Efficient Sampling for Grasp Quality Evaluation with Gaussian Process Implicit Surface Representation

Michael Laskey, Zoe McCarthy, Florian T. Pokorny, Jeff Mahler, Sachin Patil,
Danica Kragic, Jur Van Den Berg, Pieter Abbeel, and Ken Goldberg

I. Introduction

A robot in the wild is expected to encounter an object, grasp and manipulate it. In order to do so it must use its noisy sensor data to build a model of the object, for which to plan grasps on. While most grasp planning algorithms currently assume a polygonal mesh [**?**], their is no easy way to handle the uncertainty associated with partial visibility, sensor noise and surface properties such as specularity and transparency. We thus propose a Bayesian approach to this problem to formally go from point cloud to an uncertainty representation using Gaussian Processes.

An alternative to polygonal mesh representation is the signed distance field, which is zero-valued at the object surface, positive valued outside the object, and negative valued in the object interior [1]. We look at encoding uncertainty into this model with a Bayesian representation known as Gaussian Process Implicit Surface (GPIS). The key idea behind GPIS is to represent shape uncertainty as a distribution overall possible SDFs that could represent the shape. For example, GPIS-based models have large variance where data is known to be unreliable (e.g. occlusions, specularities) and lower variance where measurements are more reliable.

While a GPIS is a formal method for representation of uncertainty and can offer continuous function that can be used for local grasp optimization [**?**]. Evaluation of grasp quality through techniques like Monte-Carlo Integration on a GPIS requires an exhaustive sampling of SDFs, which is extremely computationally expensive and not suited for a real time application. We purpose an alternative to sampling SDFs by computing a distribution on explicit grasp parameters and performing Monte-Carlo integration on that instead.

We assume that a robot's m grippers approach the grasped object along straight lines (which we call lines of action), and analyze how this grasping model interacts with the probability distribution on shapes described by a Gaussian Process Implicit Surface (GPIS). We calculate and visualize the theoretical probability distributions on grasp contact points and normals and the expected center of mass induced by the GPIS and line of action. Then demonstrate how sampling from this explicit grasp distribution can be used to reduce the complexity associated with Monte-Carlo integration of the expected grasp quality over the distribution of shapes.

To further improve the efficiency of grasp quality evaluation, for a given set of grasps. We demonstrate that formulating the problem as an n-arm bandit problem allows us to formulate a policy that intelligently select which grasps to sample next. N-arm bandit problems look at the situation of an agent being presented with N slot machines, where each slot machine gives a reward from some unknown distribution. The goal of the agent is to efficiently explore the slot machines and then maximize reward by exploiting the best one. In our problem each lever pull is an evaluation of a grasp metric (e.g. Ferrari-Canny), which is computationally expensive. Thus finding a policy to minimize evaluations is essential for real time applications.

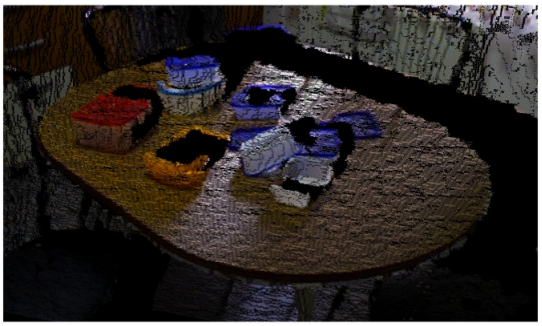

Fig. 1: An example of the noise from a Kinect-like sensor [TODO: GET AN ACTUAL KINECT IMAGE]

II. Related Work

Many different methods for representing objects exist, most of which were developed in the graphics community for making rendering and simulations computationally tractable; see [2] for an overview. Polygonal mesh models are the most common representation in robotics but these models usual assume perfectly known geometry. Extension of mesh models to include uncertainty generally assume independent noise on the vertex locations, ignoring noise correlation [3]. As an alternative, implicit surface representations are becoming increasingly popular in robotics due to their success in real-time surface modeling from range images [4],[1], [5]. These representation include a confidence estimate based on a sensor noise model, but they cannot estimate the location in the surface regions where no measurements were obtained or represent correlations between voxels [4].

Choosing grasps given a surface representation of an object usually focuses on improving grasps for a object shape [6]. Past work on grasping has focused on state uncertainty [7][**?**] [8], uncertainty in object pose [9] [10] [**?**]

or uncertainty in the location of contact with the object [11]. Robust grasp selection for parallel jaw grippers on uncertain shapes was studied by Kehoe et al. for both zero-slip push grasps [3] and slip push grasps [12] using Monte Carlo integration for the expected grasp quality, but this assumed extruded polygons with independent noise at each vertex. Christopoulus et al. sampled spline fits for 2-dimensional planar objects to measure the quality of uncertainity, but did not have a formal method for representing uncertainity and only calculated the probability of achieving force closure. .

The choice of using a Gaussian Process Implicit Surface to represent our uncertainty, stems from the fact that it provides a formal way to include various sources of noise in observations. Prior work has used uncertainty representation such as independent Gaussian noise on each vertex in a Polygonal mesh [3]. However, this doesn't provide a continuous function one can easily compute distributions on grasp parameters with. GPIS is also becoming more prevalent in the grasping community with recent work by Dragiev et al. [13] . The utilization of uncertainity around an implicit surface allows us to compute explicit distributions on contacts and surface normals, which to our knowledge has not been done formally.

[TODO: WRITE ABOUT N-ARM BANDIT LITERATURE]

III. Gaussian Process Implicit Surfaces

In this section we describe the mathematical derivation of Gaussian Process Implicit Surface representations.

A. Gaussian Process (GP) Background

Gaussian processes (GPs) are widely used in machine learning as a nonparametric regression method for estimating continuous functions from sparse and noisy data [14]. In a GP, a training set consists of input vectors $\mathcal{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^d$, and corresponding observations $\mathbf{y} = \{y_1, \ldots, y_n\}$. The observations are assumed to be noisy measurements from the unknown target function f:

$$y_i = f(\mathbf{x}_i) + \epsilon, \tag{1}$$

where $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is Gaussian noise in the observations. A zero-mean Gaussian process is completely specified by a covariance function $k(\cdot, \cdot)$, also referred to as a kernel. Given the training data $\mathcal{D} = \{\mathcal{X}, \mathbf{y}\}$ and covariance function $k(\cdot, \cdot)$, the posterior density $p(f_*|\mathbf{x}_*, \mathcal{D})$ at a test point $\mathbf{x}_*$ is shown to be [14]:

$$p(f_*|\mathbf{x}_*, \mathcal{D}) \sim \mathcal{N}\big(\mu(\mathbf{x}_*), \Sigma(\mathbf{x}_*)\big) \tag{2}$$

$$\mu(\mathbf{x}_*) = k(\mathcal{X}, \mathbf{x}_*)^\intercal (K + \sigma^2 I)^{-1} \mathbf{y} \tag{3}$$

$$\Sigma(\mathbf{x}_*) = k(\mathbf{x}_*, \mathbf{x}_*) - k(\mathcal{X}, \mathbf{x}_*)^\intercal (K + \sigma^2 I)^{-1} k(\mathcal{X}, \mathbf{x}_*)\big) \tag{4}$$

where $K \in \mathbb{R}^{n \times n}$ is a matrix with entries $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j)$ and $k(\mathcal{X}, \mathbf{x}_*) = [k(\mathbf{x}_1, \mathbf{x}_*), \ldots, k(\mathbf{x}_n, \mathbf{x}_*)]^\intercal$. This derivation can also be used to predict the mean and variance of the function gradient by extending the kernel matrices using the identities [15]:

$$\operatorname{cov}\left(f(\mathbf{x}_i), f(\mathbf{x}_j)\right) = k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\operatorname{cov}\left(\frac{\partial f(\mathbf{x}_i)}{\partial x_k}, f(\mathbf{x}_j)\right) = \frac{\partial}{\partial x_k} k(\mathbf{x}_i, \mathbf{x}_j)$$

$$\operatorname{cov}\left(\frac{\partial f(\mathbf{x}_i)}{\partial x_k}, \frac{\partial f(\mathbf{x}_j)}{\partial x_l}\right) = \frac{\partial^2}{\partial x_k \partial x_l} k(\mathbf{x}_i, \mathbf{x}_j)$$

B. Kernel Selection

The choice of kernel is application-specific, since the function $k(\mathbf{x}_i, \mathbf{x}_j)$ is used as a measure of correlation between states $\mathbf{x}_i$ and $\mathbf{x}_j$. A common choice is the squared exponential kernel:

$$k(\mathbf{x}_i, \mathbf{x}_j) = \nu^2 \exp(-\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^\intercal \Lambda^{-1} (\mathbf{x}_i - \mathbf{x}_j)) \tag{5}$$

where $\Lambda = \operatorname{diag}(\lambda_1^2, \ldots, \lambda_d^2)$ are the characteristic length scales of each dimension of $\mathbf{x}$ and ν^2 describes the variability of f. Other common kernels relevant to GPIS are the thin-plate splines kernel [16] and the Matern kernel [17].

The measurement noise parameter σ is usually estimated based on a model of the sensor used to collect measurements The vector of remaining hyperparameters $\boldsymbol{\theta} = \{\nu, \lambda_1, \ldots, \lambda_d\}$ is optimized during the training process by maximizing the log-likelihood $p(\mathbf{y}|\mathcal{X}, \boldsymbol{\theta})$, usually using gradient descent [14]. The log-likelihood function is subject to local maxima, and therefore the hyperparamter search often involves a search over the maxima found from several random initializations of the hyperparamters.

Fig. 2: Left: A surface represented as a Truncated Signed Distance Field (TSDF). Right: A GPIS reconstruction from noisy samples of the left surface's TSDF.

IV. Problem Definition

We assume a bounded 2-d rectangular workspace $\mathcal{R}$. We parameterize a single grasp g on an object with the following tuple $g = (\mathbf{c}_1, ..., \mathbf{c}_m, \mathbf{n}_1, ..., \mathbf{n}_m, \mathbf{z}, \tau)$. We have an indexing set I of m point contacts and surface normal on the object: for $i \in I$ the contact is located at c_i with surface normal n_i. The object has a center of mass z and friction coefficient τ. The line segment $\gamma(\cdot)$ has endpoints a, b that are defined as the start of the gripper and the intersection of the line with the end of the workspace respectively, as shown in Fig. 3.

We assume that the robot grippers move along the line of action until they make contact with the object and stop. Since we have uncertainty in the shape this induces a distribution on the grasp parameters subject to the line, hence $p(g) = (p(\mathbf{c}_1), ..., p(\mathbf{c}_m), p(\mathbf{n}_1), ..., p(\mathbf{n}_m), \bar{\mathbf{z}}, \tau)$. We note that τ is considered known and $\bar{z}$ is the expected center of mass, not a full probability distribution. In section V, we demonstrate how to efficiently compute the distributions on contact points and surface normals and expected center of mass.

V. Distribution of Grasp Parameters

For the following derivations we introduce the following, $\theta(x) = (\mu(x), \Sigma(x))$, where $\theta(x)$ is a tuple consisting of the mean and covariance functions given by the trained GPIS model [18] .

To calculate $p(g)$, we assume a gripper contacts approaches along a parameterized line of action, or a 1-dimensional curve in the work space, defined by $\gamma(t)$. See Fig 3, for a detailed illustration. Each gripper contact is defined by a line of action, so we assume the following tuple is provided $\Gamma = (\gamma_1(\cdot), ..., \gamma_m(\cdot))$, these approach trajectories are then used to compute a distribution on grasp parameters.

[TODO: GET A BETTER FIGURE FOR LINE OF ACTION MODEL]

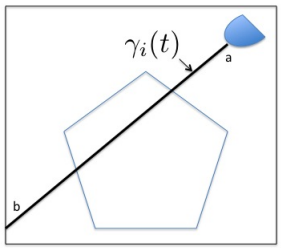

Fig. 3: Parameterized Line of Action along an object

A. Distribution on Contact Points

In our implementation we discretize along the line $\gamma(t)$ evenly but write the derivation in continuous form for generality. The probability distribution along the line $\gamma(t)$ is given by the following:

$$p\big(f(\gamma(t))|\theta(\gamma(t)) : \forall t \in [a, b]\big) = \mathcal{N}(\mu_{a:b}, \Sigma_{a:b}). \quad (6)$$

This gives the signed distance function distributions along the entire line of action in the workspace as a multivariate gaussian. We would like to find the distribution on the first contact point, which we can define as when the signed distance function $f(\gamma(t))$ is 0 and all previous times τ we have $f(\gamma(\tau)) > 0$ for $0 \leq \tau < t$. This ensures t is at the edge of the surface (having $f(\gamma(t)) = 0$) and that for all previous τ the gripper was outside of the surface (having $f(\gamma(\tau)) > 0$). We thus compute this as the joint distribution $p\big(\mathbf{c}_i = \gamma(t)\big) = p\big(f(\gamma(t)) = 0, f(\gamma(\tau)) > 0 : \forall \tau \in [0, t)\big)$. This avoids the problem of the distribution producing multiple modes along the line: one for each intersection with the surface. We now derive this distribution

$$p\big(\mathbf{c}_i = \gamma(t)\big) \propto$$
$$p\big(f(\gamma(t)) = 0\big) P\big(f(\gamma(\tau)) > 0 | f(\gamma(t)) = 0 : \forall \tau \in [0, t)\big)$$

where we only indicate proportionality and will later normalize to probability 1. Using the first product in the equation can be computed easily using the marginalization of a multivariate Gaussian distribution and the second one can be rewritten by conditioning the distribution [19].

$$p_c\big(f(\gamma(\tau)) : \forall \tau \in [0, t)\big) = p\big(f(\gamma(\tau)) | f(\gamma(t)) = 0\big)$$

The following can now be said:

$$p\big(\mathbf{c}_i = \gamma(t)\big) =$$
$$= \frac{1}{\eta} p(f(\gamma(t)) = 0) P_c\big(f(\gamma(\tau)) > 0) : \forall \tau \in [0, t)\big)$$

for the appropriate η normalization factor. The second product term can be evaluated by calculating the cumulative distribution of a multivariate Gaussian, which we calculate with the Matlab function $mvncdf$. We show again the theoretical distribution on $\mathbf{c}_i$ calculated for a given GPIS and approach direction in Fig. 4.

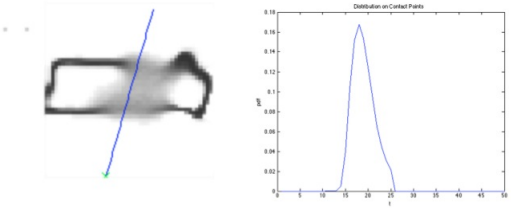

Fig. 4: Left: Grasp approach direction on an uncertain surface, represented by a Gaussian Process Implicit Surface. Right: Induced distribution on the contact point as a function of x-axis position along the approach line.

B. Distribution on Surface Normals

The distribution of surface normals $p(\mathbf{n}_i = \mathbf{v})$ can be calculate as follows. First we assume that some function exists $h(x) = (\mu_\nabla(x), \Sigma_\nabla(x))$, hence given a point $\mathbf{x}$ it returns the parameters for a Gaussian distribution around the gradient. this function can be computed via learning the gradient [15] or analytical differentiation of $f(x)$. We note

that both methods yield a Gaussian distribution. We now demonstrate how to marginalize out the contact distribution and compute $p(\mathbf{n}_i = \mathbf{v})$.

From our distribution on contact points and Bayes rule we can compute the following:

$$p(\mathbf{c}_i = \gamma(t), \mathbf{n}_i = \mathbf{v}) = p\big(\mathbf{n}_i = \mathbf{v} | \mathbf{c}_i = \gamma(t)\big) p\big(\mathbf{c}_i = \gamma(t)\big) \tag{7}$$

Now we can marginalize out the distribution on contacts:

$$p(\mathbf{n}_i = \mathbf{v}) = \int_a^b p\big(\mathbf{n}_i = \mathbf{v} | \mathbf{c}_i = \gamma(t)\big) p(\mathbf{c}_i = \gamma(t)) dt \tag{8}$$

$$p\big(\mathbf{n}_i = \mathbf{v}\big) = \int_a^b p\big(\mathbf{n}_i = \mathbf{v} | h(\gamma(t))\big) p\big(\mathbf{c}_i = \gamma(t)\big) dt \tag{9}$$

We approximate this by uniformly sampling the integral along the function $\gamma(t)$ and achieve the following:

$$p\big(\mathbf{n}_i = \mathbf{v}\big) = \sum_T p\big(\mathbf{n}_i = \mathbf{v} | h(\gamma(t))\big) p\big(\mathbf{c}_i = \gamma(t)\big) \Delta t \tag{10}$$

Grasp metrics such as Ferrari-Canny require $\mathbf{n}_i$ be normalized, or, equivalently, a member of $\mathcal{S}^{d-1}$ [6]. To account for this we project the Gaussian distribution $p\big(\mathbf{n}_i = \mathbf{v} | \mathbf{c}_i = \gamma(t)\big)$ onto $\mathcal{S}^{d-1}$. We use a projection technique developed by Olano and North [20]. In Fig. 5, we show the theoretical distribution on $\mathbf{n}_i$ calculated for a given GPIS and approach direction.

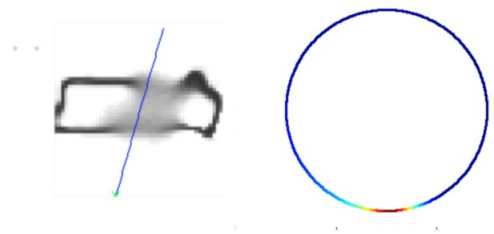

Fig. 5: Left: Grasp approach direction on an uncertain surface, represented by a Gaussian Process Implicit Surface. Right: Induced distribution on the surface normals.

C. Expected Center of Mass

We recall the quantity $\text{CDF}(f(x))(0) = \int_{-\infty}^{0} p(f(x) = s \mid \theta(x)) ds$ as the cumulative distribution function of $f(x)$ evaluated at 0 and note that it is equal to the probability that x is interior to the surface under the current observations. We assume that the object has uniform mass density and then $\text{CDF}(f(x))(0)$ is the expected mass density at x. Then we can find the expected center of mass as:

$$\bar{z} = \frac{\int_{\mathcal{R}} x CDF(f(x))(0) dx}{\int_{\mathcal{R}} CDF(f(x))(0) dx} \tag{11}$$

which can be approximated by sampling $\mathcal{R}$ uniformly in a grid and approximating the spatial integral by a sum. We show the computed density and calculated expected center of mass for a marker in Fig. 6.

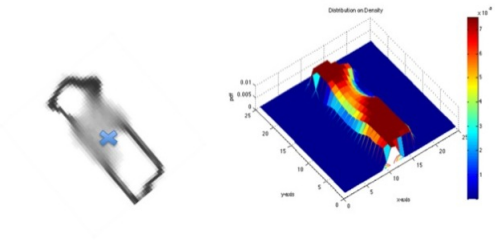

Fig. 6: Left: A surface with GPIS construction and expected center of mass (blue X) Right: The distribution on the density of each point assuming uniform center of mass

VI. Sampling from a GPIS

Prior work that consider shape uncertainty purposed sampling from distribution on the shape itself [3]. However, with a more complicated model on the shape distribution this technique would scale poorly.

Given a work space $\mathcal{R}^{n \times n}$, the complexity of sampling a work space is show to be $O(n^6)$. This is due to the necessary inversion of a $n \times n$. In typical cases n would be on the order of 100 making this very computationally intensive even in the 2D case.

If we instead sample from our proposed grasp distributions and use a line of action that is roughly equal to n, the complexity can be rewritten as $O(T * n^3)$, where T is the number of grasps that are evaluated.

VII. Adaptive Sampling for Grasp Selection

In a possible grasping scenario, one would be presented with a set G of T grasps. The goal would be to identify, which one is the best grasp given the set G and for each grasp the $p(g)$. While a naive approach to solving this problem would be to simply perform Monte-Carlo integration on each one and compute the expected grasp quality, we propose treating the problem as an n-arm bandit problem and forming a policy for selecting which grasp to sample. The motivation behind this is due to the expensive evaluation that each Ferrari-Canny computation takes [6].

An n-arm bandit problem can be defined as follows: an agent is presented with a n slot machines and receives a reward each time a lever is pulled that is drawn from the distribution associated with the slot machine. The goal of

the agent is to maximize the expected reward received, by balancing exploration and exploitation [21].

For selecting grasps we use an ϵ-greedy policy, where the greedy action is selected with probability $1-\epsilon$ and a random action is chosen otherwise. The greedy option is defined as the grasp with the highest approximate upper confidence bound v, which is computed via the following [**?**]

$$v = \bar{X} + \frac{1.96\bar{\sigma}_N}{\sqrt{n}} \tag{12}$$

VIII. Experiments

A. Validation of Grasp Distributions

To validate our theoretical calculations, we compared them to empirical distributions found via Monte-Carlo Sampling. The qualitative results for surface normals and contacts can be found in Figures 7 and 8 respectively. We present the numerical comparison with KL-divergence in Table 1, as you can see the distributions are very closely related.

Distribution	**Marker**	**Tape**	**Loofa**
p($\mathbf{c}_i$)	0.14	0.08	0.02
p($\mathbf{n}_i$)	0.35	0.12	0.16

TABLE I: The KL-Divergence for different distributions on the contact points and surface normals

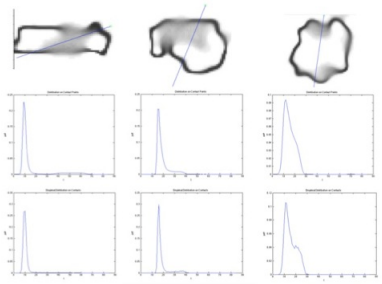

Fig. 7: Top: A surface with GPIS construction and line of action with the start pointed denoted by green x. Middle: Theoretical distribution computed on contact point Bottom: Empirically sampled distribution

B. Sampling from Grasps Vs. Shape

We first demonstrate that sampling from the two different method converges to near the same expected value, see Fig. 9. After confirming the distributions converge to the same value we show the computational complexity in Fig. 10 of the two methods for evaluating 100 grasps on an $n \times n$ grid.

C. Adaptive Sampling Technique

We consider the problem of selecting the best grasp out of a set G given a fix number of iterations I. For our experiments we look at selecting the best grasp out of a size

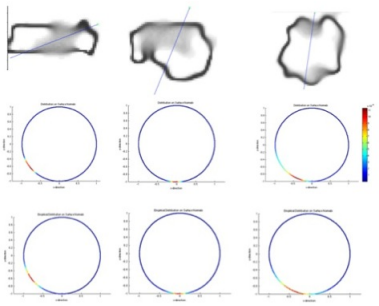

Fig. 8: Top: A surface with GPIS construction and line of action with the start pointed denoted by green x. Middle: Theoretical distribution computed for surface normals Bottom: Empirically sampled distribution

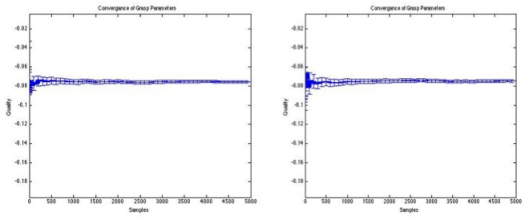

Fig. 9: Left: Sampling from grasp distributions Right: Sampling from distribution on shape

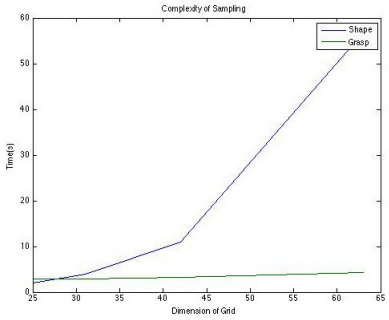

Fig. 10: Comparison of the computational complexity of the two methods with respect to the work space dimensions

of $|G| = 100$ and given a fix iteration limit of $I = 1000$. We show the results for both the marker and tape, in Fig. 11 and Fig. 12 respectively. We note that in both cases the policy was able to select the best grasp without sufficiently sample the set of possible grasps.

[TODO: ADD EXPERIMENTS DEMONSTRATING ORDERING AND TRIALS OVER LARGE NUMBER OF RUNS, ALSO FIX VISUALIZATIONS]

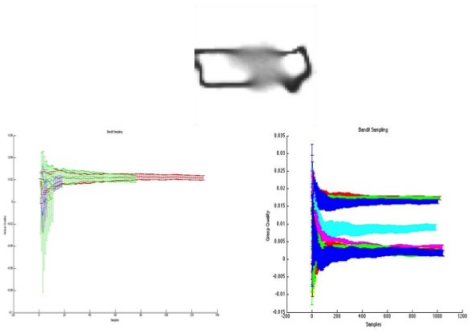

Fig. 11: Left: ϵ-greedy policy for sampling (Top 10 grasps plotted) Right: Monte-Carlo Sampling for 1000 iterations (Top 10 grasps plotted)

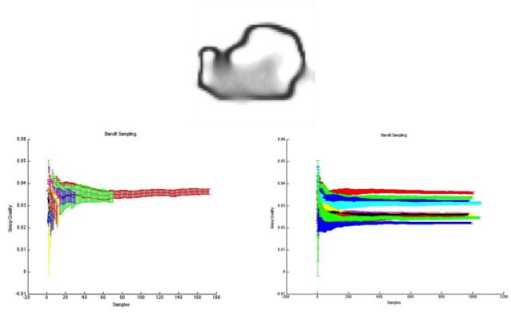

Fig. 12: Left: ϵ-greedy policy for sampling (Top 10 grasps plotted) Right: Monte-Carlo Sampling for 1000 iterations (Top 10 grasps plotted)

[TODO: WRITE CONCLUSION]

REFERENCES

[1] R. A. Newcombe, A. J. Davison, S. Izadi, P. Kohli, O. Hilliges, J. Shotton, D. Molyneaux, S. Hodges, D. Kim, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pp. 127–136, IEEE, 2011.

[2] A. Van Dam and S. K. Feiner, *Computer graphics: principles and practice*. Pearson Education, 2014.

[3] B. Kehoe, D. Berenson, and K. Goldberg, "Toward cloud-based grasping with uncertainty in shape: Estimating lower bounds on achieving force closure with zero-slip push grasps," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pp. 576–583, IEEE, 2012.

[4] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pp. 303–312, ACM, 1996.

[5] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.

[6] C. Ferrari and J. Canny, "Planning optimal grasps," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 2290–2295, 1992.

[7] K. Y. Goldberg and M. T. Mason, "Bayesian grasping," in *Robotics and Automation, 1990. Proceedings., 1990 IEEE International Conference on*, pp. 1264–1269, IEEE, 1990.

[8] F. Stulp, E. Theodorou, J. Buchli, and S. Schaal, "Learning to grasp under uncertainty," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pp. 5703–5708, IEEE, 2011.

[9] V. N. Christopoulos and P. Schrater, "Handling shape and contact location uncertainty in grasping two-dimensional planar objects," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pp. 1557–1563, IEEE, 2007.

[10] J. Felip and A. Morales, "Robust sensor-based grasp primitive for a three-finger robot hand," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 1811–1816, IEEE, 2009.

[11] Y. Zheng and W.-H. Qian, "Coping with the grasping uncertainties in force-closure analysis," *Int. J. Robotics Research (IJRR)*, vol. 24, no. 4, pp. 311–327, 2005.

[12] B. Kehoe, D. Berenson, and K. Goldberg, "Estimating part tolerance bounds based on adaptive cloud-based grasp planning with slip," in *Automation Science and Engineering (CASE), 2012 IEEE International Conference on*, pp. 1106–1113, IEEE, 2012.

[13] S. Dragiev, M. Toussaint, and M. Gienger, "Gaussian process implicit surfaces for shape estimation and grasping," in *Proc. IEEE Int. Conf. Robotics and Automation (ICRA)*, pp. 2845–2850, 2011.

[14] C. E. Rasmussen and H. Nickisch, "Gaussian processes for machine learning (gpml) toolbox," *The Journal of Machine Learning Research*, vol. 9999, pp. 3011–3015, 2010.

[15] E. Solak, R. Murray-Smith, W. E. Leithead, D. J. Leith, and C. E. Rasmussen, "Derivative observations in gaussian process models of dynamic systems," 2003.

[16] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," *Gaussian Proc. in Practice*, 2007.

[17] M. Bjorkman, Y. Bekiroglu, V. Hogman, and D. Kragic, "Enhancing visual perception of shape through tactile glances," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, pp. 3180–3186, IEEE, 2013.

[18] C. Rasmussen and C. Williams, *Gaussian processes for machine learning*. MIT Press, 2006.

[19] K. B. Petersen, "The matrix cookbook,"

[20] M. Olano and M. North, "Normal distribution mapping," *Univ. of North Carolina Computer Science Technical Report*, pp. 97–041, 1997.

[21] A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 1998.