Legged Locomotion in Challenging Terrains using Egocentric Vision

Ananye Agarwal* [1] **Ashish Kumar*** [2], **Jitendra Malik**†[2], **Deepak Pathak**†[1]
[1]Carnegie Mellon University, [2]UC Berkeley

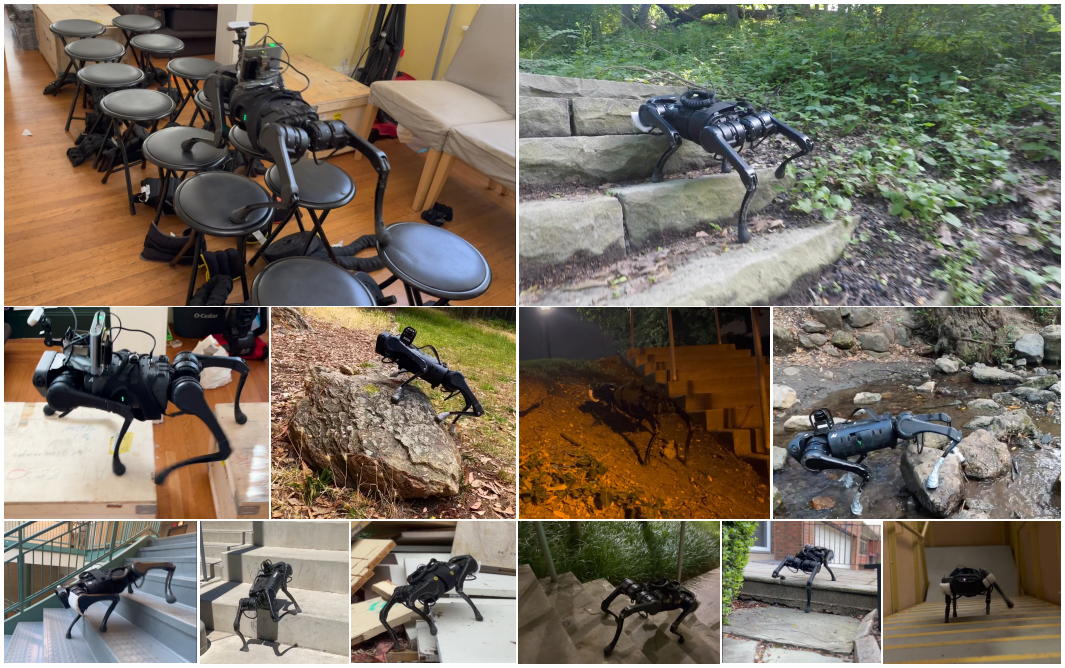


Figure 1: Our robot can traverse a variety of challenging terrain in indoor and outdoor environments, urban and natural settings during day and night using a single front-facing depth camera. The robot can traverse curbs, stairs and moderately rocky terrain. Despite being much smaller than other commonly used legged robots, it is able to climb stairs and curbs of a similar height. Videos at https://blindsupp.github.io/visual-walking/

Abstract: Animals are capable of precise and agile locomotion using vision. Replicating this ability has been a long-standing goal in robotics. The traditional approach has been to decompose this problem into elevation mapping and foothold planning phases. The elevation mapping, however, is susceptible to failure and large noise artifacts, requires specialized hardware, and is biologically implausible. In this paper, we present the first end-to-end locomotion system capable of traversing stairs, curbs, stepping stones, and gaps. We show this result on a medium-sized quadruped robot using a single front-facing depth camera. The small size of the robot necessitates discovering specialized gait patterns not seen elsewhere. The egocentric camera requires the policy to remember past information to estimate the terrain under its hind feet. We train our policy in simulation and transfer to the real world without any fine-tuning and can traverse a large variety of terrain while being robust to perturbations like pushes, slippery surfaces, and rocky terrain. Videos are at https://blindsupp.github.io/visual-walking/

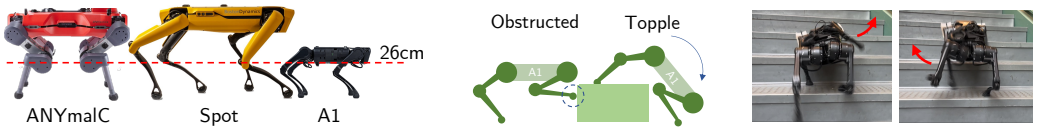
1 Introduction

Of what use is vision during locomotion? It turns out that both humans [1] and robots [2, 3] can do remarkably well at blind walking. Where vision becomes necessary is for locomotion in

*Equal Contribution. †Equal Advising.

challenging terrains like staircases or stepping stones. In this paper, we will develop this capability for a quadrupedal walking robot equipped with egocentric depth vision. We use a reinforcement learning approach trained in simulation, which we are directly able to transfer to the real world. Figure 1 and the accompanying videos shows some examples of our robot walking guided by vision.

The walking policy is trained by reinforcement learning with a recurrent neural network being used as a short term memory of recent egocentric views, proprioceptive states, and action history. Competing approaches which rely on metric localization to construct elevation maps which are noisy [4, 5, 6]. This hinders the ability of such systems to perform reliably on gaps and stepping stones.



(a) Robot size comparison (b) Challenges due to size (c) Emergent hip abduction

Figure 2: A smaller robot (a) faces challenges in climbing stairs and curbs due to the stair obstructing its feet while going up and a tendency to topple over when coming down (b). Our robot deals with this by climbing using a large hip abduction that automatically emerges during training (c).

Not having pre-programmed gait priors is useful for our relatively small robot [1] (see Figure 2). Predefined gait priors or reference motions fail to generalize to obstacles of even a reasonable height because of the relatively small size of the quadruped. The emergent behaviors like hip abduction enable our robot to climb high obstacles.

2 Method: Legged Locomotion from Egocentric Vision

We train in two phases, RL in phase 1 and supervised learning in phase 2 (Fig. 3).

Phase 1: Reinforcement Learning Given the local elevation map in front of the robot, proprioception and commanded action, we learn a policy using PPO without gait priors and with biomechanics inspired reward functions to walk on a variety of terrains. The elevation map $\mathbf{m}_t$ is passed through a MLP β to get $\tilde{\mathbf{m}}_t \in \mathbb{R}^{32}$ which is concatenated along with the rest of the observations and fed to a recurrent policy to obtain actions $\mathbf{a}_t$. Similar to [7] we generate slopes, stairs, discrete terrain and stepping stones of varying difficulty level. We randomize parameters of the simulation and add small i.i.d. gaussian noise to observations to bridge the sim2real gap and make our policy robust. See appendix for more implementation details.

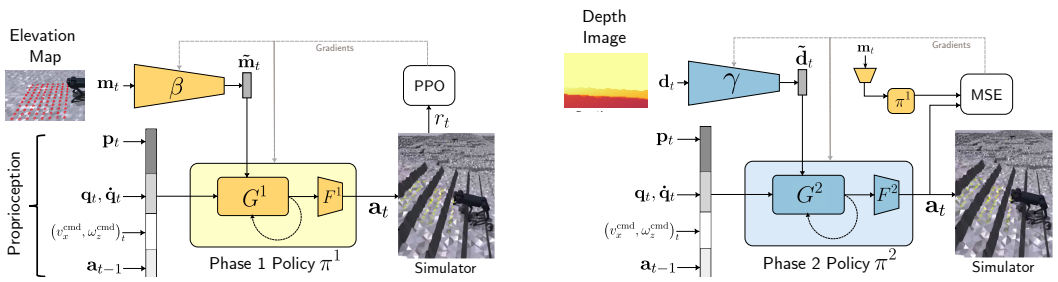
Phase 2: Supervised Learning Having learnt a useful visuomotor policy in phase 1, we can now use supervised learning to distil these into a phase 2 policy that receives depth input $\mathbf{d}_t$. We create a copy of the recurrent base policy $G^2 \leftarrow G^1, F^2 \leftarrow F^1$. The scandots compression MLP β is replaced with a convolutional depth backbone γ which processes depth map $\mathbf{d}_t$ to produce a depth latent $\tilde{\mathbf{d}}_t$. We use DAgger [8] with truncated backpropagation through time. The student can be deployed as-is on the hardware using only the available onboard compute.

3 Experimental Setup

We use the Unitree A1 robot pictured in Fig. 2. For simulation, we use the IsaacGym (IG) simulator with the legged_gym library [7] to develop walking policies. We compare against two baselines

- **Blind policy** trained without access to any scandots. This must rely on proprioception to traverse terrain and helps quantify the benefit of vision for walking.
- **Noisy** Methods which rely on elevation maps constructed using depth and tracking cameras often have large noise due to sensor deficiencies. As a result, large noise must be added during simulation as well. We distil our phase 1 policy to a student with large noise added to the elevation map. We use the noise model in [5]. This shows the benefits of directly using the depth image.

[1] A1 standing height is 40cm as measured by us. Spot, ANYmalC both are 70cm tall reported here and here.



(a) Phase 1: Reinforcement Learning

(b) Phase 2: Supervised Learning

Figure 3: We train our locomotion policy in two phases to avoid rendering depth for too many samples. (a) In phase 1, we use RL to train a policy π^1 that has access to a low-resolution elevation map that is cheap to compute. (b) In phase 2, we use π^1 to provide ground truth actions which another policy π^2 is trained to imitate. This student has access to depth map from the front camera.

Terrain	Average Distance (↑)				Mean Time to Fall (s)			
	Blind	Privileged	Noisy	Ours	Blind	Privileged	Noisy	Ours
Slopes	29.1	18.0	26.0	**31.5**	155.5	13.3	18.0	89.2
Stepping Stones	0.5	2.2	5.3	**18.1**	3.8	1.9	4.3	46.3
Stairs	11.1	7.0	12.5	**23.8**	70.6	5.8	9.6	60.5
Discrete Obstacles	22.0	18.3	26.0	**30.0**	124.1	12.7	17.7	80.6

Table 1: We measure the average distance travelled and mean time to fall for all methods on different terrains in simulation. Our method outperforms the baselines on all terrains for average distanced traveled.

4 Results and Analysis

Simulation Results We report mean time to fall and mean distance travelled before crashing for different terrain and baselines in Table 1. For each method, we train a single policy for all terrains and use that for evaluation. Although the blind policy makes non trivial progress on stairs, discrete terrain and slopes, it is significantly less efficient at traversing these terrains. On slopes our method travels a greater distance in nearly half of the time implying that the blind baseline gets stuck often. Similarly, on stairs and discrete obstacles the distance travelled by the vision baseline is much greater in a shorter amount of time. The noisy baseline has worse average distances and mean time to fall. This trend is even more significant on the stepping stones terrain where both baselines barely make any progress. The blind policy has no way of estimating the position of the stone and crashes as soon as it steps into the gap. For the noisy policy, the large amount of added noise makes it impossible for the student to reliably ascertain the location of the stones since it cannot rely on proprioception any more.

Real World Comparisons We compare the performance of our method to the blind baseline in the real world. In particular we have 4 testing setups as shows in Figure 4: Upstairs, Downstairs, Gaps and Stepping stones. We see that the blind baseline is incapable of walking upstairs beyond a few steps and fails to complete the staircase even once On downstairs, we observe that the blind baseline achieves 100% success, although it is unstable which led to the detaching of the rear right hip of the robot during our experiments. We additionally show results in stepping stones and gaps, where the blind robot fails completely. We show a 100% success on all tasks except for stepping stone on which we achieve 94% success, which is very high given the challenging setup.

Urban Environments We experiment on stairs, ramps and curbs (Fig. 1). The robot was successfully able to go upstairs as well as downstairs for stairs of height upto 24cm in height and 28cm as the lowest width. It sometimes misses a step, but shows impressive recovery behaviour and continues

Upstairs
17cm high, 30cm deep

	Success	#Stairs
Ours	100%	13
Blind	0%	2.2

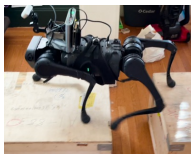
Downstairs
17cm high, 30cm deep

	Success	#Stairs
Ours	100%	13
Blind	100%	13

Stepping Stones
30cm wide, 15cm apart

	Success	#Stones
Ours	94%	9.4
Blind	0%	0

Gaps
26cm apart



	Success
Ours	100%
Blind	0%

Figure 4: We show success rates and time-to-failure (TTF) for our method and the blind baseline on curbs, stairs, stepping stones and gaps.

climbing or descending. The robot is able to climb curbs and obstacles as high as $26cm$ which is almost as high as the robot 2. This requires an emergent hip abduction movement.

Gaps and Stepping Stones We construct an obstacle course consisting of gaps and stepping stones out of tables and stools (Fig. 4). The robot achieves a 100% success rate on gaps of upto 26cm from egocentric depth and 94% on difficult stepping stones. The stepping stones experiment shows that our visual policy can learn safe foothold placement behavior even without an explicit elevation map or foothold optimization objectives. The blind baseline achieves zero success rate on both tasks and falls as soon as any gap is encountered.

Natural Environments We also deploy our policy on outdoor hikes and rocky terrains next to river beds (Fig. 1). We see that the robot is able to successfully traverse rugged stairs covered with dirt, small pebbles and some large rocks. It also avoids stumbling over large tree roots on the hiking trail. On the beach, we see that the robot is able to successfully navigate the terrain despite several slips and unstable footholds given the nature of the terrain.

5 Related Work

Legged locomotion Legged locomotion an important problem which has been studied for decades. Several classical works use model based techniques, or define heuristic reactive controllers to achieve the task of walking [9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21]. Other works use RL for walking in real and simulation [22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 3, 37, 38, 39]. However, most of these methods are blind, and only use proprioceptive signal to walk.

Locomotion from Elevation Maps To achieve visual control of walking, classical methods build metric elevation maps and plan footsteps over them [40, 41, 42, 43, 44, 45, 46, 4, 47, 48, 6, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 5, 59, 7, 60, 61, 62]. Elevation maps can be noisy or incorrect and dealing with imperfect maps is a major challenge to building robust locomotion systems. Solutions to this include incorporating uncertainty in the elevation map [40, 63, 64] and simulating errors at training time to make the walking policy robust to them [5].

Locomotion from Egocentric Depth Closest to our method is the line of work that does not construct explicit elevation maps and predicts actions directly from depth frames. [39] learn a policy that for obstacle avoidance from egocentric depth on flat terrain, [65] train a hierarchical policy which uses depth to traverse curved cliffs and mazes in simulation, [66] use lidar scans to show zero-shot generalization to difficult terrains. Yu et al. [67] train a policy to step over gaps by predicting high-level actions from egocentric depth from the head and below the torso. Relatedly, Margolis et al. [68] train a high-level policy to jump over gaps from egocentric depth using a whole body impulse controller. In contrast, we directly predict target joint angles from egocentric depth without constructing metric elevation maps.

Acknowledgments

We thank Kenny Shaw for help with the hardware. Shivam Duggal, Ellis Brown, Xuxin Cheng, Zipeng Fu, Shikhar Bahl helped with recording experiments. We thank Alex Li for proofreading. This work is supported by DARPA Machine Common Sense grant and ONR N00014-22-1-2096.

References

[1] J. M. Loomis, J. A. Da Silva, N. Fujita, and S. S. Fukusima. Visual space perception and visually directed action. *Journal of experimental psychology: Human Perception and Performance*, 18 (4):906, 1992.

[2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 2020.

[3] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. In *RSS*, 2021.

[4] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter. Perceptive locomotion in rough terrain–online foothold optimization. *RA-L*, 2020.

[5] 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.

[6] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim. Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot. In *ICRA*, 2020.

[7] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[8] 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.

[9] H. Miura and I. Shimoyama. Dynamic walk of a biped. *IJRR*, 1984.

[10] M. H. Raibert. Hopping in legged systems—modeling and simulation for the two-dimensional one-legged case. *IEEE Transactions on Systems, Man, and Cybernetics*, 1984.

[11] H. Geyer, A. Seyfarth, and R. Blickhan. Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 2003.

[12] K. Yin, K. Loken, and M. Van de Panne. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics*, 2007.

[13] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *IJRR*, 2011.

[14] A. M. Johnson, T. Libby, E. Chang-Siu, M. Tomizuka, R. J. Full, and D. E. Koditschek. Tail assisted dynamic self righting. In *Adaptive Mobile Robotics*. World Scientific, 2012.

[15] M. Khoramshahi, H. J. Bidgoly, S. Shafiee, A. Asaei, A. J. Ijspeert, and M. N. Ahmadabadi. Piecewise linear spine for speed–energy efficiency trade-off in quadruped robots. *Robotics and Autonomous Systems*, 2013.

[16] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 2014.

[17] D. J. Hyun, J. Lee, S. Park, and S. Kim. Implementation of trot-to-gallop transition and subsequent gallop on the mit cheetah i. *IJRR*, 2016.

[18] M. Barragan, N. Flowers, and A. M. Johnson. MiniRHex: A small, open-source, fully programmable walking hexapod. In *RSS Workshop*, 2018.

[19] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *IROS*, 2018.

[20] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *IROS*, 2016.

[21] C. S. Imai, M. Zhang, Y. Zhang, M. Kierebinski, R. Yang, Y. Qin, and X. Wang. Vision-guided quadrupedal locomotion in the wild with multi-modal delay randomization. *arXiv:2109.14549*, 2021.

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

[23] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *ICRA*, 2018.

[24] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg. Dynamics randomization revisited: A case study for quadrupedal locomotion. In *ICRA*, 2021.

[25] O. Nachum, M. Ahn, H. Ponte, S. S. Gu, and V. Kumar. Multi-agent manipulation via locomotion using hierarchical sim2real. In *CoRL*, 2020.

[26] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 2019.

[27] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *RSS*, 2018.

[28] J. Hanna and P. Stone. Grounded action transformation for robot learning in simulation. In *AAAI*, 2017.

[29] W. Yu, J. Tan, C. K. Liu, and G. Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *RSS*, 2017.

[30] W. Yu, C. K. Liu, and G. Turk. Policy transfer with strategy optimization. In *ICLR*, 2018.

[31] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. In *RSS*, 2020.

[32] W. Zhou, L. Pinto, and A. Gupta. Environment probing interaction policies. In *ICLR*, 2019.

[33] W. Yu, V. C. V. Kumar, G. Turk, and C. K. Liu. Sim-to-real transfer for biped locomotion. In *IROS*, 2019.

[34] W. Yu, J. Tan, Y. Bai, E. Coumans, and S. Ha. Learning fast adaptation with meta strategy optimization. *RA-L*, 2020.

[35] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, and J. Tan. Rapidly adaptable legged robots via evolutionary meta-learning. In *IROS*, 2020.

[36] I. Clavera, A. Nagabandi, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *ICLR*, 2019.

[37] Z. Fu, A. Kumar, J. Malik, and D. Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *CoRL*, 2021.

[38] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. In *ICRA*, 2022.

[39] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. In *ICLR*, 2022.

[40] P. Fankhauser, M. Bloesch, and M. Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters*, 3(4):3019–3026, 2018.

[41] Y. Pan, X. Xu, Y. Wang, X. Ding, and R. Xiong. Gpu accelerated real-time traversability mapping. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 734–740, 2019. doi:10.1109/ROBIO49542.2019.8961816.

[42] I.-S. Kweon, M. Hebert, E. Krotkov, and T. Kanade. Terrain mapping for a roving planetary explorer. In *IEEE International Conference on Robotics and Automation*, pages 997–1002. IEEE, 1989.

[43] I.-S. Kweon and T. Kanade. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):278–292, 1992.

[44] A. Kleiner and C. Dornhege. Real-time localization and elevation mapping within urban search and rescue scenarios. *Journal of Field Robotics*, 24(8-9):723–745, 2007.

[45] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter. Navigation planning for legged robots in challenging terrain. In *IROS*, 2016.

[46] A. Chilian and H. Hirschmüller. Stereo camera based navigation of mobile robots on rough terrain. In *IROS*, 2009.

[47] C. Mastalli, I. Havoutis, A. W. Winkler, D. G. Caldwell, and C. Semini. On-line and on-board planning and perception for quadrupedal locomotion. In *2015 IEEE International Conference on Technologies for Practical Robot Applications*, 2015.

[48] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter. Robust rough-terrain locomotion with a quadrupedal robot. In *ICRA*, 2018.

[49] A. Agrawal, S. Chen, A. Rai, and K. Sreenath. Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries. *arXiv preprint arXiv:2110.00891*, 2021.

[50] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng. A control architecture for quadruped locomotion over rough terrain. In *ICRA*, 2008.

[51] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal. Learning locomotion over rough terrain using terrain templates. In *IROS*, 2009.

[52] L. Wellhausen and M. Hutter. Rough terrain navigation for legged robots using reachability planning and template learning. In *IROS*, 2021.

[53] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini. Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion. In *ICRA*, 2017.

[54] O. A. V. Magana, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini. Fast and continuous foothold adaptation for dynamic locomotion through cnns. *RA-L*, 2019.

[55] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter. Real-time optimal navigation planning using learned motion costs. In *ICRA*, 2021.

[56] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti. Learning ground traversability from simulations. *RA-L*, 2018.

[57] J. Guzzi, R. O. Chavez-Garcia, M. Nava, L. M. Gambardella, and A. Giusti. Path planning with local motion estimations. *RA-L*, 2020.

[58] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. Real-time trajectory adaptation for quadrupedal locomotion using deep reinforcement learning. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[59] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5 (2):3699–3706, 2020.

[60] X. B. Peng, G. Berseth, and M. Van de Panne. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.

[61] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36(4): 1–13, 2017.

[62] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne. Allsteps: Curriculum-driven learning of stepping stone skills. In *Computer Graphics Forum*, volume 39, pages 213–224. Wiley Online Library, 2020.

[63] D. Belter, P. Łabcki, and P. Skrzypczyński. Estimating terrain elevation maps from sparse and uncertain multi-sensor data. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 715–722, 2012. doi:10.1109/ROBIO.2012.6491052.

[64] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. Robot-centric elevation mapping with uncertainty estimates. 09 2014. doi:10.1142/9789814623353_0051.

[65] D. Jain, A. Iscen, and K. Caluwaerts. From pixels to legs: Hierarchical learning of quadruped locomotion. *arXiv preprint arXiv:2011.11722*, 2020.

[66] A. Escontrela, G. Yu, P. Xu, A. Iscen, and J. Tan. Zero-shot terrain generalization for visual locomotion policies. *arXiv preprint arXiv:2011.05513*, 2020.

[67] W. Yu, D. Jain, A. Escontrela, A. Iscen, P. Xu, E. Coumans, S. Ha, J. Tan, and T. Zhang. Visual-locomotion: Learning to walk on complex terrains with vision. In *5th Annual Conference on Robot Learning*, 2021.

[68] 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.

[69] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling vision and proprioception for navigation of legged robots. *arXiv preprint arXiv:2112.02094*, 2021.

Algorithm 1 Pytorch-style pseudo-code for phase 2

Require: Phase 1 policy $\pi^1 = (G^1, F^1, \beta)$, parallel environments E, max iterations M, truncated timesteps T, learning rate η
Initialize phase 2 policy $\pi^2 = (G^2, F^2, \gamma)$ with $G^2 \leftarrow G^1$, $F^2 \leftarrow F^1$.
$n \leftarrow 0$
while $n \neq M$ **do**
  Loss $l \leftarrow 0$
  $t \leftarrow 0$
  while $t \neq T$ **do**
    $s \leftarrow E.observations$
    $a^1 \leftarrow \pi^1(s)$
    $a^2 \leftarrow \pi^2(s)$
    $l \leftarrow l + \|a^1 - a^2\|_2^2$
    $E.step\left(a^2\right)$
    $t \leftarrow t + 1$
  end while
  $\Theta_{\pi^2} \leftarrow \Theta_{\pi^2} - \eta \nabla_{\Theta_{\pi^2}} l$
  $\pi^2 \leftarrow \pi^2.detach()$
  $n \leftarrow n + 1$
end while

A Observation Space

We denote the observation space by $\mathbf{o}_t^1 = \left(\mathbf{p}_t, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, v_x^{\text{cmd}}, w_z^{\text{cmd}}, \mathbf{m}_t\right)$:

- *Proprioception* contains (1) IMU information $\mathbf{p}_t = (\boldsymbol{\omega}_t, \boldsymbol{\theta}_t)$ i.e. the angular velocity $\boldsymbol{\omega}_t \in \mathbb{R}^3$ and roll, pitch values $\boldsymbol{\theta}_t \in \mathbb{R}^2$ of the robot base (2) joint angles $\mathbf{q}_t \in \mathbb{R}^{12}$ and velocities $\dot{\mathbf{q}}_t \in \mathbb{R}^{12}$ from the servo motors. We also include the last action taken $\mathbf{a}_{t-1} \in \mathbb{R}^{12}$ as this leads to smoother policies. Actions and joint positions are normalized to lie in the range $[-1, 1]$ with zero being the mean standing position of the robot.
- *Command* contains target linear velocity in x direction $\left(v_x^{\text{cmd}}\right)_t$ and yaw angular velocity $\left(\omega_z^{\text{cmd}}\right)_t$.
- *Elevation map* is an ego-centric map of the terrain in front of the robot that is close to the field of view of the camera. In particular, it consists of the height values $\mathbf{m}_t = \{h(x, y) \mid (x, y) \in \mathcal{P}\}$ at 88 points $\mathcal{P} = \{0.3, 0.2 \ldots 1.0\} \times \{-0.5, -0.4, \ldots, 0.5\}$. Note that we add small i.i.d. gaussian noise to all input observations (except $\mathbf{a}_{t-1}$) as specified in appendix.

B Rewards

Previous work [3, 69] has shown that task agnostic energy minimization based rewards can lead to the emergence of stable and natural gaits that obey high-level commands. We use this same basic reward structure along with penalties to prevent behavior that can damage the robot on complex terrain. Now onwards, we omit the time subscript t for simplicity.

- *Absolute work penalty* $-|\boldsymbol{\tau} \cdot \mathbf{q}|$ where $\boldsymbol{\tau}$ are the joint torques. We use the absolute value so that the policy does not learn to get positive reward by exploiting inaccuracies in contact simulation.
- *Command tracking* $v_x^{\text{cmd}} - \left|v_x^{\text{cmd}} - v_x\right| - |\omega_z^{\text{cmd}} - \omega_z|$ where v_x is velocity of robot in forward direction and ω_z is yaw angular velocity (x, z are coordinate axes fixed to the robot).
- *Foot jerk penalty* $\sum_{i \in \mathcal{F}} \|\mathbf{f}_t^i - \mathbf{f}_{t-1}^i\|$ where $\mathbf{f}_t^i$ is the force at time t on the i^{th} rigid body and $\mathcal{F}$ is the set of feet indices. This prevents large motor backlash.
- *Feet drag penalty* $\sum_{i \in \mathcal{F}} \mathbb{I}\left[f_z^i \geq 1\text{N}\right] \cdot \left(\left|v_x^i\right| + \left|v_y^i\right|\right)$ where $\mathbb{I}$ is the indicator function, and v_x^i, v_y^i is velocity of i^{th} rigid body. This penalizes velocity of feet in the horizontal plane if in contact with the ground preventing feet dragging on the ground which can damage them.

- *Collision penalty* $\sum_{i \in \mathcal{C} \cup \mathcal{T}} \mathbb{I}\left[\mathbf{f}^i \geq 0.1\text{N}\right]$ where $\mathcal{C}, \mathcal{T}$ are the set of calf and thigh indices. This penalizes contacts at the thighs and calves of the robot which would otherwise graze against edges of stairs and discrete obstacles.
- *Survival bonus* constant value 1 at each time step to prioritize survival over following commands in challenging situations.

C Phase 1 policy architecture

The elevation map $\mathbf{m}_t$ is passed through a two-layer MLP β with a $\tanh$ non-linearity on the output to get $\tilde{\mathbf{m}}_t \in \mathbb{R}^{32}$. This information bottleneck forces the network to learn a low dimensional representation of terrain that is less susceptible to noise. $\tilde{\mathbf{m}}_t$ is concatenated along with the rest of the observations and fed to a recurrent policy to obtain actions $\mathbf{a}_t$.

$$\begin{aligned}
\tilde{\mathbf{m}}_t &= \beta\left(\mathbf{m}_t\right) \\
\mathbf{h}_t, \mathbf{c}_t &= G^1\left(\mathbf{p}_t, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, v_x^{\text{cmd}}, w_z^{\text{cmd}}, \tilde{\mathbf{m}}_t \mid \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right) \\
\mathbf{a}_t &= F^1\left(\mathbf{h}_t\right)
\end{aligned}$$

where G^1 is a GRU with hidden and cell state $\mathbf{h}_t, \mathbf{c}_t$ respectively and F^1 is a two layer MLP with $\tanh$ output non-linearity.

D Phase 2 architecture

We create a copy of the recurrent base policy $G^2 \leftarrow G^1, F^2 \leftarrow F^1$. The elevation map compression MLP β is replaced with a convolutional depth backbone γ which processes depth map $\mathbf{d}_t \in \mathbb{R}^{58 \times 87}$ to produce a depth latent $\tilde{\mathbf{d}}_t \in \mathbb{R}^{32}$, predicted actions $\hat{\mathbf{a}}_t$ are then computed:

$$\begin{aligned}
\tilde{\mathbf{d}}_t &= \gamma\left(\mathbf{d}_t\right) \\
\mathbf{h}_t, \mathbf{c}_t &= G^2\left(\mathbf{p}_t, \mathbf{q}_t, \dot{\mathbf{q}}_t, \mathbf{a}_{t-1}, v_x^{\text{cmd}}, w_z^{\text{cmd}}, \tilde{\mathbf{d}}_t \mid \mathbf{h}_{t-1}, \mathbf{c}_{t-1}\right) \\
\hat{\mathbf{a}}_t &= F^2\left(\mathbf{h}_t\right)
\end{aligned}$$

E Experimental Setup and Implementation Details

Pseudo-code Phase 1 is simply reinforcement learning using policy gradients. We describe the pseudo-code for the phase 2 training in Algorithm 1.

Hardware We use the Unitree A1 robot pictured in Figure 2 of the main paper. The robot has 12 actuated joints, 3 per leg at hip, thigh and calf joints. The robot has a front-facing Intel RealSense depth camera in its head. The onboard compute consists of the UPboard and a Jetson NX. The UPboard has limited CPU compute and can command the motors while the NX has a small GPU and is connected to the camera. The UPboard and Jetson are on the same local network. Since depth processing is an expensive operation we run the convolutional backbone on the Jetson's GPU and send the depth latent over a UDP socket to the UPboard which runs the base policy. The policy operates at 50Hz and sends joint position commands which are converted to torques by a low-level PD controller running at 400Hz with stiffness $K_p = 40$ and damping $K_d = 0.5$.

Simulation Setup We use the IsaacGym (IG) simulator with the legged_gym library [7] to develop walking policies. IG can run physics simulation on the GPU and has a throughput of around 2e5 time-steps per second on a Nvidia RTX 3090 during phase 1 training with 4096 robots running in parallel. For phase 2, we can render depth using simulated cameras calibrated to be in the same position as the real camera on the robot. Since depth rendering is expensive and memory intensive, we get a throughput of 500 time-steps per second with 256 parallel environments. We run phase 1 for 15 billion samples (13 hours) and phase 2 for 6 million samples (6 hours).

Environment We construct a large elevation map with 100 sub-terrains arranged in a 20×10 grid. Each row has the same type of terrain arranged in increasing difficulty while different rows have different terrain. Each terrain has a length and width of 8m. We add high fractals (upto 10cm) on flat terrain while medium fractals (4cm) on others. Terrains are shown in Figure 5 with randomization ranges described in Table 2.

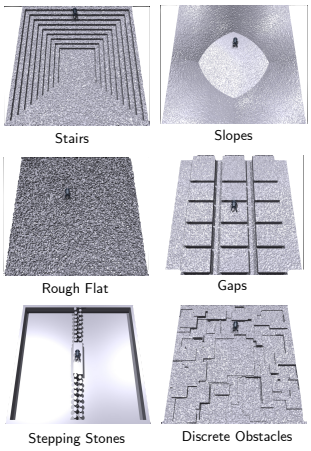


Figure 5: Set of terrain we use during training

Name	Range
Height map update frequency*	$[80\text{ms}, 120\text{ms}]$
Height map update latency*	$[10\text{ms}, 30\text{ms}]$
Added mass	$[-2\text{kg}, 6\text{kg}]$
Change in position of COM	$[-0.15\text{m}, 0.15\text{m}]$
Random pushes	Every 15s at 0.3m/s
Friction coefficient	$[0.3, 1.25]$
Height of fractal terrain	$[0.02\text{m}, 0.04\text{m}]$
Motor Strength	$[90\%, 110\%]$
PD controller stiffness	$[35, 45]$
PD controller damping	$[0.4, 0.6]$

Table 2: Parameter randomization in simulation. * indicates that randomization is increased to this value over a curriculum.

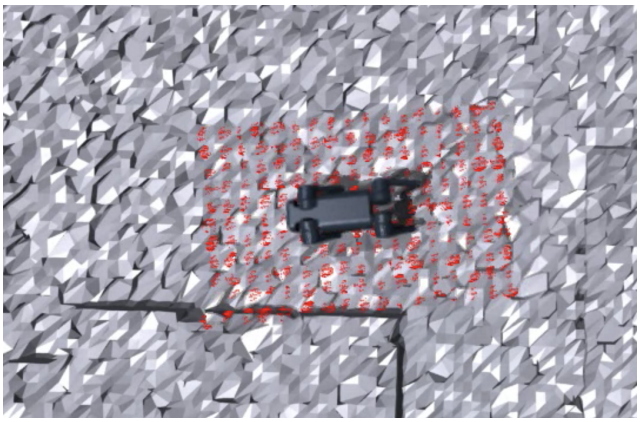


Figure 6: The privileged baseline receives terrain information from all around the robot including from around the hind feet.

Policy architecture The elevation map compression module β consists of an MLP with 2 hidden layers. The GRUs G^1, G^2 are single layer while the feed-forward networks F^1, F^2 have two hidden layers with ReLU non-linearities. The convolutional depth backbone γ consists of a series of 2D convolutions and max-pool layers.

References

[1] J. M. Loomis, J. A. Da Silva, N. Fujita, and S. S. Fukusima. Visual space perception and visually directed action. *Journal of experimental psychology: Human Perception and Performance*, 18(4):906, 1992.

[2] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter. Learning quadrupedal locomotion over challenging terrain. *Science robotics*, 2020.

[3] A. Kumar, Z. Fu, D. Pathak, and J. Malik. RMA: Rapid Motor Adaptation for Legged Robots. In *RSS*, 2021.

[4] F. Jenelten, T. Miki, A. E. Vijayan, M. Bjelonic, and M. Hutter. Perceptive locomotion in rough terrain–online foothold optimization. *RA-L*, 2020.

[5] 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.

[6] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim. Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot. In *ICRA*, 2020.

[7] N. Rudin, D. Hoeller, P. Reist, and M. Hutter. Learning to walk in minutes using massively parallel deep reinforcement learning. In *Conference on Robot Learning*, pages 91–100. PMLR, 2022.

[8] 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.

[9] H. Miura and I. Shimoyama. Dynamic walk of a biped. *IJRR*, 1984.

[10] M. H. Raibert. Hopping in legged systems—modeling and simulation for the two-dimensional one-legged case. *IEEE Transactions on Systems, Man, and Cybernetics*, 1984.

[11] H. Geyer, A. Seyfarth, and R. Blickhan. Positive force feedback in bouncing gaits? *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 2003.

[12] K. Yin, K. Loken, and M. Van de Panne. Simbicon: Simple biped locomotion control. *ACM Transactions on Graphics*, 2007.

[13] K. Sreenath, H.-W. Park, I. Poulakakis, and J. W. Grizzle. A compliant hybrid zero dynamics controller for stable, efficient and fast bipedal walking on mabel. *IJRR*, 2011.

[14] A. M. Johnson, T. Libby, E. Chang-Siu, M. Tomizuka, R. J. Full, and D. E. Koditschek. Tail assisted dynamic self righting. In *Adaptive Mobile Robotics*. World Scientific, 2012.

[15] M. Khoramshahi, H. J. Bidgoly, S. Shafiee, A. Asaei, A. J. Ijspeert, and M. N. Ahmadabadi. Piecewise linear spine for speed–energy efficiency trade-off in quadruped robots. *Robotics and Autonomous Systems*, 2013.

[16] A. D. Ames, K. Galloway, K. Sreenath, and J. W. Grizzle. Rapidly exponentially stabilizing control lyapunov functions and hybrid zero dynamics. *IEEE Transactions on Automatic Control*, 2014.

[17] D. J. Hyun, J. Lee, S. Park, and S. Kim. Implementation of trot-to-gallop transition and subsequent gallop on the mit cheetah i. *IJRR*, 2016.

[18] M. Barragan, N. Flowers, and A. M. Johnson. MiniRHex: A small, open-source, fully programmable walking hexapod. In *RSS Workshop*, 2018.

[19] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim. Mit cheetah 3: Design and control of a robust, dynamic quadruped robot. In *IROS*, 2018.

[20] M. Hutter, C. Gehring, D. Jud, A. Lauber, C. D. Bellicoso, V. Tsounis, J. Hwangbo, K. Bodie, P. Fankhauser, M. Bloesch, et al. Anymal-a highly mobile and dynamic quadrupedal robot. In *IROS*, 2016.

[21] C. S. Imai, M. Zhang, Y. Zhang, M. Kierebinski, R. Yang, Y. Qin, and X. Wang. Vision-guided quadrupedal locomotion in the wild with multi-modal delay randomization. *arXiv:2109.14549*, 2021.

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

[23] X. B. Peng, M. Andrychowicz, W. Zaremba, and P. Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *ICRA*, 2018.

[24] Z. Xie, X. Da, M. van de Panne, B. Babich, and A. Garg. Dynamics randomization revisited: A case study for quadrupedal locomotion. In *ICRA*, 2021.

[25] O. Nachum, M. Ahn, H. Ponte, S. S. Gu, and V. Kumar. Multi-agent manipulation via locomotion using hierarchical sim2real. In *CoRL*, 2020.

[26] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 2019.

[27] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *RSS*, 2018.

[28] J. Hanna and P. Stone. Grounded action transformation for robot learning in simulation. In *AAAI*, 2017.

[29] W. Yu, J. Tan, C. K. Liu, and G. Turk. Preparing for the unknown: Learning a universal policy with online system identification. In *RSS*, 2017.

[30] W. Yu, C. K. Liu, and G. Turk. Policy transfer with strategy optimization. In *ICLR*, 2018.

[31] X. B. Peng, E. Coumans, T. Zhang, T.-W. E. Lee, J. Tan, and S. Levine. Learning agile robotic locomotion skills by imitating animals. In *RSS*, 2020.

[32] W. Zhou, L. Pinto, and A. Gupta. Environment probing interaction policies. In *ICLR*, 2019.

[33] W. Yu, V. C. V. Kumar, G. Turk, and C. K. Liu. Sim-to-real transfer for biped locomotion. In *IROS*, 2019.

[34] W. Yu, J. Tan, Y. Bai, E. Coumans, and S. Ha. Learning fast adaptation with meta strategy optimization. *RA-L*, 2020.

[35] X. Song, Y. Yang, K. Choromanski, K. Caluwaerts, W. Gao, C. Finn, and J. Tan. Rapidly adaptable legged robots via evolutionary meta-learning. In *IROS*, 2020.

[36] I. Clavera, A. Nagabandi, S. Liu, R. S. Fearing, P. Abbeel, S. Levine, and C. Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. In *ICLR*, 2019.

[37] Z. Fu, A. Kumar, J. Malik, and D. Pathak. Minimizing energy consumption leads to the emergence of gaits in legged robots. In *CoRL*, 2021.

[38] L. Smith, J. C. Kew, X. B. Peng, S. Ha, J. Tan, and S. Levine. Legged robots that keep on learning: Fine-tuning locomotion policies in the real world. In *ICRA*, 2022.

[39] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang. Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers. In *ICLR*, 2022.

[40] P. Fankhauser, M. Bloesch, and M. Hutter. Probabilistic terrain mapping for mobile robots with uncertain localization. *IEEE Robotics and Automation Letters*, 3(4):3019–3026, 2018.

[41] Y. Pan, X. Xu, Y. Wang, X. Ding, and R. Xiong. Gpu accelerated real-time traversability mapping. In *2019 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 734–740, 2019. doi:10.1109/ROBIO49542.2019.8961816.

[42] I.-S. Kweon, M. Hebert, E. Krotkov, and T. Kanade. Terrain mapping for a roving planetary explorer. In *IEEE International Conference on Robotics and Automation*, pages 997–1002. IEEE, 1989.

[43] I.-S. Kweon and T. Kanade. High-resolution terrain map from multiple sensor data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):278–292, 1992.

[44] A. Kleiner and C. Dornhege. Real-time localization and elevation mapping within urban search and rescue scenarios. *Journal of Field Robotics*, 24(8-9):723–745, 2007.

[45] M. Wermelinger, P. Fankhauser, R. Diethelm, P. Krüsi, R. Siegwart, and M. Hutter. Navigation planning for legged robots in challenging terrain. In *IROS*, 2016.

[46] A. Chilian and H. Hirschmüller. Stereo camera based navigation of mobile robots on rough terrain. In *IROS*, 2009.

[47] C. Mastalli, I. Havoutis, A. W. Winkler, D. G. Caldwell, and C. Semini. On-line and on-board planning and perception for quadrupedal locomotion. In *2015 IEEE International Conference on Technologies for Practical Robot Applications*, 2015.

[48] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter. Robust rough-terrain locomotion with a quadrupedal robot. In *ICRA*, 2018.

[49] A. Agrawal, S. Chen, A. Rai, and K. Sreenath. Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries. *arXiv preprint arXiv:2110.00891*, 2021.

[50] J. Z. Kolter, M. P. Rodgers, and A. Y. Ng. A control architecture for quadruped locomotion over rough terrain. In *ICRA*, 2008.

[51] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal. Learning locomotion over rough terrain using terrain templates. In *IROS*, 2009.

[52] L. Wellhausen and M. Hutter. Rough terrain navigation for legged robots using reachability planning and template learning. In *IROS*, 2021.

[53] C. Mastalli, M. Focchi, I. Havoutis, A. Radulescu, S. Calinon, J. Buchli, D. G. Caldwell, and C. Semini. Trajectory and foothold optimization using low-dimensional models for rough terrain locomotion. In *ICRA*, 2017.

[54] O. A. V. Magana, V. Barasuol, M. Camurri, L. Franceschi, M. Focchi, M. Pontil, D. G. Caldwell, and C. Semini. Fast and continuous foothold adaptation for dynamic locomotion through cnns. *RA-L*, 2019.

[55] B. Yang, L. Wellhausen, T. Miki, M. Liu, and M. Hutter. Real-time optimal navigation planning using learned motion costs. In *ICRA*, 2021.

[56] R. O. Chavez-Garcia, J. Guzzi, L. M. Gambardella, and A. Giusti. Learning ground traversability from simulations. *RA-L*, 2018.

[57] J. Guzzi, R. O. Chavez-Garcia, M. Nava, L. M. Gambardella, and A. Giusti. Path planning with local motion estimations. *RA-L*, 2020.

[58] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis. Real-time trajectory adaptation for quadrupedal locomotion using deep reinforcement learning. In *International Conference on Robotics and Automation (ICRA)*, 2021.

[59] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter. Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning. *IEEE Robotics and Automation Letters*, 5(2):3699–3706, 2020.

[60] X. B. Peng, G. Berseth, and M. Van de Panne. Terrain-adaptive locomotion skills using deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016.

[61] X. B. Peng, G. Berseth, K. Yin, and M. Van De Panne. Deeploco: Dynamic locomotion skills using hierarchical deep reinforcement learning. *ACM Transactions on Graphics (TOG)*, 36 (4):1–13, 2017.

[62] Z. Xie, H. Y. Ling, N. H. Kim, and M. van de Panne. Allsteps: Curriculum-driven learning of stepping stone skills. In *Computer Graphics Forum*, volume 39, pages 213–224. Wiley Online Library, 2020.

[63] D. Belter, P. Łabcki, and P. Skrzypczyński. Estimating terrain elevation maps from sparse and uncertain multi-sensor data. In *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 715–722, 2012. doi:10.1109/ROBIO.2012.6491052.

[64] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart. Robot-centric elevation mapping with uncertainty estimates. 09 2014. doi:10.1142/9789814623353_0051.

[65] D. Jain, A. Iscen, and K. Caluwaerts. From pixels to legs: Hierarchical learning of quadruped locomotion. *arXiv preprint arXiv:2011.11722*, 2020.

[66] A. Escontrela, G. Yu, P. Xu, A. Iscen, and J. Tan. Zero-shot terrain generalization for visual locomotion policies. *arXiv preprint arXiv:2011.05513*, 2020.

[67] W. Yu, D. Jain, A. Escontrela, A. Iscen, P. Xu, E. Coumans, S. Ha, J. Tan, and T. Zhang. Visual-locomotion: Learning to walk on complex terrains with vision. In *5th Annual Conference on Robot Learning*, 2021.

[68] 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.

[69] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak. Coupling vision and proprioception for navigation of legged robots. *arXiv preprint arXiv:2112.02094*, 2021.