so far. We are only aware of two recent works [34, 54] that use learned models to predict the results of push and grasp actions in solving clutter removal tasks.

V. Decision-Making Algorithms

Generating the robot's behavior involves a motion planner and a controller, where the planner determines the types and parameters of manipulation operators and the controller realizes them on the physical platform. Although manipulation

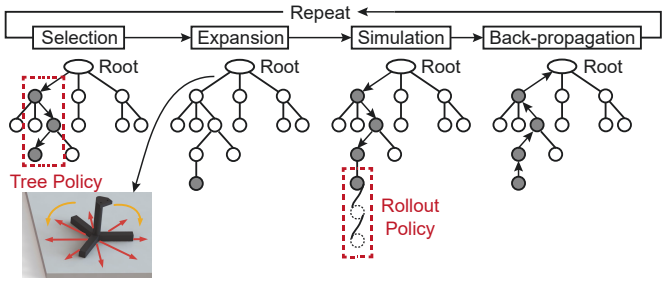


Fig. 14. We illustrate the MCTS procedure used in [120] for rearrangement planning. To choose one of the 8 object-pushing directions (red arrows on the bottom left), MCTS expands a tree with a branching factor equals 8 over a short horizon. The tree is expanded by repeating the 4 steps: 1) select a potential tree policy; 2) expand the tree policy; 3) evaluate the policy by simulating with a separate, rollout policy; 4) back-propagating and update the evaluation. Throughout this procedure, the contacts between objects are considered as side-effects and not explicitly reasoned.

control is an area of interest in its own right, existing techniques for single-object manipulation control typically suffice in the multi-object setting. Rather, the main decision-making challenge is in the motion planning stage. Unlike single-object manipulation planning where the primary challenges are grasp planning and inverse kinematics, multi-object manipulation planning demands careful sequencing of which objects to move, where to move them to [48, 70, 71, 100, 127], and sometimes the use of simultaneous contacts to move multiple objects at once [31, 55, 86, 104, 120, 126]. Due to the large search space, finding (nearly) optimal motion plans is intractable, even in a simplified setting [106], as we discuss in Section V-B.

Conceptually, the goal of multi-object manipulation is to compute control signals $u(t)$ that efficiently bring the state of the joint robot-objects system x_{init} to the goal set $\mathcal{C}_{goal}$, possibly given partial and noisy observations of x [68]. The typical setting is hierarchical, in which the motion planner reasons about a reference trajectory $x^*(t), t \in [0, T]$ and the controller chooses $u(t)$ to execute it. We note that, compared to planning, the feedback control aspect of multi-object manipulation is often a secondary consideration, making use of standard grasping, trajectory following, and force control strategies. In the vast majority of works, decision-making takes place at a higher level in the planning hierarchy, where the task divides the trajectory x^* into multiple actions. Each action is specified using a manipulation type, such as pick and place, and a small set of parameters, such as a target location. In this section, we first review major theoretic results on the computational complexity of planning problems. Then, we delve into practical solutions which often involve task-specific heuristics to scale to more complex problems.

A. Formulations of the Planning Problem

Low-level planning: Given a predictive model, it is possible to formulate the multi-object manipulation problem as a kinodynamic motion planning problem [66, 120] or Markov decision process [3, 31] over low-level motions. The changes in contacts are considered side-effects of the motion and are not reasoned about explicitly. Breadth-first search, A* search, rapidly-exploring random trees (RRT), and Monte-Carlo tree search (MCTS illustrated in Figure 14) have been employed in solving these problems. These approaches can be successful for planar pushing, but for 3D manipulation or grasping, each manipulation action has a low likelihood of succeeding, leading to a minuscule chance of success in the task overall. Instead, most authors formulate multi-object manipulation problems as searching amongst *high-level actions*.

High-level planning: In high-level planning, a collection of high-level actions is provided to the robot, such as pick and place, each of which changes the contact state of the system. Each action is further parameterized by some set of parameters, e.g., pick(X) and place(X,Y) where X is an object identity and Y is a location, and the domains of each parameter may be discrete or continuous. The responsibility of the planner is then to sequence the high level actions and their parameter choices to accomplish the task. A key benefit of this approach is that the predictive model can be drastically simplified to ignore low-level physics, and can instead model only the preconditions and postconditions of each action. For example, an object placed on a flat surface clear of obstacles will stay in the specified location.

Particularly for navigation, rearrangement, sorting, packing, and declutter tasks, reducing the planning problem to action sequencing is far more tractable than planning low-level actions, and can be solved through A* and MCTS techniques. Nevertheless, common settings still pose significant computational challenges. We cover the computational complexity of these problems and effective heuristics in the following sections.

Task-and-motion planning: In certain problems, such as tabletop rearrangement with overhead grasps, the feasibility of every high-level action is guaranteed. But in other problems, the feasibility of a high-level action depends on the existence of a feasible low-level motion. Evaluating the existence requires reasoning about the geometry and/or planning a path. For example, navigation amongst movable obstacles requires determining whether a robot can reach a target object. The general problem of sequencing actions and finding feasible motions has been studied for decades [1, 2, 13, 42] and is now known as *task-and-motion planning* [39].

As illustrated in Figure 16, task-and-motion planning (TAMP) leverages symbolic reasoning to select which high-level actions (tasks) to sequence and how to plan motions between them [13]. Each task (high-level action) is defined with preconditions and postconditions that may hold over a symbolic space (e.g., Held(X), Blocking(X,Y)) and the task planner uses STRIPS-style reasoning to guide the search towards valid task sequences [40]. Feasible motion plans for such sequences will then be verified by calling a motion planner for each action. For example, a pick-and-place sequence is broken into motion planning problems for the *transit mode*, where no object is grasped, and the *transfer mode*, where an object is moved to a target location. Moreover, as a high-level action, a pick action requires choosing a grasp of the object as well as the transit motion to reach that grasp. A place action requires choosing a placement location for the grasped object that is stable and has no collisions with either the robot or

other objects.

A key challenge in TAMP is to allocate task planning and motion planning efforts when certain motion plans are infeasible. One strategy is to plan a sequence of tasks, perform motion planning, and incorporate the feedback of failed motion plans to discourage similar task-level plans in the future [23, 121]. However, modern motion planners based on sampling-based planning are only probabilistically complete, and it is challenging to decide upon a fixed time limit. Multi-modal planning (MMP) addresses this problem by reasoning explicitly about the joint multi-step motion planning problem. Any combination of actions generates a discrete graph of configuration spaces (modes), each with their own motion constraints, that intersect at some common transition configurations [1, 51]. Sampling-based planning effort is then distributed across the modes.

Although TAMP is appealing as a general-purpose framework, it has some drawbacks for multi-object manipulation. The use of STRIPS-like symbolic information allows TAMP solvers to automatically generate effective heuristics in many cases [40], but it is challenging to represent geometric interference constraints in a symbolic fashion suitable for identifying conflict-free task sequences [32]. A special case of geometric-logic constraints has been presented in [131], where geometric constraints are piece-wise continuous with a pre-specified, finite number of constraint-switching points. Moreover, general-purpose heuristics are typically much less effective than problem-specific heuristics. Second, for many problems such as packing, the performance bottleneck is not task sequencing, but rather finding optimized action parameters in continuous domains. Finally, it is worth noting that decoupling the discrete-continuous subproblems can produce sub-optimal motion plans. In the context of dual-arm rearrangement, Shome *et al.* [114] showed this sub-optimality gap can be large with excessive transit/transfer cost.

B. Hardness of Multi-Object Manipulation

The computational complexity of multi-object manipulation has been an area of great theoretical interest, and is closely related to the multi-robot coordination problem. The problem of coordinating moving rectangular objects in a plane is at least PSPACE-hard [52]. Demaine *et al.* [24] showed that solving navigation problems in general is NP-hard. Kavraki and Kolountzakis [62] showed that finding the trajectory to decompose two planar shapes is NP-complete. Some manipulation problems are not hard to solve but difficult to achieve optimality. For example, Tang and Yu [128] showed that finding an optimal declutter plan is NP-hard. Ratner and Warmuth [106] showed that finding an optimal push plan for the 15-puzzle is intractable and the problem in general is NP-hard. Han *et al.* [48] speculated that finding the optimal stack-rearrangement plan is NP-hard. Han *et al.* [49] proved the NP-hardness of cost-optimal, (un)labeled, (non-)overlapping, table-top rearrangement problems. Moreover, it has been shown that a general collision-free trajectory planning problem is PSPACE-hard [107]. This implies that the task-and-motion planning formulation is at least PSPACE-hard.

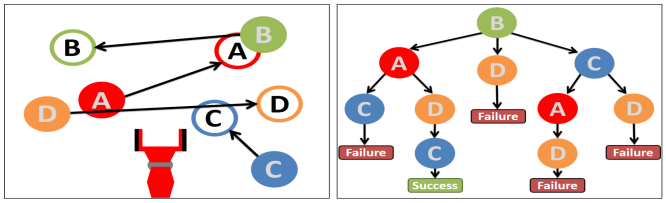


Fig. 15. (a): Krontiris and Bekris [69] considered a planar rearrangement problem, where the solid circles are the current object positions and the hollow circles are their target positions. (b): An illustration of the monotone (LP_1) assumption used in [69], in which case each object can move at most once and the planner only needs to determine the order of object movements. The size of a search tree becomes manageable under such an assumption.

Because these problems are intractable in the worst-case, theoretical analysis has turned to identifying problem subclasses that admit tractable algorithms, and to identify structures to accelerate planning for easier problem instances. In the context of arrangement, the problem classes of FP and LP [8] lend themselves to efficient solvers. We define a motion plan, for a specific object and target, to be feasible if the object can be moved to its target, considering all other objects as obstacles. The set of *flat problems* (FP) are defined as those in which every sequence of motion plans is feasible. These problems are easily solved by picking any ordering and calling motion planners. The set of *linear problems* (LP) are defined as problems in which there exists an ordering of objects that admit a sequence of feasible plans. Such problems are also known as *monotone* and are illustrated in Figure 15. The LP problem class is more challenging than FP, because it requires searching over $m!$ potential permutations of object orderings. To accelerate such a search, a *dependency graph* structure can be defined over the m obstacles which marks which obstacles block the plan of other obstacles, and the search can be efficiently restricted to sequences that obey the dependency structure [8, 70, 101].

For more general tasks, a similar problem class LP_1 was defined as the set of problems that can be solved by moving each object at most once, and was proposed originally in a navigation context [123]. Note that the target of each object is not defined in navigation tasks, so the planner chooses both the sequence and the final location of each object, and hence this problem subclass involves placement and singulation tasks. The object location should be chosen to change the connectedness of the feasible configuration space (C-space) of the robot and subsequent objects. In low-dimensional problems, the C-space connectivity for any given object arrangement can be enumerated, and obstacles affecting the connectivity can be detected as well [123]. This concept also leads to a dependency-graph-like structure that can accelerate the efficiency of search-based solvers. Divide-and-conquer and backchaining approaches can also be employed as in [95].

More complex *non-monotone* problems require moving objects aside to intermediate locations, and Ben-Shahar and Rivlin [8] introduced another rearrangement problem class LP-ϵ in which each object can be moved at most a distance ϵ from their original location. This problem class requires solving two LP problems, one to move objects to their intermediate

targets, and second to move objects to their final targets. The more general LP_k problem class allows each object to be moved up to k times, and Stilman and Kuffner [124] presented a search-based planner to address these problems. Yet more complex problems require the robot to move more than one object simultaneously, but rigorous solutions to such problems have received relatively little attention in the literature.

C. Task-Specific Solution Techniques

Despite pessimistic worst-case bounds, many tasks admit heuristics and simplifying assumptions that lead to practical solution techniques.

Analytical heuristics: Modern rearrangement planners accelerate the process of generating a discrete rearrangement plan using dependency graphs or other data structures [48, 69, 70, 71]. Krontiris *et al.* [70] use a pebble graph formulation in which unlabeled rearrangement queries may be answered via the linear-time feasibility test [4]. Krontiris and Bekris [71] can solve non-monotone instances of labeled rearrangement problems, where each object can be moved to multiple intermediary positions but stays at their final position once reached. This class of non-monotone instances is identified with the classical "Tower of Hanoi" puzzle.

A heuristic search method for pushing (sweeping) many dirt particles [31] used assignments to the closest goal as the heuristic. When assignments are mutually exclusive, i.e., no two objects can occupy the same goal, an assignment cost can be calculated as a minimization of estimated costs over any assignment of objects to goals, which can be solved using a linear program (e.g., Hungarian algorithm). This heuristic was applied for large-scale sorting-by-pushing in both randomized local search [55] and receding-horizon search [102] to scale to dozens of objects.

A heuristic for loose-packing [20] uses the overlapping volume as a heuristic for candidate object locations. A placement with lower overlap is likely to move fewer objects aside and with lower displacement than one with higher overlap.

A common heuristic for navigation problems is the use of a guide path for the robot that avoids obstacles as much as possible to identify a small set of objects that should be moved aside [17, 123]. These objects can then be focused on using heuristic manipulation strategies [17] or breadth-first search [123].

Dense packing problems have a long history in both 2D [89] and 3D bin packing [92], and are solved using search or various heuristics, such as the deepest-bottom-left-first (DBLF) heuristic. Dense-packing using robot manipulators have been considered recently with the key problem being the loading location. A heuristic greedy search using a variant of DBLF has been shown to pack more tightly given concavities found in non-convex objects [135].

Divide-and-conquer and backchaining approaches: These methods can be employed to speed up the search for object orderings [95]. Backward reasoning has also been used for placement planning to identify objects that need to be moved aside for the target object [20]. Breadth-first search is performed backward over a sequence of object pushes, and

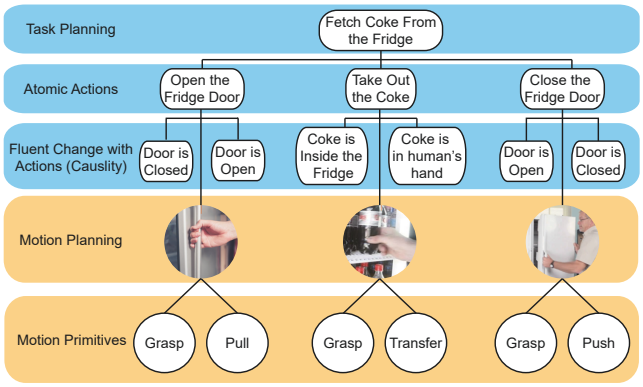


Fig. 16. We re-draw the TAMP illustration from [87] for planning a drink-fetching task. The task planner first determines the sequence of actions: open door, take the coke, and close door. Each action has pre- and post-conditions, which are input to the motion planner. The motion planner then seeks to satisfy these conditions in the parametric space of motion primitives.

restricted to objects that overlap the target placement. If a push makes a contact with another object, that the object is added to the list of candidate objects. In the simplified setting of a singulation problem where objects can be removed, recursive backchaining can determine the order of obstacle removal to reach a target object [78].

Greedy approaches are often successful in singulation and declutter problems, i.e., repeatedly selecting the next object to be moved by optimizing some value function, and then invoking a low-level planner to compute the trajectory. For singulation, prior works [14, 61, 73] rely on perception information to evaluate affordance and value of singulation actions, and pick the action with maximum value. For declutter, object selector should take into account clutter/pile stability and/or low-level planner feasibility. Prior works [100, 130] propose several heuristic selection rules based on the geometric relationships (e.g., height, contact normals) between objects. Sorting tasks, in which singulation of piles is the primary concern, can also be addressed by heuristics that determine whether to spread out a pile or perform a pick-and-place action [44]. Simple clustering metrics can be used as value functions for decluttering small particles, such as food pieces [126]. Han *et al.* [47] study picking from objects placed on a conveyor belt, and show that their optimal solution determined by an exhaustive search is approximately 10% better than a first-in-first-out heuristic. Another benefit of greedy approaches is that they can be applied both for planning and for closed-loop control. However, greedy approaches can get stuck in local minima, particularly for problems that require multi-step interactions.

Value and Q-function learning: To boost the performance of greedy approaches, several researchers have considered learning a value function or Q-function via reinforcement learning, which can then be greedily ascended. Q-functions $Q(x, a)$ are highly related to value functions $V(x)$ since they predict the value of taking an action in a given state. For example, decluttering has been cast as a reinforcement learning (RL) problem that predicts the values of singulation actions such as pushing, grasping, or sliding [10, 12, 16, 30, 60, 145]. Using simulation or real experience, rewards are awarded for

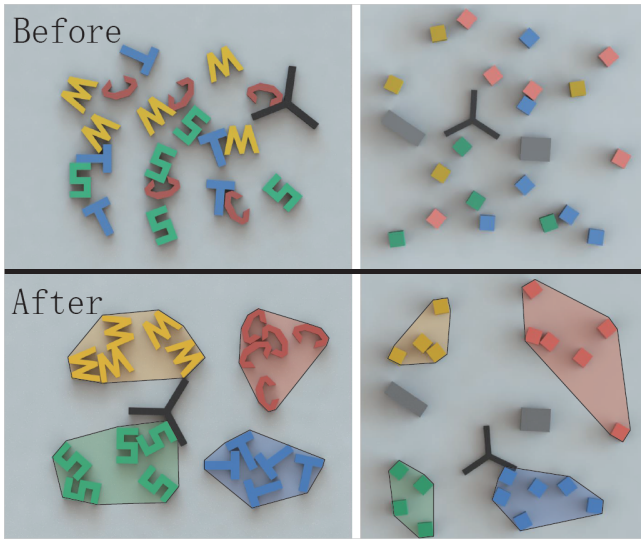


Fig. 17. Song *et al.* [120] proposed to cluster planar objects by their color. They used only push actions and successfully solve these problem instances within 200 actions.

successful grasps but penalties are awarded for unsuccessful ones. A recent trend is the use of deep networks in end-to-end methods, such as deep Q-learning, that predict values from sensor images without an intermediate state representation [10, 30, 60, 145]. These methods can use images annotated in the pixel space, e.g., with a segmentation mask [30], but otherwise avoid a separate perception step. A closely related notion to Q-functions is energy-based models, which learns an energy $E(x, a)$ such that the optimal action corresponds to an extremum of E. It has been shown in [36] that energy-based models out-perform RL on some multi-object, contact-rich manipulation tasks.

Policy learning: RL has also been used to learn manipulation policies directly. Recent works have used policy learning to address robot packing problems [53, 148]. The policy of MCTS rollout step can also be learned to improve the assessment of the value function at non-terminal nodes [120]. This approach has been shown to have considerable boost the success rate compared to random rollouts and is able to sort dozens of objects (Figure 17). Instead of RL, learning from demonstration has also been employed to learn the MCTS rollout policy for pushing rearrangement tasks [67].

Tuning or learning the action sampling strategy: The distribution of actions chosen by a sampling-based motion planner, heuristic search, and MCTS can be tuned or learned for better performance. We already discussed affordance prediction in Section III, which limits the number of actions that are explored in decision-making. Clustering heuristics were employed to identify promising actions for search in the task of decluttering dirt [31]. King *et al.* [65] employed a mixture of robot- and object-centric action samplers in a sampling-based motion planner and demonstrated superior performance compared to either sampler alone. Prioritization of object orientations that would be stable on a flat surface was shown to greatly improve the speed of dense packing [135].

D. Common Design Choices

In Table V, we summarize the common assumptions or design choices used across multiple tasks and analyze their implications. We select prior works that have explicitly claimed their assumptions. Some recent learning-based techniques might implicitly rely on these assumptions in the constructing of datasets or the design of experiments.

- *Separation* is widely used in rearrangement, decluttering, and sorting-by-packing tasks, and it assumes objects are sufficiently far from one another that grasping is always feasible and no object-object interaction occurs.
- *Finite state space* assumes that objects can only take on a discrete set of poses. This permits the use of combinatorial search which can provide a completeness or optimality guarantee within the set of allowed states.
- *Finite action space* assumes that the robot can only take finitely many possible actions, e.g., push along 8 different directions [120] or move in a grid [29]. This assumption enables search-based planners (e.g., A*, MCTS) to be applied directly without dynamic sampling of actions.
- The *sequential* assumption restricts the robot from moving more than one object at the same time. This also disallows multiple robots to operate simultaneously as in [114].
- The *planar* assumption restricts reasoning of robot and/or object movement to a plane. Perception is also simplified, since the state of the system is fully observed. In some settings, overhead grasping is allowed but the objects always return to a plane.
- The *monotone* assumption limits the number of times each object can be moved to reach their goal positions, including the LP_* problem classes. The Hanoi tower assumption is a special form that assumes objects can be moved multiple times but they stay fixed after reaching their goals [71].
- *Manipulation operator* assumptions restrict the number of possible high-level actions performed on objects. *3D Grasp* is the most general, and considers the full 3D geometry of the object and gripper. *Planar Grasp* is often used in navigation amongst movable obstacles, and allows the robot to touch the object at a single point to achieve prehensile contact. *Planar Push* uses non-prehensile pushing operations in a plane. *Shift* is a special operator used in object singulation in which the gripper touches an object from above and slides it horizontally [10].

Table V reveals the two most effective assumptions being sequential and planar, which have been used for all the tasks in both early and recent works. Yet there is a trend in recent research to lift the sequential assumption in sorting tasks and consider simultaneous object motions using planar pushing actions. Another effective way to simplify the problem is by restricting the sub-class of problems to LP_k or "Puzzle of Hanoi". Methods with such restrictions can typically provide completeness guarantees. Other assumptions (separation, finite states, and finite actions) are exclusively used for navigation and rearrangement.

E. Trends and Open Problems

Heuristic versus complete planner: Since manipulation

TABLE V
SUMMARIZING COMMON DECISION-MAKING ALGORITHM CHARACTERISTICS. WE ADD A $*$ AFTER THE PROBLEM CLASS IF A METHOD CAN FURTHER PROVIDE COMPLETENESS GUARANTEE, I.E., SOLVE ALL FEASIBLE PROBLEM INSTANCES IN THAT CLASS. THE POSSIBLE MANIPULATION OPERATORS ARE: 3DG=3D GRASP, 3DGT=3D GRASP+TOSSING, PG=PLANAR GRASP, PP=PLANAR PUSH, GP=GRASP+PUSH, GS=GRASP+SHIFT.

Manipulation Task	Reference	Separation	Finite States	Finite Actions	Sequential	Planar	Problem Class	Operator
Navigation	[123]	✓		✓	✓	✓	LP_1^*	PG
Navigation	[124]	✓		✓	✓	✓	LP_k^*	PG
Declutter	[100]				✓			3DG
Declutter	[128]			✓	✓			
Declutter	[12]							GP
Singulation	[14]				✓	✓		3DG
Singulation	[10]				✓	✓		GS
Rearrangement	[101]	✓			✓	✓	Hanoi	PG
Rearrangement	[71]	✓	✓	✓	✓	✓	Hanoi*	3DG
Rearrangement	[114]	✓	✓	✓	✓	✓	LP_1^*	3DG
Loose-Packing	[147]				✓			3DGT
Dense-Packing	[135]				✓			3DG
Navigation/ Placing	[66, 67, 68]					✓		PP
Placing	[20]				✓	✓		PP
Sorting-by-Packing/ Declutter	[129]	✓			✓			3DG
Sorting-by-Clustering/ Declutter	[31, 126]					✓		PP
Sorting-by-Clustering	[120]			✓		✓		PP

tasks are mostly NP-hard, relatively few prior works [47, 95, 114, 123, 124, 128] have analyzed planner optimality. Other works are either heuristic or only provide probabilistic completeness guarantee [65, 71, 136]. In practice, the benefits of providing completeness or optimality guarantees are not well understood. For example, MCTS can only approach optimal actions in the sampling limit. The case is similar with sampling-based motion planners. Due to the complexity of the problem and dimensionality of search spaces, planners can seldom get close to these limits in practice. On the other hand, we lack a systematic comparison of the effectiveness of various heuristics. The unanswered questions involve: Whether there exists a polynomial-time approximation scheme for NP-hard problem instances? To what extent a heuristic method performs worse than a complete or optimal algorithm? We noticed recent work [127] that shows the dynamic-programming heuristic has close-to-optimal performance in declutter problems and [114] compared the empirical performance of heuristic methods and the optimal mixed-integer programming method on rearrangement tasks, which are good starting points.

Reasoning about simultaneous object movements: The vast majority of past works assume one object moved at once, which leads to better tractability in combinatorial search, but with limited efficiency. Problems that are solvable with k objects moved simultaneously have been denoted SP^k [8]. Prior works on sorting [55, 102, 120, 126] and placement [20, 117] have enabled robots to push multiple objects simultaneously in greedy or receding-horizon. But due to the limited prediction horizon, completeness or optimality cannot be proven, leaving an open question for future theoretical work.

Learning-based approaches: Most learning-based techniques so far have used supervised learning from sampled datasets, either trained via simulation or on real systems [120, 144, 147]. Pure reinforcement learning techniques have not been extensively used in multi-object manipulations, except for [60, 64, 145]. This is presumably due to the sensitivity of reinforcement learning to parameters, insufficient data efficiency, and sim-to-real discrepancy. Moreover, end-to-end visual-motor policies are less explainable and unfriendly to parameter tuning as compared with training different sub-network modules. Indeed, several works have shown that learning can play an assistive role to guide an online planner, such as learned rollout policies that guide MCTS [120].

Mechanism and behavior co-design: The vast majority of research on multi-object manipulation assumes a fixed design of robot and end-effectors. Multi-object push with non-end-effector limbs [21] or grasp with a multi-fingered hand [50], claw, or scoop could improve task efficiency. The tool choice question is still underappreciated in the multi-object manipulation literature, and would be interesting to explore since the choice of tool geometry, material, and actuation characteristics can influence the performance and completeness of planning.

Feedback control: The vast majority of decision-making work does not explicitly reason about uncertainty and errors, and if any approach is taken at all, it is likely to involve reactive replanning. Greedy methods and receding-horizon methods, such as MCTS, are well suited to replanning. To improve the robustness of multi-object manipulation in real systems, the robot can be augmented with operators to correct for uncertainty [115], or reason explicitly about uncertainty in prediction [3, 67, 73, 136].

VI. CONCLUDING DISCUSSIONS

From its roots in theoretical planning research, multi-object manipulation has matured quickly in the last few years to achieve practical demonstrations on physical robot systems. Several open problems remain for multi-object manipulation research, and overcoming these issues will help accelerate the pace of innovation towards the goal of real-world deployment.

Combining learning and analytical approaches: Research trends are increasingly integrating learning components into perception, prediction, and decision making. End-to-end learning approaches represent one end of the spectrum, but analytical approaches still outperform learning in complex planning problems and are more practical when experience is expensive to collect. A major challenge for future work is combining the two paradigms to build adaptive, scalable, and practical systems.

Integration with open-world perception: Perception systems for multi-object manipulation typically make strong assumptions, such as known objects and a top-down or nearly occlusion-free view. To lift these assumptions and move robots out of controlled settings, a perception algorithm needs to handle a diverse range of objects and severely occluded scenarios. In addition, the perceptor should be aware of outliers including non-movable or irrelevant objects. A deeper understanding of uncertainties can also be integrated into planning and prediction. Prior works have the robot proactively push objects to determine mobility [142] and change viewpoints to discover unseen objects [57]. Another intriguing direction is to use quasistatic force analysis to help infer states of unseen objects [113]. Yet another topic that has been lightly explored is incorporating human preferences into goals for planning, such as rearranging objects to be more aesthetically pleasing in retail applications, or to maximize accessibility of frequently used items in kitchens.

Eulerian versus Lagrangian prediction: Lagrangian models have predominated the many-object manipulation community for decades, while Eulerian alternatives have been explored only recently [31, 126]. Eulerian models are known to be more efficient with a high number of objects while Lagrangian models favor fidelity and allow modeling accurate contact mechanics. However, it is unclear what is the best point of switching between the two models and how to combine their merits. A promising direction is to identify important objects as done in [116] and then use different representations for the important and less-important ones. Further, Eulerian models can seamlessly work with a perceptor in image feature space, but it is less amenable to planning algorithms that are based on object-centric representations. Further investigations of Eulerian models are needed to unlock their full potential in terms of efficiency, fidelity, and amenability to planning and perception.

Connections with continuum manipulation and scaling to many objects: There is a recent surge of interest in continuum manipulation such as liquid transfer [83, 103, 112], sand paving [64], and shaping of plastic objects [18, 85, 143]. Indeed, a continuum manipulation can be approximated as a multi-object manipulation problem by modeling the continuum as a large number of particles stitched together via material-specific constitutive laws [84, 85, 111]. With many particles, it is not possible to reason about particles one by one, and instead continuum manipulation approaches tend to use Eulerian models (see e.g., [72]) and parameterized manipulation actions (e.g., liquid transfer by pouring, sand paving by scooping, and plastic shaping by pushing). Decision-making has been guided by simple control rules such as PID control [111], trajectory optimization [103] and heuristics [18]. Similar strategies have been adopted in multi-object manipulation for declutter tasks [31, 126].

The line between multi-object and continuum manipulation problems becomes blurry as the number of objects m grows. At first glance, complete planning appears intractable with large m, unless the unit of reasoning is relaxed to include groups of objects (e.g., clusters of particles that are simultaneously movable [55]). By reasoning about groups, the action space and thus branching factor can be reduced. However, it is unclear how to determine the best decision granularity for all problems. Investigating the large m problem has the potential to make novel contributions to both the multi-object and continuum domains.

ACKNOWLEDGEMENT

This work is partially supported by an Amazon Research Award and NSF Grant #1911087. We thank all the authors to grant us the permission to use the figures in their papers.

REFERENCES

[1] R. Alami, J.-P. Laumond, and T. Siméon, "Two manipulation planning algorithms," in *WAFR Proceedings of the workshop on Algorithmic foundations of robotics*, AK Peters, Ltd., 1994, pp. 109–125.

[2] R. Alami, T. Simeon, and J.-P. Laumond, "A geometrical approach to planning manipulation tasks. the case of discrete placements and grasps," in *International Symposium on Robotics Research*, MIT Press, 1990, pp. 453–463.

[3] A. S. Anders, L. P. Kaelbling, and T. Lozano-Perez, "Reliably arranging objects in uncertain domains," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1603–1610.

[4] V. Auletta, A. Monti, M. Parente, and P. Persiano, "A linear-time algorithm for the feasibility of pebble motion on trees," *Algorithmica*, vol. 23, no. 3, pp. 223–245, 1999.

[5] D. Batra, A. X. Chang, S. Chernova, A. J. Davison, J. Deng, V. Koltun, S. Levine, J. Malik, I. Mordatch, R. Mottaghi, *et al.*, "Rearrangement: A challenge for embodied AI," vol. abs/2011.01975, 2020.

[6] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende, and k. kavukcuoglu, "Interaction networks for learning about objects, relations and physics," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., Curran Associates, Inc., 2016, pp. 4502–4510.

[7] W. Bejjani, R. Papallas, M. Leonetti, and M. R. Dogar, "Planning with a receding horizon for manipulation in clutter using a learned value function," in *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)*, 2018, pp. 1–9.

[8] O. Ben-Shahar and E. Rivlin, "Practical pushing planning for rearrangement tasks," *IEEE Transactions on Robotics and Automation*, vol. 14, no. 4, pp. 549–565, 1998.

[9] D. J. Benson and S. Okazawa, "Contact in a multi-material eulerian finite element formulation," *Computer Methods in Applied Mechanics and Engineering*, vol. 193, no. 39, pp. 4277–4298, 2004, The Arbitrary Lagrangian-Eulerian Formulation.

[10] L. Berscheid, P. Meißner, and T. Kröger, "Robot learning of shifting objects for grasping in cluttered environments," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 612–618.

[11] A. Billard and D. Kragic, "Trends and challenges in robot manipulation," *Science*, vol. 364, no. 6446, 2019.

[12] A. Boularias, J. A. Bagnell, and A. Stentz, "Learning to manipulate unknown objects in clutter by reinforcement," in *29th AAAI Conference on Artificial Intelligence, AAAI 2015 and the 27th Innovative Applications of Artificial Intelligence Conference, IAAI 2015*, AI Access Foundation, 2015, pp. 1336–1342.

[13] S. Cambon, R. Alami, and F. Gravot, "A hybrid approach to intricate motion, manipulation and task planning," *The International Journal of Robotics Research*, vol. 28, no. 1, pp. 104–126, 2009.

[14] L. Chang, J. R. Smith, and D. Fox, "Interactive singulation of objects from a pile," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3875–3882.

[15] L. Y. Chang, S. S. Srinivasa, and N. S. Pollard, "Planning pre-grasp manipulation for transport tasks," in *2010 IEEE International Conference on Robotics and Automation (ICRA)*, 2010, pp. 2697–2704.

[16] J. Chen, T. Fujinami, and E. Li, "Deep Bin Picking with Reinforcement Learning," 2017.

[17] P. C. Chen and Y. K. Hwang, "Practical path planning among movable obstacles," in *1991 IEEE International Conference on Robotics and Automation (ICRA)*, 1991, 444–449 vol.1.

[18] A. Cherubini, V. Ortenzi, A. Cosgun, R. Lee, and P. Corke, "Model-free vision-based shaping of deformable plastic materials," *The International Journal of Robotics Research*, vol. 0, no. 0, p. 0 278 364 920 907 684, 0.

[19] N. Correll, K. E. Bekris, D. Berenson, O. Brock, A. Causo, K. Hauser, K. Okada, A. Rodriguez, J. M. Romano, and P. R. Wurman, "Analysis and observations from the first amazon picking challenge," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 172–188, 2016.

[20] A. Cosgun, T. Hermans, V. Emeli, and M. Stilman, "Push planning for object placement on cluttered table surfaces," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 4627–4632.

[21] A. Cosgun, L. Ditria, S. D'Lima, and T. Drummond, "Embracing contact: Pushing multiple objects with robot's forearm," *ArXiv*, vol. abs/1906.06866, 2019.

[22] N. C. Dafle, R. Holladay, and A. Rodriguez, "In-hand manipulation via motion cones," in *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, Jun. 2018.

[23] N. T. Dantam, Z. K. Kingston, S. Chaudhuri, and L. E. Kavraki, "Incremental task and motion planning: A constraint-based approach.," in *Robotics: Science and systems*, Ann Arbor, MI, USA, vol. 12, 2016, p. 00 052.

[24] E. D. Demaine, M. L. Demaine, and J. O'Rourke, "PushPush and Push-1 are NP-hard in 2D," in *Proceedings of the 12th Annual Canadian Conference on Computational Geometry (CCCG 2000)*, Fredericton, New Brunswick, Canada, Aug. 2000, pp. 211–219.

[25] X. Deng, A. Mousavian, Y. Xiang, F. Xia, T. Bretl, and D. Fox, "Poserbpf: A rao-blackwellized particle filter for 6d object pose tracking," *Robotics: Science and Systems (RSS)*, 2019.

[26] M. Dogar and S. Srinivasa, "A framework for push-grasping in clutter," *Robotics: Science and Systems VII*, pp. 65–72, 2011.

[27] M. R. Dogar, M. C. Koval, A. Tallavajhula, and S. S. Srinivasa, "Object search by manipulation," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 4973–4980.

[28] S. Dong, D. K. Jha, D. Romeres, S. Kim, D. Nikovski, and A. Rodriguez, "Tactile-rl for insertion: Generalization to objects of unknown geometry," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6437–6443.

[29] D. Dor and U. Zwick, "SOKOBAN and other motion planning problems," *Computational Geometry: Theory and Applications*, 1999.

[30] A. Eitel, N. Hauff, and W. Burgard, "Learning to Singulate Objects Using a Push Proposal Network," *ArXiv*, vol. abs/1707.08101, 2020.

[31] S. Elliott and M. Cakmak, "Robotic cleaning through dirt rearrangement planning with learned transition models," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 1623–1630.

[32] E. Erdem, K. Haspalamutgil, C. Palaz, V. Patoglu, and T. Uras, "Combining high-level causal reasoning with low-level geometric reasoning and motion planning for robotic manipulation," in *2011 IEEE International Conference on Robotics and Automation*, IEEE, 2011, pp. 4575–4581.

[33] T. Erez, Y. Tassa, and E. Todorov, "Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 4397–4404.

[34] N. Fazeli, M. Oller, J. Wu, Z. Wu, J. B. Tenenbaum, and A. Rodriguez, "See, feel, act: Hierarchical learning for complex manipulation skills with multisensory fusion," *Science Robotics*, 2019.

[35] P. Florence, L. Manuelli, and R. Tedrake, "Self-supervised correspondence in visuomotor policy learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 492–499, 2020.

[36] P. Florence, C. Lynch, A. Zeng, O. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," *Conference on Robot Learning (CoRL)*, Nov. 2021.

[37] P. Florence, C. Lynch, A. Zeng, O. A. Ramirez, A. Wahid, L. Downs, A. Wong, J. Lee, I. Mordatch, and J. Tompson, "Implicit behavioral cloning," in *Conference on Robot Learning*, PMLR, 2022, pp. 158–168.

[38] W. Gao and R. Tedrake, "Kpam 2.0: Feedback control for category-level robotic manipulation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 2962–2969, 2021.

[39] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual review of control, robotics, and autonomous systems*, vol. 4, pp. 265–293, 2021.

[40] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling, "Ffrob: An efficient heuristic for task and motion planning," in *Algorithmic Foundations of Robotics XI*, Springer, 2015, pp. 179–195.

[41] S. Goyal, A. Ruina, and J. Papadopoulos, "Planar sliding with dry friction. part 1. limit surface and moment function," *Wear*, no. 143, 307—330, 1991.

[42] F. Gravot, S. Cambon, and R. Alami, "Asymov: A planner that deals with intricate symbolic and geometric problems," in *Robotics Research. The Eleventh International Symposium*, Springer, 2005, pp. 100–110.

[43] O. Groth, F. B. Fuchs, I. Posner, and A. Vedaldi, "Shapestacks: Learning vision-based physical intuition for generalised object stacking," in *The European Conference on Computer Vision (ECCV)*, Sep. 2018.

[44] M. Gupta and G. S. Sukhatme, "Using manipulation primitives for brick sorting in clutter," in *2012 IEEE International Conference on Robotics and Automation (ICRA)*, 2012, pp. 3883–3889.

[45] M. Halm and M. Posa, "Modeling and Analysis of Non-unique Behaviors in Multiple Frictional Impacts," in *Robotics: Science and Systems (RSS)*, Freiburg im Breisgau, Germany, 2019.

[46] P. Hämäläinen, J. Rajamäki, and C. K. Liu, "Online control of simulated humanoids using particle belief propagation," *ACM Trans. Graph.*, vol. 34, no. 4, Jul. 2015.

[47] S. D. Han, S. W. Feng, and J. Yu, "Toward fast and optimal robotic pick-and-place on a moving conveyor," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 446–453, 2020.

[48] S. D. Han, N. M. Stiffler, K. E. Bekris, and J. Yu, "Efficient, high-quality stack rearrangement," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1608–1615, 2018.

[49] S. D. Han, N. M. Stiffler, A. Krontiris, K. E. Bekris, and J. Yu, "Complexity results and fast methods for optimal tabletop rearrangement with overhand grasps," *The International Journal of Robotics Research*, vol. 37, no. 13-14, pp. 1775–1795, 2018.

[50] K. Harada and M. Kaneko, "Kinematics and internal force in grasping multiple objects," in *1998 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 1, 1998, 298–303 vol.1.

[51] K. Hauser and J.-C. Latombe, "Multi-modal planning in non-expansive spaces," *Intl. J. of Robotics Research*, vol. 29, no. 7, pp. 897–915, 2010.

[52] J. Hopcroft, J. Schwartz, and M. Sharir, "On the complexity of motion planning for multiple independent objects; pspace-hardness of the "warehouseman's problem"," *The International Journal of Robotics Research*, vol. 3, no. 4, pp. 76–88, 1984.

[53] R. Hu, J. Xu, B. Chen, M. Gong, H. Zhang, and H. Huang, "Tap-net: Transport-and-pack using reinforcement learning," *ACM Trans. Graph.*, vol. 39, no. 6, Nov. 2020.

[54] B. Huang, S. D. Han, A. Boularias, and J. Yu, "Dipn: Deep interaction prediction network with application to clutter removal," *ArXiv*, vol. abs/2011.04692, 2020.

[55] E. Huang, Z. Jia, and M. T. Mason, "Large-scale multi-object rearrangement," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 211–218.

[56] M Jaward, L Mihaylova, N Canagarajah, and D Bull, "Multiple object tracking using particle filters," in *2006 IEEE Aerospace Conference*, IEEE, 2006, 8–pp.

[57] D. Jayaraman and K. Grauman, "Learning to look around: Intelligently exploring unseen environments for unknown tasks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1238–1247.

[58] Jiaji Zhou, R. Paolini, J. A. Bagnell, and M. T. Mason, "A convex polynomial force-motion model for planar sliding: Identification and application," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 372–377.

[59] Y. Jiang, S. Moseson, and A. Saxena, "Efficient grasping from rgbd images: Learning using a new rectangle representation," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3304–3311.

[60] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, and S. Levine, "Scalable deep reinforcement learning for vision-based robotic manipulation," in *Proceedings of The 2nd Conference on Robot Learning*, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., ser. Proceedings of Machine Learning Research, vol. 87, PMLR, Oct. 2018, pp. 651–673.

[61] D. Katz, A. Venkatraman, M. Kazemi, J. A. Bagnell, and A. Stentz, "Perceiving, learning, and exploiting object affordances for autonomous pile manipulation," *Autonomous Robots*, vol. 37, no. 4, pp. 369–382, 2014.

[62] L. E. Kavraki and M. N. Kolountzakis, "Partitioning a planar assembly into two connected parts is np-complete," *Information Processing Letters*, vol. 55, no. 3, pp. 159–165, 1995.

[63] M. Kiatos and S. Malassiotis, "Robust object grasping in clutter via singulation," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1596–1600.

[64] W. Kim, C. Pavlov, and A. M. Johnson, "Developing a Simple Model for Sand-Tool Interaction and Autonomously Shaping Sand," *ArXiv*, vol. abs/1908.02745, 2019.

[65] J. E. King, M. Cognetti, and S. S. Srinivasa, "Rearrangement planning using object-centric and robot-centric action spaces," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3940–3947.

[66] J. E. King, J. A. Haustein, S. S. Srinivasa, and T. Asfour, "Nonprehensile whole arm rearrangement planning on physics manifolds," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2508–2515.

[67] J. E. King, V. Ranganeni, and S. S. Srinivasa, "Unobservable monte carlo planning for nonprehensile rearrangement tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4681–4688.

[68] J. E. King and S. S. Srinivasa, "Rearrangement Planning via Heuristic Search," *ArXiv*, vol. abs/1603.08642, 2016.

[69] A. Krontiris and K. E. Bekris, "Efficiently solving general rearrangement tasks: A fast extension primitive for an incremental sampling-based planner," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 3924–3931.

[70] A. Krontiris, R. Shome, A. Dobson, A. Kimmel, and K. Bekris, "Rearranging similar objects with a manipulator using pebble graphs," in *2014 IEEE-RAS International Conference on Humanoid Robots*, 2014, pp. 1081–1087.

[71] A. Krontiris and K. E. Bekris, "Dealing with difficult instances of object rearrangement," *Robotics: Science and Systems (RSS)*, 2015.

[72] K.-S. Ku, C.-H. An, K.-C. Li, and M.-I. Kim, "An eulerian model for the motion of granular material with a large stokes number in fluid flow," *International Journal of Multiphase Flow*, vol. 92, pp. 140–149, 2017.

[73] N. B. Kumbla, S. Thakar, K. N. Kaipa, J. Marvel, and S. K. Gupta, "Simulation based on-line evaluation of singulation plans to handle perception uncertainty in robotic bin picking," in *ASME 2017 12th International Manufacturing Science and Engineering Conference, MSEC 2017 collocated with the JSME/ASME 2017 6th International Conference on Materials and Processing*, 2017.

[74] Y. Labbé, S. Zagoruyko, I. Kalevatykh, I. Laptev, J. Carpentier, M. Aubry, and J. Sivic, "Monte-carlo tree search for efficient visually guided rearrangement planning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3715–3722, 2020.

[75] L. Ladický, S. Jeong, B. Solenthaler, M. Pollefeys, and M. Gross, "Data-driven fluid simulations using regression forests," *ACM Trans. Graph.*, vol. 34, no. 6, Oct. 2015.

[76] A. S. Lambert, M. Mukadam, B. Sundaralingam, N. Ratliff, B. Boots, and D. Fox, "Joint inference of kinematic and force trajectories with visuo-tactile sensing," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 3165–3171.

[77] D.-H. Lee, J. Kang, and P Xirouchakis, "Disassembly planning and scheduling: Review and further research," *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, vol. 215, no. 5, pp. 695–709, 2001.

[78] J. Lee, Y. Cho, C. Nam, J. Park, and C. Kim, "Efficient obstacle rearrangement for object manipulation tasks in cluttered environments," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 183–189.

[79] I. Leizea, H. Álvarez, and D. Borro, "Real time non-rigid 3d surface tracking using particle filter," *Computer Vision and Image Understanding*, vol. 133, pp. 51–65, 2015.

[80] D. I. W. Levin, J. Litven, G. L. Jones, S. Sueda, and D. K. Pai, "Eulerian solid simulation with contact," *ACM Trans. Graph.*, vol. 30, no. 4, Jul. 2011.

[81] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *J. Mach. Learn. Res.*, vol. 17, no. 1, 1334–1373, Jan. 2016.

[82] Y. Li, J. Wu, J. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *2019 IEEE International Con-*

ference on Robotics and Automation (ICRA), 2019, pp. 1205–1211.

[83] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," *arXiv preprint arXiv:2107.04004*, 2021.

[84] Y. Li, T. Lin, K. Yi, D. Bear, D. Yamins, J. Wu, J. Tenenbaum, and A. Torralba, "Visual grounding of learned physical models," in *International conference on machine learning*, PMLR, 2020, pp. 5927–5936.

[85] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *International Conference on Learning Representations*, 2019.

[86] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake, "Propagation networks for model-based control under partial observation," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019, pp. 1205–1211.

[87] J. Lin, X. Guo, J. Shao, C. Jiang, Y. Zhu, and S.-C. Zhu, "A virtual reality platform for dynamic human-scene interaction," in *SIGGRAPH ASIA 2016 virtual reality meets physical reality: Modelling and simulating virtual humans and environments*, 2016, pp. 1–4.

[88] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*, Springer, 2016, pp. 21–37.

[89] A. Lodi, S. Martello, and M. Monaci, "Two-dimensional packing problems: A survey," *European Journal of Operational Research*, vol. 141, no. 2, pp. 241 –252, 2002.

[90] J. Mahler and K. Goldberg, "Learning deep policies for robot bin picking by simulating robust grasping sequences," in *Conference on Robot Learning (CoRL)*, 2017.

[91] L. Manuelli, Y. Li, P. Florence, and R. Tedrake, "Keypoints into the future: Self-supervised correspondence in model-based reinforcement learning," in *Conference on Robot Learning (CoRL)*, 2020.

[92] S. Martello, D. Pisinger, and D. Vigo, "The three-dimensional bin packing problem," *Operations Research*, vol. 48, no. 2, pp. 256–267, 2000.

[93] C. Melhuish, A. B. Sendova-Franks, S. Scholes, I. Horsfield, and F. Welsby, "Ant-inspired sorting by robots: The importance of initial clustering," *Journal of the Royal Society Interface*, vol. 3, no. 7, pp. 235–242, 2006.

[94] C. Mitash, A. Boularias, and K. Bekris, "Physics-based scene-level reasoning for object pose estimation in clutter," *The International Journal of Robotics Research*, 2019.

[95] S. K. Moghaddam and E. Masehian, "Planning Robot Navigation among Movable Obstacles (NAMO) through a Recursive Approach," *Journal of Intelligent and Robotic Systems: Theory and Applications*, 2016.

[96] R. Mojtahedzadeh, A. Bouguerra, E. Schaffernicht, and A. J. Lilienthal, "Support relation analysis and decision making for safe robotic manipulation tasks," *Robotics and Autonomous Systems*, vol. 71, pp. 99–117, 2015.

[97] A. S. Morgan, B. Wen, J. Liang, A. Boularias, A. M. Dollar, and K. Bekris, "Vision-driven Compliant Manipulation for Reliable; High-Precision Assembly Tasks," in *Proceedings of Robotics: Science and Systems*, Virtual, Jul. 2021.

[98] C. Mucchiani and M. Yim, "A novel underactuated end-effector for planar sequential grasping of multiple objects," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020, pp. 8935–8941.

[99] C. Nam, J. Lee, Y. Cho, J. Lee, D. H. Kim, and C. Kim, "Planning for target retrieval using a robotic manipulator in cluttered and occluded environments," *ArXiv*, vol. abs/1907.03956,

[100] O. Ornan and A. Degani, "Toward autonomous disassembling of randomly piled objects with minimal perturbation," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 4983–4989.

[101] J. Ota, "Rearrangement planning of multiple movable objects by using real-time search methodology," in *2002 IEEE International Conference on Robotics and Automation (ICRA)*, vol. 1, 2002, 947–953 vol.1.

[102] Z. Pan and K. Hauser, "Decision making in joint push-grasp action space for large-scale object sorting," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[103] Z. Pan and D. Manocha, "Motion planning for fluid manipulation using simplified dynamics," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4224–4231.

[104] Z. Pan and K. Hauser, "Decision making in joint push-grasp action space for large-scale object sorting," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 6199–6205.

[105] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, pp. 8026–8037, 2019.

[106] D. Ratner and M. K. Warmuth, "Finding a shortest solution for the nxn extension of the 15-puzzle is intractable.," in *AAAI*, 1986, pp. 168–172.

[107] J. H. Reif, "Complexity of the mover's problem and generalizations," in *20th Annual Symposium on Foundations of Computer Science (sfcs 1979)*, 1979, pp. 421–427.

[108] D. Reznik and J. Canny, "A flat rigid plate is a universal planar manipulator," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 2, 1998, 1471–1477 vol.2.

[109] S. Rodriguez, M. Morales, and N. M. Amato, "Multi-agent push behaviors for large sets of passive objects," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4437–4442.

[110] R. Sarc, A. Curtis, L. Kandlbauer, K. Khodier, K. E. Lorber, and R. Pomberger, "Digitalisation and intelligent robotics in value chain of circular economy oriented waste management–a review," *Waste Management*, vol. 95, pp. 476–492, 2019.

[111] C. Schenck, "Liquids & Robots : An Investigation of Techniques for Robotic Interaction with Liquids," PhD thesis, 2018.

[112] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning robotic manipulation of granular media," in *Conference on Robot Learning (CoRL)*, 2017.

[113] T. Shao, A. Monszpart, Y. Zheng, B. Koo, W. Xu, K. Zhou, and N. J. Mitra, "Imagining the unseen: Stability-based cuboid arrangements for scene understanding," *ACM Trans. Graph.*, vol. 33, no. 6, Nov. 2014.

[114] R. Shome, K. Solovey, J. Yu, K. Bekris, and D. Halperin, "Fast, high-quality two-arm rearrangement in synchronous, monotone tabletop setups," *IEEE Transactions on Automation Science and Engineering*, 2021.

[115] R. Shome, W. N. Tang, C. Song, C. Mitash, H. Kourtev, J. Yu, A. Boularias, and K. E. Bekris, "Tight robot packing in the real world," *ArXiv*, vol. 1903.0098, 2020.

[116] T. Silver, R. Chitnis, A. Curtis, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling, "Planning with learned object importance in large problem instances using graph neural networks," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 13, pp. 11 962–11 971, May 2021.

[117] C. Song and A. Boularias, "Object rearrangement with nested nonprehensile manipulation actions," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 6578–6585.

[118] C. Song and A. Boularias, "A probabilistic model for planar sliding of objects with unknown material properties: Identification and robust planning," in *2020 IEEE/RSJ International*

Conference on Intelligent Robots and Systems (IROS), 2020, pp. 5311–5318.

[119] ——, "Learning to slide unknown objects with differentiable physics simulations," *Robotics: Science and Systems (RSS)*, 2020.

[120] H. Song, J. A. Haustein, W. Yuan, K. Hang, M. Y. Wang, D. Kragic, and J. A. Stork, "Multi-Object Rearrangement with Monte Carlo Tree Search: A Case Study on Planar Nonprehensile Sorting," *ArXiv*, vol. abs/1912.07024, 2019.

[121] S. Srivastava, E. Fang, L. Riano, R. Chitnis, S. Russell, and P. Abbeel, "Combined task and motion planning through an extensible planner-independent interface layer," in *2014 IEEE international conference on robotics and automation (ICRA)*, IEEE, 2014, pp. 639–646.

[122] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM Review*, vol. 42, no. 1, pp. 3–39, 2000.

[123] M. Stilman and J. Kuffner, "Navigation among movable obstacles: Real-time reasoning in complex environments," in *4th IEEE/RAS International Conference on Humanoid Robots, 2004.*, vol. 1, 2004, 322–341 Vol. 1.

[124] M. Stilman and J. Kuffner, "Planning among movable obstacles with artificial constraints," *The International Journal of Robotics Research*, vol. 27, no. 11-12, pp. 1295–1307, 2008.

[125] J. Stüber, C. Zito, and R. Stolkin, *Let's Push Things Forward: A Survey on Robot Pushing*, 2020.

[126] H. J. T. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," *Algorithmic Foundations of Robotics XIV*, pp. 347–363, Feb. 2021.

[127] W. N. Tang, S. D. Han, and J. Yu, "Computing high-quality clutter removal solutions for multiple robots," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[128] W. N. Tang and J. Yu, "Taming combinatorial challenges in optimal clutter removal tasks," *ArXiv*, vol. abs/1905.13530, 2019.

[129] A. K. Tanwani, N. Mor, J. Kubiatowicz, J. E. Gonzalez, and K. Goldberg, "A fog robotics approach to deep robot learning: Application to object recognition and grasp planning in surface decluttering," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 4559–4566.

[130] S. Temtsin and A. Degani, "Decision-making algorithms for safe robotic disassembling of randomly piled objects," *Advanced Robotics*, 2017.

[131] M. Toussaint, "Logic-geometric programming: An optimization-based approach to combined task and motion planning," in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI'15, Buenos Aires, Argentina: AAAI Press, 2015, 1930–1936.

[132] A. Vardy, "Accelerated patch sorting by a robotic swarm," *Proceedings of the 2012 9th Conference on Computer and Robot Vision, CRV 2012*, no. August, pp. 314–321, 2012.

[133] A. Vardy, G. Vorobyev, and W. Banzhaf, "Cache consensus: Rapid object sorting by a robotic swarm," *Swarm Intelligence*, vol. 8, no. 1, pp. 61–87, 2014.

[134] K. Wada, S. Kitagawa, K. Okada, and M. Inaba, "Instance segmentation of visible and occluded regions for finding and picking target from a pile of objects," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 2048–2055.

[135] F. Wang and K. Hauser, "Stable bin packing of non-convex 3d objects with a robot manipulator," in *2019 IEEE International Conference on Robotics and Automation (ICRA)*, 2019, pp. 8698–8704.

[136] F. Wang and K. Hauser, "Robot packing with known items and nondeterministic arrival order," *Robotics: Science and Systems (RSS)*, 2019.

[137] N. Watters, D. Zoran, T. Weber, P. Battaglia, R. Pascanu, and A. Tacchetti, "Visual interaction networks: Learning a physics simulator from video," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., Curran Associates, Inc., 2017, pp. 4539–4547.

[138] ——, "Visual Interaction Networks: Learning a Physics Simulator from Video," in *Advances in Neural Information Processing Systems 30*, I Guyon, U. V. Luxburg, S Bengio, H Wallach, R Fergus, S Vishwanathan, and R Garnett, Eds., Curran Associates, Inc., 2017, pp. 4539–4547.

[139] M. Wilson, "Developments in robot applications for food manufacturing," *Industrial Robot: An International Journal*, 2010.

[140] L. L. Wong, L. P. Kaelbling, and T. Lozano-Pérez, "Manipulation-based active search for occluded objects," in *2013 IEEE International Conference on Robotics and Automation*, 2013, pp. 2814–2819.

[141] K. Xu, H. Huang, Y. Shi, H. Li, P. Long, J. Caichen, W. Sun, and B. Chen, "Autoscanning for coupled scene reconstruction and proactive object analysis," *ACM Trans. Graph.*, vol. 34, no. 6, Oct. 2015.

[142] ——, "Autoscanning for coupled scene reconstruction and proactive object analysis," *ACM Trans. Graph.*, vol. 34, no. 6, Oct. 2015.

[143] H. Yin, A. Varava, and D. Kragic, "Modeling, learning, perception, and control methods for deformable object manipulation," *Science Robotics*, vol. 6, no. 54, 2021.

[144] K. Zakka, A. Zeng, J. Lee, and S. Song, "Form2fit: Learning shape priors for generalizable assembly from disassembly," *ArXiv*, vol. abs/1910.13675, 2019.

[145] A. Zeng, S. Song, S. Welker, J. Lee, A. Rodriguez, and T. Funkhouser, "Learning synergies between pushing and grasping with self-supervised deep reinforcement learning," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 4238–4245.

[146] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," *arXiv preprint arXiv:2010.14406*, 2020.

[147] A. Zeng, S. Song, J. Lee, A. Rodriquez, and T. A.Funkouser, "Tossingbot: Learning to throw arbitrary objects with residual physics," *Robotics: Science and Systems (RSS)*, 2019.

[148] H. Zhao, Q. She, C. Zhu, Y. Yang, and K. Xu, "Online 3d bin packing with constrained deep reinforcement learning," in *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, AAAI Press, 2021, pp. 741–749.

[149] J. Zhou, Y. Hou, and M. T. Mason, "Pushing revisited: Differential flatness, trajectory planning, and stabilization," *The International Journal of Robotics Research*, vol. 38, no. 12-13, pp. 1477–1489, 2019.

[150] J. Zhou, M. T. Mason, R. Paolini, and D. Bagnell, "A convex polynomial model for planar sliding mechanics: Theory, application, and experimental validation," *The International Journal of Robotics Research*, vol. 37, no. 2-3, pp. 249–265, 2018.

[151] J. Zhou, R. Paolini, A. M. Johnson, J. A. Bagnell, and M. T. Mason, "A probabilistic planning framework for planar grasping under uncertainty," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 2111–2118, 2017.