Accessible AI in Robotic Navigation

Samantha Wurm[1]*, Beatrice Casey[2], Brendan O'Toole[3]

Abstract
We present Project SPIKE – an artificially intelligent robot trained to navigate physical spaces by viewing them in virtual simulations on Microsoft's platform: Project Malmo. As the AI research community pushes for artificial general intelligence (AGI), economic, time-sensitive, and accessible experimentation platforms are needed that support the development of intelligent machines that learn to solve real-life tasks in virtual environments. Using free, open-source software like Project Malmo and ACT-R to train our agent provides a suitable virtual environment to train the robot's cognition while avoiding the time burden of running physical trials to collect training-set data. Project SPIKE allows for accessibility in the robotics community as all of the software and hardware that we used can be purchased for less than $1200, thus dramatically lowering the barrier to entry of working in this field. In this demo we present the development of our robot, SPIKE, and its capabilities. We created our robot using an iRobot in tandem with the Jetson TX2 and a ZED camera. We gave our robot navigation capabilities using Project Malmo, Act-R, and TensorFlow.

Keywords
Artificial Intelligence — Accessibility — Robotics

[1] *Department of Computer Science, Bucknell University, Pennsylvania, United States*
[2] *Department of Computer Science, Bucknell University, Pennsylvania, United States*
***Corresponding author**: sew025@bucknell.edu

Contents

1. Cost Analysis

As one of the main purposes of Project SPIKE is to increase economic accessibility to working in embodied AI, we chose parts that were widely available and relatively inexpensive as compared to other models currently used in the field. The software includes Project Malmo (free, open source), ACT-R (free, open source), and TensorFlow (free, open source). The hardware includes the NVIDIA Jetson TX2 Developer Kit ($599), the Stereolabs ZED camera ($349), and the iRobot Create 2 ($199). The total cost of all of the software and hardware for building this artificially intelligent system is $1,147 (prices pulled December 16th, 2021). The current estimated cost of building an AI robot in the US starts at $20,000. This is a decent decrease, and could be helpful to those in the field with less resources readily available.

2. Hardware

In this paper, we display the opportunities created by integrating multiple hardware systems together. In this section, each piece of hardware will be explained individually, in order to further elucidate how it, specifically, contributes to the larger robotic capabilities that result after integration. 1. NVIDIA Jetson TX2 Developer Kit: A "supercomputer". It is the most powerful "embedded AI computing device" [NVIDIA Developer] This is the brain of SPIKE as a whole. It is where the training takes place and is the entity that determines how SPIKE will move. 2. Stereolabs ZED camera: This camera mimics human eyes, and is thus the eyes of SPIKE. It takes in the data around it and captures depth and motion allowing SPIKE (more specifically the Jetson) to have more information about the world around it. 3. iRobot Create 2: The iRobot is essentially a roomba without the vacuum. It is therefore able to navigate a space forwards, backwards, etc. This is the 'body' of SPIKE. It is what will be carrying it throughout the space. The iRobot receives instructions from the Jetson as to how it will move around.

3. Building the Robot

Considering the importance of each of these parts, as described, the process of building robot SPIKE is a delicate one.

The steps that must be followed in order to assemble the robot go as follows: All of these parts need to be connected. So, it is necessary to build some type of structure that will house each of these individual elements. While there are many ways to do this, the simplest (and most cost-effective) would be to cut a wooden plate with a circumference equivalent to the iRobot. Initially, SPIKE was to have two 'layers', or plates, however it was found to be too unstable and unbalanced. One plate or layer makes SPIKE much sturdier. Two metal rods are needed to hold up the wooden plate above the iRobot. The metal rods used are square and have an 'indent' in the middle that allows for the head of a screw to slide through this cavity. Additionally, they have a hollow center (again, where a single screw can fit through). Additionally, L-brackets are needed to attach the metal rods to the plates. These L brackets slide in the cavity of the metal rod. As mentioned above, screws are necessary to actually hold the robot together. The iRobot has designated places where one can drill a hole through its plastic covering. Using these places, mark the wooden plate so that it lines up with them and using screws that fit the L-brackets, drill holes to attach the L-brackets to the wooden plate in the specified places. To make sure the L-brackets are solidly connected to the metal rods, use a longer screw (NOT self-tapping) and screw it in tightly to the side of the L-bracket that is connected to the metal rod such that the end of the screw is pressed firmly against the rod and any attempt to slide the rod off the L-bracket will fail. This completes creating the stand. To connect this stand to the robot, remove the plastic cover on the iRobot and drill a hole in the designated places (that match the places where the holes on the wooden panel were drilled). Using long, non self-tapping screws, twist the screw through these holes so they are poking straight up from the plastic cover. Re-attach the plastic covering to the iRobot. It might be a tight fit based on the size of the screw head used, but it should be able to snap back on regardless. Now, using the hollow center of the rods, slide the stand onto the screws. This completes the assembly of the structure. Some things to note are: Hold SPIKE at the bottom (i.e. by the iRobot), given that the screws are not holding it down in place The structure might 'wobble' slightly. To fix this, you would need to get a screw with a bigger diameter such that it fits tightly and exactly in the center of the metal rod. To assemble SPIKE, place the ZED camera on the wooden plate, closer to the edge so that the plate does not get in the view of the ZED. Place the Jetson on top of the iRobot itself. Most likely it will be necessary to zip-tie wires to the metal rods so the robot does not trip on them and so they are out of the way. It will be necessary to also have a USB hub. The Jetson only has one USB-3 port, however the ZED and the iRobot have to connect to it via USB-3. Having additional ports on the hub will also make it convenient for any other attachments one would wish to add.

4. Project Malmo

In this demo, project SPIKE is a single-agent system that takes in information about its surroundings and acts in accordance with simple movement goals. The Project Malmo platform is beneficial for training SPIKE, as it was designed to support research in robotic experimentation and AI-agent-training specifically. It provides a realistic, sizable, dynamic environment for an agent to be trained in, and in sum implements the important characteristics of a successful AGI environment, as outlined in "The Malmo Platform for Artificial Intelligence Experimentation" [Matthew Johnson, Katja Hofmann, Tim Hutton, David Bignell], and in "AGI Environments, Tasks, and Agents" [Laird and Wray III, 2010], refined by [Adams et al., 2012], detailed below. C1. The environment is complex, with diverse, interacting and richly structured objects. This is supported by exposing the full, rich structure of the Minecraft game. C2. The environment is dynamic and open. The platform supports infinitely-varied environments and missions, including, e.g., navigation, survival, and construction. C3. Task-relevant regularities exist at multiple time scales. Like real-world tasks, missions in Malmo can have complex structure, e.g. A construction project requires navigation, mining resources, composing structures, etc. C4. Other agents impact performance. Both AI-AI and human-AI interaction (and collaboration) are supported. C5. Tasks can be complex, diverse and novel. New tasks can be created easily, so the set of possible tasks is infinite. C6. Interactions between agent, environment and tasks are complex and limited. Perception and action couple environment and agents. Several abstraction levels are provided to vary complexity within this framework. C7. Computational resources of the agent are limited. Real-time interaction naturally constrains available resources. Additional constraints can be imposed if required. C8. Agent existence is long-term and continual. This is naturally provided by persistent Minecraft worlds, supporting long-term agent development and lifelong learning. The above requirements were paramount to consider when choosing the virtual environment of training for SPIKE due to the guiding design principle that governed the ideation and development of our robot: accessibility. Because Project Malmo works cross-platform, cross-language, uses popular data formats, and is open-source, it is easily accessible to individuals with all different tools or skills available at their fingertips. Furthermore, because the Project SPIKE documentation and code will also be open source, the accessibility and transparency of AI-robotics will be further underlined by this implementation. Moreover, in this demo, SPIKE was built to navigate the computer science lab in Bucknell University's building Dana room 116. Therefore, we created a simulation (to-size) of this room, which SPIKE explored as a virtual agent and learned how to navigate. We created SPIKE as a simple virtual agent in Project Malmo, with no goals or special capabilities. In order to create an environment that could mimic a space in which there are dynamic changes, moving furniture, and possibly roaming people, we needed an easily adaptable and complex environment platform. When creating our room in Project Malmo, we did not include the precise locations of each piece

of furniture or person, because SPIKE needs to learn higher level rules in order to constantly tackle novel challenges and integrate new information into its system. In order to implement a decision-making system for SPIKE to navigate the space that we created in Project Malmo, as a Project Malmo agent, we used the cognitive architecture ACT-R.

5. ACT-R

ACT-R is a cognitive architecture that mimics human thinking. It uses "task knowledge to perform a task thereby predicting and explaining the steps of cognition that form human behavior" [Ritter et al., 2018]. ACT-R uses Buffers to model cognition. It has buffers for visual, working and procedural memory, and buffers to see and act upon the model's environment. These buffers contain characteristics that are similar to human cognition, such as decay in memory (e.g. when not using or thinking about something for a while, a human tends to forget or not remember that thing), what can happen between buffers and other things. There are Productions, which creates rules. It matches an if-then pattern: if a rule is matched, then take this action. Productions resemble procedural task knowledge. Chunks represent declarative memory objects and are represented in ACT-R as symbols. These symbols can have values and attributes, and the more often it is accessed, the less time it takes the next time it is accessed. This represents human cognition in that the more we learn or think about a subject, the easier it is to remember or execute this knowledge [Ritter et al., 2018]. Connecting this to SPIKE, ACT-R helps it make decisions using Productions (if there is a wall, turn right/left) and chunks to allow SPIKE to navigate a room and learn about its environment. The more often SPIKE sees a wall, the more often that chunk is activated and thus the less time it would take for it to know which way to turn and what to do.

6. Training a Robotic Agent

In order to train our agent, we needed to integrate the work that we had done in Project Malmo and in ACT-R. After creating our virtual environment and agent in Project Malmo, and implementing cognitive rules and learning skills in ACT-R, we taught SPIKE how to create and implement rules for navigating space. By training the robot in a virtual environment, we could run thousands of trials for SPIKE to navigate virtual space, learn and modify movement rules, and reframe in very little time. Once SPIKE could navigate the virtual room in a sensible manner without bumping into things, we could set up SPIKE to navigate the physical room that we were working in on the iRobot base. Rules that our robot followed include: When there is a wall, turn left; When there is an object in front of you, continue to turn right until there is no object in front of you, then go straight; If there is a door, do not go through it but turn left.

7. Cloud Point Software

Utilizing Stereolabs' ZED camera, SPIKE was capable of gathering and analyzing input visual data. The data which SPIKE captured included: side-by-side RGB images, depth maps, and point cloud data. This data was obtained using Sterolabs' ZED SDK via Python. Point cloud data was the data SPIKE primarily focused on as the visual representation of the environment. This data was stored as Polygonal files (.ply). Utilizing Open3D's 3D image processing library in Python, the polygonal data were able to be cleaned, transformed, and analyzed. Open3D provided a library of functions to clean the data through use of statistical outlier functionality. Open3D also provided functionality to analyze the data and implement processes such as support identification and object detection.

8. Cognitively Inspired Robotics

Through the use of the Open3D Python library, SPIKE was able to process point cloud data and derive symbolic representations of the robots environment which could be used in a cognitively inspired fashion to generate actions. Processing the data allowed for a few main outputs. First, rather than using standard depth measurements to understand distance, SPIKE utilized functionality to identify planes with the largest support within the data. This allowed SPIKE to identify surrounding walls and other large obstacles. Second, SPIKE utilized density based clustering to group local point cloud clusters. This allowed SPIKE to begin to identify "objects" as smaller clusters of point cloud nodes.

9. Integration

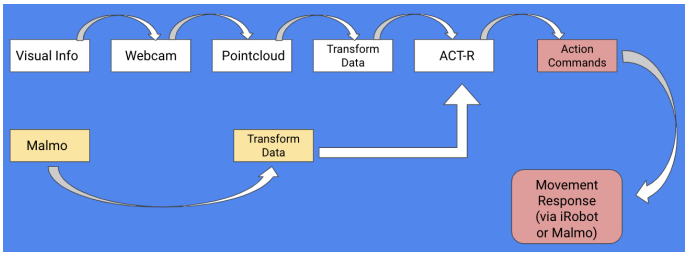

Figure 1. Flow of Information through Our Robotic System

The visual above displays the flow of information through SPIKE's system. The system is first trained using the flow portrayed in the yellow boxes, and then, when the agent is ready to explore the physical world after mastering the virtual world, the agent's flow is determined by the structure explained in the white boxes. Both processes yield the same actionable results, as shown in the pink boxes. In each scenario, SPIKE's cognition process begins by intaking data (pixels in Project Malmo or visual information/depth information from the ZED Stereolabs camera lens). In virtual training, this data can be directly sent to some cognition system or architecture, like

ACT-R, in order to be transformed into symbols. These symbols (or 'chunks' in ACT-R) are utilized for the final step, which is the creation of "Action Commands." SPIKE uses Action Commands to navigate the virtual world. In the physical world, visual data cannot be directly sent to a cognitive architecture system. Therefore, visual data is sent to Point Cloud or some ML classifier system, and transformed into usable data. This data is then, like in virtual training, transformed into symbols in a cognitive architecture and translated into the learning and implementation of Action Commands. By first training the agent in the yellow lane of flow, and then transferring to the white lane of flow, the Action Commands learned while in virtual training to stay with the agent, thus saving time, money, and energy. The agent is able to translate data and learn/implement cognitive rules more quickly and efficiently in the physical environment after learning them in the virtual environment. By first training SPIKE in the virtual environment, it gives SPIKE a foundational knowledge of its surroundings and will thus allow it to learn about changes to the environment quicker. For example, if SPIKE comes into contact with a person in the room it has already learned about, it will still know about the space it is in, and can learn what to do about this change easier than if it was completely disoriented in its environment.

10. Final Thoughts

We present Project SPIKE – an artificially intelligent robot trained to navigate physical spaces by viewing them in virtual simulations on Microsoft's platform: Project Malmo. The design of SPIKE exposes the agent to a dynamic, reliable 3D learning environment where it can explore navigation and learn cognitive techniques before entering the physical world. Using an agent like SPIKE, researchers can explore difficult new problems presented in the AI field economically and with ease. The result of this project is a robot with which individuals of more diverse backgrounds and specialties can become involved in AI innovation, and thus increase not only the efficiency but the ethical and equitable nature of the field of AI.

References

Adams et al., 2012 S. Adams, I. Arel, J. Bach, R. Coop, R. Furlan, B. Goertzel, J.S. Hall, A. Samsonovich, M. Scheutz, M. Schlesinger, et al. Mapping the landscape of human-level artificial general intelligence. AI Magazine, 33(1):25–42, 2012

Home. ACT. (n.d.). Retrieved January 23, 2022, from http://act-r.psy.cmu.edu/

Laird and Wray III, 2010 J.E. Laird and R.E. Wray III. Cognitive architecture requirements for achieving AGI. AGI, pages 79–84, 2010.

Matthew Johnson, Katja Hofmann, Tim Hutton, David Bignell, 2016 , Matthew Johnson, Katja Hofmann, Tim Hutton, David Bignell. The Malmo Platform for Artificial Intelligence Experimentation.

NVIDIA Developer Jetson TX2 module. NVIDIA Developer. (2021, April 14). Retrieved January 23, 2022, from https://developer.nvidia.com/embedded/jetson-tx2

Ritter et al., 2018 Ritter, F. E., Tehranchi, F., Oury, J. D. (2018, December 7). Act-r: A cognitive architecture for modeling cognition. Wiley Interdisciplinary Reviews. Retrieved January 23, 2022, from https://wires.onlinelibrary.wiley.com/doi/full/10.1002/wcs.1488