An Integrated Framework for Robust Human-Robot Interaction

Dr. Mohan Sridharan
Stochastic Estimation and Autonomous Robotics Laboratory
Department of Computer Science
Texas Tech University, Lubbock, TX 79409
mohan.sridharan@ttu.edu

ABSTRACT

Developments in sensor technology and sensory input processing algorithms have enabled the use of mobile robots in real-world domains. As they are increasingly deployed to interact with humans in our homes and offices, robots need the ability to operate autonomously based on sensory cues and high-level feedback from non-expert human participants. Towards this objective, this chapter describes an integrated framework that jointly addresses the learning, adaptation and interaction challenges associated with robust human-robot interaction in real-world application domains. The novel probabilistic framework consists of: (a) a bootstrap learning algorithm that enables a robot to learn layered graphical models of environmental objects and adapt to unforeseen dynamic changes; (b) a hierarchical planning algorithm based on partially observable Markov decision processes (POMDPs) that enables the robot to reliably and efficiently tailor learning, sensing and processing to the task at hand; and (c) an augmented reinforcement learning algorithm that enables the robot to acquire limited high-level feedback from non-expert human participants, and merge human feedback with the information extracted from sensory cues. Instances of these algorithms are implemented and fully evaluated on mobile robots and in simulated domains using vision as the primary source of information in conjunction with range data and simplistic verbal inputs. Furthermore, a strategy is outlined to integrate these components to achieve robust human-robot interaction in real-world application domains.

Key Terms: Bootstrap learning, Hierarchical POMDP, Augmented reinforcement learning, Autonomous robots, Human-robot interaction, Visual processing, Wheeled robots.

1. INTRODUCTION

Mobile robots are increasingly being used in real-world application domains such as surveillance, navigation and healthcare due to the availability of high-fidelity sensors and the development of state of the art algorithms to process sensory inputs. As we move towards deploying robots in our homes and offices, i.e., domains with a significant amount of uncertainty, there is a need for enabling robots to learn from sensory cues and limited feedback from non-expert human participants. Human-robot interaction (HRI) poses many challenges such as autonomous operation, safety, engagement, robot design and interaction protocol design (Tapus, Mataric, & Scassellati, 2007). *The focus of this chapter is on robust autonomy using sensory cues and high-level feedback from non-expert human participants.* Many algorithms have been developed for autonomous operation based on sensory inputs, and for learning from manual training and domain knowledge. Real-world domains characterized by partial observability, non-deterministic action outcomes and unforeseen dynamic changes make it difficult for a robot to operate without any human feedback. At the same time, human participants may not have the expertise and time to provide elaborate and accurate feedback in complex domains (Fong, Nourbakhsh, & Dautenhahn, 2003; Thrun, 2004). Recent research has hence focused on enabling a robot to acquire human feedback when needed and merge human inputs with the information extracted from sensory cues. However, these algorithms require elaborate domain knowledge or fail to model the unreliability of human inputs, limiting their use to simplistic simulated domains or specific real-world applications (Knox & Stone, 2010; Rosenthal, Veloso, & Dey, 2011).

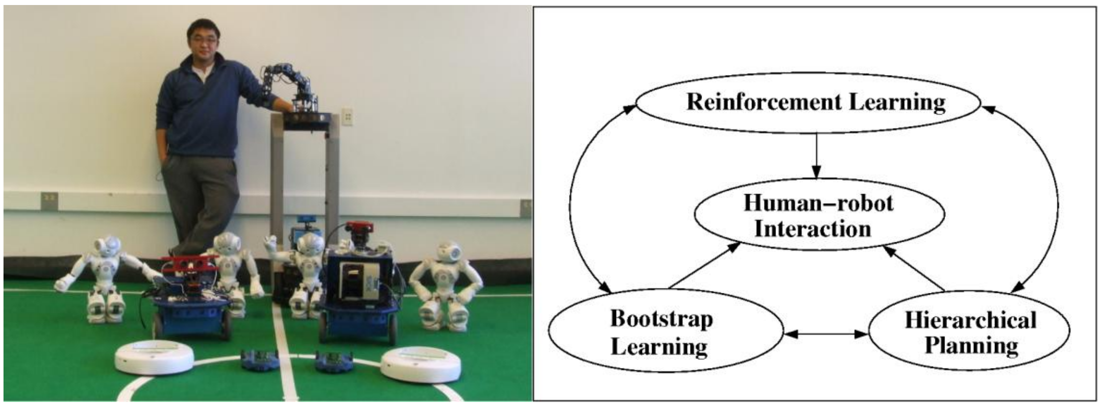

Figure 1: (Left) Examples of robot platforms relevant to the research described in this chapter; (Right) Integrated framework that uses the dependencies between learning, adaptation and interaction to achieve synergetic autonomy in real-world HRI.

As an illustrative example, consider the robots in Figure 1(left) deployed to interact with humans in offices and homes. Such real-world domains are characterized by unforeseen dynamic changes, e.g., existing objects move, novel objects are introduced and the environmental factors change unpredictably. Assume that sensory cues consist primarily of vision (monocular and stereo) in conjunction with range data and simplistic verbal inputs. Also assume that the robots do not manipulate domain objects and do not have physical contact with humans. Each robot is equipped with core algorithms to process sensory cues with varying levels of reliability and computational complexity. Non-expert human participants provide limited high-level feedback in the form of simplistic verbal inputs that reinforce the robot's actions or resolve ambiguities identified by the robot. Although it is not feasible to process all inputs or model the entire domain and still respond to dynamic changes, each robot has to exploit relevant sensory cues to operate reliably. Given such a scenario, this chapter focuses on the following key questions:

- *How to best enable a robot to adapt learning, sensing and processing to different scenarios and participants?*
- *How to best enable a robot to seek limited high-level feedback from non-expert human participants, and robustly merge human inputs with the information extracted from sensory inputs?*

While sophisticated algorithms have been developed for the learning, adaptation and interaction challenges in isolation, the integration of these subfields to enable robust HRI remains an open problem, even as it presents new opportunities to address the existing challenges in the subfields (AAAI Symposium, 2012). This chapter describes a novel probabilistic framework that seeks to answer the questions listed above by jointly addressing the associated learning, adaptation and interaction challenges. The framework is composed of the following components:

- **Bootstrap Learning:** robots use sensory cues to autonomously and incrementally learn probabilistic graphical models of environmental objects. These learned models enable robots to detect and adapt to unforeseen changes.

- **Hierarchical Planning:** a novel hierarchical decomposition of partially observable Markov decision processes enables robots to automatically adapt learning, sensing and information processing to each of a wide range of tasks.
- **Reinforcement Learning:** robots acquire high-level feedback from non-expert humans (based on need and availability) and merge the information extracted from human feedback with the information extracted from sensory inputs.

As shown in Figure 1(right), these components inform and guide each other, e.g., planning and limited human feedback can constrain learning to objects and events relevant to the task at hand, while learning can help automate planning. The remainder of the chapter is organized as follows. Section 2 motivates the integrated framework for HRI by discussing related work in learning, planning and interaction. Instances of the individual components are then described in the context of visual inputs in Sections 3.1-3.3. These instantiations are accompanied by experimental results of evaluating the corresponding algorithms in simulated domains and on wheeled and humanoid robots deployed in indoor domains. Furthermore, this chapter outlines a strategy (in Sections 3.1-3.3, Section 3.4) to integrate the individual components to achieve the desired target of robust human-robot interaction.

2. RELATED WORK

The proposed framework uses vision as a major source of information. This section motivates the integrated framework for robust human-robot interaction by discussing the limitations of existing algorithms for vision-based learning, planning and interaction.

2.1 Vision-based Learning and Planning

A robot vision system typically includes segmentation, recognition and scene understanding. Computer vision research has produced many algorithms for segmentation (Caselles, Kimmel, & Shapiro, 1997; Comaniciu & Meer, 2002; Felzenswalb & Huttenlocher, 2004), and many robot domains use labeled data to map pixels to color labels. Algorithms have also been developed to characterize (and hence recognize) objects using local image gradients (Lowe, 2004; Mikolajczyk & Schmid, 2004); appearance models (Arashloo & Kittler, 2011; Fergus, Perona, & Zisserman, 2003); hierarchical decomposition of parts (Fidler, Boben, & Leonardis, 2008); or visual cortical mechanisms (Serre, Wolf, Bileschi, Riesenhuber, & Poggio, 2007). Recent research in computer vision has also provided algorithms that use contextual cues learned from images for object recognition (Li, Parikh, & Chen, 2011). Many robot applications use these algorithms in conjunction with temporal cues and 3D range input (e.g., Kinect, RGB-D cameras, Lidar) for object recognition and scene understanding (Lai, Bo, Ren, & Fox, 2011). However, many of these algorithms are computationally expensive, sensitive to environmental changes, or require extensive domain-specific information.

Sensitivity to environmental factors is a major challenge to the use of visual features, e.g., object recognition algorithms based on image gradients or color distributions are sensitive to illumination, and stereo maps are sensitive to texture-less surfaces (Hartley & Zisserman, 2004). Algorithms developed to provide robustness to changes in environmental factors such as illumination typically require prior knowledge of illuminations and object properties and are computationally expensive (Finlayson, Hordley, & Hubel, 2001; Lammens, 1994). Since a mobile robot has to deal with unexpected changes, robot vision algorithms tend to track changes in visual feature distributions and update parameters of learned models (e.g., mixture of Gaussians) over time (Sridharan & Stone, 2007; Thrun, 2006). However, adaptation to unforeseen changes continues to be a major challenge to the use of robots in the real-world.

In parallel to the research on vision-based learning, many algorithms have been developed for automatic speech recognition and understanding using grammars and probabilistic sequential reasoning methods (Brick & Scheutz, 2007; Guedon, 2005; Rabiner, 1989), resulting in many HRI applications. However, these algorithms require significant prior knowledge, cannot adapt to dynamic changes or do not build strong associations between language and other modalities for human-robot interaction.

A mobile robot in real-world application domains such as offices and homes cannot observe the entire domain or process all sensory inputs. Planning algorithms have hence been developed to sequence sensing and information processing operators based on high-level goals. Modern AI planning algorithms that relax the limiting constraints of classical algorithms (Ghallab, Nau, & Traverso, 2004) have been used in many applications (Brenner & Nebel, 2009; Petrick & Bacchus, 2004; Talamadupula, Benton, Kambhampati, Schermerhorn & Scheutz, 2010). Probabilistic planning algorithms have also been designed for tasks such as visual gesture and object recognition (Li, Bulitko, Greiner, & Levner, 2003). In parallel, active vision algorithms have been developed for sensor placement and multi-target tracking (Kreucher, Kastella, & Hero, 2005), submodular functions have been used for sensor placement (Krause, Singh, & Guestrin, 2008) and visual target recognition has been posed as an information maximization task (Butko & Movellan, 2008). However, many of these methods require manual supervision and many visual planning tasks are not submodular. In recent years, partially observable Markov decision processes (POMDPs) have been used to plan sensory processing for behavior control, navigation and grasp planning on robots (Brook, Ciocarlie, & Hsiao, 2011; Hoey et al., 2010). Although good performance has been achieved using a hierarchy in POMDPs and other planning formulations (Marthi, Russell, & Wolfe, 2009; Pineau, Montemerlo, Pollack, Roy, & Thrun, 2003), a large portion of the data for hierarchy and model creation has to be manually encoded. To enable planning in complex domains, recent work has focused on integrating knowledge representation and logical reasoning (Chen et al., 2010; Galindo, Fernandez-Madrigal, Gonzalez, & Saffioti, 2008), and on switching between classical and probabilistic planning for robot applications (Gobelbecker, Gretton, & Dearden, 2011; Kaelbling & Lozano-Perez, 2011). Researchers have also explored tractable representations for hierarchical POMDPs (e.g., dynamic Bayes nets and factored MDPs) (Theocharous, Murphy, & Kaelbling, 2004; Toussaint, Charlin, & Poupart, 2008), but these algorithms are computationally expensive for complex application domains.

2.2 Human-robot Interaction

Developments in sensory input processing algorithms and cognitive architectures (Anderson et al., 2004; Scheutz, Schermerhorn, Kramer, & Anderson, 2007) have aided the use of robots and software agents in a wide range of applications such as human-computer interaction, elderly care and interaction with autistic children (Canemero, 2010; Robins et al., 2004). Research consortia are focusing on cognitive human-robot interaction (CogX Project, 2011), where information from different cues (e.g., vision and speech) are bound based on predetermined rules. Researchers are also integrating computational cognitive models, multiple spatial representations and sensory cues to enable human-robot collaboration, e.g., in a reconnaissance task (Kennedy et al., 2007). However, adaptive visual processing, speech understanding, knowledge representation and optimal use of human inputs are still open challenges to natural HRI (Cantrell, Scheutz, Schermerhorn, & Wu, 2010).

Two broad design approaches typically characterize HRI efforts: biologically inspired design mimics social behavior and uses theories in life sciences and social sciences, while functional design builds computational models to match the domain's social interaction needs. The limited applicability of existing HRI design guidelines causes designers to develop context-specific guidelines and evaluation methods for each domain (Thrun, 2004). An appealing approach is to analyze domain needs and use social exchange concepts to guide HRI design choices (Lawler, 2001; Wagner & Arkin, 2008).

HRI researchers have developed sophisticated algorithms for enabling a robot to operate autonomously based on sensory cues. Some algorithms use computational models of social interactions between humans, modeling the perceived outcomes of the association and the evolving short-term and long-term constraints (Kleinberg & Tardos, 2008). Research shows that robots learn better when they consider social and environmental cues in addition to mimicking the actions of a partner (Cakmak, DePalma, Arriaga, & Thomaz, 2010). Similarly, research on interactions between robots and toddlers shows that the credibility of interactions is a major contributor to the social significance assigned to a robot (Meltzoff, Brooks, Shon, & Rao, 2010). Recent research also indicates that a socially assistive robot can use verbal and visual feedback to positively impact intrinsic motivation of elderly to perform physical or cognitive tasks (Fasola & Mataric, 2010). There has also been considerable work on using embodied relational (virtual) agents in health care (Bickmore, Schulman, & Yiu, 2010; Rizzo, Parsons, Buckwalter, & Kenny, 2010). However, a key limitation of these (existing) algorithms is that they predominantly use manually-encoded domain knowledge in specific applications, and the use of robots in complex real-world domains continues to be an open challenge.

In parallel to the work on autonomous learning from sensory cues, significant research has been performed to enable a robot to learn from human demonstrations (Argall, Chernova, Veloso, & Browning, 2009; Grollman, 2010; Zang, Irani, Zhou, Isbell, & Thomaz, 2010). These approaches build mathematical models based on research in related fields such as control theory, biology and psychology, and theories of human learning and social interactions among humans. However, extensive manual training requires participants with substantial knowledge of the domain and the robot's capabilities. Although humans can provide useful information about tasks and domain, it is typically difficult for human participants to possess the expertise and time to provide elaborate and accurate feedback in complex domains.

Widespread use of robots in the real-world requires the ability to interact with non-experts (Clarkson & Arkin, 2006; Yanco, Drury, & Scholtz, 2004). In recent times, there has been some work in agent domains and on robots to use limited high-level human feedback when it is available or necessary, e.g., the *CoBot* that seeks human help to navigate to desired locations (Rosenthal et al., 2011), or the reinforcement learning-based *TAMER* framework that combines human and environmental feedbacks in simulated game domains (Knox & Stone, 2010). However, existing methods require elaborate prior knowledge of the specific task and domain or do not model the unreliability of human inputs.

Summary: Existing learning and planning algorithms have enabled the use of robots in specific applications, but they make strong assumptions regarding the task and domain, require extensive manual feedback and are computationally expensive. Existing methods for HRI have predominantly focused on teaching the robot to perform specific tasks or on enabling the robot to learn from sensory cues. Although it is intractable for the robot to learn complex models of all domain objects and events in all scenarios, it has to use the relevant information and respond in real-time to dynamic changes. Our framework addresses these challenges by exploiting the dependencies between learning, adaptation and interaction. As a result, mobile robots are able to incrementally learn object models, adapt sensing and information processing to the task at hand, and acquire and use high-level inputs from non-expert human participants based on need and availability.

3. INTEGRATED FRAMEWORK FOR HRI

The framework described in this chapter seeks to achieve robust HRI by enabling robots to operate autonomously when possible, acquiring and utilizing feedback from non-expert human participants based on need and availability. Consider the illustrative example in Section 1, where a mobile robot equipped with sensors and information processing algorithms is deployed in an office. High-level human feedback is in the form of simplistic verbal inputs that provide positive or negative reinforcement of the robot's actions, or make a choice from multiple options posed by the robot. The integrated framework consists of three components. Section 3.1 describes a bootstrap learning algorithm that enables a mobile robot to learn layered graphical models of objects and adapt to changes. Next, Section 3.2 describes a hierarchical planning algorithm that uses partially observable Markov decision processes to automatically adapt sensing, learning and processing to the task at hand. Finally, Section 3.3 describes an augmented reinforcement learning algorithm that enables a mobile robot to acquire and robustly merge high-level human feedback with the information extracted from sensory cues. As stated earlier, each section also describes how the individual algorithms in the integrated framework inform and guide each other. Furthermore, Section 3.4 illustrates the software architecture for the integrated framework in the context of the learning and planning algorithms described in Sections 3.1-3.2.

3.1 Bootstrap Learning

Figure 2 (left) shows an instance of bootstrap learning for visual inputs, where the mobile robot autonomously and incrementally: (a) learns the domain map and layered graphical object models; (b) uses the map and object models to learn visual feature models; and (c) uses the visual feature models to detect and adapt to unforeseen dynamic changes.

Existing simultaneous localization and mapping (SLAM) algorithms are used by the robot to learn and revise the domain map (Davison, Reid, Morton, & Stasse, 2007; Grisetti, Stachniss, & Burgard, 2006). Human input can be used to provide semantic labels to locations in the map. Learning of object models is then based on the observation that many real-world objects tend to possess a unique characteristics (e.g., colors and parts) and trace well-defined motion patterns, although these characteristics and patterns are not known in advance. In addition, given a learned map of the domain, the *interesting* objects are typically those that can move. Candidate objects in the images are hence identified using motion cues, i.e., by tracking local image gradient features (used in visual SLAM) and clustering the features based on relative velocity. Next, discriminative and descriptive local, global and temporal visual cues with complementary properties are extracted from these candidate image regions to populate the object models. For instance, in Figure 2 (right), gradient features, connection potentials between gradient features, graph-based image segments and color distributions are the features under consideration. The second layer of the object model represents a higher level of abstraction for robustness, e.g., relative spatial arrangement of local gradients, neighborhood relationships of connection potentials between gradient features (using Markov random fields), part-based models of image segments and second-order image statistics of color distributions. These learned models are revised incrementally over subsequent images and used to recognize stationary or moving objects.

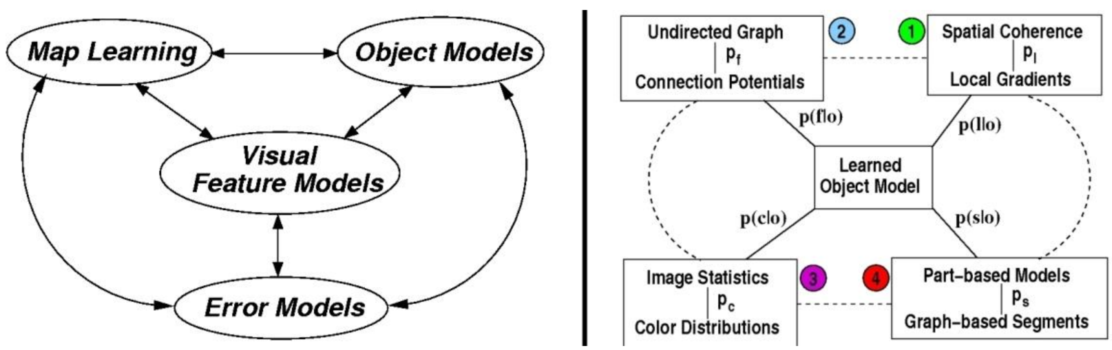

Figure 2: (Left) Visual feature models, environmental map and object models bootstrap off of each other to incrementally refine the individual models; (Right) Layered graphical models with belief propagation are used to represent domain objects.

An instance of this bootstrap learning approach was used to learn models for objects in different categories, e.g., *box, book, airplane, robot, car* and *human*, with about 5-6 different models learned for subcategories within each category. These experiments were conducted over a set of approximately 1000 images, which included images captured by the wheeled robots in Figure 1 (left) and images from computer vision benchmark datasets (e.g., *Pascal VOC 2006*). The robot autonomously learned models for moving objects and used the models to recognize stationary and moving objects in subsequent images. Experimental results indicate a high classification accuracy of 90% averaged over all categories (and subcategories within a category). Classification errors correspond to images where a sufficient number of unique features were not detected or matched with the learned object models due to motion blur or the fact that some images provide long-shots of the objects (Li & Sridharan, 2012; Li, Sridharan & Zhang, 2011). Figure 3 shows an example of using the learned object models to recognize a target object against a complex background (*blue box* on book shelf). Merging probabilistic evidence provided by individual components of the learned object models regarding occurrence of the corresponding objects in the image regions enables the robot to exploit the complementary properties of different visual cues. As a result, the robot robustly recognizes the target object (*blue box* in this example) in the appropriate image region.

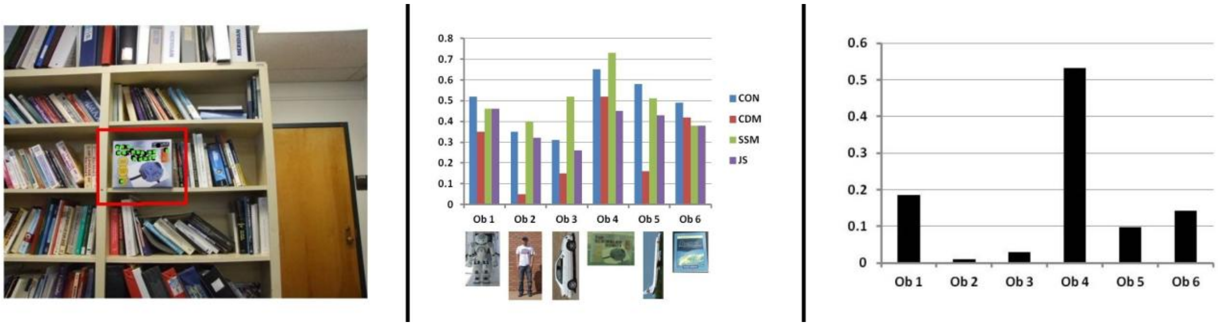

Figure 3: (Left) Test image of a box in a complex background; (Center) Individual match probabilities—the best subcategories within each category are shown along the x-axis; and (Right) Net match probabilities across different object categories—merging evidence from different components of the learned object models results in robust object recognition .

The learned map and object models are used to maintain revised models of the underlying visual features. For instance, distributions of image gradients are organized as histograms to characterize objects, while color distributions are modeled as Gaussian mixture models. Statistical bootstrap tests are used to automatically determine suitable models for different visual feature distributions. The feature models are then revised incrementally using pixels from images of objects recognized using the learned object models. When changes in object configuration or environmental factors (e.g., illumination) cause unforeseen changes in feature distributions, the learned feature models and object models are used to correlate sensor values to these factors based on the hypothesis that images from an environmental state (e.g., specific illumination) have measurably similar distributions in relevant feature spaces. A representation is learned automatically for environmental factors using visual features. The learned feature models and representations are then used to detect and adapt to changes in the corresponding environmental factors. Instances of this learning and adaptation approach has been implemented and evaluated on legged and wheeled robots that autonomously learned color distributions from domain objects (with known/learned positions and color labels) and used the learned color distribution models to detect and adapt to illumination changes (Sridharan & Stone, 2009; Sridharan & Stone, 2007). Color distributions were modeled as Gaussian mixture models and histograms, and each illumination was represented by: a mapping from pixel values (of images in that illumination) to object color labels, probability density functions (pdfs) in color space and a distribution of distances between these pdfs. When minor illumination changes caused a drift in color distributions and a slow decay of capabilities such as segmentation, the robot automatically extracted image pixels corresponding to known objects to revise the color distribution models. When sudden (or large) illumination changes caused large shifts in color distributions, the average distance between color distributions from the new illumination and the learned color distributions were well outside the range of the expected distribution of distances. If the change was to an illumination that had been modeled, the robot smoothly transitioned to using the corresponding color models for subsequent operations. On the other hand, if the change corresponded to a new illumination, the robot augmented existing color and object models to account for this novelty. This learning and adaptation approach can be used to model other visual features and environmental factors.

In practical domains, it is infeasible for the robot to observe the entire domain from a single position. The robot hence uses the learned object models, feature models and domain map to learn stochastic models that predict: (a) motion errors for different motion patterns; and (b) the likelihood of learning feature models in different locations. Motion planning then simultaneously maximizes the probability of learning desired feature (and object) models and minimizes localization errors (Ghallab, Nau & Traverso, 2004). For instance, the color distribution learning and illumination adaptation approach described above used such a motion planning algorithm to enable the robot to plan motion sequences that placed the robot in the vicinity of objects suitable for learning the color distributions (Sridharan & Stone, 2007). Furthermore, the bootstrap learning algorithm has been used to fuse stereo (visual) cues with range information on wheeled robots operating in indoor and outdoor environments (Murarka, Sridharan, & Kuipers, 2008; Sridharan & Li, 2009).

Real-world application domains are likely to contain a large number of objects that can be represented using different features, making it a challenge to learn appropriate object models (Hoiem, Efros, & Hebert, 2007). The integrated framework will address this challenge using the relationships between components of the framework. Bootstrap learning will thus be made feasible using: (a) planning to identify relevant features for characterizing domain objects relevant to the task at hand (Section 3.2); and (b) human feedback for reinforcement and disambiguation (Section 3.3). Conditional probability distributions can also be learned to model relationships between and within the layers of the object models, building richer object descriptions. Furthermore, the learning and adaptation algorithms can be revised to work with other visual (or non-visual) sensory cues.

3.2 Hierarchical Planning

In large, complex domains, a robot cannot process all sensory inputs or learn models for all domain objects. At the same time, robust operation in such domains requires that the robot make best use of all the relevant information. Based on evidence in animals and robots (Horswill, 1993; Land & Hayhoe, 2001), an appealing approach is to retain capabilities for many tasks, direct sensing to relevant locations, and consider the reliability and complexity of available algorithms to automatically determine the sequence of algorithms appropriate for any given task. This objective can be posed as a planning task and as an instance of probabilistic sequential decision-making. More specifically, this section describes the use of partially observable Markov decision processes (POMDPs) to tailor sensing and information processing to the task at hand. POMDPs elegantly model the partial observability and non-determinism of robot application domains. However, POMDP formulations of large real-world domains soon become intractable due to the exponential state explosion of such domains and the high computational complexity of even approximate POMDP solvers (Ong, Png, Hsu & Lee, 2010). A novel hierarchical decomposition is hence incorporated—Figure 4 (left) shows an instance for visual sensing and information processing. For a specific task, the high-level (HL) POMDP computes the sequence of 3D scenes to be analyzed. For a chosen scene, the intermediate-level (IL) POMDP analyzes snapshots (e.g., images) of the scene by choosing a sequence of salient regions of interest (ROIs) to be examined. Each ROI is modeled as a lower-level (LL) POMDP that computes the best sequence of algorithms to be applied on the ROI. Belief propagation between levels of the hierarchy and generation of suitable POMDP models in all levels of the hierarchy occurs autonomously at run-time. Furthermore, the hierarchy is augmented with a communication layer (CL) that enables each robot to merge the information extracted from sensory cues with the information communicated by one or more teammates (Zhang & Sridharan, 2012).

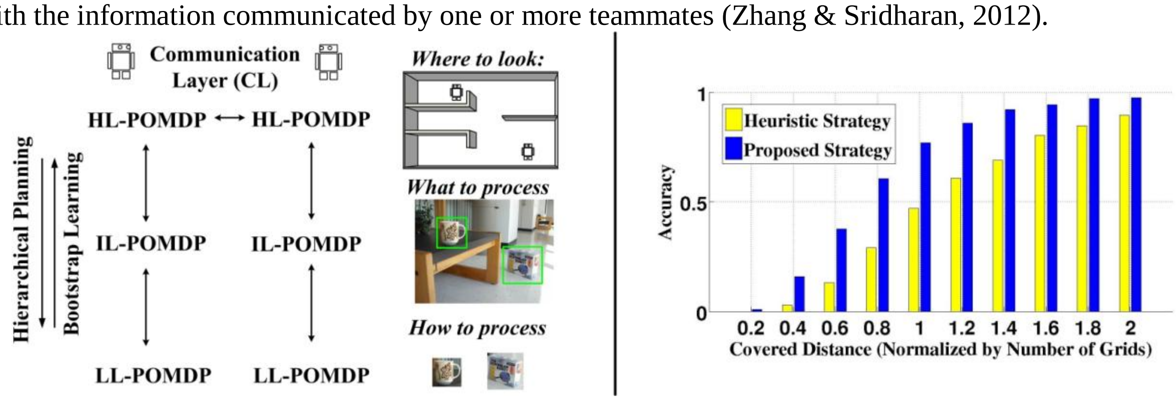

Figure 4: (Left) Layered POMDP hierarchy for visual sensing and information processing on a team of robots; (Right) Visual search based on constrained convolutional policies is more efficient than ad-hoc heuristic search strategies.

Consider the task of visually locating a human or an object in an office with multiple rooms. The HL-POMDP represents the 3D area as a 2D occupancy grid, which forms the state space. Since the true underlying state cannot be observed with certainty, a probability distribution over the grid represents the current belief and any prior knowledge about the resident's location (i.e., the *belief state*). The HL-POMDP's actions cause a robot to move to specific grids and analyze 3D scenes. Planning involves finding the best *policy* that maps belief states to stochastic action choices. The challenge is that application domains can result in large state spaces and these state spaces can change in response to domain changes. For efficient operation over large areas, shift and rotation symmetries of visual search are exploited to learn a convolutional policy kernel from the policy for a grid map of a small region. The policies for grid maps of large areas are then generated automatically (at run-time) by performing an inexpensive convolution operation with the learned policy kernel. Action utilities in the HL-POMDP are modeled as the expected information gain, i.e., the reduction in entropy of the corresponding belief vectors. In addition, the observation functions of the HL-POMDP are computed automatically based on the learned observation functions of lower levels of the hierarchy. During plan execution, the computed policy is used to repeatedly choose an action and update beliefs based on the observed outcome. A key benefit of this approach is that domain map changes (e.g., objects are moved or doors are closed) are addressed automatically by suitably re-weighting the computed policy. In addition, the cost of robot motion is modeled by re-weighting the learned policy to trade-off distance of travel against the likelihood of finding the desired targets. Figure 4 (right) shows results where a robot located targets in a 15×15 simulated grid—each point in the figure is the average over 1000 trials, with the convolutional policy computed from a 5×5 policy kernel. As seen in Figure 4 (right), for any desired accuracy (along the *y-axis*), convolutional policies locate target objects much faster than heuristic (i.e., greedy) search policies.

Once the robot moves to a chosen grid-cell, it analyzes snapshots (e.g., images) of the scene. To locate a target, ROIs in a snapshot, shown enveloped in green rectangles in Figure 4 (left), can be processed using a wide range of visual operators based on bootstrap-learned (object) models, e.g., object recognition operators that use gradient features or parts. However, the POMDP in the joint space of image ROIs soon becomes intractable, e.g., there are approximately 50000 states for just three ROIs and two actions with six outcomes (each). The POMDP hierarchy partially ameliorates this state explosion challenge by modeling each ROI with an LL-POMDP, and using an IL-POMDP to select the ROI to be analyzed further using the corresponding LL policies. The IL-POMDP hence controls the application of relevant processing algorithms to examine all the ROIs in an image of the chosen scene. The IL-POMDP model parameters (e.g., reward specification and observation functions) are generated automatically at run-time based on the corresponding LL policies and propagated belief. Similarly, relevant LL-POMDP models are learned automatically for any image ROI using bootstrap learning and minimal human supervision. Furthermore, each robot probabilistically merges current beliefs with the beliefs communicated by teammates, enabling the team of robots to collaborate robustly (despite unreliable communication) to achieve a shared objective, e.g., find one or more target objects in the domain.

Instances of the IL and LL of the hierarchy have enabled robots to collaborate with humans to jointly manipulate and converse about tabletop objects (Sridharan, Wyatt & Dearden, 2010; Sridharan, Wyatt & Dearden, 2008). Instances of the entire hierarchy have enabled mobile robots to locate target objects in dynamic indoor domains such as offices (Zhang & Sridharan, 2012; Zhang & Sridharan, 2011). These experiments indicate that the hierarchical planning algorithm significantly reduces planning time in comparison to the POMDP in the joint space of all ROIs, as shown in Figure 5 (left). The hierarchical planning approach is also as efficient as state of the art contingency planners while providing substantially higher reliability (Sridharan, Wyatt & Dearden, 2010). Furthermore, robots are able to share information to collaborate robustly with teammates despite unreliable sensing and communication. As shown in Figure 5 (right), belief sharing in conjunction with the POMDP hierarchy enables the team to identify targets with high accuracy in a much smaller number of action steps in comparison to an ad-hoc collaboration strategy.

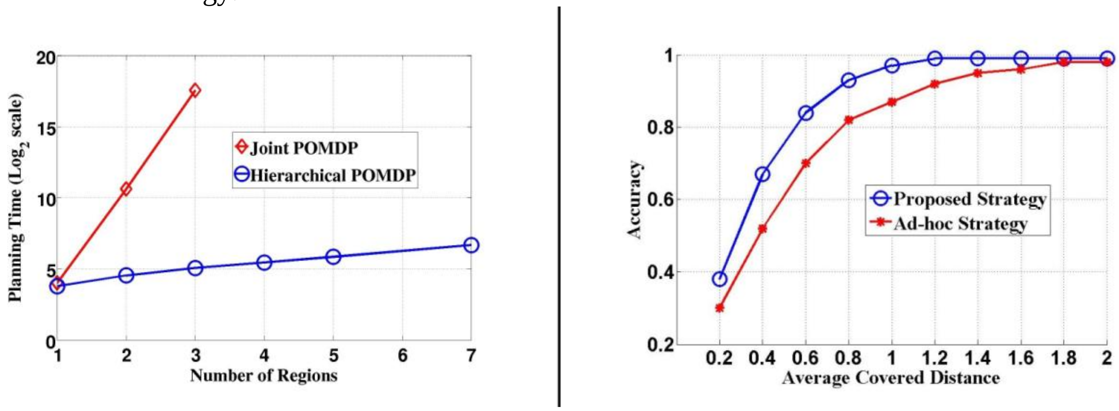

Figure 5: (Left) Hierarchical POMDP significantly reduces planning time in comparison to the POMDP over the joint state space of all image ROIs; (Right) Merging beliefs obtained by processing sensory cues with the communicated beliefs enables a team of robots to localize targets accurately while traveling a much smaller distance than with an ad-hoc probabilistic collaboration strategy.

In real-world application domains with large state spaces and dynamic changes, automated planning and decision-making is a challenge. This challenge will be addressed using the hierarchical planning algorithm in conjunction with other components of the integrated framework. Mobile robots will then be able to adapt learning and planning to the domain and the corresponding tasks by: (a) representing and revising domain knowledge, performing logical reasoning and acquiring information from other robots or humans (e.g., feedback in the form of reinforcement and disambiguation) (Zhang, Bao & Sridharan, 2012); and (b) identifying relevant objects that need to be learned and features that will capture the most information about these objects.

3.3 Augmented Reinforcement Learning

For widespread deployment of robots to interact with humans in real-world domains, robots equipped with the learning and planning algorithms described above still need a strategy to acquire and use limited feedback from non-expert human participants. This objective poses two questions: (Q1) how best to robustly merge high-level human input with the information extracted from sensory cues? and (Q2) when and how should human feedback be acquired?

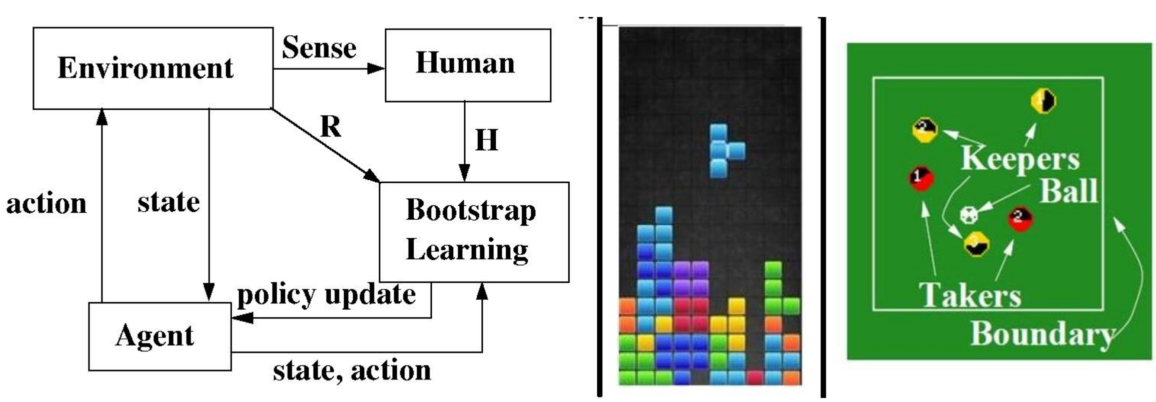

Figure 6: (Left) Augmented reinforcement learning enables the robot to bootstrap off of high-level human feedback and environmental feedback in the form of sensory inputs; (Center) Single-agent Tetris domain; and (Right) Multiagent 3vs.2 Keepaway domain.

Tasks that require an agent or a robot to learn from repeated interactions with the environment can be posed as a Reinforcement learning (RL) problem. RL is a well-established computational approach, where the desired task is modeled as a Markov decision process (MDP) and an agent repeatedly performs actions to receive a state estimate and a reward signal (Sutton & Barto, 1998). The RL framework has been used in many application domains to enable agents and robots to learn suitable action policies (i.e., mapping from states to actions). As stated earlier, we consider high-level feedback from non-expert human participants—for ease of explanation, this section only considers positive or negative reinforcement of actions, e.g., *yes/no* feedback. Including *human feedback* **H** in the RL framework is a challenge because **H** may not fit in the same range as *environmental feedback* **R** obtained from sensory inputs. In addition, **H** may be in response to a set of past (or even future) states and actions. Figure 6 (left) shows the augmented reinforcement learning (ARL) approach that is used to answer Q1, i.e., to robustly merge **R** and **H**. In the absence of human feedback, the robot uses the standard RL formulation, i.e., a baseline RL algorithm is used to learn an action policy by observing the effects of actions performed in various states. When human feedback is available, the robot uses automatically-computed performance measures (e.g., time for task completion) to bootstrap off of the two feedback signals and incrementally revise their relative contributions to the action choice policy. Specifically, the robot estimates parameters of a function that merges **R** and **H** such that the actions chosen by the resultant policy maximize the performance measure(s): $\mathrm{argmax}_{a \in A}\ f(\mathrm{R}, \mathrm{H})$. For ease of explanation, consider the weighted linear combination function: $\mathrm{argmax}_{\mathrm{a} \in \mathrm{A}} \{\mathrm{w}_\mathrm{r}\ \mathrm{R} + \mathrm{w}_\mathrm{h}\ \mathrm{H}\}$ in the fully observable simulated game domains of Tetris and Keepaway soccer—see Figure 6 (center) and Figure 6 (right). The objective in *Tetris* is to maximize episode length by dropping blocks such that they complete and hence clear lines. In multiagent *Keepaway*, the objective is to maximize episode length by enabling keepers to retain ball possession from the takers. In the absence of human feedback, the agent(s) in these domains learn a policy from **R** (using the baseline RL algorithm) and invoke the top N action policies proportional to their relative ability to maximize episode length. When human feedback is available, the weights corresponding to feedback signals (w_h and w_r) are continuously and incrementally revised based on the degree of match between **H** and **R**, and their relative ability to maximize episode length. The individual feedback signals are thus merged to provide the overall action choice policy. Figure 7 shows results of experiments in the Tetris and Keepaway domains, using high-level feedback from four human participants 2-5 times per episode (the *yes/no* feedback signals are mapped to real-valued rewards). Figure 7 (left) shows the result of using a weighted linear combination function in the Tetris domain, using *policy gradient* (Sutton & Barto, 1998) as the baseline RL algorithm. The ARL approach to merge **R** and **H** significantly increases the episode length in comparison to using **R** or **H** (not shown in Figure) individually (Sridharan, 2011).

Next, Figure 7 (right) shows experimental results in the 3 vs. 2 Keepaway domain, using the SMDP version of $Sarsa(\lambda)$ (Stone, Sutton & Kuhlmann, 2005) as the baseline algorithm. This domain changes too quickly for instantaneous human feedback of the agents' action choices. A gamma distribution is hence learned experimentally to model typical human response times. This distribution is used for credit assignment over past states and actions. As seen in Figure 7 (right), using the ARL approach significantly increases episode duration in comparison to the individual feedback signals. In addition, using the learned gamma function for credit assignment further improves the episode duration. Furthermore, when different humans participating in the experimental trials provide intentionally incorrect feedback, agents are able to recover by revising weights of feedback signals (Sridharan, 2011).

Since the ARL approach uses belief distributions computed in hierarchical planning to estimate state, Q2 is answered using information-theoretic measures and bootstrap learning algorithms. The state with the maximum belief is considered to be the true state and the entropy of belief distributions is used as a measure of uncertainty. Asking for human input is modeled as a sensing action that is sequenced to maximize information gain (Zhang, Bao & Sridharan, 2012), similar to the topmost level of the POMDP hierarchy described in Section B.3.2.

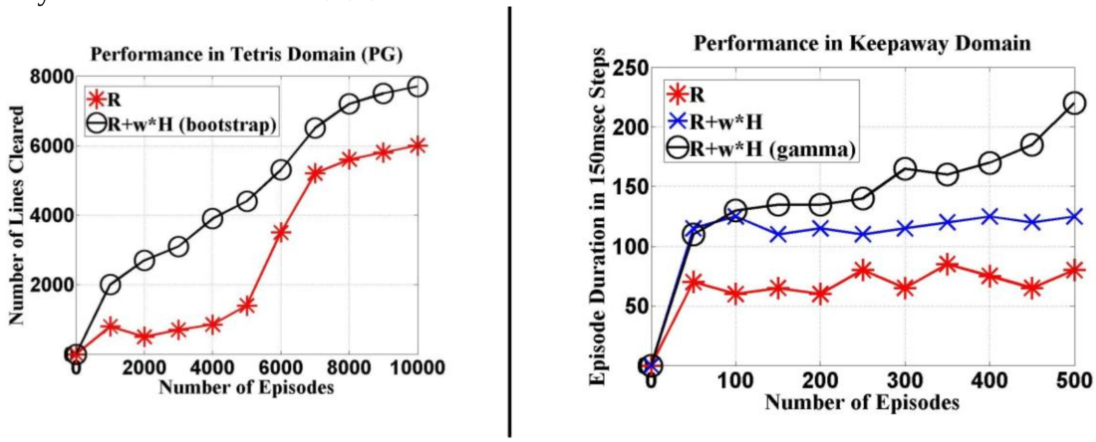

Figure 7: Results of proof of concept experiments in simulated domains: (Left) single-agent Tetris domain; and (Right) multiagent Keepaway domain. Merging human and environmental feedbacks significantly increases the episode length in comparison to individual feedback mechanisms. Probabilistic credit assignment over past states and actions further improves performance.

The ARL approach has been described (above) in the context of simulated agent domains. RL typically requires knowledge of state and an estimate of the transition and reward functions—these are not readily available in robot application domains. The integrated framework will address this challenge by defining rewards based on information gain and global performance measures, states based on the belief states used in hierarchical planning (Section B.3.2), and transition functions based on bootstrap learning and limited domain knowledge (Section B.3.1). Bootstrap learning will also provide the models necessary to estimate the likelihood of obtaining relevant information through different query types. Furthermore, the ARL algorithm will be used in conjunction with an algorithm that learns associations between visual and verbal object descriptions, enabling simplistic natural language interactions between humans and agents (Swaminathan & Sridharan, 2011). The robot will thus be able to initiate and sustain interactions with appropriate human participants.

3.4 Integration Overview

Finally, consider the architecture that integrates the bootstrap learning, hierarchical planning and augmented reinforcement learning algorithms described above. To enable modular software development, all algorithms were implemented using the popular Robot Operating System (ROS) (Quigley et al., 2009). Figure 8 presents a subset of the architecture that focuses on visual bootstrap learning and hierarchical (probabilistic) planning—the corresponding graph was generated by the ROS command-line option: <*rxgraph*>. The individual nodes are described below.

The hierarchical planning algorithm (Section 3.2) is placed within the *vs_planner* node, while the visual bootstrap learning algorithm (Section 3.1) is placed within the *vs_vision* node. Communication between nodes is achieved by publishing topics, i.e., by passing messages. The *vs_vision* node repeatedly processes input images to learn relevant visual object models. The learned object models are used to recognize objects in subsequent images, populating the <*v_pack*> package that is sent to the *vs_planner* node. This package contains the ID of each detected object, in addition to the distance and bearing of the object (relative to the robot) and a (probability) measure of the uncertainty associated with the observation of the object. These observations are used to perform belief updates within the planning module, as described in Section 3.2. Belief updates occur: (1) when the robot arrives at a desired grid cell and processes one or more images of the scene—belief updates consider presence and absence of the target object; or (2) when the robot detects the target by processing images while moving to a desired grid cell. After the belief update, the planner node sends the coordinates of any desired grid cell to the movement control node *move_base* (in the *goal* message) and waits for a response. The *move_base* node receives the current domain map from the *map_server* (which can also perform simultaneous localization and mapping—SLAM) and laser range information from *hokuyo_node*, which contains the driver for the laser range finder. The *move_base* node also receives navigation goals (if any) from humans through *navigation_goals*, in addition to pose and odometry information from *amcl* and *erratic_base_driver* respectively. The *erratic_base_driver* provides the robot-specific coordinate frames (in *tf*) and the driver for the specific robot platform used in these experiments (e.g., the *erratic* wheeled robot in Figure 1). The *amcl* node performs localization using particle filters to provide the pose estimate. The *move_base* node uses A* search to find a path to the desired grid cell and provides linear and angular velocity commands to the robot's driver (in *cmd_vel*). These commands result in the robot's motion and one of three responses: *arrived*, *canceled* or *not-arrived*. The *arrived* response is received when the robot reaches the desired location, while the *canceled* response represents unexpected cancellation of the motion command. The *not-arrived* response is usually the result of a dynamic change in the environment, e.g., closing a door makes an office unavailable to the robot. Additional nodes are used (in a similar manner) to create instances of other algorithms, e.g., for augmented reinforcement learning using human feedback.

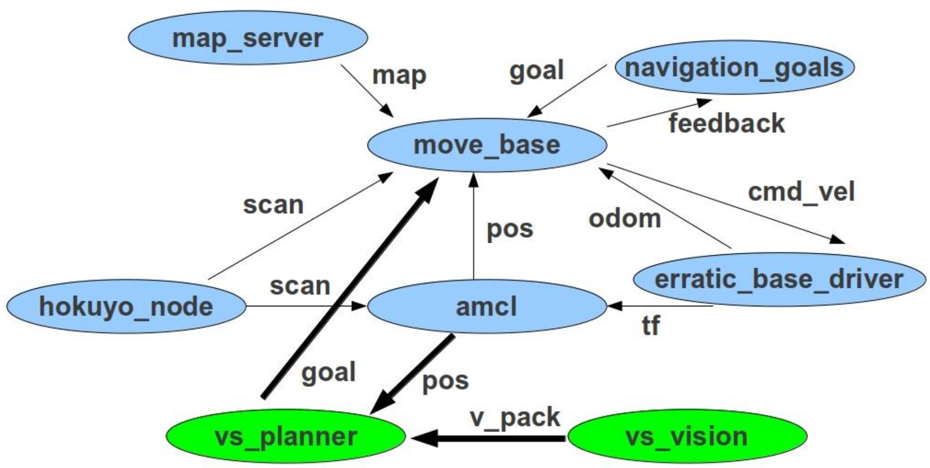

Figure 8: A ROS-based framework for integrating different components. Interaction between hierarchical planning, visual bootstrap learning and other control modules is illustrated.

This architecture has been successfully implemented and used on wheeled robots deployed in indoor office domains (Zhang & Sridharan, 2012). The modular architecture makes it easy to revise specific algorithms (e.g., for autonomous learning or planning) and to use the architecture on other robot platforms in different application domains. Furthermore, other nodes can be added (as and when required) to create instances of algorithms that augment the existing components in the integrated framework. Results of recent experimental trials, including some images and video demos can be viewed online: http://www.cs.ttu.edu/~smohan/RobotAssist.html

4. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

This chapter described a novel integrated (probabilistic) framework that jointly addressed the learning, adaptation and interaction challenges associated with robust human-robot interaction in real-world application domains. This framework consists of three components: (1) a bootstrap learning algorithm that enables mobile robots to autonomously learn layered graphical models of environmental objects, and to detect and adapt to unforeseen domain changes; (2) a hierarchical planning algorithm based on partially observable Markov decision processes that enables a team of robots to collaborate robustly by sharing beliefs and automatically adapting sensing and information processing to the task at hand; and (3) an augmented reinforcement learning algorithm that enables robots to acquire limited high-level feedback from non-expert human participants, and to robustly merge human feedback with the information extracted from sensory cues. Instances of these algorithms have been implemented in a modular software architecture and evaluated on mobile robots and simulated agents interacting with non-expert human participants in indoor office domains and multiagent game domains.

As stated in Section 1 (and illustrated in Sections 3.1-3.4), the integrated framework enables the individual algorithms to inform and guide each other, posing novel challenges and providing new opportunities to address the tough challenges in the individual fields. Future work will investigate the full integration of the bootstrap learning, hierarchical planning and augmented reinforcement learning algorithms. For instance, planning will be used to choose the objects and events relevant to the tasks that need to be performed, and to identify features suitable for modeling these objects, e.g., to select visual features that provide the most information about the objects of interest. Similarly, bootstrap learning will be used to autonomously learn the model parameters required to automate hierarchical planning in complex domains. Future work will also focus on building richer object descriptions by integrating visual and verbal cues, resulting in natural language interactions between robots and humans.

As mobile robots are increasingly deployed to interact with humans in real-world application domains such as homes and offices, there is a pressing need for enabling robots to operate autonomously by learning from sensory cues and high-level feedback from non-expert human participants. The integrated framework described in this chapter represents a significant (and novel) step towards this long-term goal of robust human-robot interaction in a wide range of real-world application domains.

ACKNOWLEDGMENTS

The author thanks Shiqi Zhang, Xiang Li, Batbold Myagmarjav and Mamatha Aerolla for their help in implementing the algorithms, performing the experimental trials and gathering the results reported in this chapter. The author is also grateful to the following colleagues for many discussions that have guided the research reported in this chapter: Peter Stone, Ian Fasel, Benjamin Kuipers, Jeremy Wyatt, Richard Dearden, Nick Hawes and Aaron Sloman. The research reported in this chapter was supported in part by the ONR Science of Autonomy award N00014-09-1-0658.

REFERENCES

AAAI Spring Symposium on Designing Intelligent Robots: Reintegrating AI (SS02). (2012). (Retrieved October 2011 from http://www.aaai.org/Symposia/Spring/sss12symposia.php#ss02)

Aboutalib, S., & Veloso, M. (2010, October). Multiple-Cue Object Recognition in Outside Datasets. *IEEE International Conference on Intelligent Robots and Systems (IROS)* (p. 4554-4559), Taipei.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An Integrated Theory of the Mind. *Psychological Review*, 111(4), 1036-1060.

Arashloo, S. R., & Kittler, J. (2011). Energy Normalization for Pose-Invariant Face Recognition Based on MRF Model Image Matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33, 1274–1280.

Argall, B., Chernova, S., Veloso, M., & Browning, B. (2009). A Survey of Robot Learning from Demonstration. *Robotics and Autonomous Systems*, 57(5), 469-483.

Bickmore, T., Schulman, D., & Yiu, L. (2010). Maintaining Engagement in Long-term Interventions with Relational Agents. *Journal of Applied Artificial Intelligence; special issue on Intelligent Virtual Agents*, 24(6), 648-666.

Brenner, M., & Nebel, B. (2009). Continual Planning and Acting in Dynamic Multiagent Environments. *Journal of Autonomous Agents and Multiagent Systems*, 19(3), 297-331.
Brick, T., & Scheutz, M. (2007, March). Incremental Natural Language Processing for HRI. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)* (p. 263-270). Washington D.C., USA.
Brook, P., Ciocarlie, M., & Hsiao, K. (2011). Collaborative Grasp Planning with Multiple Object Representations. In *International Conference on Robotics and Automation (ICRA).*
Butko, N. J., & Movellan, J. R. (2008). I-POMDP: An Infomax Model of Eye Movement. In *International Conference on Development and Learning (ICDL)*.
Cakmak, M., DePalma, N., Arriaga, R., & Thomaz, A. (2010). Exploiting Social Partners in Robot Learning. *Autonomous Robots*, 29, 309-329.
Canemero, L. (2010). *The HUMAINE Project*. (Retrieved October 2010 from http://emotion-research.net/)
Cantrell, R., Scheutz, M., Schermerhorn, P., & Wu, X. (2010). Robust Spoken Instruction Understanding for HRI. In *ACM/IEEE International Conference on Human-Robot Interaction (HRI)*.
Caselles, V., Kimmel, R., & Shapiro, G. (1997). Geodesic Active Contours. *International Journal of Computer Vision*, 22(1), 61-79.
Chen, X., Ji, J., Jiang, J., Jin, G., Wang, F., & Xie, J. (2010, May 10-14). Developing High-Level Cognitive Functions for Service Robots. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Toronto, Canada.
Clarkson, E., & Arkin, R. C. (2006). Applying Heuristic Evaluation to Human-Robot Interaction Systems (Tech. Rep.). Georgia Institute of Technology GIT-GVU- 06-08.
CogX. (2011). Cognitive Systems that Self-Understand and Self-Extend. (Retrieved October 2011 from http://cogx.eu/)
Comaniciu, D., & Meer, P. (2002). Mean Shift: A Robust Approach Towards Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 603-619.
Davison, A. J., Reid, I. D., Morton, N. D., & Stasse, O. (2007). MonoSLAM: Real-Time Single Camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1052–1067.
Fasola, J., & Mataric, M. (2010, August). Robot Motivator: Increasing User Enjoyment and Performance on a Physical/Cognitive Task. In *International Conference on Development and Learning (ICDL)*. Ann Arbor, USA.
Felzenswalb, P. F., & Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2), 167-181.
Fergus, R., Perona, P., & Zisserman, A. (2003). Object Class Recognition by Unsupervised Scale-Invariant Learning. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.
Fidler, S., Boben, M., & Leonardis, A. (2008). Similarity-based Cross-Layered Hierarchical Representation for Object Categorization. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*.
Finlayson, G., Hordley, S., & Hubel, P. (2001, November). Color by Correlation: A Simple, Unifying Framework for Color Constancy. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11), 1209-1221.
Fong, T., Nourbakhsh, I., and Dautenhahn. K. (2003). A Survey of Socially Interactive Robots. *Robotics and Autonomous Systems*, 42(3-4):143–166.
Galindo, C., Fernandez-Madrigal, J.-A., Gonzalez, J., & Saffioti, A. (2008). Robot Task Planning using Semantic Maps. *Robotics and Autonomous Systems*, 56(11), 955-966.
Ghallab, M., Nau, D., & Traverso, P. (2004). *Automated Planning: Theory and Practice*. San Francisco, CA 94111: Morgan Kaufmann.
Gobelbecker, M., Gretton, C., & Dearden, R. (2011). A Switching Planner for Combined Task and Observation Planning. In *National Conference on Artificial Intelligence (AAAI)*. San Francisco, USA.
Grisetti, G., Stachniss, C., & Burgard, W. (2006). Improved Techniques for Grid Mapping with Rao-Blackwellized Particle Filters. *IEEE Transactions on Robotics*, 23(1):34–46.

Grollman, D. (2010). Teaching Old Dogs New Tricks: Incremental Multimap Regression for Interactive Robot Learning from Demonstration. Unpublished doctoral dissertation, Department of Computer Science, Brown University.
Guedon, Y. (2005). Hybrid Markov/semi-Markov Chains. *Computational Statistics and Data Analysis*, 49, 663–688.
Hartley, R., & Zisserman, A. (2004). *Multiple View Geometry in Computer Vision (2nd Ed.)*. Cambridge University Press.
Hoey, J., Poupart, P., Bertoldi, A., Craig, T., Boutilier, C., & Mihailidis, A. (2010). Automated Handwashing Assistance for Persons with Dementia using Video and a Partially Observable Markov Decision Process. *Computer Vision and Image Understanding*, 114(5), 503-519.
Hoiem, D., Efros, A., & Hebert, M. (2007). Recovering Surface Layout from an Image. *International Journal of Computer Vision*, 75(1):151–172.
Horswill, I. (1993). Polly: A Vision-Based Artificial Agent. In *National Conference on Artificial Intelligence (AAAI)* (p. 824-829).
Kaelbling, L., & Lozano-Perez, T. (2011). Domain and Plan Representation for Task and Motion Planning in Uncertain Domains. In *IROS 2011 Workshop on Knowledge Representation for Autonomous Robots*.
Kennedy, W., Bugajska, M., Marge, M., Adams, W., Fransen, B., Perzanowski, D., et al. (2007). Spatial Representation and Reasoning for Human-Robot Interaction. In *Twenty-Second Conference on Artificial Intelligence (AAAI)* (p. 1554-1559), Toronto, Canada.
Kleinberg, J., & Tardos, E. (2008, May 17-20). Balanced Outcomes in Social Exchange Networks. In *ACM Symposium on Theory of Computing*.
Knox, W. and Stone, P. (2010, May). Combining Manual Feedback with Subsequent MDP Reward Signals for Reinforcement Learning. *In International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.
Krause, A., Singh, A., & Guestrin, C. (2008). Near-optimal Sensor Placements in Gaussian Processes: Theory, Efficient Algorithms and Empirical Studies. *Journal of Machine Learning Research*, 9, 235-284.
Kreucher, C., Kastella, K., & Hero, A. (2005). Sensor Management Using An Active Sensing Approach. *IEEE Transactions on Signal Processing*, 85(3), 607-624.
Lai, K., Bo, L., Ren, X., & Fox, D. (2011, May 9-13). Sparse Distance Learning for Object Recognition Combining RGB and Depth Information. In *International Conference on Robotics and Automation (ICRA)*. Shanghai, China.
Lammens, J. M. G. (1994). A Computational Model of Color Perception and Color Naming. Doctoral dissertation, Computer Science Department, State University of New York at Buffalo, NY.
Land, M. F., & Hayhoe, M. (2001). In What Ways do Eye Movements Contribute to Everyday Activities? *Vision Research*, 41:3559–3565.
Lawler, E. J. (2001). An Affect Theory of Social Exchange. The American Journal of Sociology, 107(2):321–352.
Li, C., Parikh, D., & Chen, T. (2011, November 6-13). Extracting Adaptive Contextual Cues from Unlabeled Regions. In *International Conference on Computer Vision (ICCV)*. Barcelona, Spain.
Li, L., Bulitko, V., Greiner, R., & Levner, I. (2003). Improving an Adaptive Image Interpretation System by Leveraging. In *Australian and New Zealand Conference on Intelligent Information Systems*.
Li, X., & Sridharan, M. (2012, June 5). Vision-based Autonomous Learning of Object Models on a Mobile Robots. In *Autonomous Robots and Multirobot Systems* (ARMS) Workshop at the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Valencia, Spain.
Li, X., Sridharan, M., & Zhang, S. (2011, May 9-13). Autonomous Learning of Vision-based Layered Object Models on Mobile Robots. In *International Conference on Robotics and Automation (ICRA)*. Shanghai, China.
Lowe, D. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2), 91-110.

Marthi, B., Russell, S., & Wolfe, J. (2009, August). *Angelic Hierarchical Planning: Optimal and Online Algorithms*, Technical Report UCB/EECE-2009-122, EECS Department, University of California Berkeley.
Meltzoff, A., Brooks, R., Shon, A., & Rao, R. (2010, October-November). Social Robots are Psychological Agents for Infants: A Test of Gaze Following. *Neural Networks*, 23(8-9), 966-972.
Mikolajczyk, K., & Schmid, C. (2004). Scale and Affine Invariant Interest Point Detectors. International Journal of Computer Vision (IJCV), 60(1), 63-86.
Murarka, A., Sridharan, M., & Kuipers, B. (2008). Detecting Obstacles and Drop-offs using Stereo and Motion Cues for Safe Local Motion. In *International Conference on Intelligent Robots and Systems (IROS)*, Nice, France.
Ong, S. C., Png, S. W., Hsu, D., & Lee, W. S. (2010, July). Planning Under Uncertainty for Robotic Tasks with Mixed Observability. *International Journal of Robotics Research*, 29(8):1053–1068.
Pineau, J., Montemerlo, M., Pollack, M., Roy, N., & Thrun, S. (2003). Towards Robotic Assistants in Nursing Homes: Challenges and Results. *Robotics and Autonomous Systems, Special Issue on Socially Interactive Robots*, 42(3-4), 271-281.
Petrick, R., & Bacchus, F. (2004). Extending the Knowledge-Based Approach to Planning with Incomplete Information and Sensing. In *International Conference on Automated Planning and Scheduling (ICAPS)*.
Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., & Ng, A. (2009). ROS: An Open-Source Robot Operating System. In ICRA Workshop on Open Source Software.
Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2), 257-286.
Rizzo, A., Parsons, T., Buckwalter, G., & Kenny, P. (2010, March 21). A New Generation of Intelligent Virtual Patients for Clinical Training. In *IEEE Virtual Reality Conference*. Waltham, USA.
Robins, B., Dautenhahn, K., Boekhorst, R., Billard, A., Keates, S., Clarkson, J., et al. (2004). *Effects of Repeated Exposure of a Humanoid Robot on Children with Autism*. Springer-Verlag.
Rosenthal, S., Veloso, M., and Dey, A. (2011, August). Learning Accuracy and Availability of Humans who Help Mobile Robots. *Twenty-Fifth Conference on Artificial Intelligence (AAAI)*, San Francisco, USA.
Scheutz, M., Schermerhorn, P., Kramer, J., & Anderson, D. (2007). First Steps Towards Natural Human-Like HRI. *Autonomous Robots*, 22(4), 411-423.
Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., & Poggio, T. (2007, March). Robust Object Recognition with Cortex-Like Mechanisms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3).
Sridharan, M. (2011, December 18-21). Augmented Reinforcement Learning for Interaction with Non-Expert Humans in Agent Domains. In *International Conference on Machine Learning Applications (ICMLA)*, Honolulu, Hawaii.
Sridharan, M., & Li, X. (2009). Learning Sensor Models for Autonomous Information Fusion on a Humanoid Robot. In *IEEE-RAS International Conference on Humanoid Robots*. Paris, France.
Sridharan, M., & Stone, P. (2009). Color Learning and Illumination Invariance on Mobile Robots: A Survey. *Robotics and Autonomous Systems (RAS)*, Volume 57, number 6-7, pages 629-644.
Sridharan, M., & Stone, P. (2007). Global Action Selection for Illumination Invariant Color Modeling. In *International Conference on Intelligent Robots and Systems (IROS)*. San Diego, USA.
Sridharan, M., Wyatt, J., & Dearden, R. (2010). Planning to See: A Hierarchical Approach to Planning Visual Actions on a Robot using POMDPs. *Artificial Intelligence*, 174, 704-725.
Sridharan, M., Wyatt, J., & Dearden, R. (2008, September). HiPPo: Hierarchical POMDPs for Planning Information Processing and Sensing Actions on a Robot. In *International Conference on Automated Planning and Scheduling (ICAPS)*. Sydney, Australia.
Stone, P., Sutton, R., & Kuhlmann, G. (2005). Reinforcement learning for RoboCup Soccer Keepaway. *Adaptive Behavior*, 13:165–188.

Sutton, R. L., & Barto, A. G. (1998). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, USA.
Swaminathan, R., & Sridharan, M. (2011, December). *Towards Natural Human-Robot Interaction using Multimodal Cues*. Technical Report, Department of Computer Science, Texas Tech University.
Talamadupula, K., Benton, J., Kambhampati, S., Schermerhorn, P., & Scheutz, M. (2010). Planning for Human-Robot Teaming in Open Worlds. *ACM Transactions on Intelligent Systems and Technology*, 1(2), 14:1-14:24.
Tapus, A., Mataric, M., and Scassellati, B. (2007, March). The Grand Challenges in Socially Assistive Robotics. *Robotics and Automation Magazine, Special Issue on Grand Challenges in Robotics*, 14(1):35–42.
Theocharous, G., Murphy, K., & Kaelbling, L. P. (2004). Representing Hierarchical POMDPs as DBNs for Multi-scale Robot Localization. In *International Conference on Robotics and Automation (ICRA)*.
Thrun, S. (2006). Stanley: The Robot that Won the DARPA Grand Challenge. Field Robotics, 23(9):661–692.
Thrun, S. (2004). Toward a Framework for Human-robot Interaction. *Human-Computer Interaction*, 19(1):9-24.
Toussaint, M., Charlin, L., & Poupart, P. (2008). Hierarchical POMDP Controller Optimization by Likelihood Maximization. In *Uncertainty in AI (UAI)*.
Wagner, A., & Arkin, R. (2008). Analyzing Social Situations for Human-Robot Interaction. Interaction Studies, 9(2):277–300.
Yanco, H. A., Drury, J. L., & Scholtz, J. (2004). Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition. *Human-Computer Interaction*, 19(1), 117–149.
Zang, P., Irani, A., Zhou, P., Isbell, C., & Thomaz, A. (2010). Using Training regimens to Teach Expanding Function Approximators. In *International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (p. 341-348).
Zhang, S., Bao, F. S., & Sridharan, M. (2012, June 5). ASP-POMDP: Integrating Non-monotonic Logical Reasoning and Probabilistic Planning on Mobile Robots. In *Autonomous Robots and Multirobot Systems* (ARMS) Workshop at the International Conference on Autonomous Agents and Multiagent Systems (AAMAS), Valencia, Spain.
Zhang, S., & Sridharan, M. (2012, June 4-8). Active Visual Search and Collaboration on Mobile Robots. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. Valencia, Spain.
Zhang, S., & Sridharan, M. (2011, August 7-8). Visual Search and Multirobot Collaboration on Mobile Robots. In *International Workshop on Automated Action Planning for Autonomous Mobile Robots*. San Francisco, USA.