Associating Grasp Configurations with Hierarchical Features in Convolutional Neural Networks

Li Yang Ku, Erik Learned-Miller, and Rod Grupen

***Abstract*— In this work, we provide a solution for posturing the anthropomorphic Robonaut-2 hand and arm for grasping based on visual information. A mapping from visual features extracted from a convolutional neural network (CNN) to grasp points is learned. We demonstrate that a CNN pre-trained for image classification can be applied to a grasping task based on a small set of grasping examples. Our approach takes advantage of the hierarchical nature of the CNN by identifying features that capture the hierarchical support relations between filters in different CNN layers and locating their 3D positions by tracing activations backwards in the CNN. When this backward trace terminates in the RGB-D image, important manipulable structures are thereby localized. These features that reside in different layers of the CNN are then associated with controllers that engage different kinematic subchains in the hand/arm system for grasping. A grasping dataset is collected using demonstrated hand/object relationships for Robonaut-2 to evaluate the proposed approach in terms of the precision of the resulting preshape postures. We demonstrate that this approach outperforms baseline approaches in cluttered scenarios on the grasping dataset and a point cloud based approach on a grasping task using Robonaut-2.**

I. INTRODUCTION

In this work, we provide a solution for posturing Robonaut-2's anthropomorphic hand and arm for grasping based on visual information. Experiments conducted by Goodale and Milner [1] have shown that humans are capable of pre-shaping their hands to grasp an object based exclusively on visual information from an object. There are, in general, many possible kinds of grasps for each object; here we will focus on generating a specific grasp that is similar to a demonstrated grasp.

Convolutional neural networks (CNNs) have attracted a great deal of attention in the computer vision community and have outperformed other algorithms on many benchmarks. However, applying CNNs to robotics applications is non-trivial for two reasons. First, collecting the quantity of robot data typically required to train a CNN is extremely difficult. In this work, we learn a mapping from visual features extracted from a CNN trained on ImageNet [2] to grasp configurations of the entire anthropomorphic hand/arm and demonstrate that a small set of grasping examples is sufficient for generalizing grasps across different objects of similar shapes. Second, the final output of a CNN contains little location information from the observed object, which is essential for grasping. We take advantage of the hierarchical nature of the CNN to identify a set of features we call *hierarchical CNN features* that capture the hierarchical support relations between filters in different CNN layers. The 3D positions of such features can be identified by tracing activations of high-level filters backwards in the CNN to discover the locations of important structures that direct hand/arm control. Figure 1 shows the overall architecture and how such hierarchical CNN features are mapped to a point cloud to support manipulation.

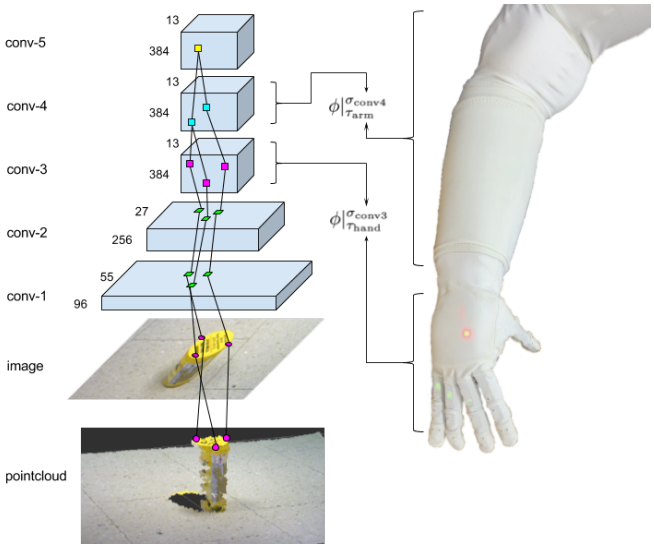

Fig. 1. Overall architecture of the proposed system. The input for this example is the RGB image and point cloud of a yellow jar. Tuples of the yellow dots (conv-5), the cyan dots (conv-4), and the magenta dots (conv-3) represent hierarchical CNN features. These features can be traced back to the image input and mapped to the point cloud to support manipulation. $\phi|_{\tau_{\text{hand}}}^{\sigma_{\text{conv3}}}$ represents the function that controls hand motor resources τ_{hand} based on conv-3 layer features σ_{conv3} and $\phi|_{\tau_{\text{arm}}}^{\sigma_{\text{conv4}}}$ represents the function that controls arm motor resources τ_{arm} based on conv-4 layer features σ_{conv4}. ϕ represents the potential function computed from input σ whose derivative with respect to motor resources τ constitute a controller.

We collected a grasping dataset using demonstrated hand/object relationships for Robonaut-2 [3] to evaluate the proposed approach in terms of the precision of the resulting preshape postures. We show that this proposed approach outperforms alternative approaches in cluttered scenarios on this dataset.

This approach is then tested in a grasping experiment on Robonaut-2 which hierarchical CNN features in the 4th and 5th layers are associated with the arm and hand controllers respectively. To our knowledge, this is the first work that associates features in different CNN layers with controllers that engage different kinematic subchains in the hand/arm systems. We demonstrate that the proposed approach out-

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

performs a baseline approach using the object point cloud. Figure 2 shows Robonaut-2 pre-shaping its hand before grasping novel cylindrical and cuboid objects.

Fig. 2. Robonaut-2 pre-shaping its hand before grasping an object. Notice that in the air duster example the fingers are pre-shaped to form an enveloping grasp and in the box example the fingers are pre-shaped to grasp the faces of the box.

II. RELATED WORK

The idea that our brain encodes visual stimuli in two separate regions was first proposed by Schneider [4]. Ungerleider and Mishkin further discovered the ventral and dorsal streams and proposed the hypothesis often known as the distinction of "what" and "where" between the two visual pathways [5]. However, in 1992 Goodale and Milner proposed an alternative perspective on the functionality of these two visual pathways based on many observations made with patient DF [6]. Patient DF developed a profound visual form agnosia due to damage to her ventral stream. Despite DF's inability to recognize the shape, size and orientation of visual objects, she is capable of grasping the object with accurate hand and finger movements. Based on a series of experiments [1], Goodale and Milner proposed the *perception-action model*, which suggests that the dorsal pathway provides action-relevant information about the structural characteristic of objects in addition to their position. Our work is inspired by these observations and associates grasp configurations with visual features instead of object identities.

Recently, studies on this perception-action model in the field of cognitive science have shifted from identifying functional dissociation between ventral and dorsal streams and, instead, focus on how these two streams interact [7]. Attentional blink experiments [8] suggest that object recognition processes in the ventral stream activate action-related processing in the dorsal stream, which leads to increased attention to objects that present an affordance relationship with the observed object. This interaction between the ventral and dorsal streams disappears when the stimuli are words rather than images. This result is consistent with our proposition that features learned for recognition tasks provide useful visual cues for grasp actions.

A large body of literature has focused on solving for grasp configurations. Sahbani *et al*. divide grasp synthesis approaches into analytic and empirical [9] while Bohg *et al*. further categorize grasp methodologies based on prior object knowledge, object feature, task, etc. [10]. Our work belongs to a set of empirical approaches that try to grasp familiar objects using both 2D and 3D features based on demonstrations. We compare our work to research that generates robot grasps from visual information. In work by Saxena *et al*., a single grasp point is identified using a probabilistic model conditioned on a set of visual features such as edges, textures, and colors [11]. Similar work uses contact, center of mass, and force closure properties based on point cloud and image information to calculate the probability of a hand configuration successfully grasping a novel object [12]. Platt *et al*. uses online learning to associate different types of grasps with the object's height and width [13]. A shape template approach for grasping novel objects was also proposed [14]. The authors introduced a shape descriptor called a height map to capture local object geometry for matching a point cloud generated by a novel object to a known grasp template. Other work uses a geometric approach for grasping novel objects based on point clouds [15]. An antipodal grasp is determined by finding cutting planes that satisfy geometric constraints. A similar approach based on local object geometry was also introduced [16].

Our approach associates CNN features trained on the ImageNet dataset with a demonstrated grasp. Hierarchical features consider both the local pattern and higher level structures that it belongs to. This visually-guided grasp can be successful even when there is insufficient 3D point cloud data. For example, a side grasp on a cylinder can be inferred even when only the top face is observed. In [17], a deep network trained on 1035 examples is used to determine a successful grasp based on RGB-D data. Grasp positions are exhaustively generated and evaluated. Our approach localizes features in a pre-trained CNN and generates grasp points based on a small set of grasping examples.

Several authors have applied CNNs to robotics. In [18], 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 our work, we also use filters in the third convolution layers; but unlike the previous work, we consider their relationship with higher layer filters. In [19], an autoencoder is used to learn spatial information of features in a neural network. Our approach finds the multi-level receptive field of a hierarchical feature in a particular image by repeatedly back tracing along a single path. In [20], 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. In contrast, our approach considers multi-objective configurations capable of control for more sophisticated and higher degree-of-freedom hand/arm systems like Robonaut-2.

A great deal of research has been done on understanding the relationship between CNN filter activation and the input image. In [21], deconvolution is used to find what pixels activate each filter. In other work, the gradient of the network response with respect to the image is calculated to obtain a saliency map used for object localization [22]. In [23], an

approach that adds a guidance signal to backpropagation for better visualization of higher level filters is also introduced. In our work, backpropagation is performed on a single filter per layer to consider the hierarchical relationship between filters. Recent work by Zhang *et al*. introduces the excitation backprop that uses a probabilistic winner-take-all process to generate attention maps for different categories [24]. Our work localizes features based on similar concepts.

Some authors have explored using intermediate filter activation in addition to the the response of the output layer of a CNN. 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 labelling [25]. Our approach groups filters in different layers based on their hierarchical activation instead of just the spatial relationship. In [26], 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 [27], the last layer is used to identify the object instance while the fifth convolution layer is used to determine the *aspect* of an object. In our work we consider a feature as the activation of a lower layer filter that causes a specific higher layer filter to activate and plan grasp poses based on these features.

Our work is also inspired by *deformable part models* [28] where a class model is composed of smaller models of parts; e.g., wheels are parts of a bicycle. We view CNNs similarly; if a higher layer filter response represents a high level structure, the lower layer filter responses that contribute to this higher layer response can be seen as representing local parts of this structure and may provide useful information for manipulating an object.

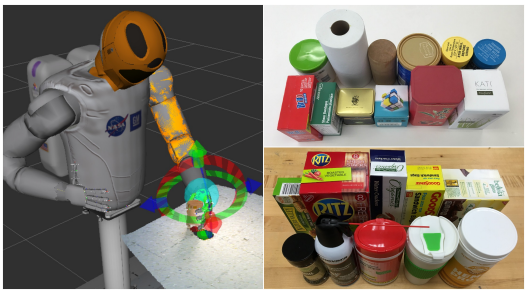

Fig. 3. Left: the data collection interface where the robot arm and hand is adjusted to the grasp pose. Right top: The set of objects used in the R2 grasping dataset. Right bottom: The set of novel objects used in the grasping experiment.

III. DATASET

In this work, we created the R2 grasping dataset that contains grasp records of each demonstrated grasp. A grasp record contains the point cloud and RGB image of the target object observed from the robot's viewpoint, and the Cartesian pose of each joint in the Robonaut-2 hand in the camera frame. The data is collected using an Asus Xtion camera and the Robonaut-2 simulator [29]. The object is placed on a flat surface where the camera is about 70 cm above and looking down at a 55 degree angle. We manually adjust the left robot arm and each finger of the left hand so that the robot hand can perform a firm grasp on the object. For cuboid objects, we adjust the thumb tip and index finger tip to the front and back faces of the cuboid and about 3cm away from the left edge of the face. For cylindrical objects, we adjust the thumb tip and index finger to perform an enveloping grasp on the object.

We collected a total of 120 grasping examples of twelve different objects. Six of the objects are cylindrical and six of them are cuboids as shown in Figure 3. The same object is presented at different orientations and under different lighting conditions. In addition to grasping examples with a single object, we also created 24 grasping examples in cluttered scenarios. Twelve of them include a single cylindrical object and twelve of them include a single cuboid object. The joint poses of the hand while grasping these objects are also recorded. Figure 7 shows a few examples of the cluttered test set.

IV. APPROACH

To learn how to grasp a previously unseen object, we use a CNN model similar to Alexnet [31] introduced by Yosinski [30] and implemented with Caffe [32] to identify visual features that represent meaningful characteristics of the geometric shape of an object to support grasping. The network is trained on ImageNet and not retrained for the grasping task. In this section, we describe how to identify features that activate consistently among the training data. We call these identified features that capture the hierarchical support relations between filters in different CNN layers *hierarchical CNN features*. We discover the locations of these features by repeatedly tracing filter activations backwards in the CNN along a single path. We then discuss how grasp points, Cartesian grasp positions of end effectors, are generated based on offsets between features and robot end effectors.

A. Identifying Consistent Features

Given a set of grasp records that demonstrates grasping objects with similar shapes, our goal is to find a set of visual features that activate consistently. Our assumption is that some of these features represent meaningful geometric information regarding the shape of an object that supports grasping. For each observed point cloud, we segment the flat supporting surface using Random Sample Consensus (RANSAC) [33] 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.

Next, filters that activate consistently over similar object types (*consistent filters*), in the fifth convolution (conv-5)

layer are identified. The top N^5 filters that have the highest sum of log responses $\sum_{a \in A_i^5} \log a$ among all the grasping examples of the same type of grasp are identified, where A_i^5 is the activation map of filter f_i^5, and f_i^m represents the i^{th} filter in the m^{th} convolutional layer. In this work, only one type of grasp is demonstrated for the same type of object. We observe that many filters in the conv-5 layer represent high level structures and fire on box-like objects, tube-like objects, faces, etc. Knowing what type of object is observed can determine the type of the grasp but is not sufficient for grasping, since boxes can have different sizes and presented in different pose. However, if lower level features exist such as edges or vertices, robot fingers can be placed relative to them.

CNN features 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. For example, we found that filter f_{87}^5 in the conv-5 layer in the CNN represents a box shaped object. This filter is a combination of several filters in the fourth convolution (conv-4) layer. Among these filters, f_{190}^4 and f_{133}^4 represents the lower right corner and top left corner of the box respectively. Filter f_{190}^4 is also a combination of several filters in the third convolution (conv-3) layer. Among these filters, f_{168}^3 and f_{54}^3 represents the diagonal right edge and diagonal left edge of the lower right corner of the box. The activation map of filter f_{168}^3 will respond to all diagonal edges in an image, but if we only look at the subset of units of f_{168}^3 that has a hierarchical relationship with f_{190}^4 in the conv-4 layer and f_{87}^5 in the conv-5 layer, local features that correspond to meaningful parts of a box-like object can be identified. Instead of representing a feature with a single filter in the conv-3 layer, our approach uses a tuple of filters to represent a feature, such as (f_{87}^5, f_{190}^4, f_{168}^3) in the previous example. We call such features *hierarchical CNN features*. Within a hierarchical CNN feature, a higher level filter is called the parent filter of a lower level filter. Figure 4 shows the visualization of several hierarchical CNN features identified in cuboid and cylindrical objects, including the features in the previous example.

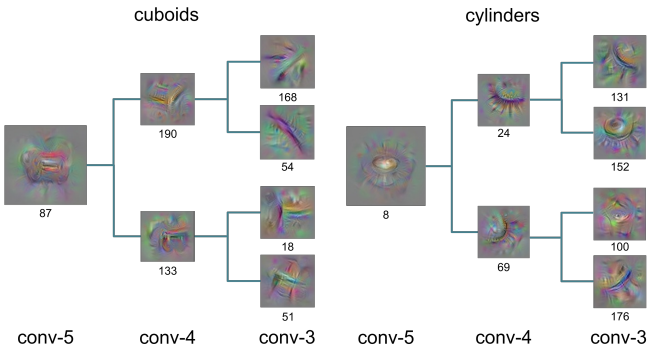

Fig. 4. Hierarchical CNN feature visualizations among cuboid objects (left) and cylinders (right). Each square figure is the visualization of a CNN filter while the edges connect a lower layer filter to a parent filter in a higher layer. The numbers under the squares are the corresponding filter indices. Filters are visualized using the visualization tool introduced in [30]. Notice that the lower level filters represent local structures of a parent filter.

In order to identify consistent filters in layers lower than the conv-5 layer, the max response of the activation map A_i^5 for each of the consistent filters f_i^5 in the conv-5 layer is first identified. For each consistent filter f_i^5, all responses except for the maximum a_{max}^5 are zeroed out and backpropagated to obtain the gradient of the max response with respect to the conv-4 layer: $G^4 = \partial a_{max}^5 / \partial$conv-4. The filters that contribute to f_i^5 consistently in the conv-4 layer are identified by picking the top N^4 filters that have the highest sum of log gradients $\sum_{g \in G_j^4} \log g$, where G_j^4 represents components of the gradient G^4 that corresponds to filter f_j^4. With the same procedure we then find the top N^3 filters that contribute to f_j^4 consistently by calculating the gradient with respect to the conv-3 layer: $G^3 = \partial g_{max}^4 / \partial$conv-3, where g_{max}^4 is the maximum partial derivative of G_j^4. For each hierarchical CNN feature, backpropagation is only performed on one unit per convolutional layer. This process traces backward along a single path recursively and yields a tree structure of hierarchical features.

To locate this feature in 3D we then backpropagate to the image and map the mean of the response location to the corresponding 3D point in the point cloud. The *response* of a hierarchical CNN feature is defined as the maximum gradient of the lowest layer filter in the tuple, e.g. g_{max}^4 for feature (f_i^5, f_j^4). Figure 5 shows an example of results of performing backpropagation along a single path to the image layer from features in the conv-5, conv-4, and conv-3 layers. The conv-3 layer features can be interpreted as representing lower level features such as edges and corners of the cuboid object.

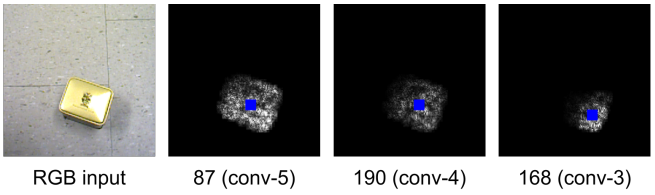

Fig. 5. Localizing features in different layers. The color image is the input image. The following images from left to right represent the location map of hierarchical CNN features (f_{87}^5), (f_{87}^5, f_{190}^4), and (f_{87}^5, f_{190}^4, f_{168}^3) obtained from backpropagating the feature response along a single path to the image layer. The blue dot in each location map represents the mean of the response locations. The mean response locations of conv-3 layer features are located closer to edges and corners of the cuboid object compared to the conv-4 and conv-5 features. The conv-3 layer features can be interpreted as representing local structures of the cuboid object.

B. Generating Grasp Points

For each target object we generate grasp points for the index finger, thumb, and the arm. Grasp points are the locations that the corresponding end effectors should be positioned for a successful grasp and are calculated based on a set of relative positions between 3D features and end effectors. For each type of object we store a set of possible grasp offsets for each end effector and the corresponding feature. A grasp offset is determined by the relative position from a robot end effector to a feature based on a demonstrated grasp. Although modeling a type of grasp based on the offsets between grasp

points and feature positions may not result in consistent grasp points when dealing with objects of different sizes at different orientations, there usually exists a subset of feature points that are close to the end effector consistently therefore result in smaller offset variances. In this work, we associate the robot hand frame to features in the conv-4 layer and the robot index finger and thumb to features in the conv-3 layer. This mapping is inspired by the observation that humans are capable of moving their arms toward the object for grasp without recognizing detailed features; the precise locations of low level features are only needed when the finger locations need to be accurate.

Not all features that fire consistently on the same object type are good features for planning actions. For example, when grasping a box, positioning the index finger, which contacts the back edge of the box, relative to the front edge of the box will result in positions with high variance, since the size of the box may vary. In contrast, positioning the index finger relative to the back edge of the box will result in a lower position variance. For each end effector position we pick the top N hierarchical CNN features that have the lowest 3D position variance among training examples and have the same parent filter in the conv-5 layer. The other features are then removed. The final set of features have the same conv-5 filter f_i^5. We found that restricting all the hierarchical CNN features to have the same parent filter allows our approach to perform well in a cluttered scenario.

During testing, the hierarchical CNN features associated with grasp offsets are first identified. The 3D positions of these features are then located through backpropagation to the 3D point cloud. A set of possible grasp positions are then calculated based on the grasp offsets and the 3D positions of the corresponding hierarchical CNN features. The grasp points for the robot hand frame, and end points for the thumb and index finger are then determined by the weighted mean position of the corresponding set of possible grasp positions with the feature responses as weights. Figure 6 shows examples of the grasp points and the set of possible grasp positions on different objects.

V. EXPERIMENTS

We ran two sets of experiments. In the first set, we evaluate the difference between the calculated grasp points and the ground truth in the R2 grasping dataset. In the second set, we test our approach by grasping novel objects using Robonaut-2 [3] and compare the number of successful grasps with a baseline approach.

A. Experiments on the R2 Grasping Dataset

In these experiments the performance of our approach on the R2 grasping dataset is analyzed. First, the accuracy of grasp points is measured with cross-validation. Second, approaches with and without the hierarchical CNN features are compared.

1) Cross-Validation Results: Cross-validation is applied by leaving out one object instance at a time during training and testing on the left out object by comparing the calculated grasp points to the ground truth. The distance between the example position and the targeted position of the hand frame, index finger tip, and thumb tip is calculated and shown in Table I. The average grasp position error for the hand frame is higher than the thumb and index finger; this is likely because the positions of local features alone are not sufficient to predict an accurate position for the hand frame. However, since the hand frame is not contacting the object, its position is less crucial for a successful grasp. Figure 6 shows a few results of cross-validation on different objects with different pose and lighting. Similar to the training data, the cuboid objects are grasped at positions closer to the left side while the cylindrical objects are grasped such that the fingers would wrap around the cylinder.

TABLE I
AVERAGE GRASP POSITION ERROR ON CYLINDRICAL AND CUBOID OBJECTS IN METERS.

	cylindrical objects						
left hand	cetaphil jar	wood cylinder	maxwell can	blue jar	paper roll	yellow jar	average
thumb tip	0.0197	0.0164	0.0134	0.0202	0.0136	0.0147	0.0163
index tip	0.011	0.0111	0.023	0.017	0.0155	0.0126	0.015
hand frame	0.0128	0.0393	0.0234	0.0183	0.0328	0.0173	0.024
	cuboid objects						
left hand	cube box	redtea box	bandage box	twinings box	brillo box	tazo box	average
thumb tip	0.0094	0.0101	0.0093	0.0095	0.0088	0.0111	0.01
index tip	0.0135	0.0207	0.0278	0.0134	0.0098	0.0157	0.0168
hand frame	0.0241	0.0203	0.0195	0.0177	0.0177	0.0285	0.0213

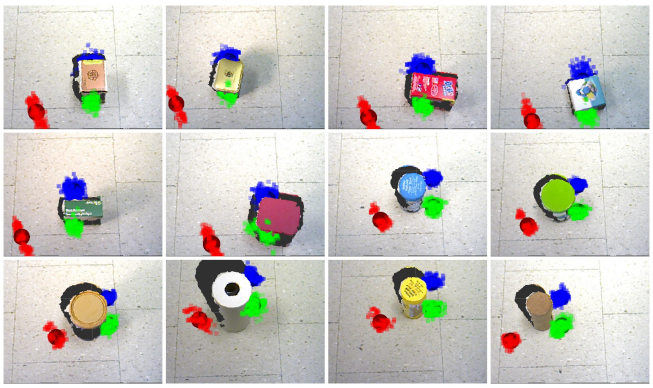

Fig. 6. Sample cross-validation results for single object scenario. The red, green, and blue spheres represent the calculated grasp points for the hand frame and endpoint positions for the thumb and index finger of the left robot hand. The grasp points are the weighted mean of the colored dots that each represents a possible grasp position based on one training example. Notice that for the cuboid object the grasp points for the thumb and index finger are located on the opposing face and about 3cm away from the left edge of the face as it was trained. For the cylinder object the grasp points for the thumb and index finger are on the right side of the cylinder to form an enveloping grasp. The black pixels are locations behind the point cloud that are not observable.

2) Comparison: To evaluate the proposed hierarchical CNN feature (hier-feat), we compare cross-validation results with four alternative approaches that do not consider relationships between layers. The first approach is a baseline

that uses the same set of features identified by the proposed approach but only considers the lowest level filter of the hierarchical CNN features, therefore removing relationship with higher level CNN filters. The grasp points are then generated based on offsets to these individual filters in each layer. The second approach (indv-filter) also associates grasp points with individual CNN filters instead of hierarchical CNN features but learns the set of filters that fire consistently instead of using the same set of features used by the proposed approach. Similar to the second approach, the third alternative (conv5-filter) identifies individual CNN filters instead of hierarchical CNN features but only considers filters in the conv-5 layer. The fourth approach (conv5-max) identifies the top five consistent filters in the conv-5 layer and uses the one that has the max response during testing. To make the comparison fair, we also remove filters other than the top N hierarchical CNN features that have the lowest position variance among training examples for the first three comparative approaches. We use $N^5 = N^4 = N^3 = 5$ and $N = 15$ in this experiment.

The results are shown in the first row of Table II. The proposed approach performs better than all four alternatives. However the difference is not significant, this is because the lower level filter that has the maximum response is mostly the same with or without restricting it to have the same parent filter when only one object is presented. In the next test, we show that associating low level filters with high level filters has a greater advantage when low level features may be generated from different high level structures, i.e., when there is clutter present. The fact that our approach outperforms the conv5-max approach shows the benefit of using lower level features to higher level features on planning actions. In the absence of clutter, the conv5-filter approach performs well because although filters in the conv-5 layer are more likely to represent higher level object structures, many of them also represent low-level features like corners and edges.

Hierarchical CNN features are most useful when the scene is more complex and the same lower layer filter fires at multiple places. Since the proposed approach limits the filters in the conv-3 layer to have the same parent filter in the conv-5 layer, only lower layer features that belong to the same high level structure are considered. These five approaches are further tested on the cluttered test set. A test case is considered successful if the distance errors of the thumb tip and index finger tip are both less than 5cm and the hand frame error is less than 10cm. The results are shown in the second row of Table II. Figure 7 shows a few example results on cluttered test cases. The proposed approach performs significantly better than the baseline, indv-filter, and conv5-filter approaches since filters may fire on different objects without constraining it to a single high level filter. Figure 8 shows two comparison results between the proposed approach and the baseline approach.

B. Experiments on Robonaut-2

This section describes the evaluation of the proposed pre-shaping algorithm based on the percentage of successful grasps on a set of novel objects on Robonaut-2 [3]. Details on the experimental setting, the hierarchical controller used for pre-shaping, and results are explained.

TABLE II

COMPARISON ON ALTERNATIVE APPROACHES.

	hier-feat	base-line	indv-filter	conv5-filter	conv5-max
Single Object Experiment: cross validation average grasp position error (cm)	1.719	1.805	2.114	1.755	2.138
Cluttered Experiment: number of failed cluttered cases (24 total)	2	20	13	19	2

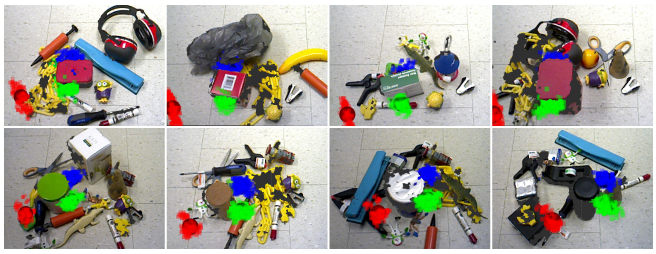

Fig. 7. Examples of grasping in a cluttered scenario. The red, green, and blue spheres represent the grasp points of the hand frame, thumb tip, and index finger tip of the left robot hand. The grasp points are the weighted mean of the colored dots that each represents a possible grasp position based on one training example. The top row is trained on grasping cuboid objects and the bottom row is trained on grasping cylindrical objects. Notice that our approach is able to identify the only cuboid or cylinder in the scene and generate grasp points similar to the training examples.

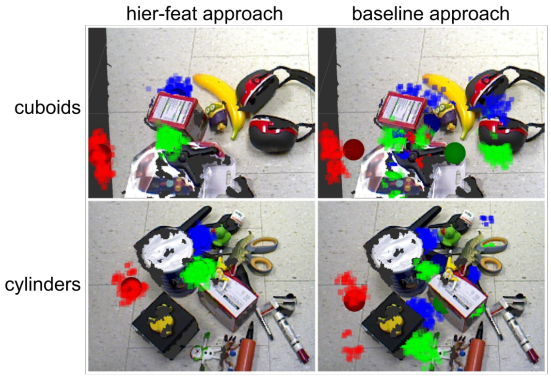

Fig. 8. Comparison in a cluttered scenario. Notice that the colored dots are scattered around in the baseline approach since the highest response filter in conv-3 or conv-4 layer are no longer restricted to the same high level structure.

1) Experimental Setting: For each trial, a single object in the novel object set is placed on a flat surface within the robot's reach. Given the object image and point cloud, the robot moves its wrist and fingers to the pre-shaping pose. After reaching the pre-shaping pose, the hand changes to a pre-defined closed posture and tries to pick up the object by moving the hand up vertically. A grasp is considered to be successful if the object did not drop after the robot tries to pick it up. A total of 100 grasping trials on 10 novel objects with our proposed approach and a comparative

baseline approach are tested. The novel objects used in this experiment are shown in Figure 3.

TABLE III
GRASP SUCCESS RATE ON NOVEL OBJECTS BASED ON 5 TRIALS PER OBJECT.

	cylindrical objects					
	tumbler	wipe package	basil container	hemp protein	duster	average
point cloud approach	40%	60%	0%	20%	40%	36%
hier-feat (our's)	100%	100%	100%	100%	100%	100%
	cuboid objects					
	cracker box	ritz box	bevita box	bag box	energy bar box	average
point cloud approach	80%	20%	60%	60%	60%	52%
hier-feat (our's)	100%	80%	100%	100%	100%	96%

2) Hierarchical Controller: A hierarchical controller constructed from hierarchical CNN features in different CNN layers is implemented to reach the pre-shaping pose. Given the object image and point cloud, our approach generates targets for the robot hand frame, index finger tip, and thumb tip. The hand frame target is determined based on hierarchical CNN features in the conv-4 layer while the thumb tip and index finger tip target is determined based on hierarchical CNN features in the conv-3 layer. The pre-shaping is executed in two steps. First, the arm controller moves the arm such that the distance from the hand frame to the corresponding grasp point is minimized. Once the arm controller converges, the hand controller moves the wrist and fingers to minimize the sum of squared distances from the index finger tip and thumb tip to their corresponding target.

These controllers are based on the control basis framework [34] and can be written in the form $\phi|_{\tau}^{\sigma}$, where ϕ is a potential function that describes the sum of squared distances to the targets, σ represents sensory resources allocated, and τ represents the motor resources allocated. In this work, the hand controller is represented as $\phi|_{\tau_{\text{hand}}}^{\sigma_{\text{conv3}}}$, where τ_{hand} is the hand motor resources and σ_{conv3} is the conv-3 layer hierarchical CNN features; the arm controller is represented as $\phi|_{\tau_{\text{arm}}}^{\sigma_{\text{conv4}}}$, where τ_{arm} is the arm motor resources and σ_{conv4} is the conv-4 layer hierarchical CNN features.

3) Results: The proposed algorithm is compared to a baseline point cloud approach that moves the robot hand to a position where the object point cloud center is located at the center of the hand after the hand is fully closed. The results are shown in Table III. Among the 50 grasping trials only one grasp failed with the proposed approach due to a failure in controlling the index finger to the target position. This demonstrates that the proposed approach has a much higher probability of success in grasping novel objects than the point cloud based approach. Figure 9 shows Robonaut-2 grasping novel objects during testing.

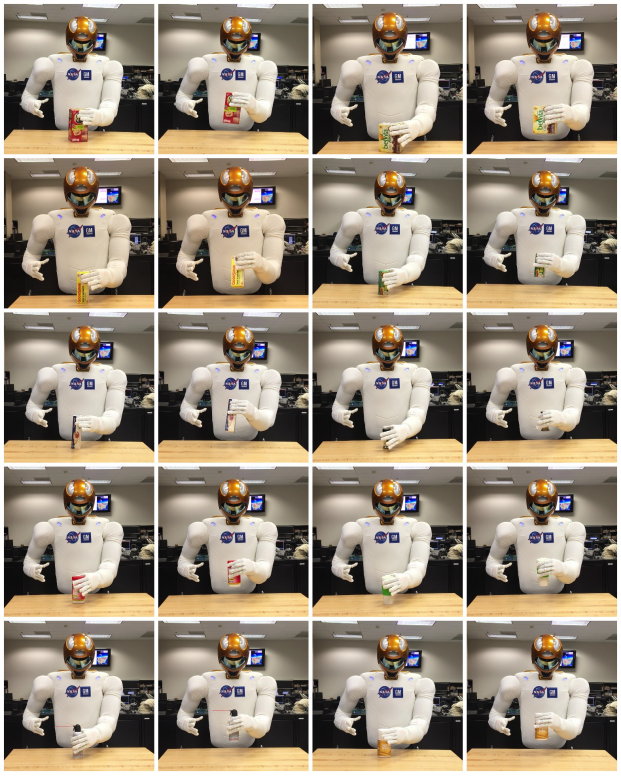

Fig. 9. Robonaut-2 grasping 10 different novel objects. The first and third columns show the pre-shaping steps while the second and fourth columns show the corresponding grasp and pickup. The cuboid objects are grasped on the faces while the cylinder objects are grasped such that the object is wrapped in the hand.

VI. DISCUSSION

Past studies have shown that CNN features are only effective for object recognition when most of the filter responses are used. In [35], it is shown that to achieve 90% object recognition accuracy on the PASCAL dataset most object classes require using at least 30 to 40 filters in the conv-5 layer. Research effort to invert CNN features [36] has also shown that most information is contained in a population of features with moderate responses and not in the most highly activated filters. In this work, we demonstrate that a subset of hierarchical CNN features that have the same parent filter in the conv-5 layer may be sufficient for controlling the robot end effector for grasping common household objects with simple shapes. Although using all filters may provide subtle information for object classification, we argue that when interacting with objects, the strength of CNN features lies in their hierarchical nature and a small set of filters are sufficient to support manipulation.

However, the current work is limited to objects with simple shapes that can be represented by a single filter in the conv-5 layer. In future work, we plan to extend to more complicated objects by learning higher layer filters that represent more complicated object types.

VII. CONCLUSION

In this work, we tackle the problem of pre-shaping a human-like robot hand for grasping based on visual input.

The hierarchical CNN feature that captures the hierarchical relationship between filters is introduced. The proposed approach first identifies hierarchical CNN features that are active consistently among the same type of grasps and localizes them by backpropagating the response along a single path to the point cloud. The arm controller is then associated with features in the conv-4 layer while the hand controller is associated with features in the conv-3 layer to perform a grasp. The proposed approach is evaluated on the collected dataset and show significant improvement over approaches that do not associate filters in different layers in cluttered scenarios. This solution is further tested in a grasping experiment where a total of 100 grasp trials on novel objects are performed on Robonaut-2. The proposed approach results in a much higher percentage of successful grasps compared to a point cloud based approach.

VIII. ACKNOWLEDGMENT

We are thankful to our colleagues Dirk Ruiken, Mitchell Hebert, Michael Lanighan, Tiffany Liu, Takeshi Takahashi, and former members of the Laboratory for Perceptual Robotics for their contribution on the control basis framework code base. We are also grateful to Julia Badger 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] M. Goodale and D. Milner, *Sight unseen: An exploration of conscious and unconscious vision*. OUP Oxford, 2013.

[2] 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.

[3] 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.

[4] G. E. Schneider, "Two visual systems." *Science*, 1969.

[5] M. Mishkin, L. G. Ungerleider, and K. A. Macko, "Object vision and spatial vision: two cortical pathways," *Trends in neurosciences*, vol. 6, pp. 414–417, 1983.

[6] M. A. Goodale and A. D. Milner, "Separate visual pathways for perception and action," *Trends in neurosciences*, vol. 15, no. 1, pp. 20–25, 1992.

[7] R. D. McIntosh and T. Schenk, "Two visual streams for perception and action: Current trends," *Neuropsychologia*, vol. 47, no. 6, pp. 1391–1396, 2009.

[8] M. Adamo and S. Ferber, "A picture says more than a thousand words: Behavioural and erp evidence for attentional enhancements due to action affordances," *Neuropsychologia*, vol. 47, no. 6, pp. 1600–1608, 2009.

[9] A. Sahbani, S. El-Khoury, and P. Bidaud, "An overview of 3d object grasp synthesis algorithms," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 326–336, 2012.

[10] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesisa survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2014.

[11] A. Saxena, J. Driemeyer, and A. Y. Ng, "Robotic grasping of novel objects using vision," *The International Journal of Robotics Research*, vol. 27, no. 2, pp. 157–173, 2008.

[12] A. Saxena, L. L. Wong, and A. Y. Ng, "Learning grasp strategies with partial shape information." in *AAAI*, vol. 3, no. 2, 2008, pp. 1491–1494.

[13] R. Platt, R. A. Grupen, and A. H. Fagg, "Re-using schematic grasping policies," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*. IEEE, 2005, pp. 141–147.

[14] A. Herzog, P. Pastor, M. Kalakrishnan, L. Righetti, J. Bohg, T. Asfour, and S. Schaal, "Learning of grasp selection based on shape-templates," *Autonomous Robots*, vol. 36, no. 1-2, pp. 51–65, 2014.

[15] A. t. Pas and R. Platt, "Using geometry to detect grasps in 3d point clouds," *arXiv preprint arXiv:1501.03100*, 2015.

[16] L. E. Zhang, M. Ciocarlie, and K. Hsiao, "Grasp evaluation with graspable feature matching," in *RSS Workshop on Mobile Manipulation: Learning to Manipulate*, 2011.

[17] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *The International Journal of Robotics Research*, vol. 34, no. 4-5, pp. 705–724, 2015.

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

[19] 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.

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

[21] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer vision–ECCV 2014*. Springer, 2014, pp. 818–833.

[22] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[23] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[24] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," in *European Conference on Computer Vision*. Springer, 2016, pp. 543–559.

[25] 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.

[26] 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.

[27] E. Wilkinson and T. Takahashi, "Efficient aspect object models using pre-trained convolutional neural networks," in *Humanoid Robots (Humanoids), 2015 IEEE-RAS 15th International Conference on*. IEEE, 2015, pp. 284–289.

[28] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1627–1645, 2010.

[29] P. Dinh and S. Hart, "NASA Robonaut 2 Simulator," 2013, [Online; accessed 7-July-2014]. [Online]. Available: http://wiki.ros.org/nasa_r2_simulator

[30] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," in *Deep Learning Workshop, International Conference on Machine Learning (ICML)*, 2015.

[31] 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.

[32] 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.

[33] 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.

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

[35] P. Agrawal, R. Girshick, and J. Malik, "Analyzing the performance of multilayer neural networks for object recognition," in *Computer Vision–ECCV 2014*. Springer, 2014, pp. 329–344.

[36] A. Dosovitskiy and T. Brox, "Inverting convolutional networks with convolutional networks," *arXiv preprint arXiv:1506.02753*, 2015.