Extreme Parkour with Legged Robots

Anonymous

Abstract: Humans can perform parkour by traversing obstacles in a highly dynamic fashion requiring precise eye-muscle coordination and movement. Getting robots to do the same task requires overcoming similar challenges. Classically, this is done by independently engineering perception, actuation, and control systems to very low tolerances. This restricts them to tightly controlled settings such as a predetermined obstacle course in labs. In contrast, humans are able to learn parkour through practice without significantly changing their underlying biology. In this paper, we take a similar approach to developing robot parkour on a small low-cost robot with imprecise actuation and a single front-facing depth camera for perception which is low-frequency, jittery, and prone to artifacts. We show how a single neural net policy operating directly from a camera image, trained in simulation with large-scale RL, can overcome imprecise sensing and actuation to output highly precise control behavior end-to-end. We show our robot can perform a high jump on obstacles 2x its height, long jump across gaps 2x its length, do a handstand and run across tilted ramps, and generalize to novel obstacle courses with different physical properties.

1 Method

We wish to train a single neural network that goes directly from raw depth and onboard sensing to joint angle commands. To train adaptive motor policies, recent approaches use two-phase student teacher training [1, 2, 3, 4]. Later works [5, 6] introduce regularized online adaptation (ROA) to collapse this into a single phase. To train the vision backbone, a similar teacher-student framework is employed [7, 8, 9] where a teacher trained with privileged scandots information is distilled to a student with access to depth. In this paper, we use ROA for adaptation and two-phase training for the vision backbone but introduce key modifications for the challenging task of extreme parkour.

First, since parkour requires diverse behaviors to traverse different obstacles it is challenging to engineer reward functions specific to each. We present a simple, unified reward formulation from which diverse behaviors emerge automatically and are perfectly adapted to the terrain geometry. Second, during parkour the robot needs to be able to choose its own direction as opposed to following human-specified ones. For instance, when jumping across tilted ramps, it needs to jump on the first ramp at a very specific angle and then change directions immediately which is impossible for a human to provide. Instead, we provide directions in phase 1 using suitably placed waypoints and in phase 2 we train a network to predict these oracle heading directions from depth information.

1.1 Unified Reward for Extreme Parkour

The rewards used in [7] do not transfer directly to the parkour case. The robot cannot follow arbitrary direction commands and instead must have the freedom to choose the optimal direction. Instead of randomly sampling directions, we compute direction using waypoints placed on the terrain as

$$\hat{\mathbf{d}}_w = \frac{\mathbf{p} - \mathbf{x}}{\|\mathbf{p} - \mathbf{x}\|} \tag{1}$$

where $\mathbf{p}$ is the next waypoint location and $\mathbf{x}$ is robot location in the world frame. The velocity tracking reward is then computed as

$$r_{tracking} = \min(\langle \mathbf{v}, \hat{\mathbf{d}}_w \rangle, v_{cmd}) \tag{2}$$

where $\mathbf{v} \in \mathbb{R}^2$ is the robot's current velocity in world frame and $v_{cmd} \in \mathbb{R}$ is the desired speed. Note that [7] tracks velocity in the base frame but world frame is used. This is done to prevent the robot from exploiting the reward and learning the unintended behavior of turning around the obstacle.

7th Conference on Robot Learning (CoRL 2023), Atlanta, USA.

While the above reward is sufficient for diverse parkour behavior, for challenging obstacles the robot tends to step close to the edge to minimize energy usage. This behavior is risky and does not transfer well to real settings. We therefore add a term to penalize foot contacts near terrain edges.

$$r_{clearance} = -\sum_{i=0}^{4} c_i \cdot M[p_i] \tag{3}$$

c_i is 1 if ith foot touches the ground. M is a boolean function which is 1 $i$ff the point p_i lies within 5cm of an edge. p_i is the foot position for each leg.

The rewards defined above typically lead to a gait that uses all four legs. However, a defining feature of parkour is walking in different styles that are aesthetically pleasing but may not be biomechanically optimal. To explore this diversity, we introduce a term to track a desired forward vector using the same inner product design principle, which can be controlled by the operator at test time.

$$r_{stylized} = W \cdot \left[0.5 \cdot \langle \hat{\mathbf{v}}_{fwd}, \hat{\mathbf{c}} \rangle + 0.5\right]^2 \tag{4}$$

where $\hat{\mathbf{v}}_{fwd}$ is the unit vector pointing forward along the robot's body, $\hat{\mathbf{c}}$ is also a unit vector indicating the desired direction and W is a binary number to switch the reward on/off. In our case, we train the robot to do a handstand and choose $\hat{\mathbf{c}} = [0,0,-1]^T$. W is sampled randomly in $\{0,1\}$ at training and controlled via remote at deployment. We also use the additional regularization terms from [6].

1.2 Reinforcement Learning from Scandots (Phase 1)

We use the above rewards to learn a policy using model-free RL [10] in simulation. This policy takes as input, the proprioception $\mathbf{x}$, scandots $\mathbf{m}$, target heading $\hat{\mathbf{d}}$, walking flag W and commanded speed v^{cmd}. We use regularized online adaptation (ROA) [5] to train an adaptation module to estimate environment properties. We create a set of tilted ramps, gaps, hurdles and high step terrains (Fig. **??**), and arrange them in increasing difficulty as in [7]. To aid exploration, robots are first initialized in easy levels. They are promoted to harder ones if they traverse more than half the length, and demoted to an easier one if they travel less than half the expected distance $v^{\text{cmd}}T$ (T is episode length).

1.3 Distilling Direction and Exteroception (Phase 2)

The phase 1 policy relies on two pieces of information not directly available on the real robot. First, exteroceptive information is only available in the form of depth images from a front-facing camera instead of scandots. Second, there is no expert to specify waypoints and target directions, these must be inferred from the visible terrain geometry. We use supervised learning to obtain a deployable policy which automatically estimates these quantities. For exteroception, similar to the RMA architecture in [7] we replace the scandots input to the base policy with a convnet-GRU pipeline that accepts depth. This network is trained using DAgger [11], with ground truth actions from the phase 1 policy. We use student predicted motor commands to step the environment. We initialize the actor network with a copy from phase 1 to minimize the drift when we directly step the environment with student actions. However for predicted heading, the depth encoding network is not pre-trained. Directly using predicted heading as observation could result in catastrophic distribution drift leading to incorrect action labels from the teacher. To overcome this issue, we propose to use a mixture of teacher and student (MTS). Concretely, the heading command the student observes

$$obs_\theta = \begin{cases} \theta pred, & \text{if } |\theta pred - \hat{\mathbf{d}}_w| < 0.6 \\ \hat{\mathbf{d}}_w, & \text{otherwise} \end{cases}$$

where $\theta pred$ and $\hat{\mathbf{d}}_w$ are the desired yaw angle from prediction and oracle, respectively. obs_θ is the yaw angle the policy observes.

References

[1] A. Kumar, Z. Fu, D. Pathak, and J. Malik. Rma: Rapid motor adaptation for legged robots. *Robotics: Science and Systems*, 2021.

[2] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning robust perceptive locomotion for quadrupedal robots in the wild. *Science Robotics*, 7(62):eabk2822, 2022.

[3] Y. Song, K. Shi, R. Penicka, and D. Scaramuzza. Learning perception-aware agile flight in cluttered environments, 2023.

[4] J. Fu, Y. Song, Y. Wu, F. Yu, and D. Scaramuzza. Learning deep sensorimotor policies for vision-based autonomous drone racing, 2022.

[5] Z. Fu, X. Cheng, and D. Pathak. Deep whole-body control: learning a unified policy for manipulation and locomotion. In *Conference on Robot Learning*, pages 138–149. PMLR, 2023.

[6] X. Cheng, A. Kumar, and D. Pathak. Legs as manipulator: Pushing quadrupedal agility beyond locomotion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023.

[7] A. Agarwal, A. Kumar, J. Malik, and D. Pathak. Legged locomotion in challenging terrains using egocentric vision. In *6th Annual Conference on Robot Learning*, 2022.

[8] R. Yang, G. Yang, and X. Wang. Neural volumetric memory for visual locomotion control. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1430–1440, 2023.

[9] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. Kim, and P. Agrawal. Learning to jump from pixels. *arXiv preprint arXiv:2110.15344*, 2021.

[10] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[11] 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*, 2011.