Learning Robot Grasping from 3-D Images with Markov Random Fields

Abdeslam Boularias, Oliver Kroemer, Jan Peters

***Abstract*— Learning to grasp novel objects is an essential skill for robots operating in unstructured environments. We therefore propose a probabilistic approach for learning to grasp. In particular, we learn a function that predicts the success probability of grasps performed on surface points of a given object. Our approach is based on Markov Random Fields (MRF), and motivated by the fact that points that are geometrically close to each other tend to have similar grasp success probabilities. The MRF approach is successfully tested in simulation, and on a real robot using 3-D scans of various types of objects. The empirical results show a significant improvement over methods that do not utilize the smoothness assumption and classify each point separately from the others.**

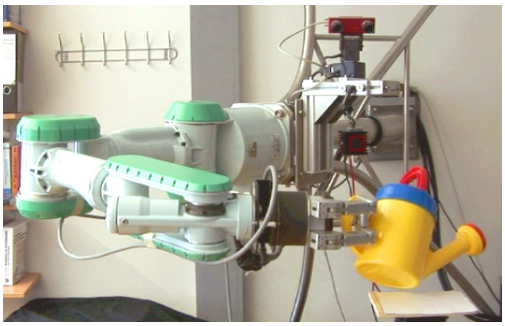

Fig. 1. Barrett hand equipped with a SwissRanger time-of-flight camera

I. INTRODUCTION

A wide range of domestic tasks for service robots are based on object manipulation. Examples include collecting objects, loading or unloading a dishwasher, and opening doors. These tasks require the robot to localize objects and to efficiently grasp them. Grasping is therefore one of the most fundamental problems in robotics. This problem is particularly challenging in unstructured environments, since even familiar types of objects contain a range of shapes and sizes. Despite these variations, humans can learn how to grasp objects from a small number of examples, and efficiently generalize the learned skills to grasping novel objects. In this paper, we propose a probabilistic framework for teaching an autonomous robot to grasp new objects. The robot is equipped with a 3-D vision system, such as a Kinect or a time-of-flight camera.

Given the importance of grasping for robots, a variety of approaches have been proposed [1]. Until the last decade, most of these techniques relied on complete and accurate 3-D models of the objects, in order to apply analytical methods from mechanics [2]. Building accurate models for new objects is difficult and often requires laser scanning the objects. Additionally, surface properties, such as friction and compliance are essential for these approaches. However, these properties are not easy to measure, and are often modelled as being uniform for a whole object.

An alternative approach is the use of statistical methods for learning to grasp. These methods have received increased attention in recent years [3]–[6]. For example, de Granville et al. [3] explored the problem of representing the orientation of a hand as it approaches an object, and demonstrated the feasibility of extracting canonical grasps from a human demonstration. Canonical grasps were represented using clustering based on mixture distributions. Another approach [6] consists of combining analytical and empirical methods by segmenting an object into a set of superquadratics and then learning which ones are more suitable for grasping.

Vision-based methods have also been widely explored. Earlier work on grasping using vision was based on modeling an object as a set of primitive shapes, such as spheres, cylinders, cones and boxes, and then using a set of rules for generating grasp positions and orientations [7]. Pelossof et al [8] used an SVM to learn a grasp quality measure, where the grasping parameters correspond to the degrees of freedom of a hand. Rao et al. [9] used 3-D scan data points of a given object in order to segment it, and then used a classifier to select only graspable segments based on their color and geometric features.

Saxena et al. [4] also showed that machine learning methods can be successfully applied to grasp novel objects. More specifically, they use 2-D images of the same object taken from different angles and learn a logistic function that predicts the position of a good grasping point based on its visual features. A grasping point is defined as a small region on the surface of the object that a human, using a two-fingered pinch grasp, would choose to grasp it. Recently, Jiang et al. [10] showed how to learn a *grasp rectangle* from 3-D images and use it for estimating a full 7-dimensional gripper configuration. This grasp rectangle is defined by taking into account features of both the object and the gripper. Another vision-based approach [6] first segments the object using the gaussian curvature as an indicator of the separation points. The segments are approximated with superquadratic primitive shapes. A neural network is then trained to learn which segments can be used for grasping.

Abdeslam Boularias, Oliver Kroemer and Jan Peters are with the Max-Planck Institute for Intelligent Systems in Tübingen, Germany. Oliver Kroemer and Jan Peters are also affiliated with the Technische Universitaet Darmstadt, Intelligent Autonomous Systems Group, Darmstadt, Germany. {abdeslam.boularias,oliverkro,jan.peters}@tuebingen.mpg.de
The authors contributed equally to this work.

Along the same line, Bohg and Kragic [11] have developed a method for detecting a grasping point on an object by analyzing it in a monocular image and reconstructing the 3-D representation based on a stereo view. They applied shape context as a visual feature descriptor that relates the global shape of an object to a point. An SVM is then used for classifying the points.

A common drawback of these methods is that they learn a function that assigns to each point a label (class), independently from the labels assigned to its neighboring points. While this approach is computationally efficient, the high quality of the produced results requires that the features describing the points are sufficiently informative about the corresponding class. Unfortunately, this crucial assumption is not generally valid. For example, points on different parts of an object may have similar features. Noise in the extracted features can also result in labels that are not locally consistent. However, such problems can be efficiently solved using Markov Random Fields (MRFs) for enforcing neighbor points to have the same label [12]–[15]. Therefore, we consider in this paper a class of MRFs, called Associative Markov Networks (AMNs), for learning to grasp.

Given a 3-D point cloud describing the surface of the object to be grasped, we use an AMN to represent the probability distribution on the possible assignments of labels (good grasping point, bad grasping point) to the different points. The AMN is trained by maximizing the likelihood of positive and negative examples on one or many objects. The object is then grasped by positioning the center of the end-effector on a positively labeled point.

The remainder of this paper is structured into three sections. In the next, the general MRF and AMN models are presented. In Section II, we show how the AMN model is used for learning grasping positions. Section III presents the empirical performance of our approach, evaluated on the GraspIt! simulator, as well as on a Mitsubishi PA-10 with a Barrett Hand, and a SwissRanger time-of-flight camera, as shown in Figure 1.

II. BACKGROUND

A Markov Random Field (MRF) is a graphical model widely used for representing joint probability distributions. The MRF defines a probability distribution over N discrete variables $Y = \{y_1, \ldots, y_n\}$. Each variable corresponds to the label of a vertex in a graph $(\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. Each node is assigned to a label y_i from a set of possible labels $\mathcal{L}$. Therefore, the MRF defines a probability distribution over $\mathcal{L}^N$. In our case, the vertices of the graph are the 3-D scan points of an object, and the edges connect each point to its k-nearest neighbors. The label y_i indicates whether grasping the object at point i is successful or not, thus $y_i \in \{+1, -1\}$, where $y_i = +1$ implies that i is a good point for grasping the object and $y_i = -1$ implies that i is a bad grasping point.

Performing inference on general graphical models is NP-hard due to the exponential output space of possible solutions [16]. Therefore, we focus on a particular tractable class of MRFs called pairwise Markov networks, where potentials ϕ_i and $\phi_{i,j}$ are associated with each node $i \in \mathcal{V}$ and each edge $(i,j) \in \mathcal{E}$. A node potential $\phi_i(y_i)$ is a non-negative real number that indicates the preference of labeling the node i with the label y_i independently from the other nodes, while an edge potential $\phi_{i,j}(y_i, y_j)$ indicates the preference of labeling the neighbor nodes i and j with the labels y_i and y_j simultaneously. The joint distribution on the labels $Y = (y_1, \ldots, y_n)$ is given by

$$P(Y) = \frac{1}{Z} \prod_{i \in \mathcal{V}} \phi_i(y_i) \prod_{(i,j) \in \mathcal{E}} \phi_{i,j}(y_i, y_j),$$

where Z is a partition function defined as

$$Z = \sum_{Y' \in \mathcal{L}^N} \prod_{i \in \mathcal{V}} \phi_i(y'_i) \prod_{(i,j) \in \mathcal{E}} \phi_{i,j}(y'_i, y'_j). \quad (1)$$

Furthermore, we restrict our model to a class of pairwise MRFs known as Associative Markov Network (AMN) described by Taskar et al. [17]. In this model, edges connecting points that have different labels are associated with a constant potential equal to 1. In other terms, we set $\phi_{i,j}(y_i, y_j) = 1$ if $y_i \neq y_j$. Consequently, only the potentials of edges connecting points with the same labels are considered in this model, they are given by $\phi_{i,j}(y)$ where $y_i = y_j = y$.

We define $X = \{x_i, x_{ij}\}$ to be the features extracted from the 3-D point cloud, where $x_i \in \mathbb{R}^{d_n}$ is the vector of features that describes the grasp at point i, and $x_{ij} \in \mathbb{R}^{d_e}$ is the vector of features for edge (i,j), which describes the relation between the grasps performed at points i and j. The AMN model uses the log-linear function for representing a potential as a function of the features, i.e. $\log \phi_i(y) = w_{n,y}^T x_i$ and $\log \phi_{ij}(y) = w_{e,y}^T x_{ij}$, where $w_{n,y} \in \mathbb{R}^{d_n}$ are the weights used for nodes that are labeled by y, and $w_{e,y} \in \mathbb{R}^{d_e}$ are the weights used for adjacent nodes that are both labeled by y. This formulation assumes that all the nodes share the same weights and all the edges share the same weights as well. The log-probability of the joint distribution on the labels is

$$\log P(Y) = \sum_{i \in \mathcal{V}} w_{n,y_i}^T x_i + \sum_{\substack{(i,j) \in \mathcal{E} \\ \text{s.t. } y_i = y_j}} w_{e,y_i}^T x_{ij} - \log Z_w(X),$$

where $Z_w(X)$ is the partition function from Equation 1.

Since y_i is binary, we indicate by w_n the weights of labeling the point i as a good grasp point, and use $-w_n$ as the weights of labeling the point i as a bad grasp point. We also use a unique edge weight vector w_e for all the labels. Thus, the importance of having similar labels for adjacent points is the same for successful and unsuccessful grasps.

III. LEARNING GRASPS WITH MARKOV RANDOM FIELDS

In this section, we show how to use Markov Random Fields for learning to find good grasping points. First, we show how to learn the parameters w_n and w_e using examples of good and bad grasping points, manually provided by a human (Figure 2). Then, we show how to infer the positions of the good grasping points on a given new object.

A. Learning the parameters of the model

We provide the learning algorithm with a set of positive and negative grasp examples performed on one or many objects. To simplify the notations, we consider only one training object represented by a graph $(\mathcal{V}^{training}, \mathcal{E}^{training})$, where $\mathcal{V}^{training}$ and $\mathcal{E}^{training}$ are restricted to the labeled points and the edges connecting them.

From the Representer Theorem [18], [19], we know that the parameters w_n and w_e that maximize the log-likelihood are given by a linear function of the provided examples

$$\begin{aligned} w_n &= \sum_{i \in \mathcal{V}^{training}} \theta_n(i) x_i, \\ w_e &= \sum_{\substack{(i,j) \in \mathcal{E}^{training} \\ \text{s.t. } y_i = y_j}} \theta_e(i,j) x_{ij}, \end{aligned}$$

where $\theta_n(i), \theta_e(i,j) \in \mathbb{R}$.

Therefore, the log-probability of the joint distribution on the labels for an arbitrary new object, represented by a graph $(\mathcal{V}, \mathcal{E})$, is given by

$$\begin{aligned} \log P(Y) = &\sum_{i' \in \mathcal{V}} \sum_{i \in \mathcal{V}^{training}} y_{i'} \theta_n(i) k_n(i,i') \\ &+ \sum_{\substack{(i',j') \in \mathcal{E} \\ y_{i'} = y_{j'}}} \sum_{\substack{(i,j) \in \mathcal{E}^{training} \\ y_i = y_j}} \theta_e(i,j) k_e(\langle i,j \rangle, \langle i',j' \rangle) \\ &- \log Z_\theta(X), \end{aligned} \tag{2}$$

where k_n and k_e are kernel functions used for measuring similarities between the points and the edges respectively, they are defined as $k_n(i,i') = x_i^T x_{i'}$, and $k_e(\langle i,j \rangle, \langle i',j' \rangle) = (x_{ij})^T x_{i'j'}$.

In the training phase, the graph $(\mathcal{V}, \mathcal{E})$ is replaced by $(\mathcal{V}^{training}, \mathcal{E}^{training})$, and Equation (2) defines the log-likelihood of the examples as a concave function of the parameters θ_n and θ_e, which is then maximized using the BFGS optimization method. Given that the examples usually cover separate subsets of each object, we decompose each graph $(\mathcal{V}^t, \mathcal{E}^t)$ into its weakly connected components (components where a path exists between every two points), and separately calculate the function $Z_\theta(X_c^t)$ of each component $c \in \mathcal{C}^t$.

B. Finding suitable points for grasping new objects

Given the learned parameters θ_n and θ_e and a new object, we want to find the labels $Y = (y_1, \ldots, y_n)$ that have the highest probability (Equation (2)). This problem, called the inference problem, can be solved in a polynomial time using techniques such as graph min-cut [20] and linear programming relaxation [17] when the labels y_i are binary, which is the case in this paper. We adopt the graph min-cut approach. For more details on how to transform the inference problem to a min-cut one, we refer the reader to Ben Taskar, 2004 [19]. Finally, one grasping point is uniformly sampled from the set of valid grasp points.

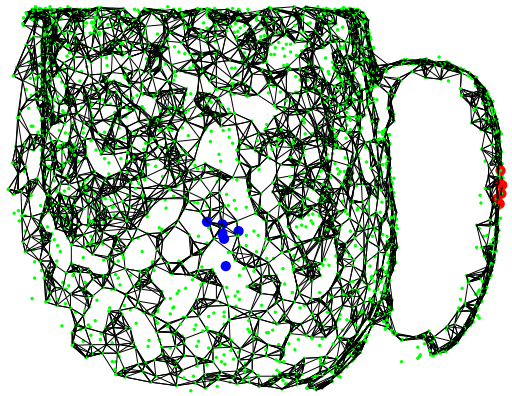

Fig. 2. A point cloud of the object used for training. A total of 12 points are manually labeled. Red points indicate possibly good grasp locations, while blue points indicate likely bad grasp positions. Notice that the Markov network covers only a randomly chosen subset of the points.

IV. EXPERIMENTS

In this section, we present our experiments on learning autonomous grasps. We compare the Markov network to a logistic regression model, which is obtained by removing the edges and training the model using only the node features. A quadratic kernel (the squared dot product of the feature vectors) is used as a similarity measure in Equation (2).

A. Feature extraction

The features, calculated from a 3-D point cloud, should contain sufficient information for measuring the similarity, in terms of success or failure, between grasps performed on different points. A grasp is specified by the center of the hand during the closure and the direction of the hand. We use the surface normal vector at the grasp point to indicate the direction of the hand. The success of this particular type of grasps depends on the local geometry of the object around the center of the hand. We use two types of features to capture this information. The first one, that we call the *density feature*, roughly indicates the shape of the object around a given point. For each point i, we consider the points of the object that are inside the three balls $b_{i,1}, b_{i,2}$ and $b_{i,3}$ centered at i with radiuses $r_1 \leq r_2 \leq r_3$. We then compute the percentage of the points lying inside b_1, between b_1 and b_2, and between b_2 and b_3. These three features are mainly used for distinguishing handle-like shaped parts from the others. The second type of features are called the *spectral features*, they are commonly used in spectral analysis of point clouds [13]. For each point i, we compute the scatter matrix formed by the 3-D coordinates of the points inside the ball $b_{i,3}$. We define $\lambda_3 \geq \lambda_2 \geq \lambda_1$ to be the eigenvalues of the scatter matrix. The spectral features correspond then to λ_1, $\lambda_2 - \lambda_1$ and $\lambda_3 - \lambda_2$. Finally, we normalize the spectral features $\lambda_2 - \lambda_1$ and $\lambda_3 - \lambda_2$ by dividing them by the maximal one. We additionally consider a bias feature of 1.0 for all the points. For the edge features, we noticed that the best performance of our approach was obtained by simply using a constant feature $x_{i,j} = 1.0$ for all the edges.

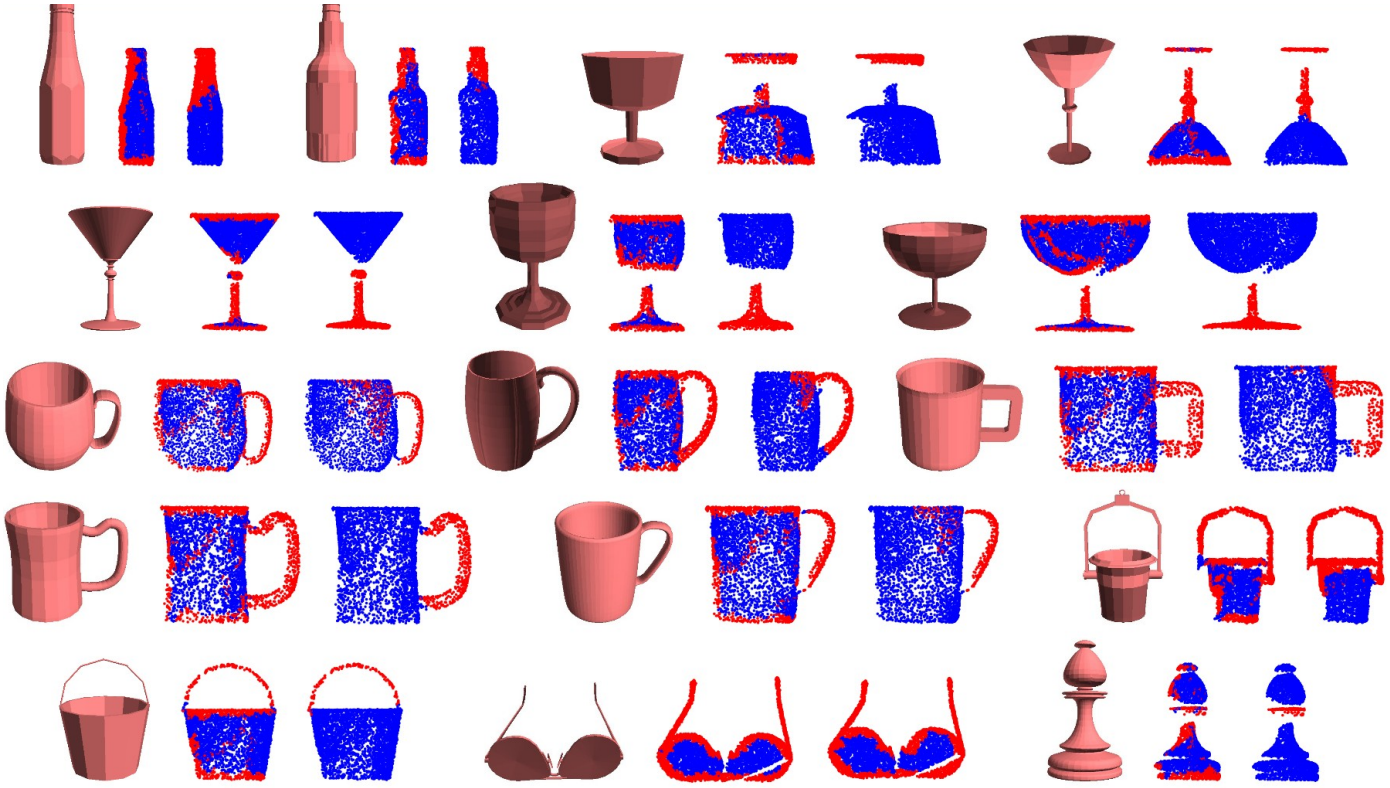

Fig. 3. Objects used for testing. Points in red indicate predictions of good grasp positions, and those in blue indicate predictions of bad positions for precision grasps. For each object, the middle figure corresponds to the results obtained using the logistic regression, and the right figure is the results of the associative Markov network.

B. Princeton Shape Benchmark

The first experiment consists in using 3-D models of objects, taken from the *Princeton Shape Benchmark* [21], for evaluating the performance of our approach. We simulated 3-D scans of the provided models by placing a virtual camera in the scene, and uniformly sampling ≈ 6000 points from the visible surface of the object. A gaussian white noise of variance 4×10^{-6} was added to the coordinates of each point.

While all the sampled points are used for calculating the features, only 20% of them are used as nodes of the Markov network. The degree of the Markov networks is set to six, i.e. each point is connected to its six nearest neighbors. We found out that a smaller degree leads to a weak propagation of information between the points, while in some cases, a higher degree results in forcing distant points to have the same label. The radius r_1 is set to 1 cm for all the objects, r_2 is set to $1.1 \times r_1$ and $r_3 = 1.1 \times r_2$.

Given the small number of features that we used, a small number of examples was sufficient for training the Markov network. Figure 2 shows an object used for the training. Six points on the handle are manually labeled as positive grasp examples, whereas six other points on the surface of the cup are labeled as negative grasp examples. The proposed approach therefore does not require large amounts of labeled data. In fact, the complexity of the inference algorithm is polynomial in the number of examples.

Figure 3 shows the results of classifying points of different objects. As expected, the AMN classifies mostly only the points on handle-like shaped parts as positive grasp points. In some cases, the AMN misclassifies points that are close to the handle. One possible solution to this problem is to include the curvature of the surface in the features of the edges, such that neighbor points that are located on different parts of the object are not penalized for having different labels. The majority of the points that are misclassified by the logistic regression are located on parts that locally look like handles. These parts often correspond to the line between the visible and the occluded surfaces of the object. This problem does not occur often in the AMN approach because the points located between the visible and the occluded surfaces have neighbors on the visible surface, and they are forced to have a similar label as their neighbors. The runtime of the AMN classifier is not significantly higher than the runtime of the logistic regression. The average runtime of classifying all the points of an object, using the objects in Figure 3, is 56.3 ± 27.9 seconds for AMNs compared to 50.6 ± 25.0 seconds for the logistic regression (using Matlab).

C. Evaluations with the GraspIt! simulator

GraspIt! is an interactive grasping simulator that can import a wide variety of hand and object models and can evaluate the grasps formed by these hands [22]. We used it to evaluate grasping different objects from the Princeton Shape Benchmark, using the results of the classification process described in the previous subsection.

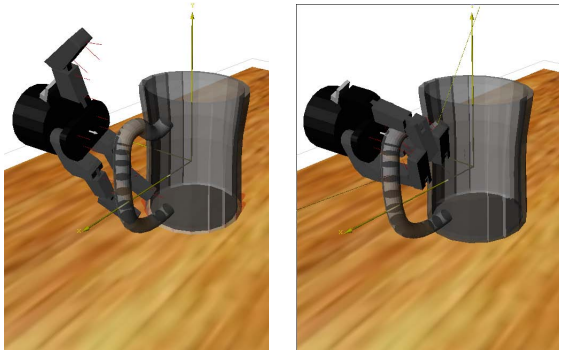

Fig. 4. Grasping a cup around a point suggested by the Markov network

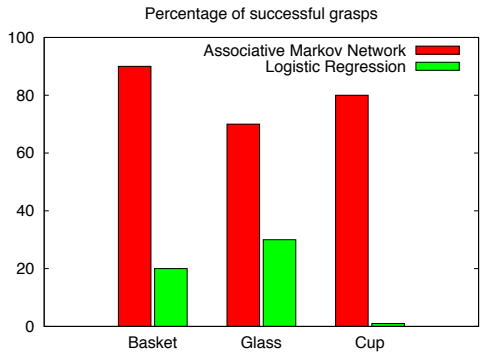

Fig. 5. Percentage of successful grasps using the GraspIt! simulator

We use the Barrett model for the hand, and define a grasp by the parameters $(i, \vec{d}, \vec{o})$. The first parameter, i, is a 3-D point on the surface of the object. The vector $\vec{d}$ specifies the direction from the center of the hand to the grasping point. The vector $\vec{o}$ indicates the orientation of the thumb with the hand fully open. The grasping action itself is a basic closing of the hand, without grasp-specific preshaping (Figure 4).

In our experiments, the grasping points are uniformly sampled from those classified as good grasping points. The direction vector is the opposite of the surface normal at the grasping point. The orientation of the thumb corresponds to the normal vector of the plan formed by the direction vector and the first eigenvector of the scatter matrix, defined by the points surrounding the grasping point. The only parameter that we needed to hand-tune for each grasp was the magnitude of the direction vector, which defines the distance between the center of the hand and the grasping point.

Figure 5 shows the success rate of grasping three objects, using 10 grasping points for each object. A grasp is considered as successful if the object does not fall after lifting it from the table. Most of the points suggested by the Markov Network are located on the handle-like shaped parts, and they correspond to good positions for grabbing the objects, which explains the improved performance of this method. For instance, none of the points sampled by the logistic regression was on the handle of the cup (the third object), and the orientation of the thumb was mostly in a wrong direction, leading to a zero success rate. This is mainly due to two factors. First, the eigenvectors associated to a point located on a flat surface (in contrast to a handle) are not good indicators of the hand orientation. Second, the object tends to move away from the hand when one finger starts pushing it before the others.

D. Experiments with the Barrett hand

We empirically evaluated the proposed method on a Mitsubishi PA-10 with a 3-finger Barrett hand, equipped with a SwissRanger time-of-flight camera. As with the GraspIt! simulator, we define a grasp by a point on the surface of the object, a direction vector, and the orientation of the thumb with the hand fully open. The points are uniformly sampled from those classified as good grasping points. The direction and the orientation are computed from the surface normal

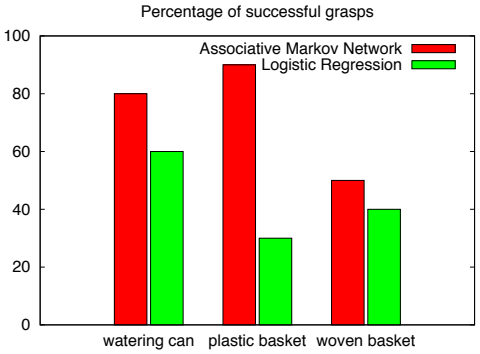

Fig. 6. Percentage of successful grasps using a Mitsubishi PA-10 with a Barrett hand, equipped with a SwissRanger time-of-flight camera.

and the eigenvectors of the point cloud, as in the GraspIt! experiment. The distance between the center of the hand and the grasping point in the pre-shape configuration (before closing the hand) was however kept fixed for all the objects. Moreover, the parameters used for computing the features of the objects are the same as those used for the objects from the Princeton Shape Benchmark. Therefore, no parameters have been hand-tuned in this experiment.

Figure 6 shows the success rate of grasping three objects, using 10 grasps for each object. A grasp is considered as successful only if the object does not fall after being lifted. The objects are a watering can, a plastic basket and a woven basket (Figure 7). As in the GraspIt! experiment, the Markov network method achieved a higher success rate than the logistic regression. This is due to the fact that the Markov network more consistently located handle-like shaped parts of the objects. Grasping objects is often easier when performed on the handles, especially for objects made from a material with a low friction factor, such as metal or plastic. The logistic regression found the handles, but also unsuitable locations due to edge artifacts.

The third object was particularly challenging for both methods. This object is made from a compliant material, consequently, the position of the handle was changing during the grasp. This object has also relatively small handles, and the distance of the hand from the handle was not adjusted. Nevertheless, the performance of our approach was slightly higher than the logistic regression in this case too. This object

Fig. 7. The objects used for grasping in the robot experiments

demonstrates that the approach is applicable to objects that have a different shape each time they are grasped.

V. CONCLUSION AND FUTURE WORK

We have presented a probabilistic approach for predicting the success of a grasp performed at a given point on the surface of an object. The objects are represented by 3-D point clouds, taken with a time-of-flight camera. These points are usually noisy and provide only a partial view of the object. Moreover, the features of the points taken separately do not provide sufficient information about their suitability for grasping the object. For example, points on different parts of the same object may have similar features. Therefore, we have used a graphical model, known as Associative Markov Network, for classifying the points. We have experimentally evaluated this method in three settings. In the first, we used the Princeton Shape database for training the Markov network and visualizing the classified point clouds of a range of objects. We compared the results of the Markov network with those of logistic regression. The results show that the Markov network is more robust to the noise and to the partial view of the object. In the second experiment, we used the GraspIt! simulator to measure the efficiency of grasping the objects from the points that have been suggested as good grasp points in the first experiment. In the last experiment, we used a robot equipped with a Barrett hand and a SwissRanger time-of-flight camera to evaluate the results of the classification. Here again, the results show the power of graphical models in detecting graspable parts of a novel object, based on simple features and only a few examples of successful and unsuccessful grasps.

However, the experiments on the robot show that most of the failed grasps were caused by inadequate orientation of the fingers, distance between the hand and the object during the grasp, and direction of approaching the grasping point. Therefore, these parameters should also be learned in a future work, although it is not yet clear if they should be learned jointly with the positions of the grasping points, or post-hoc, based on the regions found using the method proposed in this paper. Another cause of the failures was the false handles generated by self-occlusion. This problem can be solved with a Bayesian approach where the uncertainty on the positions of the points is explicitly modeled, and virtual nodes corresponding to possible hidden points are added to the Markov network.

REFERENCES

[1] Antonio Bicchi and Vijay Kumar. Robotic Grasping and Contact: A Review. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'00)*, 2000.

[2] Karun Shimoga. Robot Grasp Synthesis Algorithms: A Survey. *The International Journal of Robotics Research*, 15:230–266, 1996.

[3] Charles de Granville, Joshua Southerland, and Andrew Fagg. Learning Grasp Affordances Through Human Demonstration. In *Proceedings of the International Conference on Development and Learning*, 2006.

[4] Ashutosh Saxena, Lawson Wong, and Andrew Ng. Learning grasp strategies with partial shape information. In *Proceedings of the 23rd national conference on Artificial intelligence (AAAI'08)*.

[5] Oliver Kroemer, Renaud Detry, Justus Piater, and Jan Peters. Combining Active Learning and Reactive Control for Robot Grasping. *Robotics and Autonomous Systems*, 58:1105–1116, 2010.

[6] Sahar El-Khoury and Anis Sahbani. A New Strategy Combining Empirical and Analytical Approaches for Grasping Unknown 3D Objects. *Robotics and Autonomous Systems*, 58:497–507, 2010.

[7] Andrew T. Miller, Steffen Knoop, Henrik I. Christensen, and Peter K. Allen. Automatic Grasp Planning Using Shape Primitives. In *Proceedings of the 2003 IEEE international conference on Robotics and Automation (ICRA'03)*, pages 1824–1829, 2003.

[8] Rafael Pelossof, Andrew Miller, Peter Allen, and Tony Jebara. An SVM Learning Approach to Robotic Grasping. In *Proceedings of the 2004 IEEE international conference on Robotics and Automation (ICRA'04)*, pages 3512–3518.

[9] Deepak Rao, Quoc V. Le, Thanathorn Phoka, Morgan Quigley, Attawith Sudsang, and Andrew Y. Ng. Grasping Novel Objects with Depth Segmentation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'10)*, 2010.

[10] Yun Jiang Stephen Moseson and Ashutosh Saxena. Learning grasp strategies with partial shape information. In *Proceedings of the 2011 IEEE international conference on Robotics and Automation (ICRA'11)*.

[11] Jeannette Bohg and Danica Kragic. Learning Grasping Points with Shape Context. *Robotics and Autonomous Systems*, 58:362–377, 2010.

[12] Dragomir Anguelov, Ben Taskar, Vassil Chatalbashev, Daphne Koller, Dinkar Gupta, Geremy Heitz, and Andrew Ng. Discriminative learning of Markov random fields for segmentation of 3d scan data. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR'05)*, pages 169–176, 2005.

[13] Daniel Munoz, Nicolas Vandapel, and Martial Hebert. Onboard contextual classification of 3-D point clouds with learned high-order Markov random fields. In *Proceedings of the 2009 IEEE international conference on Robotics and Automation (ICRA'09)*, 2009.

[14] Pushmeet Kohli, Pawan Kumar, and Philip Torr. P3 and beyond: Solving energies with higher order cliques. In *IEEE International Conference on Computer Vision and Pattern Recognition (ICCVPR'07)*, 2007.

[15] Nathan Ratliff, Drew Bagnell, and Martin Zinkevich. Online subgradient methods for structured prediction. In *In Eleventh International Conference on Artificial Intelligence and Statistics (AISTATS'07)*, 2007.

[16] Venkat Chandrasekaran, Nathan Srebro, and Prahladh Harsha. Complexity of inference in graphical models. In *Proceedings of the Proceedings of the Twenty-Fourth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI'08)*, pages 70–78, 2008.

[17] Ben Taskar, Vassil Chatalbashev, and Daphne Koller. Learning associative markov networks. In *Proceedings of the Twenty-First International Conference on Machine Learning (ICML'04)*, 2004.

[18] Yasemin Altun, Alex J. Smola, and Thomas Hofmann. Exponential Families for Conditional Random Fields. In *Proceedings of the 20th Annual Conference on Uncertainty in Artificial Intelligence (UAI'04)*, pages 2–9, 2004.

[19] Ben Taskar. *Learning Structured Prediction Models: A Large Margin Approach*. PhD thesis, Stanford University, CA, USA, 2004.

[20] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:2001, 1999.

[21] Michael Kazhdan Philip Shilane, Patrick Min and Thomas Funkhouser. The Princeton Shape Benchmark. *Shape Modeling International*, Genova, Italy, June 2004. http://shape.cs.princeton.edu/benchmark/.

[22] Andrew Miller and Peter K. Allen. Graspit!: A Versatile Simulator for Robotic Grasping. *IEEE Robotics and Automation Magazine*, 11:110–122, 2004. http://www.cs.columbia.edu/ cmatei/graspit/.