Hierarchical Fingertip Space for Multi-fingered Precision Grasping

Kaiyu Hang, Johannes A. Stork and Danica Kragic

***Abstract*— Dexterous in-hand manipulation of objects benefits from the ability of a robot system to generate precision grasps. In this paper, we propose a concept of Fingertip Space and its use for precision grasp synthesis. Fingertip Space is a representation that takes into account both the local geometry of object surface as well as the fingertip geometry. As such, it is directly applicable to the object point cloud data and it establishes a basis for the grasp search space. We propose a model for a hierarchical encoding of the Fingertip Space that enables multilevel refinement for efficient grasp synthesis. The proposed method works at the grasp contact level while not neglecting object shape nor hand kinematics. Experimental evaluation is performed for the Barrett hand considering also noisy and incomplete point cloud data.**

I. Introduction and Contributions

Research in robotic grasping ranges from the sensory perception problem [1]–[3] to task level grasp planning [4]. For applications such as dexterous in-hand manipulation, precision grasping is a necessary requirement [5]–[9]. The synthesis of precision grasps has been in particular addressed in [10]–[14] but in a rather limited manner. In this paper, we address the problem of generating precision grasps on objects of complex shapes and propose the following:

- A concept of *Fingertip Space* – an integrated representation of object/fingertip contacts space that takes into consideration both local object geometry and fingertip shape. It directly operates on the object point cloud and establishes a basis for the grasp search space.
- A hierarchy of the *Fingertip Space* for multilevel refinement of grasps allowing for an efficient search of stable grasps.

Our work is motivated by the fact that most of the contemporary object representation approaches concentrate on the global rather than local surface properties and are therefore not suitable for generating precision grasps. Examples include Reeb graph [15], Medial Axis [16], topological features [17], primitive shapes [3], [18] and approximated parametrized volumes [19]–[21]. Inspired by [22], which has proven that nearby grasps with certain bounded contact differences are also bounded in grasp quality, it enables us to consider *Fingertip Unit*, which preserves surface local information, as the basis of *Fingertip Space*. On the other hand, hierarchical representation of the *Fingertip Space* provides a multi-resolution global view of the target object to facilitate the grasp planning in an efficient way. In comparison with the widely used sampling based precision grasp planners [11], [23], [24], our representation makes the grasp planning more reliable on complex shapes. Moreover, reachability is an important component ensuring that the synthesized grasp is applicable [25], [26]. By sampling and encoding feasible hand configurations, we approximate the reachability manifold non-parametrically to produce reachable grasps. Finally, the execution of the synthesized grasp is computed similarly to [27].

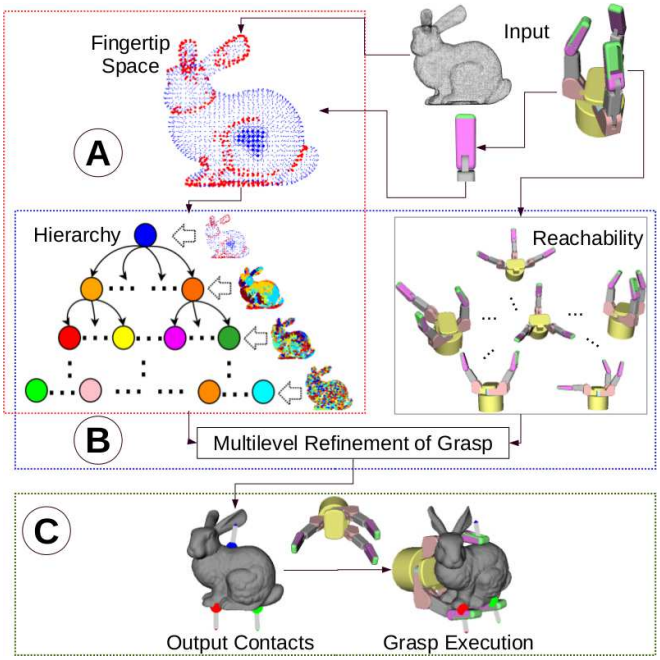

Fig. 1. System pipeline: Given an object point cloud and a robotic hand as input, our system (A) extracts a fingertip space directly from the object point cloud and builds a hierarchical representation of it. (B) By incorporating the fingertip space hierarchy and a hand reachability measure, the multilevel refinement procedure searches for a feasible combination of contacts with an initial hand configuration. (C) In the end, the synthesized grasp is realized by local contact positions optimization with respect to the synthesized contacts.

The system pipeline is depicted in Fig. 1: we assume that the friction coefficients are known and that the center of mass of the object is the centroid of its point cloud. The rest of this paper is organized as follows: In Sec. II, we formulate the problem of precision grasp synthesis in the context of our system. In Sec. III, we introduce the extraction of *Fingertip Space* and its hierarchy, which is shown as block (A) in Fig. 1. Multilevel refinement of grasps, shown as block (B), is described in Sec. IV along with *Stochastic Hill Climbing*. In Sec. V, we describe details of our system implementation and grasp execution (Block (C)) and present the experimental evaluation. We conclude the work together and introduce ideas for the future work in Sec. VI.

The authors are with the Computer Vision and Active Perception Lab, Centre for Autonomous Systems, School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden, {kaiyuh, jastork, dani}@kth.se. This work was supported by FLEXBOT (FP7-ERC-279933).

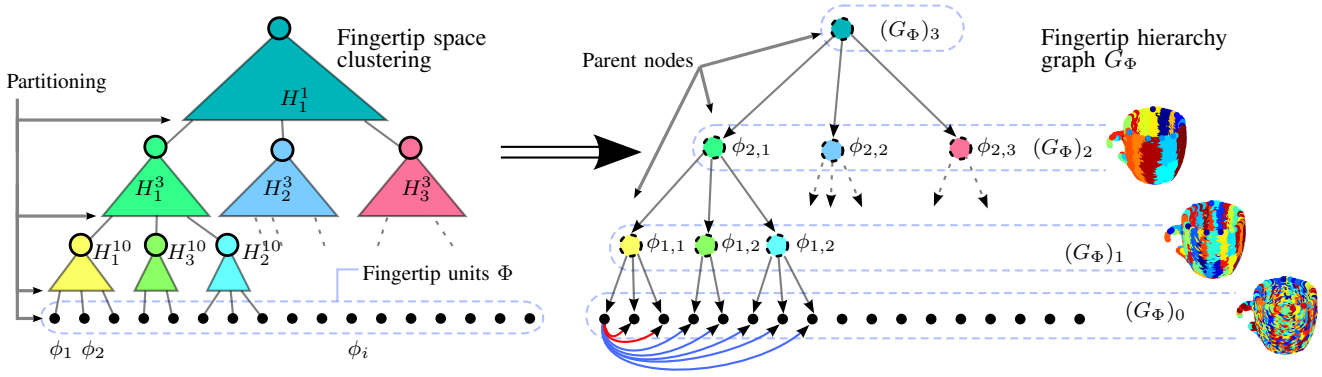

Fig. 2. Formation of the fingertip hierarchy exemplified for four levels. *Left:* An AHC clustering tree is used to retrieve a partitioning of the fingertip space into $|\Phi|$, 10, 3, and 1 cells. For each cell a circle symbolizes the representative fingertip unit. *Right:* The representative units are used as parents in a DAG. Edges to siblings (in red) and to cousins (in blue) are only shown for the fingertip unit $\phi_1 = \phi_{0,1}$.

II. Problem Formulation and Notation

We begin by presenting the notation in the table below and then continue with the formalization of the problem.

n_g	Number of fingers of a robot hand
$\mathcal{L}$	Set of all contact locations
$g = (C_g, Joint_g, Pose_g)$	Fingertip grasp with contacts C_g, joint values $Joint_g$ and hand pose $Pose_g$
$\Phi = \{\phi_i\}_i$	Fingertip Space of fingertip units ϕ_i
$d_\Phi \colon \Phi \times \Phi \to \mathbb{R}$	Distance measure on fingertip units
$\bar{\Phi} = \Phi \dot\cup \{\phi_{i,j}\}_{i,j}$	Fingertip Space with additional representative parents $\phi_{i,j}$
G_Φ	Fingertip Hierarchy graph
$(G_\Phi)_i$	ith hierarchy level induced by G_Φ
$\mathcal{S}$	Fingertip grasp solution space
$\mathcal{S}_i$	ith hierarchy level solution space
$Q \colon C_g \to \mathbb{R}$	Grasp quality function
$R \colon C_g \to \mathbb{R}^+$	Reachability function
$\mathcal{P} = \{(p_i, n_i)\}_i$	Point cloud with unit length normals
$N_{p_i}^r$	Point neighborhood within radius r
$H_k^{z_i} \subseteq \Phi$	Cell k of a z_i-element partitioning of Φ
$\widehat{\mathcal{M}} = \{m_{\hat{g}}\}_{\hat{g}}$	Sampled reachability manifold of sampled grasps $\hat{g}$

For synthesizing precision grasps, which we refer to as fingertip grasps in this work, we next introduce how to construct a representation that hierarchically integrates global and local features of the object and fingertips based on the *Fingertip Space*. Starting from the top level of the hierarchy, our system starts from an initial grasping pose and then optimizes the contacts through the hierarchy in a coarse-to-fine manner to finally produce a stable and reachable fingertip grasp.

A. Fingertip Grasps

We consider fingertip grasps for a hand with n_g fingers, formalized as the tuple $g = (C_g, Joint_g, Pose_g)$. We refer to contacts between the robot fingertips and the object as $C_g = \{c_i\}_{i=1}^{n_g}$, the values for the end effector joints as $Joint_g$, and the position and orientation of the hand as $Pose_g$. A fingertip grasp provides one individual contact, c_i, for each fingertip which means that for each finger there exists one location $l_i \in \mathcal{L}$ on the object that is in contact with the fingertip. If no value for $Joint_g$ exists ensuring that the fingertips can exert force onto the object via the individual contacts, C_g is considered *not reachable*.

B. Fingertip Space

The set of potential contact locations, $\mathcal{L}$, is large but many locations are not viable due to the local surface geometry. To keep grasp synthesis tractable, we propose a finite discrete set of locations on the object, $\Phi = \{\phi_i\}_i$ that consists of only viable locations and denote it as *Fingertip Space*. The elements, ϕ_i of this space are named *Fingertip Units*. Thus, viable grasp locations take into account local object surface *and* fingertip geometry. Sec. III provides more details on this.

Provided a similarity measure for fingertip units, it is possible to assign structure to the space $\Phi \subset \bar{\Phi}$ in form of a directed acyclic graph $G_\Phi = \left(\bar{\Phi}, E_{\bar{\Phi}}\right)$. We define G_Φ such that similar fingertip units are pairwise connected by edges and introduce new parent units as representatives of all their descendants in Φ. The set $\bar{\Phi} = \Phi \dot\cup \left\{\phi_{i,j}\right\}_{i,j}$ consists of fingertip units and introduced ancestor units. The symbol $\phi_{i,j}$ with $i > 0$ denotes the jth parent unit in the ith level of the hierarchy, representing all fingertip units that are commonly represented by its children in the $(i-1)$th level. As shown in Fig. 2 on the right, the resulting hierarchy of fingertip space has a single root and similar fingertip units are connected. In the graph, we refer to members of Φ as elements $\phi_{0,j}$ and denote with $(G_\Phi)_i = \bigcup_j \phi_{i,j}$ the ith hierarchy level induced by G_Φ, e.g. especially we have $(G_\Phi)_0 = \Phi$. Connected nodes from the same level are neighbors, and connecting pairs of parent units can be exploited (see Sec. III-B).

The graph G_Φ and the induced hierarchy levels $(G_\Phi)_i$ form our object representation. For fingertip grasp synthesis, the above definitions efficiently provide relevant information by adding an explicit similarity-based structure to the fingertip units space: *i)* Similar units are directly connected.

ii) Dissimilar units are found by considering the units represented by dissimilar distant ancestors. *iii)* An increasingly coarser representation is found by considering the members or levels further up in the hierarchy. *iv)* Similar fingertip units are collected under a common parent.

C. Fingertip Grasp Selection by Optimization

To synthesize a feasible fingertip grasp on an object, it is necessary to select locations from $\mathcal{L}$ that afford stable contacts and ensure reachability. We first search for stable and reachable contacts and after that check if there are solutions for $Joint_g$ and $Pose_g$ that realize the grasp. By approximating the set of all possible contacts with Φ as described in Sec. II-B, we can formalize the contacts as $C_g = \left(\phi_1, \phi_2, \ldots, \phi_{n_g}\right) \in \mathcal{S}$. Thereby, we denote $\mathcal{S} = \prod_{k=1}^{n_g} \Phi_k$ as the solution space consisting of fingertip spaces of different robot fingers Φ_k. All surface locations that do not support the placement of a specific fingertip are disregarded.

Given a measure of grasp quality in terms of fingertip units $Q\left(C_g\right) \in \mathbb{R}$ and a measure of reachability, $R\left(C_g\right) \in \mathbb{R}^+$ we can formulate an optimization objective in terms of solution space elements as $\theta\left(C_g\right) = Q(C_g) + \alpha R(C_g)$. Here, we assumed perfect reachability for $R(C_g) = 0$ and set $0 > \alpha \in \mathbb{R}$. The optimization problem is then given as

$$C_g^* = \underset{C_p \in \mathcal{S}}{\operatorname{argmax}} \quad \theta\left(C_g\right) \tag{1}$$

Concretely, we do not solve Eq. 1 directly but formulate a hierarchy of increasingly approximated problem instances as explained Sec. IV-C. Grasp synthesis is finalized in continuous coordinates by inverse kinematics for the selected contacts $C_g^* \in \mathcal{S}$ as described in Sec. IV-D. If the resulting grasp is obstructed or not reachable, we start a new search with a different initialization.

III. Fingertip Space Representation

In this section we explain fingertip unit extraction from arbitrary point clouds with normals using a simple finger model. We first provide a definition of fingertip units in terms of input data and elaborate on the fingertip hierarchy which is used in Sec. IV-C.

A. Extraction of Fingertip Units

In Sec. II-B we only state a qualitative definition of fingertip units as locations on the object that allows the placement of a fingertip. Observing an arbitrary point cloud $\mathcal{P} = \left\{(p_i, n_i)\right\}_i$ with normal vector estimates, we need to extract a finite set of such locations by investigating $\mathcal{P}$ while taking a finger model into account. For the purpose of this work, we focus on Barrett hand and consider contacts where the inside of the distal links rests on the object surface. The fingertip model describes a flat circular region located at the center of the distal link's inner surface and has radius r, as shown in Fig. 3.

For a point p_i to support the placement of the fingertip orthogonal to its normal n_i, all neighboring points within radius r need to support the fingertip as well. This can be formulated as a limit criterion on variance of point positions

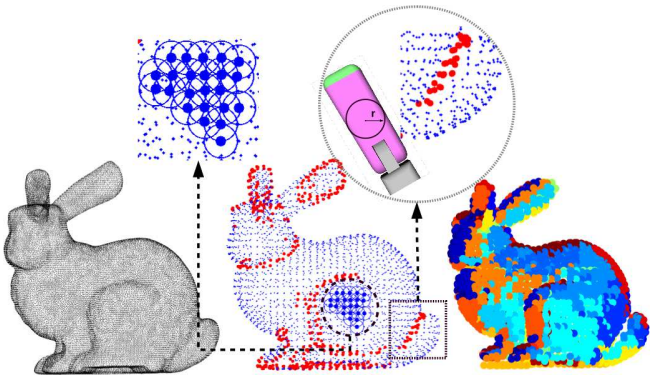

Fig. 3. From *left* to *right:* Point cloud, fingertip space (in blue), and rejected points (in red). Partitioning of similar fingertip unites into 20 cells. *Magnified:* Red marks points rejected due to variance criterion. Comparing to finger size, it is obvious that the red positions cannot stabilize contacts.

and normals of the point neighborhood, $N_{p_i}^r = \{(p_j, n_j) \in \mathcal{P} \mid \left\|p_i - p_j\right\| < r\}$, and used to reject unfit points. Fingertip space and units are now defined in terms of the following filter:

$$\Phi = \left\{(p_i, n_i) \in \mathcal{P} \mid \operatorname{Var}\left(N_{p_i}^r\right) < \lambda\right\} \tag{2}$$

A fingertip unit is thus a position and a normal, $\phi = (p, n)$, where the neighborhood N_p^r satisfies the statistical variance criterion Var.

B. Hierarchy of Fingertip Space

In Sec. II-B we have augmented the fingertip space with a similarity-based graph G_Φ and the hierarchy levels $(G_\Phi)_i$ it induces. To compute a G_Φ, we employ agglomerative cluster analysis where each fingertip unit initially forms a singleton cluster. Agglomerative Hierarchical Clustering (AHC) of the set Φ with the distance measure d_Φ is a bottom up procedure that results in a clustering tree by iteratively merging the two most similar clusters. This dendrogram can be accessed to obtain a partitioning of Φ into $z \in \mathbb{N}$ clusters or cells, e.g. $\Phi = H_1^z \dot{\cup} H_2^z \dot{\cup} \ldots \dot{\cup} H_z^z$. For a partitioning of $z' > z$ cells each of the cells $H_i^{z'}$ is strictly contained in exactly one of the cells H_j^z.

We exploit this property to construct the hierarchy in G_Φ by computing a sequence of l partitions with $|\Phi| = z_0 > z_1 > \cdots > z_l = 1$ number of cells. For each cell $H_j^{z_i} \subseteq \Phi$ with $i > 0$, we create a representative fingertip unit $\phi_{i,j} \in \bar{\Phi}$ from the median position and the mean normal of all contained fingertips units. Parent-child edges are introduced for each two fingertips units $\phi_{i,j}$ and $\phi_{i-1,k}$ with $i > 0$ if the child's cell is contained in the parent's cell.

$$\forall\, H_k^{z_{i-1}} \subseteq H_j^{z_i} \colon (\phi_{i,j}, \phi_{i-1,k}) \in E_{\bar{\Phi}} \tag{3}$$

Additionally we connect all siblings nodes and introduce edges to all nodes who's parents are siblings. This process is exemplified in Fig. 2.

Concretely, we are interested in grasp similarity for search and require similar fingertip units to be grouped together. The

admittedly crude fingertip distance measure of Eq. 4 provides plausible results in terms of positions and normals.

$$d_\Phi(\phi_i, \phi_j) = \left\| (p_i - p_j) + \eta(n_i - n_j) \right\| \tag{4}$$

The parameter $\eta \in \mathbb{R}^+$ balances position and normal. Larger η induces more parallel or flat geometry and small η results in compactly shaped cells but allows more normal vector variance. Furthermore, we need to specify the number of levels l and the number of nodes per level z_i in order to get cutoff values from AHC. For simplicity, we base the number of nodes per level on an incrementation ration and m_{l-1}. Note that $m_l = 1$ and $m_0 = |\Phi|$. Fig. 3 shows 20 cells of different size with $\eta = 3$.

IV. Grasp Synthesis

In Sec. II-C we have stated the discrete version of our grasp synthesis as a combinatorial optimization problem. This section serves to describe our choice of reachability measure R and grasp quality function Q for Eq. 1. We also describe the optimization procedure using the multilevel refinement metaheuristic.

A. Grasp Stability Metric

All functions in Eq. 1 are defined on sets of fingertip units $\phi_i = (p_i, n_i)$. It is therefore convenient to focus on quality measures for point contacts as many approaches to robotic grasping are based on force analysis and the concept of force-closure [28], [29]. There, the forces exerted by the robot and friction of the surfaces are considered. For Q, we choose to evaluate the force-closure property of a grasp with the L^1 grasp quality measure Q_μ reported in [11] that employs the Coulomb friction model. The grasp quality $Q = Q_\mu$ is a function of all contact positions and normals, the center of mass of the object and the friction coefficient $\mu \in \mathbb{R}^+$. We can thus directly refer to the fingertip units for point contacts $p_i \in \mathbb{R}^3$ and inward-pointing unit surface normals $n_i \in \mathbb{R}^3$. A grasp is force-closed if Q_μ is larger than zero.

B. Reachability Measure

For the optimization objective in Eq. 1 we require a non-binary reachability measure R of a set of fingertip units that relates to values for $Joint_g$ and $Pose_g$. However, computing an approximate inverse kinematic solution and measuring the residual error in each optimization step is computationally infeasible. Instead, we consider an approximation of the fingertip reachability manifold $\widehat{\mathcal{M}}$ and assume a free-floating hand model. Feasible hand configurations are generated by rejection sampling and their fingertip positions and normals are recorded in a affine invariant encoding. To this end, we retain a vector, $m_{\hat{g}}$, of pairwise fingertip distances and normal differences from a sampled grasp $\hat{g}$ and keep the associated $Joint_{\hat{g}}$ values. For a grasp hypothesis C_g we can calculate the encoding m_g and access the nearest neighbor in encoding space $m_{\hat{g}} \in \widehat{\mathcal{M}}$. If the manifold is sampled sufficiently, the differences between m_g and $m_{\hat{g}}$ can be consider as the reachability residual

$$R(C_g) = \left\| m_g - m_{\hat{g}} \right\| \tag{5}$$

This reachability measure relies on the distances in an encoding space and we are aware that better techniques exists, e.g., density estimation as in [30] that takes into account also object-level impedance control. However, our R is used in a heuristic way to reduce the searched space for Eq. 1 and to initialize hand configurations $Joint_g := Joint_{\hat{g}}$ and has served sufficiently for this purposes.

C. Multilevel Refinement Optimization

As in our previous work [31], we apply the multilevel refinement metaheuristic [32] for a hierarchy of combinatorial optimization problems. The fingertip space object representation defined in Sec. II-B offers a multiresolution view of the object and can be exploited for refinement search. We refer to the fingertip hierarchy levels $(G_\Phi)_i$ to form increasingly approximated instances of the solution space. This is achieved by defining $\mathcal{S}_i = \prod_{k=1}^{n_g} \left(G_{\Phi_k}\right)_i$ as the solution space on the ith refinement level. On each level i, a solution C_g^i is initialized by extending the solution of the previous level $i+1$ and optimizing it, resulting with a solution $C_g^* = C_g^0$ in the search space $\mathcal{S}_0 = \mathcal{S}$.

In this context the individual optimization problems are usually optimized using local optimization methods for convex problems [33]. As argued in our previous work, [31] we cannot expect convex objective manifolds for complex objects. Furthermore, the result of hill climbing techniques is heavily dependent on initialization in non-convex solution spaces. Different algorithms [34], [35] have been proposed to escape local minima. In this work, we adopt stochastic hill climbing [36].

In this algorithm, the objective function is not directly used to improve the current solution. Instead, the change between two solutions C_g and $C_{g'}$ is conditioned on the probability stated in Eq. (6).

$$Pr(C_g, C_{g'}) = \left(1 + \exp \frac{\theta(C_g) - \theta(C_{g'})}{\zeta}\right)^{-1} \tag{6}$$

The search randomness is determined by ζ. Large values make the steps completely random, whereas the algorithm degenerates to hill climbing when ζ is very small.

Our basic optimization procedure is shown in Alg. 1. The function $\mathrm{rand}(0,1)$ produces uniformly distributed real numbers between 0 and 1. Children and neighbors of grasps are created from the respective children and neighbors of the constituting fingertip units in the graph G_Φ as defined in Sec.II-B.

D. Grasp Realization

The optimization procedure described in Sec. IV-C results in a grasp comprised of discrete fingertip units C_g^*. For this grasp, the sampling-based reachability measure from Sec. IV-B provides the joint configuration $Joint_{\hat{g}}$ of the closest recorded grasp $\hat{g}$ in encoding space. The optimization procedure described below employs a continuous optimization for $Joint_g$ and $Pose_g$ in terms of C_g^* to close the gap between discretization, sampling and applicable continuous solutions. This is achieved by first approximately aligning the hand to

Algorithm 1 Multilevel refinement with stochastic hill climbing for grasp synthesis

Input: ζ, maxIter, G_Φ, θ
Output: grasp g

for $i = l-1$ to 0 **do**
  if $i = l-1$ **then** ▷ Initialization
    $C_g^i \leftarrow$ random from $\mathcal{S}_i$
  else ▷ Extension
    $C_g^i = \underset{C_g \text{child of } C_g^{i+1}}{\text{argmax}} \ \theta(C_g)$
  end if
  for 1 to maxIter **do** ▷ Refinement
    $C_g \leftarrow$ some neighbor of $C_g^i \in \mathcal{S}_i$
    if $Pr(C_g^i, C_g) \geq \text{rand}(0,1)$ **then**
      $C_g^i \leftarrow C_g$
    end if
  end for
end for

the grasping pose with an affine transform between C_g^* and fingertips of $\hat{g}$, and then locally optimizing simulated contact positions.

The initial affine transform can be found by minimizing the Euclidean error for the known correspondences between C_g^* and the fingertips of $\hat{g}$. An example of initial hand alignment is shown in Fig. 4. As can be seen in Fig. 4, not all fingers have an initial single surface contact. For this reason, we first open the colliding fingers using proportional joint value increments and then close all fingers until contact to get simulated contact positions C_g^+. We then turn to gradient decent to minimize the error between the positions of C_g^+ and C_g^* for which we compute the gradient numerically.

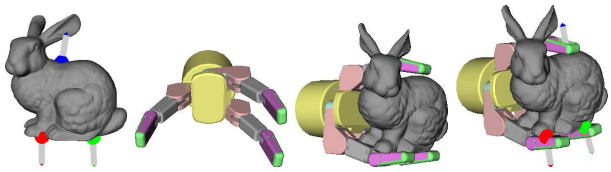

Fig. 4. An example of initial alignment and grasp realization. *Left to right:* Marker positions represent C_g^*. The initial joint values for the grasp from $\widehat{\mathcal{M}}$. Hand alignment by affine transform. Final grasp after contact optimization.

V. Experimental Evaluation

In this section, we first provide implementation details and then present the results of evaluation. The evaluations have been conducted in OpenRave [24] on six objects : *Stanford Bunny* [37], *Plane* [38] and *Waschmittel* [39], as well as *Cup, Spoon* and *Milk Box* scanned by ourself.

A. Implementation Details

As described in Sec. II, *Fingertip Units* are locations on the object surface where fingertip contacts are viable and the *Fingertip Space* is a finite set of *Fingertip Units*. If two contacts have similar locations and orientations, they would have similar contributions to the grasp stability [22]. Therefore, prior to fingertip space extraction, we uniformly subsampled the object point cloud to produce fingertip unit candidates. Concretely, the subsampling was done in the scale of half of the fingertip unit size on the point cloud.

For the reachability measure $R(C_g)$ in the objective function $\theta(C_g)$, a set of feasible grasps were sampled and encoded. For the sake of efficiency, we saved all the codes in a kd-tree and consider the Euclidean distance between codes as the reachability residual.

After Alg. 1 has synthesized a grasp hypothesis, we discard the hypothesis and restart the algorithm if: a) grasp hypothesis is unstable, or b) the reachability residual is too large or c) it is not collision-free. The collision is checked by firstly aligning the configured robot hand to the grasping pose by the affine transform described in Sec. IV, and then in the simulation check whether the hand has collisions at positions other than the fingertips.

In all the experiments shown below, we use $l = 4$ layers, $m_{l-1} = 20$ and $\eta = 1$ for constructing the hierarchy of the fingertip space. We set $\alpha = 0.4$ to weight between $Q(C_g)$ and $R(C_g)$ in $\theta(C_g)$, and set $maxIter = 100$ for grasp refinement.

B. Fingertip Space Extraction and System Evaluation

Different definitions of fingertip units result in different fingertip spaces. As described in Sec. III, a fingertip unit in this work is defined as a circular area at the center of the distal link and has radius r. In this section, we show two different definitions of fingertip units of the Barrett hand and their corresponding fingertip spaces, and then we evaluate the performance of the system using these two fingertip spaces respectively.

In Fig. 5, the fingertip unit is located at the center of the distal link in both rows expressing the representative position of a fingertip. In the upper row, the radius of the circular area is the distance between the center and the long edge of the distal link, denoted as r_1, whereas in the lower row, the radius is the distance between the center and the short edge of the distal link, denoted as r_2. As we can see from their corresponding fingertip space, r_2 is indeed a more restricted condition that requires a larger area on the object to fit the fingertip, and in the meanwhile it results in a much sparser fingertip space.

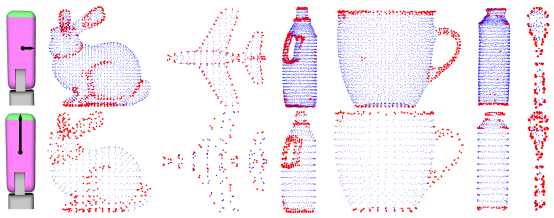

Fig. 5. Fingertip Space with different fingertip unit sizes.

Fig. 6 records the statistics of our system evaluation. Recall that the initialization of the system is random in this work and there is also randomness in the stochastic hill

climbing procedure, the results generated by system can be different between each single run of the system. Therefore, we ran it 100 times on each of the six test objects using fingertip unit sizes of both r_1 and r_2 and investigate the averaged performance. We can see that the result for the plane model is much worse than others. This is due to the fact that the plane has many parts that are highly concave and that the Barrett hand is coupled with only 4 DoFs, the grasp realization is therefore much more difficult. For the same reason, it is much easier to expect collisions between the robot hand and the plane surface and more search iterations are therefore required. However, it is intuitive that if a more dexterous hand is employed, it is easier for us to deal with more complex object shapes. The averaged time per iteration is related to the size of extracted fingertip space Φ, which is shown in the parenthesis in the first column: the larger fingertip space an object has, the more time it takes to search for a precision grasp.

It is worth noting that the performance of the system is generally better when fingertip radius was set to r_2. Because given the same fingertip embodiment, a larger fingertip unit makes it safer to stabilize a contact. Fig. 7 displays some example stable grasps synthesized by the system.

Object(radius: #Units)	Stable(%)	Rounds	Time/Round
Bunny(r_1: 3276)	98	1.73	13.08s
Bunny(r_2: 293)	**100**	1.64	6.03s
Plane(r_1: 579)	75	3.16	8.35s
Plane(r_2: 96)	**89**	3.98	5.97s
Waschmittel(r_1: 4236)	95	1.51	17.25s
Waschmittel(r_2: 644)	**95**	1.22	9.13s
Cup(r_1: 3068)	98	1.42	13.00s
Cup(r_2: 730)	**97**	1.51	8.19s
Spoon(r_1: 91)	88	1.83	8.22s
Spoon(r_2: 49)	**91**	1.91	3.31s
Milk Box(r_1: 3936)	100	2.69	13.68s
Milk Box(r_2: 842)	**100**	3.73	9.98s

Fig. 6. Statistics of algorithm evaluation. *Stable(%)*: The percentage of stable grasps after the grasps were executed. *Rounds*: The averaged rounds of Alg. 1 to successfully output a good grasp, note that Alg. 1 is restarted if the final check is not satisfied. *Time/Round*: The averaged time in seconds that one round of the algorithm takes.

C. Positioning Error Tolerance of Fingertip Space

In Fig. 8, grasps are shown with their realized contacts (green) and synthesized contacts (red). The realized grasps are usually a bit different from what was synthesized, both in contact positions and normals. This is due to the fact that the reachability measure employed in Sec. IV is an approximation of the real reachability manifold and that the Barrett hand is not dexterous enough to always sufficiently deal with non-zero reachability residual. In this section, we examine whether the synthesized grasps will remain stable if the final executions of them have positioning errors with respect to synthesized contacts.

The experiments have been conducted on *Stanford Bunny, Plane* and *Waschmittel* models by assuming that the positioning errors are within one and two fingertip unit sizes, given the fact that the positioning errors recorded in our experiments were smaller than two fingertip unit size. Similarly to the concept of Independent Contact Regions [40], we consider a grasp as tolerant to positioning errors if all contacts can be freely positioned within a certain range without losing stability. In this experiment, 100 grasps have been synthesized on all three objects, and contacts within the error limit were sampled and the percentages of sampled nearby stable grasps were recorded for each grasp. Test results are shown in Fig. 9 as the percentages of the nearby stable grasps with standard deviation on the bar plot.

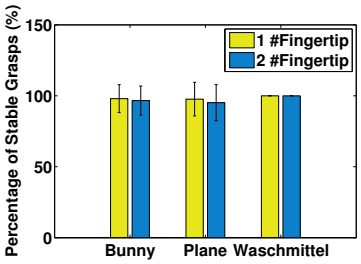

Fig. 9. Positioning error tolerance test results: percentages of stable nearby grasps given positioning errors within one and two fingertip unit sizes.

As can be seen from the results, neighbors of synthesized stable grasps remain stable with high probabilities. This is an evidence for the fact that synthesized grasps are tolerant to small positioning errors and that our reachability measure retains the relevant information. This can be explained by the fingertip space extraction: since fingertip units are positions where the object surface is smooth, small positioning errors will not heavily influence contact positions and normals, and the grasp stability is therefore also not heavily influenced, which can be referred back to our motivation in Sec. I.

D. Precision Grasp Synthesis with Noisy Data

In this section, we examine the performance of our algorithm considering noisy sensory data. As shown in Fig. 10, we scanned the *Stanford Bunny, Plane* and *Waschmittel* models using a virtual 3D sensor while adding Gaussian noise in the viewing direction. For the extraction of fingertip units, the fingertip size was set to r_2. As we can see, the fingertip space becomes different comparing to noise-free objects. However, it is worth to note that, although the objects are noisy, the extracted fingertip units are still retaining the property of flatness and smoothness.

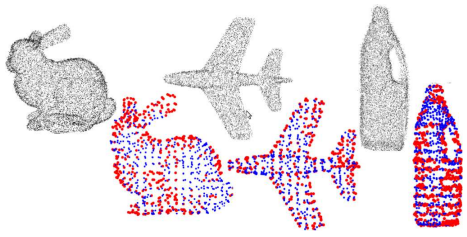

Fig. 10. Noisy objects used in experiments and their corresponding fingertip space.

Fig. 11 records the statistics of 100 runs of our approach. Grasps were synthesized using noisy data and the final

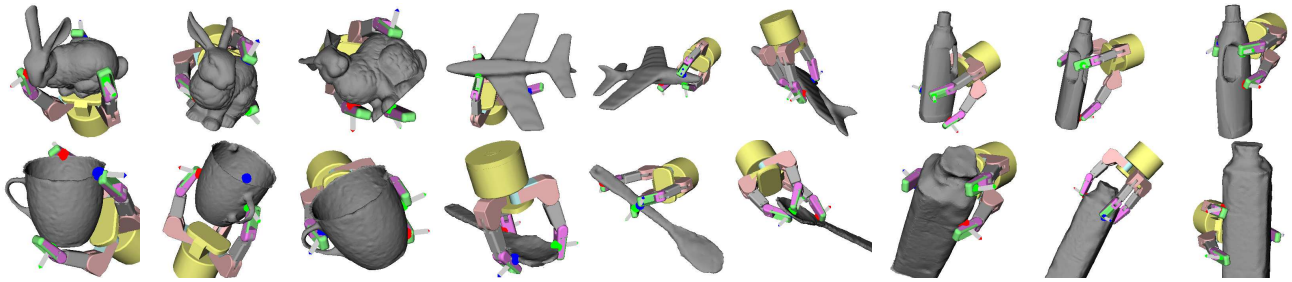

Fig. 7. Example precision grasps synthesized by the algorithm.

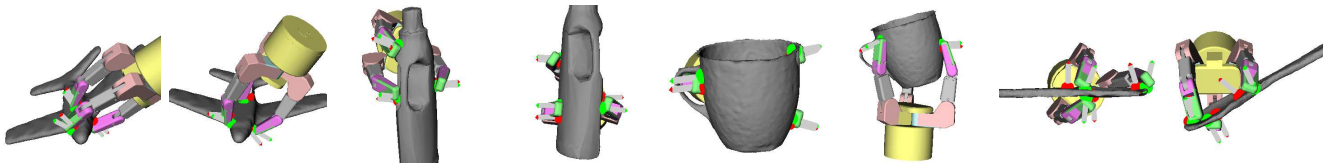

Fig. 8. Positioning errors in grasp realization.

grasp qualities are computed after the synthesized grasps have been executed on the perfect objects. The result shows that the percentage of stable grasps have been decreased in comparison to the noise-free experiments, however, the system can still synthesize stable precision grasps.

Object(#Units)	Stable(%)	Rounds	Time/Round
Bunny(122)	92	2.12	7.63s
Plane(111)	83	4.16	7.24s
Waschmittel(582)	90	2.05	9.75s

Fig. 11. Statistics of algorithm evaluation with noise. *Stable(%)*: The percentage of stable grasps after the grasps were executed. *Rounds*: The averaged rounds of Alg. 1 to successfully output a good grasp, note that Alg. 1 is restarted if the final check is not satisfied. *Time/Round*: The averaged time in seconds that one round of the algorithm takes.

E. Grasp Synthesis with Partially Observed Data

It is difficult to observe complete point clouds of target objects in real applications. In this section, we simulate partial views of objects by setting locations of a virtual camera, and then we show example stable grasps synthesized by the system, see Fig. 12.

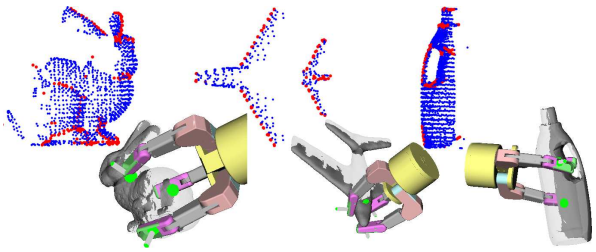

Fig. 12. Upper: Fingertip space of the partially observed objects. Lower: Grasps synthesized on partially observed objects. Unobserved parts on the object are shown in transparency.

As shown in the examples, grasps can still be successfully synthesized and the contacts are only synthesized for visible positions. This is because the fingertip space extraction and the hierarchy construction operate directly on the observed point cloud and does not require the object to be completely observed. This implies another advantage of the proposed object representation that the system is able to synthesize precision grasps as long as the observed parts of the object are graspable. In the real applications, if no successful grasps can be synthesized by the system from a single view of the object, the robot can move to a different position to find graspable parts.

F. An Example of Grasp Synthesis and Realization

In this section, we present an example of grasp optimization and execution.

Multilevel Grasp Optimization

As the refinement procedure in Alg. 1 aims at improving the objective function $\theta(C_g)$, it searches for larger $Q(C_g)$ and smaller $R(C_g)$ values.

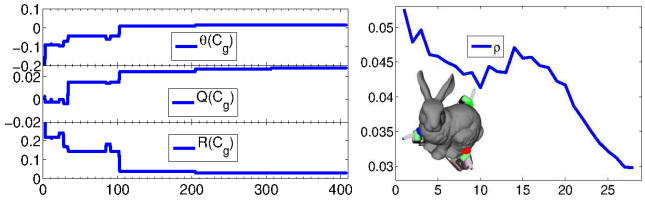

Fig. 13. *Left:* Records of multilevel grasp optimization. *Right:* Records of contact positions optimization, $\rho = \left\| C_g^+ - C_g^* \right\|$. The horizontal axes are number of iterations in both figures.

Fig. 13 displays one example of $\theta(C_g)$, $Q(C_g)$ and $R(C_g)$ curves of Alg. 1 applied on the Bunny model. We can see that the $\theta(C_g)$ value is generally increasing with a few decreases due to the randomness in the Alg. 1, and that the $Q(C_g)$ value is also generally increasing. However, the $R(C_g)$ value is decreasing but sometimes increasing, this is because the search procedure was attempting many different joint configurations to fit a grasp while balancing between other objectives. Next, we apply the contacts optimization to realized the grasp with synthesized contacts.

Contact Positions Optimization

As shown in Fig. 13, ρ value is generally decreasing during the gradient descent but is occasionally overshooting. The overshots are due to the joint space of robot hand and the object surface is very complicated and has many local optima. After the contact positions optimization is done, the final stable precision grasp was achieved as shown on the right. It is worth to mention that as the fingertips' positions after affine transform was already very close to the desired position, the gradient descent did not need many steps to converge.

VI. Conclusion

In this paper, we have proposed a concept of *Fingertip Space*, which is an integrated representation of both object local geometry and fingertip geometry, and shown its use in precision grasp synthesis. By building a hierarchical representation of the fingertip space, we have enabled multilevel refinement for precision grasp synthesis. Our experimental evaluation with a Barrett hand has shown that the fingertip space and its hierarchy is a viable and efficient representation for precision grasp synthesis, and that the multilevel refinement facilitates the search procedure. We have also evaluated the positioning errors tolerance of our system, as well as demonstrated examples of our system working with noisy and incomplete data. In the future, we are planning to implement our system on a real robot and additionally make the modular system more compact and flexible for different robot embodiments and search algorithms to be plugged in.

References

[1] Z. Liu, L. B. Gueta, and J. Ota, "A strategy for fast grasping of unknown objects using partial shape information from range sensors," *Advanced Robotics*, vol. 27, no. 8, pp. 581–595, 2013.

[2] M. Popovic, D. Kraft, L. Bodenhagen, E. Baeski, N. Pugeault, D. Kragic, T. Asfour, and N. Krüger, "A strategy for grasping unknown objects based on co-planarity and colour information," *Robotics and Autonomous Systems*, vol. 58, no. 5, pp. 551 – 565, 2010.

[3] K. Huebner, "BADGrA toolbox for box-based approximation, decomposition and GRasping," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 367 – 376, 2012.

[4] Y. Bekiroglu, D. Song, L. Wang, and D. Kragic, "A probabilistic framework for task-oriented grasp stability assessment," in *IEEE ICRA*, 2013.

[5] J.-P. Saut, A. Sahbani, S. El-Khoury, and V. Perdereau, "Dexterous manipulation planning using probabilistic roadmaps in continuous grasp subspaces," in *IEEE/RSJ IROS*, 2007.

[6] L. Han and J. Trinkle, "Dexterous manipulation by rolling and finger gaiting," in *IEEE ICRA*, 1998.

[7] K. Hertkorn, M. Roa, and B. Ch, "Planning in-hand object manipulation with multifingered hands considering task constraints," in *IEEE ICRA*, 2013.

[8] T. Hasegawa, K. Murakami, and T. Matsuoka, "Grasp planning for precision manipulation by multifingered robotic hand," in *IEEE SMC*, vol. 6, 1999, pp. 762–767.

[9] M. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE RAS*, vol. 5, no. 3, pp. 269–279, 1989.

[10] C. Borst, M. Fischer, and G. Hirzinger, "Calculating hand configurations for precision and pinch grasps," in *IEEE/RSJ IROS*, 2002.

[11] ——, "Grasping the dice by dicing the grasp," in *IEEE/RSJ IROS*, 2003.

[12] ——, "Efficient and precise grasp planning for real world objects," in *Multi-point Interaction with Real and Virtual Objects*, 2005, vol. 18, pp. 91–111.

[13] C. Rosales, J. Porta, R. Suarez, and L. Ros, "Finding all valid hand configurations for a given precision grasp," in *IEEE ICRA*, 2008.

[14] J.-P. Saut and D. Sidobre, "Efficient models for grasp planning with a multi-fingered hand," *Robotics and Autonomous Systems*, vol. 60, no. 3, pp. 347 – 357, 2012.

[15] J. Aleotti and S. Caselli, "A 3d shape segmentation approach for robot grasping by parts," *IEEE RAS*, vol. 60(3), pp. 358–366, 2012.

[16] M. Przybylski, T. Asfour, and R. Dillmann, "Planning grasps for robotic hands using a novel object representation based on the medial axis transform," in *IEEE/RSJ IROS*, 2011.

[17] F. T. Pokorny, J. A. Stork, and D. Kragic, "Grasping objects with holes: A topological approach," in *IEEE ICRA*, 2013.

[18] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *IEEE ICRA*, 2003.

[19] G. Biegelbauer and M. Vincze, "Efficient 3d object detection by fitting superquadrics to range image data for robot's object manipulation," in *IEEE ICRA*, 2007.

[20] T. T. Cocias, S. M. Grigorescu, and F. Moldoveanu, "Multiple-superquadrics based object surface estimation for grasping in service robotics," in *Optimization of Electrical and Electronic Equipment (OPTIM)*, 2012, pp. 1471–1477.

[21] D. Berenson, R. Diankov, K. Nishiwaki, S. Kagami, and J. Kuffner, "Grasp planning in complex scenes," in *IEEE-RAS Humanoids*, 2007.

[22] F. T. Pokorny and D. Kragic, "Classical grasp quality evaluation: New theory and algorithms," in *IEEE/RSJ IROS*, 2013.

[23] N. Niparnan and A. Sudsang, "Fast computation of 4-fingered force-closure grasps from surface points," in *IEEE/RSJ IROS*, 2004.

[24] R. Diankov, "Automated construction of robotic manipulation programs," Ph.D. dissertation, Carnegie Mellon University, Robotics Institute, 2010.

[25] Z. XUE and R. DILLMANN, "Efficient grasp planning with reachability analysis," *International Journal of Humanoid Robotics*, vol. 08, no. 04, pp. 761–775, 2011.

[26] F. Zacharias, C. Borst, and G. Hirzinger, "Online generation of reachable grasps for dexterous manipulation using a representation of the reachable workspace," in *IEEE ICRA*, 2009.

[27] M. T. Ciocarlie and P. K. Allen, "Hand posture subspaces for dexterous robotic grasping," *The International Journal of Robotics Research*, vol. 28, no. 7, pp. 851–867, 2009.

[28] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE ICRA*, 1992.

[29] A. Bicchi and V. Kumar, "Robotic grasping and contact: A review," in *IEEE ICRA*, 2000.

[30] M. Li, Y. Bekiroglu, D. Kragic, and A. Billard, "Learning of grasp adaptation through experience and tactile sensing," in *IEEE/RSJ IROS*, 2014.

[31] K. Hang, J. A. Stork, F. T. Pokorny, and D. Kragic, "Combinatorial optimization for hierarchical contact-level grasping," in *IEEE ICRA*, 2014.

[32] C. Walshaw, "Multilevel refinement for combinatorial optimisation problems," *Annals of Operations Research*, vol. 131, no. 1-4, pp. 325–372, 2004.

[33] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, ser. Prentice Hall series in artificial intelligence. Prentice Hall, 2010.

[34] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[35] P. Révész, *Random Walk in Random and Non-random Environments*. World Scientific, 2005.

[36] B. P. Gerkey, S. Thrun, and G. Gordon, "Parallel stochastic hill-climbing with small teams, in l.e.parker et al., eds," in *Multi-Robot Systems: From Swarms to Intelligent Automata*, 2005.

[37] G. Turk and M. Levoy, "Zippered polygon meshes from range images," in *Proceedings of the 21st Annual Conference on Computer Graphics and Interactive Techniques*, 1994.

[38] X. Chen, A. Golovinskiy, and T. Funkhouser, "A benchmark for 3D mesh segmentation," *ACM SIGGRAPH*, vol. 28, no. 3, 2009.

[39] B. Len, S. Ulbrich, R. Diankov, G. Puche, M. Przybylski, A. Morales, T. Asfour, S. Moisio, J. Bohg, J. Kuffner, and R. Dillmann, "Opengrasp: A toolkit for robot grasping simulation," in *Simulation, Modeling, and Programming for Autonomous Robots*, ser. Lecture Notes in Computer Science, 2010, vol. 6472, pp. 109–120.

[40] M. Roa and R. Suarez, "Computation of independent contact regions for grasping 3-d objects," *Robotics, IEEE Transactions on*, vol. 25, no. 4, pp. 839–850, 2009.