Sensory-Invariance Driven Action as a Method for Autonomous Grounding in a Pan-Tilt Camera

Erica Bolan[1] and Yoonsuck Choe[2]

[1]Department of Computer Science, Carleton College, Northfield MN 55057

[2]Department of Computer Science, Texas A&M University, College Station TX 77843

bolane@carleton.edu, choe@cs.tamu.edu

***Abstract*—Creation of truly autonomous agents is a primary goal of Artificial Intelligence. An algorithm using sensory invariance as a criterion for motion response has been previously proposed, and its effectiveness demonstrated. We integrate this algorithm with a pan-tilt webcam to demonstrate the algorithm's ability to interact with a natural, dynamic environment.**

I. Introduction

One of the central goals of the Aritifical Intelligence field is the development of truly autonomous agents that are able to intelligently interact with their environments, reacting to incoming stimuli appropriately. For the agent to remain autonomous, its learning must be internal, with no rules or meanings fed to it by the designer. The agent must receive a stimulus from its environment, represent the stimulus internally, and subsequently extract meaning from representation. We know that the brain is able to ground itself in this manner when it receives a visual stimulus; the neurons from which the brain extracts information about its environment do not have direct access to that environment. Yet somehow, the brain is still able to extract the structure and meaning of a visual stimulus, based only on its internal sensory state. An algorithm proposed in previous work suggests that action driven by sensory state invariance is a possible solution to how the brain accomplishes this feat, and is therefore a plausible approach for to take in the design of an artificial agent. See [1] and [2]. In our work here, we have implemented this proposed algorithm, known as Sensory-Invariance Driven Action (SIDA), on a webcam with pan and tilt capabilities. The remainder of this paper details the design of the agent, SIDA, and our experiments, and concludes with a discussion of results and directions for future work.

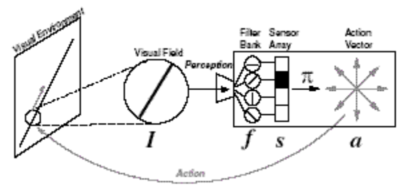


Fig. 1. A model of the agent and its environment [2]

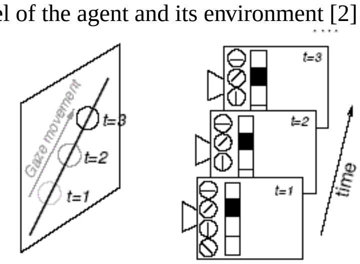


Fig. 2. The SIDA model [2]

II. The Agent

The agent is closely modeled after our knowledge of the human brain. The brain has a set of sensory filters to filter visual input, as well as a set of motor primitives. We have equipped our agent with a similar set of sensory filters and motion primitives. The sensory filters effectively remove the agent's direct access to the external stimulus, forcing it to act based solely on its internal sensory state (Figure 1). Figure 2 demonstrates SIDA in action. Note that the actions taken by the agent reflect the underlying structure of the input image, yet the ”brain” of the agent has no direct access to the stimulus in its raw form. We chose to implement this agent using a Logitech OrbitView webcam as a physical representation of the agent. The camera (Figure 3) sits on a stable base and can be directed, via modifications to its source code, to adjust its yaw and pitch (horizontal and vertical tilt) to certain degrees. The cam-

Fig. 3. The Logitech OrbitView camera used as an agent

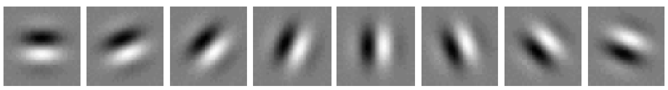


Fig. 4. Gabor filters for an agent with 8 sensory states [2]

era constantly refreshes its input, meaning that our agent was effectively interacting with a live, natural environment. We trained the camera using the process described in the next section, and then allowed it to explore its environment freely, recording its input stimuli and corresponding gaze trajectories. The sensory filters found in the human brain were mimicked within our agent using a series of oriented Gabor filters, as suggested in [2]. The agent is given n sensor filters at "conception," with n=8, 16, or 32 for the various experiments. The filters are represented visually in Figure 4; see [2] for a description of the mathematics behind these filters. Each sensory filter corresponds to a pair of motion primitives. For example, the vertical filter corresponds to movement in the direction of 90 degrees or 270 degrees.

III. Training the Agent

This section closely follows the manner outlined in previous work; see [2] for details. Training was performed off-line, using a natural, static image. The reward table generated during training was then used in the experiments outlined below.

A. Preprocessing

The first step in training was to process the input stimulus in a fasion similar to the fashion in which the brain processes visual stimuli. [2] proposes using a Difference-of-Gaussian filter for image preprocessing, and our work uses the filters described in detail there. Figure 5 shows the visual stimulus used for training, after being filtered.

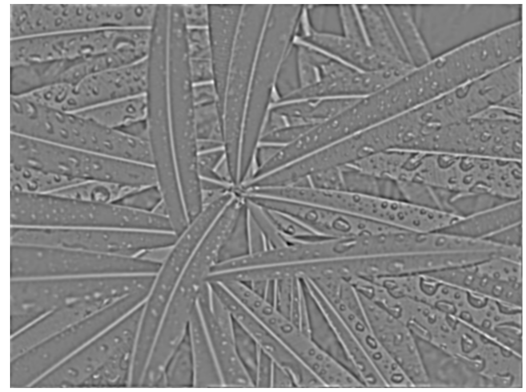


Fig. 5. A DoG-filtered image [2]

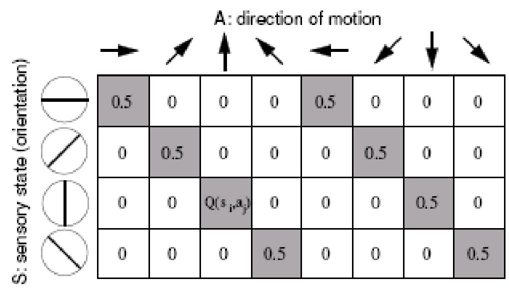


Fig. 6. An ideal Q table for an agent with n=4 sensory states and 8 corresponding motor primitives [2]

B. Learning

As proposed in [2], we used a Q-learning algorithm to train the agent. A sample Q table is shown in Figure 6, for an agent with only 4 sensory states. The table is initialized to random values, and updated using the following algorithm:

1) Given current sensory state s_t, randomly choose an action a_t.
2) If a_t equals $\arg\max_{a \in A} Q(s_t, a)$,
   a) then perform action a_t,
   b) else perform action a_t with probability proportional to $Q(s_t, a_t)$.
3) Repeat until an action has been performed.

IV. Experiments and Results

After training was completed offline, the resulting reward tables were fed into the agent for integration

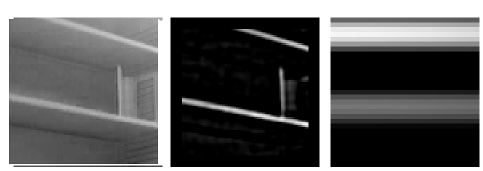
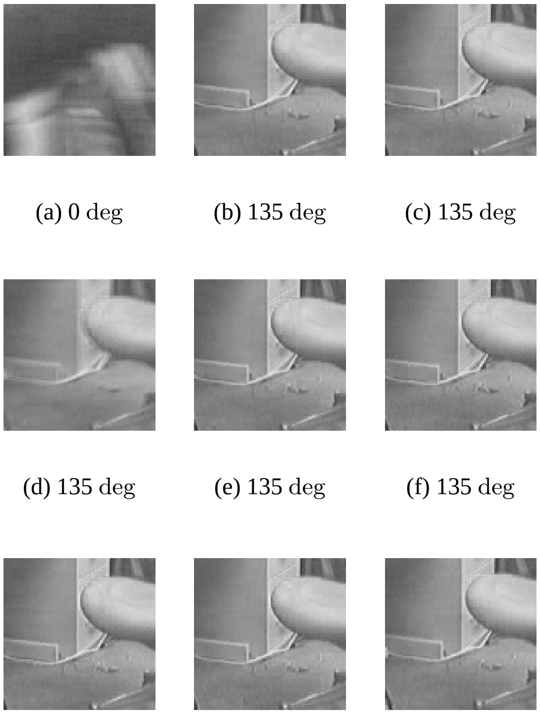


Fig. 7. An input block, its DoG filter, and its corresponding sensory state



(a) 0 deg (b) 135 deg (c) 135 deg

(d) 135 deg (e) 135 deg (f) 135 deg

(g) 135 deg (h) 135 deg (i) 90 deg

Fig. 8. Nine consecutive image inputs and their corresponding sensory states for an agent with 8 sensory filters

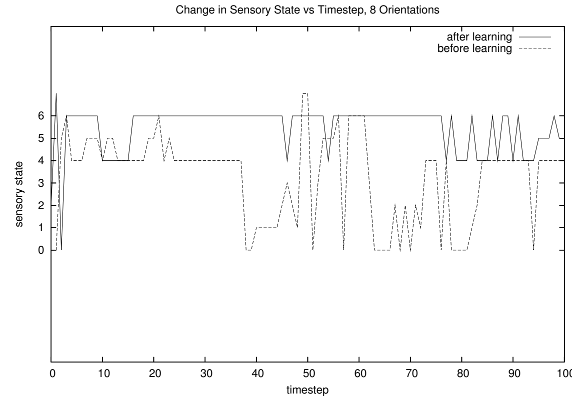


Fig. 9. Change in sensory state over 100 timesteps in an agent with 8 sensory filters

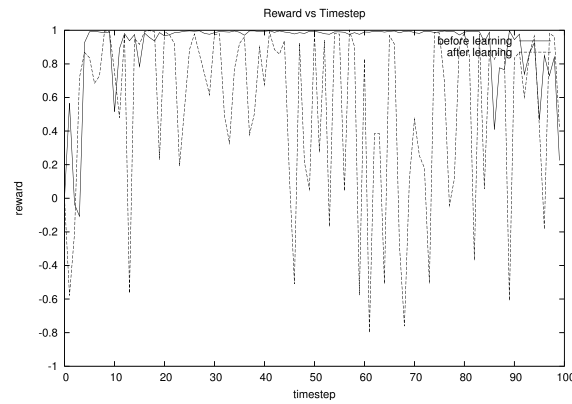
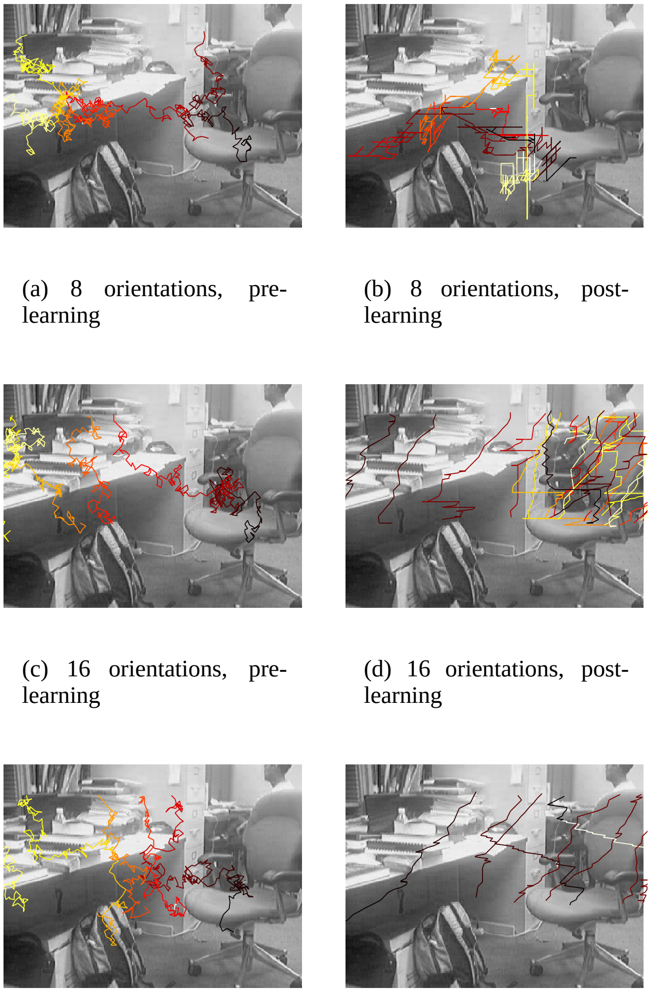


Fig. 10. Reward over time in agents before and after learning. A value of 1 represents maximum reward, -1 represents minimum reward.

with the camera. The agent/camera was pointed in an initial direction and then allowed to interact freely with its environment for 1,000 timesteps, recording its gaze trajectories. The environment was a typical office environment, containing desks, computers, file cabinets, and so forth. (The environment at the location at which the camera was initially directed can be seen in Figure 7, underlying the gaze trajectory data.) The camera recorded a 320×240 pixel image, and sampled a 93×93 chunk of that image, centered at its gaze location. The input was processed in the manner described in the previous sections. Using that filtered input, the agent determined its current sensory state and the camera, using the reward table generated previously, determined the optimal action to take. The camera's gaze was shifted according to that action, and the image was re-sampled, beginning the process again. Figure 7 shows a sample input block (part of a bookshelf, in this case), its corresponding DoG-filtered image, and the Gabor filter determined to most closely match.

Figure 8 shows the 93×93 image chunks for nine consecutive timesteps of the camera's exploration of the environment after training. The sensory state determined to correspond to the current input is also shown. Note that the agent successfully maintains its sensory state in steps (b)-(i). The sensory state here seems to correspond to the main feature of the image blocks, which is the arm of the chair. The actions taken by the camera will move along a line corresponding to 135 degrees, thus reflecting the structure of the input. Figure 9 shows the sensory state over the first 100 timesteps for the agent with 8 sensory states, before and after learning. After the learning process, sensory states are clearly maintained for longer durations, and the changes that do occur are more gradual.

Analyzing the reward from timestep to timestep shows that the agent is able to maintain a relatively high rate of reward as it re-directs its gaze. In the graphs in Figure 10, reward is determined to be the degree of sensory invariance from one action to the next. A reward of 1.0 is maximum, representing no change in the sensory state, while reward of -1.0 is minimum, representing a complete change in sen-



(a) 8 orientations, pre-learning

(b) 8 orientations, post-learning

(c) 16 orientations, pre-learning

(d) 16 orientations, post-learning

(e) 32 orientations, pre-learning

(f) 32 orientations, post-learning

Fig. 11. Gaze trajectories for agents with 8, 16, and 32 sensory states, before and after learning

sory state. The mean reward for the agent before learning was .4496, while the mean reward for the agent after learning was .8992. The reward is reasonably high before learning; this can be accounted to the fact that the camera's movements were limited to a very small distance, effectively narrowing the amount of variance possible from action to action. However, there is a significant jump in the average reward after learning, indicating that the agent has learned to maximize its reward efficiently.

The most significant results lie in the patterns of the camera's gaze trajectories as it directs itself around the environment. Figure 11 shows the camera's gaze trajectories before and after learning, mapped onto a visual representation of its environment. Before learning, the trajectory is random and undirected, and does not appear to correspond with the underlying image. After learning, though, the trajectories follow a more intelligent pattern and are more directed. It is important to note here that the background image is an estimate; as the camera shifted its center of gaze, the visual input shifted accordingly. As the number of sensory filters grew, the range of possible movements grew, allowing the camera more freedom in its explorations and allowing it to roam further from the direction in which it was intially pointed. This means that the gaze trajectories overlaid on the image do not correspond directly to the background. However, it can still be seen quite clearly that the agent is able, after learning, to direct its gaze in more consistent trajectories. The agent equipped with 8 sensory states shows particular success in modeling its input. The gaze trajectories here reflect the structure of the underlying stimulus convincingly. One oddity to note is that when the camera reached the edge of the physical limits of its horizontal or vertical gaze, it was instructed to jump back slightly, often re-locating itself in a spot very close to where it was previously. It would then take the same sequence of actions that it did upon previously visiting this area of the image, resulting in the seemingly parallel lines seen in (d). Regardless of this problem, though, it is clear from comparisons of the agent before and after learning that the camera is gleaning structural knowledge of its environment and acting appropriately, based only on its internal sensory state.

V. Conclusions

Our work here has demonstrated the effectiveness of SIDA when integrated with a physical agent, in this case a camera with pan and tilt capabilities. Our results show that the algorithm can be fully integrated with an agent in a manner that allows it to interact intelligently with a live, natural environment. Through the camera, the agent is able to maintain a high degree of sensory invariance as it explores its world, and is thus able, through its actions, to map the structure of the input–*using only its internal sensory state to guide its gaze, and thus maintaing a state of autonomous grounding*. The agent, trained using SIDA and integrated with a camera, was able to successfully interact with a live environment based only on its training on a static (and very dissimilar) image. These are the main contributions of our work.

A possible direction for future work is to allow the camera to train in real-time, updating its reward table on the fly as it interacts with a live, constantly changing environment. This could potentially allow for more realistic reward tables and more precise mappings of gaze trajectories to input.

Acknowledgments

Thank you to the Computing Research Assocation and Distributed Mentor Program for their sponsorship in making this research possible.

References

[1] Y. Choe and S. Bhamidipati. Autonomous acquisition of the meaning of sensory states through sensory-invariance driven action. In *Biologically Inspired Approaches to Advanced Information Technology*, pages 176–188. Springer, Berlin, 2004.

[2] Y. Choe and N. Smith. Motion-based autonomous grounding: Inferring external world properties from internal sensory states alone. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 936–941, 2006.