2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

Automatic Relational Scene Representation For Safe Robotic Manipulation Tasks

Rasoul Mojtahedzadeh, Abdelbaki Bouguerra and Achim J. Lilienthal
Center of Applied Autonomous Sensor Systems (AASS), Örebro University, Sweden

Abstract— **In this paper, we propose a new approach for automatically building symbolic relational descriptions of static configurations of objects to be manipulated by a robotic system. The main goal of our work is to provide advanced cognitive abilities for such robotic systems to make them more aware of the outcome of their actions. We describe how such symbolic relations are automatically extracted for configurations of box-shaped objects using notions from geometry and static equilibrium in classical mechanics. We also present extensive simulation results as well as some real-world experiments aimed at verifying the output of the proposed approach.**

I. Introduction

In robotic manipulation tasks, it is desirable to pick up and move objects safely to their target position. Essential for safe robotic manipulation is the ability to predict and reason about the effects of possible manipulation actions. In industrial automation the need to predict the effects of actions on the fly is avoided with known, and well engineered environments. However, in non-engineered environments, it is a challenge to reliably predict the outcomes of robot actions. One example is dealing with shipping containers, which are filled with loose, unpredictably stacked and only partially known goods. Fig. 1 shows two such difficult configurations. Planning for a safe sequence of unloading actions to empty the container requires analyzing the relations between the objects in contact with each other.

(a)

(b)

Fig. 1: Two example snapshots of configurations of objects inside shipping containers at unloading sites.

In artificial intelligence (AI) planning community, it is usually assumed that symbolic representations of the planning problem are available (usually encoded by an expert). We propose a method to relax such an assumption for a specific class of objects. The key idea is to develop an automatic cognitive process to enable robotic systems to construct symbolic relational representations of static configuration of box-shaped objects. Since the relations between objects in a scene are independent of the manipulators constraints, the method is independent of the type of manipulator in use. The proposed scene representation captures *support* and *action* (*reaction*) relations between objects (e.g. object A supports object B), and therefore is useful for high level AI symbolic reasoning tools such as action planning.

The approach we propose in this paper comprises two major steps. The first step is a geometric analysis aimed at extracting act-react relations between those objects that are in contact with each other. Since the configurations are assumed to be static, the act-react relations are identified solely based on analyzing how the weight of one object might act (push) on another object. In the second step, a more elaborate analysis based on static equilibrium conditions is performed. The idea is to check if removing one object in contact with another can result in a violation of the static equilibrium of the latter. To do this, we model the problem as a non-linear optimization program that contains among its constraints the static equilibrium conditions as well as constraints about possible unknown friction parameters.

This work is part of a larger research effort aiming at making the process of unloading cargo containers autonomous and with more advanced cognitive abilities[1]. We emphasize that we do not address the problem of object detection and pose estimation, and therefore, we make the assumption that information about the objects composing the scene is available to our algorithm as a set of geometrical attributes (size and pose). Such data, for example, can be obtained by utilizing object detection and pose estimation algorithms. Moreover, we only consider rigid box-shaped objects (i.e carton boxes), which are among the most popular packagings for shipment of goods in containers [1].

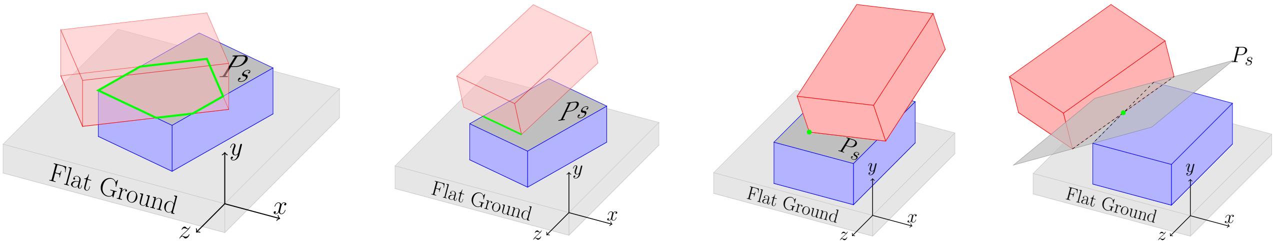
The rest of the paper is organized as follows. Related work is discussed in Section II. Section III outlines the different steps used to extract relational representation between blocks. Section IV presents experimental results of applying the proposed method on simulated and real scene data. Section V concludes the paper.

II. Related Work

Existing related research falls mostly in the category of robotic "bin-picking", where the goal is to localize and pick

[1] `http://www.roblog.eu`

978-1-4673-6357-0/13/$31.00 ©2013 IEEE



(a) Face-On-Face (b) Edge-On-Face (c) Vertex-On-Face (d) Edge-Cross-Edge

Fig. 2: Illustration of contact types and the corresponding separating planes (P_s) between two blocks.

objects from a bin using 3D range and visual perception. While some researchers have focused on developing object recognition and pose estimation algorithms for bin-picking systems [2]–[4], some research has been carried out to address and improve grasping in robotic systems [5]–[7]. In comparison with the types of objects inside shipping containers, which are usually small to large size carton boxes, the bin objects are normally small to medium size rigid mechanical parts to be assembled. A small amount of displacement of such parts due to picking up one from the bin is usually tolerable and non-harmful to the parts. In contrast, a displacement due to unloading an object from a container may result in some adjacent objects falling down. A robotic unloading system should have a cognitive level to analyze the configuration of objects and reasonably plan for a safe sequence of objects to be unloaded.

Other works address building spatial relational descriptions of scenes, where the focus is more on where objects are located with respect to each other (e.g., object `A` is north of object `B`) [8]–[10]. Such primitive symbolic descriptors are originally developed to describe relations between entities (e.g. buildings) in GIS (Geographic Information System) maps, and therefore cannot be used to predict the interaction between objects due to laws of Physics.

III. Relational Scene Representation

To explain the relational scene representation, we first present a set of definitions and assumptions that are used throughout the paper.

- **Block** is a set of 3D points in $\mathbb{R}^3$ inside a rectangular cuboid. We assume that the center of mass of the block and the geometric centroid of the cuboid coincide. We also assume that blocks are rigid and non-deformable, i.e., they cannot penetrate into each other.
- **Flat ground** is a large block that cannot be moved and on which other blocks can sit. Here, we assume that the gravity force is perpendicular to the flat ground. In practice this can be, for example, the floor of the container.
- **Configuration** is a set of blocks and the flat ground. In a *static configuation*, all the blocks are motionless.
- **Reference frame,** without loss of generality, is a fixed frame with xz-plane representing the side of the flat ground facing up, i.e. the gravity force direction is opposite to that of y-axis.

A. Contact Point-Set Network

Our approach to relational representation of a static configuration of blocks relies on extracting contact points between the blocks. This is motivated by the fact that in a static configuration (where the earth gravity is the only force acting on objects, and the gravitational forces between objects are negligible) the points of action of the weight forces between objects, and consequently the corresponding torques are determined by the contact points and the mass distribution of the objects.

The contact points are used to build a graph structure, where vertices are the blocks and edges represent the set of points at contact between the blocks. We call such graph *contact point-set network* (CPSN). We consider four major types of geometrically possible contacts between two blocks and compute them in the following order:

1) Face-On-Face. This type of contact happens when "a face of one block and a face of another block partly or completely coincide". The result is a polygonal area with minimum 3 and maximum 8 vertices (see Fig. 2a).
2) Edge-On-Face. This happens when "an edge of one block partly or completely touches a face of another block". The result is a line segment (see Fig. 2b).
3) Vertex-On-Face. This happens when "a vertex of one block touches a face of another block". The result is a single point (see Fig. 2c).
4) Edge-Cross-Edge. This happens when "an edge of one block intersects with, but is not parallel to an edge of another block". The result is a single point (see Fig. 2d)

Since we consider static configurations, we do not consider unstable contacts such as "a vertex of one block touches a vertex of another block".

B. Geometrical Analysis

The objective at this stage is to identify and label blocks in contact as *acting* (or *reacting*) according to the geometric configuration between two blocks. From Newton's third law of motion, we know for two objects `A` and `B` in contact, that if object `A` exerts a force on `B`, then `B` exerts a force, which is equal in magnitude but opposite in direction on `A`; we call `A` "*acting object*" and `B` "*reacting object*".

Our geometrical analysis is based on extracting the separating plane between two blocks in contact. Since a block is a convex set, for each pair of blocks in contact, according to *hyperplane separation* and *supporting hyperplane* theorems [11], there exists a separating plane, which divides 3D space into two half-spaces such that the separating plane contains (and is identified by) the contact points and each half-space contains only one of the blocks. Fig. 2 shows separating planes for the discussed contact types.

In our analysis, we label the two half-spaces as *positive and negative sides* of the separating plane. A half-space is labeled as positive, resp. negative, side, if the y-component of the separating plane's normal vector at that side is strictly positive, resp. negative. In the case of a perpendicular separating plane to the xz-plane (i.e., the y-component of the normal vector is zero), the half-spaces are not labeled.

To identify which block acts on another, we first ignore all the other blocks in contact with the two blocks in question. The acting block and the reacting block is determined according to the following proposition.

Proposition 1: For two blocks **A** and **B** in contact, if their separating plane is not perpendicular to the flat ground (i.e., xz-plane), then the positive side of the separating plane contains the acting block, and the negative side contains the reacting block. We represent such a symbolic relation as `ACT(A,B)` which is read as "**A** acts on **B**". For a sketch of the proof of the proposition, see the appendix.

There are many situations where the extracted `ACT` relations are not enough to decide which object can be removed and not cause the other objects to fall. An illustration of such a case is shown in Fig. 3, where block `A` is the only one not reacting to any other block. Such information can naively used to suggest removing `A` first. However, the weight of `B` produces a torque about the contact axis (`B`,`D`) which is canceled by the action force that `A` exerts on `B` at contact point(`A`,`B`). This means that removing `A` causes `B` to fall down. Besides the case where the geometrical reasoning cannot be applied (i.e., the separating plane is perpendicular to the flat earth), this example suggests that interaction between more than two objects requires a more complex analysis, which we introduce in the next section.

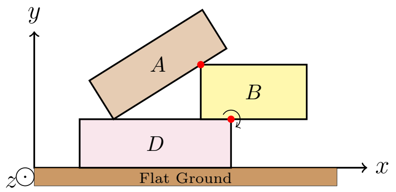


Fig. 3: An example configuration where geometric analysis is not enough to predict the effect of removing block `A`.

C. Static Equilibrium Analysis

We use static equilibrium conditions to anticipate the effect of removing a block from a static configuration. From classical mechanics [12], an object is in static equilibrium if and only if:

- The vector sum of all external forces is zero.
- The vector sum of all torques (due to the external forces) about any pivot point is zero.
- The linear momentum of the object is zero.

In the context of this paper, if removing block `A` from a static configuration leads to unsatisfied static equilibrium conditions for another block `B` in contact with `A`, then `A` is said to *support* `B`. This relation is symbolically denoted by `SUPP(A,B)`. It is important to note that two blocks in contact can both support each other, i.e., it is possible to extract two relations such as `SUPP(A,B)` and `SUPP(B,A)` (see objects `A` and `B` in example configuration in Fig. 3).

In order to determine the static equilibrium of blocks, values of the necessary physical quantities such as masses of objects and their distribution, as well as the friction factors are required to be known. However, in our case such physical quantities are not available. Moreover, in 3D space, where the objects are not mathematically idealized points, configurations of objects often represent a statically indeterminate mechanical system, i.e., a system in which the number of unknown forces is greater than the number of independent equations. Static equilibrium equations are often insufficient to determine unknown forces acting on a block, even if the physical quantities are known.

Since we are interested in symbolic relations between objects and not the exact numeric computation of unknown forces, we model static equilibrium analysis of a target block as an optimization problem, where the goal is to find a feasible solution that satisfies a set of predefined constraints. First, we define the problem formally as follows,

Problem. Given a target block X, a set of blocks $\mathbf{Y} = \{Y_1, \ldots, Y_p, Z\}$ in contact with X, the corresponding CPSN of X, the separating planes between X and Y_i, and the set of `ACT` relations identified from geometrical reasoning on X with respect to Y_i, determine if X is in static equilibrium after removing Z from $\mathbf{Y}$.

We assume a unit mass for the target block, and define a *force-group* to be a set of forces that a block exerts on another block at their contact points. Each force-group contains either one, two, or between 3 to 8 forces depending on the contact type (see section III-A). Each force in a force-group is attached to one point that is called a force-point. A force-point can be the point of a single-point contact, or the end point of a line-segment contact, or one of the vertices of a polygon contact (see Fig. 4). The forces in a force-group are assumed to have the same direction, but their magnitudes may differ.

For each block Y_i in $\mathbf{Y}$, depending on its `ACT` relation with X, three possibilities are considered:

a) X acts on Y_i: Here, for each force-point j, the corresponding force $\vec{F}_{i,j} \in G_i$, is resolved into two components $< \vec{F}^n_{i,j}, \vec{F}^t_{i,j} >$: a perpendicular force to, and a tangential (friction) force in the separating plane respectively. We limit the amount of friction by introducing a friction factor, $\mu_{i,j}$, such that $\|\vec{F}^t_{i,j}\| \leq \mu_{i,j}\|\vec{F}^n_{i,j}\|$ and $\mu_{i,j} \leq \mu_{\text{MAX}}$. The value of μ_{MAX} is specified as a constraint. The direction of all the tangential forces is determined by an angle parameter, α_i. To summarize, one optimization parameter for the angle of friction forces of G_i and three parameters (magnitude of normal force, magnitude of friction force, and the friction factor) for each point j in G_i are defined.

b) Y_i acts on X: In this case, for each force-point j in the force-group, G_i, we assign a weight, $\vec{W}_{i,j} =$

$(0, -w_{i,j}, 0)$ where $w_{i,j} > 0$. Thus, for each $\vec{W}_{i,j}$ one optimization parameter is defined.

c) The ACT *relation between X and Y_i is not given:* We neglect friction and consider solely normal forces that Y_i may exert on X. Thus, for each force-point j in the contact force-group, G_i, one optimization parameter is defined.

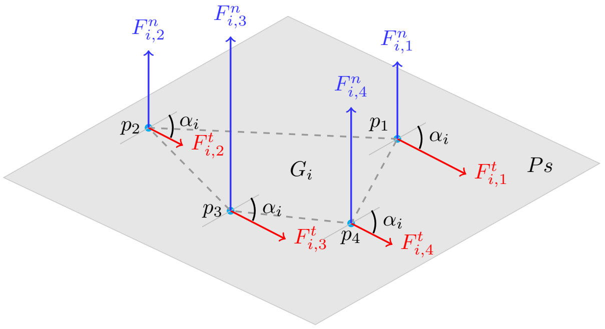


Fig. 4: A force-group G_i with 4 contact force-points p_j (polygon contact type), friction forces F^ti, j, normal forces $F^n_{i,j}$ and angle α_i.

For the sake of modeling, the weight of block X is represented as the force-group G_0 with a single force, $\vec{W}_{0,1} = (0, -g, 0)$, at the centroid of X where $g > 0$ is the magnitude of the earth's gravity. Mathematically, the static equilibrium conditions for block X can be written as below,

$$\vec{F}_{\text{total}} = \sum_{i=0}^{p} \left(\sum_{j=1}^{q_i} \vec{F}_{i,j} \right) = \mathbf{0}$$

$$\vec{\tau}_{\text{total}} = \sum_{i=0}^{p} \left(\sum_{j=1}^{q_i} \vec{r}_{i,j} \times \vec{F}_{i,j} \right) = \mathbf{0}$$

where p is the number of force-groups (including the weight of X), q_i is the number of force-points in the i-th force-group, $\vec{F}_{i,j}$ is the j-th force in the i-th force-group, $\vec{r}_{i,j}$ is the moment arm from the centroid of X to the point of action of $\vec{F}_{i,j}$, and consequently $\vec{r}_{i,j} \times \vec{F}_{i,j}$ is the torque about the centroid of X due to the external force $\vec{F}_{i,j}$.

We define the objective function of the optimization problem to be the sum of the absolute values of the total force and total torque components along the reference frame axes:

$$f_{\text{obj}}(.) = |F_x| + |F_y| + |F_z| + |\tau_x| + |\tau_y| + |\tau_z|$$

and formally define the optimization problem as

$$\begin{aligned} \text{minimize} \quad & f_{\text{obj}} \\ \text{subject to} \quad & 0 < \mu_{i,j} \leq \mu_{\max} \\ & \|\vec{F}^t_{i,j}\| \leq \mu_{i,j} \|\vec{F}^n_{i,j}\| \\ & 0 \leq \alpha_i \leq 2\pi \\ & F_x = 0, F_y = 0, F_z = 0 \\ & \tau_x = 0, \tau_y = 0, \tau_z = 0 \end{aligned}$$

where $i = 1, \ldots, p$; $j = 1, \ldots, q_i$, and

$f_{\text{obj}}(.)$	The objective function
F_x, F_y, F_z	x, y, and z components of $\vec{F}_{\text{total}}$
τ_x, τ_y, τ_z	x, y, and z components of $\vec{\tau}_{\text{total}}$
$\vec{F}^n_{i,j}$	The normal component of $\vec{F}_{i,j}$
$\vec{F}^t_{i,j}$	The friction component of $\vec{F}_{i,j}$
$\mu_{i,j}$	The friction factor at point of action i, j
$\mu_{\max}$	The maximum acceptable friction factor

The minimum of $f_{\text{obj}}(.)$ is zero, which is satisfied if there exists a consistent solution.

The existence of at least one solution that satisfies all the constraints of the optimization problem implies that there exists one possible set of forces and acceptable friction factors that can satisfy the static equilibrium conditions of the target block. The implication, however, is valid as long as the assumptions on friction factors and mass distributions are close to real values. Thus, finding a solution just tells us that it is possible that X remains at rest. On the other hand, if there is no solution for the problem, then it is impossible for X to preserve its static equilibrium which means that block Z supports block X. In other words we can state that SUPP(Z,X) is true.

Please note that if the approach fails to extract a SUPP relation between two connected blocks A and B, then we cannot deduce that there is no support relation between A and B. In such case, the support relation is simply not known to hold between A and B.

D. Relational Representation Of Configurations of Blocks

The final output of the proposed method is a graph representing ACT and SUPP relations that exist between the blocks of the static configuration. The vertices of the graph represent the blocks, while edges between vertices represent the ACT and SUPP relations. Examples of such graphs are shown in Fig. 7. Such symbolic relational representation can then be used by high-level AI symbolic reasoners to decide on the safest sequence of unloading a container.

IV. Results

In this section, we present experimental results that we carried both in simulation and in real world. The aim of the simulation experiments was to analyze the execution time of the different stages of the approach on a large number of random configurations. The real world scenarios aimed at validating the constructed representation of the scene.

A. Random Configurations

We developed a scene configuration generator based on physics simulation. The simulator generates random configurations of blocks inside a container (see Fig. 5). The data of physical quantities such as mass, friction factors and so forth, in addition to the collision shape description of the objects (i.e. block dimensions), are set as minimum-maximum intervals. The attributes of the generated objects are uniformly sampled from the given intervals.

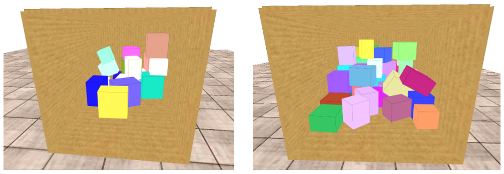
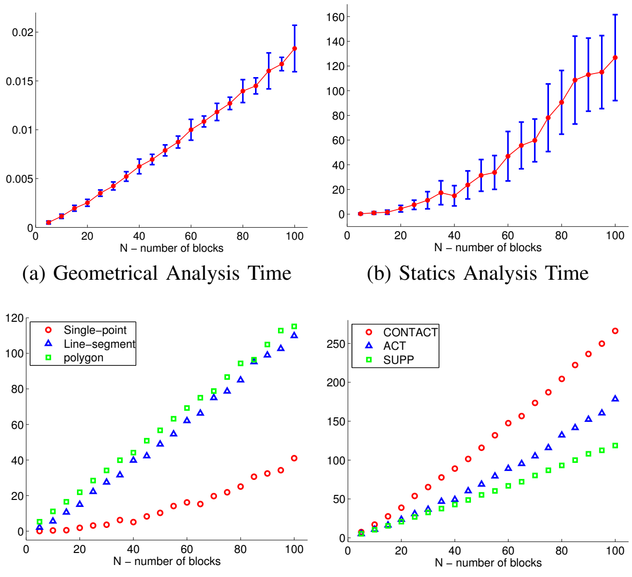

(a) (b)

Fig. 5: Two configurations generated by the simulator.

We considered a varying number of blocks N ($N = 5i, i = 1, \ldots, 20$), where for each value of N, we generated 20 random configurations, giving a total of 400 configurations. For each configuration, we recorded the time for the geometrical and static equilibrium analysis. As a figure of complexity of the generated configurations, we recorded the number of contacts between blocks, the number of extracted ACT and SUPP relations, and the number of contact types (i.e., single-point, line-segment and polygon contacts).



(a) Geometrical Analysis Time (b) Statics Analysis Time

(c) Contact Types (d) Relations

Fig. 6: Results of randomly generated configuration by the simulator. In (a) and (b) the vertical axes are time in seconds. In (c) and (d) the vertical axes are the average of the number of contact types and relations respectively.

Fig. 6a, resp. Fig. 6b, shows the average time taken by the geometrical, resp. static equilibrium, analysis. We can notice that the geometrical analysis takes very short time and increases linearly with the number of objects. The static equilibrium analysis requires much more time and increases polynomially. The time consumption of the static equilibrium analysis is due to the nonlinear optimization solver, that is called for each target object. Nevertheless, for realistic scenarios, we expect the number of objects extracted by any object-detection algorithm to be small, and thus, the performance of the static equilibrium analysis remains acceptable. In addition, the static equilibrium analysis stage can be parallelized.

Fig. 6c depicts the average number of contact types with respect to the number of blocks. As expected, the number of single-point contacts which are the result of less stable configurations (vertex-on-face and edge-cross-edge) is noticeably lower than the number of face-on-face and edge-on-face contact types.

The average number of extracted relations between blocks is shown in Fig. 6d. The number of support relations increases linearly. For $N < 20$, the number of ACT and SUPP relations are close to each other, that is, for each ACT(X,Y) relation, it is more probable to have a corresponding SUPP(Y,X) relation. However, as the number of blocks increases, the number of corresponding support relations diverges from that of act relations.

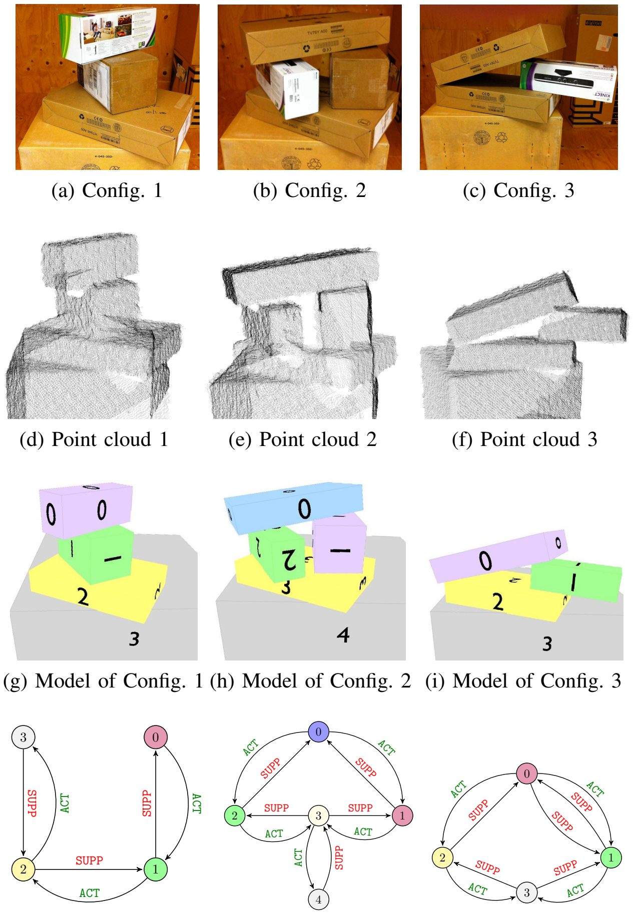
B. Real World Configurations

We examined the proposed approach on three real-world configurations of carton boxes placed in a mock-up container. The goal was to verify that the generated relations were consistent with the true configuration of the objects in the real world. We collected and registered 3D point clouds using two Kinect sensors placed at two different angles of views (looking at the scene from left and right sides). Since the focus of the paper is not on object detection, we performed the detection of the boxes and their attributes by a simple procedure consisting of registering models of boxes to the generated point clouds and manually refining the pose estimation process. Fig. 7 shows the real-world configurations with their models and the corresponding constructed relational scene representation.

In the real world configurations, the flat ground is modeled by a static block (the gray bottom block in Figures 7g to 7i). The first configuration is simple and contains three boxes stacked on top of each other (see Fig.7a). Its relational scene representation is shown in Fig. 7j where each ACT(X,Y) relation has a corresponding SUPP(Y,X) relation. The second configuration (see Fig.7b) has more interactions between blocks, where more than one block (1 and 2) are acting on another (block 3), and reacting against other block (block 0). Fig.7k depicts the corresponding representation. In the third configuration (see Fig.7c) there is a bidirectional SUPP relation between two blocks (0 and 1). This implies that no box can be removed without causing another box to fall as a side effect (see Fig.7l).

V. Summary and Future Work

We propose a new approach to analyze and represent configurations of cuboid-shaped objects in terms of abstract symbolic relations. The representation uses a minimal set of relations to capture possible physical interactions between objects in contact with each other. The proposed symbolic relational representation can be readily used by high-level AI reasoning paradigms to predict the effects of moving objects in contact with each other. To the best of the authors' knowledge, this issue has not been addressed before. The



(a) Config. 1 (b) Config. 2 (c) Config. 3

(d) Point cloud 1 (e) Point cloud 2 (f) Point cloud 3

(g) Model of Config. 1 (h) Model of Config. 2 (i) Model of Config. 3

(j) Graph of Config. 1 (k) Graph of Config. 2 (l) Graph of Config. 3

Fig. 7: Results of real world configurations. The first row shows images of the configurations. The registered point clouds are shown in the second row. In the third row, the refined pose of the models are drawn. At the bottom, the relational scene representations are shown as graphs of ACT and SUPP relations.

proposed approach constitutes a step forward in terms of bringing cognitive reasoning abilities to the area of robotic manipulation. Nevertheless, there are open issues which need to be addressed in future work. One intention is to extend the approach to deal with objects that can be supported by unseen objects due to occlusion and uncertainty in their attributes (pose and size).

APPENDIX

Proof of Proposition 1 Without loss of generality, we assume that the positive side of the separating plane contains object $\mathbf{A}$ and the negative side contains $\mathbf{B}$ (see Fig. 8). If $\mathbf{n}$ is the normal vector of the separating plane in the positive side, i.e., the y-component of the normal vector is strictly positive, and $\mathbf{w}(0, -w, 0)$, $w > 0$ is the weight of a point in $\mathbf{B}$, we show that none of points in $\mathbf{B}$ can exert force on $\mathbf{A}$ due to their weights. To do this, we compute the projection of the weight, $\mathbf{w}$, on the normal vector $\mathbf{n}$,

$$\mathbf{w}^{\mathbf{n}}_{\text{proj}} = (\mathbf{w} \cdot \mathbf{n})\mathbf{n} = -(wn_y)\mathbf{n}$$

Since $w > 0$ and $n_y > 0$, $\mathbf{w}$ has no contribution towards the positive side, and hence no force exertion on $\mathbf{A}$. Similarly, we can show that for all weights of points in $\mathbf{A}$, there exists a non-zero force contribution towards the negative side, i.e, ACT(A,B).

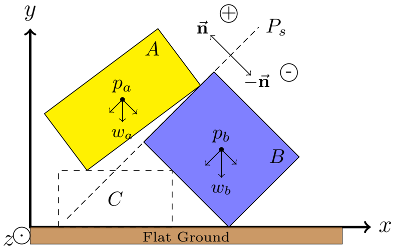


Fig. 8: Extracting the possible ACT relation between two blocks A and B in contact (Proposition 1).

REFERENCES

[1] W. Echelmeyer, A. Kirchheim, A. J. Lilienthal, H. Akbiyik, and M. Bonini, "Performance indicators for robotics systems in logistics applications," in *IROS Workshop on Metrics and Methodologies for Autonomous Robot Teams in Logistics (MMARTLOG)*, 2011.

[2] K. Rahardja and A. Kosaka, "Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues," in *1996 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE Press, 1996, pp. 1448–57.

[3] S. Fuchs, S. Haddadin, M. Keller, S. Parusel, A. Kolb, and M. Suppa, "Cooperative bin-picking with time-of-flight camera and impedance controlled dlr lightweight robot iii," in *2010 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE Press, 2010, pp. 4862–4867.

[4] J. Kirkegaard and T. B. Moeslund, "Bin-picking based on harmonic shape contexts and graph-based matching," in *Proceedings of the 18th Int. Conf. on Pattern Recognition - Volume 02*, ser. ICPR '06. IEEE Computer Society, 2006, pp. 581–584.

[5] L. P. Ellekilde, J. Jørgensen, D. Kraft, N. Krüger, J. Piater, and H. Petersen, "Applying a Learning Framework for Improving Success Rates in Industrial Bin Picking," in *2012 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*. IEEE Press, 2012, pp. 1637–1643.

[6] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *2007 IEEE/RAS Int. Conf. on Humanoid Robots*, December 2007, pp. 42–48.

[7] N. Curtis and J. Xiao, "Efficient and effective grasping of novel objects through learning and adapting a knowledge base," in *2008 IEEE/RSJ Int. Conf. on Intelligent Robots and System*, 2008, pp. 2252–2257.

[8] D. V. Pullar and M. J. Egenhofer, "Toward formal definitions of topological relations among spatial objects," in *Proceedings of the 3rd Int. Symposium on Spatial Data Handling*, O. Columbus, Ed. International Geographical Union, IGU, August 1988, pp. 225–241.

[9] J. Chen, D. Liu, H. Jia, and C. Zhang, "Cardinal direction relations in 3d space," in *Knowledge Science, Engineering and Management, Second Int. Conf., KSEM*, 2007, pp. 623–629.

[10] C. Li, J. Lu, C. Yin, and L. Ma, "Qualitative spatial representation and reasoning in 3d space," in *Proceedings of the 2009 Second Int. Conf. on Intelligent Computation Technology and Automation - Volume 01*, ser. ICICTA '09. IEEE Computer Society, 2009.

[11] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004, isbn: 9780521833783.

[12] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics 9th Edition Volume 2 Chapters 18-37 for So Methodist Univ*. John Wiley & Sons, 2011, isbn: 9781118115626.