frontiers in
PSYCHOLOGY

ORIGINAL RESEARCH ARTICLE
published: 22 November 2010
doi: 10.3389/fpsyg.2010.00173

Credit assignment in multiple goal embodied visuomotor behavior

Constantin A. Rothkopf[1]* and Dana H. Ballard[2]

[1] *Frankfurt Institute for Advanced Studies, Goethe University, Frankfurt am Main, Germany*
[2] *Department of Computer Science, University of Texas at Austin, Austin, TX, USA*

Edited by:
Anna M. Borghi, University of Bologna, Italy

Reviewed by:
Sascha Topolinski, Universität Würzburg, Germany
Stephen J. Flusberg, Stanford University, USA

****Correspondence:***
Constantin A. Rothkopf, Frankfurt Institute for Advanced Studies, Ruth-Moufang-Strasse 1, 60438 Frankfurt am Main, Germany.
e-mail: rothkopf@fias.uni-frankfurt.de

The intrinsic complexity of the brain can lead one to set aside issues related to its relationships with the body, but the field of embodied cognition emphasizes that understanding brain function at the system level requires one to address the role of the brain-body interface. It has only recently been appreciated that this interface performs huge amounts of computation that does not have to be repeated by the brain, and thus affords the brain great simplifications in its representations. In effect the brain's abstract states can refer to coded representations of the world created by the body. But even if the brain can communicate with the world through abstractions, the severe speed limitations in its neural circuitry mean that vast amounts of indexing must be performed during development so that appropriate behavioral responses can be rapidly accessed. One way this could happen would be if the brain used a decomposition whereby behavioral primitives could be quickly accessed and combined. This realization motivates our study of independent sensorimotor task solvers, which we call modules, in directing behavior. The issue we focus on herein is how an embodied agent can learn to calibrate such individual visuomotor modules while pursuing multiple goals. The biologically plausible standard for module programming is that of reinforcement given during exploration of the environment. However this formulation contains a substantial issue when sensorimotor modules are used in combination: The credit for their overall performance must be divided amongst them. We show that this problem can be solved and that diverse task combinations are beneficial in learning and not a complication, as usually assumed. Our simulations show that fast algorithms are available that allot credit correctly and are insensitive to measurement noise.

Keywords: modules, reinforcement, learning, credit assignment, reward

INTRODUCTION

Very early on it was recognized that to realize the sophisticated decisions that humans routinely make, their brains must have some kind of internal model (Tolman, 1948). One of the key figures in the modern day was Neisser (1967) who refined the idea of an internal cognitive architecture. Current systems codify experts' knowledge, e.g. (Anderson, 1983; Laird et al., 1987; Langley and Choi, 2006; Sun, 2006). The principal feature of these systems is their use fine-grained rules with variables and bind them by pattern matching. Their broad intent is to search for a sequence of rules that will solve a problem. Despite the challenging difficulties involved, expert systems have achieved notable successes, particularly in intellectual problems where the symbol bindings can be intuited, such as in algebraic problem solving (Ritter et al., 1998). However a crucial area that these systems have tackled more secondarily is that of perception and action[1].

In contrast, diverse communities in robotics and psychology have been working on cognitive architectures that take a more integrated approach to vision and action, and both have recognized that the ultimate model architecture will have a hierarchical structure (e.g., Brooks, 1986; Newell, 1990; Firby et al., 1995; Arkin, 1998; Bryson and Stein, 2001). Robotics researchers in particular have gravitated to a modular three-tiered structure that models strategic, tactical and detail levels in complex behavior (Bonasso et al., 1997). Embodied cognition integrates elements from all these advances but in addition places a special stress on the body's role in computation. It emphasizes that the brain cannot be understood in isolation as so much of its structure is dictated by the body it finds itself in and the world that the body has to survive in (Ballard et al., 1997b; Roy and Pentland, 2002; Barsalou, 2009; Glenberg, 2010). This has important implications for cognitive architectures, because the brain can be dramatically simpler than it could ever be without its encasing milieu. The reason is that the brain does not have to replicate the natural structure of the world or the special ways of interacting with it taken by the body but instead can have an internal structure that implicitly and explicitly anticipates these commitments. Research in this area has shown that using simulated figures in realistic virtual environments can make delicate manipulation problems of limited clearance more readily solvable (Badler et al., 1993, 1999) and revealed the economies of state needed to interact in dynamic environments (Terzopoulos et al., 1994;

[1]For example in Anderson's ACT-R, vision is appended as a subsystem, with the ability to search for parts of the image by coordinates or feature, its rules being based on Treisman (1980) and Trick and Pylyshyn (1994).

Terzopoulos, 1999; Sprague et al., 2007). Moreover, these compact state descriptions of sensorimotor interactions lend themselves to being modeled with reinforcement learning (RL).

Considerable empirical evidence has demonstrated that activity in human and animal brains can be related to variables in models of RL. The data comprises single cell activity in reward related visuomotor behavior in monkeys (Schultz et al., 1997b; Schultz, 2000) and BOLD activity using fMRI in humans during reward related and cognitive control tasks (Gottfried et al., 2003; Haruno and Kawato, 2006; Pessiglione et al., 2006). Although typical RL models can handle such small problems, they have the drawback that they do not scale up to large problems since the state spaces grow exponentially in the number of state variables. Furthermore, RL is mostly applied to individual tasks and not to tasks with multiple goals and task combinations. These problems have made it difficult to apply RL to realistic settings, with the result that the state spaces considered are generally small.

The scaling issue can be addressed by exploiting the structure present in a complex task through some form of factorization. While some previous work has developed techniques to learn such structure within a complex task (Guestrin et al., 2003), another approach is to start from independent tasks (Singh and Cohn, 1998; Sprague and Ballard, 2003) and consider their combinations. Imagine that you are late getting out of bed in the morning and have to quickly get ready and try to catch the bus. You have to get dressed, gather your things, run down the street toward the bus stop while avoiding other pedestrians, etc. That is, you have to pursue multiple goals at once. Our premise is that each of these goals has some intrinsic value to the overall enterprise, and that the brain has to know what these values are in order to juggle contingencies. Another important reason for knowing the value of the component modules is that this knowledge allows different combinations of multiple active modules to be used in many different tasks.

The computational difficulty arising from the modular approach is that the obtained reward needs to be attributed correctly in order for the modules to learn their respective contribution to the momentary reward. In many settings it only is reasonable to assume that a global signal of reward is available. Thus different active reinforcement learning modules have the problem of dividing the global reward up between them. Our focus is this problem. By solving this credit assignment problem correctly, individual modules can learn their respective contribution to achieving the current task combination.

Our robust solution to credit assignment succeeds by assuming that each module has access to the estimated sum of the reward estimates of other active modules. We derive formulas for estimates of reward that, assuming properties of the duration of episodes during which the concurrent goals are not changing, converge rapidly to their true values. We demonstrate that the algorithm can solve the credit assignment problem in a variant of a classical animal foraging problem in the literature (Singh and Cohn, 1998) as well as a more complex case of a human avatar learning multi-tasking in a virtual environment (Sprague et al., 2007). Thus, we show how a well-established reward-dependent learning algorithm that has been successful in modeling animal and human visuomotor and cognitive learning can be extended to learn solutions to multiple goal visuomotor behavior.

MATERIALS AND METHODS

REINFORCEMENT LEARNING BACKGROUND

A standard formalism for describing the brain's programs is that of Markov Decision Processes (MDPs). An individual MDP consists of a 4-tuple (S,A,T,R) with S being the set of possible states, A the set of possible actions, T the transition model describing the probabilities $P(s_{t+1}|s_t, a_t)$ of reaching a state s_{t+1} when being in state s_t at time t and executing action a_t, and R is a reward model that describes the expected value of the reward r_t, which is distributed according to $P(r_t|s_t, a_t)$ and is associated with the transition from state s_t to some state s_{t+1} when executing action a_t.

The goal of RL is to find a policy π that maps from the set of states S to actions A so as to maximize the expected total discounted future reward through some form of learning. The dynamics of the environment T and the reward function R are not known in advance and an explicit reward function R is learned from experience. RL algorithms effectively assign a value $V^\pi(s)$ to each state, which represents this expected total discounted reward obtainable when starting from the particular state s and following the policy π thereafter. Where γ is a scalar factor that discounts future rewards, $V^\pi(s)$ can be described by:

$$V^\pi(s) = E^\pi\left(\sum_{t=0}^{\infty}\gamma^t r_t\right) \tag{1}$$

Alternatively, the values can be parameterized by state and action pairs, denoted by $Q^\pi(s, a)$. Where Q^* denotes the value associated with the optimal policy π^*, the optimal achievable reward from a state s can be expressed as $V^*(s) = \max_a Q^*(s, a)$ and the Bellman optimality equations for the quality values can be formulated as:

$$Q^*(s,a) = \sum_r rP(r\,|\,s,a) + \gamma\sum_{s'\in S} P(s'\,|\,s,a)\max_{a'} \quad Q^*(s',a') \tag{2}$$

Temporal difference learning (Sutton and Barto, 1998), uses the error between the current estimated values of states and the observed reward to drive learning. In its related Q-learning form, the estimate of the value of a state-action pair is adjusted by this error δ_Q using a learning rate α:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha\delta_Q \tag{3}$$

Two important expressions for δ_Q are (1) the original Q-learning rule (Watkins, 1989) and (2) SARSA (Rummery and Niranjan, 1994). The first is an off-policy rule, i.e., it uses errors between current observations and estimates of the values for following an optimal policy, while actually following a potentially suboptimal policy during learning. SARSA[2] is an on-policy learning rule, i.e., the updates of the state and action values reflect the current policy derived from these value estimates. As SARSA allows one to follow a suboptimal policy in the course of learning, it is well-matched for use with modules, which cannot always depend on following their own policy recommendations. Its learning rule is given by:

$$\delta_Q = r_t + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t). \tag{4}$$

[2]SARSA is an acronym for the quintuple s_t, a_t, r_{t+1}, s_{t+1}, a_{t+1} denoting the actual trajectory followed.

Evidence that both the Q-learning and the SARSA error signals are represented in the brain of animals and humans have been provided in numerous experiments (Schultz et al., 1997b; Schultz, 2000; Morris et al., 2006).

INDIVIDUAL TASK SOLUTIONS: MODULES

The essential architectural commitment is that the required behaviors can be realized with separate MDP modules. The primary assumption is that, to a first approximation, such modules are activated in subsets whose members either do not interfere with each other (Guestrin et al., 2003; Russell and Zimdars, 2003; Sprague et al., 2007), or, if they do, then the interference can be handled in a way that approximates the result one would obtain from the complete state space that included all the active module state values[3]. We first describe the equations that govern the situation wherein the modules are completely independent, then show the modifications for embodiment wherein modules have to agree on the action selected, and finally show the notation used to describe the situation where the instantaneous reward is only known for the total subset of active modules and not for individuals.

Embodied module definitions

An independent RL module with its own actions can be defined as an MDP, i.e., the ith module is given by

$$\mathcal{M}_i = \{S_i, A_i, T_i, R_i\} \tag{5}$$

where the subscripts denote that the information is from the ith MDP. The states of the different modules are assumed all non-overlapping. In such a case, the optimal value function is readily expressible in terms of the component value functions and the states and actions are fully factored so that there is no overlap between states and additionally the following two conditions hold. Where $s = \{s^{(1)}, \ldots, s^{(M)}\}$ is the combined state of the M modules and similar notation is used for **a** and **r**,

$$P\left(s_{t+1} \mid s_t, a_t\right) = \prod_{i=1}^{M} P\left(s_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right) \tag{6}$$

$$P\left(r_{t+1} \mid s_t, a_t\right) = \prod_{i=1}^{M} P\left(r_{t+1}^{(i)} \mid s_t^{(i)}, a_t^{(i)}\right) \tag{7}$$

These two conditions can be used together with Eq. 2 in order to arrive at the result:

$$Q\left(s_t, a_t\right) = \sum_{i=1}^{N} Q\left(s_t^{(i)}, a_t^{(i)}\right) \tag{8}$$

If Eqs. 6 and 7 hold and all the rewards are known, the action maximizing Eq. 8 can be selected and is guaranteed to be optimal. In this decomposed formulation, each module can follow its own policy π^i, mapping from the local states s^i to the local actions a^i. This case is appropriate for a multi-agent setting when each module can be identified with a separate agent that may be expected to act independently.

However, our focus is the embodied cognition setting, where single agent pursues multiple goals that are divided up between multiple independent modules that a single agent can activate concurrently (Humphrys, 1996; Karlsson, 1997; Singh and Cohn, 1998; Sprague and Ballard, 2003). The consequence is that the action space is shared, so that all active modules must choose a single action. Thus the embodiment requires some form of action selection in order to mediate the competition between the possibly rivalrous actions proposed by individual modules. We use the probabilistic softmax action selection:

$$P\left(a_t^{(j)} \mid Q\left(s_t^{(1)}, a_t\right), \ldots, Q\left(s_t^{(N)}, a_t\right)\right) = \frac{e^{Q\left(s_t^{(j)}, a_t^{(j)}\right)/\tau}}{\sum_{i=1}^{M} e^{Q\left(s_t^{(i)}, a_t^{(i)}\right)}/\tau} \tag{9}$$

to choose the action, and once it has been selected, it is used for all modules. This type of action selection has been shown to model human behavior well in a variety of single goal decision making tasks (Daw et al., 2006; Rangel and Hare, 2010). The parameter τ controls the balance between exploration and exploitation during learning and usually decreases over time to reflect the shift toward less exploratory decisions over the course of learning. Note that in this formulation we propose to select a decision based on the combined value that the modules predict. This is different from Doya et al. (2002) where all modules contribute weighted by how well each module is predicting the dynamics of the world, irrespective of value or overall outcome. This is also different from Daw et al. (2005) where a single controller selects the next action alone based solely on the uncertainty of the current value estimates of a state so that each module needs to represent the same, full set of state variables.

This model has been very effective in representing human performance in the case where the multiple tasks are to walk down a sidewalk while simultaneously staying on the sidewalk, picking up litter objects and avoiding obstacles. **Figure 1**, a replication of Sprague et al. (2007), shows the results of the learning via RL of separate modules for each of these three tasks by an avatar model embedded in the same environment. The top panels in the figure show the discounted reward values as a function of the state space in front of the agent. The bottom panels show the respective policies. Note that for each of the modules the state estimate is different, as a consequence of the disposition of the agent in the environment and the relative positions of surrounding objects. **Figure 1** illustrates the action selection issue that crops up with the use of modules: actions recommended by individual modules may be different, requiring resolution by the use of Eq. 9.

Finally we can address the new constraint we are after and that is that the individual rewards due to each module are not known, but only the global reward is supplied to the agent at each time step. Using only this information, the agent needs to compute the share of the credit for each module. To describe this situation formally, we can write

$$M_i = \{S_i, A_i, T_i, G_{\mathcal{M}(t)}\} \tag{10}$$

[3]One always has to worry about whether the state definition does indeed capture all the relevant information. Formally one tackles this by appealing to additional structure of the *partially-observable* MDP that contains probabilistic machinery to represent the fact that being in any particular state is uncertain and has only an associated probability. However in the embodied cognition setting this extra machinery may not always be required, as extensive sensori-motor feedback, can render the uncertainties in the state estimate manageable with standard estimation techniques, such as Kalman filters as is done here. Nonetheless, the presented solution based on MDPs could be extended to consider belief states.

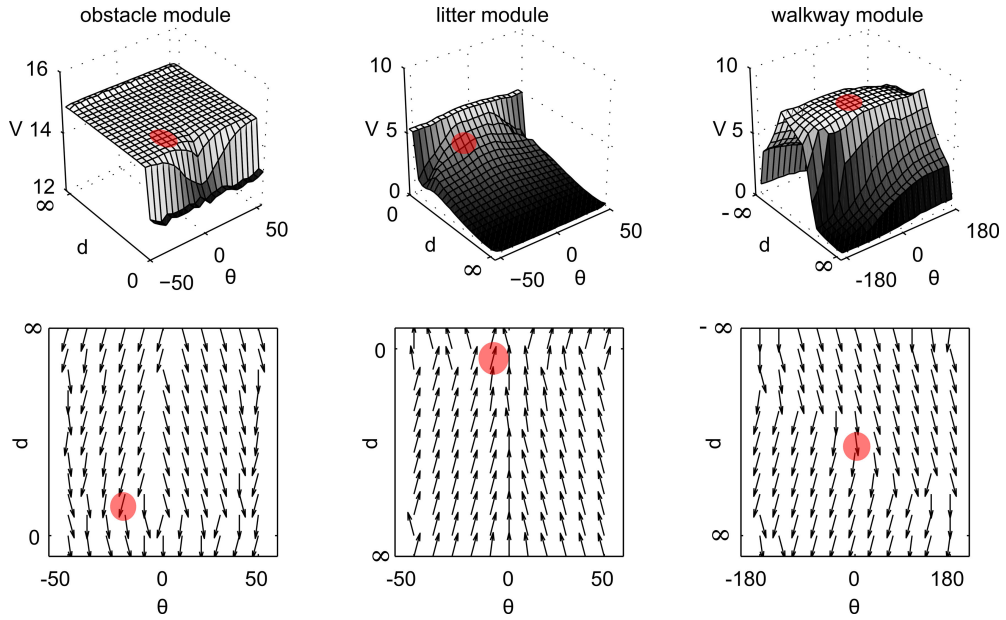

FIGURE 1 | Value functions (Top row) and their associated policies (Bottom row) for each of three modules. These functions for obstacle avoidance, litter collection, and sidewalk preference in left to right order have been learned by a virtual avatar walking along a sidewalk strewn with litter and obstacles. The red disk marks the state estimate uncertainty for each of them for a particular moment in the traverse.

where the subscript $\mathcal{M}(t)$ in $G_{\mathcal{M}(t)}$ denotes the modules that are *active* at time step t. In later formulae we abbreviate this as G_t for economy.

Evidence for modules

Direct measurements of brain activity provide a plethora of evidence that the segments in a task take the form of specialized modules. For example the Basal Ganglia circuitry shows specific neural circuits that respond to short components of a larger task (Schultz et al., 1997; Hikosaka et al., 2008). Moreover, embodied cognition studies provide much additional evidence. Studies of dual task performance provide evidence that separate task representations compete for shared resources, such as internal resources (Franco-Watkins et al., 2010) or eye gaze (Shinoda et al., 2001).

One compelling example is that of Rothkopf and Ballard (2009) that measures human gaze fixations during the navigation task that has separate trophic (picking up litter objects) and anti-trophic (avoiding obstacles) components. The overall setting is that of our virtual environment and uses identical litter and obstacle shapes that are only distinguished by object color. When picking up the object as litter, subjects' gaze is allocated to the center of the object, but when avoiding the same object, subjects' gaze fall on the objects' edges. **Figure 2** shows both of these cases. The inference is that when approaching an object, subjects use the expanding flow field to home in on it, but when avoiding an object, subjects use the different strategy of rotating about an edge of it. These two different strategies would be efficiently handled by individual task solutions, i.e., different modules, so that individual solutions can be learned and reused in combinations.

Modules and gaze arbitration

Another justification for independent modules is that they provide an elegant model for the disposition of gaze. Owing to the small visual angle of the human fovea, approximately 1°, gaze is not easily shared in servicing different tasks, and must be allocated amongst them. Arbitrating gaze requires a different approach than arbitrating control of the body. Reinforcement learning algorithms are best suited to handling actions that have direct consequences for a task. Actions such as eye fixations are difficult to put in this framework because they have only indirect consequences: they do not change the physical state of the agent or the environment; they serve only to obtain information.

A much better strategy than the straightforward RL protocol is to choose to use gaze to service the behavior that has *the most to gain* by being updated. The advantage of doing so is that uncertainty in the state information is reduced, leading to better policy choices. As time evolves, the uncertainty of the state of a module grows, introducing the possibility of low rewards. Deploying gaze to estimate that state more accurately reduces this risk, as shown in **Figure 3**.

Estimating the cost of uncertainty is equivalent to estimating the expected cost of incorrect action choices that result from uncertainty. Given that the Q-functions are known, and that Kalman filters can provide the necessary distributions over the state variables, it is straightforward to estimate this factor, loss_b, for each behavior b by sampling, using the following analysis. The loss value can be broken down into the losses associated with the uncertainty for each particular behavior b:

$$\text{loss}_b = E\left[\max_a \left(Q_b(s_b,a) + \sum_{i\in B, i\neq b} Q_i^E(s_i,a)\right)\right] - \sum_i Q_i^E(s_i,a_E). \quad (11)$$

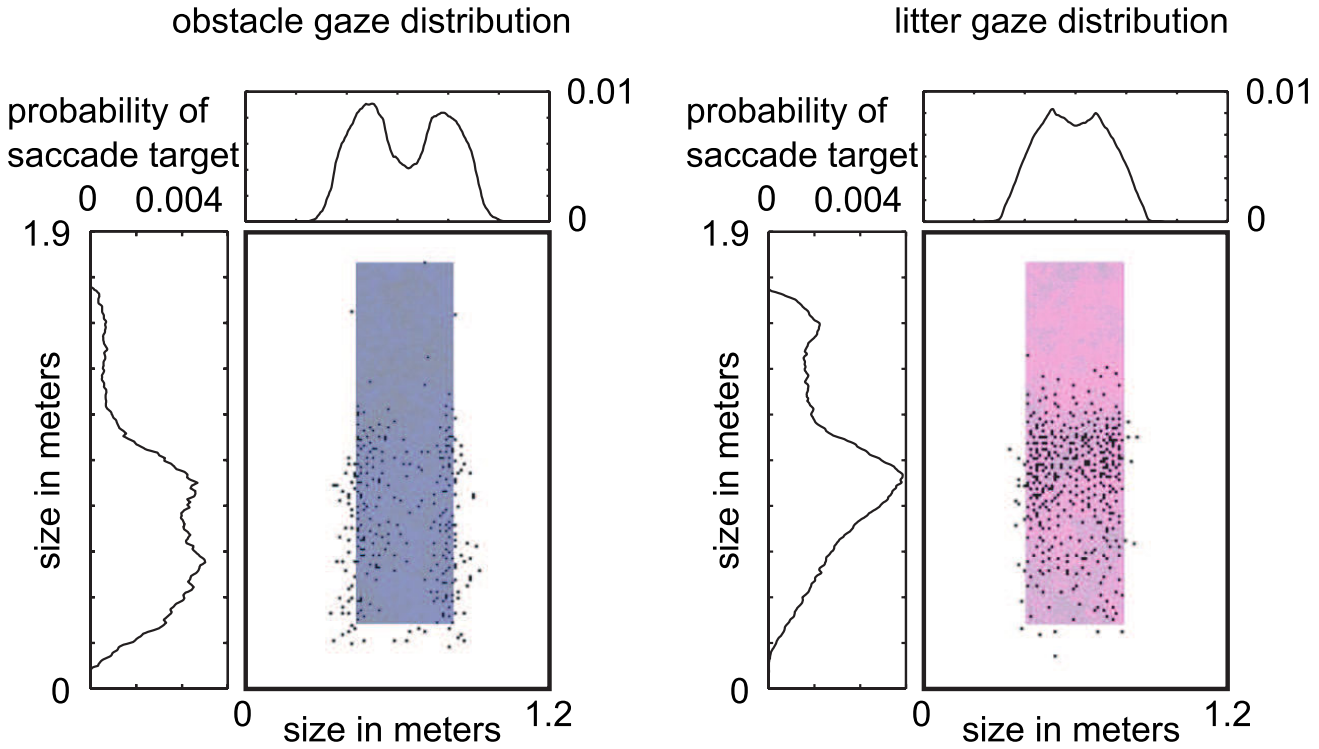

FIGURE 2 | Human gaze data for the same environment showing striking evidence for visual routines. Humans in the same environment as the avatar precisely manipulate gaze location depending on the specific task goal. The small black dots show the location of all fixation points on litter and obstacles. When avoiding obstacles (left) gaze points cluster at the edges of the object. When picking up a similar object (right) gaze points cluster on the center. From Rothkopf and Ballard (2009).

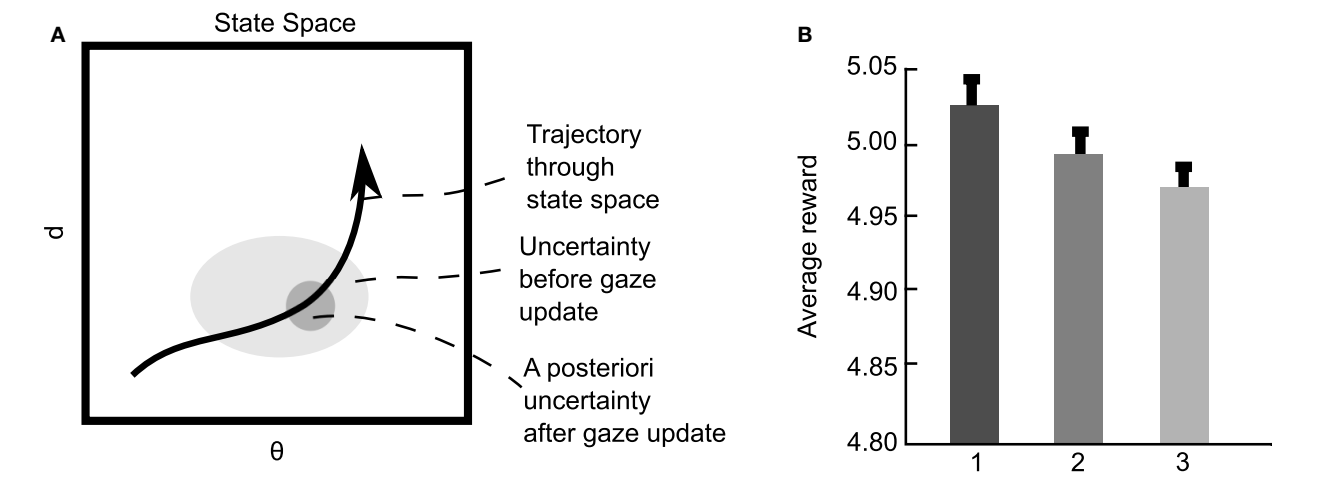

FIGURE 3 | Module-based gaze allocation. Modules compete for gaze in order to update their measurements. **(A)** A caricature of the basic method for a given module. The trajectory through the agent's state space is estimated using Kalman filter that propagates estimates in the absence of measurements and, as a consequence, build up uncertainty (large shaded area). If the behavior succeeds in obtaining a fixation, state space uncertainty is reduced (smaller shaded area). The reinforcement learning model allows the value of reducing uncertainty to be calculated. **(B)** The Sprague model out performs other models. Bars, left to right: Sprague model (1), round-robin (2), random selection (3).

Here, the expectation on the left is computed only over s_b. The value on the left is the expected return if s_b were known but the other state variables were not. The value on the right is the expected return if none of the state variables are known. The difference is interpreted as the cost of the uncertainty associated with s_b. The maximum of these values is then used to select which behavior should be given control of gaze.

The module activation protocol

Our central assumption is that an overall complex problem can be factored into a small set of MDPs, but any given factorization can only be expected to be valid for some transient period. Thus, the set of active modules is expected to change over time as the actions taken direct the agent to different parts of the composite state space. This raises two issues that we finesse: (1) How is a module activated? We assume that the sensory information provides a trigger as to when a module will be helpful. (2) How many modules can be active at a time? Extensive research on the capacity of humans to multi-task suggest that this number might be small, approximately four (Luck and Vogel, 1997). Taking both these constraints into consideration in our simulations, we use trigger features and use the value of four as a bound on the number of simultaneously active modules. Although this module activation protocol will allow the modules to learn as long as they sample their state-action spaces sufficiently often, there is still the question of how often to use it with respect to the SARSA algorithm. If it is used at every time step, the modules chosen will have little time to explore their problem spaces and adjust their Q-values. Thus for the length of module activation, we introduce the notion of an *episode* with an associated length parameter Δ (see **Figure 4**). In general this constraint should be soft as the module composition may have to be changed to deal with important environmental exigencies, but for our simulations we use a constant value. During each episode, only a subset of the total module set is active. The guiding hypothesis is that in the timecourse of behavior, a certain set of goals is pursued and therefore the corresponding modules that are needed to achieve these goals become active and those that correspond to tasks that are not pursued become inactive (Sprague et al., 2007).

In addition to the episode, we need two other assumptions:

1. The sum of the current estimates of the reward across an entire subset is accessible to each individual module in the subset at each moment by assumption;
2. The sampled subsets collectively must span the module space because the reward calculations demand this;

and the consequences of a module being activated are that:

1. It has used an associated procedure, such as a visual routine (Ullman, 1984; Ballard et al., 1997), to compute the initial state the module is in. In our examples we assume or supply a routine that does this;
2. Its Q-values are included in the sum indicated in Eq. 9 used to select an action, and
3. It influences the global reward that is handed out at every time step.

THE CREDIT ASSIGNMENT ALGORITHM

Each active module represents some portion of the composite state space and contributes through the selection of the composite action through Eq. 9, but without some additional constraint they only have access to a global performance measure, defined as the sum of the individual rewards collected by all of the $\mathcal{M}$ active modules at each time step:

$$G_t = \sum_{i \in \mathcal{M}} r_t^{(i)}. \tag{12}$$

The central problem that we tackle is how to learn the composite Q-values $Q^{(i)}(s^{(i)}, a)$ when only global rewards G_t are directly observed, but not the individual values $\{r_t^{(i)}\}$ (see **Figure 4**).

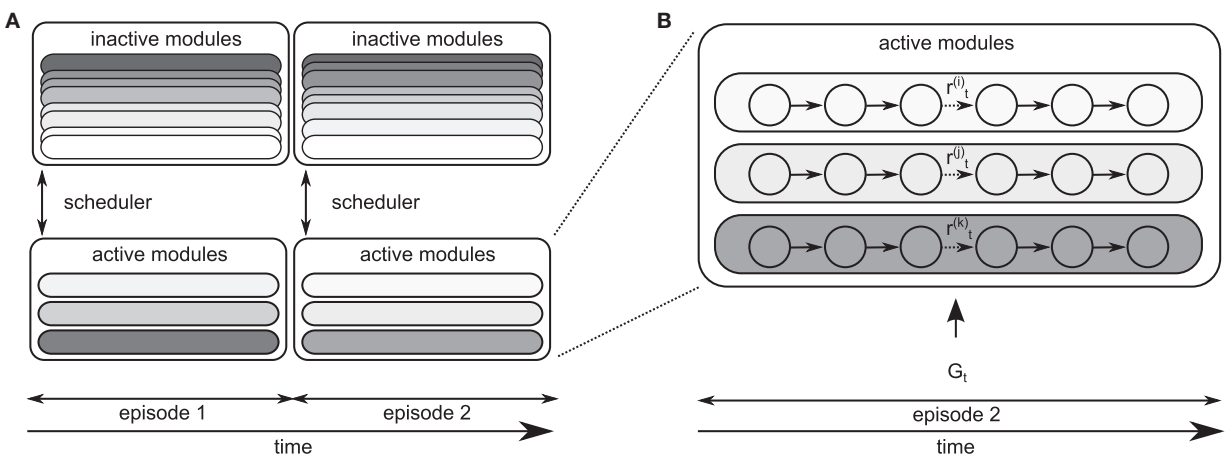

FIGURE 4 | Schematic representation of the modular credit assignment problem. (A) In any period during behavior there is only a subset of the total module set that is active. We term these periods episodes. In the timecourse of behavior, modules that are needed become active and those that are no longer needed become inactive. The vertical depicts two sequential episodes of three modules each, denoted with different shadings. The vertical arrows denote the scheduler's action in activating and deactivating modules. Our formal results only depend on each module being chosen sufficiently often and not on the details of the selection strategy. The same module may be selected in sequential episodes. **(B)** A fundamental problem for a biological agent using a modular architecture. At any given instant, shown with dotted lines, when multiple modules are active and only a global reward signal G is available, the modules each have to be able to calculate how much of the rewards is due to their activation. This is known as the credit assignment problem. Our setting simplifies the problem by assuming that individual reinforcement learning modules are independent and communicate only their estimates of their reward values.

The key additional constraint that we introduce is an assumption that the system can use the sum of current reward estimates from the modules that are co-active at any instant. This knowledge leads to the idea to use the different sets to estimate the difference between the total observed reward G_t and the sum of the current estimates of the individual rewards of the concurrently running behaviors. Credit assignment is achieved by bootstrapping these estimates over multiple task combinations, during which different subsets of behaviors are active. Dropping the temporal subscript for convenience, this reasoning can be formalized as requiring the individual behaviors to learn independent reward models $r^{(i)}(s^{(i)}, a)$. The current reward estimate for one particular behavior i, is obtained as

$$\hat{r}^{(i)} \leftarrow \hat{r}^{(i)} + \beta\delta_{r^{(i)}} \tag{13}$$

where the error on the reward estimates δ_r is calculated as the difference between the global reward and the sum of the component estimates:

$$\delta_{r^{(i)}} = G - \sum_{j\in\mathcal{M}} \hat{r}^{(i)} \tag{14}$$

so that Eq. 13 becomes:

$$\hat{r}^{(i)} \leftarrow \hat{r}^{(i)} + \beta\left(G - \sum_{j\in\mathcal{M}} \hat{r}^{(j)} \right)$$

which can be informatively rewritten as:

$$\hat{r}^{(i)} \leftarrow (1-\beta)\hat{r}^{(i)} + \beta\left(G - \sum_{j\in\mathcal{M}, j\neq i} \hat{r}^{(j)} \right) \tag{15}$$

To interpret this equation: Each module should adjust its reward estimate by a weighted sum of its own reward estimate and the estimate of its reward inferred from that of the other active modules. Together with the module activation protocol and Δ, Eq. 15 represents the core of our solution to the credit assignment problem. When one particular subset of tasks is pursued, each active behavior adjusts the current reward estimates $\hat{r}_i$ in the individual reward functions according to Eq. 15 at each time step. Over time, the set of tasks that have to be solved will change, resulting in a different set of behaviors being active, so that a new adjustment is applied to the reward functions according to Eq. 15. This bootstrapping process therefore relies on the assertion that the subsets of active behaviors visits all component behaviors.

The component Q-values for the state-action pairs of the individual behaviors are learned using the above estimates of the individual reward functions. Given the current reward estimates obtained through repeated application of Eq. 15, the SARSA algorithm is used to learn the component Q-functions:

$$Q_i\left(s_t^{(i)}, a_t^{(i)}\right) \leftarrow Q_i\left(s_t^{(i)}, a_t^{(i)}\right) + \alpha\delta_{Qi} \tag{16}$$

where δ_{Qi} now contains these estimates $\hat{r}_t^{(i)}$ and is given by:

$$\delta_{Q_i} = \hat{r}_t^{(i)} + \gamma Q_i\left(s_{t+1}^{(i)}, a_{t+1}^{(i)}\right) - Q_i\left(s_t^{(i)}, a_t^{(i)}\right) \tag{17}$$

The usage of an on-policy learning rule such as SARSA is necessary as noted in Sprague and Ballard (2003), because the arbitration process specified by Eq. 9 may select actions that are suboptimal for one or more of the modules. A feature of the SARSA algorithm is that it estimates the values of the policy that is actually used for control.

A concern one might have at this point is that since the rewards and the policies based on them are varying in separate algorithms, the net result might be that neither estimate converges. However it can be proved that this is not the case as long as $(k-1)\beta < 1$ where k is the maximum number of modules active at any one time. Furthermore convergence in the reward space is very rapid as shown by the simulations (Rothkopf and Ballard, 2010, (in press)).

Dealing with uncertainty

During the computation, the modules' MDPs are typically in different states of completion and consequently have different levels of uncertainty in their reward estimates. Unfortunately if Eq. 15 is used with a single fixed β value, this means, that on a particular task combination, all component behaviors will weight reward estimates in the same way, independent of how well component behaviors have already estimated their share. Thus a drawback of the fixed β updating scheme is that it is possible for a behavior to unlearn good reward estimates if it is combined with other behaviors whose reward estimates are far from their true values.

The problem of combining different modules' reward estimates that have different states of uncertainty can be fixed by considering the respective uncertainties in the estimates of the respective rewards separately. Thus one can have individual β_i values for each module reflect their corresponding reward estimates of uncertainty values. Assuming that the between-module fluctuations are uncorrelated, one can express the gain for each reward estimate in terms of the individual uncertainties in the respective reward estimates $(\sigma^{(i)})^2$:

$$\begin{aligned}\beta_i &= \frac{\left(\sigma^{(i)}\right)^2}{\sum_{j=1}^{N}\left(\sigma^{(j)}\right)^2} \\ &= \frac{\left(\sigma^{(i)}\right)^2}{\sum_{j\neq i}^{N}\left(\sigma^{(j)}\right)^2 + \left(\sigma^{(i)}\right)^2}\end{aligned} \tag{18}$$

where the last term in the denominator is variance in the observation noise.

Expanding the sum in the denominator in Eq. 18 suggests the second approximation, in which each individual module uses an on-line estimate for the variance $\Sigma_{j\neq i}^{N}(\sigma^{(j)})^2$ by tracking the variance in the difference between the global reward G and the sum of the reward estimates of the other modules $\Sigma_{j\neq i}\hat{r}^{(j)}$. In the following simulations each module tracked this difference using a recursive least squares estimator with exponential forgetting and maintained the uncertainty about the rewards of individual state-action pairs $(\sigma^{(i)})^2$ locally[4].

[4]The exponential forgetting factor of the recursive least squares estimator is chosen so that it reflects the timescale of the switching of the behaviors. The idea is that whenever the composition of tasks is changed the estimates will also change, because the sum of the estimates will change with the set of learners. The equation that is used is: $\lambda = 1 - (1/(\nu^{2/3}))$, where λ is the forgetting factor and ν is the time order of the sequence. For the simulations presented, the time order was established as the expected value of the number of iterations over which an individual module is switched on.

RESULTS

We demonstrate the algorithm on two separate problems. The first is a classic predator and food source problem that uses 15 different food sources and 5 predators. The second is the multi-tasking problem we described earlier of an agent in a simulated three-dimensional world walking on a sidewalk while avoiding obstacles and picking up litter (Sprague and Ballard, 2003). For all these simulations, the RL learning parameter α was 0.1. The first experiment uses both constant β values from the set {0.01, 0.1, 0.2, 0.5} as well as the variance weighted β computed according to Eq. 18. The experiment involving learning to navigate along a path while avoiding obstacles and approaching targets uses a constant β value of 0.1.

MODULE SELECTION IN THE FACE OF MULTIPLE REWARD SOURCES

This problem is described in Singh and Cohn (1998) where the authors explore the use of multiple tasks in a grid-world problem. This single-agent problem comes close to representing a problem that would have to be addressed by a biological agent since the action space is shared by the modules.

In the original formulation, an agent moves on a 5×5 grid. Possible actions move the agent in the eight compass directions. Moves at the edge of the grid-world which would result in the agent leaving the grid result in the agent staying in the current position. The grid is populated by three food items and one predator. The picking up of a food item results in a reward of one unit and the repositioning of the food item to a new and empty location. The world is also populated by a predator, which moves every other time unit toward the current position of the agent. The agent obtains a reward of 0.5 units for every time step during which it does not collide with the predator. Each learner represents the position of the respective food item or predator, i.e., there are 625 states for each of the component learners, and a total of four learners were always active in order to solve the four component tasks (see **Figure 5**).

Previously Singh and Cohn (1998) and Sprague and Ballard (2003) used this task in multi-goal learning but both studies used individual rewards that were delivered for each task as separate reward signals.

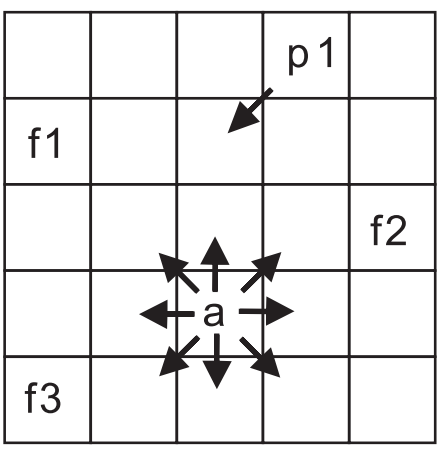

FIGURE 5 | Predator-prey grid-world example following SinghCohn,1998. An agent is located on a 5x5 grid and searches to find three different food sources f1 to f3 and tries to avoid the predator p, which moves every other time step toward the agent.

Here the problem was modified so that the reward each behavior sees is only the global sum of the individual rewards. Furthermore, at the beginning of each episode, three food sources are selected randomly according to a uniform distribution over a total of 15 different food sources. Similarly, at the beginning of each episode, one predator is selected randomly from a pool of five different predators according to a uniform distribution, so that during every episode a total of three food sources and one predator are present, as in the original problem. The Δ for each episode is 50 iterations.

Simulations were run for different values of β and with a variance weighted β for comparison. The rewards for all foods and predators were set to the values of the original Singh and Cohn problem (Singh and Cohn, 1998) in order to be able to compare the present results with the original problem formulation, which allowed a maximum average reward per episode of four units. **Figure 6** shows the average reward earned at each time step and demonstrates the improvement over learning as well as the superiority in the speed of acquiring maximal reward for the variance weighted learner.

Figure 6 furthermore demonstrates that for intermediate learning rates for β between 0.01 and 0.2, the reward estimates approach the true reward values and similarly the error in all computed value functions decreases. This is assessed by computing the RMS error between all reward estimates and the true rewards, as well as the RMS error between the true and learned value functions over time. By contrast, a learner with a learning rate of $\beta = 0.5$ does not converge on the correct reward model over the course of the simulations. Accordingly, this learner does neither approach the correct value function as shown in **Figure 6** nor approach the average reward collected by the other learners. Again, the variance weighted learner is able to learn the reward model faster than the learners using a constant learning rate β so that the error in both the reward estimates as well as in the value function decrease fastest.

LEARNING WALKWAY NAVIGATION IN A VIRTUAL 3D ENVIRONMENT

This problem uses a humanoid agent navigating in a realistic virtual reality environment that has three dimensionality. The agent uses simulated vision to compute features from the environment that define each module's state space. Also the agent's discrete state spaces must guide it successfully through the much more fine-grained environment. The walkway navigation task was first considered by Sprague et al. (2007) where a factorized solution was presented. However, that solution was obtained by delivering each of the individual learners their respective reward; that is, the agent received three separate rewards, one for the walkway following module, one for the obstacle avoidance module, and one for the litter picking up module. This problem was re-coded here but with the additional constraint of only global reward being observed by all modules in each task combination. The global reward was always the linear sum of the rewards obtained by the individual modules according to Eq. 12.

The parameterization of the statespace is shown in **Figure 7**. Each module represents the states with a two-dimensional vector containing a distance and an angle. For the picking up and the avoidance behaviors, these are the distance to the closest litter object and obstacle respectively and the signed angle between the current heading direction and the direction toward the object. The distance is scaled logarithmically similarly to the original setup (Sprague et al., 2007) and the resulting distance d_i is then discretized into 21

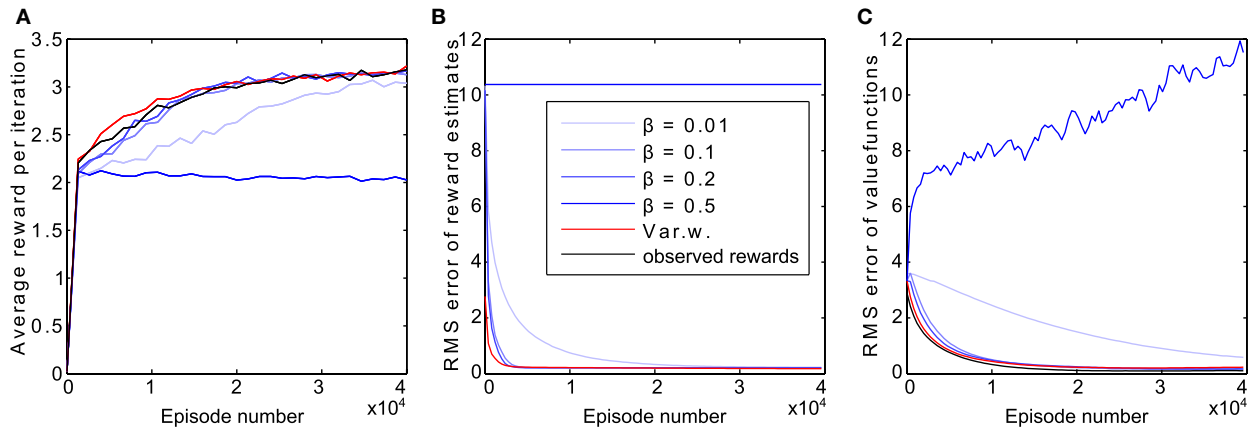

FIGURE 6 | Comparison of learning progress. (A) Effect of different learning rates and the variance weighted learning ("Var.w.") on the accumulated reward over episodes for the simulated foraging agent for the case of only knowing global reward Gt. For comparison, a case where the rewards given to each module were known is shown as the "observed rewards" trace. **(B)** Root mean squared error error between the true rewards and the reward estimates of all behaviors over episodes. The three curves correspond to the different learning rates β in Eq. 15. **(C)** Root mean squared error between the true value function and the learned value functions of all behaviors over trials.

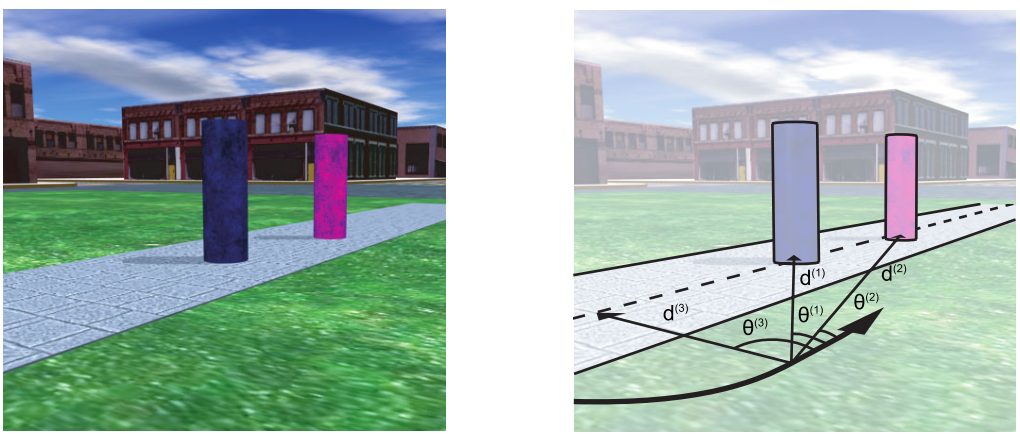

FIGURE 7 | The walkway navigation tasks. Left: typical view from the agent while navigating along the walkway. The three possible tasks are following the walkway, avoiding obstacles, which are the dark cylinders, and picking up litter, corresponding to the light cylinders. Right: Schematic representation of the statespace parameterization for the learners. Each module represents the distance to the closest object in the field of view and the angle between the current heading direction and the object center axis. The module learning the walkway behavior uses the signed distance to the midline of the walkway and the angle between the heading direction and the vector in the direction of the walkway.

possible values between 0 and infinite distance. The angles within the field of view, i.e., with a magnitude smaller than 50° are similarly discretized to 21 values. The walkway statespace is slightly different from Sprague et al. (2007) in that it represents all positions of the agent relative to the walkway for all possible walking directions. Finally, instead of 3 possible actions as in Sprague et al. (2007) the current simulations use 5 actions corresponding to steering at one of the five angles $\{-15, -7.5, 0, 7.5, 15\}$ with additive Gaussian noise of variance $\sigma^2 = 1$. To learn policies and Q-values, different subsets of modules were selected for different episodes and the correct global reward supplied for each individual subset.

The basic time unit of computation was chosen to be 300 ms, which is the average duration of a fixational eye movement. Human subjects took an average duration of 1 min an 48 s to carry out these tasks, which is approximately 325 intervals of 300 ms. Therefore, each episode consists of $\Delta = 325$ discrete time steps. At the beginning of each episode it is determined which tasks have high priority. During each episode, it is equally probable that either two or three tasks are pursued. For each episode between 35 and 40 obstacles are used, together with a similar number of litter objects.

The reward values displayed as a function of the state space locations are shown in **Figure 8A**. Starting from random values and receiving only global reward at each step, the agent's modules are able to arrive at good estimates of the true reward. The accuracy of these estimates is shown in **Figure 8B**.

The value functions and policies of these simulations are shown in **Figure 9**, at both the first iteration with random initial values and after learning, when the agent has walked the walkway for 1000 episodes.

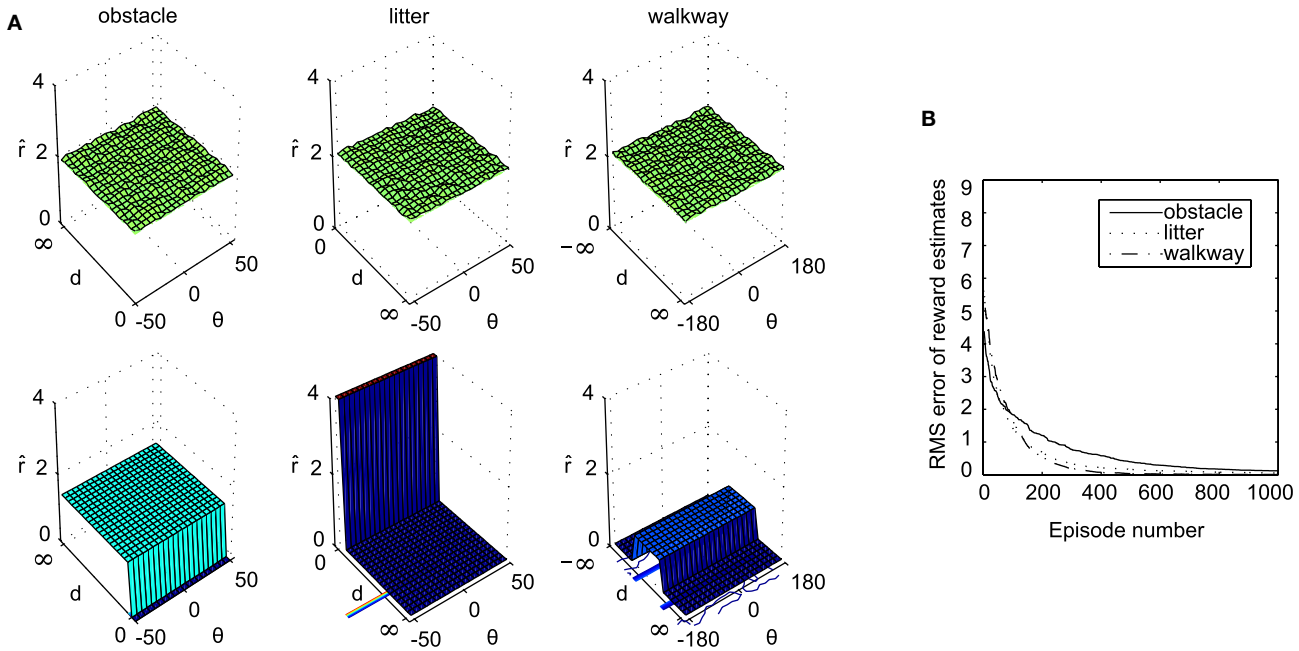

FIGURE 8 | Reward calculations for the walkway navigation task for the three component behaviors using the credit assignment algorithm. (A) Top row: Initial estimates of the reward functions. Bottom row: Final reward estimates. **(B)** Time course of learning reward functions for each of the three component behaviors. RMS error between true and calculated reward as a function of iteration number.

As can be seen from the representation of the reward estimates, the individual behaviors have learned the true rewards of their respective tasks, where not intersecting with an obstacle results in a reward of one unit, intersecting a litter object gives four units of reward, and staying on the walkway results in a reward of 0.8 units. The figures of the reward estimates also demonstrate that a function approximation scheme should be better at capturing structure in the reward space such as smooth reward landscapes, reward functions with only one state being rewarded, or separate areas with discrete rewards.

DISCUSSION

The primary contribution of this paper is to describe a way that individual task solutions can be learned by individual modules with independent state variables while pursuing multiple goals and observing only the global reward. The proposed method relies on the agent carrying out multiple task combinations over time, which enables the correct learning of individual rewards for the component tasks. Accordingly, carrying out multiple concurrent task combinations is not a complication but enables learning about the rewards associated with individual tasks. The key constraints, motivated by the need for a system that would potentially scale to a large library of behaviors, are (1) the overall system must be structured such that the system could achieve its goals by using only a subset of its behavioral repertoire at any instant, (2) the reward gained by this subset is the total of that earned by its component behaviors, and (3) the modules must be used in linearly independent combinations. The use of modules allows the rewards obtained by reinforcement to be estimated on-line. In addition this formulation lends itself to use the uncertainties in current reward estimates for combining them amongst modules, which speeds convergence of the estimating process.

The linear independence constraint (the second additional assumption in the module activation protocol) is important as without it, the Q-values and their corresponding value functions V cannot be correctly computed. To see this, note that adding a constant to the reward function does not change the policy (Ng et al., 1999) but changes the value function. When γ is near unity, a small additive constant c into the reward results in a large difference between the corresponding value function V' and the original V as shown by the following:

$$V'^{\pi}(s) = \frac{c}{1-\gamma} + V^{\pi}(s)$$

Thus, although it may be possible to learn some of the policies for component modules for one particular task combination, the value functions will be corrupted by a large bias, which will be especially problematic when new task combinations are to be solved. The reward estimates will be biased such that they have to be relearned, but will again be biased.

In our venue small numbers of behaviors that are appropriate for the current situation are selected in an on-line faction. In this situation it is essential to get the Q-values right. An algorithm that models other modules' contribution as pure noise will compute the correct policy when *all* the behaviors/agents are active but this result will not extend to active *subsets* of modules and behaviors because incorrect Q-values, when used in subsets, will cause chaotic behavior.

Considering the demonstrations, the closest work to our walkway simulation would be that of Warren and Fajen (2004) who was the first to quantitatively address the question of human dynamic trajectories in such a venue. The trajectories generated

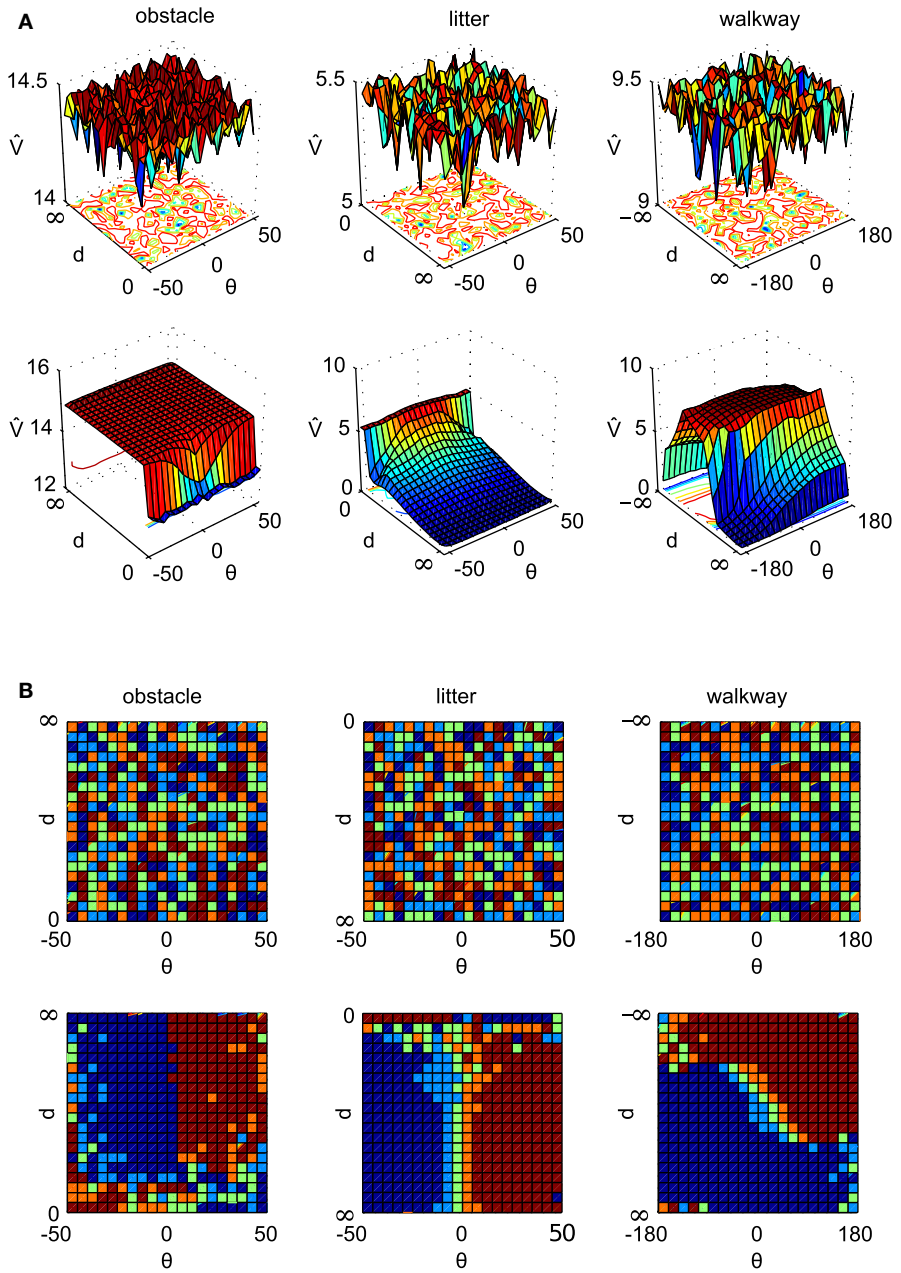

FIGURE 9 | Representations of value functions and policies in the walkway navigation task for the three component behaviors. (A) Top row: initial value function estimates $\hat{V}(s)$. Bottom row: final value estimates. **(B)** Representations of policies. Top row: initial policy estimates $\hat{\pi}(s)$. Bottom row: final policy estimates. The navigation actions are coded as follows: left turns are red, straight ahead is light green, right turns are blue.

by both models are qualitatively similar, but Warren's are curve fit to underlying differential equations and so far are not connected with concepts of reward. In principal it is easy to show differences by having objects to be picked up with different reward values. Warren's formalism has no way of expressing this so all the data for attractor and repulser objects would have to be refit.

In its formalism, the present work is related to earlier approaches that start out with compositional solutions to individual problems and then devise methods in order to combine a large number of such elemental solutions (e.g., Meuleau et al., 1998; Singh and Cohn, 1998). Both approaches are concerned with learning solutions to large MDPs by utilizing solutions or partial solutions to smaller component MDPs. In Meuleau et al. (1998) the solutions to such components are heuristically combined to find an approximate solution to the composite MDP by exploiting assumptions on the structure of the joint action space. A way of learning a composite MDP from individual component MDPs by merging has been described in Singh and Cohn (1998). However, the composite problem is solved in a more ad hoc way using bounds on the state values derived from the state values of the individual component MDPs.

Attempts to overcome the scaling problem in more elegant ways than *ab initio* factoring try to exploit inherent structure in the problem (Dayan and Hinton, 1992; Parr and Russell, 1997; Sutton et al., 1999; Barto and Mahadevan, 2003; Vigorito and Barto, 2010). Factoring can use graphical models that express conditional independencies can reduce the size of the variables necessary for a full description of the problem at hand (Boutilier and Goldszmidt, 2000; Guestrin et al., 2003). The approach by Sallans and Hinton (2004) can also be conceptualized as exploiting the statistical structures of the state and action spaces. Doya et al. (2002) and Samejima et al. (2003) use a number of actor-critic modules and learn a linear combination of the controllers for the local approximation of the policy. All these methods constitute advances and our method is extensible to them to the extent that they can be encapsulated into modules that are made explicit and the issues related to module activation are addressed.

The credit assignment problem is an important problem in embodied cognition as a behaving animal in a complex environment has to solve this problem. While common laboratory experiments consider mostly single tasks with reward being unambiguously associated with the single task component, in the natural environment multiple concurrent goals have to be solved and the contributions of the individual actions to the total observed reward have to be learned. In this setting, the ability to assign credit correctly may confer an additional advantage. When the brain encodes new behaviors, it needs to know their values. Since the brain uses its own internally generated secondary reward signals such as dopamine to estimate these values, there is the delicate issue of how to keep the overall system in calibration. The credit assignment algorithm suggests a partial solution: by using global reward and concurrent subsets of active modular behaviors, the rewards can at least be held to a global consistency.

ACKNOWLEDGMENTS

The research reported herein was supported by NIH Grants RR009283, EY019174 and MH060624. Constantin A. Rothkopf was additionally supported by EC MEXT-project PLICON and by the German Federal Ministry of Education and Research within the Bernstein Focus: Neurotechnology research grant.

REFERENCES

Anderson, J. (1983). *The Architecture of Cognition*. Cambridge, MA: Harvard University Press.

Arkin, R. (1998). *Behavior Based Robotics*. Cambridge, MA: MIT Press.

Badler, N., Palmer, M., and Bindiganavale, R. (1999). Animation control for real-time virtual humans. *Commun. ACM* 42, 64–73.

Badler, N. I., Phillips, C. B., and Webber, B. L. (1993). *Simulating Humans: Computer Graphics Animation and Control*. New York, NY: Oxford University Press.

Ballard, D. H., Hayhoe, M. M., Pook, P. K., and Rao, R. P. N. (1997). Deictic codes for the embodiment of cognition. *Behav. Brain Sci.* 20, 723–767.

Barsalou, L. (2009). Simulation, situated conceptualization, and prediction. *Phil. Trans. R. Soc. B* 364, 1281–1289.

Barto, A. G., and Mahadevan, S. (2003). Recent advances in hierarchical reinforcement learning. *Discrete Event Dyn. Syst.* 13, 41–77.

Bonasso, R. P., Firby, J. R., Gat, E., Kortenkamp, D., Miller, D. D. P., and Slack, M. G. (1997). Experiences with an architecture for intelligent, reactive agents. *J. Exp. Theor. Artif. Intell.* 9, 237–256.

Brooks, R. (1986). A robust layered control system for a mobile robot. *IEEE J. Robot. Autom.* 2, 14–23.

Bryson, J. J., and Stein, L. A. (2001). "Modularity and design in reactive intelligence," in *International Joint Conference on Artificial Intelligence*, Seattle, WA.

Daw, N. D., Niv, Y., and Dayan, P. (2005). Uncertainty-based competition between prefrontal and dorsolateral striatal systems for behavioral control. *Nat. Neurosci.* 8, 1704–1711.

Daw, N. D., O'Doherty, J. P., Dayan, P., Seymour, B., and Dolan, R. J. (2006). Cortical substrates for exploratory decisions in humans. *Nature* 441, 876–879.

Dayan, P., and Hinton, G. E. (1992). "Feudal reinforcement learning," in *Advances in Neural Information Processing Systems,* (San Francisco, CA: Morgan Kaufmann Publishers Inc.) 5, 271–278.

Dearden, R. Boutilier, C., and Goldszmidt, M. (2000). Stochastic dynamic programming with factored representations. *Artif. Intell.* 121, 49–107.

Doya, K., Samejima, K., Katagiri K., and Kawato, M. (2002). Multiple model-based reinforcement learning. *Neural Comput.* 14, 1347–1369.

Firby, R. J., Kahn, R. E., Prokopowicz, P. N., and Swain, M. J. (1995). "An architecture for vision and action," in *International Joint Conference on Artificial Intelligence* (Montreal, Canada; Morgan Kaufmann Publishers Inc.), 72–79.

Franco-Watkins, A. M., Rickard, T. C., and Pashler, H. (2010). Taxing executive processes does not necessarily increase impulsive decision making. *Exp. Psychol.* 57, 193–201.

Glenberg, A. M. (2010). Embodiment as a unifying perspective for psychology. *Wiley Interdiscip. Rev. Cogn. Sci.* 1, 586–596.

Gottfried, J. A., O'Doherty, J., and Dolan, R. J. (2003). Encoding predictive reward value in human amygdala and orbitofrontal cortex. *Science* 301, 1104–1107.

Guestrin, C. E., Koller, D., Parr, R., and Venkataraman, S. (2003). Efficient solution algorithms for factored MDPs. *J. Artif. Intell. Res.* 19, 399–468.

Haruno, M., and Kawato, M. (2006). Different neural correlates of reward expectation and reward expectation error in the putamen and caudate nucleus during stimulus-action-reward association learning. *J. Neurophysiol.* 95, 948–959.

Hikosaka, O., Bromberg-Martin, E., Hong, S., and Matsumoto, M. (2008). New insights on the subcortical representation of reward. *Curr. Opin. Neurobiol.* 18, 203–208.

Humphrys, M. (1996). "Action selection methods using reinforcement learning," in *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior*, eds P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S. W. Wilson (Cambridge, MA: MIT Press, Bradford Books), 135–144.

Karlsson, J. (1997). *Learning to Solve Multiple Goals*. PhD thesis, University of Rochester, Rochester.

Laird, J. E., Newell, A., and Rosenblum, P. S. (1987). Soar: an architecture for general intelligence. *Artif. Intell.* 33, 1–64.

Langley, P., and Choi, D. (2006). Learning recursive control programs from problem solving. *J. Mach. Learn. Res.* 7, 493–518.

Luck, S. J., and Vogel, E. K. (1997). The capacity of visual working memory for features and conjunctions. *Nature* 390, 279–281.

Meuleau, N., Hauskrecht, M., Kim, K.-E., Peshkin, L., Kaelbling, L., Dean, T., and Boutilier, C. (1998). "Solving very large weakly coupled Markov decision processes," in *AAAI/IAAI*, 165–172.

Morris, G., Nevet, A., Arkadir, D., Vaadia, E., and Bergman, H. (2006). Midbrain dopamine neurons encode decisions for future action. *Nat. Neurosci.* 9, 1057–1063.

Neisser, U. (1967). *Cognitive Psychology*. New York: Appleton-Century-Crofts.

Newell, A. (1990). *Unified Theories of Cognition*. Cambridge, MA: Harvard University Press.

Ng, A. Y., Harada, D., and Russell, S. (1999). "Policy invariance under reward transformations: theory and application to reward shaping," in *Proceedings of the Sixteenth International Conference on Machine Learning*, Bled, Slovenia.

Parr, R., and Russell, S. (1997). "Reinforcement learning with hierarchies of machines," in *Advances in Neural Information Processing Systems*, M. I. Jordan, M. J. Kearns, and S. A. Solla. (Cambridge, MA: MIT Press), 1043–1049.

Pessiglione, M., Seymour, B., Flandin, G., Dolan, R. J., and Frith, C. D. (2006). Dopamine-dependent prediction errors underpin reward-seeking behaviour in humans. *Nature* 442, 1042–1045.

Rangel, A., and Hare, T. (2010). Neural computations associated with goal-directed choice. *Curr. Opin. Neurobiol.* 20, 262–270.

Ritter, S., Anderson, J. R., Cytrynowicz, M., and Medvedeva, O. (1998). Authoring content in the pat algebra tutor. *J. Interact. Media Educ.* 98, 1–30.

Rothkopf, C. A., and Ballard, D. H. (2009). Image statistics at the point of gaze during human navigation. *Vis. Neurosci.* 26, 81–92.

Rothkopf, C. A., and Ballard, D. H. (2010). "Learning and coordinating reper-

toires of behaviors: credit assignment and module activation," in *Intrinsically Motivated Cumulative Learning in Natural and Artificial Systems,* Eds G. Baldassarre and M. Mirolli (in press).

Roy, D. K., and Pentland, A. P. (2002). Learning words from sights and sounds: a computational model. *Cogn. Sci.* 26, 113–146.

Rummery, G. A., and Niranjan, M. (1994). "On-line Q-learning using connectionist systems," Technical Report CUED/F-INFENG/TR 166, Cambridge University Engineering Department, Cambridge.

Russell, S., and Zimdars, A. (2003). "Q-decomposition for reinforcement learning agents," in *Proceedings of the International Conference on Machine Learning*, Washington, DC.

Sallans, B., and Hinton, G. E. (2004). Reinforcement learning with factored states and actions. *J. Mach. Learn. Res.* 5, 1063–1088.

Samejima, K., Doya, K., and Kawato, M. (2003). Inter-module credit assignment in modular reinforcement learning. *Neural Netw.* 16, 985–994.

Schultz, W. (2000). Multiple reward signals in the brain. *Nat. Rev. Neurosci.* 1, 199–207.

Schultz, W., Dayan, P., and Montague, P. R. (1997). A neural substrate of prediction and reward. *Science* 275, 1593–1599.

Shinoda, H., Hayhoe, M. M., and Shrivastava, A. (2001). What controls attention in natural environments? *Vis. Res.* 41, 3535–3546.

Singh, S., and Cohn, D. (1998). How to dynamically merge Markov decision processes. *Neural Inf. Process. Syst.* 10, 1057–1063.

Sprague, N., and Ballard, D. (2003). "Multiple-goal reinforcement learning with modular sarsa(0)," in *International Joint Conference on Artificial Intelligence*, Acapulco.

Sprague, N., Ballard, D., and Robinson, A. (2007). Modeling embodied visual behaviors. *ACM Trans. Appl. Percept.* 4, 1–23.

Sun, R. (2006). *Cognition and Multi-Agent Interaction*, chapter 4. Cambridge: Cambridge University Press, 79–99.

Sutton, R. S., and Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. Cambridge, MA: MIT Press.

Sutton, R. S., Precup, D., and Singh, S. P. (1999). Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artif. Intell.* 112, 181–211.

Terzopoulos, D. (1999). Artificial life for computer graphics. *Commun. ACM* 42, 32–42.

Terzopoulos, D., Tu, X., and Grzeszczuk, R. (1994). Artificial fishes: autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artif. Life*, 1, 327–351.

Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychol. Rev.* 55, 189–208.

Treisman, A. M. (1980). A feature-integration theory of attention. *Cogn. Psychol.* 12, 97–136.

Trick, L. M., and Pylyshyn, Z. W. (1994). Why are small and large numbers enumerated differently? A limited-capacity preattentive stage in vision. *Psychol. Rev.* 101, 80–102.

Ullman, S. (1984). Visual routines. *Cognition* 18, 97–157.

Vigorito, C. M., and Barto, A. G. (2010). Intrinsically motivated hierarchical skill learning in structured environments. *IEEE Trans. Auton. Ment. Dev.* 2, 132–143.

Warren, W. H., and Fajen, B. R. (2004). Behavioral dynamics of human locomotion. *Ecol. Psychol.* 16, 61–66.

Watkins, C. J. C. H. (1989). *Learning from delayed rewards*. PhD thesis, University of Cambridge, Cambridge.

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 23 July 2010; paper pending published: 23 August 2010; accepted: 28 September 2010; published online: 22 November 2010.

Citation: Rothkopf CA and Ballard DH (2010) Credit assignment in multiple goal embodied visuomotor behavior. Front. Psychology ***1****:173. doi: 10.3389/fpsyg.2010.00173*

This article was submitted to Frontiers in Cognition, a specialty of Frontiers in Psychology.