

# Object Sensitive Grasping of Disembodied Barrett Hand

Neil Traft and Jolande Fooken  
University of British Columbia

## Abstract

The proposed goal of this project was to be able to adapt the grasp shape of a robotic hand, based on previous experience, to produce a more stable and dexterous grasp. Following this goal, we looked for different ways a robotic hand could build upon its previous grasping experience.

We explored the idea of employing the adaptive grasping algorithms of Humberston and Pai [1], but we found some limitations to implementing this work on the Barrett robotic arm and hand. Instead, we opted to try to classify or recognize the object being grasped using modern statistical techniques. We were able to train a neural network to recognize objects in both training and test data sets with a high degree of accuracy (in some cases over 99%). However, when these grasps were repeated on the robot, we were unable to obtain any kind of reliable recognition. The reasons for this may be due to a variety of factors which will be discussed in the following paper.

We conclude that the idea of using high-level knowledge about an object to choose strategies for grasping is justified and realizable. However, using neural networks as a tool for encoding this knowledge may not be viable.

## I Introduction

When humans perform simple grasping task in every day life, they depend on a combination of their visual system as well as their sensorimotor memory. Hereby, the human hand relies on about 17000 mechanoreceptive tactile units [2] embedded in the hairless skin of the palm that are able to give feedback in response to e.g. touch, pressure or vibrations, constantly adapting fingertip forces and grasping strength. Lifting up an object, such as a cup or a pen, is consequently followed by a cascade of sensory signal generation and processing [3].

In humans, visual information of the objects properties during grasping is important, however not essential [4]. Consequently, a lot of research effort has been put into tactile-driven approaches for robotic grasp control [5] [6]. The main challenge remains yet to find a dexterous robotic grasping technique that can cope with the wide range of different grasping contexts. In other words, to mimic natural human grasping behavior as accurately as possible.

Conventionally, there are two approaches to developing grasping strategies and algorithms. While the first one uses geometric object models, i.e. calculates a geometry-based, object-specific optimal hand posture, the second approach solely depends on tactile feedback upon contact with the object being grasped. Both

approaches have the drawback that each grasp will be performed independently of the previous grasp experience. In contrast to this, humans use previous grasping information to *preshape* their grasp. (The simple example of a person lying in bed at night and reaching for a glass of water as opposed to a phone or a book illustrates this.) Accordingly, more recent ideas integrate some kind of *grasp experience* into the planning of the subsequent grasp [7] [8].

## Grasp Preshaping

The main idea of grasp adaptation is to use previously acquired grasping knowledge to improve future grasping strategies. One possible method we proposed was to try to equalize time-to-contact across all fingers, based on previous grasp shapes. This idea was inspired by the adaptive grasping algorithms of Humberston and Pai [1]. However, in the course of the project we found many limitations to adapting this specific algorithm to the Barrett robotic arm and hand.

As the Barrett Hand is equipped with 1-DOF finger joints, preliminary preshaped grasps were very similar. Most of the grasping action is governed by the automatic TorqueSwitch<sup>TM</sup> mechanism in each finger [10]. Also, since the objects being grasped were not fixed to the pedestal, the hand had the tendency to push them around until all three fingers were making contact simultaneously.

In addition to the problems inherent in the hardware, we faced temporary technical difficulties with respect to the torque sensor data collection. As the torque sensor readouts would have been crucial in identifying the different times of contact for each finger, we finally discarded the idea of preshaping the Barrett Hand.

## Object Recognition

Our other main interest was in how the properties of the object being grasped would influence the sensor output. The Barrett Hand is equipped with a rich set of sensors which cover three different modalities. This is analogous to the mechanoreceptors of the human fingertips, which themselves cover at least three different modalities: strain, vibration, and rate of change [2]. In our case, the modalities are mass (given by the force torque sensor), geometric shape (inferred from finger joint positions), and pliancy (given by tactile pressure sensors). We expect that the combination of three such orthogonal modalities will constitute a fairly unique description of an object.

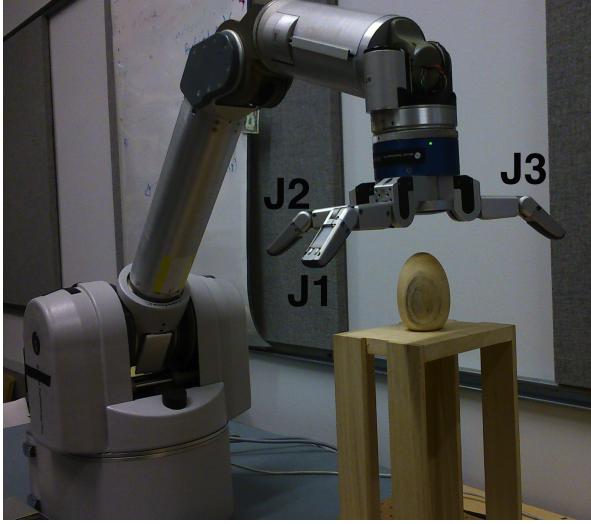
Given such compelling sensor feedback, would the system be able to recognize an object from a predefined trained set? We ultimately opted to bring statistical classification to bear on this question. This approach is motivated by the idea that once the system has more high-level information about the type of object it is sensing, it can employ grasps/strategies suited to that particular type of object. A key part of using previous experiences is being able to sort and categorize those experiences.

## II Methods

### System Overview

The system consists of the 7-DOF Barrett WAM robot arm and 4-DOF Barrett BH-280 Hand from Barrett Technology, Inc (compare figure 1). The robot is equipped with one 6-DOF wrist torque sensor, three 1-DOF finger joint torque sensors, and four 24-DOF tactile pressure sensors, making for a total of 105 independent

sensor inputs. Given such rich sensory input, we hoped to obtain feature vectors which exhibit statistically significant differences between different grasp shapes. All the sensors read at 125 Hz. Most afferent inputs in humans run at less than 60 Hz, so this rate is sufficient to mimic physiological driven grasping approaches [9].

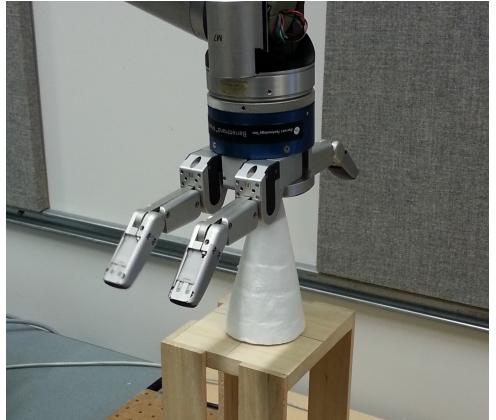


**Figure 1** 7-DOF Barrett WAM robot arm and 4-DOF Barrett BH-280 Hand from Barrett Technology, Inc.

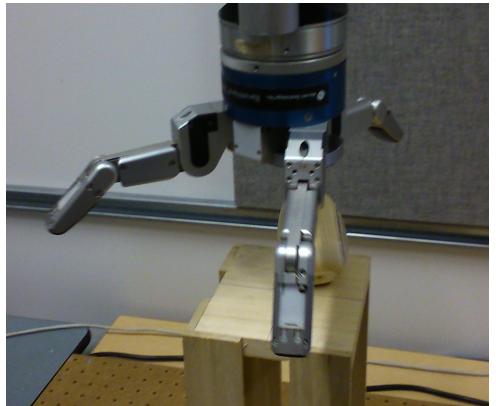
## Software Architecture

The software for running our experiments on the Barrett WAM and Hand is a menu-based command-line program that makes it easy to record sensor data and test out differently trained neural networks. The hand's home position as well as the initial grasp positions are predefined and can be called individually. On the main operational level, the menu lets the user choose one of 5 different grasp types:

## Top-down Prismatic Precision



## Top-down Tripod Precision



## Side-on Heavy Wrap



**Side-on Power Grip** as well as  
**Side-on Prismatic Precision**



Thus, each grasp is a combination of two factors: the position of the hand and the position of the fingers. The position of the hand, referred to as the "target position", can be either from the side (side-on) or from above (top-down). The position of fingers 1 and 2 can be at 0° (prism), 30° (tripod), or 180° (wrap). When the user chooses one of the above grasps, the robot follows a fixed sequence of states:

**First** Move to preparatory position.

**Second** Prepare (preshape) the hand for the particular grasp type (prism, tripod, or wrap).

**Third** Move to the target position (side-on or top-down).

**Fourth** Close the hand on the object.

**Fifth** Lift the object briefly and return it to the pedestal.

**Sixth** Release the object and retreat to the preparatory position.

During steps 3-5, the following sensors are recorded and logged to disk:

- WAM joint positions
- Finger joint positions (outer link)
- Finger torques
- 3D wrist forces
- Palm and finger tactile pressures

The software is structured such that all menu options are executed asynchronously. The user always retains control and can cancel the current sequence at any time. Over time working with the robot we also found it necessary to add various facilities for identifying the name of the object currently being grasped, resetting the hand/WAM if they have controller issues, and recording a failed grasp. We use these annotations to sort and label our sensor data samples.

## Data Collection

The objects grasped varied in shape, size, symmetry, texture, weight, pliancy, and firmness. For details see Figure 2. Each object was grasped several times with 3 (if possible) different grip strategies (Top-down Prismatic Precision, Heavy Wrap, and Power Grip).

Data from many trials of grasping these objects were collected into log files. These files were then imported into MATLAB and sorted by object and grasp type. From these files, we took only the interval during which the object was being grasped.

The finger joint positions were used to determine the time interval for sensor sampling. Initially, the finger torques were used for this purpose, but later in the project we encountered technical difficulties in communicating with the



**Figure 2** Set of grasped objects on which classification was performed. (1) styrofoam ball, (2) soft foam, (3) styrofoam cone, (4) foam square, (5) wood block, (6) plush octopus, (7) foam butterfly, (8) packaging tape, (9) rattan ball, (10) cookie cutter, (11) wooden egg, (12) football, (13) foam star, (14) drinking bottle, (15) cube, and (16) bean bag.

strain gages, and this method had to be altered. Luckily, the joint position data proved sufficient. As the two positional changes (i.e. the maximum and minimum of the derivative) mark start and end time point of each grasp, the specific time stamps for each trial could be calculated and used to find only the relevant data set for the neural network analysis. Subsequently, each object was assigned a label to run a classification.

## Neural Network

To classify a grasp, the sensor data were normalized and used to train a three layer neural network. We read a total of 103 sensor values, and classified among 16 possible objects. Additionally, we formed a class ‘Failed Grasp’ to which we assigned all failed grasps independent of the object, making for a total of 17 classes.

Therefore, the neural network consists of a

103 node input layer, a 25 node hidden layer, and a 17 node output layer. The implementation was largely based on the one found in Andrew Ng’s Machine Learning course [12]. Since this single hidden layer with 25 nodes was enough to perform robust character recognition in [12], it was deemed a satisfactory configuration for our purpose as well. Figure 3 depicts the nodes in the three layers of the neural network.

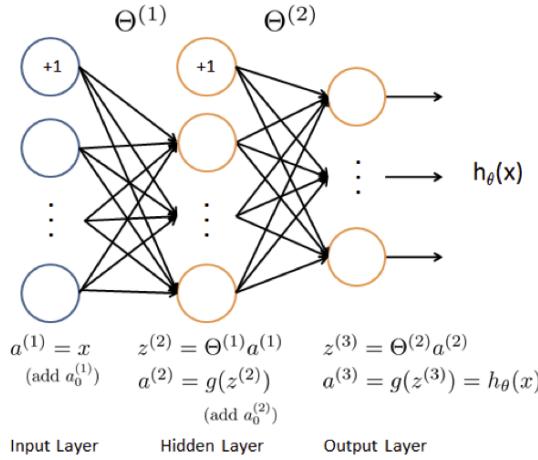
After the first round of experiments, we used the labeled data we collected to train a separate neural network for each of the three chosen grasps. All 103 features were separately normalized before training. For each sensor  $i$ , the column vector  $\mathbf{v}_i$  of all samples becomes

$$\mathbf{v}_i = \frac{\mathbf{v}_i - \text{mean}(\mathbf{v}_i)}{\text{std}(\mathbf{v}_i)}$$

Neural networks were trained using a cost function  $J(\Theta)$  similar to  $K$  regularized logistic regressions, where  $K$  is the number of classes, 17. We also added a regularization term with an adjustable weight  $\lambda$ . The training algorithm iteratively finds the parameters  $\Theta$  which minimize the cost function  $J(\Theta)$  by computing the cost function gradient  $\frac{\partial J(\Theta)}{\partial \Theta}$  in the neural network at each step over all training examples. For full details of the algorithm, see [12].

We produced a variety of different networks with different regularization weights from  $\lambda = 1$  to  $\lambda = 1000$ . 20% of the collected time points from each grasp were set aside as a test set to verify our results.

At this point a module was added to the software which predicted the object being grasped, given one or more samples of the above sensor data. The final version of the software printed out the name of the object which it “sensed”, while lifting the object from the



**Figure 3** A three layer neural network for the classification of grasped objects from finger pose, wrist force, and tactile pressure data. Taken from Programming Exercise 4 in [11].

pedestal. This was implemented by taking a single time slice of sensor data while grasping an object, and feeding it forward through each layer of the neural network. In each layer,

$$\mathbf{a}^{(i+1)} = \text{sigmoid}([\mathbf{e} \ \mathbf{a}^{(i)}] \Theta^{(i)})$$

where  $\mathbf{a}^{(1)} \in \mathbb{R}^{m \times 103}$  = the sensor samples  
 $\mathbf{a}^{(i)}$  = output of layer  $i$   
 $\Theta^{(i)}$  = parameters of layer  $i$   
 $\mathbf{e}$  = column of all ones  
 $\text{sigmoid}(z) = \frac{1}{1 + e^{-z}}$

The final prediction is taken as the label which was assigned the maximum probability by the output layer:

$$\text{object} = \text{index of } \max(\mathbf{a}^{(3)})$$

### III Results

#### Sensor Data

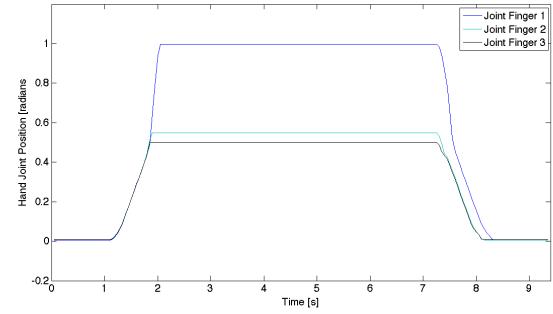
The output data of the various sensors may indicate different modalities of the object being grasped. While the force torque sensor will react strongly to the weight of the object, the finger joint positions are more sensitive to the shape. Orthogonal to either of these, the tactile pressure sensors will give information of the object's compliance.

If we had obtained finger torque measurements, these would have additionally contributed to our picture of both the shape and the compliance of the object. Unfortunately, the data recording of the finger strain gages caused major technical difficulties so that it could not be done consistently. This was a severe setback, as the finger torques are the most sensitive measure to initial contact with the object. Not only did this halt our plans for a grasp preshaping algorithm, it also forced us to reprogram part of our data collection method.

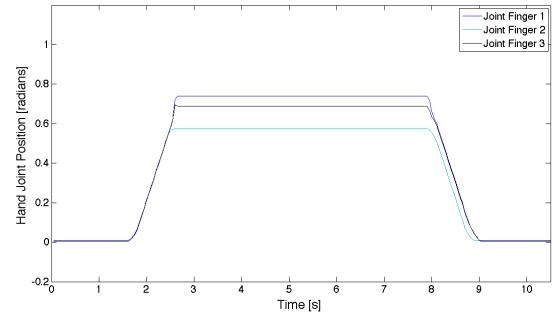
Still, the remaining sensors give us quite a full and diverse description of an object. Let us discuss the finger positions. Figure 4 shows the joint position profile while grasping the cone under each of the respective grasp types. The positional change is recorded in radians over the time span of the grasp. If we examine these graphs we can gain insight into the nature of both the grip and the object being grip.

Once the grasp is initiated, the position of each joint increases steadily until the object is fully enclosed in the hand. The joints remain at this maximal position until the object is released. The most interesting grip in this scenario is the top-down prismatic, where the

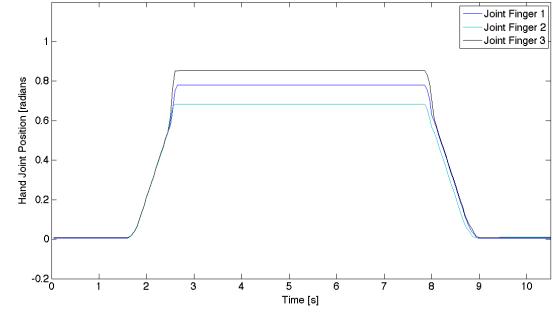
hand grips the rather thin top of the cone. As the physical gap between fingers 1 and 2 is larger than the upper circumference of the cone, we can see that finger 1 fails to make full contact and thus moves further than finger 2. As opposed to this, in the side-on grips the fingers just wrap around the base of the cone, thereby only showing small positional variations. These differing scenarios play themselves out clearly in the sensor data.



(a) Top-down Prismatic Precision



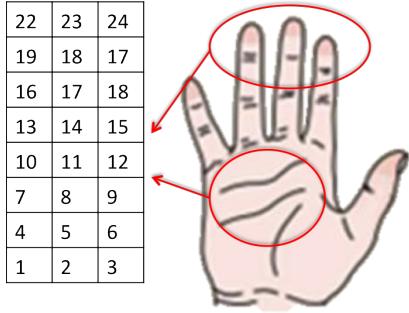
(b) Side-on Heavy Wrap



**Figure 4** Finger joint position in radians versus time for the (a) Prismatic Precision Grasp, the (b) Heavy Wrap, and the (c) Power Grip.

We were especially interested in the output of the tactile pressure sensors. The three Barrett fingers as well as the palm are provided with a tactile sensor array, consisting of 24 pressure sensors. The sensors are arranged in an 8x3 array. In the following, the sensor cell will be referenced according to the enumeration given in figure 9. Note: the distal finger tip is always depicted at the top of the map, while the bottom

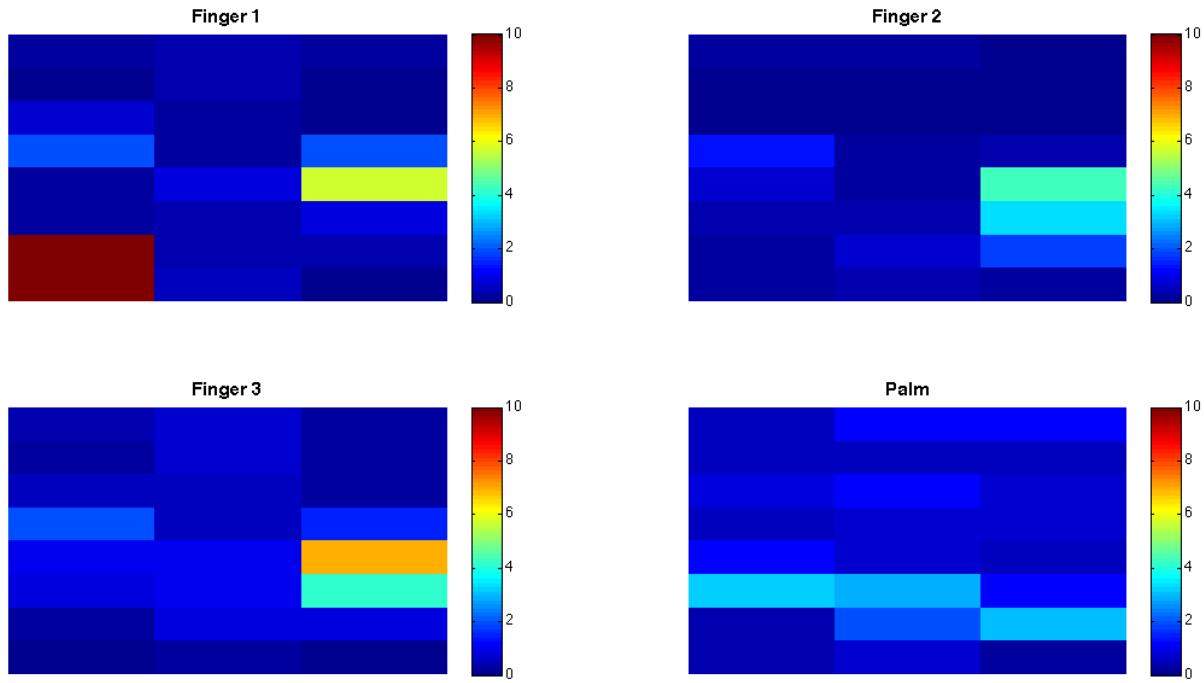
cells represent the proximal end of the finger tip.



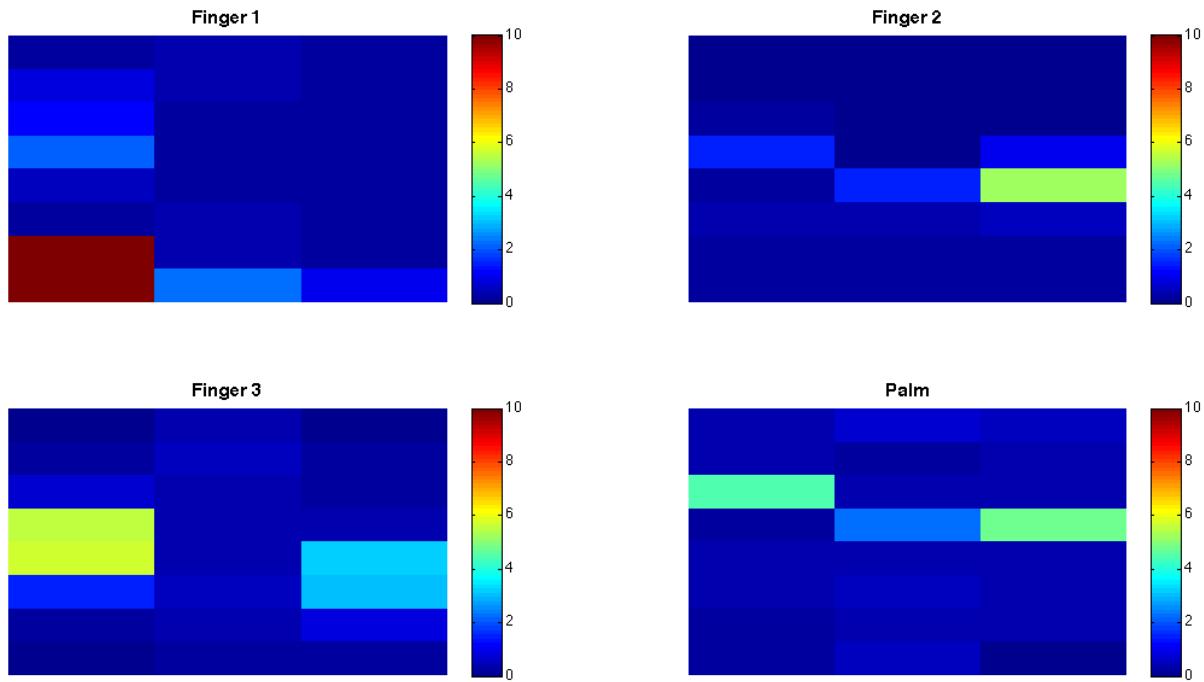
**Figure 5** Sensor arrays in finger tips and palm with 24 sensors each.

For each cell the mean pressure value during grasping was calculated for each finger and the palm. We then compared different characteristics of the material to see which showed the most prominent feature in the pressure maps. Generally, all pressure values were recorded in  $\frac{N}{cm^2}$ . First, we compared the object's shape. Figure 6 shows two objects of similar weight: an upright square wood block (compare object 5 in figure 2) and a round water bottle (compare object 14 in figure 2), gripped with the Heavy Wrap.

The first observation is that for both objects, cells 1 and 4 in finger 1 show significantly higher pressures than all other cells. This was consistent through all measurements for all grips and all objects of that specific data collection. We therefore conclude that something was blocking/triggering these cells leading to faulty data output. Consequently, we did not use these two cells to train the neural network.



(a) Side-on Heavy Wrap grip of square wood block (object 5)



(b) Side-on Heavy Wrap grip of round water bottle (object 14)

**Figure 6** Pressure maps of fingers 1-3 and palm of Barrett hand. Pressures are recorded in  $\frac{N}{cm^2}$  and plotted as a mean over the grasping trial for each respective cells. The Side-on Heavy Wrap compared for a square and round object.

The pressure profiles differ slightly, mainly for the palm and for finger 1. As the object is being grasped from the side-on, the fingers wrap around it tightly. Accordingly, pressure sensors 7-18 are most prominent. Interestingly, the pressure is higher on the finger side than in the middle, which is most likely due to the bulky (rather square) shape of the Barrett Hand.

Even though the pressure maps show some distinctive features, the difference is not as striking as one might expect. However, unlike the human finger, the Barrett finger only has two phalanges, therefore it is not able to bend at the distal interphalangeal joint [3]. The sensor array will therefore only touch one side of the wood log or the bottle, respectively, making it insensitive to the object's shape.

Next, we compared the pressure profiles of two objects of the same shape, but different weight, surface structure, and slightly different size (compare figure 7). The two balls (rattan ball styrofoam ball) were gripped from top-down with the prismatic precision grasp. Again, sensors 1 and 4 showed faulty pressure data and were ignored. Unfortunately, sensor 3 of finger 1 also began to give unreasonable high feedback in the course of the data collection and was thus ignored.

Again, the pressure maps show similar features. Most of the feedback is observed for the cells (7-18) in the middle of the finger. The pressure in J3 is slightly higher as finger 3 has to compensate for the two fingers (J1 and J2) gripping from the opposite site.

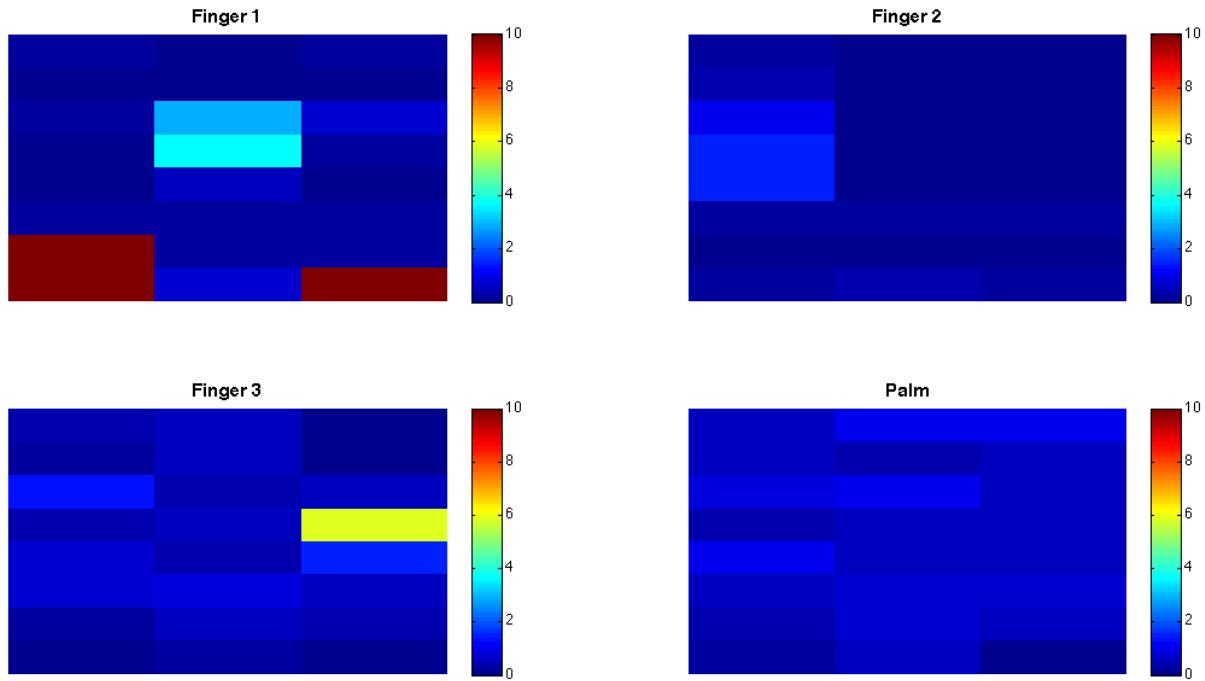
It is interesting that the pressures while gripping the lighter styrofoam ball are slightly higher than the respective pressures while grasping the rattan ball. This is most likely due to the smaller size of the styrofoam ball. The fingers can con-

sequently close further around the ball and thus tighten the grip. Additionally, the styrofoam ball has a smooth surface so that the fingers can tightly wrap around the surface, while the rattan ball has a rough surface that impede a tight grip.

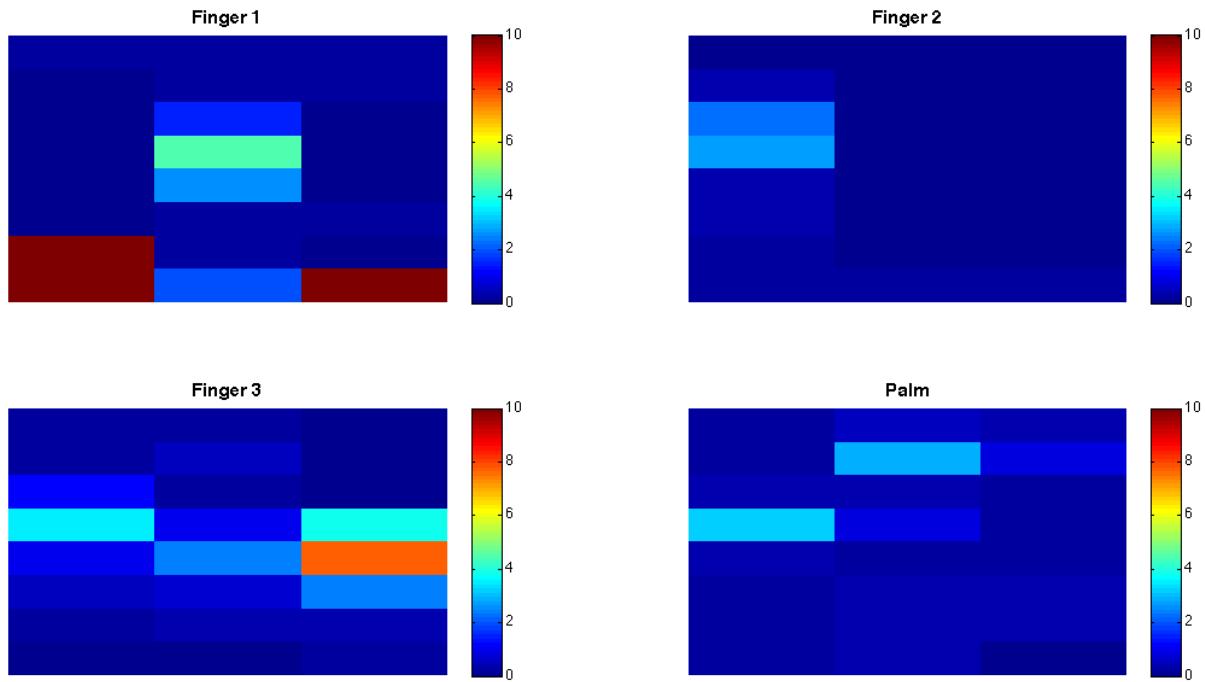
The important factor of compliance becomes even more apparent, when comparing the pressure maps of a soft and hard piece of foam (compare figure 8) for a side-on power grip. The two objects, soft foam (object 2) as well as foam square (object4), were similar in size, shape, and weight. However, while the soft foam was very compliant, the foam square was rather firm.

Note that again sensors 1 and 4 of finger 1 give faulty feedback and were not taken into account for any analysis. As the fingers approach side-on, they grab the square-shaped foam pieces longitudinally. The power grip really closed around the object until the foam was tightly squished. For the firm foam the two fingers push the square into an angled asymmetric position, so that the palm only receives pressure on one side, while J3 is pushing back hard with the distal end of the finger tip. As opposed to this, the pressure sensors show no response to the soft foam. This was consistent for all grasps of the soft foam and other soft objects, such as plush octopus (object 6).

We conclude that the compliance of the object is the main factor which influences the response in the pressure sensors. The size of the object will influence how tightly it can be gripped and thus also show its effect, though indirectly. Grasping small objects such as the cube or the bean bag with the power grip, finger 1 will not make contact at all, thus leaving the pressure readout blank.

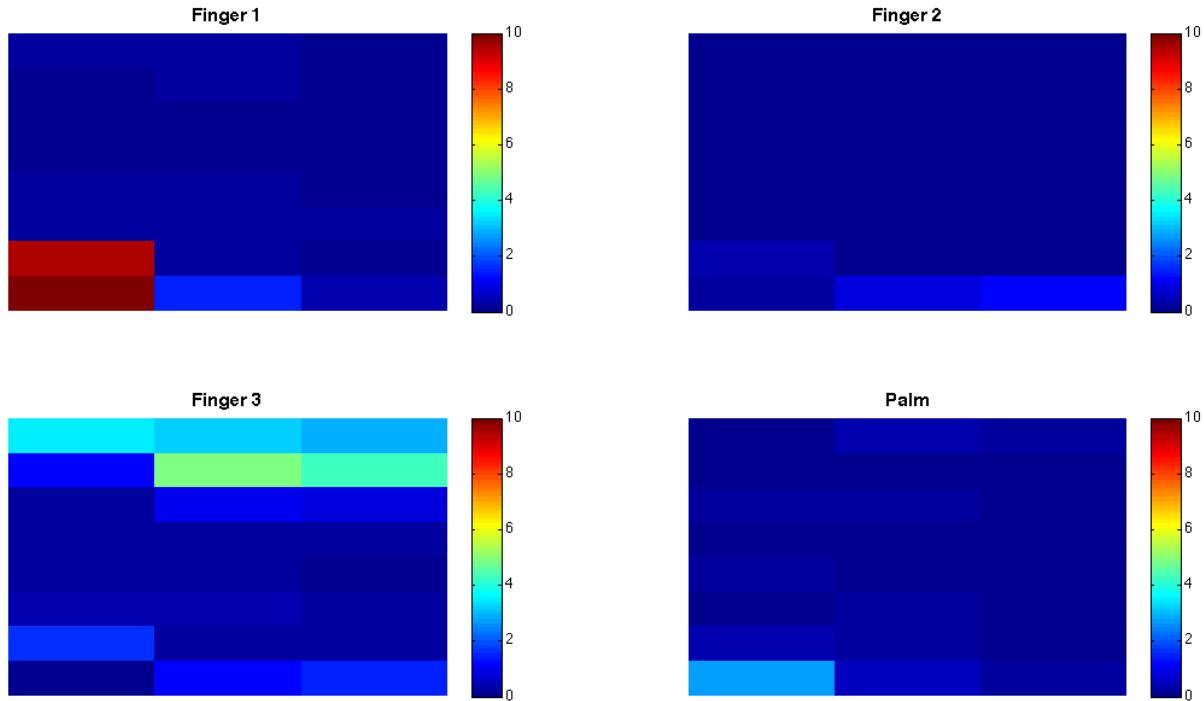


(a) Top-down Prismatic Precision grip of rattan ball (object 9)

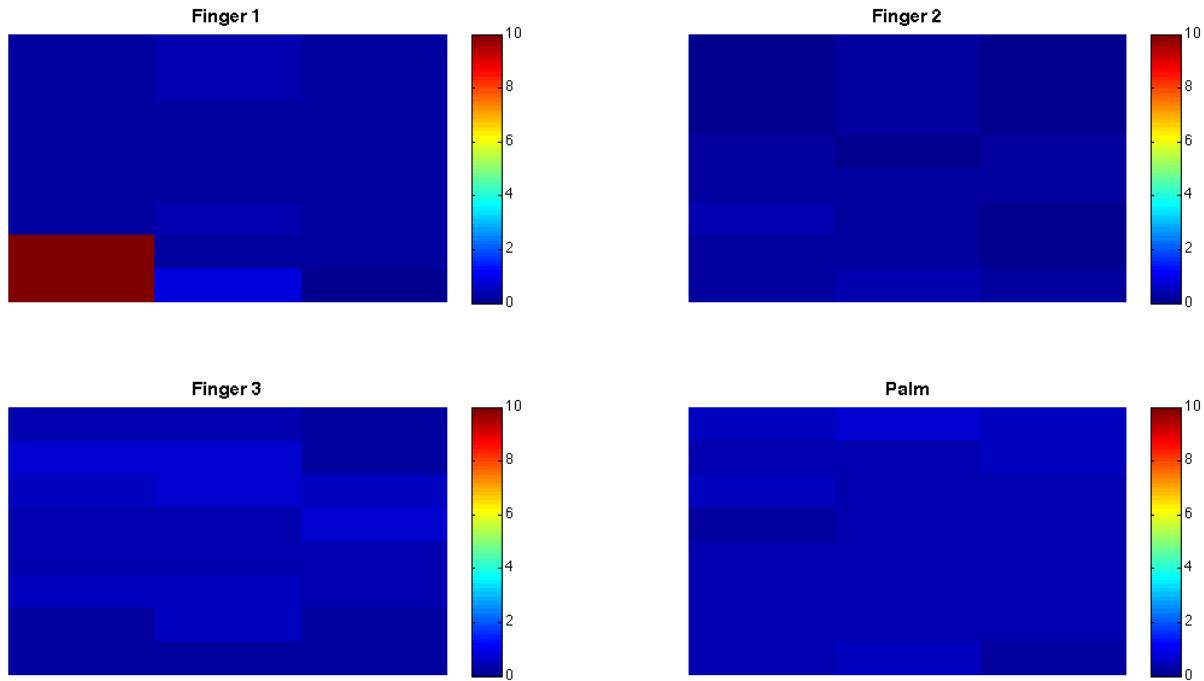


(b) Top-down Prismatic Precision grip of styrofoam ball (object 1)

**Figure 7** Pressure maps of fingers 1-3 and palm of Barrett Hand. Pressures are recorded in  $\frac{N}{cm^2}$  and plotted as a mean over the grasping trial for each respective cells. The Top-down Prismatic Precision grip compared for two round objects with different weights and surface characteristics.



(a) Side-on Power Grip of hard foam square (object 4)



(b) Side-on Power Grip of soft foam square (object 2)

**Figure 8** Pressure maps of fingers 1-3 and palm of Barrett hand. Pressures are recorded in  $\frac{N}{cm^2}$  and plotted as a mean over the grasping trial for each respective cells. The Side-on Power Grip compared for a soft and a hard square piece of foam.

## Neural Network

The goal of the neural network analysis was to assign labels to certain objects and train the hand to familiarize itself with the sensor response generated by each object. In this way it should be able to recognize an object while grasping it, and then plan the next grasp accordingly. We tried to implement this method and improve its performance in the course of the project. Each step listed below was taken because the neural network analysis of the previous version showed no meaningful results (i.e. when performed on the robot it was not able to name an object correctly).

- Initially we attempted training on raw, unnormalized sensor data. The network performed poorly, attaining only 16% accuracy even on the training set itself. We found that normalization of the data was absolutely crucial for training the neural network to good accuracy.
- After normalization, the data set was split into training (80%) and test data (20%) randomly. For both training and validation the neural network analysis reached suspiciously high accuracy of more than 96%.
- Faulty pressure readouts were detected and removed. We considered average read outs of over  $15 \frac{N}{cm^2}$  as faulty, especially when identical readings were observed over many different objects and grasp types.
- The high accuracy on the test set and poor performance in the real world indicated to us that we had overfit the data. To mitigate this issue we increased the regularization weight. This led to a lower accuracy of

training and validation step in the neural network analysis and seemed to favor certain objects for the respective grasp types. It did not significantly improve real-world performance, but did give us some insight to be discussed below.

- As the number of trials for some objects were significantly higher than for others, we excluded these objects to provide a more balanced data set. This also seemed to afford no improvement.

Despite all our efforts to improve the neural network analysis, we were unable to obtain any kind of reliable performance. The fact that performance on the validation set nearly always matches performance on the training set *should* indicate that no overfitting occurred. However, it may be the case that the validation data was in fact too similar to the training data, since they were acquired as different time slices of the same grasp, rather than being taken from totally different grasp samples. This suspicion is emphasized by the 99% accuracy of the neural network, a strong indication for overfitting the data.

Tests on the robot confirmed this. Objects could not be recognized correctly at all. However, for each grasp type, there were two or three select objects which would be identified correctly a majority of the time. The system seemed to prefer these objects and named these repeatedly, independent of which object was being grasped.

Adjusting the regularizer gave us a lot of insight into this phenomenon. We were able to expose this behavior in the test data by using very heavy regularization. Running the network with  $\lambda = 100$  led to a much lower accuracy for the neural network training—about 36%. When

we examined the individual predictions for each object, we found that there were a few objects which dominated the predictions. These objects received all of the predictions, thereby exhibiting 100% recall (true positives / actual positives) but very low precision (true positives / predicted positives). The other objects therefore had 0% recall.

To summarize, the neural network analysis did not work for the given training and validation data. It appears to be too heavily biased toward certain objects, though we are still unsure as to why. This behavior is usually not evident in the test set. Our conclusion is that either there were not enough individual grasps sampled, or the method does not hold for the desired task.

## IV Limitations

Even if our method had performed as well on the robot as it did on the test set, there were many limitations to using this approach for object recognition.

First and foremost, the method is limited only to objects which have been observed before. It is designed only to recognize a known object; it does not encode any higher-level characteristics which can then be observed in new objects.

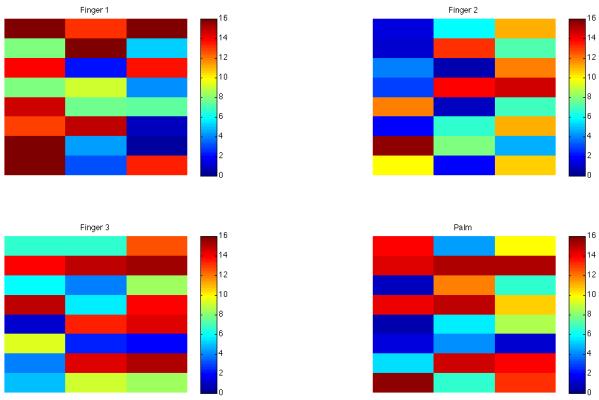
The method is also highly dependent on object orientation and size. The neural network needs to have been trained with a grasp of an object in a *particular* orientation in order to be able to recognize that object the next time it is observed in that orientation.

Nor is it invariant to the shape of the hand, since we use raw sensor data rather than extracting a feature vector from local keypoints,

as is typical in computer vision. Because of this dependence on hand and finger pose, in order to perform recognition over all the grasp types shown in Section II, it was necessary to train a *separate* neural network for *each* type, and use the network which corresponds to the current grasp when performing predictions.

We also consider the method likely to break down when number of objects increases. Classification gets significantly harder the more classes you have to decide between (not to mention training becomes much costlier). There is likely to be some point at which splitting hairs between similar classes becomes intractable.

Another drawback is the rather inaccurate tactile sensing. Object shapes do not necessarily show up in the pressure maps of the tactile sensor arrays. Due to the small spatial resolution of the sensor arrays, localization of shape contours is coarse. In addition, the sensors gave repeatedly random feedback (compare figure ??) which made it difficult to rely on their information. Due to the tightness of the grasps, contact surfaces with the fingers are often broad. These particular sensors probably call for a treatment very different from the edge/corner detection of computer vision algorithms.



**Figure 9** Faulty pressure read out exemplary for top-down prismatic precision grip. Such readout occurred repeatedly for each respective grasp.

## V Conclusion and Future Work

The neural network, as we implemented it, did not yield valid predictions of the object being grasped. The question that remains is, if the method is not suitable for this specific setup or if the data set used to train and validate the network were too similar. More questions remain: Why does the network prefer certain objects for certain grasps? What feature influences the neural response? Was there one feature that dominated all other sensor input? To answer all these questions the opacity of the neural network needs to be uncovered.

Possibilities for the future are numerous. Now, that the system is up and running, more data samples could be collected to have different sets for neural network training and validation. An important issue would be to fix the lack of finger torque data. In addition to the current setup, a setup with fixed objects could be further explored. Our original idea to implement a preshaping of the hand could then maybe be performed in a combination of initial contact and

object recognition.

In the course of the project, we became painfully aware of the difficulty of collecting sufficient data for doing statistical techniques on robot experiences. Robots are often slow, and collecting recordings of their experiences in the world is time consuming and resource-intensive. One major lesson from all the setbacks is that there may be more to gain from using what is known about human motor control, rather than unpredictable and black-box statistical techniques. Until robots are in widespread use around the world, there may not be enough variety of experiences for them to learn from by brute force alone. We should instead start from a known point using existing knowledge of human haptics and optimize from there.

## References

- [1] B. Humberston and D. Pai, “Interactive Animation of Precision Manipulations with Force Feedback,” *Draft*, 2013.
- [2] S. Lederman, *Encyclopedia of human biology*, vol. 8. 2 ed., 1997.
- [3] G. Tortora and B. Derrickson, *Principles of Anatomy and Physiology*. 13 ed., 2011.
- [4] P. Jenmalm and R. Johansson, “Visual and Somatosensory Information about Object Shape Control Manipulative Finter-tip Forces,” *The Journal of Neuroscience*, vol. 17, pp. 4486–4499, 1997.
- [5] M. Lee and H. Nicholls, “Tactile sensing for mechatronics – a state of the art survey,” *Mechatronics*, vol. 9, pp. 1–31, 1999.
- [6] H. Yousef, M. Boukallel, and K. Althoefer, “Tactile sensing for dexterous in-hand manipulation in robotics – A review,” *Sensors and Actuators A: Physical*, vol. 167, pp. 171–187, 2011.
- [7] J. Steffen, R. Haschke, and H. Ritter, “Experience-based and Tactile-driven Dynamic Grasp Control,” *International Conference on Intelligent Robots and Systems*, vol. 17, pp. 2938–2943, 2007. San Diego.
- [8] P. Pastor, L. Righetti, M. Kalakrishnan, and S. Schaal, “Online Movement Adaptation Based on Previous Sensor Experiences,” *International Conference on Intelligent Robots and Systems*, vol. 17, pp. 365–371, 2011. San Francisco.
- [9] R. Howe, “Tactile sensing and control of robotic manipulation,” *Advanced Robotics*, vol. 8, pp. 245–261, 1993.
- [10] Barrett and Technologies, *BH8-Series User Manual Firmware Version 4.4.x*.
- [11] A. Ng, “Machine Learning ,” 2013.
- [12] A. Ng, “Machine Learning ,” 2013.