# Hierarchical Scene Graphs and Contact-Aware Behavior Trees for Learning and Executing Bimanual Manipulation

Kumar Manas*†, Franziska Herbert*†, Georgia Chalvatzaki*†‡
*Interactive Robot Perception & Learning (PEARL) Lab, TU Darmstadt, Germany
†Hessian.AI, Darmstadt, Germany
‡Robotics Institute Germany (RIG)

***Abstract*—Scene graphs have become a standard representation for semantic scene understanding in robotics, yet they typically encode only static spatial and categorical relations. We argue that for contact-rich bimanual manipulation, contact and affordance relations must be elevated to first-class entities within the scene graph rather than being handled exclusively at the controller level. We present a concrete architectural proposal in which a hierarchical scene graph is augmented with explicit contact states, affordance edges, and temporal interaction dynamics, and show how this enriched representation naturally interfaces with behavior trees for reactive bimanual execution. Using supermarket shelf replenishment as a running example, we illustrate how the proposed representation captures the structure of bimanual sequencing: from bilateral approach and asymmetric first contact, through grasp establishment and shared transport, to supported placement and release. We describe our ongoing pipeline for extracting these contact-augmented graph sequences from human demonstrations and discuss the design choices, open challenges, and expected benefits of treating contact as semantic state.**

***Index Terms*—contact-rich manipulation, scene graphs, behavior trees, bimanual manipulation, imitation learning**

## I. Introduction and Position

Scene graphs provide a structured, queryable representation of environments that links objects, regions, and their semantic relations [4], [5]. In manipulation, they have been used to encode what objects are present, where they are located, and how they relate spatially [5]. However, *how* a robot physically interacts with an object—the evolution of contact, the stability of a grasp, the coordination between two arms—is typically represented only at the controller level through force thresholds, distance checks, or learned policies.

We argue that this separation creates a gap. When contact information is invisible to the task planner, the system cannot reason about intermediate manipulation states: Is the left gripper approaching while the right is already in contact? Has a stable bilateral grasp been established? Is the object being supported at the goal, or merely placed? These questions are critical for bimanual manipulation, where success depends on coordinated sequencing, stable contact maintenance, and timely recovery from failures [1], [6].

**Our position** is that contact should be modeled as a first-class semantic entity in the scene graph—alongside spatial relations like `on`, `inside`, and `left_of`—so that planners, execution monitors, and learning algorithms can query it directly. We make this argument concrete through three contributions:

1) A **contact-augmented scene graph formulation** that extends hierarchical scene graphs with explicit contact states, affordance edges, and temporal interaction dynamics (Section III).
2) A **design for behavior-tree execution** whose conditions query contact-level graph relations rather than raw sensor thresholds, enabling reactive bimanual coordination (Section IV).
3) A **demonstration pipeline** (in progress) that extracts contact-augmented graph sequences from human bimanual demonstrations, illustrated through supermarket shelf replenishment (Section V).

Throughout the paper, we use a single running example—a bimanual robot restocking a supermarket shelf—to ground each design choice in a concrete manipulation scenario.

## II. Related Work

**Scene graphs for robotics.** 3D scene graphs [4] encode hierarchical spatial and semantic structure, and systems such as Hydra [5] construct them in real time. These representations have been extended with neural feature fields for joint visual-semantic-spatial reasoning [11]. However, existing formulations typically stop at static spatial relations and do not model how contact evolves during manipulation.

**Contact in manipulation.** Contact modeling is well studied in physics engines [2], [3] and in hand-object interaction understanding from video [7], [8]. Yet these representations are rarely connected to the task-level planning graph. Our proposal bridges this gap by embedding contact states directly into the scene graph that the planner queries.

**Task structure from demonstrations.** Learning task graphs from human demonstrations [10] and error-aware imitation from teleoperation [9] provide complementary approaches for recovering manipulation structure. We build on these by targeting the specific structure needed for bimanual sequencing: which arm initiates contact, when bilateral engagement occurs, and how support transitions unfold.

**Behavior trees.** Behavior trees (BTs) [6] offer modular, reactive execution with natural support for fallbacks and

recovery. Our contribution is not the use of BTs per se, but the design of BT conditions that query contact-level scene graph relations, making the execution layer directly aware of interaction state.

## III. Contact-Augmented Scene Graph

### A. Running Example: Shelf Replenishment

Consider a robot that must restock cereal boxes onto a cluttered supermarket shelf from a supply cart. The task requires the robot to perceive the shelf layout, select a target item, coordinate a bimanual grasp (the box is too wide for one gripper), transport it to the shelf, and place it in a gap between existing items—all while maintaining stable contact and avoiding collisions with neighboring products. This example is representative because it involves constrained geometry, object affordances, clutter, and the full lifecycle of bimanual contact from approach to release.

### B. Graph Structure

Our graph is hierarchical (Fig. 1). At the **environment level**, nodes represent regions of interest: the shelf unit, individual shelf layers, the supply cart, and aisle zones. At the **object level**, nodes represent manipulable items (the target cereal box, neighboring products) and support surfaces (shelf boards, cart surface). At the **interaction level**, nodes represent the robot's end-effectors and, crucially, explicit **contact-point entities** that track the state of physical interaction between each end-effector and the objects it touches.

Each node carries semantic and geometric attributes (type, pose, dimensions, orientation, support region). We distinguish four families of edges:

- **Structural:** `located_in`, `is_on`, `part_of`, `supports`—encoding the containment and support hierarchy (e.g., cereal box `is_on` shelf layer, shelf layer `part_of` shelf unit).
- **Spatial:** `left_of`, `right_of`, `in_front_of`, `above`—capturing layout relative to the target.
- **Affordance:** `graspable`, `pushable`, `requires_bimanual`—encoding action possibilities. The cereal box has `requires_bimanual` due to its width; a small jar on the same shelf has `graspable` (single-hand).
- **Contact (our addition):** `getting_close`, `contact`, `grasping`, `moving_together`, `stable`, `released`—encoding the temporal evolution of physical interaction.

The last family is the core of our proposal. Rather than tracking contact only as a continuous signal in the controller, these edges make interaction state visible to any module that queries the graph.

### C. Why Contact Needs to Be in the Graph

To see why controller-level contact tracking is insufficient, consider the moment during shelf replenishment when the left gripper has contacted the cereal box but the right gripper is still approaching. In a standard scene graph, the box is simply `is_on` the shelf; no relation captures the asymmetric partial engagement. A behavior tree querying this graph cannot distinguish "no contact yet" from "one arm engaged, other approaching" from "bilateral grasp established"—yet these states require different execution strategies (continue approach vs. close grippers vs. begin transport).

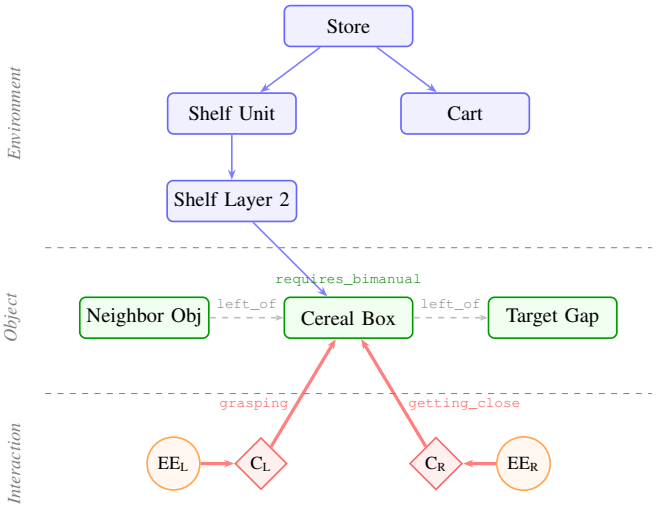


Fig. 1. Contact-augmented hierarchical scene graph for shelf replenishment. Blue: environment nodes; green: object nodes; orange: end-effectors; red diamonds: contact-point entities with contact-state edges (bold red). Here the left gripper has established a grasp while the right is still approaching—an asymmetric intermediate state that is invisible without explicit contact representation.

By adding contact-point nodes and contact-state edges, the graph at this moment contains: $\text{EE}_\text{L} \xrightarrow{\texttt{grasping}}$ cereal box and $\text{EE}_\text{R} \xrightarrow{\texttt{getting\_close}}$ cereal box. This makes the asymmetric state explicit and queryable.

### D. Contact State Transitions

Contact states evolve temporally. We define transitions using end-effector-to-object distance $d_{\text{ee,obj}}$ and relative velocity $\mathbf{v}_{\text{rel}}$:

$$\texttt{grasping} \iff d_{\text{ee,obj}} < \epsilon_{\text{grasp}} \ \wedge \ \|\mathbf{v}_{\text{rel}}\| < v_{\text{thresh}}. \quad (1)$$

The temporal derivative $\dot{d}(t) \approx (d(t) - d(t-1))/\Delta t$ allows inferring `getting_close` (negative $\dot{d}$, above threshold) and detecting loss of contact (`released`) when $d$ increases after a `grasping` state. For bimanual coordination, `moving_together` is established when both end-effectors satisfy `grasping` and the object velocity matches the coordinated end-effector velocity; `stable` requires additionally that force residuals remain below a margin during transport or placement.

These are intentionally simple criteria. The point is not to propose a novel contact detector, but to argue that *whatever* contact detection is used, its output should be represented as a semantic graph relation rather than remaining a hidden controller variable.

## IV. Behavior Trees Grounded in Contact State

We use behavior trees (BTs) [6] for execution because they are modular, interpretable, and naturally support fallback-based recovery. Our specific design choice is to **bind BT conditions to scene-graph queries** over contact relations rather than to raw sensor thresholds.

For the shelf replenishment example, the bimanual grasp subtree checks:

- `BothGrippersGrasping()`: queries whether *both* contact-point nodes have `grasping` edges to the target.
- `MovingTogether()`: queries whether the `moving_together` relation holds (bilateral transport is stable).
- `ObjectSupportedAtGoal()`: queries whether the target has a new `is_on` relation with the destination shelf layer and the `stable` contact state is active.

Fallback nodes handle intermediate states: if only one gripper has contact, the tree retries the other arm's approach rather than aborting; if `moving_together` is lost during transport, the tree pauses and re-establishes stable engagement.

This design decouples the BT structure from domain-specific thresholds. The same subtree pattern (approach $\rightarrow$ establish bilateral grasp $\rightarrow$ verify coordinated motion $\rightarrow$ place $\rightarrow$ verify support) can be reused for a different domain—such as bimanual luggage handling on a conveyor—by changing only the graph content (object types, spatial layout) while keeping the contact-query conditions identical.

## V. Extracting Contact-Augmented Graphs from Demonstrations

A key motivation for making contact explicit in the graph is that it enables learning bimanual sequencing structure from human demonstrations. We are building a pipeline (Fig. 2) that converts first-person video of supermarket replenishment into contact-augmented graph sequences:

1) **Hand-object interaction extraction:** From video, we detect hands and objects in contact using established methods [7] and bimanual affordance extraction [8] to recover which hand is touching which object and when.
2) **Scene graph construction:** Detected objects and surfaces are mapped to graph nodes with structural and spatial edges derived from shelf geometry.
3) **Contact-state annotation:** The hand-object contact signals are mapped to our contact-state vocabulary (`getting_close`, `grasping`, `moving_together`, etc.), producing a temporally ordered sequence of graph snapshots.
4) **Primitive sequence extraction:** Transitions between graph states are segmented into manipulation primitives (reach, grasp, transport, place, release) following the semantic-geometric task graph formulation of Herbert et al. [10].

**Current status.** We have collected several first-person replenishment demonstrations in a test store, covering multiple participants and scenarios. Hand-object interaction extraction and scene graph construction are operational. Contact-state annotation and the full integration with BT execution are under active development. We emphasize that the contribution of this paper is the *representational argument and architectural design*, not yet the empirical validation of the pipeline.

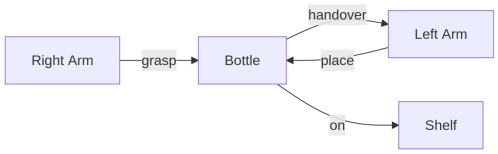


Fig. 2. Pipeline for extracting contact-augmented graph sequences from human demonstration video. Each stage converts continuous video into increasingly structured representations, culminating in primitive sequences that can initialize behavior tree execution.

**What we expect to learn.** Bimanual demonstrations contain latent sequencing decisions—whether to clear neighboring items first, whether to reorient before insertion, whether to use one or both hands depending on object size. By representing these demonstrations as contact-augmented graph sequences rather than raw trajectories, we aim to recover these decisions as graph-level patterns (e.g., a `pushable` affordance edge triggers a clearing primitive before the main grasp sequence) that transfer to new shelf configurations.

## VI. Discussion: Open Challenges and Expected Benefits

**Expected benefits.** Making contact a semantic graph entity has several design advantages: (i) planners can reason about intermediate bimanual states that are otherwise hidden; (ii) BT conditions become domain-independent queries rather than domain-specific thresholds; (iii) demonstration data can be structured at the contact-transition level, enabling more transferable sequence learning than trajectory-level imitation.

**Robustness under perception noise.** A clear challenge is that contact states inferred from vision and proprioception are noisy. Hysteresis on state transitions (requiring sustained threshold satisfaction before switching) and temporal smoothing can mitigate spurious transitions, but a systematic study of failure modes is needed. Incorporating tactile sensing would provide a more reliable contact signal and is a natural extension.

**Scalability of the relation vocabulary.** Our current vocabulary of contact states is manually defined. As the range of manipulation tasks grows, this vocabulary may need to be extended or learned. Whether a fixed, human-designed set of contact relations suffices for a broad class of bimanual tasks, or whether task-specific relations emerge from data, is an open question.

**Integration with learned models.** The interface between demonstration-derived task models and online BT execution is underspecified in our current design. Concretely, the learned

model could output a most-likely next primitive given the current graph state (a greedy policy), a ranked distribution over primitives (allowing the BT to select based on feasibility), or a full sub-task graph that the BT instantiates as a subtree. Evaluating these alternatives is a priority for our upcoming experimental work.

**Cross-domain transfer.** We have discussed the architecture primarily through shelf replenishment. A secondary domain—airport luggage handling, involving larger objects with stronger bilateral coordination requirements—uses the same graph structure and BT conditions with different object-level content. Whether the shared representation provides practical transfer benefits, or whether domain-specific tuning dominates, remains to be validated.

## VII. Conclusion

We have argued that contact-rich bimanual manipulation requires contact to be represented as a first-class semantic entity in the scene graph, not only as a controller-level signal. Through a concrete architectural proposal and a running shelf-replenishment example, we showed how contact-augmented scene graphs interface naturally with behavior trees for reactive bimanual execution and with demonstration pipelines for learning manipulation structure. The main open challenges are robustness under perception noise, scalability of the contact vocabulary, and the precise integration interface between learned task models and online execution. We hope this position motivates further work on bridging semantic scene representations and physical interaction reasoning for manipulation.

## Acknowledgement

This research is funded by the EU Horizon Europe Project "MANiBOT" (101120823).

## References

[1] M. T. Mason, *Mechanics of Robotic Manipulation*. MIT Press, 2001.

[2] E. Todorov, T. Erez, and Y. Tassa, "MuJoCo: A physics engine for model-based control," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[3] J. Carpentier, F. Valenza, N. Mansard, *et al.*, "Pinocchio: Fast forward and inverse dynamics for poly-articulated systems," https://stack-of-tasks.github.io/pinocchio, 2015–2021.

[4] I. Armeni *et al.*, "3D scene graph: A structure for unified semantics, 3D space, and camera," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2019.

[5] N. Hughes *et al.*, "Hydra: A real-time spatial perception system for 3D scene graph construction and optimization," in *Robotics: Science and Systems (RSS)*, 2022.

[6] M. Colledanchise and P. Ögren, *Behavior Trees in Robotics and AI: An Introduction*. CRC Press, 2018.

[7] D. Shan *et al.*, "Understanding human hands in contact at internet scale," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[8] M. Heidinger, S. Jauhri, V. Prasad, and G. Chalvatzaki, "2HandedAfforder: Learning precise actionable bimanual affordances from human videos," in *Proc. IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, 2025, pp. 14743–14753.

[9] J. Wong *et al.*, "Error-aware imitation learning from teleoperation data for mobile manipulation," in *Conf. on Robot Learning*, 2021.

[10] F. Herbert *et al.*, "Learning semantic-geometric task graph-representations from human demonstrations," *arXiv preprint arXiv:2601.11460*, 2026.

[11] C. Maurer *et al.*, "UniFField: A generalizable unified neural feature field for visual, semantic, and spatial uncertainties in any scene," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2026.