BECCA:

A functional model of the human brain for arbitrary task learning

Brandon Rohrer*,
J. Dan Morrow,
Fred Rothganger,
Patrick Xavier

Sandia National Laboratories

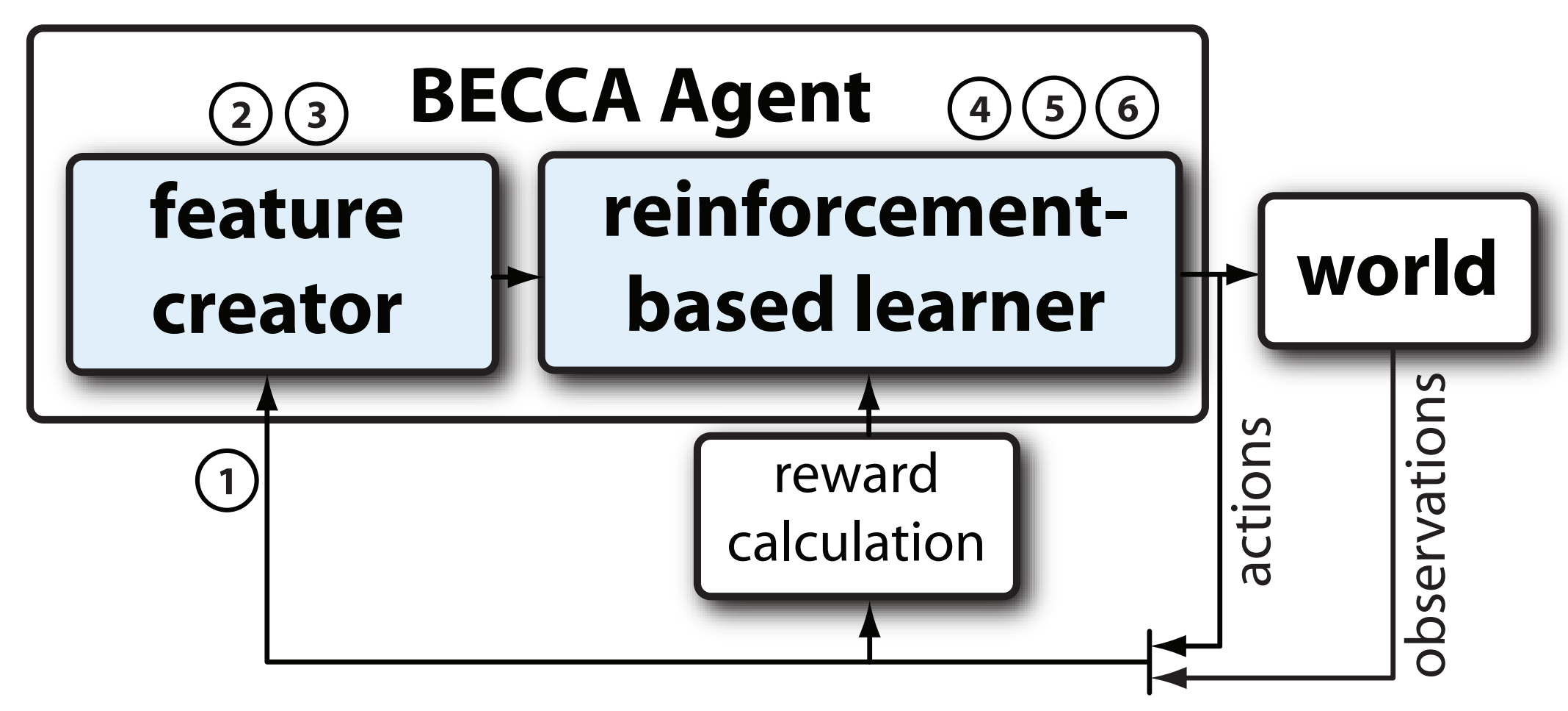
* For correspondence: brrohre@sandia.gov
For code and pulications: www.sandia.gov/~brrohre

Abstract

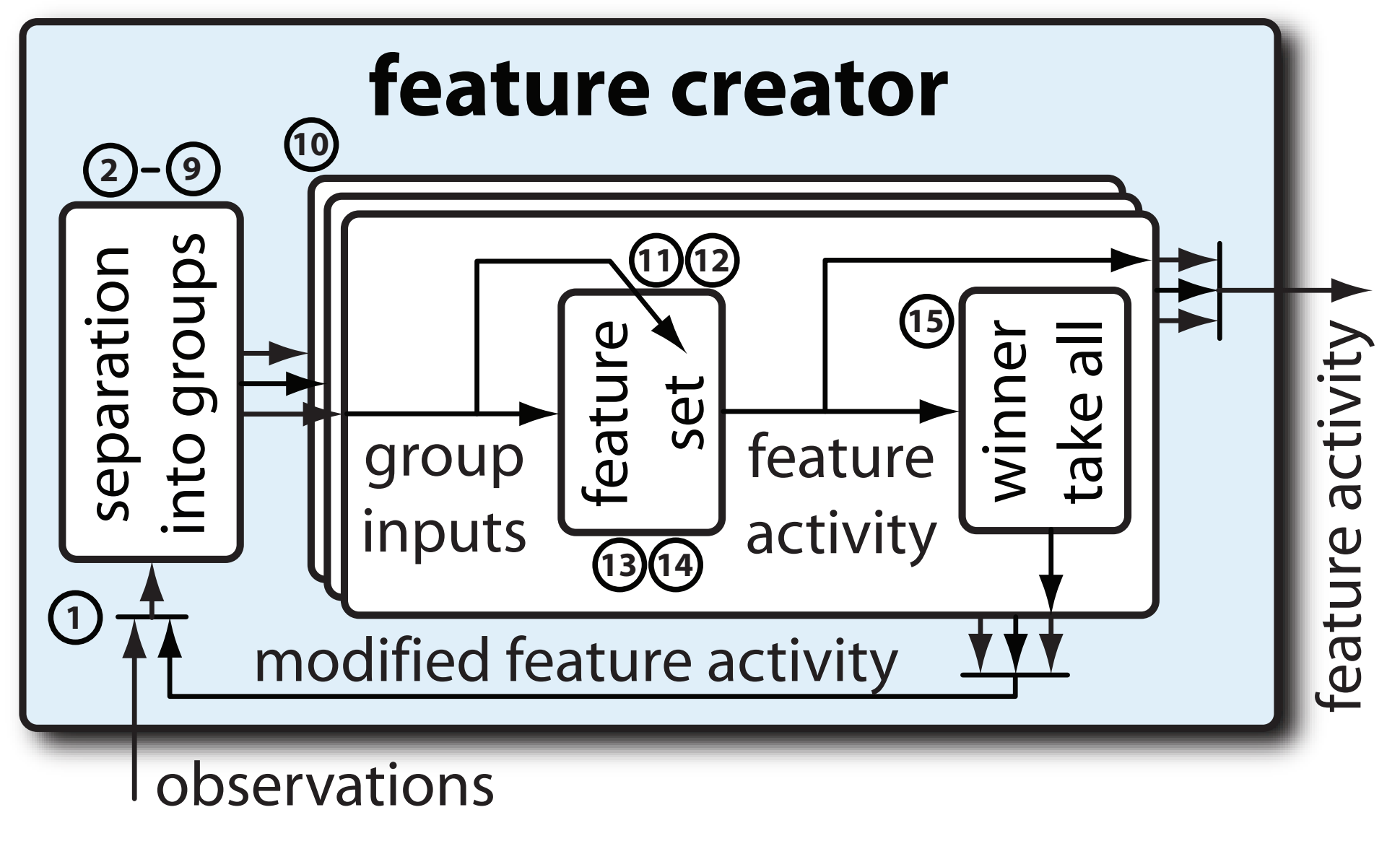
BECCA, a brain-emulating cognition and control architecture, attempts to model aspects of brain function critical to performing goal-driven tasks. It is a broad model, since many brain activities can be characterized as the performance of goal-driven tasks. Cognitive activities captured by BECCA include feature creation, feature recognition, multi-modal data fusion, attention, model creation, prediction, planning, and volitional action.

The Problem

At its highest level, BECCA is a solution to the general reinforcement learning problem. While receiving a sequence of sensory inputs, it chooses a sequence of actions trying to maximize the total reward accumulated. BECCA solves this problem without having any model of its world or even of its own embodiment. This is also a description for the problem that humans solve in learning how to pursue goals in the world, general **natural world interaction**.



At each timestep, the BECCA agent completes one iteration of the sensinglearning-planning-acting loop, consisting of six major steps: 1) Reading in observations and reward. 2) Updating its feature set. 3) Expressing observations in terms of features. 4) Predicting likely outcomes based on an internal model. 5) Selecting an action based on the expected reward of likely outcomes. 6) Updating its model.



Block diagram of the feature creator, illustrating its operation. Numbered labels refer to steps in Algorithm 1.

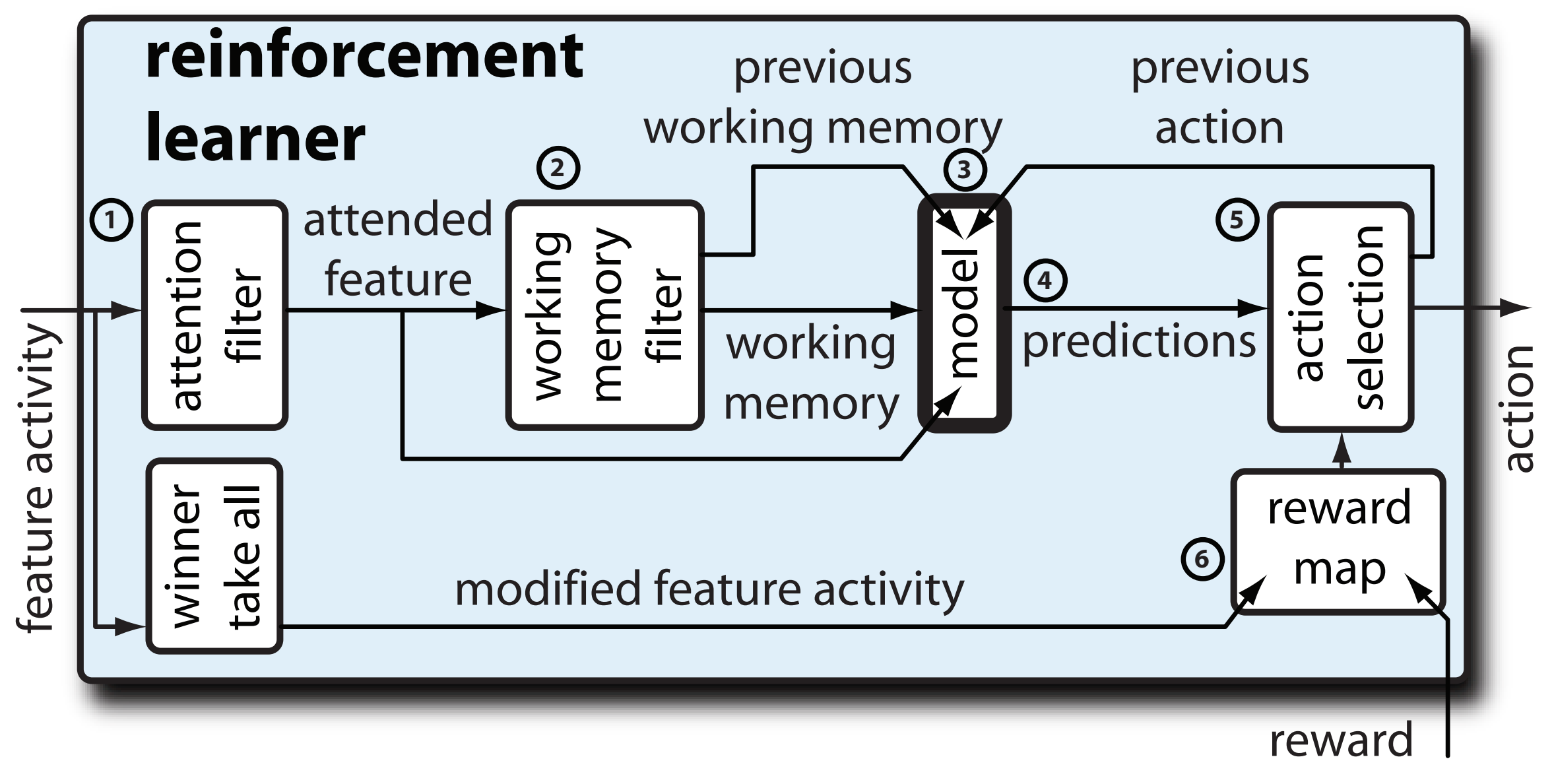
Algorithm 1 Feature Creator

```
Input: observation vector
Output: feature_activity vector

1:  form input vector by concatenating observation
        and previous modified_feature_activity
2:  update estimate of correlation between inputs
3:  if MAX(correlation) > $threshold_1$ then                creates a new group
4:      add the two input elements achieving the
            maximum correlation to the new group
5:      while NOT(stop_condition_met) do
6:          find mean_correlation between each
                remaining input and group members
7:          if MAX(mean_correlation) > $threshold_2$ then
8:              add the corresponding input element to the group
9:          else stop_condition_met
10: for each group do
11:     if MIN (COSINE DISTANCE (input, features)) > $threshold_3$ then
12:         add normalized input to set of features
13:     for each feature do
14:         feature_activity = feature · input
15:     modified_feature_activity = WINNER-TAKE-ALL(feature_activity)
```

Approach

BECCA incorporates two novel algorithms: an unsupervised feature creation algorithm and a model-based reinforcement learning algorithm. The feature creation algorithm allows its perception to grow in sophistication during the course of its operation. It builds low level inputs into higher level features, and then builds those into still higher level features until useful, abstract concepts are created. The reinforcement learning algorithm records sequences of the high level features in order to build a model of the system being controlled and its environment. It then uses that model, together with the reward history, to make predictions and choose actions likely to lead to more reward.



Block diagram of the reinforcement learner, illustrating its operation. Numbered labels refer to steps in Algorithm 2.

Algorithm 2 Reinforcement Learner

```
Input: feature_activity vector
Output: action vector

1:   attended_feature is the feature with MAX(feature_activity)
2:   decay working_memory and add attended_feature
3:   update model, consisting of cause–effect transition pairs
3.1:     cause = NORMALIZE(previous_action + previous_working_memory)
3.2:     effect = NORMALIZE(attended_feature)
3.3:     effect_matches = all transition pairs in model with matching effect
3.4:     if MAX(SIMILARITY(effect_matches, cause) < $threshold_4$)
3.5:         add new cause–effect pair to the model
3.6:     else
3.7:     increment count of nearest cause–effect pair
4:   get predictions from model
4.1:     for each model entry do
4.2:         weighted_effect = effect × LOG(count + 1) ×
                 SIMILARITY(cause, working_memory)
4.3:     expected_effect = NORMALIZE(weighted_effects)
5:   select action
5.1:     expected_reward = expected_effect × reward_map
5.2:     find cause–effect pair associated with MAX(expected_reward)
5.2:     find action associated with cause–effect pair
5.3:     on a fraction, $\alpha$, of time steps,
             generate a random exploratory action
6:   update reward_map using modified_feature_activity and reward
```

Implications

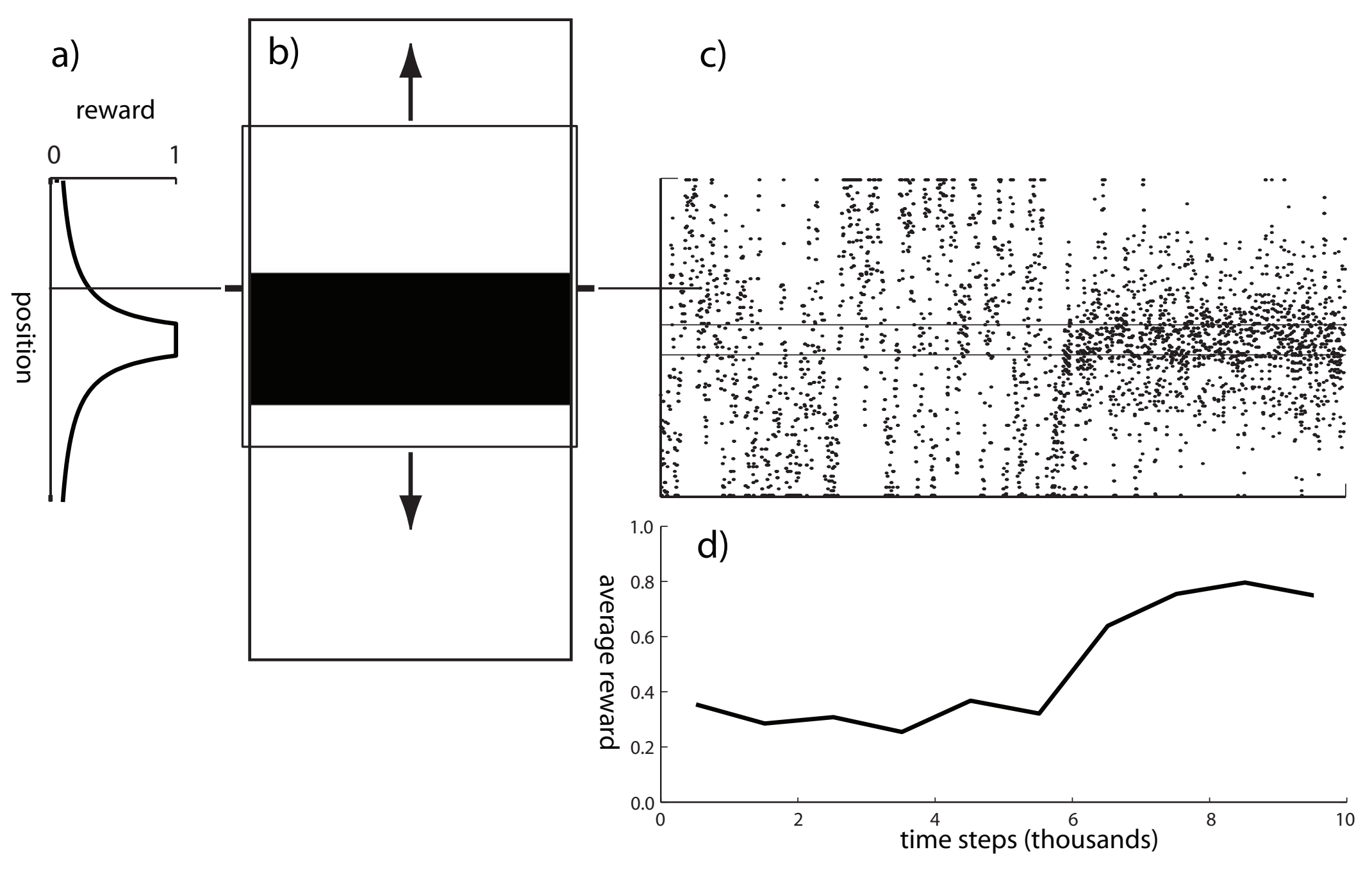
BECCA was designed to handle a broad set of natural interaction tasks, of which the visual servoing task is a simple example. By mimicking the hypothesized functions of multiple brain regions, it has demonstrated:

1. **Autonomous feature creation**
2. **Reward assignment to features**
3. **Modeling of its environment**

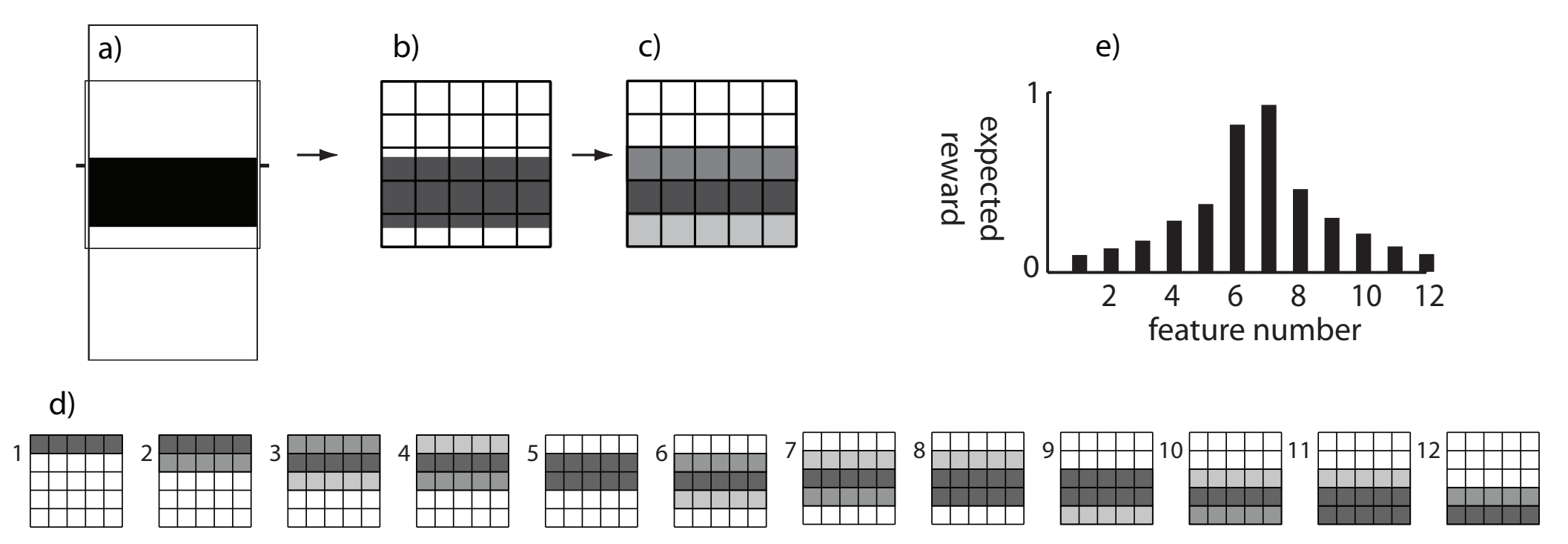
Because BECCA agents make no explicit assumptions about the world, they are potentially applicable to a wide variety of tasks and applications.

Demonstration

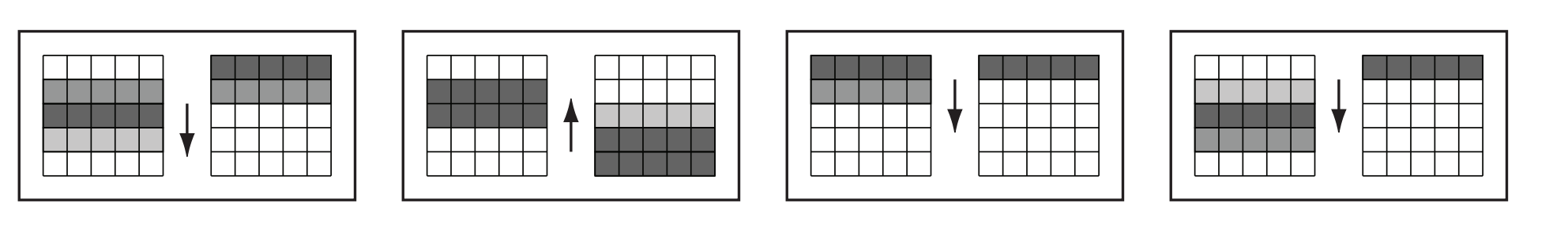
A BECCA agent panned a virtual camera along a mural. The camera produced a 5x5 pixel virtual image of its field of view. Each 50-element observation comprised the 25 pixel values, v, as well as their complement, 1-v. The actions set consisted of 15 discrete movement steps in either direction. BECCA was rewarded for directing its "gaze" at the center of the mural.



A one-dimensional visual tracking task. a) The reward signal as a function of the agent's gaze direction. b) The task environment. The white field with black bar provided a visual world. The frame representing the agent's field of view and the direction of its gaze is also shown. The agent executed panning movements to adjust its gaze direction. c) The agent's gaze direction history. Fine lines indicate the region resulting in maximum reward. After approximately 6000 time steps the gaze focused more on the high reward region. d) Average reward per time step. After approximately 6000 time steps the average reward roughly doubled.



Feature creation in the visual tracking task. a) BECCA's visual field at each time step spanned only a portion of the mural. b) Although the visual field was 200 x 200 pixels, it was pixelized into a 5 x 5 pixel array. c) Pixelization was achieved by averaging the pixel values within each superpixel. d) The visual tracking task resulted in twelve features. e) The reward associated with each feature in BECCA's experience corresponded closely to the reward expected, based on the reward-position function in panel a, above.



The most common sequences recorded within the model. Each arrow indicates the direction of gaze shift and results in image motion in the opposite direction.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.