

Grasp Planning for Flexible Production with Small Lot Sizes based on CAD models using GPIS and Bayesian Optimization

Jianjie Lin, Markus Rickert and Alois Knoll

Abstract—Grasp planning for multi-fingered hands is still a challenging task due to the high nonlinear quality metrics, the high dimensionality of hand posture configuration, and complex object shapes. Analytical-based grasp planning algorithms formulate the grasping problem as a constraint optimization problem using advanced convex optimization solvers. However, these are not guaranteed to find a globally optimal solution. Data-driven based algorithms utilize machine learning algorithm frameworks to learn the grasp policy using enormous training data sets. This paper presents a new approach for grasp generation by formulating a global optimization problem with Bayesian optimization. Furthermore, we parameterize the object shape utilizing the Gaussian Process Implicit Surface (GPIS) to integrate the object shape information into the optimization process. Moreover, a chart defined on the object surface is used to refine palm pose locally. We introduced a dual optimization stage to optimize the palm pose and contact points separately. We further extend the Bayesian optimization by utilizing the alternating direction method of multipliers (ADMM) to eliminate contact optimization constraints. We conduct the experiments in the graspi! Simulator that demonstrates the effectiveness of this approach quantitatively and qualitatively. Our approach achieves a 95% success rate on various commonly objects with diverse shapes, scales, and weights.

I. INTRODUCTION

Only a limited number of small and medium-sized enterprises in Europe use robot systems in production, mainly dealing with small lot sizes and requiring a more flexible production process. It is, however, very time-consuming and expensive to adapt a robot system to a new production line, and it requires expert knowledge for deploying such a system, which, however, is not commonly available in shop floor workers [1]. Intuitive programming is currently proposed on the market for accelerating programming and remedying the problems caused by a lack of expert knowledge. Moreover, the service robots use semantic knowledge in combination with reasoning, and inference [2] to solve the declarative goal. Automatically synthesizing a robot program based on the semantic product, process, and resource descriptions enable an automatic adaptation to new processes. In this process, the recognition of objects and parts in the environments is involved, which is typically designed in CAD systems and described via a boundary representation [3]. Due to the small lot size production of SMEs, it is not feasible to train the objects over a long period of time by using the data-driven approaches. Based on this observation, it will accelerate the deploying time if we grasp the object firstly in a simulator

Jianjie Lin, Markus Rickert and Alois Knoll are with Robotics and Embedded System, Department of Informatics, Technische Universität München, Munich, Germany jianjie.lin@tum.de, {rickert, knoll}@in.tum.de

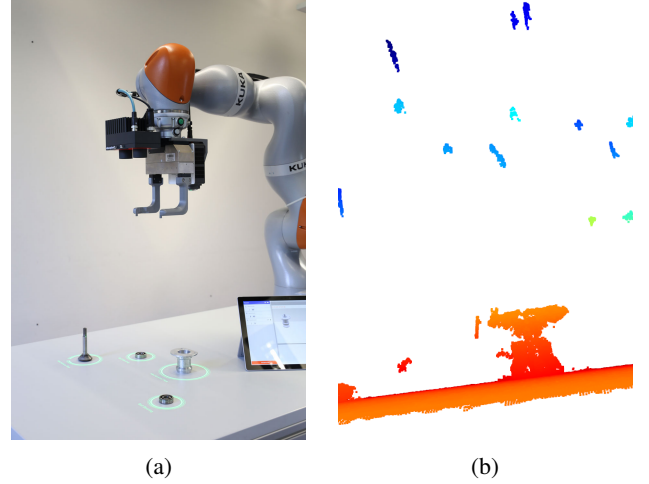


Fig. 1: Robot setup for assembling a gear box with (a) a lightweight robot and a 3D camera sensor, (b) point cloud scene of mechanical gearbox parts on the table.

with the CAD models and then transfer the preplanned grasp to the real world with a 6D pose estimation [4]. Fig. 1 shows an example of such a grasping use case for a mechanical gearbox together with a point cloud scene captured by the 3D camera sensor attached to the robot.

Dexterous robotic grasping planning has been an active research subject in the robotic community over the past decades. Grasping is essential in many areas such as industrial factories and household scenarios. There have many different kinds of robotic hands, i.e., traditional parallel-jaw grippers, complex multi-fingered hands, or even vacuum-based end effectors. The goal of grasp planning aims to find a proper contact on the object and an appropriate posture of the hand related to the object to maximize grasp quality. This is a challenging task, especially for multi-fingered hands, due to different kinds of object shapes, the complicated geometric relationship between robotic hands and objects, and the high dimensionality of hand configurations. Grasp planning can be divided into analytic approaches [5] on the one side and empirical or data-driven approaches on the other side [6]. The analytic grasp synthesis approach is usually formulated as a constrained optimization problem over criteria that measure dexterity, equilibrium, and stability and exhibit a certain dynamic behavior. Besides, it requires the analysis of statically-indeterminate grasps [5] and under-actuated systems. The latter describes hands, in which the number of the controlled degrees of freedom is fewer than the number of

contact forces, therefore further increasing the complexity of grasp synergies. One common assumption made in analytic methods is that precise geometric and physical models are available to the robot. Furthermore, optimizing grasp quality with constraints based on a convex optimization solver such as SQP can not guarantee finding a good grasp. In contrast to analytic approaches, the empirical or data-driven approaches rely on sampling the grasp candidate either from a data set or by first learning a grasp quality and then selecting the best by ranking them according to some specific metric. In this work, we will study how to optimize the palm pose and contact point in the same framework by utilizing a global Bayesian optimization solver under consideration constraints. Gaussian Process Implicit Surface Atlas (GPIS-Atlas) is used to parameterize the diverse shape. Therefore the geometry information can be integrated into the Bayesian optimization framework. Furthermore, GPIS-Atlas has the capability to describe the perfect geometry model in the form of a CAD model or the noisy point clouds [7]. In the work [4], the 6D pose is estimated between an object in the form of CAD model and the corresponding point clouds taken from an Ensenso Camera. Therefore, for finding an appropriate grasp pose for this object, we can directly apply our Bayesian optimization framework with the CAD model instead of working on noisy point clouds. After that, we transform the grasp pose using the 6D pose transformation from [4].

II. RELATED WORK

Multi-fingered hand grasp planning is still challenging due to the high dimensionality of hand structure and complex graspable object shapes. Automatic grasp planning is a difficult problem because of the vast number of possible hand configurations. Several different approaches have been proposed to find an optimal grasp pose over the past decades. Goldfeder et al. [8] introduced a database-backed grasp planning, which uses shape matching to identify known objects in a database with are likely to have similar grasp poses [8]. Ciocarlie et al. presented [9] Eigengrasp planning defines a subspace of a given hand’s DOF space and utilizes the Simulated Annealing planner to find an optimized grasp. Miller et al. [10] proposed a primitive shape-based grasp planning which generates a set of grasps by modeling an object as a set of shape primitives, such as spheres, cylinders, cones, and boxes. Pelosso et al. [11] use an approach based on Support Vector Machines that approximate the grasp quality with a new set of grasp parameters. It considers grasp planning as a regression problem by given a feature vector, which should be defined heuristically. With the continuous success of deep learning vision, researchers utilize deep learning, also in combination with reinforcement learning, to learn a grasp quality directly from an image via large training data sets [12]. Levine et al. [13] used between 6 and 14 robots at any given point in time to collect data in two months and train a convolutional neural network to predict grasp success for a pick-and-place task with a parallel-jaw gripper. Mahler et al. [14] proposed a Dex-Net-based deep learning framework using a parallel-jaw gripper or vacuum-

based end effector learn a grasp policy based on millions of grasp experiments. Kalashnikov et al. [15] introduced a scalable self-supervised vision-based reinforcement learning framework to train a deep neural network Q-function by leveraging over 580k real-world grasp attempts. However, the deep learning-based algorithm can only take the 2d image as an input, and the trained neural network cannot be easily transferred to another robotic hand configuration. Varley et al. [16] obtained the geometry representation of grasping objects from point clouds using a 3D-CNN. Ten et.al [17] is the state of the art 6 DOF grasp planner (GPD). Liang et.al. [18] proposed an end-to-end PointNetGPD to detect the grasp configuration from a point sets. Mousavian et al. introduced a 6DOF GraspNet by sampling a set of grasping using a variational autoencoder. In addition to deep learning, the Bayesian optimization-based algorithm in [19] can consider uncertainty in input space to find a safe grasp region by optimizing the grasp quality. Furthermore, it utilizes unscented transformation-based Bayesian optimization (UBO), a popular nonlinear approximation method, to explore the safe region. However, UBO considers only the palm pose optimization without considering the contact point. We present a grasp planning approach in this work, where we combine Bayesian optimization with an analytical approach. We use the Grasp Wrench Space (GWS) [20] as grasp quality metric, which calculates the convex hull over discretized friction cones from the individual contact wrench spaces of all contacts. Due to the GWS metric’s complexity, we explore the potential of Bayesian optimization to optimize this highly-nonlinear grasp quality problem. Since the hand posture (hand palm pose) and hand configuration can be considered separately because the finger’s contact points on the object surface only depend on the hand posture and forward kinematics of hand joints, we propose a dual-stage approach: In the first stage, we optimize the hand palm pose without considering hand configuration, and in the second stage, we use the result of the first stage and optimize the contact points on the object surface. Our approach optimizes a hand palm pose $T \in SE(3)$ regarding its grasp quality. For this, we present the rotation in hyperspherical coordinates instead of a rotation matrix or quaternion. We further parameterize the object surface as a Gaussian Process Implicit Surface (GPIS) [7] and use a k -D tree to find the closest point between the current palm pose and object surface. Based on GPIS, we can further compute a chart C and the corresponding normal vector N_C on this nearest point. Utilizing this chart, we can make a local adaption of the palm pose to find a better location concerning the object surface. In the second stage, we convert the problem of solving constraints between the contact points and the object surface to querying the GPIS given a known contact point. Since the standard framework of Bayesian optimization cannot solve this constraint optimization problem, we use the Alternating Direction Method of Multipliers (ADMM) [21] to assist the contact point optimization by decomposing the whole problem into a set of subproblems.

III. PROBLEM FORMULATION

In general, to define a grasp, we need two sets of variables: the intrinsic variables to define the hand degrees of freedom (DOF) and the extrinsic variables to define the hand's position relative to the target object. Grasp planning is used to find the optimized contact points and an associated hand configuration to maximize grasp quality. The contact point on the object surface is denoted as $\mathbf{c} = [\mathbf{c}_1, \dots, \mathbf{c}_n]$, where $\mathbf{c}_i \in \text{SE}(3)$, and n is the number of fingers. We will assume that contact happens on the fingertip, and one finger only has one contact on the object surface \mathcal{O} . The finger joint of the hand configuration is described as $\mathbf{q} = [\mathbf{q}_1, \dots, \mathbf{q}_m]$, where m is the DOF. Note that some finger joints are under-actuated (passive joint). Therefore the number of finger joints is not equal to the DOF. The pose of the palm is represented by $\mathbf{T}_{\text{palm}}(\mathbf{R}, \mathbf{t})$. Mathematically, the optimized problem can be formulated as:

$$\begin{aligned} & \max_{\mathbf{c}, \mathbf{q}, \mathbf{T}_{\text{palm}}} Q(\mathbf{c}, \mathbf{q}, \mathbf{T}_{\text{palm}}) & (1a) \\ \text{s.t. } & \mathbf{c} = \text{FK}_{\text{palm2c}}(\mathbf{q}, \mathbf{T}_{\text{palm}}) \in \mathcal{O} & (1b) \\ & q_{\min, i} \leq q_i \leq q_{\max, i}, \quad i = 1 \dots m, & (1c) \end{aligned}$$

where $Q(\mathbf{c}, \mathbf{q}, \mathbf{T}_{\text{palm}})$ is the GWS, which is a 6-dimensional convex polyhedron, the epsilon and volume quality metric introduced by Ferrari and Canny [20]. The epsilon quality is defined as the minimum distance from the origin to any of the hyperplanes defining the boundary of the GWS. In contrast, the volume quality is the volume of GWS. $\text{FK}_{\text{palm2c}}$ is the forward kinematics from the palm pose to the contact points. The formulation (1b) constrains all contact points on the object surface \mathcal{O} . Furthermore, a contact is defined as any point where two bodies are separated by less than the contact threshold ϵ_c , but not interpenetrating. In this work, the object surface will be parameterized by using GPIS to easily check if the contact point satisfies the constraint (1b). The problem in (1) is a high-dimensional nonlinear constraint problem, besides the gradient of the objective function and constraints cannot be analytically computed. Further, the convex optimization solver can only find a local minimum. Based on those observations, Bayesian optimization is applied to find a near-global optimization solution for these problems. Since the palm pose and contact point are nearly independent, we can switch between optimizing the palm pose and the contact point.

IV. BAYESIAN OPTIMIZATION FOR GRASP PLANNING

Bayesian optimization is a global optimization method, which can be used to solve the problem

$$\mathbf{x}_{\text{optimized}} = \max_{\mathbf{x} \in \mathcal{X}} f_{\text{obj}}(\mathbf{x}), \quad (2)$$

where the objective function $f_{\text{obj}}(\mathbf{x})$ is a black-box function or a function which is expensive to evaluate and $\mathcal{X} \subseteq \mathbb{R}^D$ is a bounded domain. We use the Latin Hypercube Sampling (LHS) to get the initial sampling, and save it as data set $\mathcal{D}_{0:t-1} = \{(\mathbf{x}_0, y_0), \dots, (\mathbf{x}_{t-1}, y_{t-1})\}$, and learn a Gaussian process model (GP). The essential step is to choose

an appropriate acquisition function. Here, we use Expected Improvement [22], which is defined as

$$\text{EI}(\mathbf{x}) = \mathbb{E}[\max(f_{\text{obj}}(\mathbf{x}) - f_{\text{obj}}(\mathbf{x}^+), 0)], \quad (3)$$

where \mathbb{E} is the expectation function, $f_{\text{obj}}(\mathbf{x}^+)$ is the best observation with the location \mathbf{x}^+ so far. The $\text{EI}(\mathbf{x})$ can be evaluated analytically under the GP model as

$$\text{EI}(\mathbf{x}) = \mathbb{1}(\sigma(\mathbf{x})) \left((\mu(\mathbf{x}) - f_{\text{obj}}(\mathbf{x}^+) - \xi) \Phi(Z) + \sigma(\mathbf{x}) \phi(Z) \right),$$

with $Z = \mathbb{1}(\sigma(\mathbf{x})) \left(\frac{\mu(\mathbf{x}) - f_{\text{obj}}(\mathbf{x}^+) - \xi}{\sigma(\mathbf{x})} \right)$, where $\mathbb{1}(x)$ is the indicator function that is equal to 0 for $x \leq 0$ and equal to 1 otherwise. The mean $\mu(\mathbf{x})$ and standard deviation $\sigma(\mathbf{x})$ are defined in the GP posterior at \mathbf{x} , and Φ and ϕ are the CDF (cumulative distribution function) and PDF (probability density function) of the standard normal distribution. ξ is a parameter which balances between the exploration and exploitation. The objective function in our algorithm for optimizing the grasp contact points is grasp quality which consist of epsilon and volume quality.

$$f_{\text{obj}}(\mathbf{x}) = (\mathbb{1}(q_\epsilon) q_\epsilon + \lambda q_{\text{volume}}), \quad (4)$$

where λ is a predefined parameter. The Matérn covariance function ($\nu = 5/2$) is chosen as kernel function for the Gaussian process model in the Bayesian optimization and can be described as: $K_{5/2}(d) = \sigma^2 \left(1 + \frac{\sqrt{5}d}{\rho} + \frac{5d^2}{3\rho^2} \right) \exp\left(-\frac{\sqrt{5}d}{\rho}\right)$ with hyper parameter σ and length-scale ρ . The parameter d is the distance between two query points. Since we need to optimize the palm pose, which is interpreted as a transformation. We define the distance between two transformation matrices as $\Delta_T = \|\mathbf{t}_1 - \mathbf{t}_2\| + \|\log(\mathbf{R}_1^T \mathbf{R}_2)\|_F / \sqrt{2}$, where the term $\|\log(\mathbf{R}_1^T \mathbf{R}_2)\|_F / \sqrt{2}$ is the geodesic distance defined in the Riemann manifold. To optimize the hyperparameter, the algorithm RProp (Resilient Propagation) [23] is applied, a popular gradient descent algorithm that only uses the signs of gradients to compute updates and can dynamically adapt the step size for each weight independently.

The object surface in our algorithm is described as a GPIS and every point lying on the surface in the set \mathcal{X} , should satisfy the equality constraints $\mathcal{X} = \{\mathbf{x} \in \mathbb{R}^3 : f_{\text{GPIS}}(\mathbf{x}) = 0\}$. Furthermore, the tangent space of each point on these surface will be computed by using

$$\begin{bmatrix} \nabla f_{\text{GPIS}}^T(\mathbf{x}_i) \\ \Phi_i^T(\mathbf{x}_i) \end{bmatrix} \Phi_i(\mathbf{x}_i) = \begin{bmatrix} 0 \\ I \end{bmatrix}, \quad (5)$$

where $\Phi_i(\mathbf{x}_i) \in \mathbb{R}^{3 \times 2}$ is the basis of tangent space at the location \mathbf{x} and $\nabla f_{\text{GPIS}}(\mathbf{x}_i) \in \mathbb{R}^{3 \times 1}$ is the gradient of implicit function for \mathbf{x} . By using $\Phi_i(\mathbf{x})$, we can map \mathbf{x}_i to \mathbf{x}'_i with $\mathbf{x}'_i = \mathbf{x}_i + \Phi_i(\mathbf{x}_i) \mathbf{u}_i$, where $\mathbf{u}_i \in \mathbb{R}^{2 \times 1}$ is a point in the local coordinate on this chart. The sample value \mathbf{x}'_i is shown as black dots in Fig. 2a, the tangent vector on the chart is shown as red and green arrows, while the blue arrow shows the normal vector.

A. Hand Palm Pose Optimization (HPP-Opt)

To optimize the palm pose, we need to take the transformation $T_{\text{palm}}(\mathbf{R}, \mathbf{t}) \in \text{SE}(3)$ into account, where the rotation matrix \mathbf{R} can be interpreted by using a unit quaternion \mathbf{q} . Since the unit quaternion manifold $\mathcal{M}_{\mathbb{H}}$ is an Riemannian manifold, by virtual equality of $\mathcal{M}_{\mathbb{H}}$ and 4D unit hypersphere $S^3 = \{\mathbf{x} \in \mathbb{R}^{3+1} : \|\mathbf{x}\| = 1\}$, the quaternion \mathbf{q} can be represented in the hyperspherical coordinates with ϕ, ψ, θ where ϕ, ψ range over $[0, \pi]$ and θ ranges over $[0, 2\pi)$. Employing hyperspherical coordinates, the constraints of \mathbf{q} disappear. Therefore, the palm pose optimization is converted to an unconstrained optimization, and it can be mathematically formulated as

$$\max_{\phi, \psi, \theta, \mathbf{t}_{\text{palm}}} f_{\text{obj}}(\mathbf{q}, T_{\text{palm}}(\mathbf{R}(\phi, \psi, \theta), \mathbf{t}_{\text{palm}})). \quad (6)$$

To reduce the search space, we constrain the palm pose between two bounding boxes, represented by an axis-aligned minimum bounding box (AABB), denoted as $\mathcal{X}_{\text{AABB},i}$ with $i \in \{1, 2\}$. The smaller one is shown as an orange cube in Fig. 2a. We define the variable $\mathbf{x}_{\text{palm}} = [t, \phi, \psi, \theta]^T \in \mathbb{R}^{6 \times 1}$. Sampling a point between two bounding box cannot be formulated mathematically, therefore, we use an ellipsoid $\mathcal{E}_1 = (a_1, b_1, c_1, a_0, b_0, c_0)$ to approximate $\mathcal{X}_{\text{AABB},1}$ and another ellipsoid $\mathcal{E}_2 = (a_2, b_2, c_2, a_0, b_0, c_0)$ to approximate the bigger bounding box $\mathcal{X}_{\text{AABB},2}$. As the result, the palm pose sample $\mathbf{t} = [t_x, t_y, t_z]$ can be formulated as

$$\begin{aligned} t_x &= a_0 + r * a_1 \sin(\theta) \cos(\phi) \\ t_y &= b_0 + r * b_1 \sin(\theta) \sin(\phi) \\ t_z &= c_0 + r * c_1 \cos(\theta), \end{aligned}$$

where $m_0 = \frac{k_{\min,1} + k_{\max,1}}{2}$, $m_1 = \frac{k_{\max,1} - k_{\min,1}}{2}$, $m \in \{a, b, c\}$, and $k \in \{x, y, z\}$. The parameter r is a random variable

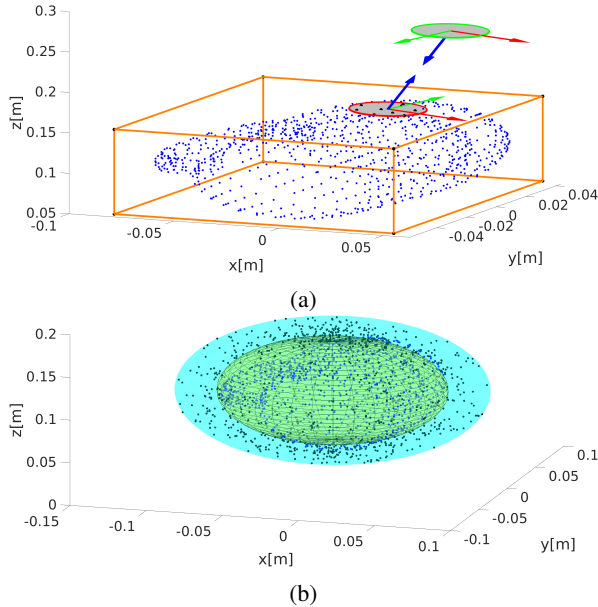


Fig. 2: Visualization of (a) Tangent space on a mug surface and (b) sampling for initial palm pose.

Algorithm 1 Palm Pose Optimization HPP-Opt

Require: $n_{\text{dim}}, n_{\text{iter}}$

- 1: Get LHS $\mathcal{D}_{0:t-1} = \{(\mathbf{x}_0, y_0), \dots, (\mathbf{x}_{t-1}, y_{t-1})\}$
- 2: Fit the Gaussian process Model $p(y|\mathbf{x}, \mathcal{D}_{0:t-1})$
- 3: Optimized hyper Parameter Rrop
- 4: **for** $t = 1$ to n_{iter} **do**
- 5: $\mathbf{x}_{\text{palm},t} = \arg \max_{\mathbf{x}} \text{EI}(\mathbf{x}|\mathcal{D}_{0:t-1})$
- 6: Find a collision-free contact and get $\mathbf{q}, \mathbf{x}_{\text{palm}}$
- 7: $\mathcal{G} = \text{GPISAtlas}(f_{\text{GPIS}}, \mathbf{x}_{\text{palm}})$
- 8: $\{\mathcal{G}_{\text{best}}(\mathbf{x}_{\text{palm},t}, \mathbf{q}), y_{\text{max},t}\} = \arg \max_{\mathcal{G}} f_{\text{obj}}(\mathbf{q}, \mathbf{x}_{\text{palm}})$
- 9: Add the new sample $\mathcal{D}_{0:t} = \{\mathcal{D}_{0:t-1}, (\mathbf{x}_{\text{palm},t}, y_{\text{max},t})\}$
- 10: update Gaussian process model (GP)
- 11: **end for**
- 12: $\mathbf{x}_{\text{palm}}^+ = \arg \max_{\mathbf{x}_i \in \mathcal{X}_{0:t}} f_{\text{obj}}(\mathbf{x}_i)$
- 13: **return** $\mathbf{x}_{\text{palm}}^+$

which ranges over $[1, r_{\text{max}}]$, where

$$r_{\text{max}} = \sqrt{\frac{1}{\left(\frac{a_1 \sin(\theta) \cos(\phi)}{a_2}\right)^2 + \left(\frac{b_1 \sin(\theta) \sin(\phi)}{b_2}\right)^2 + \left(\frac{c_1 \cos(\theta)}{c_2}\right)^2}}.$$

The parameter θ ranges over $[0, \pi]$ and θ ranges over $[0, 2\pi)$. The two ellipsoids are shown in Fig 2b. The problem in (6) can be solved by using Bayesian optimization, as shown in Alg. 1. The solution found by Bayesian optimization is based on the probability of best grasp distribution. To improve the performance, a local adaption based on GPIS-Atlas is proposed.

1) *GPIS-Atlas based local adaption:* The first step is to find the closest point from the current palm pose to the object using k -D tree algorithm. Assuming we found the pose $\mathbf{t}_{\text{closest}}$, a chart C_i with the center point $\mathbf{t}_{\text{closest}}$ is created by solving (5). We denoted the outward unit normal vector of this chart C_i as $\mathbf{N}_{\text{closest}}$ and the robot hand posture is designed to orient to the direction of a chart normal \mathbf{N}_C . We apply the following approaches to get the pose \mathbf{T} . Assuming that the normal vector of hand in the initial state points to the z -axis \mathbf{n}_z , the hand is currently in local frame \mathcal{H}_1 with rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{H}_1}$ with respect to the world frame \mathcal{W} . The next hand configuration should point to the normal direction \mathbf{N}_C in local frame \mathcal{H}_{C_i} , therefore the corresponding rotation matrix ${}^{\mathcal{W}}\mathbf{R}_{\mathcal{H}_{C_i}}$ can be interpreted in angle-axis representation $[n_{\text{axis}}, \theta_{\text{axis}}]$ with $\theta_{\text{axis}} = \text{atan2}(\|\mathbf{n}_z \times \mathbf{N}_C\|, \mathbf{n}_z \cdot \mathbf{N}_C)$ and $n_{\text{axis}} = \mathbf{n}_z \times \mathbf{N}_C$. As a result, we can transform the local frame \mathcal{H}_1 to \mathcal{H}_{C_i} with the rotation transformation as ${}^{\mathcal{H}_1}\mathbf{R}_{\mathcal{H}_{C_i}} = {}^{\mathcal{W}}\mathbf{R}_{\mathcal{H}_1}^T {}^{\mathcal{W}}\mathbf{R}_{\mathcal{H}_{C_i}}$. Furthermore, the translation of the palm pose is defined as $\mathbf{t}_{\text{palm}} = \mathbf{t}_{\text{closest}} + \lambda \mathbf{N}_{\text{closest}}$. This means that the new palm pose \mathbf{x}_{palm} is parallel to the chart C_i with the distance $\|\lambda\|$, as shown in Fig. 2a. The parameter is optimized so that the hand is not colliding with the object. The whole transformation is defined as $\mathbf{T}_{\text{palm}} = \mathbf{T}(\mathbf{I}, \mathbf{t}_{\text{palm}}) \mathbf{T}({}^{\mathcal{W}}\mathbf{R}_{\mathcal{H}_{C_i}}, \mathbf{0}) \mathbf{T}(\mathbf{R}_z, \mathbf{0})$. The transformation $\mathbf{T}(\mathbf{R}_z, \mathbf{0})$ is used to further guarantee no collision. Furthermore, we can define a point set on the chart C_i as \mathcal{X}_{C_i} and randomly choose a sample $\mathbf{t}_{\text{sample}} \in \mathcal{X}_{C_i}$ as $\mathbf{t}_{\text{closest}}$.

Algorithm 2 GPISAtlas() function for local adaption

Require: $f_{\text{GPIS}}, \mathbf{x}_{\text{palm,current}}$

- 1: Set $\mathbf{t} = \mathbf{x}_{\text{palm,current}}(0 : 2)$
- 2: Get $\mathbf{t}_{\text{closest}} = \text{KD}(\mathbf{t}_{\text{palm,current}})$
- 3: Set $\mathcal{G} = \{\}, \lambda \leftarrow 0, \theta_z \leftarrow 0, \lambda_{\text{max}}, n_{\text{seed}}$
- 4: Compute Chart C_i and $\mathbf{N}_{\text{closest}}$ by equation 5
- 5: **for** $n = 0$ to n_{seed} **do**
- 6: $\mathbf{t}'_{\text{closest}} = \mathbf{t}_{\text{closest}} + \Phi_i(\mathbf{t}_{\text{closest}})\mathbf{u}_{\text{rand}}$
- 7: **while** CheckCollision **do**
- 8: $\mathbf{t}_{\text{palm}} = \mathbf{t}'_{\text{closest}} + \lambda \mathbf{N}_{\text{closest}}$
- 9: **if** $\lambda < \lambda_{\text{max}}$ **then**
- 10: $\mathbf{T}_{\text{palm}} = \mathbf{T}_1(I, \mathbf{t}_{\text{palm}})\mathbf{T}_2({}^{\mathcal{W}}\mathbf{R}_{\mathcal{H}_{C_i}}, \mathbf{0})$
- 11: $\lambda \leftarrow \lambda + \Delta_{\lambda}$
- 12: **else**
- 13: $\mathbf{R}_z = \text{Angleaxis}(0, 0, 1, \theta_z)$
- 14: $\theta_z \leftarrow \theta_z + \Delta_{\theta_z}$
- 15: $\mathbf{T}_{\text{palm}} = \mathbf{T}_1(I, \mathbf{t}_{\text{palm}})\mathbf{T}_2({}^{\mathcal{W}}\mathbf{R}_{\mathcal{H}_{C_i}}, \mathbf{0})\mathbf{T}_3(\mathbf{R}_z, \mathbf{0})$
- 16: **end if**
- 17: **end while**
- 18: Execute the Grasp, get hand configuration \mathbf{q}
- 19: $\mathcal{G} = \mathcal{G} \cup \{\mathbf{x}_{\text{palm}}, \mathbf{q}\}$
- 20: **end for**
- 21: **return** \mathcal{G}

B. ADMM-Based Contact Point Optimization (ADMM-CP-Opt)

The contact point optimization is used to find a set of desired joints \mathbf{q} for each finger and the contact points \mathbf{c} on the object surface. It can be described as

$$\max_{\mathbf{c}, \mathbf{q}} f_{\text{obj}}(\mathbf{q}, \mathbf{T}_{\text{palm}}) \quad (7a)$$

$$\text{s.t. } \mathbf{c} = \text{FK}(\mathbf{q}, \mathbf{T}_{\text{palm}}), \quad (7b)$$

$$|f_{\text{GPIS}}(\mathbf{c}_k)| \leq \epsilon_c, \quad k = 1 \dots n \quad (7c)$$

$$q_k \in [q_{\min, k}, q_{\max, k}], \quad k = 1 \dots m, \quad (7d)$$

where FK calculates the forward kinematics, and n represents the number of contact points on the object surface. The parameter m is the DOF of a hand. A contact point is represented as a transformation \mathbf{T}_{C_i} . However, each finger has fewer joints than 6, which results in an underestimated system. Consequently, we cannot directly calculate the inverse kinematics based on the contact points, and it is not possible to arbitrarily move the fingertip on the object surface. To relax the constraints, we will not fix the palm pose, but constrain the palm pose on the chart $C_{\text{palm}, i}$, which is parallel to chart C_i on the object surface, therefore $\mathbf{T}_{\text{palm}} \in C_{\text{palm}, i}$. Since the equality constraints cannot be solved by using Bayesian optimization, the Alternating Direction Method of Multipliers (ADMM) based Bayesian optimization [21] is utilized to solve the contact pose optimization problem (7) with the new formulation

$$\max_{\mathbf{q} \in \mathcal{B}} f_{\text{obj}}(\mathbf{q}, \mathbf{T}_{\text{palm}}) + g_c(\mathbf{q}, \mathbf{T}_{\text{palm}}), \quad (8)$$

with $g_c(\mathbf{q}, \mathbf{T}_{\text{palm}}) = \mu \sum_{i=0}^n c_i(\mathbf{q}, \mathbf{T}_{\text{palm}})^2$ and $c_i(\mathbf{q}, \mathbf{T}_{\text{palm}}) = |f_{\text{GPIS}}(\text{FK}_i(\mathbf{q}, \mathbf{T}_{\text{palm}}(\mathbf{R}, \mathbf{t})))| - \epsilon_c$. In order to solve (8),

Algorithm 3 ADMM-Based Contact Point Optimization (ADMM-CP-Opt)

Require: $f_{\text{GPIS}}, \mathbf{T}_{\text{palm}}(\mathbf{R}, \mathbf{t}_{\text{palm}})$

- 1: Compute Chart C_i and get $\Phi_i(\mathbf{t}_{\text{palm}})$ solving equation 5
- 2: **for** $n = 0$ to n_{seed} **do**
- 3: $\mathbf{t}' = \mathbf{t}_{\text{palm}} + \Phi_i(\mathbf{t}_{\text{palm}})\mathbf{u}_{\text{rand}}$
- 4: **for** iter = 0 to \max_{iter} **do**
- 5: Solve $\mathbf{q}^{k+1} = \text{BO}_q(f_{\text{obj}}(\mathbf{q}) + \frac{\rho}{2} \left\| \mathbf{q} - \mathbf{z}^k + \frac{\mathbf{y}^k}{\rho} \right\|_2^2)$
- 6: Solve $\mathbf{z}^{k+1} = \text{BO}_z(g_c(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{q}^{k+1} - \mathbf{z} + \frac{\mathbf{y}^k}{\rho} \right\|_2^2)$
- 7: Update $\mathbf{y}^{k+1} = \mathbf{y}^k + \rho(\mathbf{q}^{k+1} - \mathbf{z}^{k+1})$
- 8: Get $\epsilon_1 = \left\| \mathbf{q}^{k+1} - \mathbf{z}^{k+1} \right\|_2, \epsilon_2 = \left\| \rho(\mathbf{z}^{k+1} - \mathbf{z}^k) \right\|_2$
- 9: **if** $\epsilon_1 \leq \epsilon^{\text{primal}}$ and $\epsilon_2 \leq \epsilon^{\text{dual}}$ **then break**
- 10: **end if**
- 11: $\mathbf{y}^k \leftarrow \mathbf{y}^{k+1}, \mathbf{z}^k \leftarrow \mathbf{z}^{k+1}$
- 12: Update BO_q, BO_z
- 13: **end for**
- 14: **end for**
- 15: **return** $\mathbf{q}^{k+1}, \mathbf{T}_{\text{palm}}(\mathbf{R}, \mathbf{t}')$

ADMM introduces an auxiliary variable \mathbf{z} , resulting in

$$\max_{\mathbf{q}, \mathbf{z} \in \mathcal{B}} f_{\text{obj}}(\mathbf{q}, \mathbf{T}_{\text{palm}}) + g_c(\mathbf{z}, \mathbf{T}_{\text{palm}}) \quad (9a)$$

$$\text{s.t. } \mathbf{q} = \mathbf{z}. \quad (9b)$$

In the following, we neglect \mathbf{T}_{palm} in f_{obj} and g_c . By applying Augmented Lagrangian function for equation (9), a new objective function is formulated as

$$\mathcal{L}_{\rho}(\mathbf{q}, \mathbf{z}, \mathbf{y}) = f_{\text{obj}}(\mathbf{q}) + g_c(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{q} - \mathbf{z} + \frac{\mathbf{y}}{\rho} \right\|_2^2 \quad (10)$$

Therefore, we can solve $f_{\text{obj}}(\mathbf{q})$ and $g_c(\mathbf{z})$ by alternating over the following sub problems:

$$\mathbf{q}^{k+1} = \arg \max_{\mathbf{q}} f_{\text{obj}}(\mathbf{q}) + \frac{\rho}{2} \left\| \mathbf{q} - \mathbf{z}^k + \frac{\mathbf{y}^k}{\rho} \right\|_2^2 \quad (11a)$$

$$\mathbf{z}^{k+1} = \arg \max_{\mathbf{z}} g_c(\mathbf{z}) + \frac{\rho}{2} \left\| \mathbf{q}^{k+1} - \mathbf{z} + \frac{\mathbf{y}^k}{\rho} \right\|_2^2 \quad (11b)$$

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \rho(\mathbf{q}^{k+1} - \mathbf{z}^{k+1}). \quad (11c)$$

The optimal condition is defined as $\left\| \mathbf{q}^{k+1} - \mathbf{z}^{k+1} \right\|_2 \leq \epsilon^{\text{primal}}$ and $\left\| \rho(\mathbf{z}^{k+1} - \mathbf{z}^k) \right\|_2 \leq \epsilon^{\text{dual}}$, where ϵ^{primal} and ϵ^{dual} are two predefined optimality tolerances. Each sub problem is solved by using Bayesian optimization. The algorithm is summarized in Alg. 3.

C. Integration of HPP-Opt and ADMM-CP-Opt

The HPP-Opt and ADMM-CP-Opt are in detail introduced in the previous section. This section will integrate both optimizations in one framework and aim to find a near-global optimized grasp. The integration process is summarized in Alg. (4). Since HPP-Opt does not consider the hand finger configuration and the grasp is executed relying on the function of *AutoGrasp* from Graspit! [24], we executed ADMM-CP-Opt by using the result of HPP-Opt. The final solution combines the palm pose from HPP-Opt and the hand finger from ADMM-CP-Opt. It can seem that the ADMM-CP-Opt has at least the one solution as HPP-Opt.

Algorithm 4 Integration of HPP-Opt and ADMM-CP-Opt

Require: $n_{\text{dim}}, n_{\text{iter}}$

- 1: Get LHS $\mathcal{D}_{0:t-1} = \{(\mathbf{x}_0, y_0), \dots, (\mathbf{x}_{t-1}, y_{t-1})\}$
- 2: Fit the Gaussian process Model $p(y|\mathbf{x}, \mathcal{D}_{0:t-1})$
- 3: Optimized hyper Parameter Rrop
- 4: **for** $t = 1$ to n_{iter} **do**
- 5: $\mathbf{x}_{\text{palm},t} = \arg \max_{\mathbf{x}} \text{EI}(\mathbf{x}|\mathcal{D}_{0:t-1})$
- 6: Find an no collision contact, and get $\mathbf{q}, \mathbf{x}_{\text{palm}}$
- 7: $\mathcal{G}_1 = \text{GPISAtlas}(f_{\text{GPIS}}, \mathbf{x}_{\text{palm}})$
- 8: $\{\mathcal{G}_{\text{best}}(\mathbf{x}_{\text{palm},t}, \mathbf{q}), y_{\text{max},t}\} = \arg \max_{\mathbf{q}_1} f_{\text{obj}}(\mathbf{q}, \mathbf{x}_{\text{palm}})$
- 9: $\mathcal{G}_2(\mathbf{x}'_{\text{palm},t}, \mathbf{q}') = \text{Admm-CP-Opt}(f_{\text{GPIS}}, T_{\text{palm}}(\mathbf{x}_{\text{palm},t}))$
- 10: $y'_{\text{max},t} = f_{\text{obj}}(\mathbf{q}', \mathbf{x}'_{\text{palm}})$
- 11: Add new sample $\mathcal{D}_{0:t} = \{\mathcal{D}_{0:t-1}, (\mathbf{x}'_{\text{palm},t}, y'_{\text{max},t})\}$
- 12: update Gaussian process model (GP)
- 13: **end for**
- 14: $\mathbf{x}_{\text{palm}}^+ = \arg \max_{\mathbf{x}_i \in \mathcal{X}_{0:t}} f_{\text{obj}}(\mathbf{x}_i)$
- 15: **return** $\mathbf{x}_{\text{palm}}^+$

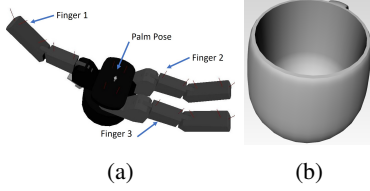


Fig. 3: Visualization of (a) Barret hand and (b) Object mug.

V. EXPERIMENT

Simulation results are introduced in this section to verify the effectiveness of our algorithm. The experiment is executed in the platform Graspit! [24] by using Barret hand, as shown in Fig. 3a. Barret hand has three fingers, finger one has two joints, where the last joint is under-actuated. Finger two and three have one common joint. Each finger has three joints where both fingers have a passive joint. Therefore the Barret hand fingers have totaled 4 DOFs. The hand palm pose is denoted as a 6-dimensional vector. Therefore the whole Barret hand system has totaled 10 DOFs. The experiment's graspable object is stored in the mesh file, such as one example: mug in Fig. 3b. The GPIS describes the object by using the mesh triangle vertices as the input. Our approach achieves a 95% success rate on various commonly used objects with diverse appearances, scales, and weights compared to the other algorithm. All evaluations were performed on a laptop with a 2.6 GHz Intel Core i7-6700HQ and 16 GB of RAM.

A. Experiment on HPP-Opt

In this section, the algorithm HPP-Opt is compared with other grasp planning. The first grasp planning approach is a simulated annealing grasp planner using an auto grasp quality energy as a search strategy, which behaves like a random grasp planning. The second approach uses an EigenGrasp planner combined with a simulated annealing solver to guide potential quality energy. The Bayesian optimization and simulated annealing both are global optimization solvers, where

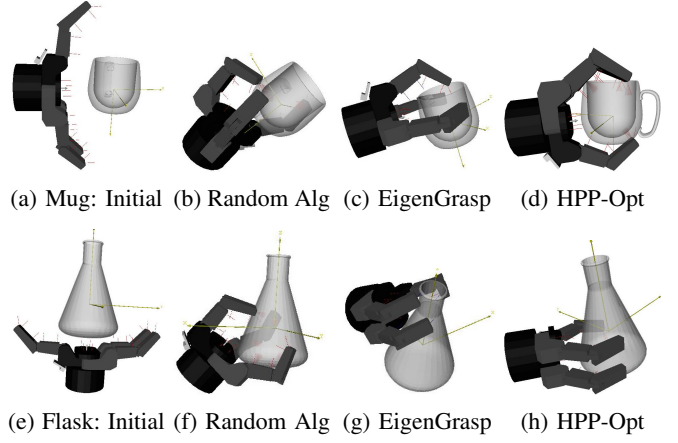


Fig. 4: Comparison of different Grasp planning Algorithm examples for different objects with the multi-fingered hand

TABLE I: Evaluation of different grasp planning algorithm for all data sets. The results is an average of first 20 best Grasp Candidates. A greater value of epsilon and volume means a more stable grasp. The best result is highlighted in *green*

quality	Algorithms							
	random Alg		EigenGrasp		HPP-Opt		ADMM-CP-Opt	
	q_e	q_{volume}	q_e	q_{volume}	q_e	q_{volume}	q_e	q_{volume}
Mug	0	9.5352e-05	1.7281e-04	7.1981e-04	0.0555	0.0097	0.06	0.0161
Flask	0	3.9603e-05	4.9050e-04	2.0142e-04	0.0107	0.0014	0.0107	0.0026
Phone	0	3.9209e-05	0.0017	4.6942e-04	0.0142	0.0035	0.0142	0.0042
Sphere	0.0042	0.0019	0	0.0011	0.0495	0.0121	0.050	0.0279
Bishop	0	8.1935e-05	0.0012	9.7708e-05	0.0094	0.0011	0.0094	0.0016

the latter one is a probabilistic technique for approximating the global optimum of a given function. We compare the algorithms with different kinds of shapes. The experiment is conducted in a fixed time of 20 seconds, and we average the first 20 best grasp candidates. The best grasp candidate of mug and flask from whole grasp candidates are visualized in Fig. 4. The initial state of Barret hand and object are randomly defined, shown in Fig. 4a and 4e. It can be seen that the hand configuration selected by random grasp planning is skewed to the object, and contact points on the surface are not properly, and the resulting quality is also worse. The results of EigenGrasp show a better solution where the palm pose is trying to parallel to the object surface. The palm pose selected by HPP-Opt is more reasonable in comparison to the other two algorithms. We show the quantitative result is in table (I). In different kinds of geometry shapes, our algorithm can achieve a much more stable grasp than other algorithms. The epsilon quality achieved by the first approach is almost zero besides in the case of a sphere. The epsilon quality by Eigen grasps a minimal value. On average, the epsilon quality of HPP-Opt is 28.5 times greater than Eigen grasp's epsilon quality. And the HPP-Opt's volume quality is 32.1417 times greater than Eigen Grasp's volume quality. Furthermore, we show the best grasp candidate from the first 20 best grasp candidates in Table II.

TABLE II: The best grasp from 20 grasp candidates. The best result is highlighted in *green*

quality	Algorithms							
	random Alg		EigenGrasp		HPP-opt		ADMM-CP-Opt	
	q_e	q_{volume}	q_e	q_{volume}	q_e	q_{volume}	q_e	q_{volume}
Mug	0	0.0012	0.0065	0.0011	0.1039	0.0151	0.1158	0.0340
Flask	0	0.0004	0.0130	0.0031	0.0449	0.0018	0.0452	0.0019
Phone	0	0.0004	0.0310	0.0007	0.0208	0.0198	0.0335	0.0006
Sphere	0.0844	0.0301	0	0.0086	0.0914	0.0362	0.1676	0.0728
Bishop	0	0.0012	0.0164	0.0004	0.0390	0.0020	0.0444	0.0045

B. Experiment of integration HPP-Opt and ADMM-CP-Opt

In section V-A, hand palm pose is optimized based on the Bayesian optimization algorithm combining with local adaption, and hand finger configuration is set based on the function of *AutoGrasp* from Graspit [24]. The principle of *AutoGrasp* is to close each hand finger DOF at a rate equal to a predefined speed factor multiple with its default velocity, and the movement is stopped at the contact point. Therefore ADMM-CP-Opt is used to assist the HPP-Opt to find a better hand finger configuration. The comparison result is visualized in Fig. 5. In the case (5a), fingers two and three are too close and grasp the bottom of the flask. As a consequence, the resulting triangle has a small internal angle. Applying ADMM-CP-Opt splits the fingers two and three by maximizing epsilon and volume quality and makes the resulting triangle closer to the equilateral triangle with a more stable grasp. The same improvement happen in the Fig. 5b - Fig. 5f as well. The solution founded by ADMM-CP-Opt is trying to make the resulting triangle as closer as the equilateral triangle. In Table I, ADMM-CP-Opt improves the volume quality of HPP-Opt. And In table II, ADMM-CP-Opt shows a better grasp than HPP-Opt in most cases under epsilon and volume quality metrics.

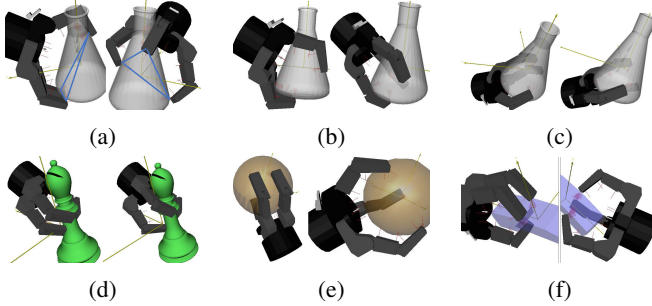


Fig. 5: The comparison result of HPP-Opt and ADMM-CP-Opt. In each sub figure, the left one is result of HPP-Opt, and the right one is improved by using ADMM-CP-Opt

VI. CONCLUSION AND OUTLOOK

We propose a new algorithm for grasp planning with multi-fingered hands by optimizing the hand palm pose and hand finger configuration separately. Using a global Bayesian optimization solver, no initial configuration is required, which shows superiority over the convex optimization solver. We propose a dual-stage optimization process by considering

the independence of the hand palm pose and finger configuration. In the first stage, we utilize a GPIS to describe the graspable object so that collision checking of contact points can be integrated into the optimization framework. Furthermore, the chart on the object surface can be computed using the GPIS, which can be used to explore the local information of objects. Two ellipsoids are used to define the palm pose constraints domain. Relying on the first stage result, we apply an ADMM based Bayesian optimization to optimize the contact points. The whole process will switch between HPP-Opt and ADMM-CP-Opt. We collect the best 20 Grasps, and the final grasp is selected by ranking the grasp candidates under consideration of epsilon and volume quality. In this work, we describe only the object in GPIS. In future work, we can describe the hand part into GPIS so that the collision checking can be converted to a problem by querying the distance between two surfaces. Besides, the robot arm is not considered in the grasping scenario. It will be interesting to integrate the constraints of robot arm manipulability in the objective function.

REFERENCES

- [1] A. Perzylo, M. Rickert, B. Kahl, N. Somani, C. Lehmann, A. Kuss, S. Profanter, A. B. Beck, M. Haage, M. R. Hansen, M. T. Nibe, M. A. Roa, O. Sornmo, S. G. Robertz, U. Thomas, G. Veiga, E. A. Topp, I. Kessler, and M. Danzer, "SMERobotics: Smart robots for flexible manufacturing," *IEEE Robotics and Automation Magazine*, vol. 26, no. 1, pp. 78–90, 2019.
- [2] M. Tenorth, A. Perzylo, R. Lafrenz, and M. Beetz, "The RoboEarth language: Representing and exchanging knowledge about actions, objects and environments," in *Proc. of the IEEE Intl. Conf. on Robotics and Automation*, May 2012, pp. 1284–1289.
- [3] A. Perzylo, N. Somani, M. Rickert, and A. Knoll, "An ontology for cad data and geometric constraints as a link between product models and semantic robot task descriptions," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, Sept. 2015, pp. 4197–4203.
- [4] J. Lin, M. Rickert, and A. Knoll, "6D pose estimation for flexible production with small lot sizes based on cad models using gaussian process implicit surfaces," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.
- [5] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000, pp. 348–353.
- [6] J. Bohg, A. Morales, T. Asfour, and D. Kragic, "Data-driven grasp synthesis—A survey," *IEEE Transactions on Robotics*, vol. 30, no. 2, pp. 289–309, 2013.
- [7] O. Williams and A. Fitzgibbon, "Gaussian process implicit surfaces," in *Proceedings of the Gaussian Processes in Practice Workshop*, June 2006.
- [8] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2009, pp. 1710–1716.
- [9] M. T. Ciocarlie, "Low-dimensional robotic grasping: Eigengrasp subspaces and optimized underactuation," Ph.D. dissertation, Columbia University, 2010.
- [10] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2003, pp. 1824–1829.
- [11] R. Pelossof, A. Miller, P. Allen, and T. Jebara, "An svm learning approach to robotic grasping," in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2004, pp. 3512–3518.
- [12] E. Jang, S. Vijayanarasimhan, P. Pastor, J. Ibarz, and S. Levine, "End-to-end learning of semantic grasping," *arXiv preprint arXiv:1707.01932*, 2017.
- [13] S. Levine, P. Pastor, A. Krizhevsky, J. Ibarz, and D. Quillen, "Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection," *The International Journal of Robotics Research*, vol. 37, no. 4–5, pp. 421–436, 2018.

- [14] J. Mahler, F. T. Pokorny, B. Hou, M. Roderick, M. Laskey, M. Aubry, K. Kohlhoff, T. Kröger, J. Kuffner, and K. Goldberg, “Dex-net 1.0: A cloud-based network of 3d objects for robust grasp planning using a multi-armed bandit model with correlated rewards,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 2016, pp. 1957–1964.
- [15] D. Kalashnikov, A. Irpan, P. Pastor, J. Ibarz, A. Herzog, E. Jang, D. Quillen, E. Holly, M. Kalakrishnan, V. Vanhoucke, *et al.*, “Qt-opt: Scalable deep reinforcement learning for vision-based robotic manipulation,” *arXiv preprint arXiv:1806.10293*, 2018.
- [16] J. Varley, C. DeChant, A. Richardson, J. Ruales, and P. Allen, “Shape completion enabled robotic grasping,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2017, pp. 2442–2447.
- [17] A. ten Pas, M. Gualtieri, K. Saenko, and R. Platt, “Grasp pose detection in point clouds,” *The International Journal of Robotics Research*, vol. 36, no. 13-14, pp. 1455–1473, 2017.
- [18] H. Liang, X. Ma, S. Li, M. Görner, S. Tang, B. Fang, F. Sun, and J. Zhang, “Pointnetgpd: Detecting grasp configurations from point sets,” in *Proceedings of the International Conference on Robotics and Automation*, 2019, pp. 3629–3635.
- [19] J. Nogueira, R. Martinez-Cantin, A. Bernardino, and L. Jamone, “Un-scented bayesian optimization for safe robot grasping,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 1967–1972.
- [20] C. Ferrari and J. F. Canny, “Planning optimal grasps,” in *Proceedings of the IEEE International Conference on Robotics and Automation*, 1992, pp. 2290–2295.
- [21] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, *et al.*, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.
- [22] D. R. Jones, M. Schonlau, and W. J. Welch, “Efficient global optimization of expensive black-box functions,” *Journal of Global Optimization*, vol. 13, no. 4, pp. 455–492, Dec 1998.
- [23] M. Blum and M. A. Riedmiller, “Optimization of gaussian process hyperparameters using Rprop,” in *Proceedings of the European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, Apr. 2013, pp. 339–344.
- [24] A. T. Miller and P. K. Allen, “Graspit! a versatile simulator for robotic grasping,” *IEEE Robotics and Automation Magazine*, vol. 11, pp. 110–122, 2004.