From Semantic Understanding to Geometric Features: Using Foundation Models for Novel Robotic Tasks

Nizan Mashall, Erez Karpas, Miriam Zacksenhouse

Technion – Israel Institute of Technology

Abstract

Foundation Models contain implicit knowledge about objects and how to use them. Our system makes this knowledge explicit by detecting key geometric features (vertices, edges, and planes) and defining task related coordinate systems on the object. This information enables model-based planners to execute actions without requiring task-specific training, even with novel objects. The system utilizes five pre-trained foundation models, enabling zero-shot capabilities and a fully automated process.

Introduction

Performing robotic manipulations in household environments represents one of the greatest challenges in robotics. While humans effortlessly execute thousands of daily tasks, from stirring soup to dispensing salt, robots struggle to replicate this versatility. While certain fundamental skills (e.g., grasping, locomotion) require high dexterity, most daily tasks can be decomposed into simple geometric transformations - primarily rotations around specific axes or translations along defined vectors. For instance, turning a doorknob or pressing a button can be represented as a single rotational or linear motion respectively.

The complexity of these tasks lies not in their physical execution but in the semantic understanding of the objects and their use. Semantic information can be manually programmed for specific tasks and known objects, but the vast diversity of household objects and tasks makes manual programming or learning approaches—which require explicit training for each task—impractical, expensive, and potentially unreliable.

Recent integration of Large Language Models (LLMs) with robotic systems has shown remarkable capabilities in breaking down high-level tasks into actionable steps. However, their potential to expand the variety of actionable tasks using semantic understanding and common sense reasoning remains largely untapped.

This demonstration introduces a novel zero-shot system that bridges this gap by integrating multiple state-of-the-art (SOTA) artificial intelligence (AI) models to detect task-relevant geometric features based on semantic understanding, enabling the execution of simple tasks. For task requests specified in natural language, the system provides geometric features- such as vertices, edges, or planes- and their relevant direction vectors, which can be used to generate trajectories for executing novel tasks using novel objects.

The proposed system introduces three fundamental features that enable trajectory planning for novel tasks:

1. **Object-Level Semantics:** Translating semantic knowledge about how objects should be used into specific geometric features (planes, edges, vertex) and their task-relevant coordinate systems.
2. **Affordance-Based Generalization:** Enabling operation across variants within object categories by identifying fundamental geometric features that support characteristic actions (e.g., the striking face common to most hammer variants for impact tasks, or the dispensing openings present in most salt dispensers for controlled release, independent of manufacturer-specific designs).
3. **Zero-Shot Execution:** Leveraging foundation models' knowledge base—derived from extensive training data—to understand object usage patterns, enabling identification of task-relevant geometric features for novel tasks without task-specific training.

Background

Traditional robotic planning frameworks employ a two-layer architecture: a low-level skills layer providing basic capabilities (e.g., picking, pushing) and a high-level task planner for sequencing these skills. While effective for predefined tasks and objects, this architecture struggles with a critical limitation: the lack of task-oriented semantic knowledge. For instance, placing a mug requires identifying its bottom surface, while pouring necessitates detecting its top opening. While such semantic information can be manually programmed for known objects, extracting this knowledge for unknown objects in novel tasks presents a real challenge.

Current approaches attempt to address this challenge through learning-based methods:

Vision-Language-Action Models

Recent work has introduced Vision-Language-Action Models(VLAs), exemplified by RT-2 (Brohan et al. 2023, 2022),

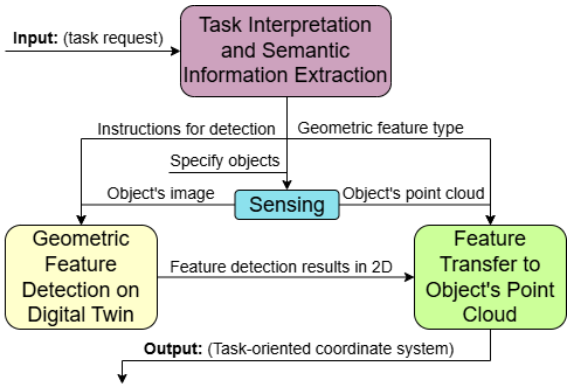

Figure 1: Overview of the proposed system for task-relevant geometric feature detection. The system architecture comprises three main stages, visually distinguished by colors: Task Interpretation and Semantic Information Extraction (purple) which processes natural language input to generate detection instructions and specify geometric feature types, digital twin feature detection (yellow) which performs geometric feature detection through image processing, and feature transfer (green) which maps detected features to the real object's point cloud. The sensing block (blue) represents the function of acquiring images and point clouds of real objects, which is assumed as given in this demonstration. Directional arrows indicate information flow between components. The system takes a natural language task request as input and produces task-oriented coordinate systems as output.

which learn from demonstration to directly map natural language instructions and visual inputs to robotic actions. This method avoids the need for explicit semantic information by learning from demonstration datasets and using visual processing for scene understanding to generate direct action for robot control. While these models achieve interesting performance, they present several limitations:

- **Data Requirements:** Training requires extensive data collection (17 robot-months with 13 robots for RT-2).
- **Control Architecture Limitations:** Limited ability to integrate with specially trained low-level controllers, and resource-intensive processing, results in low-frequency commands.
- **Black-Box Nature:** Deep learning architecture makes decision-making processes difficult to monitor and validate.

LLM-Based Program Generation

Another approach, demonstrated by Microsoft (Vemprala et al. 2024), uses LLMs like ChatGPT (Achiam et al. 2023) to generate robot programs through a structured pipeline. This approach defines high-level function libraries for robot control, translates natural language into sequences of these functions, and incorporates human feedback for safety and quality assurance.

While this approach enables natural language interaction, it remains constrained to high-level planning and cannot generate new skills or bridge the gap to low-level control effectively.

Our System

Method

Our system aims to extract task-oriented semantic information from natural language instructions without requiring task-specific training or demonstration. The key objective is to identify task-relevant geometric features that can be used for task execution. We propose a novel approach that leverages multiple AI models to extract and utilize this semantic information.

As illustrated in Figure 1, our system processes natural language task requests through three sequential stages using five integrated foundation models. In the first stage, an LLM processes input and generates detection prompts. In the second stage, a text-to-image model (T2I) creates digital twins in task-specific contexts, and vision-language models (VLMs) detect task-relevant geometric features with a segmentation model isolating specific regions. In the final stage, a 3D reconstruction model converts 2D features to point cloud representations for mapping detected features to the real object. The implementation of each stage proceeds as follows:

Task Interpretation and Semantic Information Extraction: The process begins with natural language commands from users. In Figure 2 step 1, an example task query from users is shown. The LLM processes these commands to identify required objects, determine necessary actions, and specify relevant geometric feature types. Additionally, the LLM generates two types of prompts: one for scene image generation and another for the VLM to detect regions of interest.

Geometric Feature Detection on Digital Twin: Using a T2I model with prompts generated by an LLM, the system creates four images showing objects in their task-specific context. VLM then uses LLM-generated prompts to detect the relevant geometric features, which are subsequently segmented using a segmentation model. Using a digital twin in context allows the system to overcome the limitations of direct detection, which is often unsuccessful. Subsequently, another VLM agent evaluates the segmented results across all four images, scoring each detection to select the highest-quality result. Figure 2 demonstrates this multi-stage detection process using the example of hammering a nail into a wall. In step 2, LLM-generated prompts guide T2I 3 to create photorealistic representations of objects in their functional context. In step 3, the VLM uses additional LLM-generated prompts to identify task-relevant geometric features, indicated by red bounding boxes. In step 4, the bounded regions are segmented and highlighted in blue, representing critical geometric features-in this case, the hammer's striking surface.

Feature Transfer to Object's Point Cloud: The registration of the geometric features of the digital twin with the real object comprises several key steps, as illustrated in Fig-

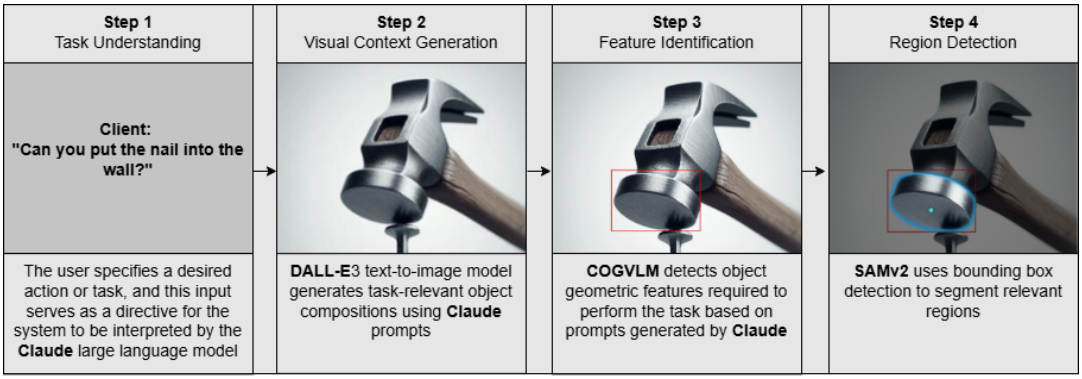

Figure 2: Pipeline for second stage: visual geometric feature detection on digital twins. The system flows through four steps: task parsing, synthetic image generation (DALL-E3), feature detection (CogVLM), and region segmentation (SAMv2). The hammer example demonstrates the detection of task-relevant geometric features on digital twins.

ure 3. The process integrates three primary inputs: the VLM-derived segmented digital twin, the sensor-captured real object point cloud, and the LLM-provided geometric feature type (vertex, edge, or plane). The segmented digital twin undergoes 2D to 3D transformation by 3D reconstruction model, preserving critical geometric features identified in the previous stage.

The registration pipeline begins with data preparation to ensure point cloud compatibility. This includes noise filtering, downsampling the digital twin to a hull point cloud, and random upsampling of the real object to achieve point count parity. Registration proceeds through a coarse-to-fine approach, initially employing Iterative Closest Point (ICP) point-to-plane (Besl and McKay 1992) optimization for coarse alignment. Fine registration followed by segmenting the point cloud into principal components using K-means (Yadav and Sharma 2013) clustering, followed by scaling and centering of individual segments. After registration, the segmentation performed on the digital twin is propagated to find the corresponding points in the real object.

Through final optimization, the coordinate system is established based on the geometric feature type specified by the LLM, positioned either on a single point, at the midpoint of an edge, or at the center of a plane. This coordinate system orientation provides the essential direction to perform the robotic manipulation task.

System Implementation Challenges

Detecting task-relevant geometric features through morphological methods lacks generalizability, as each task requires its own specialized algorithm. By leveraging semantic knowledge of how objects are used, semantic detection offers a more general approach. However, when implementing semantic detection, such as finding a hammer's striking face, we discovered limitations in both T2I models and VLMs when identifying specific geometric features (planes, edges, or vertices). Despite extensive testing of various methods, direct extraction of this information using VLMs or T2I models proved unsuccessful. Nevertheless, we found that VLMs could successfully identify object parts when contextualized within task-specific relationships with other objects. Figure 4 compares these methods and demonstrates that while queries about specific hammer parts fail, task-related queries successfully detect the relevant areas.

Another significant challenge lies in the registration of point clouds between our real object point cloud and its digital twin (e.g., different variants of hammers). These point clouds often exhibit substantial variations, making registration particularly challenging. Traditional approaches such as ICP and learning-based methods showed poor performance. To address this, we developed a coarse-to-fine registration method that proved highly effective for everyday household objects and hand tools.

Demonstration

A classic use case demonstrating this system's capabilities is the task of hammering a nail into a wall. Given the command "please put a nail in the wall," the LLM identifies the hammer and nail as necessary objects and hammering as the required action.

The LLM then generates an image generation prompt: "Show only hammer and nail. The hammer and nail in position just before the hammer strikes the nail. The metal hammer head should be clearly visible, with the flat striking face oriented directly towards the camera. The striking face of the hammer head should be barely touching the top of the nail head, with no gap between them. Use a plain white background to isolate the hammer, nail, and striking face clearly."

For geometric feature detection, the LLM provides a second prompt: "Where is 'the hammer surface which touches the nail' in the image?"

Figure 2 shows the progression: the generated image in step 2 and the detected geometric feature indicated by a red bounding box in step 3. Figure 3 illustrates the registration process between the real object and its digital twin, with the bottom right image showing the final detected plane and its normal vector in the real point cloud. This output provides essential information for planning and executing novel robotic tasks.

In our implementation, we employed Claude LLM for

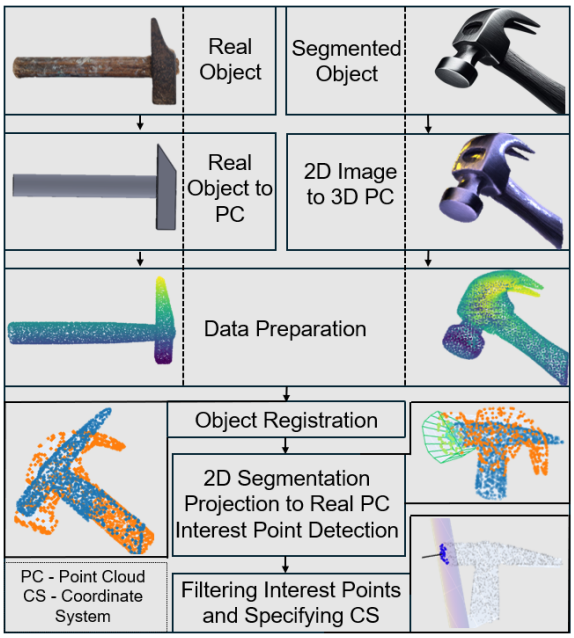

Figure 3: Pipeline for third stage: geometric feature registration between digital twin and real object. The system processes both the real object (left) and segmented digital twin (right) through multiple steps: point cloud generation, data preparation, object registration, and coordinate system specification. Each step is visualized using a hammer example, showing the progression from initial inputs to final geometric feature alignment.

task interpretation and prompt generation. Digital twin generation was performed using DALL-E 3. Vision-language tasks were facilitated by CogVLM (Wang et al. 2023), with Claude serving as a secondary agent. For segmentation, we utilized SAMv2 (Ravi et al. 2024), a model capable of zero-shot segmentation without requiring task-specific training. Furthermore, SF3D (Boss et al. 2024) was used for efficient mesh reconstruction from single images.

Conclusion

We presented a system that leverages foundation models to extract semantic information and convert it into geometric features required for robotic task execution. Our approach works with novel objects and tasks without task-specific training.

In future work, we aim to extend the system to extract additional execution parameters, including geometric constraints, velocity, force, and other physical requirements. We plan to generate sensor monitoring code on demand for closed-loop control and explore integration with model-based planners to create an end-to-end system-from natural language commands to performing novel tasks.

This work contributes to advancing autonomous robots in complex real-world settings.

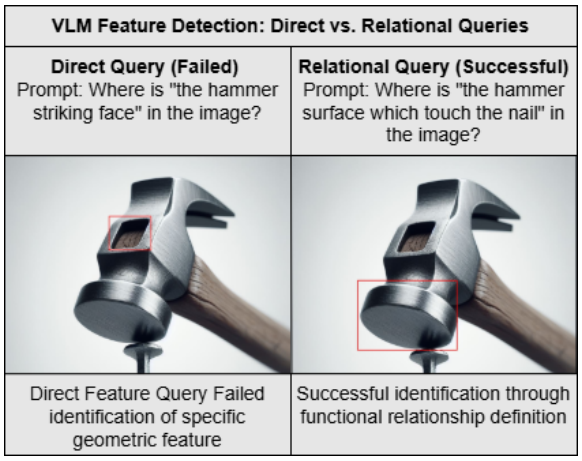

Figure 4: Demonstration of VLM feature detection limitations and solutions. While direct queries about specific object parts fail (left), reformulating the query to focus on object relationships enables successful feature detection (right).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Besl, P. J.; and McKay, N. D. 1992. Method for registration of 3-D shapes. In *Sensor fusion IV: control paradigms and data structures*, volume 1611, 586–606. Spie.

Boss, M.; Huang, Z.; Vasishta, A.; and Jampani, V. 2024. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. *arXiv preprint arXiv:2408.00653*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Chen, X.; Choromanski, K.; Ding, T.; Driess, D.; Dubey, A.; Finn, C.; et al. 2023. Rt-2: Vision-language-action models transfer web knowledge to robotic control. *arXiv preprint arXiv:2307.15818*.

Brohan, A.; Brown, N.; Carbajal, J.; Chebotar, Y.; Dabis, J.; Finn, C.; Gopalakrishnan, K.; Hausman, K.; Herzog, A.; Hsu, J.; et al. 2022. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*.

Ravi, N.; Gabeur, V.; Hu, Y.-T.; Hu, R.; Ryali, C.; Ma, T.; Khedr, H.; Rädle, R.; Rolland, C.; Gustafson, L.; et al. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Vemprala, S. H.; Bonatti, R.; Bucker, A.; and Kapoor, A. 2024. Chatgpt for robotics: Design principles and model abilities. *IEEE Access*.

Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; Song, X.; et al. 2023. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Yadav, J.; and Sharma, M. 2013. A Review of K-mean Algorithm. *Int. J. Eng. Trends Technol*, 4(7): 2972–2976.