Legged Robot Agility Guided by Human Gesture

Taerim Yoon[a], Dongho Kang[b], Jin Cheng[b], Minsung Ahn[c], Stelian Coros[b] and Sungjoon Choi[a]

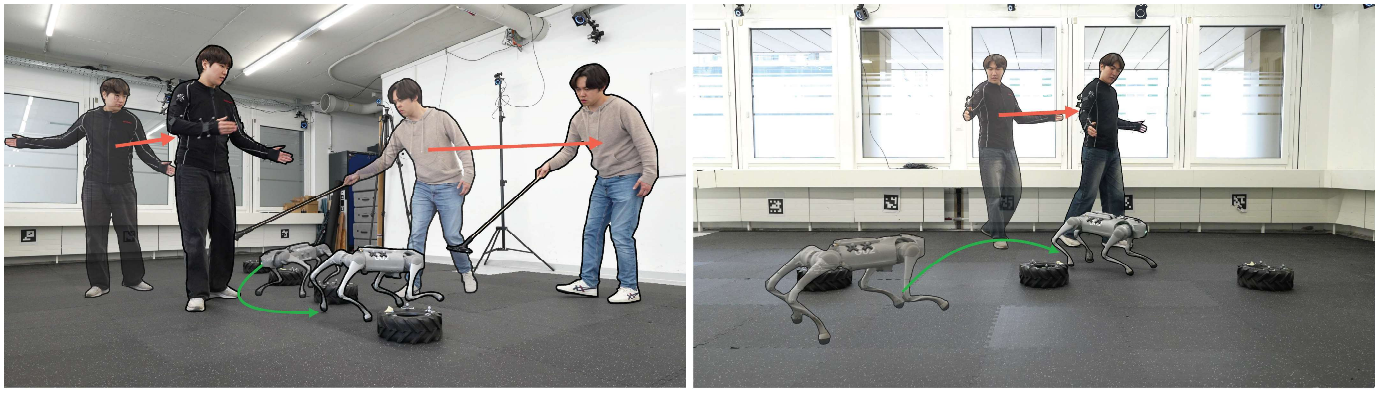

(a) Data Collection (b) Control Command with Gesture

Fig. 1: (a) Illustration of the data collection procedure. The person on the left, wearing a motion capture suit, makes a gesture command to the robot, while the person on the right uses a teaching rod to guide the dog in performing the desired movement. (b) In the inference stage, the training rod is no longer needed, and only the person in the motion capture suit controls the robot using gestures.

***Abstract*—This work introduces an approach for guiding agile-legged robots through cluttered environments and complex obstacle courses using human body gestures. Our framework, Interactive Robot Parkour, enables robots to interpret human gestures as high-level commands and execute precise, responsive movements in real-time. At the core of our method is the use of real-world interaction data between humans and robots. This data is used to learn a mapping between gestures and desirable navigation paths within reconstructed scenes. The scene reconstruction, built in a digital twin fashion, facilitates data-efficient learning and effective adaptation to new environments. We validate our approach through simulated and real-world parkour tasks, demonstrating that robots can reliably respond to human intentions with minimal engineering effort to teach new gesture commands. Video footage of our preliminary results is available at *https://youtu.be/F62xzzdsljc*.**

I. Introduction

As legged robots become more integrated into our daily lives, their ability to understand and collaborate with humans is gaining increasing attention across a wide range of real-world applications [1]. Typically, these robots are controlled by users using input devices such as joysticks or keyboards, which define the target velocity in the desired moving direction. However, relying on such input devices is not always practical, as they demand continuous attention and precise control [2]. For example, industrial workers wearing thick protective gloves may find manipulating input devices difficult or even impossible.

[a]These authors are with the Robot Intelligence Lab in the Department of Artificial Intelligence at Korea University, South Korea.

[b]These authors are with the Computational Robotics Lab in the Department of Computer Science at ETH Zurich, Switzerland.

[c]Minsung Ahn is with the Robotics and Mechanisms Laboratory in the Department of Mechanical and Aerospace Engineering at UCLA, USA.

To seek an alternative, we turn to human-animal cooperation in agile navigation. Animals possess the ability to understand social cues and execute fine, agile movements, while humans can communicate high-level commands through various forms of non-verbal interaction. Among these, body gestures stand out as a particularly effective means of guiding legged animals. A noticeable example is the *dog agility competition*, where a human trainer and a dog work as a coordinated team—the trainer uses gestures to indicate which obstacles to overcome and how to navigate the course. In such scenarios, a verbal command like “Go there” to a robot can be ambiguous, often requiring additional context to clarify the intended target. In contrast, a simple gesture can directly and intuitively express spatial intent, making communication more efficient and natural by providing clear visual cues about direction and goal location.

Inspired by this animal ability, we aim to facilitate human-robot cooperation, where human gestures serve as high-level commands, and the robot responds with appropriate movements when faced with obstacles, as shown in Figure 1b. To this end, we propose the Interactive Robot Parkour (IRP), which leverages human gesture data paired with ground-truth human intentions represented as goal points. While real-world data collection enables operators to interact with the robot in a more natural and realistic manner, we further enhance the dataset through digital-twin scene reconstruction, as shown in Figure 3. This augmentation allows the robot agent to learn more precise movements and adapt effectively to new scenarios.

We demonstrate our method across several representative robot navigation scenarios, supported by empirical analysis showing that the proposed data augmentation approach helps

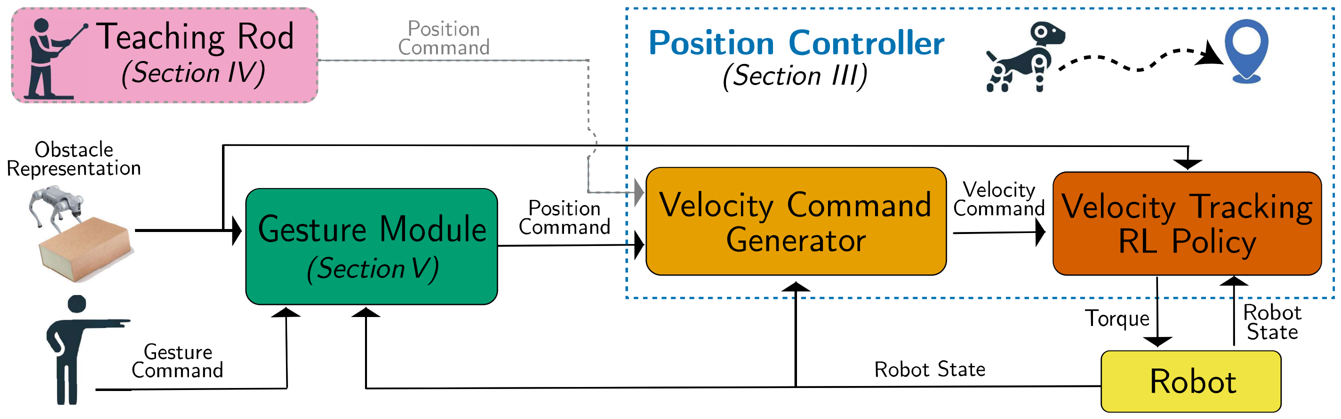


Fig. 2: Overview of interactive robot parkour

the robot agent better recognize obstacles, follow the correct sequence, and successfully achieve the task goal. For example, in the *Zigzag through tires* task, our method outperforms alternative approaches by achieving a higher success rate, successfully navigating through multiple tires without making contact with any of them.

In summary, key contributions of this work are as follows:

- We present Interactive Robot Parkour (IRP), a novel approach for gesture-based control of legged robots navigating cluttered environments.
- We propose a data augmentation method that involves reconstructing interaction scenes in simulation from real-world data to enhance learning efficiency and improve the adaptability of trained models.
- We present evaluation protocols and baselines showing improved performance in gesture-driven agility tasks requiring precise spatial understanding.

II. Overview

Consider a quadruped robot whose base position and orientation are denoted as $\mathbf{p}_\mathrm{b} \in \mathbb{R}^3$ and $\mathbf{h} \in \mathbb{H}$, respectively, with $\mathbb{H}$ indicating the space of unit quaternions. We define the robot's linear and angular velocities, along with the joint velocity, are represented as $\mathbf{v} \in \mathbb{R}^3$, $\mathbf{w} \in \mathbb{R}^3$, and $\dot{\boldsymbol{\theta}} \in \mathbb{R}^M$, respectively. The generalized coordinate of the robot is defined as $\mathbf{q} = [\mathbf{p}_\mathrm{b}, \mathbf{h}, \boldsymbol{\theta}]$ and its time derivative as $\dot{\mathbf{q}} = [\mathbf{v}, \mathbf{w}, \dot{\boldsymbol{\theta}}]$. Following this the state of the robot is defined as $\mathbf{x} = [\mathbf{q}, \dot{\mathbf{q}}]^T$. Additionally, we define the human gesture as $\mathbf{m}$ consisting of six keypoint positions at the shoulders, elbows, and wrists on both the left and right sides. Furthermore, we define $\mathbf{o}$ to represent obstacles, which can be either a box or a cylinder. The representation $\mathbf{o}$ includes position $\mathbf{p}_o$ and quaternion $\mathbf{h}_o$. For a box, $\mathbf{o}$ also includes dimensions (l^x, l^y, l^z), while for a cylinder, it includes radius r and height l^z.

Our goal is to interpret human intentions through gestures $\mathbf{m}$ and control the robot accordingly based on its states $\mathbf{x}$ and surroundings obstacles $\mathbf{o}$. To achieve this, we develop a system that involves collecting interaction data between humans and the robot, as well as training methodologies.

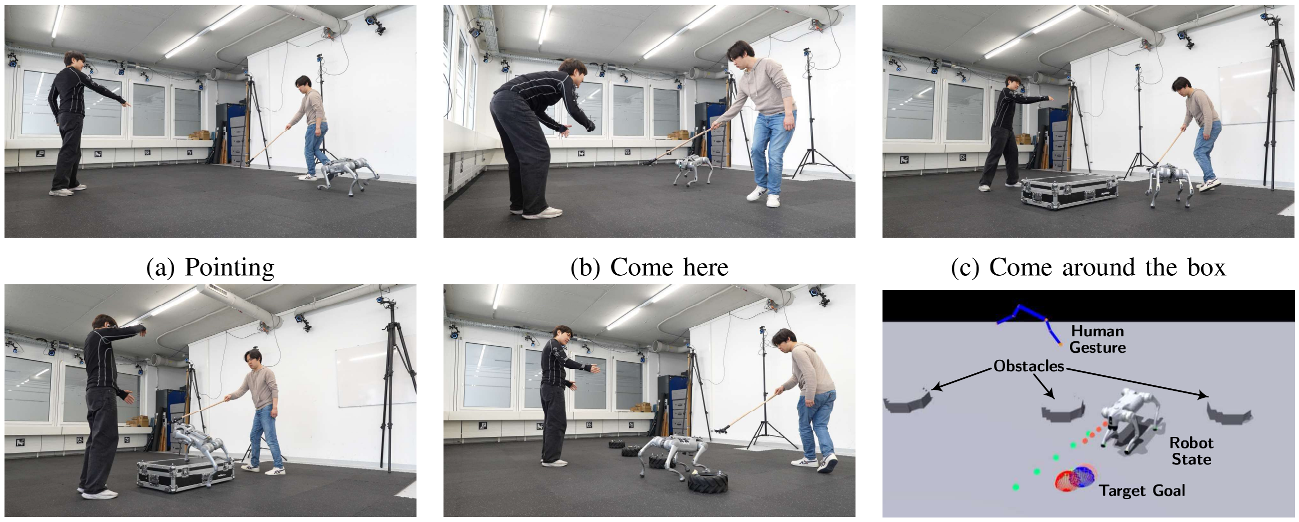
As illustrated in Figure 1, IRP consists of three sequential stages. First, we train a low-level controller using Reinforcement Learning (RL) to navigate through obstacles. Next, we introduce a teaching rod as a physical guide, which the robot follows to facilitate interaction. We collect data on human interactions using both the teaching rod and gestures while the robot operates under the position controller, as shown in Figure 1a. Finally, we train a gesture inference module to control the robot using gestures without the teaching rod, as demonstrated in Figure 1b. Details of these three stages will be provided in Section III, Section IV, and Section V, respectively.

III. Positional Controller

The goal of the positional controller is to control the robot so that it can navigate to the target position while overcoming obstacles. As illustrated in Figure 2, it consists of a velocity generator and controller. A velocity generator outputs the heading velocity command $v_\mathrm{com} = |\mathbf{g}^{xy} - \mathbf{p}^{xy}|_2$ and the orientation command $\phi_\mathrm{com} = \mathrm{yaw}(R^T(\mathbf{g} - \mathbf{p}))$, where $\mathbf{p}$ and $\mathbf{g}$ denote the robot's base and target positions, respectively, and $(\cdot)^{xy}$ indicates ground-projected points. In essence, the velocity generator functions as a proportional mechanism, guiding the robot toward the target position using the velocity controller.

The velocity controller is developed based on the work of Cheng et al. [3] and is designed to follow velocity-level commands $v_\mathrm{com} \in [-1.0, 1.0]$ m/s and angular velocity commands $w_\mathrm{com} \in [-\pi/3, \pi/3]$ rad/s while navigating through obstacles. We train this controller using Isaac Gym [4], leveraging high-speed simulation for efficient learning with GPU (RTX-3090). The RL policy utilizes a scandot representation of obstacles, which is acquired in the real world via motion capture cameras (OptiTrack Prime X22), ensuring precise environmental perception.

The velocity controller must be robust when overcoming the obstacles, as its failure would compromise the entire system. In this regard, we make two key modifications in policy learning and system design to enhance tracking performance. Firstly, we introduced a stand boolean in the state representation. When the given velocity command v is below the threshold $v_\mathrm{th} = 0.1$m/s, the stand boolean is set to 1; otherwise, it is set to 0. This explicit signal demonstrated an improved



(a) Pointing (b) Come here (c) Come around the box

(d) Jump over this box (e) Zigzag through tires (f) Reconstructed Scene

Fig. 3: Illustration of data collection procedures for five interaction scenarios. (a) *Pointing* and (b) *Come here* depict human-robot interactions in open space. (c) *Come around the box* and (d) *Jump over this box* illustrate interactions involving a box obstacle. (e) *Zigzag through tires* demonstrates interaction with multiple tire obstacles. (f) These scenes are reconstructed in simulation to augment data for efficient training.

transition from walking to standing and vice versa. Secondly, we employed a Kalman filter-based state estimator [5] that fuses data from joint encoders, foot contact sensors, and global position measurements obtained from motion capture cameras. This state estimator reliably provides state values for the control policy and enables data collection in the global frame, allowing full-scene reconstruction in simulation.

IV. Collecting Interaction Data

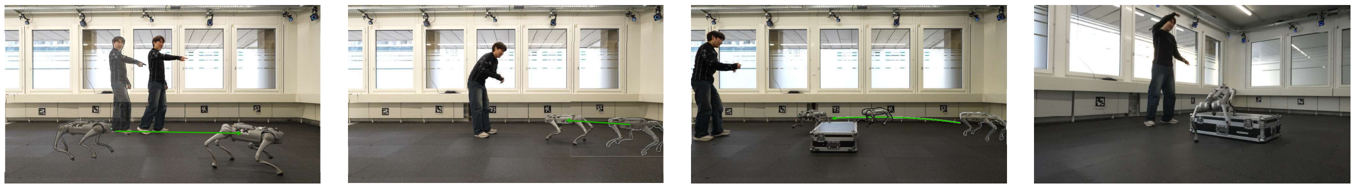
Our data collection approach is inspired by a dog training technique called *luring*, which uses a treat to guide the dog toward a desired behavior. For example, a trainer may lure a dog to a specific location with a snack while simultaneously making a pointing gesture. Over time, the dog learns to associate the gesture with the intended action and can perform the behavior without needing the treat as a prompt. Similarly, we employ a teaching rod to guide the robot, mimicking the role of the treat in luring. The robot follows the rod using a position controller from Section III, gradually learning the meaning of the gestures so it can eventually respond without requiring the rod.

Specifically, our goal is to collect interaction data, denoted as D, comprising human gestures $\mathbf{m}$, robot states $\mathbf{x}$, obstacle representations $\mathbf{o}$, and target goals $\mathbf{g}$, recorded at a frequency of 10Hz. As shown in Figure 3, we define five interaction scenarios between humans and the robot: *Go there*, *Come here*, *Jump over this box*, *Come around the box*, and *Zigzag through tires*. Two humans involved in this data collection follow these scenarios, where they adjust more fine behaviors on the spot through active verbal communication. Among the scenarios, *Pointing* is one of the simplest gesture commands. The commander points to a location, directing the robot to move there. In *Following*, the commander raises either their left or right hand. If the left hand is raised, the robot moves to the left side of the commander; if the right hand is raised, it moves to the right side. In the *Come here* scenario, the commander waves their hands toward themselves, signaling the robot to approach. For scenarios involving obstacles, *Jump over this box* requires the commander to wave their hand over their head, instructing the robot to jump over the box. In *Come around the box*, the commander waves their hand around the box, guiding the robot to detour around it rather than jumping over. Lastly, in *Zigzag through tires*, the commander waves their hand left and right, signaling the robot to weave through the tires in a zigzag pattern. In particular, this data contains sufficient information to fully reconstruct the interaction scene in simulation, as shown in Figure 3f.

V. Training Gesture Module

Our goal is to leverage collected interaction data to train a gesture module that maps human gesture commands to corresponding target goals. By training this module, the robot becomes capable of interpreting gestures and performing desired tasks autonomously, eliminating the need for a teaching rod, as illustrated in Figure 1b.

Training the gesture module presents several challenges. Primarily, data efficiency is a significant concern since our data is gathered directly from real-world interactions, inherently restricting the available amount. Standard imitation approaches, such as Behavior Cloning [6], train the gesture module according to an expert's distribution. This approach, however, is prone to error accumulation, leading to significant misbehavior when the robot encounters states not covered by the training data, which can be referred to as *drifting*. In this case, the robot may completely disregard human gestures and start wandering aimlessly, or it might attempt unnecessary maneuvers, such as jumping over obstacles like boxes or tires, when it should avoid doing so.



(a) Pointing (b) Come here (c) Come around the box (d) Jumpover this box

Fig. 4: Illustrations of the interactive parkour model deployed in the real world. Human gestures control the robot in real-time, enabling the robot to navigate through obstacles according to the operator's intentions.

Method	Baseline (m)	Ours (m)
Point	2.7774	**2.3170**
Come here	**0.9176**	1.0101
Zigzag tires	7.1376	**1.2915**
Jump box	9.4244	**5.0746**
Come around	4.9384	**3.2200**

TABLE I: Comparison of baseline and proposed method based on positional error using ground-truth interaction data. Lower values indicate better performance.

To address these issues, our approach introduces data augmentation through simulation-based scene reconstruction. Specifically, we reconstruct interaction scenarios in a simulated global frame, incorporating obstacles and human gestures using timestamped real-world interaction data. Within these simulations, we intentionally provoke the drifting behavior of the gesture module by deploying it alongside a low-level controller. When the simulated agent inevitably begins to drift, we generate augmented target positions by computing the offset between the ground-truth goal position $\mathbf{g}_{\text{gt}}$ and the current simulated position of the drifting agent $\mathbf{p}_{\pi}$. Formally, the augmented data point is calculated as $\mathbf{g}_{\text{gt}} - \mathbf{p}_{\pi}$. This augmentation technique shares similarities with DAgger [7], as it effectively trains the gesture module to recognize and recover from drifted states, thereby enhancing robustness and generalization in real-world scenarios.

Another challenge is ensuring precise robot movements relative to obstacles. For instance, when jumping over a box, the robot must approach it perpendicularly to the edges to achieve stable navigation. Similarly, when zigzagging through tires, the robot should pass between them without making contact. We mitigate this issue by decoupling the timing of the simulated human gestures and target goals from the simulation clock. Specifically, the next gesture and its corresponding target goal appear only after the robot reaches the current target, which we refer to as conditional framing. In contrast, a baseline approach with synchronized simulation and data timelines results in rapidly changing goals before the robot can reach them. This can potentially cause the robot to mistakenly jump over the tires due to improper data augmentation. By implementing our timing strategy, the robot demonstrates refined and accurate navigation through obstacles.

VI. Preliminary Results

We evaluate the performance of the gesture module by comparing the ground truth base positions from our interaction dataset against the simulated robot positions. In detail, we deploy the robot in the reconstructed scenes from the interaction data for 60 s, where the gesture commands are also derived from the interaction data. On the one hand, the baseline method is Behavior Cloning (BC) [6], which utilizes supervised learning to train the gesture module. On the other hand, our proposed method adopts data augmentation through scene reconstruction across all scenarios. Additionally, we apply the conditional framing technique from Section V for scenarios involving obstacles.

We evaluate the position error using the L1 distance. Specifically, we simulate the robot while providing human gestures based on prerecorded data. We then measure the positional difference between the robot in the simulator and the ground-truth base position using the collected data. As the results shown in Table I, the proposed method demonstrates improved performance overall. Particularly, the proposed method excels in scenarios involving obstacles with an average improvement of 3.97 m, while also showing comparable results for scenarios without obstacles with an average improvement of 0.18 m.

We also showcase the interactive parkour deployed in the real world [1]. As illustrated in Figure 4, the deployed robot can follow gesture commands from a human operator. For pointing gestures, the robot moves directly toward the indicated area, as shown in Figure 4a. If the operator swings their arms toward themselves, the robot approaches gently, as depicted in Figure 4b. An outward swing from the box prompts it to circumnavigate the obstacle before approaching, as demonstrated in Figure 4c. A swing of the arms over the head directs the robot to the middle of the box to execute a jump, as shown in Figure 4d.

VII. Future Work

Our framework enables robots to interpret human gestures and perform agile movements, supporting intuitive human-robot interaction. It also allows the evaluation of positional errors accumulated by the deployed agent, providing a foundation for benchmarking collaboration. Since the collected data is sufficient for full scene reconstruction, it holds strong potential for broader quantitative analysis. However, our current evaluation methods do not fully leverage this. In future work, we plan to develop more systematic metrics to better assess human-robot collaboration and contribute to standardized benchmarks.

[1] Video footage is available at *https://youtu.be/F62xzzdsljc*

References

[1] C. S. Bellicoso *et al.*, "Advances in real-world applications for legged robots," *Journal of Field Robotics*, vol. 35, no. 8, pp. 1311–1326, 2018.

[2] M. A. Goodrich and A. C. Schultz, "Human–robot interaction: A survey," *Foundations and Trends in Human–Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[3] X. Cheng, K. Shi, A. Agarwal, and D. Pathak, "Extreme parkour with legged robots," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2024, pp. 11 443–11 450.

[4] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.

[5] M. Bloesch, M. Hutter, M. A. Hoepflinger, S. Leutenegger, C. Gehring, C. D. Remy, and R. Siegwart, "State estimation for legged robots-consistent fusion of leg kinematics and imu," *Robotics*, vol. 17, pp. 17–24, 2013.

[6] D. A. Pomerleau, "Efficient training of artificial neural networks for autonomous navigation," *Neural computation*, vol. 3, no. 1, pp. 88–97, 1991.

[7] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.