Extraction of Physically Plausible Support Relations to Predict and Validate Manipulation Action Effects

Rainer Kartmann*, Fabian Paus*, Markus Grotz and Tamim Asfour

***Abstract*—Reliable execution of robot manipulation actions in cluttered environments requires that the robot is able to understand relations between objects and reason about consequences of actions applied to these objects. We present an approach for extracting physically plausible support relations between objects based on visual information which does not require any prior knowledge about physical object properties, e. g. mass distribution or friction coefficients. Based on a scene representation enriched by such physically plausible support relations between objects, we derive predictions about action effects. These predictions take into account uncertainty about support relations and allow applying strategies for safe bimanual object manipulation when needed. The extraction of physically plausible support relations is evaluated both in simulation and in real world experiments using real data from a depth camera, whereas the handling of support relation uncertainties is validated on the humanoid robot ARMAR-III.**

***Index Terms*—Perception for Grasping and Manipulation, Semantic Scene Understanding, RGB-D Perception**

I. Introduction

ROBOTS operating in dynamic and cluttered environments must be able to infer a physically plausible scene representation, which allows to leverage the environment for manipulation tasks and ensure a successful action execution. In cluttered scenes, a pure geometric reasoning about the scene is not sufficient to plan and execute actions. Therefore, the robot must be able to acquire and utilize knowledge about the structure of the scene, i. e. how objects physically interact with each other, in order to generate feasible and safe action plans.

Human perception automatically combines visual features, prior knowledge, and basic physical constraints into a plausible model of the scene. The latter part is known as naive physics (see [1], [2]) since the human brain does not accurately model all physical phenomena at work. Rather, a simplified model based on prior knowledge about action-effect relations is employed when acting in the world. Furthermore, by identifying ambiguities and uncertainties in their scene understanding, humans are able to interact with the environment to verify hypotheses about objects, their relations and possible interactions with them, and to acquire new knowledge about the scene structure. Inspired by the concept of naive physics, we have investigated how a robot can extract physically plausible support relations between entities in the scene based only on geometric reasoning. Since our approach does not require any prior knowledge about the physical properties of an object, we need to explicitly take the uncertainty of support relations into account.

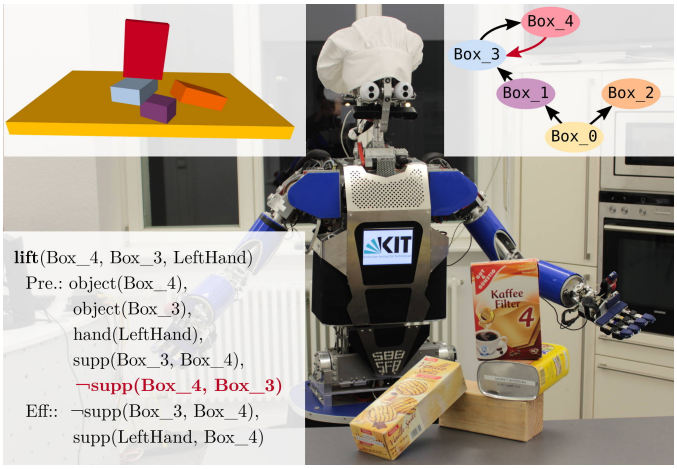


Fig. 1: The humanoid robot ARMAR-III segmented the scene based on RGB-D images. The support relations are visualized in a graph, in which uncertain edges are marked red. In order to lift the coffee filters on top of the pile, a precondition requires no top-down support.

Consider the scene in Fig. 1. The robot's task is to lift the coffee filters box on top of an object stack. The robot extracted a geometric scene representation using RGB-D images from its depth camera. Based on these 3D geometric shapes, physically plausible but not necessarily accurate support relations are extracted and represented as a directed graph, in which objects are the nodes and support relations are the edges. The edge from the top-most object to the box underneath is marked red, indicating that such support relation is uncertain. Lifting the coffee filters (`Box_4`) might cause falling of the underlying box (`Box_3`). We can predict these action effects by observing the preconditions of the lift action, which require the non-existence of a support relation between the lifted object and the box underneath (highlighted in red in Fig. 1). Instead of not executing the action, our approach employs a safe manipulation strategy using one hand to execute the action, and the second hand to secure potentially falling objects. Further, we use force measurements obtained from force-torque sensors in the robot's wrists to detect whether an object

Manuscript received: February, 24, 2018; Revised May, 25, 2018; Accepted July, 12, 2018.

This paper was recommended for publication by Editor Han Ding upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the German Research Foundation (DFG) as part of the Transregional Collaborative Research Centre Invasive Computing (SFB/TR 89) and by the European Union's Horizon 2020 Research and Innovation programme under grant agreement No. 731761 (IMAGINE).

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. {paus, asfour}@kit.edu

*The first two authors contributed equally to this work.

Digital Object Identifier (DOI): see top of this page.

fell into the second hand and update the support relations accordingly. To this end, we propose bimanual manipulation strategies to cope with the inherent uncertainty about the consequences of action executions.

There are two main contributions of this work. First, we extend the concept of support relations by uncertainty detection based on a support polygon analysis, see Section III. This allows the robot to detect potential top-down support relations. Second, we propose a bimanual manipulation strategy to cope with situations arising from an uncertain support relation, see Section IV. Depending on the interaction result, the robot validates or discards the support hypothesis. Section V shows our evaluation results. We evaluate the extracted support relations in simulated environments and on recorded RGB-D views of scenes with varying complexity. Thereby, we demonstrate how uncertainty detection improves recall without inflating precision compared to other approaches. In order to evaluate the safe manipulation strategy, we conduct experiments with the humanoid robot ARMAR-III [3].

II. Related Work

In Section II-A, we discuss semantic scene understanding approaches which extract symbolic support relations from images. Subsequently, Section II-B addresses related work about stability reasoning under gravity. We cover relevant approaches for predicting action effects on the perceptual level in Section II-C, whereas Section II-D describes the prediction of symbolic changes.

A. Semantic Scene Understanding

Scene understanding aims at describing a scene by its qualitative structure and spatial relationships [4]. In our previous work [5], we utilized a graph-based semantic scene representation based on neighborhood relations between geometric primitives to allow execution of interaction possibilities in a shared autonomous operation. In this work, however, we go beyond modeling spatial relationships to consider physical support between structures. Silberman et al. [6] rely on RGB-D images to identify support relations for indoor environments by utilizing a MAP inference approach and linear programming. The authors exploit the Manhattan world assumption for their scene understanding approach. However, this assumption is not feasible for all scenarios. Further, only one supporting object is considered in their work. The idea of a support relation between objects has been extended by Panda et al. [7], [8]. The authors identify support relations to plan an action sequence to remove objects in order to get access to a specific object. Besides neighborhood relationships, they also consider the relationship type, i.e. support from below, support from the side, or containment. While their approach relies on handcrafted rules and depends on a structure classifier, no physical principles are considered in their work.

B. Stability Reasoning under Gravity

The general idea of stability reasoning is to incorporate physics by considering stability under gravity. In [9], [10], a stability relationship analysis between objects is proposed. Using a voxel-based scene model the authors cluster volumetric primitives to physically stable objects using a stability function. Support relations can be used to improve other vision-based tasks. Jia et al. [11] utilize a derived stability relationship to improve scene segmentation based on a volumetric box representation of the scene. Similar to the work of Panda et al., the relation includes discrete types such as surface on-top, partial on-top, or side support. However, planning and safely executing action plans have not been considered so far. Furthermore, in such scenarios, the assumption of having a complete world model or sufficient information about the objects is no longer valid. Mojtahedzadeh et al. [12] have investigated the safe removal from an unstructured pile in shipping containers based on geometric object representation.

C. Perceptual Prediction of Action Effects

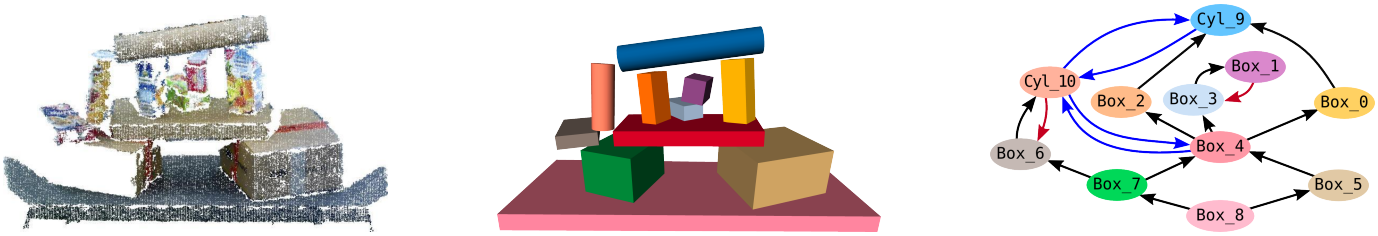
Another relevant scientific area deals with dynamics anticipation. Given an action, the goal is to predict the dynamic behavior of the scene. Fromm et al. utilize a dynamic simulation to predict the cost of an action in terms of undesired motions of inactive objects in [13]. Sophisticated physics models or dynamic simulations as used in [12] and [13] rely on prior knowledge about the physical properties of the objects, such as mass distribution and friction coefficients. However, these properties typically vary between different scenes and may not be available in unknown environments. Deep neural networks have been used to predict action effects on the perceptual level [14], [15]. Byravan et al., for example, designed a deep neural network architecture called SE3-net to segment a scene into objects and predict their rigid motion, given a 3D point cloud and an action vector [16].

D. Symbolic Prediction of Action Effects

In order to predict the symbolic effects of actions, different methods have been proposed in the literature. Kernel perceptrons can be used to learn the effects of actions [17]. They learn the difference in state after an action has been applied. Object-Action-Complexes (OACs) were proposed to model the expected change on an object as a consequence of an action ([18], [19]). The unpredictability of an action is defined as the difference between the actual change and the expected change when executing an action. In their work, they assume that actions are implemented fairly optimal, i. e. they do not introduce undesired side effects in addition to the expected change. In [20], the authors discover effect categories of actions using unsupervised clustering methods. By learning the relation between object features and the effect categories, they can categorize objects based on the generated effects of available actions.

III. Extraction of Support Relations

In this work, we abstract the environment using basic 3D geometric shapes like boxes, cylinders, and spheres extracted from segmented RGB-D data. Both the extraction of support relations and the presented manipulation strategy require an



(a) Point cloud (b) Objects (c) Support graph

Fig. 2: Shapes are extracted from a 3D point cloud captured by an RGB-D camera (a) and used as input. Given the 3D geometry and pose of each object (b), we extract binary support relations and represent them as a directed graph (c). Black edges represent certain bottom-up support, blue edges are unknown and red edges are potential top-down support relations.

object representation consisting of geometry and pose. First, we register multiple views into a single globally consistent point cloud. To speed up subsequent steps, we then segment the scene into plausible disjunct parts. To this end, we resort to state-of-the-art methods available in the point cloud library (PCL) [21]. We utilize a RANSAC model fitting approach for each segment to determine its 3D geometric shape. To address the issue of occlusion and missing data, we modified the RANSAC box fitting approach of Garcia et al. [22], so that we only need two visible faces instead of six. Fig. 2 depicts an example scene, the extracted 3D geometric shapes, and the derived support relations.

Our approach extends [12], which uses a Static Equilibrium Analysis (SEA) to determine support relations. SEA requires all objects and the range of their physical properties to be known. Since objects are extracted from RGB-D images, our method needs to handle incomplete knowledge. Furthermore, we add explicit handling of objects with almost vertical separating planes to further improve accuracy. We provide a publicly available implementation of our method[1].

A. Definition and Representation of Support Relations

Given a set of objects $\mathcal{O}$ represented by their geometric 3D shape, our goal is to extract binary support relations $\texttt{SUPP}(\cdot\,,\cdot)$ between each object pair. We follow the definition of support relation given in [12]: For two objects $\texttt{A},\texttt{B} \in \mathcal{O}$ we denote $\texttt{SUPP}(\texttt{A},\texttt{B})$ iff removing $\texttt{A}$ causes $\texttt{B}$ to lose its motionless state, i. e. $\texttt{A}$ supports $\texttt{B}$. For instance, $\texttt{B}$ falls when $\texttt{A}$ is removed. This definition incorporates physical object interactions and enables us to assess the scene's stability when certain actions are performed.

To represent support relations of a given object set $\mathcal{O}$, we define a support graph whose nodes represent objects and whose edges indicate support. Support relations are transitive, i. e. there are indirect support relations between objects without direct contact. However, it is more practical to model only direct relations explicitly and infer indirect relations as needed. Therefore, a support graph is a directed graph $\mathcal{G}_s = (V, E)$ whose transitive closure $\langle\mathcal{G}_s\rangle = (V, \vec{E})$ models the support relations with

$$V = \mathcal{O}, \quad \vec{E} = \{(\texttt{A},\texttt{B}) \in \mathcal{O} \times \mathcal{O} \mid \texttt{SUPP}(\texttt{A},\texttt{B})\} \ . \tag{1}$$

[1]https://gitlab.com/h2t/semantic-object-relations

B. Act Relation Heuristic

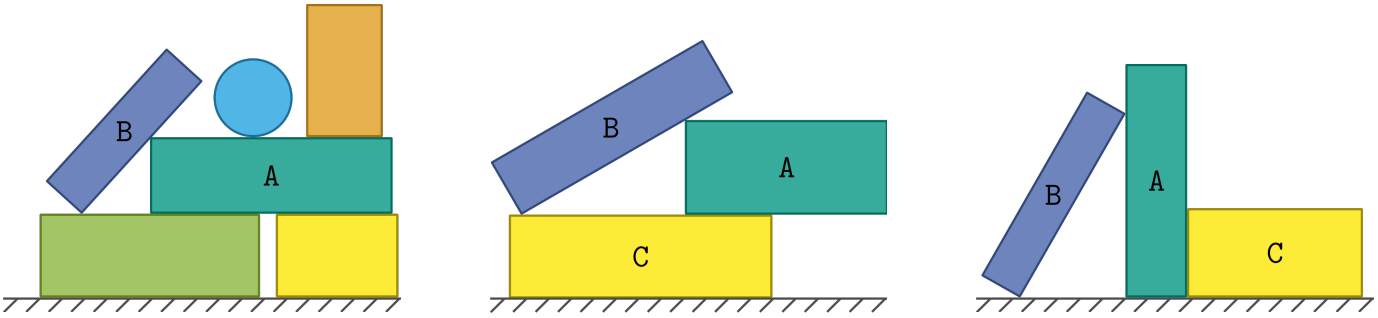
In the first step of the support analysis, act relations $\texttt{ACT}(\cdot\,,\cdot)$ as proposed by [12] are computed. Let $\texttt{A},\texttt{B} \in \mathcal{O}$ be two objects with given pose and geometry. Motivated by Newton's third law of motion, $\texttt{ACT}(\texttt{A},\texttt{B})$ indicates that, due to gravity, $\texttt{A}$ is exerting a force on $\texttt{B}$. In this case, $\texttt{A}$ is called acting and $\texttt{B}$ is called reacting. First, the contacts between $\texttt{A}$ and $\texttt{B}$ are computed. To deal with perceptual inaccuracies, we increase the size of the objects by a small margin $\delta_c \in \mathbb{R}_{\geq 0}$ for contact detection. Using the contact points and normals, the plane $\mathcal{P}_{\text{sep}}$ separating $\texttt{A}$ and $\texttt{B}$ is constructed. If $\texttt{A}$ is above $\mathcal{P}_{\text{sep}}$ and $\texttt{B}$ is below, we set $\texttt{ACT}(\texttt{A},\texttt{B})$, and vice versa if $\texttt{B}$ is above $\mathcal{P}_{\text{sep}}$.

When objects are located horizontally next to each other as shown in Fig. 3c, it might not be clear which object is acting, reacting, or whether a force is exerted at all. In these cases, the separating plane will be almost vertical. As an extension to [12], we introduce a threshold $\alpha_{\max}$. If the separating plane's rotation angle relative to the vector of gravity is below $\alpha_{\max}$, we mark the act relations between the respective objects as unknown. We will evaluate different values for $\alpha_{\max}$ in Section V.

After computing $\texttt{ACT}(\cdot\,,\cdot)$ for each pair of objects, we use it to generate our first support hypothesis by setting

$$\texttt{SUPP}(\texttt{A},\texttt{B}) \Leftrightarrow \texttt{ACT}(\texttt{B},\texttt{A}) \ . \tag{2}$$

In other words, we state that an object $\texttt{B}$ is supported by another object $\texttt{A}$ if $\texttt{B}$ acts on $\texttt{A}$. This heuristic is valid in many scenes where objects are stacked on top of each other (see Fig. 3a), thereby offering a good basis for a support hypothesis. However, there are cases of support which are not covered by act relations. For instance, consider the configuration shown in Fig. 3b. Clearly, $\texttt{B}$ acts on $\texttt{A}$, and thus $\texttt{SUPP}(\texttt{A},\texttt{B})$ according to the act heuristic. In addition, it seems likely that $\texttt{A}$ falls if $\texttt{B}$ is removed. As $\texttt{A}$ does not act on $\texttt{B}$, this possible support is not found by the act heuristic. By vision alone, one cannot know whether $\texttt{B}$ supports $\texttt{A}$. If most of $\texttt{A}$'s mass is located at its left side, resting on $\texttt{C}$, it might not need support from $\texttt{B}$ to be stable under gravity. Still, this uncertainty must be detected and taken into account when executing actions affecting a possible support between $\texttt{B}$ and $\texttt{A}$. The following section explains how we detect uncertain support relations.



(a) (b) (c)

Fig. 3: (a) Scene with solely bottom-up support. All support relations are correctly determined by the act hypothesis. (b) Scene with possible top-down support from B to A. Depending on the mass distribution of A, it may fall or not when B is removed. (c) Scene with vertical separating planes. While A supports B, there is no support between A and C.

C. Detection of Uncertain Support Relations

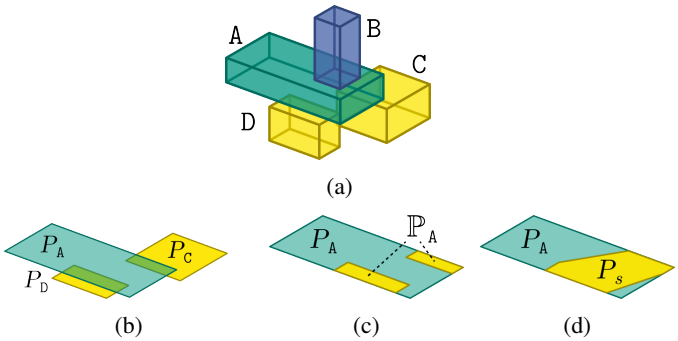
Approaches addressing top-down support typically assume a uniform mass distribution [12], [13]. We present a purely geometric approach for detecting potential top-down support. Let $\mathcal{G}_s = (V, E)$ be the support graph resulting from applying (2) to the computed act relations. Let $\mathtt{A}, \mathtt{B} \in V$ be two objects with $(\mathtt{A}, \mathtt{B}) \in E$, i. e. A supports B according to their act relation. In order to decide whether B may also support A, we consider how likely it is that A falls when B is removed. For example in Fig. 3a, A is well supported by the objects below it and will not fall when removing B. In Fig. 3b, however, A is badly supported by the object below it. Since A is at rest nonetheless, it seems likely that it is supported by B.

Thus, to decide whether A may fall when B is removed, we examine how well A is supported by the other objects. The process is visualized in Fig. 4. First, we project A to the ground plane, resulting in a 2D polygon $P_{\mathtt{A}}$. If an object has round faces or edges, it is approximated by a triangle mesh. Second, each object C supporting A is projected onto the ground plane as well, and its projection polygon $P_{\mathtt{C}}$ is intersected with $P_{\mathtt{A}}$. Then, the intersecting areas of all supporting objects are combined into the set of polygons $\mathbb{P}_{\mathtt{A}} = \{P_{\mathtt{A}} \cap P_{\mathtt{C}} \mid \mathtt{A}, \mathtt{C} \in \mathcal{O},\ \mathtt{SUPP}(\mathtt{C}, \mathtt{A})\}$ representing the directly supported area of A. The support polygon P_s is the convex hull of the polygons in $\mathbb{P}_{\mathtt{A}}$.

Finally, the support area ratio r_s is computed as the proportion of the supported area of A to its total area

$$r_s = \frac{\text{area}(P_s)}{\text{area}(P_{\mathtt{A}})} \quad . \tag{3}$$

Note that $r_s \in [0, 1]$, where $r_s = 1$ if A is fully supported, and $r_s = 0$ if A is not supported at all, i. e. A is floating. If r_s is below a threshold $r_{s,\min}$, A is considered potentially unstable. In this case, a new edge $(\mathtt{B}, \mathtt{A})$ labeled as uncertain is added to E. If r_s is above the threshold $r_{s,\min}$, A is likely well supported by the objects below it and will stay at rest when B is removed, so no edge is added. This reasoning is performed for all pairs of objects, adding edges to $\mathcal{G}_s$ where potential support is detected. The supported area ratio is a simple heuristic which does not require explicit assumptions about the mass distribution of objects. A more sophisticated model could use the supported volume ratio or estimate the object's center of gravity which requires more prior knowledge.



(a) (b) (c) (d)

Fig. 4: Example of support polygon construction. (a) The object constellation. A supports B and is supported by C and D. (b) The objects A, C and D are projected onto the ground, creating polygons $P_{\mathtt{A}}$, $P_{\mathtt{C}}$ and $P_{\mathtt{D}}$, respectively. (c) $P_{\mathtt{C}}$ and $P_{\mathtt{D}}$ are intersected with $P_{\mathtt{A}}$, constructing $\mathbb{P}_{\mathtt{A}}$. (d) The support polygon P_s is the convex hull of the polygons in $\mathbb{P}_{\mathtt{A}}$.

Uncertainty detection (abbr. UD) requires a base support extraction method (see Section III-B) to determine supporting objects used in the support polygon calculation.

IV. Handling Uncertainties in Action Effects

In [12], they generated an action plan by greedily choosing to remove the object which causes the least motion. They execute the action even if it would cause major disturbances. Our approach detects potentially unsafe actions and employs a safe bimanual manipulation strategy to cope with these situations.

A. Detecting Unsafe Actions

We consider two operators `push` and `lift`, which the robot can execute in order to manipulate a pile of objects. However, our methods can be easily extended to more sophisticated manipulation actions. Listing 1 specifies operators, their preconditions, and effects. We use the predicate `OBJECT(A)` to denote that A is an object with which the agent can interact. The predicate `HAND(H)` identifies the end-effector H used

```
push(A, B, H): Push object A on object B
               using hand H
  Precon.: OBJECT(A), OBJECT(B), HAND(H)
           SUPP(B,A), ¬SUPP(A,B)
  Effects: --

lift(A, B, H): Lift object A from object
               B using hand H
  Precon.: OBJECT(A), OBJECT(B), HAND(H),
           SUPP(B,A), ¬SUPP(A,B)
  Effects: ¬SUPP(B,A), SUPP(H,A)
```

Listing 1: Definition of operators, i. e. actions which the robot can execute, including their preconditions (`Precon.`) and effects (`Effects`) using STRIPS notation.

to execute the action, and `SUPP(A,B)` requires a support relation between objects `A` and `B`.

Uncertainties in the extraction of support relations might lead to the execution of actions, which violate the defined preconditions or have additional undesired effects on the world state. We focus on the undesired effects on the support graph, i. e. pushing or lifting an object causes other objects to change their support relations. We use the previously described uncertainty detection method (UD) to identify potential top-down support relations. If the action contains a precondition that relies on the non-existence of an uncertain support relation involving the object to be manipulated, we can still execute the action using a safer bimanual manipulation strategy. Consider the case of lifting `B` in Fig. 3b. UD adds an uncertain support relation $\texttt{SUPP}(\texttt{B},\texttt{A})$ which may cause the undesired effect of `A` falling. Using the second hand of the humanoid robot, we can prevent these undesired changes in the scene structure.

B. Safe Manipulation Strategy

Given an object set $\mathcal{O}$, a support graph, an action a on a target object $\texttt{T} \in \mathcal{O}$ and an object $\texttt{S} \in \mathcal{O}$ with uncertain support relation $\texttt{SUPP}(\texttt{T},\texttt{S})$, we want to execute action a, prevent any undesired effects caused by the existence of $\texttt{SUPP}(\texttt{T},\texttt{S})$ and detect whether $\texttt{SUPP}(\texttt{T},\texttt{S})$ was true in the initial scene.

We solve this problem by using one hand $H_\texttt{T}$ to execute the primary action on the target object `T`, and the other hand $H_\texttt{S}$ to secure the supporting object `S`. If `S` would fall after or during the action execution, instead of falling unpredictably, a new support edge $\texttt{SUPP}(H_\texttt{S},\texttt{S})$ from the securing hand $H_\texttt{S}$ to the supporting object `S` is added. Using force-torque sensors in the hand of the robot we can decide whether $\texttt{SUPP}(H_\texttt{S},\texttt{S})$ needs to be added to the graph.

Algorithm 1 shows the implementation of the safe manipulation strategy. First, we decide which hand executes the action and which hand secures the supporting object. The action hand is chosen by a simple heuristic. If the target object's position is to the right of the robot's base, we choose the right hand and vice versa. Then, we can calculate the secure and target poses $\mathbf{p}_\texttt{S}$ and $\mathbf{p}_\texttt{T}$ needed to execute the action safely.

Algorithm 1: Safe Manipulation Strategy

Input: H_R, H_L: Right and left end-effector
$\texttt{T},\texttt{S}$: Target and supporting object
a: Action to be executed
$F_{\max}$: Force threshold for fall detection
$\mathcal{G}_s = (\mathcal{O}, E)$: Estimated support graph
Output: $\mathcal{G}_{\text{new}}$: Resulting support graph
$\mathcal{G}_{\text{corr}}$: Corrected initial support graph

$H_\texttt{T}, H_\texttt{S} = \text{ChooseHands}(H_R, H_L, \texttt{T}, \texttt{S})$;
$\mathbf{p}_\texttt{S} = \text{CalculateSecurePose}(H_\texttt{S}, \texttt{S})$;
$\mathbf{p}_\texttt{T} = \text{CalculateTargetPose}(a, H_\texttt{T}, \texttt{T})$;
$\mathbf{p}_{\text{pre},\texttt{T}} = \text{CalculatePrepose}(a, H_\texttt{T}, \mathbf{p}_\texttt{T})$;
$\mathbf{q}_{\text{Body}} = \text{SolveBimanualIK}(H_\texttt{T}, \mathbf{p}_{\text{pre},\texttt{T}}, H_\texttt{S}, \mathbf{p}_\texttt{S})$;
$\text{MoveBody}(\mathbf{q}_{\text{Body}})$;
$FT = \text{CreateFilteredForceTorqueSensor}(H_\texttt{S})$;
$\text{ExecuteAction}(a, \mathbf{p}_\texttt{T})$;
$E_{\text{new}} = E_{\text{corr}} = E$;
if $FT.\text{MaxForceValue}.z > F_{\max}$ **then**
  $E_{\text{new}} = E \cup \{(H_\texttt{S}, \texttt{S})\}$;
else
  $E_{\text{corr}} = E \backslash \{(\texttt{T}, \texttt{S})\}$;
end
return $((\mathcal{O}, E_{\text{new}}), (\mathcal{O}, E_{\text{corr}}))$;

Before executing the action, the robot should move its end-effector to a suitable prepose $\mathbf{p}_{\text{pre},\texttt{T}}$. We solve the IK for the kinematic chain consisting of both arms and a hip joint, to allow bimanual manipulation. Using the resulting joint values $\mathbf{q}_{\text{Body}}$, the robot moves its end-effectors to the secure pose and action prepose. Before we execute the desired action, the force-torque sensor of the supporting hand is started and a median derivative filter is applied to detect changes more easily. If the filtered force value in the z-direction (down) exceeds the predefined force threshold $F_{\max}$, the unsafe support edge existed and undesired side effects were prevented by adding $\texttt{SUPP}(H_\texttt{S},\texttt{S})$. Otherwise, the edge $\texttt{SUPP}(\texttt{T},\texttt{S})$ did not exist in the initial scene and can be removed from the corrected graph.

In order to compensate for noise and drift in the force-torque sensors of our robot, we use a median derivative filter to detect the fall of an object onto the securing hand. Let $F_w(k)$ be the latest $w \in \mathbb{N}$ force sensor values at time step $k \in \mathbb{N}$:

$$F_w(k) = \big\{f(k), f(k-1), \ldots, f(k-w+1)\big\}, \tag{4}$$

where $f(k) \in \mathbb{R}$ is the reported force value along the direction of gravity at time step k. Let $\text{median}(F)$ return the median of a finite set $F \subset \mathbb{R}$. Then, the filtered value $\hat{f}(k) \in \mathbb{R}$ at time step $k \in \mathbb{N}$ is determined as follows:

$$\hat{f}(k) = \text{median}(F_v(k)) - \text{median}(F_w(k)) \tag{5}$$

where $v, w \in \mathbb{N}$ are window sizes with $w \gg v$. The filtered value will be close to zero if the applied force does not change. However, a caught object will result in a significant peak (for an example see Fig. 5).

V. Evaluation

We evaluated both the extraction of support relations as well as the handling of uncertainties in action effects. Simulation

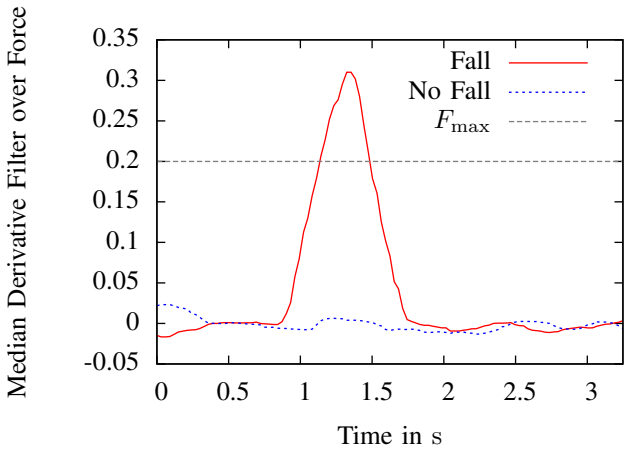


Fig. 5: Filtered force values during a push action. A fallen object produces a clear peak, while the filtered force values stay around zero when the object remains on the pile.

and real-world experiments were conducted on the humanoid robot ARMAR-III [3] using the ArmarX [23] robotic framework. A video showing our experiments is attached to this paper and available online[2].

A. Extraction of Support Relations

We evaluate our extraction methods on multiple scenes containing piles of objects on a table. The RGB-D images for the real world scenes were recorded using an ASUS Xtion Pro. For the simulation, a depth camera was simulated. All support hypotheses $\mathcal{G}_{\mathrm{Hyp}} = (\mathcal{O}_{\mathrm{Hyp}}, E_{\mathrm{Hyp}})$ were constructed using a contact margin $\delta_c = 10\,\mathrm{mm}$ and a threshold for the support area ratio $r_{s,\min} = 70\,\%$. We used three base strategies which can be combined with uncertainty detection:

$\alpha 0$: No separating plane angle threshold ($\alpha_{\max} = 0°$)
$\alpha 10$N: Within threshold, assume no support ($\alpha_{\max} = 10°$)
$\alpha 10$S: Within threshold, assume support ($\alpha_{\max} = 10°$)

For each scene, we created a ground truth support graph $\mathcal{G}_{\mathrm{GT}} = (\mathcal{O}_{\mathrm{GT}}, E_{\mathrm{GT}})$. In the simulation, we determined support relations of each object by registering what other objects move due to the removal of the inspected object. For the real scenes, we annotated the support relations by hand. All ground truth graphs were transitively reduced to remove redundant edges. We manually matched objects in $\mathcal{O}_{\mathrm{Hyp}}$ and $\mathcal{O}_{\mathrm{GT}}$, since we are only interested in the extracted support edges. Table I shows precision and recall for all scenes and each of the four strategies calculated as

$$\mathrm{Prec} = \frac{|E_{\mathrm{GT}} \cap E_{\mathrm{Hyp}}|}{|E_{\mathrm{Hyp}}|}, \quad \mathrm{Rec} = \frac{|E_{\mathrm{GT}} \cap E_{\mathrm{Hyp}}|}{|E_{\mathrm{GT}}|} \quad . \tag{6}$$

Selected evaluation scenes are presented in Table II. The $\alpha 10$N strategy is a strict improvement over $\alpha 0$, demonstrating the usefulness of the threshold $\alpha_{\max}$. Almost all support edges detected by $\alpha 10$N are correct. However, it misses more edges than the other strategies resulting in the worst recall. $\alpha 10$S produces a higher recall than $\alpha 10$N due to adding unknown edges to the support hypothesis. Yet, precision is reduced considerably since not all of the added edges are correct. Adding UD improves recall without negatively affecting precision in scenes containing top-down support (e. g. S4, R2, R7). As can be expected, $\alpha 10$S+UD adds the most edges out of all strategies and therefore achieves the best recall. It also adds the most false positives, resulting in the worst precision. Overall, $\alpha 10$S is too conservative and adds too many edges. $\alpha 10$N+UD achieves high precision and recall values by adding only potential top-down support edges. It offers the best compromise between correctness and completeness of the support hypothesis. Therefore, we propose that the $\alpha 10$N+UD strategy is most suitable for scene understanding in cluttered environments. We will also see its benefits for robotic manipulation in Section V-B.

Further, we noticed that due to visual occlusions some of the extracted objects were too small. The support hypothesis was affected in two ways:

1) The extracted objects were not in contact, hence no support was found. For example, this is the case in scene R8, where `Box_8` and `Box_15` seem to float, and `Box_13` is not supported by `Box_10`.
2) Some objects were considered to be badly supported because there seemed to be no object below them. Consequently, UD generated wrong support edges. This effect can be observed in scene R8, where `Box_5` is possibly supported by `Box_2` and `Box_3` according to the UD strategy.

If the input point cloud was more complete and the objects' extents were more accurately estimated, these errors could be reduced.

B. Handling Uncertainties in Action Effects

In order to validate the handling of uncertainties, we conducted experiments on the humanoid robot ARMAR-III. We purposefully created scenes that contained uncertain top-down support relations to trigger the safe manipulation strategy. The robot was given the task of either lifting or pushing an object which potentially supports another object underneath it. We chose the same parameters for the support extraction as in Section V-A, configured the force-torque filter with $v = 11$, $w = 101$ (corresponding to time spans of about $300\,\mathrm{ms}$ and $3000\,\mathrm{ms}$ in our setup) and set the force detection threshold to $F_{\max} = 0.2$.

In the first scenario, the task was to lift the coffee filters on top of the yellow container (see Fig. 1). We shifted weights inside the container to alter the mass distribution creating both cases of existing and non-existing top-down support. The second scenario involved a push action where we changed the positioning of the supporting blue cereal box to provoke both support cases. The robot was able to prevent the fall of the bottom object in the case of real top-down support and detect the fall of the object using its force-torque sensors (see Fig. 5). In case of no top-down support, we detected that the object did not fall on the robot's hand.

[2]https://youtu.be/iEw-mDmnRGE

Strategy		**S1**	**S2**	**S3**	**S4**	**R1**	**R2**	**R3**	**R4**	**R5**	**R6**	**R7**	**R8**	**VL**	**VP**	**Mean**
α0	Prec	1.00	0.71	0.71	0.80	1.00	0.78	0.67	0.86	0.86	1.00	1.00	0.75	1.00	0.86	0.86
	Rec	1.00	1.00	1.00	0.89	1.00	0.78	0.67	0.86	0.86	1.00	0.75	0.75	0.83	0.75	0.87
α10N	Prec	1.00	0.83	1.00	0.89	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.94	1.00	1.00	0.98
	Rec	1.00	1.00	1.00	0.89	1.00	0.78	0.67	0.86	0.86	1.00	0.75	0.75	0.83	0.75	0.87
α10S	Prec	1.00	0.63	0.56	0.73	1.00	0.73	0.75	0.93	0.81	1.00	1.00	0.58	1.00	0.88	0.83
	Rec	1.00	1.00	1.00	0.89	1.00	0.89	1.00	0.93	0.93	1.00	0.75	0.75	0.83	0.88	0.92
α0+UD	Prec	1.00	0.71	0.71	0.75	1.00	0.80	0.67	0.81	0.82	0.75	1.00	0.67	1.00	0.88	0.83
	Rec	1.00	1.00	1.00	1.00	1.00	0.89	0.67	0.93	1.00	1.00	1.00	0.80	1.00	0.88	0.94
α10N+UD	Prec	1.00	0.83	1.00	0.90	1.00	1.00	1.00	0.76	0.93	0.75	1.00	0.80	1.00	1.00	0.93
	Rec	1.00	1.00	1.00	1.00	1.00	0.89	0.67	0.93	0.93	1.00	1.00	0.80	1.00	0.88	0.93
α10S+UD	Prec	1.00	0.63	0.56	0.75	1.00	0.75	0.75	0.74	0.78	0.75	1.00	0.53	1.00	0.89	0.79
	Rec	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.80	1.00	1.00	0.99

TABLE I: Precision (Prec) and recall (Rec) of extracted support relations for all scenes and strategies. S1 – S4 are simulated and R1 – R8 are real scenes. VL and VP are our validation scenarios (for lifting and pushing, respectively).

VI. Conclusion

We presented an approach for the extraction of physically plausible support relations between objects in cluttered scenes. Our approach allows detecting possible top-down support relations, which need to be carefully handled during action execution. To this end, we proposed a safe manipulation strategy utilizing both arms of a humanoid robot to simultaneously execute the given task and secure potentially falling objects. We showed that physically plausible scene understanding is possible without complete modeling of the scene dynamics and physical object properties. With geometric reasoning based on naive physics, a robot can plan safe manipulation actions. The scenes we have considered include piles of objects on top of a table. In future work, we will extend the work towards a wider variety of scenes focusing on cluttered and unstructured environments and apply the approach to whole-body loco-manipulation actions.

References

[1] P. J. Hayes, *The naive physics manifesto*. Université de Genève, Institut pour les études sémantiques et cognitives, 1978.

[2] S. Vosniadou, "On the nature of naive physics," *Reconsidering conceptual change: Issues in theory and practice*, pp. 61–76, 2002.

[3] T. Asfour, K. Regenstein, P. Azad, J. Schroder, A. Bierbaum, N. Vahrenkamp, and R. Dillmann, "ARMAR-III: An integrated humanoid platform for sensory-motor control," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2006, pp. 169–175.

[4] B. Rosman and S. Ramamoorthy, "Learning spatial relationships between objects," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1328–1342, 2011.

[5] M. Grotz, P. Kaiser, E. E. Aksoy, F. Paus, and T. Asfour, "Graph-based visual semantic perception for humanoid robots," in *IEEE-RAS International Conference on Humanoid Robots (Humanoids)*, 2017, pp. 869–875.

[6] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *European Conference on Computer Vision*. Springer, 2012, pp. 746–760.

[7] S. Panda, A. H. A. Hafez, and C. V. Jawahar, "Learning support order for manipulation in clutter," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 809–815.

[8] S. Panda, A. H. Abdul Hafez, and C. V. Jawahar, "Single and multiple view support order prediction in clutter for manipulation," *Journal of Intelligent & Robotic Systems*, vol. 83, no. 2, pp. 179–203, 2016.

[9] B. Zheng, Y. Zhao, J. Yu, K. Ikeuchi, and S.-C. Zhu, "Scene understanding by reasoning stability and safety," *International Journal of Computer Vision*, vol. 112, no. 2, pp. 221–238, 2015.

[10] B. Zheng, Y. Zhao, J. C. Yu, K. Ikeuchi, and S.-C. Zhu, "Beyond point clouds: Scene understanding by reasoning geometry and physics," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3127–3134.

[11] Z. Jia, A. C. Gallagher, A. Saxena, and T. Chen, "3D reasoning from blocks to stability," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 5, pp. 905–918, 2015.

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

[13] T. Fromm and A. Birk, "Physics-based damage-aware manipulation strategy planning using scene dynamics anticipation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 915–922.

[14] P. Agrawal, A. V. Nair, P. Abbeel, J. Malik, and S. Levine, "Learning to poke by poking: Experiential learning of intuitive 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. 5074–5082.

[15] A. Eitel, N. Hauff, and W. Burgard, "Learning to singulate objects using a push proposal network," in *Proc. of the International Symposium on Robotics Research (ISRR)*, Puerto Varas, Chile, 2017.

[16] A. Byravan and D. Fox, "SE3-nets: Learning rigid body motion using deep neural networks," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 173–180.

[17] K. Mourao, R. P. Petrick, and M. Steedman, "Using kernel perceptrons to learn action effects for planning," in *International Conference on Cognitive Systems (CogSys)*, 2008, pp. 45–50.

[18] F. Wörgötter, A. Agostini, N. Krüger, N. Shylo, and B. Porr, "Cognitive agents — a procedural perspective relying on the predictability of object-action-complexes (OACs)," *Robotics and Autonomus Systems*, vol. 57, no. 4, pp. 420–432, 2009.

[19] N. Krüger, C. Geib, J. Piater, R. Petrick, M. Steedman, F. Wörgötter, A. Ude, T. Asfour, D. Kraft, D. Omrčen, A. Agostini, and R. Dillmann, "Object-action complexes: Grounded abstractions of sensorimotor processes," *Robotics and Autonomous Systems*, vol. 59, pp. 740–757, 2011.

[20] E. Ugur and J. Piater, "Bottom-up learning of object categories, action effects and logical rules: From continuous manipulative exploration to symbolic planning," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 2627–2633.

[21] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2011, pp. 1–4.

[22] S. Garcìa, "Fitting primitive shapes to point clouds for robotic grasping," Master of Science Thesis, Royal Institute of Technology, Sweden, 2009.

[23] N. Vahrenkamp, M. Wächter, M. Kröhnert, K. Welke, and T. Asfour, "The robot software framework ArmarX," *Information Technology*, vol. 57, no. 2, pp. 99–111, 2015.

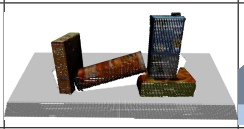
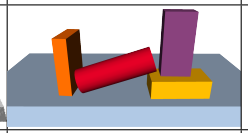
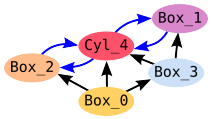
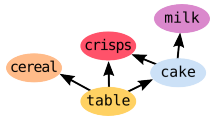
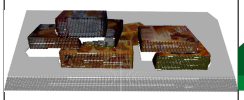
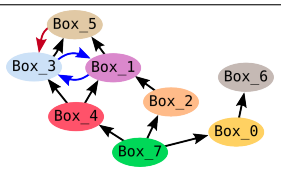
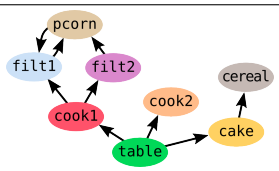
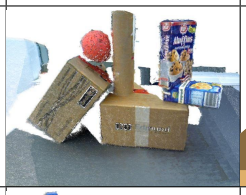
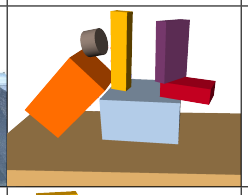
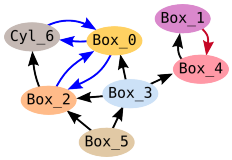
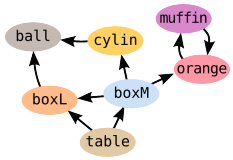
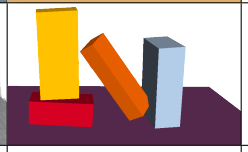
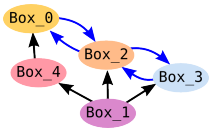
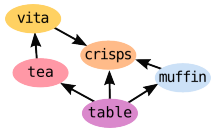
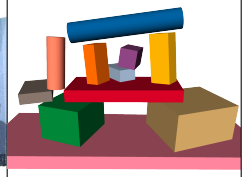
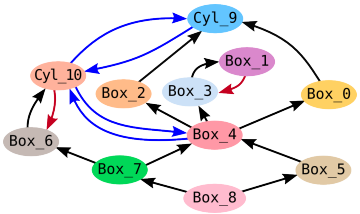
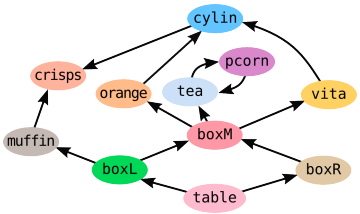
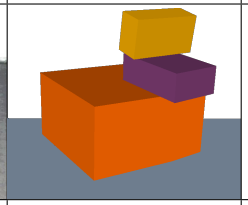
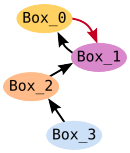
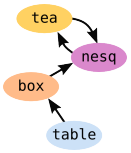
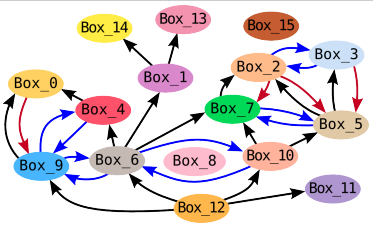
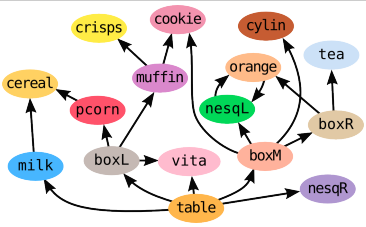
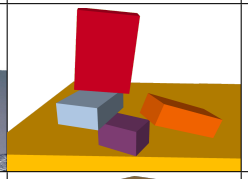
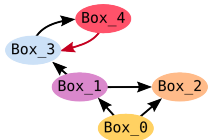
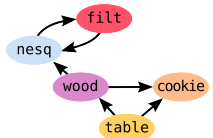
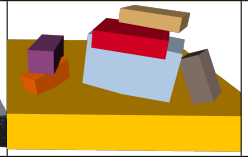
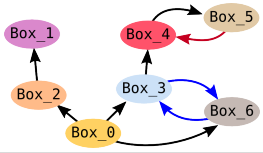
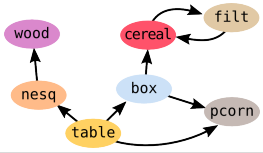
Scene	Point Cloud	Extracted Objects	Support Hypothesis	Support Ground Truth
S3				
S4				
R2				
R3				
R5				
R7				
R8				
VL				
VP				

TABLE II: Selected evaluation scenes. We show the recorded point cloud, the extracted objects, our support hypotheses and the ground truth support graph. Support hypotheses for all four strategies are visualized by color coding differing edges. Black edges were generated by the α10N strategy and are also part of the three other strategies. Blue edges are part of the graph if we use α10S, while red edges are added by UD. The scenes not shown in this table are slight variations of their neighbors, e. g. R1 is a simpler version of R2 without the unknown and uncertain edges.