An Aspect Representation for Object Manipulation Based on Convolutional Neural Networks

Li Yang Ku, Erik Learned-Miller and Rod Grupen

***Abstract*— We propose an intelligent visuomotor system that interacts with the environment and memorizes the consequences of actions. As more memories are recorded and more interactions are observed, the agent becomes more capable of predicting the consequences of actions and is, thus, better at planning sequences of actions to solve tasks. In previous work, we introduced the aspect transition graph (ATG) which represents how actions lead from one observation to another using a directed multi-graph. In this work, we propose a novel aspect representation based on hierarchical CNN features, learned with convolutional neural networks, that supports manipulation and captures the essential affordances of an object based on RGB-D images. In a traditional planning system, robots are given a pre-defined set of actions that take the robot from one symbolic state to another. However symbolic states often lack the flexibility to generalize across similar situations. Our proposed representation is grounded in the robot's observations and lies in a continuous space that allows the robot to handle similar unseen situations. The hierarchical CNN features within a representation also allow the robot to act precisely with respect to the spatial location of individual features. We evaluate the robustness of this representation using the Washington RGB-D Objects Dataset and show that it achieves state of the art results for instance pose estimation. We then test this representation in conjunction with an ATG on a drill grasping task on Robonaut-2. We show that given grasp, drag, and turn demonstrations on the drill, the robot is capable of planning sequences of learned actions to compensate for reachability constraints.**

I. INTRODUCTION

In this work, we consider an agent that interacts with the environment and memorizes the consequences of actions. As more memories are recorded and more interactions are observed, the agent becomes more capable of predicting the consequences of actions and better at planning sequences of actions to solve tasks. In previous work [1], we introduced the aspect transition graph (ATG), which represents how actions lead to new observations using a directed multi-graph consisting of aspect nodes and action edges. An aspect is defined as an observation that is memorized by the robot. This concept is inspired by experiments done in the field of human psychophysics and neurophysiology, which suggest that humans memorize a set of canonical views of an object instead of maintaining a single object-centered model [2] [3]. We demonstrated that with an ATG model, the robot is capable of executing a sequence of actions to solve tasks in a simple environment consisting of boxes with fiducial markers. In this work, we propose a novel aspect representation that supports manipulation in more complicated scenarios and captures the essential affordances of an object based on RGB-D images.

Li Yang Ku, Erik Learned-Miller and Rod Grupen are with the College of Information and Computer Sciences, University of Massachusetts Amherst, Amherst, MA 01003, USA {lku,elm,grupen}@cs.umass.edu

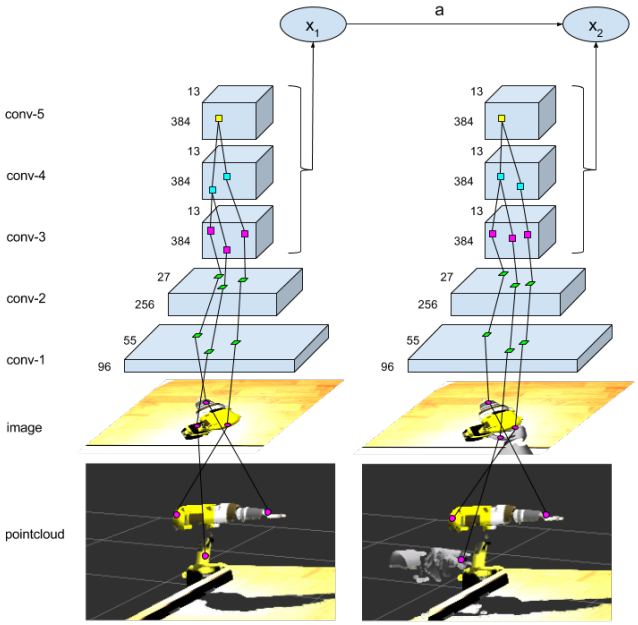

Fig. 1. Architecture of two aspect nodes within an ATG. Each aspect node x stores a set of hierarchical CNN features that captures the relationship between filters in the conv-3, conv-4, and conv-5 layer of a pre-trained CNN. This set of hierarchical CNN features are mapped to the point cloud through backpropagation and supports action a as reference points for relative movements.

The hierarchical CNN features applied in this work are extracted from middle layer filters rather than the final layer of a CNN. The positions of these features can be located by backpropagating filter responses to the point cloud (See Section III-A). Identifying 3D feature positions allows us to plan precise actions relative to these local features and adjust to small variations. Based on these feature locations a pose descriptor that specifies the relative positions between these features and a location descriptor that models the distance between features and robot frames are created. These descriptors allow us to distinguish observations generated by different poses of an object.

In a traditional planning system, robots are given a predefined set of actions that take the robot from one symbolic state to another. However symbolic states often lack the flexibility to generalize across similar situations. Our proposed representation is grounded in the robot's observations and lies in a continuous space that allows the robot to handle

similar but new situations.

We test the robustness of this aspect representation using the Washington RGB-D Objects Dataset and show that it achieves state of the art results for instance pose estimation. We further incorporate this representation with an ATG model and test on a drill grasping task on Robonaut-2 [4]. Figure 1 shows part of this integrated architecture. We show that given a small set of grasp, drag, and turn demonstrations on the drill through teleoperation, the robot is capable of planning learned actions to extend its reach in grasping and manipulation tasks.

II. RELATED WORK

In this section, we first compare with related work in object pose estimation based on RGB-D images and approaches that use features extracted from layers beyond a CNN's final layer. We then discuss CNN applications in robotics and previous work in learning from demonstration. Last, we talk about research that have shown to extend the robot's reach through planning and our previous work with ATGs.

Because of the increased availability of inexpensive RGB-D sensors, a great deal of research has been done to identify the pose of an object using outputs from these sensors. In [5], gradient and shape kernel descriptors are extracted from RGB and depth images for an object pose tree to determine the object pose. In [6], sparce coding is used to learn hierarchical feature representations for intensity, RGB, depth, and 3D surface normals. In [7], depth images are colorized based on the depth and fed into a CNN trained on RGB images; the pose is then determined based on CNN features from both the RGB image and the colorized depth image. These approaches all treat the depth image as additional information that is concatenated to the RGB image. In this work, we consider a more integrated approach where features extracted from the RGB image are mapped to a location using the point cloud generated from the depth image. A descriptor of the relative positions of RGB features is then generated.

Many authors have explored using filter activations other than the final layer of a CNN. In [8], hypercolumns, which are defined as the activation of all CNN units above a pixel, are used on tasks such as simultaneous detection and segmentation, keypoint localization, and part labeling. The hierarchical CNN features used in this work group CNN filters in different layers based on their hierarchical support relations instead of spatial relationship. In [7], the last two layers of two CNNs, one that takes an image as input and one that takes depth as input, are used to identify object category, instance, and pose. In our work, we locate mid-layer CNN filters through backpropagation and determine the object pose based on these feature positions.

Several authors have applied CNNs to robotics. In [9], visuomotor policies are learned using an end-to-end neural network that takes images and outputs joint torques. A three layer CNN is used without any max pooling layer to maintain spatial information. In [10], an autoencoder is used to learn spatial information of features of a neural network. Our approach finds the receptive field that causes a high-level response in a particular image through backpropagation. In [11], a CNN is used to learn what features are graspable through 50 thousand trials collected using a Baxter robot. The final layer is used to select 1 out of 18 grasp orientations. Although end-to-end learning is an elegant approach, it is often not practical to collect the same amount of training data for grasping that is required by successful CNNs for computer vision. Instead of training CNNs that output robot actions directly based on image inputs, our approach learns the relative position between features and end point goals for arms and fingers, which makes it possible to learn complicated end effector poses with very few demonstrations. This work uses a CNN model similar to Alexnet [12] introduced by Yosinski [13] that is implemented in Caffe [14] and trained on ImageNet [15].

A great deal of work has been done on learning from demonstration. In [16], human demonstrations are segmented and interpreted as pre-defined symbolic subtasks under macro operators and mapped to primitive robot actions on a jar opening task with two arms. In [17], relevant features based on joints, end effector positions, and relative positions to the object are extracted from demonstrations and used for generalization. In [18], the task is focused on extracting low-level motor primitives to encode the demonstrated trajectory. In [19], key frames of a trajectory are extracted from demonstration and connected with spline during testing. Our approach focuses on learning movements with respect to visual features and considers how actions change observations.

In [20] and [21], it is also shown that a robot's reach can be extended by planning non-prehensile actions. In [20], a set of primitive actions and their effects are defined for a sampling based planning algorithm. In [21] and [22], the set of actions are learned through multiple stages based on pre-defined rewards. In this work, the actions and their effects are learned directly from demonstrations.

The aspect transition graph (ATG) used in this work was first introduced in Sen's work [23]. In [24], a mechanism for learning these models without supervision and a method for applying belief space planning on it is introduced. In [25] and [26], the robustness of the ATG is further improved with visual servoing and the ability to handle stochastic actions.

III. REPRESENTATION

This paper introduces a novel aspect representation based on hierarchical CNN features that supports manipulation. A primary step in this system is to match the current observation to an aspect, which we defined to be a stored observation in memory, so that the robot can apply learned actions to the current situation. We represent an aspect as a set of hierarchical CNN features, an appearance descriptor, a pose descriptor, and a location descriptor. This representation is used to identify the aspect that affords the same type of interactions given the current observation and allows the robot to manipulate the object based on the feature locations when combined with an ATG. The following sections discuss how hierarchical CNN features are used to define this

representation.

A. Hierarchical CNN Features

Although CNNs have outperformed other approaches on object recognition benchmarks, identifying the category of an object is not enough for manipulation. Knowing visual feature locations in 3D is also crucial for interacting with it precisely. The hierarchical CNN features introduced in our previous work [27] exploit the fact that CNNs are by nature hierarchical; a filter in a higher layer with little location information is a combination of lower level features with higher spatial accuracy. 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 such as (f_i^5, f_j^4, f_k^3), where f_i^5, f_j^4, and f_k^3 represent the i^{th} filter in the 5^{th} convolutional layer, the j^{th} filter in the 4^{th} convolutional layer, and the k^{th} filter in the 3^{rd} convolutional layer respectively. The responses of a filter f_j^m are restricted to the portion that contributes to the filter f_i^{m+1}.

The key observation in our work is the following. A lower layer CNN filter often represents a local structure of a more complicated structure represented by a higher layer filter. The hierarchical CNN feature identifies local structures at each level that are related to a hierarchical "part-based" representation of an object that each afford opportunities for control and interaction.

For each observed point cloud, we segment the flat supporting surface using Random Sample Consensus (RANSAC) [28] and create a 2D mask that excludes pixels that are not part of the object in the RGB image. This mask is dilated to preserve the boundary of the object. During the forward pass, for each convolution layer we check each filter and zero out the responses that are not marked as part of the object according to the mask. This approach removes filter responses that are not part of the object. The masked forward pass approach is used instead of directly segmenting the object in the RGB image to avoid sharp edges caused by segmentation.

During the backward pass, the top N^j filters in the j^{th} convolutional (conv-j) layer that have the highest sum of log responses are identified. For each of these filters we find the maximum response of the activation map, zero out all other responses and backpropagate this max unit reponse to the conv-$(j-1)$ layer. We then find the top N^{j-1} filters that have the highest sum of log partial derivatives obtained from backpropagation. For each of these N^{j-1} filters, we take the gradient of backpropagation and zero out everything except for the maximum partial derivative. The same procedure is used recursively to find N^{j-2} filters in the conv-$(j-2)$ layer. This process back traces along a single path recursively and yields a tree structure of hierarchical CNN features. In our drill-grasping experiment we pick $N^5 = N^4 = N^3 = 3$. After identifying a set of hierarchical CNN features, we further locate a feature in 3D by backpropagating to the image and mapping the mean of the response location to the corresponding 3D point in the point cloud. We define the response of a hierarchical CNN feature as the maximum partial derivative of the lowest layer filter in the feature tuple. This response is proportional to the amount this lower layer filter contributes to the higher layer filter activation in this hierarchical CNN feature.

B. Descriptors

Based on the response and 3D location of the hierarchical CNN features extracted, our approach generates an appearance descriptor r, a pose descriptor q, and a location descriptor l for each aspect. The appearance descriptor r is a set of hierarchical CNN feature responses based on the feature tuple. Our assumption is that aspects similar to the current observation have similar appearances and therefore similar hierarchical CNN features and responses.

The pose descriptor q is a set of relative 3D positions in the camera frame between each pair of hierarchical CNN features. If H is the set of all possible hierarchical CNN features and r contains $|h|$ responses of a subset $h \subset H$ of hierarchical CNN features, then q contains $|h| \times |h-1|/2$ XYZ differences. Assuming that aspects similar to the observation should have similar poses, we use the relative location of the features to further distinguish aspects that have the same features but are oriented differently.

The location descriptor l is a set of distances from the centroid of the hierarchical CNN features to a set of predefined robot frames. In this work, the robot shoulder and palm frames are used. The location descriptor captures the object position and distinguishes aspects that are reachable.

Our goal is to find the aspect x in memory most likely to match the current observation z based on the descriptors. Let $p(x|z)$ be the probability that aspect x is generated from the same state that generates observation z. We can then calculate this probability through Bayes' rule $p(x|z) \propto p(z|x) \cdot p(x)$, which the likelihood is modeled as

$$p(z|x) = p(r^z|r^x) \cdot p(q^z|q^x) \cdot p(l^z|l^x). \tag{1}$$

Here r^z and r^x are the appearance descriptors of the observation z and the aspect x, q^z and q^x are the pose descriptor of z and x, and l^z and l^x are the location descriptor of z and x.

We model $p(r^z|r^x)$ as the geometric mean of the probabilities $p(r_n^z|r_n^x)$ of individual appearance descriptor values r_n^z and r_n^x. The probability $p(r_n^z|r_n^x)$ is modeled as a Generalized Gaussian Distribution (GGD) of the value difference between r_n^z and r_n^x scaled by their sum.

$$p(r^z|r^x) = \Big(\prod_{r_n^x \in r^x \vee r_n^z \in r^z} p(r_n^z|r_n^x) \Big)^{\frac{1}{N}}, \tag{2}$$

$$p(r_n^z|r_n^x) = GGD(r_n^x - r_n^z; \alpha = r_n^x + r_n^z), \tag{3}$$

where N is the number of appearance descriptors. A missing descriptor value $r_n \notin r$ is considered to be zero. Different individual appearance descriptors r_n refer to different hierarchical CNN feature tuples. The GGD function is defined as

$$GGD(y; \mu, \alpha, \beta) = \frac{1}{Z(\alpha, \beta)} \cdot e^{-(\frac{|y-\mu|}{\alpha})^\beta}, \tag{4}$$

where μ is the mean parameter, α is the scale parameter, β is the shape parameter, and $Z(\alpha, \beta)$ is the partition function. μ is set to zero and β is set to 0.1 in this work. We found that a shape parameter β that produces a heavier tail performs better than a standard Gaussian on the Washington RGB-D Objects dataset.

The pose descriptor likelihood $p(q^z|q^x)$ is modeled similarly, with

$$p(q^z|q^x) = \Big(\prod_{q_n^x \in q^x \wedge q_n^z \in q^z} p(q_n^z|q_n^x) \Big)^{\frac{1}{N}}, \tag{5}$$

$$p(q_n^z|q_n^x) = GGD(q_n^x - q_n^z; \alpha = C_q), \tag{6}$$

where q_n^x and q_n^z are the individual pose descriptor values, and N is the number of pose descriptors. Different individual pose descriptors refer to different pairs of hierarchical CNN features and XYZ coordinates in the camera frame. C_q is a constant we set to 0.1.

The location descriptor likelihood $p(l^z|l^x)$ is also modeled as

$$p(l^z|l^x) = \Big(\prod_{l_n^x \in l^x \wedge l_n^z \in l^z} p(l_n^z|l_n^x) \Big)^{\frac{1}{N}}, \tag{7}$$

$$p(l_n^z|l_n^x) = GGD(l_n^x - l_n^z; \alpha = C_l), \tag{8}$$

where l_n^x and l_n^z are the individual location descriptor values, and N is the number of location descriptors. Different individual location descriptors refer to different robot frames. C_l is a constant set equal to C_q.

C. Aspect Transition Graph

Next we briefly review the formal definition of an ATG from our prior work. An aspect transition graph (ATG) object model in our work 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 aspect nodes. We define an aspect as an observation that is stored in the model. 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. In this work, the aspect representation is used to denote aspect nodes and action edges store the relative positions between robot frames and the hierarchical CNN features in aspect x_1. An ATG models how an action that acts relative to observed features would change the observation. The hierarchical CNN features stored in an aspect node may include features that are not on the object. For example, when the robot places its fingers on top of an object, a feature that is generated by both the fingers and the object may be observed.

IV. EXPERIMENTAL RESULTS

We perform two sets of experiments. First, the proposed aspect representation is evaluated on an object pose estimation dataset. The major objective of this first set of experiments is to obtain a quantitative measurement on the ability to identify observations with similar affordances. Second, the aspect representation integrated with ATG is tested on a drill grasping task. We show that given a set of grasp, drag, and turn examples, Robonaut-2 is capable of extending its reach by planning a sequence of learned actions. This demonstration is the first that extends our previous work to a real world scenario and shows the robot's ability to plan using actions learned directly from observations.

A. Pose Estimation

The goal of the proposed aspect representation is to identify an observation in memory that is similar to the current observation and, thus, affords similar actions. Therefore, we test the proposed representation on instance pose recognition on the Washington RGB-D Objects dataset [29] under the assumption that an object's pose and affordance are strongly correlated; the same object usually supports the same set of actions when placed in similar orientations. We show that our approach achieves state of the art results in accuracy.

1) Experimental Settings: The Washington RGB-D dataset contains RGB images, depth images, point clouds, and masks for 300 objects. Each object is placed on a turntable and approximately 250 frames are captured for each elevation angle $(30°, 45°, 60°)$. Every 5th frame is labeled with the approximated turntable angle. The $30°$ and $60°$ frames are used for training and the $45°$ frames are used for testing. The goal of the instance pose estimation task is to identify the turntable angles of frames taken at $45°$ elevation angle. We did not experiment on category pose estimation since we are mostly interested in identifying the aspect of a specific object instance in this work. We preprocessed the depth images to fill in empty values with the values of the closest pixels and generated point clouds based on the processed depth images.

2) Approach: During testing, we treat the frames in the training set as aspects x and the test frame as observations z. The prerecorded frame in memory that has the closest turntable angle to the current observation should also afford the most similar set of actions. Hence, the angle of the frame in the training set that has the maximum posterior probability $p(x|z)$ given the test frame is chosen as the estimated angle. For each frame that is labeled with the turntable angle, we extract the hierarchical CNN features and generate the appearance and pose descriptor. We set $N^5 = 30$ and $N^4 = 5$ as the number of extracted hierarchical CNN features in the conv-5 and conv-4 layer. We do not include conv-3 layer features to reduce the test time. The location descriptor is not used in this experiment since there is no need to identify the object location with respect to the robot.

3) Results: Since the distribution of angle differences are skewed across objects, both the average error and the median error are used for evaluation. We compare against three other reported approaches, (a) object pose tree with kernel descriptors [5], (b) hierarchical matching pursuit [6], and (c) pre-trained CNN with RGB and depth image [7]. Our approach achieves a $38.1°$ average pose error and a $16.3°$ median pose error; both of these numbers achieve state of the art results as shown in Table I. Note that while our experiment evaluates every test frame in the test set, other

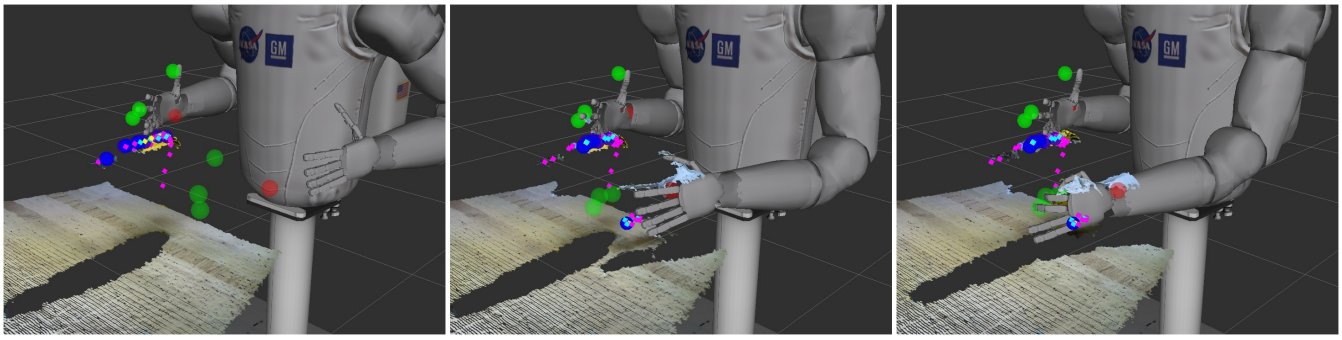

Fig. 2. Relative movements based on hierarchical CNN features for grasping a drill. The figures from left to right represent a sequence of actions moving relative to a set of hierarchical CNN features. The yellow, cyan, and magenta dots represent the detected hierarchical CNN features in the conv-5, conv-4, and conv-3 layers respectively. Most of the detected features are generated by the top of the drill, while in the second and third figure, some features are generated by the visible part of the left hand. The red and green spheres represent the target positions for the robot palms and fingers. A target position is calculated based on the mean of a set of feature positions plus the corresponding offsets. The blue spheres highlight the set of hierarchical CNN features that the target position for the index finger is associated to.

works only evaluate on frames that are correctly classified as the correct instance. Many of the errors are due to objects in the dataset that have similar appearance across multiple viewpoints. We argue that these objects often support the same set of actions when they are oriented at visually similar poses and only need to be represented by one aspect. For example, the orientation of an orange is not important to the robot as long as the robot learned to manipulate it from one similar observation.

Work	Angular Error (°)	
	MedPose(I)	AvgPose(I)
OPTree, Lai *et al.* [5]	30.2	57.1
HMP, Bo *et al.* [6]	18.0	44.8
CNN: RGB-D, Schwarz *et al.* [7]	18.7	42.8
Hierarchical CNN feature (Our approach)	**16.3**	**38.1**

TABLE I
MEDIAN AND AVERAGE INSTANCE POSE ESTIMATION ERROR ON WASHINGTON RGB-D OBJECTS DATASET.

B. Drill Manipulation

We test the proposed aspect representation with ATG on a drill grasping task on Robonaut-2. We demonstrated that with the proposed aspect representation the current observation can be associated with the aspect that supports the same set of actions. Through a few demonstrated manipulations on a drill, the robot is able to grasp the drill, in a position that is normally out of reach by combining learned actions in sequence.

1) Experimental Settings: The goal of the task is to grasp the drill handle correctly with the left robot hand based on 8 demonstrated manipulation action sequences on a drill on Robonaut-2 through teleoperation. Three of the demonstrated action sequences are grasp action sequences, where we place the drill at three different orientations and teleoperate the robot to hold the drill handle with its left hand. Four of the demonstrated actions are drag action sequences, where we place the drill at the right side of the robot and teleoperate the right hand to drag the drill from the right side to the center. The other demonstrated action sequence is a turn action, where we place the drill such that the tip of the drill is facing toward the right side of the robot and teleoperate the right hand to turn the drill such that the tip of the drill is facing away from the robot.

For each test trial, the drill is placed on the table in front of the robot. The robot can manipulate the drill until it successfully grasps it on the handle. For example, if the drill is placed on the right side of the table and not reachable with the left hand, the robot can drag the drill closer with its right hand and grasp it with its left hand. If the drill ends up in a pose that is no longer graspable or if the robot tries to grasp and fails, we consider the trial a failure. The experiment is evaluated based on the number of successful grasps where the robot fingers surround the handle such that the tip of the drill is facing outward from the wrist.

2) Learning From Demonstration: An ATG is generated for each of the 8 demonstrations by indicating the start and end of the actions executed during teleoperation; an aspect node that stores the proposed aspect representation is created between each action. For example, one demonstrated turn example is a three action sequence of moving the hand to a pre-turn pose, pushing the drill tip to turn the drill, and moving the hand back. We define an action as a relative movement to a set of features. In this experiment, we pick the top three filters with the highest responses in the conv-5, conv-4, and conv-3 layers ($N^5 = N^4 = N^3 = 3$) for extracting the hierarchical CNN features. Each aspect representation is composed of an appearance descriptor with 39 hierarchical CNN feature response values, a pose descriptor with 741 XYZ differences between these 39 features, and a location descriptor with 4 distance values from the centroid of these features to the robot's palms and shoulders. An action edge that connects from aspect x_t to x_{t+1} in an ATG stores an action that is configured by the position offset between the robot end effectors and the set of corresponding hierarchical CNN features in aspect x_t. The top K features in x_t that the robot end effectors are closest to after executing the action is chosen. We set $K = 5$ in this test.

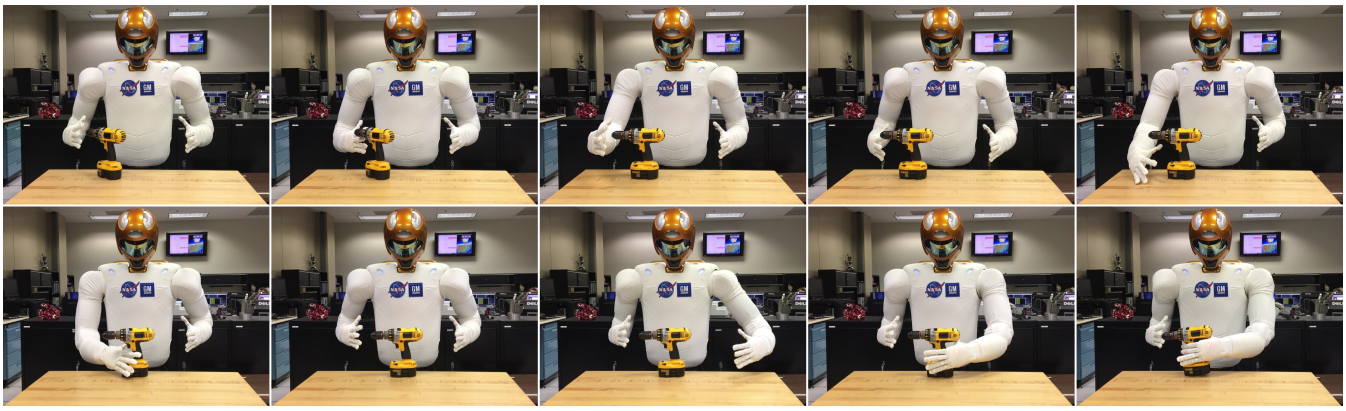

Fig. 3. Sequence of actions in one grasping test trial. The images are ordered from left to right then top to bottom. The initial pose of the drill is at an angle that is not graspable and is located too far right for the left hand to reach. Therefore the robot turns the drill then drags it to the center before grasping with its left hand.

In this experiment, the arm controllers are associated with the conv-4 layer hierarchical CNN features while the hand controllers are associated with the conv-3 layer hierarchical CNN features. We manually define this hierarchical correspondence where higher level features are associated with higher level actions that require less accuracy. We did not compare with different mappings in this experiment but we believe this design allows the robot to plan more efficiently in different levels of abstraction based on the task requirement; an action that does not require high precision would only need a rough location reference. Figure 2 shows a sequence of relative movements generated while grasping the drill.

The 8 ATGs created from demonstration are combined into one ATG that represents all the manipulations the robot memorized for different aspects of the drill. Both the drag and turn action sequences conclude in a state where the drill is on the table with no contact with the robot; since the orientation and location of the drill is uncertain after these actions we connect the last aspect node of the ATGs corresponding to these demonstrations to an intermediate node, and connect this node to the first aspect of all of the 8 ATGs created. The three grasp action sequences end up in a state where the drill is grasped correctly in the robot hand; since our task is to reach such state, we connect the last aspect nodes of the ATGs corresponding to these demonstrations to an aspect that indicates that the drill is grasped. The final ATG contains 31 aspect nodes and 32 action edges.

3) Approach: For each trial, we first identify the aspect node x in the combined ATG that has the highest posterior probability $p(x|z)$ given the current observation z. The prior probability $p(x)$ is set to be uniform among aspect nodes that are the first aspect of each demonstration. The maximum a posteriori (MAP) aspect node $p(x|z)$ can therefore be determined by calculating $p(z|x) \cdot p(x)$. The next action is then chosen based on the first action edge on the shortest path from the MAP aspect node to the goal aspect node.

For each action, the target positions for the palms are determined by the mean position of a set of conv-4 layer features plus the corresponding offsets. The same approach is used to determine target positions for the fingers and thumbs from a set of conv-3 layer features plus their respective offsets.

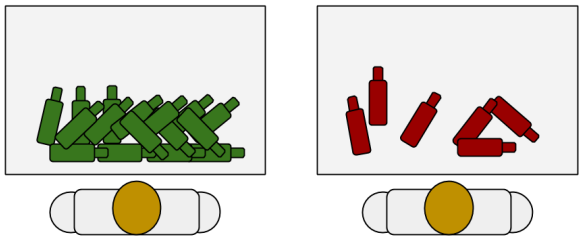

Fig. 4. Initial drill poses that the robot succeeded and failed in grasping with its left hand during testing. The green drill poses in the left figure shows the succeeded poses and the red drill poses in the right figure shows the failed poses. Our approach allows the robot to grasp drills located at position that is normally out of reach.

Each action is executed in two steps. First, the arm controllers move the arms such that the distance from the palms to their corresponding targets are minimized. Once the arm controllers converge, the hand controllers move the wrists and fingers to minimize the sum of distance from the index finger tip, middle finger tip, and thumb tip to their corresponding target. These controllers are based on the control basis framework [30] and can be written in the form $\phi|_{\tau}^{\sigma}$, where ϕ is a potential function that describes the sum of distance to the targets, σ represents sensory resources allocated, and τ represents the motor resources allocated. The posterior probability $p(x|z)$ is recursively updated based on the Bayesian filtering algorithm [31] after each action and observation. The next action is always chosen based on the MAP aspect node after the update.

4) Experimental Results: We performed 22 grasping trials in this experiment and our approach successfully grasped the drill on the handle 16 times. Among 11 of the successful trials, the robot turned or dragged the drill with its right hand before grasping it with its left hand. Figure 3 shows one of the trials that the robot executed both turning and dragging

before grasping the drill. The initial poses of the drill that the robot succeeded or failed in grasping are shown in Figure 4. Three of the failed trials are due to the robot trying to grasp the drill while the drill is placed at a pose almost within reach. We believe that adding more demonstrations or the ability to recover from error would improve the performance on this task. Calculating the aspect representation takes about 3 seconds on a desktop computer with the NVIDIA GTX 780 graphics card. Matching the current aspect to a memorized aspect has a complexity of $O(n)$, where n is the number of memorized aspects; this matching process takes less then a second in this experiment.

V. CONCLUSION

In this work, we introduce an aspect representation based on hierarchical CNN features that supports manipulation and allows generalization to similar observations. We evaluate the robustness of this representation using the Washington RGB-D Objects dataset for instance pose estimation. We further combine this representation with the ATG model and experiment on a drill grasping task on Robonaut-2, where the goal is to grasp the drill placed in multiple poses based on a small set of grasp, drag, and turn actions demonstrated to the robot. This demonstration is a significant extension of our previous work to a real world environment and shows the robot's ability to plan a sequence of actions learned directly from observations.

VI. ACKNOWLEDGMENT

We are thankful to our colleagues Dirk Ruiken, Mitchell Hebert, Michael Lanighan, and members of the Laboratory for Perceptual Robotics for their contribution on the control basis framework. We are also grateful to Julia Badger, Will Baker, and 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] L. Y. Ku, S. Sen, E. G. Learned-Miller, and R. A. Grupen, "Aspect transition graph: an affordance-based model," *Second Workshop on Affordances: Visual Perception of Affordances and Functional Visual Primitives for Scene Analysis, at the European Conference on Computer Vision*, 2014.

[2] S. Edelman and H. H. Bülthoff, "Orientation dependence in the recognition of familiar and novel views of three-dimensional objects," *Vision research*, vol. 32, no. 12, pp. 2385–2400, 1992.

[3] H. H. Bülthoff and S. Edelman, "Psychophysical support for a two-dimensional view interpolation theory of object recognition," *Proceedings of the National Academy of Sciences*, vol. 89, no. 1, pp. 60–64, 1992.

[4] M. A. Diftler, J. Mehling, M. E. Abdallah, N. A. Radford, L. B. Bridgwater, A. M. Sanders, R. S. Askew, D. M. Linn, J. D. Yamokoski, F. Permenter *et al.*, "Robonaut 2-the first humanoid robot in space," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*. IEEE, 2011, pp. 2178–2183.

[5] K. Lai, L. Bo, X. Ren, and D. Fox, "A scalable tree-based approach for joint object and pose recognition." in *AAAI*, 2011.

[6] L. Bo, X. Ren, and D. Fox, "Unsupervised feature learning for rgb-d based object recognition," in *Experimental Robotics*. Springer, 2013, pp. 387–402.

[7] M. Schwarz, H. Schulz, and S. Behnke, "Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1329–1335.

[8] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 447–456.

[9] S. Levine, C. Finn, T. Darrell, and P. Abbeel, "End-to-end training of deep visuomotor policies," *arXiv preprint arXiv:1504.00702*, 2015.

[10] C. Finn, X. Y. Tan, Y. Duan, T. Darrell, S. Levine, and P. Abbeel, "Deep spatial autoencoders for visuomotor learning," *reconstruction*, vol. 117, no. 117, p. 240, 2015.

[11] L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50k tries and 700 robot hours," *arXiv preprint arXiv:1509.06825*, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[16] R. Zöllner, T. Asfour, and R. Dillmann, "Programming by demonstration: dual-arm manipulation tasks for humanoid robots." in *IROS*, 2004, pp. 479–484.

[17] 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)*, vol. 37, no. 2, pp. 286–298, 2007.

[18] A. J. Ijspeert, J. Nakanishi, and S. Schaal, "Learning attractor landscapes for learning motor primitives," Tech. Rep., 2002.

[19] B. Akgun, M. Cakmak, K. Jiang, and A. L. Thomaz, "Keyframe-based learning from demonstration," *International Journal of Social Robotics*, vol. 4, no. 4, pp. 343–355, 2012.

[20] J. Barry, K. Hsiao, L. P. Kaelbling, and T. Lozano-Pérez, "Manipulation with multiple action types," in *Experimental Robotics*. Springer, 2013, pp. 531–545.

[21] S. Sen, G. Sherrick, D. Ruiken, and R. Grupen, "Choosing informative actions for manipulation tasks," in *Humanoid Robots (Humanoids), 2011 11th IEEE-RAS International Conference on*. IEEE, 2011, pp. 721–726.

[22] S. Hart, S. Sen, and R. A. Grupen, "Intrinsically motivated hierarchical manipulation." in *ICRA*, 2008, pp. 3814–3819.

[23] S. Sen, "Bridging the gap between autonomous skill learning and task-specific planning," Ph.D. dissertation, University of Massachusetts Amherst, 2013.

[24] 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.

[25] 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.

[26] 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*. IEEE, 2015, pp. 1096–1101.

[27] L. Y. Ku, E. G. Learned-Miller, and R. A. Grupen, "Associating grasp configurations with hierarchical features in convolutional neural networks," *arXiv preprint arXiv:1609.03947*, 2016.

[28] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[29] 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*. IEEE, 2011, pp. 1817–1824.

[30] M. Huber, "A hybrid architecture for adaptive robot control," Ph.D. dissertation, University of Massachusetts Amherst, 2000.

[31] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.