Integration of Robotic Perception, Action, and Memory

Li Yang Ku, Erik Learned-Miller, and Rod Grupen
College of Information and Computer Sciences
University of Massachusetts Amherst, Amherst, MA
{lku, elm, grupen}@cs.umass.edu

1. Introduction

To act autonomously in an unstructured environment, it would be beneficial for a robot to integrate its perception and past experience to handle a wide variety of situations. However, traditional approaches generally represent action and perception separately—as object models in computer vision and as action templates in robot controllers. Due to this separation, the robot can only interact with objects based on learned models when the object label is identified. Interacting based on object labels is not only vulnerable to recognition errors but also limits how past experiences can be generalized to novel situations.

In the book "On Intelligence" [4], Hawkins introduces the memory-prediction framework and proposes that intelligence should be measured by the capacity to memorize and predict patterns. Hawkins asserts that *"Your brain receives patterns from the outside world, stores them as memories, and makes predictions by combining what it has seen before and what is happening now"*. The proposed framework extends this concept and shows the capability of a memory model that integrates action and perception. With this integrated model, a robot would be capable of solving tasks by predicting perceptual action consequences based on memory and observation. This paper gives a broad overview of the proposed framework, and reviews our previous work that has investigated various components of the framework.

2. Framework

Figure 1 shows a modified conceptual diagram of the neocortex taken from the book "On Intelligence". Blocks with the same vertical positions represent neurons of the same cortex layer and arrows represent the direction of the information flow based on neuron connections. A neuron in a higher layer represents more abstract notions while a neuron in a lower layer represents simpler features. For example, visual neurons in a higher layer have larger receptive fields, represent object categories, and change more slowly over time. In this figure, memory regions that connect sensory neurons and motor neurons of the same layer are added to the original diagram. These memory regions associate neurons across modalities and can be used to infer bottom up signals that are missing. The connection loops within memory regions indicate predictions made based on observations, motor commands, and past memories. These memory regions have connections similar to the pyramidal neurons in the neocortex that have many connections within the same layer and an extended axon that sends signal to distant regions. However, these conjectured connections of the memory region are not based on neurological discoveries but on computational structures that shown to be practical in solving robotic tasks. The colored blocks and connections are implemented in the proposed framework and tested on robotic systems. In the following, we describe the memory model and the hierarchical structure in this diagram and show how they can be learned from demonstrations efficiently.

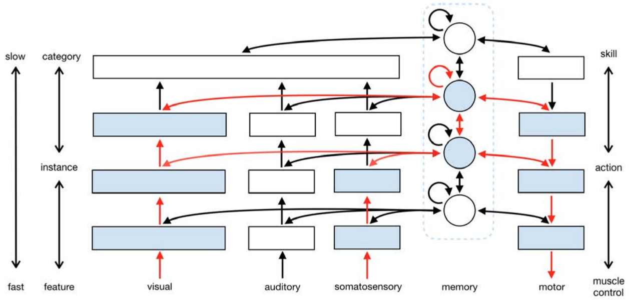

Figure 1. A modified conceptual diagram of layers and connections in the neocortex where the highlighted memory regions are added to the original diagram introduced in the book "On Intelligence" [4]. The filled layers and red connections are implemented in the proposed framework.

2.1. Memory model

In computer vision, there are two common types of object models used for identification. One represents objects in 2D and the other in 3D. However, neither of these incorporates information regarding how perceptions of objects change in response to actions. A robot that recognizes objects with traditional models knows nothing more than the label of the object. It is clear that humans have a different kind of object understanding—they can often predict the state and appearance of an object after an action.

Instead of an independent object recognition system, the proposed framework uses an integrated model called *aspect transition graph* (ATG) that fuses information acquired from sensors and robot actions to achieve better recognition and understanding of the environment. An ATG is a memory model that memorizes past experiences on how actions change *aspects*, observations stored in the model, and thus, maps observable states and actions to predicted future observable states.

An ATG is represented using a directed multigraph $G = (\mathcal{X}, \mathcal{U})$, composed of a set of aspect nodes $\mathcal{X}$ connected by a set of action edges $\mathcal{U}$ that capture the probabilistic transition between aspects. An action edge u is a triple (x_1, x_2, a) consisting of a source node x_1, a destination node x_2 and an action a that transitions between them. Note that there can be multiple action edges (associated with different actions) that transition between the same pair of nodes. Figure 2 shows an example of an ATG model of a cube.

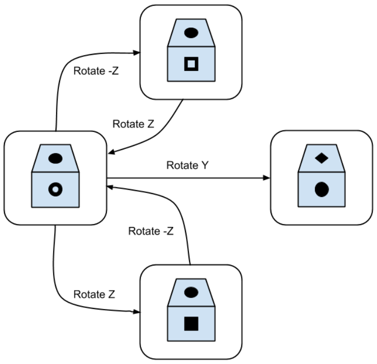

Figure 2. Example of an incomplete aspect transition graph (ATG) of a cube object that has a pattern on each face. Each aspect is consists of observations of two faces of the cube. Each edge represents an action that transitions between observations.

This memory model can be used to plan actions in partially observed environments. In previous work, we consider a simultaneous object modeling and recognition (SOMAR) task, where the robot has to model a given object while trying to recognize it [11]. An information theoretic planner that reduces uncertainty over objects by executing actions that reduces the expected entropy the most is proposed. The expected entropy is calculated based on the predicted action outcome stored in the ATG memory models. We showed that this approach outperforms a random action planner.

The ATG model is also shown to be able to handle uncertainties in stochastic environments in previous work [10]. Through fine-grained transitions, we show that errors can be detected early by comparing observation with the predicted action outcome. Transition probabilities is added to action edges in an ATG for actions that may result in random observations and errors can be handled accordingly. Surprising events that are not modeled in the memory are also handled by resetting the belief among aspects to the prior distribution; the robot would then re-examine the situation and identify possible solutions. We show that this approach results in more efficient actions and more robust results on a task that requires the robot to manipulate a box till it sees certain faces.

In [8], an ATG that considers a continuous observation space is introduced. Aspects are redefined as the set of observations within ϵ difference of a stored observation and the region of attraction is the set of observations that a closed-loop controller can converge to an aspect. Based on the funnel metaphor for closed-loop controllers introduced by Burridge [1], we introduce the slide metaphor for open-loop controllers that are used to represent action edges in an ATG model. A funnel may converge from a large set of robot states to a smaller subset, while a slide may end up in many different states due to noise. However, if a funnel-slide-funnel structure is constructed carefully such that the end of the slide is within the mouth of a funnel, we can guarantee a sequence of actions to succeed even when open-loop actions are included. This structure is tested on a tool grasping task where visual servoing is used to represent the funnel. We show that this structure reduces error significantly.

Traditional grasping approaches such as the willow garage grasping pipeline [15] usually separates action planning from object recognition, where actions are executed based on object poses and labels generated from the vision module. In [5], we propose an alternative grasping approach where the observation is matched to the most similar aspect in the ATG memory model; actions are then executed based on action edges connected from this aspect. This approach does not require an explicit object pose of the object and allows the robot to act directly based on observation. We tested on a drill grasping task based on memorized grasping examples.

2.2. Hierarchical structure

Neural networks with hierarchical structures, such as Convolutional Neural Networks (CNNs), have outperformed other approaches on many benchmarks in computer vision. However applying them to robotics is nontrivial for two reasons. First, the final output of a CNN contains little location information, which is essential for manipulation. Second, collecting the quantity of robot data required to train a CNN is quite difficult.

The proposed framework tackles these challenges using the hierarchical CNN feature introduced in our previous work [9]. Hierarchical CNN features are extracted from a CNN trained on image classification therefore only require a small set of action examples. Instead of representing a feature with a single filter in a certain CNN layer, hierarchical CNN features use a tuple of filter indices to represent a feature. These features capture the hierarchical relationship between filters in different layers and can represent local parts of an object such as the right edge of the lower right corner of a box's top face. Hierarchical CNN features can be localized by back propagating filter responses along a single path to the input image and then mapped to a 3D point in the point cloud. This process traces backward recursively and yields a tree structure of hierarchical CNN features.

We consider a grasping task where the goal is to posture an anthropomorphic hand and arm for grasping based on visual information. A dataset consists of 120 grasping examples of six cylindrical and six cuboid objects is collected. Each example consists of the image, input point cloud, and joint configuration of the pregrasp pose. To map hierarchical CNN features to grasp pose, features that fire consistently are first identified among objects of the same class (cuboids or cylinders.) Features that have low offset variances to end effectors (index finger, thumb, and hand) among examples are then selected. By restricting the selected hierarchical CNN features to have the same high level filter, features will all be associated with the same object. Figure 3 show that without considering the hierarchical relationship, low level filters will fire on different objects in a cluttered scenario.

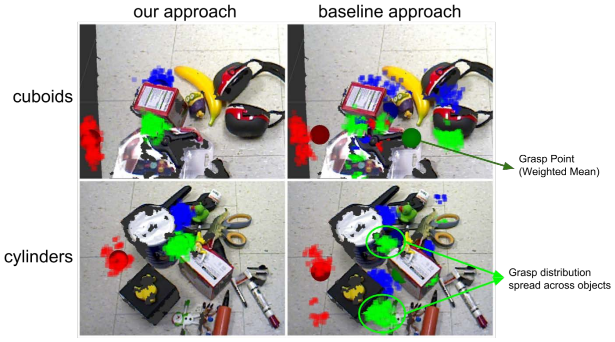

Figure 3. Comparison in a cluttered scenario. The red, green, and blue dots represent proposed grasp points for the hand frame, thumb tip, and index finger tip of the left robot hand. Notice that the colored dots are scattered around in the baseline approach since the highest response filter in conv-3 or conv-4 layer are no longer restricted to the same high level structure.

These selected hierarchical CNN features are then associated with a hierarchical controller that controls different kinematic subchains hierarchically. In this work, hierarchical CNN features in the fourth convolutional layer is associated with the arm controller and hierarchical CNN features in the third convolutional layer is associated with the hand controller. The intuition behind these relations is that when moving the arm, a rough location of the object is sufficient and the detail object information is only needed when placing fingers. We evaluated this approach on 50 grasping trials on 10 novel objects and show significant improvement over a point cloud based approach.

This hierarchical CNN feature is further combined with proprioceptive feedback and force feedback to form a hierarchical aspect representation in [7]. This aspect representation is used to represent the stored observation in an ATG model and can be used to model the appearance, pose, and location of an object and the force feedback that the robot have perceived. This aspect representation is evaluated on the Washington RGB-D Objects dataset [12] on instance pose recognition and achieved state of the art result.

2.3. Learning from demonstration

Learning from demonstration (LfD) is an attractive approach due to its similarity to how humans teach each other. However, most work on LfD has focused on learning the demonstrated motion [13], action constraints [14], and/or trajectory segments [3] [2] and has assumed that object poses can be identified correctly. This assumption may be true in industrial settings, but does not in general hold in unstructured environments.

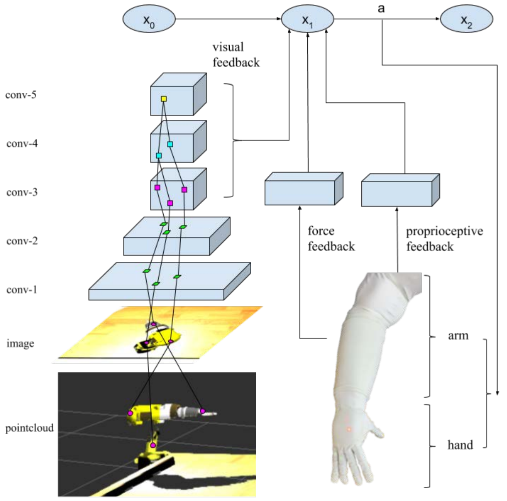

Figure 4. The sensorimotor architecture driving transitions in the ATG framework. The aspect representation stored in an aspect node x is based on visual, force, and proprioceptive feedback. These information is used to parameterize action a for controlling the arm and hand motors.

In previous work [6], we present an integrated approach that treats identifying informative features as part of the learning process. This gives robots the capacity to manipulate objects without fiducial markers and to learn actions focused on salient parts of the object. Instead of defining actions as relative movements with respect to the object pose, our actions are based on spatial relationships between features. Based on the demonstration type provided by the operator, informative features that support actions can be identified automatically. Figure 4 shows the overall architecture. Through learning from demonstration, the ATG memory model, hierarchical aspect representation, and connections to the hierarchical controller can be learned together efficiently.

This framework is demonstrated on a challenging bolt tightening task where the robot has to grasp the ratchet, tighten a bolt, and put the ratchet back into a tool holder with a small set of demonstrations. We show that the accuracy of mating the socket with the bolt can be increased with multiple examples. This learning from demonstrations approach is also tested on a drill grasping task in [7], where the goal is to grasp the drill on the handle with the robot's left hand. If the drill is out of reach, the robot has to plan a sequence of actions using both arms to extend its reachability based on grasping, rotating, and dragging actions learned from demonstrations. Figure 5 shows one of the trials that the robot executed both turning and dragging before grasping the drill.

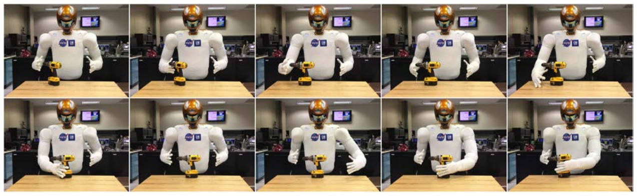

Figure 5. Sequence of actions in one grasping test trial. The images are ordered from left to right then top to bottom.

3. Conclusions

The goal of this work is to present a framework that allows robots to solve tasks in an unstructured environment through predicting perceptual action consequences based on memory and observation. In this paper, we provide an overview of a series of work that explores parts of this framework. By predicting perceptual action consequences based on memory and perception, the proposed framework can accomplish a variety of challenging tasks under a unified framework. These results can be seen as support to the conjectured connections between sensory neurons, motor neurons, and memory regions in the proposed neocortex model shown in Figure 1.

4. Acknowledgment

We are thankful to our colleagues and former members of the Laboratory for Perceptual Robotics for their contribution on the control basis framework code base. We are also grateful to the Robonaut Team for their support on Robonaut-2. This material is based upon work supported under Grant NASA-GCT-NNX12AR16A and a NASA Space Technology Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Aeronautics and Space Administration.

References

[1] R. R. Burridge, A. A. Rizzi, and D. E. Koditschek. Sequential composition of dynamically dexterous robot behaviors. *The International Journal of Robotics Research*, 18(6):534–555, 1999.

[2] S. Calinon and A. Billard. A probabilistic programming by demonstration framework handling constraints in joint space and task space. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 367–372. IEEE, 2008.

[3] S. Calinon, F. Guenter, and A. Billard. On learning, representing, and generalizing a task in a humanoid robot. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 37(2):286–298, 2007.

[4] J. Hawkins and S. Blakeslee. *On intelligence*. Macmillan, 2007.

[5] L. Y. Ku, M. Hebert, E. Learned-Miller, and R. Grupen. Object manipulation based on memory and observation. In *First Workshop on Object Understanding for Interaction, at the International Conference on Computer Vision*, 2015.

[6] L. Y. Ku, S. Jordan, J. Badger, E. G. Learned-Miller, and R. A. Grupen. Learning to use a ratchet by modeling spatial relations in demonstrations. *arXiv preprint*, 2018.

[7] L. Y. Ku, E. Learned-Miller, and R. Grupen. An aspect representation for object manipulation based on convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 794–800. IEEE, 2017.

[8] L. Y. Ku, E. G. Learned-Miller, and R. A. Grupen. Modeling objects as aspect transition graphs to support manipulation. *International Symposium on Robotics Research*, 2015.

[9] L. Y. Ku, E. G. Learned-Miller, and R. A. Grupen. Associating grasp configurations with hierarchical features in convolutional neural networks. In *Intelligent Robots and Systems (IROS), 2017 IEEE International Conference on*. IEEE, 2017.

[10] L. Y. Ku, D. Ruiken, E. Learned-Miller, and R. Grupen. Error detection and surprise in stochastic robot actions. In *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*, pages 1096–1101. IEEE, 2015.

[11] L. Y. Ku, S. Sen, E. G. Learned-Miller, and R. A. Grupen. Action-based models for belief-space planning. *Workshop on Information-Based Grasp and Manipulation Planning, at Robotics: Science and Systems*, 2014.

[12] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 1817–1824. IEEE, 2011.

[13] P. Pastor, H. Hoffmann, T. Asfour, and S. Schaal. Learning and generalization of motor skills by learning from demonstration. In *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, pages 763–768. IEEE, 2009.

[14] C. Pérez-D'Arpino and J. A. Shah. C-learn: Learning geometric constraints from demonstrations for multi-step manipulation in shared autonomy. In *IEEE International Conference on Robotics and Automation*, 2017.

[15] M. Wise and M. Ciocarlie. ICRA Manipulation Demo, 2010. [Online; accessed 19-September-2015].