Efficient Audio-Visual Navigation via Reinforcement Learning

Zihui Xue
zx2878
sherryxue@utexas.edu

Yuedong Yang
yy9466
albertyoung@utexas.edu

Abstract: This project investigates the task of audio-visual navigation, where the agent is provided with visual observations (*e.g.*, depth images) and audio cues (*e.g.*, a phone ringing) and asked to navigate to a goal location. To start with, we evaluate the role of the visual and audio modality by comparing a visual agent, an audio agent and an audio-visual agent. Next, we aim to improve the energy efficiency of an audio-visual agent. Motivated by the fact that depth sensors usually consume more power than microphones, we propose to turn on the depth sensor of the agent only when it is necessary. We train a naive audio-visual agent as the baseline and an intelligent audio-visual agent that decides when to turn on the depth sensor itself. We conduct experiments on the Replica dataset and show great performance of our designed energy-efficient audio-visual agent. [1]

1 Introduction

Recent years have witnessed great progress of research on embodied agents operating in simulated environments for various tasks. Among them, *navigation* is of particular importance due to its high practical value and various applications. The PointGoal Navigation task [1] requires an agent to navigate to target coordinates that are relative to the agent's position. With no ground-truth map of the environment provided, the agent takes actions based on its sensory signals (*e.g.*, RGB images, depth images and GPS+Compass signals).

Reinforcement Learning (RL) is one prevalent approach in solving this task. Decentralized Distributed Proximal Policy Optimization (DD-PPO) [2] achieves near perfect navigation performance on unseen environments. However, the visual agent only adopts visual and location information, and is deaf to the environment. Joining the rapid progress of audio-visual learning [3], one recent line of work [4, 5, 6] investigates audio-visual navigation in 3D environments. Soundspaces [4] define the AudioPointGoal task, where the agent not only receives relative coordinates of the target location, but also hears an audio (*e.g.*, phone ringing) emitted from the goal.

In this project, we aim to investigate audio-visual navigation thoroughly and answer the following two questions: (1) Is the additional audio modality beneficial for the navigation task? (2) What's the contribution of the visual modality in audio-visual navigation? Towards this goal, we implement a visual agent, an audio agent and an audio-visual agent with Proximal Policy Optimization (PPO) [7] and compare their performance. Our results demonstrate that the audio-visual agent achieves satisfactory performance.

Motivated by the positive influence of audio information in navigation, we proceed to explore the topic from the perspective of computational and energy efficiency. Since the audio modality provides rich information to the agent and introduces lighter computations compared with the visual modality, can we occasionally turn off the agent's camera for computation and energy savings? We envision that embodied AI systems will run on mobile devices such as autonomous robots and drones with

[1] Presentation video available at https://www.youtube.com/watch?v=LOrfifPg8eE, code available at https://github.com/zihuixue/RLProj.

limited compute resources and battery life. Such application scenarios call for solutions that are not only accurate, but also computationally cheap.

In this project, we also made our attempts towards the topic of efficient audio-visual navigation. To be more precise, we design an audio-sparse-visual agent that takes sounds and temporally sparse depth images as the input. Instead of having continuous depth images, our agent "occasionally" turns on its camera to receive depth images; audio information is available at every step. Experiments on the Replica dataset demonstrate great performance of our proposed energy-efficient audio-visual agent.

2 Related Work

2.1 Navigation

Navigation has been well-studied with many prior works. These works can be classified into two categories: two-stage method and single-stage method.

Two-stage methods divide the navigation problem into mapping and planning two stages, and study them separately. In mapping stage, robot builds an accurate metric map or topological map of the environment with simultaneously localization and mapping (SLAM) algorithms. [8, 9, 10] build metric maps containing point cloud of feature points; [11, 12] build geological topology map representing the environment with a graph where nodes represent empty spaces and edges represent connectivity; [13, 14, 15] build object-centric semantic maps where landmarks are semantically meaningful objects. Recently, [16, 17, 18] combines all these three types of maps and produce a five-level hierarchical map. In planning stage, robots schedule a path towards the destination in the map built in previous stage with searching algorithms like A-star [19] and its variants[20]. [21] provides a thorough survey of these works.

One stage methods aim to find the target automatically via active exploration of the environment. Classic methods [22, 11, 23] use hand-crafted heuristics (e.g. searching the closest frontier) to guide the exploration. These methods cannot understand the environment and search aimlessly. Recently, with the development of robot learning, deep neural networks are introduced for a more efficient search by utilizing the semantic information. [24] explores and navigates to the target with a differentiable mapper and a differentiable planner running in parallel. The mapper updates the belief about the world and the planner decide the next exploration location until the target is found. [25] extends [24] by defining the ultimate target as the "long-term goal" and introducing a series of "short-term goal" during exploration for guidance.

2.2 Audio-visual Navigation

Human perceive the world in a multimodal way, *e.g.*, through vision, sound, touch, etc. While current robots adopt visual information (*i.e.*, RGB and depth images) for navigation, they are deaf to the environment. In fact, sound is an important source of information and often provide critical cues in real applications. Motivated by the finding, Chen *et al.* [4] introduce the task of audio-visual navigation, where the robot receives visual and audio sensory data for navigation. They design a deep learning architecture that maps audio, visual and GPS observations to agent actions and train the network with PPO. Gan *et al.* [5] propose to utilize sound to infer the goal location and then follow a path planner that gives a sequence of actions based on the agent's observations. Chen *et al.* [6] propose a framework that learns to set waypoints for audio-visual navigation with an acoustic memory.

3 Method

3.1 Task Setup

We consider the navigation task [26] - an agent is initialized at a random starting position and orientation in an environment and required to navigate to target coordinates; no ground-truth map is available. The agent needs to use its sensory input for navigation. We list four navigation tasks below, categorized based on the agent's input modality.

(1) PointGoal: The agent is provided with an idealized GPS and compass so that it can access their location coordinates. The agent's relative distance to the starting position is provided in the form of a displacement vector (Δ_x, Δ_y). In addition, the agent is equipped with a depth sensor placed at a height of 1.5m from the center of the agent's base and oriented forward; the sensor provides depth images of size 256 × 256 and has a 90 degree field of view.

(2) AudioPointGoal-Blind: The agent is not provided with depth sensors but still receives the displacement vector for navigation. In addition, the agent hears audio from a sounding target and the audio is updated as the agent moves.

(3) AudioPointGoal: The agent is equipped with GPS+compass, depth sensors as well as microphones to receive audio input.

(4) AudioPointGoal-Sparse: The agent is still equipped with all three sensors, but we encourage the agent to only turn on its depth sensor when necessary.

Note that for simplicity, we do not consider agents with RGB sensors. As shown in previous works [4], the agent equipped with a depth sensor performs better in navigation than the agent with an RGB sensor. Therefore, we choose depth images to represent the visual modality in our study.

The first three tasks have been studied in previous works [1, 4], and we follow their task settings. For our proposed AudioPointGoal-Sparse task, we present two solutions.

To begin with, we train a naive audio-sparse-visual agent; we uniformly sample depth frames along the temporal dimension (*i.e.*, feed the agent a depth image every T steps, T being a constant value). Thus the agent's action space and reward are identical to a regular audio-visual agent.

Next, we aim to make our audio-sparse-visual agent more intelligent. We increase its action space by adding an action "turn on the depth sensor" to enable it to receive depth images in the next step. Formally, the action space consists of eight actions: *Turn left*, *Turn right*, *Move forward*, *Stop*, *Turn left and turn on depth sensor*, *Turn right and turn on depth sensor*, *Move forward and turn on depth sensor*, *Stop and turn on depth sensor*. The action space of a regular audio-visual agent and our naive audio-sparse-visual agent corresponds to the first four actions. The turning actions result in 10 degree turns and *Move Forward* takes the agent forward by 0.5m.

In terms of the reward, for the naive audio-sparse-visual agent, we follow previous works [1, 4] and use the dense reward function below:

$$r_t = d_{t-1} - d_t + \lambda \tag{1}$$

where r_t denotes the reward at time t and d_t is the geodesic distance to goal at time t. λ is a constant slack penalty to encourage efficient paths, and we choose it to be -0.01. An episode ends when the agent takes the stop action and an additional +10 reward is gained if the agent is within 2m of the goal. For the audio-sparse-visual agent, it needs to decide whether to turn on its depth sensor or not. To prevent it from always choosing the *Turn on* action and to encourage sparsity, we add a penalty term of -0.3 in the reward if the agent chooses to turn on its depth sensor.

We consider two evaluation metrics: (1) Success. Percentage of episodes in which the agent calls stop within 0.2m of the goal location. (2) Success weighted by inverse normalized Path Length (SPL). SPL accounts for both success and path efficiency by weighting a binary success indicator by the normalized ratio of the agent's path length and the shortest path.

An episode is specified based on the scene, the agent's start location, orientation and its goal location. Additionally, for tasks where agents are equipped with microphones, a source audio waveform is specified in an episode.

3.2 Model Training

Inspired by the remarkable performance PPO achieves in navigation tasks [2], we also train our audio-sparse-visual agents with PPO.

Input format. Our audio-sparse-visual agent takes input from three modalities: (1) an audio provided in the form of spectrograms, we obtain two tensors of shape $65 \times 65 \times 2$ and $65 \times 26 \times 2$; (2) a depth image of size $128 \times 128 \times 1$; (3) a 2-dimensional displacement vector showing the agent's relative distance to the goal location.

Policy model architecture. Since an agent receives multimodal input of different shapes, we adopt a fusion architecture as in [4]. First, we have three separate encoders to map observations to a common representation space. We use an audio convolution neural network [27] (CNN), an image CNN and an identity mapping for audio, visual and location input, respectively. Then we concatenate multimodal features and apply a gated recurrent unit [28] (GRU). Last, we use two linear layers for actor and critic outputs of the model, which correspond to policy distribution and state value, respectively.

4 Results

4.1 Experimental Setup

We evaluate our method on the Replica dataset [29]. It consists of 18 highly-photo-realistic (HDR) indoor scene reconstructions at room and building scale. Moreover, we include audio renderings provided in Soundspaces [4]. The number of training and test episodes is $0.1M$ and 1000, respectively.

The experiments are conducted on a desktop with with AMD Threadripper 3970X CPU and two NVIDIA RTX 3090 GPUs. We implement the policy network with PyTorch [30]. The optimizer is Adam [31] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. Learning rate is set as $2.5e - 4$ and rewards are discounted with a decay of 0.99. We train agents for 4M steps. We use Habitat[2] as our simulation platform and implement our code based on SoundSpaces[3].

4.2 Results

We first investigate how audio and visual modality can help in navigation. Table 1 presents a comparison of an audio agent, a visual agent and an audio-visual agent. All agents are equipped with a GPS+compass sensor (*i.e.*, have access to their relative coordinates to the goal).

Table 1: A comparison of PointGoal, AudioPointGoal-Blind and AudioPointGoal.

Task	Sensor Modality	Success	SPL
PointGoal	Depth	0.822	0.711
AudioPointGoal-Blind	Audio	0.740	0.552
AudioPointGoal	Depth + Audio	0.804	0.722

From the table, we can clearly see that the visual modality plays an important role in navigation: the audio agent (*i.e.*, AudioPointGoal-Blind) performs considerably worse than the other two agents. While the audio-visual agent (*i.e.*, AudioPointGoal) indeed leads to a higher SPL compared with

[2] https://github.com/facebookresearch/habitat-lab

[3] https://github.com/facebookresearch/sound-spaces

the visual agent (*i.e.*, PointGoal), the improvement is not so significant. The results indicate that our audio-visual agent does not learn to make informative decisions from audio input very well. This may be related to insufficient training time. As our compute resources is limited and running one experiment is very time-consuming, we run a limited number of steps for each agent. Another possible reason is our current policy model architecture does not combine multimodal observations very well. In the future, we plan to further evaluate this problem and see if there's improvement space for an audio-visual agent. We will conduct experiments on other datasets, repeat experiment runs and redesign our policy network to better fuse multimodal features.

Next, we evaluate audio-sparse-visual agents for our proposed AudioPointGoal-Sparse task. To be specific, we train three agents: (1) a naive audio-sparse-visual agent that receives depth observations every 2 steps; we term it as *naive audio-sparse-visual-2*; (2) a naive audio-sparse-visual agent that receives depth observations every 5 steps, denoted by naive *audio-sparse-visual-5*; (3) an intelligent *audio-sparse-visual* agent that incorporates "turn on the depth sensor" in its action space. The results are summarized in Table 2.

Table 2: A comparison of PointGoal, AudioPointGoal-Blind and AudioPointGoal.

Task	Agent	Success	SPL
AudioPointGoal	*audio-visual*	0.804	0.722
AudioPointGoal-Sparse	*naive audio-sparse-visual-2*	0.786	0.693
AudioPointGoal-Sparse	*naive audio-sparse-visual-5*	0.650	0.545
AudioPointGoal-Sparse	*audio-sparse-visual*	0.842	0.710

Considering the two naive agents, we observe performance degradation, especially for *naive audio-sparse-visual-5*. Since the agent can only access one depth image every 5 steps, it does not learn to utilize depth images for action well. The *naive audio-sparse-visual-2* has a slighter lower SPL (*i.e.*, -0.029) than a regular audio-visual agent. The slight decrease indicates that depth images may not be needed at every step; this finding may be utilized for deployment benefits.

In addition, Table 2 demonstrates great performance of our proposed *audio-sparse-visual* agent. It outperforms the two naive agents and only requires a "turn on depth sensor" rate of 0.246. This is similar to the sampling frequency of *audio-sparse-visual-5* (*i.e.*, 0.2), yet we observe greatly increased SPL (*i.e.*, from 0.545 to 0.710). Moreover, compared with the regular audio-visual agent, our agent has a slightly slower SPL but only requires depth samples approximately every 4 steps. Note that the "turn on" action is determined by the agent and depth observations are not uniformly fed to the agent. These results demonstrate potentials of our approach in efficient audio-visual navigation.

5 Conclusion

The project focuses on audio-visual navigation with RL. We aim to evaluate the contribution of the audio and visual modality in navigation and implement an audio agent, a visual agent and an audio-visual agent with PPO. Moreover, we propose a new task AudioPointGoal-Sparse from the perspective of computational and energy efficiency. We design agents that occasionally turn on the camera sensor for observations in an attempt to save computation and battery on edge devices. We conduct experiments on the Replica dataset and compare different agents we train. The results demonstrate potentials of efficient audio-visual navigation.

References

[1] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[2] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra. DD-PPO: Learning near-perfect pointgoal navigators from 2.5 billion frames. In *International Conference on Learning Representations (ICLR)*, 2020.

[3] H. Zhu, M.-D. Luo, R. Wang, A.-H. Zheng, and R. He. Deep audio-visual learning: A survey. *International Journal of Automation and Computing*, 18(3):351–376, 2021.

[4] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman. Soundspaces: Audio-visual navigaton in 3d environments. In *ECCV*, 2020.

[5] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 9701–9707. IEEE, 2020.

[6] C. Chen, S. Majumder, Z. Al-Halah, R. Gao, S. K. Ramakrishnan, and K. Grauman. Learning to set waypoints for audio-visual navigation. *ICLR*, 2021.

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

[8] R. Mur-Artal and J. D. Tardós. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE transactions on robotics*, 33(5):1255–1262, 2017.

[9] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. Montiel, and J. D. Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

[10] P. Geneva, K. Eckenhoff, W. Lee, Y. Yang, and G. Huang. Openvins: A research platform for visual-inertial estimation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4666–4672. IEEE, 2020.

[11] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto. Sparse 3d topological graphs for micro-aerial vehicle planning. In *International Conference on Intelligent Robots and Systems (IROS)*, pages 1–9. IEEE, 2018.

[12] A. A. Ravankar, A. Ravankar, T. Emaru, and Y. Kobayashi. A hybrid topological mapping and navigation method for large area robot mapping. In *Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 1104–1107. IEEE, 2017.

[13] L. Nicholson, M. Milford, and N. Sünderhauf. Quadricslam: Dual quadrics from object detections as landmarks in object-oriented slam. *IEEE Robotics and Automation Letters*, 4(1):1–8, 2018.

[14] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss. Suma++: Efficient lidar-based semantic slam. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4530–4537. IEEE, 2019.

[15] S. Yang and S. Scherer. Cubeslam: Monocular 3-d object slam. *IEEE Transactions on Robotics*, 35(4):925–938, 2019.

[16] Y. Chang, Y. Tian, J. P. How, and L. Carlone. Kimera-multi: a system for distributed multi-robot metric-semantic simultaneous localization and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 11210–11218. IEEE, 2021.

[17] A. Rosinol, M. Abate, Y. Chang, and L. Carlone. Kimera: an open-source library for real-time metric-semantic localization and mapping. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1689–1696. IEEE, 2020.

[18] A. Rosinol, A. Gupta, M. Abate, J. Shi, and L. Carlone. 3d dynamic scene graphs: Actionable spatial perception with places, objects, and humans. *arXiv preprint arXiv:2002.06289*, 2020.

[19] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

[20] A. Stentz. Optimal and efficient path planning for partially known environments. In *Intelligent unmanned ground vehicles*, pages 203–220. Springer, 1997.

[21] B. Patle, A. Pandey, D. Parhi, A. Jagadeesh, et al. A review: On path planning strategies for navigation of mobile robot. *Defence Technology*, 15(4):582–606, 2019.

[22] M. Keidar and G. A. Kaminka. Robot exploration with fast frontier detection: Theory and experiments. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 113–120, 2012.

[23] C. Wang, H. Ma, W. Chen, L. Liu, and M. Q.-H. Meng. Efficient autonomous exploration with incrementally built topological map in 3-d environments. *IEEE Transactions on Instrumentation and Measurement*, 69(12):9853–9865, 2020.

[24] S. Gupta, J. Davidson, S. Levine, R. Sukthankar, and J. Malik. Cognitive mapping and planning for visual navigation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017.

[25] D. S. Chaplot, D. Gandhi, S. Gupta, A. Gupta, and R. Salakhutdinov. Learning to explore using active neural slam. *arXiv preprint arXiv:2004.05155*, 2020.

[26] P. Anderson, A. Chang, D. S. Chaplot, A. Dosovitskiy, S. Gupta, V. Koltun, J. Kosecka, J. Malik, R. Mottaghi, M. Savva, et al. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018.

[27] K. O'Shea and R. Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[29] J. Straub, T. Whelan, L. Ma, Y. Chen, E. Wijmans, S. Green, J. J. Engel, R. Mur-Artal, C. Ren, S. Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[30] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[31] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.