2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Three-Finger Precision Grasp on Incomplete 3D Point Clouds

Ilaria Gori[1], Ugo Pattacini[1], Vadim Tikhanoff[1] and Giorgio Metta[1]

***Abstract*— We present a novel method for three-finger precision grasp and its implementation in a complete grasping toolchain. We start from binocular vision to recover the partial 3D structure of unknown objects. We then process the incomplete 3D point clouds searching for good triplets according to a function that accounts for both the feasibility and the stability of the solution. In particular, while stability is determined using the classical force-closure approach, feasibility is evaluated according to a new measure that includes information about the possible configuration shapes of the hand as well as the hand's inverse kinematics. We finally extensively assess the proposed method using the stereo vision and the kinematics of the iCub robot.**

I. INTRODUCTION

Object manipulation is certainly one of the keys enabling technologies in a variety of different robotic domains. Manipulation is also a fertile research topic due to its intrinsic complexity. The dimensionality of the hand and arm's configuration space and the difficulty of retrieving accurate visual priors are perhaps the two major factors beneath its complexity. In Napier's taxonomy [1] grasp actions are classified in power grasps and precision grasps. In power grasp [2], [3] and [4] the object and the hand share large contact areas, hindering further movements of the fingers. On the other hand, in precision grasp, the object is contacted with the tips of the fingers. Power grasps are particularly suited when the object does not need to be handled precisely, whereas precision grasps represent the best choice when specific tasks have to be fulfilled as e.g. in tool use. This work focuses on precision grasp.

Generating a precision grasp requires finding a set of contact points on the object that are *stable* and *feasible*, given the hand's size and kinematics, its material and contact properties as well as its overall joints stiffness. A set of contact points is said to be *force-closure* if it can resist external forces and moments without dropping the object. From now on, when we mention stable grasps, we will always intend force-closure grasps. Further, a grasp is feasible when its configuration, defined as the finger joint angles along with the position and the orientation of the end-effector, allows appropriate contacts at the desired locations on the object. As finding complete solutions turns to be difficult, the literature abounds of partial solutions focusing on isolated aspects of the problem. For example, the earlier proposals only formalize the question of stability, assuming that the contact points were given [5] [6] [7]. More recently, the work of Erkan et al. [8] and Rao and colleagues [9] attack the problem by trying to learn good grasps on the basis of empirical experience using machine learning techniques. However they neither account for stability nor feasibility of the grasp. Yet other methods start from computing a feasible hand configuration for a set of given stable points [10] [11]. There are also "hybrid" approaches [12] that try to provide contact points that are stable, feasible or both. Our work falls squarely into the class of hybrid approaches, tackling the full grasping problem by computing stable contact points, and determining a hand configuration that guarantees feasibility.

To the best of our knowledge, all the precision grasp hybrid approaches use either synthetic objects or complete point clouds of known objects to compute reliable grasps [13], [14], [15]. We can certainly store and manage information on 3D models of a very large number of objects; nevertheless, because of the variety of objects' sizes and shapes, a robotic grasping strategy should be also robust to unknown objects. This implicitly extends the robot's skills to situations where recognition fails to classify the object. Being able to deal with unknown objects contributes to the overall quality of the robot's grasping skills. This line of reasoning leads us to the consideration that the robot can only use incomplete 3D shape information as acquired from a single viewpoint unless, in the presence of an unknown object, it starts a complicated exploration and data collection procedure. Assuming that this is not desirable, we can then reason on the fact that the robot cannot use points in the occluded parts of the object. Our method is designed to work with incomplete 3D point clouds resulting from a single view of the object, albeit nothing prevents us from exploiting the additional information where better models are available.

Another popular procedure that often characterizes hybrid approaches is the approximation of the 3D data with specific elementary shapes, as for example, boxes [16], shape primitives [17] or superquadrics [13], [18]. This approach allows handling regular objects quite precisely although, in many cases, it may introduce an additional source of uncertainty on the contact point determination. On highly irregular objects, the error of the approximation procedure coupled with the intrinsic noise of the sensors (vision) and the mechanical error of the hand becomes the recipe for disaster. Our first contribution is a method based on the raw collection of the 3D points, without employing any object modeling.

Most of the hybrid approaches find contact points on the object surface, while assuring their stability using standard force-closure criteria [15], [19], [13]. Nevertheless, they do

*This work was supported by the European FP7 ICT project No. 270490 (EFAA), project No. 270273 (Xperience) and project No. 288382 (Poeticon ++)

[1]iCub Facility, Istituto Italiano di Tecnologia, Via Morego 30, Genova {ilaria.gori, ugo.pattacini, vadim.tikhanoff, giorgio.metta}@iit.it

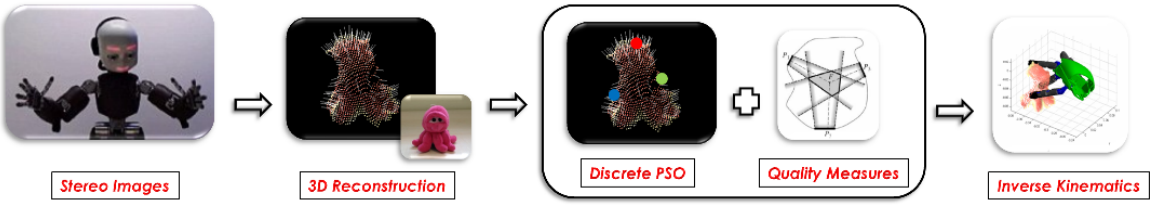

Fig. 1. The figure illustrates the entire pipeline of our algorithm. We reconstruct the object in 3D obtaining a point cloud along with its surface normals. The point cloud is explored in order to find candidate triplets that satisfy specific criteria. We then compute, for each candidate triplet, the inverse kinematics of the hand. Finally, we select the best grasp on the basis of the manipulability measure computed on the whole robot's arm.

not take into account the physical characteristics of the robotic hand; importantly, this limitation may result in unfeasible configurations. In this sense our second contribution is a measure of the robot's hand physical properties in evaluating the stability of the contact points. Privileging fingers configurations that lead more often to feasible grasps can upfront increase the probability of success. Subsequently, the inverse kinematics of the hand helps to prune unreachable configurations altogether.

There are a few hybrid methods that search for stable and feasible contact points simultaneously [18], [14]. However they require complete 3D models of the objects. They limit their analysis to the hand, assuming that it can move freely around the object: this is not sufficient to make sure that the robot can reach the desired position, as the hand is connected to robot's body, which lies in a specific position with respect to the object. Our last contribution is a preliminary attempt to account for the robot's position when selecting the best grasp configuration, in order to use the proposed algorithm on real robots and specifically on humanoids.

We identified a roadmap that progressively evolve from grasping in purely simulated environments to complete real world case-studies with a taxonomy determined by the accuracy of the information that is available to the robot (i.e. complete 3D point cloud vs. single view): (A) Complete 3D point clouds with simulated grasps; (B) partial 3D point clouds with simulated grasps; (C) partial 3D point clouds with the grasp actually executed by the real robot. As far as we know, all the precision grasp hybrid approaches presented in the literature were developed either on synthetic objects or on complete point clouds, exploiting the advantage afforded by the perfect knowledge of the object. In this paper we focus on scenario B, developing a method that is closer to be utilized on a real robot in realistic everyday conditions.

II. METHOD

El-Khoury [18] supports the view that finding a *stable* and *feasible* set of contact points on an object is the primary goal when dealing with precision grasp. He further claims that the simultaneous fulfillment of these two conditions brings robustness to the grasp retrieval procedure. This implies that both stability and feasibility have to be calculated on the same type of variables, either discrete of continuous. In El-Khoury's work, the optimization is continuous since the object is approximated with a superquadric and the hand configuration is also continuous (i.e. the kinematics). On the contrary, since we do not make use of any surface fitting, our object models belong to the discrete domain and cannot be explored jointly with the space of the hand configurations. Grasping has thus to be made of two distinct routines that find a sufficient number of candidate points (triplets) and solve the inverse kinematic problems for the hand (see Fig. 1).

In the first phase, we seek for a number of candidate triplets – sets of three contact points with corresponding normals – under the condition that they allow the generation of a good grasp (according to the stability criterion). Since running the inverse kinematics on every single triplet is computationally expensive, we need to guarantee that the candidate triplets are selective. As a first approximation we privilege triplets that are certainly within the grasping envelope of the iCub's hand, as it is pointless to ask the iCub to grasp a large object (given its child size). Similarly the iCub cannot position its first three fingers at an equal angle with respect to the center of the object.

Once a sufficient number of candidate sets of contact points has been found, we select the best one on the basis of the feasibility of the grasp configuration. We compute the inverse kinematics of the hand for each candidate triplet and evaluate the physical feasibility of the solution. Finally, we consider the position of the robot with respect to the object, and we select the best grasp among those that do not imply critical or singular configurations of robot's arm and torso.

The remainder of the paper is structured as follows: Sec. III illustrates the selection of the candidate triplets, Sec. IV describes the determination of the best grasp using inverse kinematics, Sec. V presents the experimental results and Sec. VI draws the conclusion and illustrates future improvements.

III. TRIPLETS EXTRACTION

We start from an object lying in front of our robot, iCub [20]. The first step of our algorithm consists in exploring the visible portion of the object and selecting a number of best candidate triplets. We first reconstruct the object in 3D from a single viewpoint, obtaining an incomplete 3D point cloud. We then estimate surface normals on each point, and we sample the cloud assuming that close points have similar normals. On the sampled cloud, we apply a variant of the Discrete Particle Swarm Optimization algorithm [21], finding

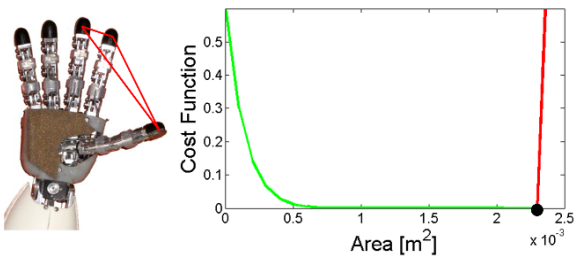

Fig. 2. Cost function accounting for the robot's hand dimension. From the picture it is possible to infer that bigger areas are privileged. The green line represents the cost function for acceptable areas. Nevertheless, if the area is bigger than the maximum area that can be covered by the hand – which is represented by the big black dot – the measurement grows up dramatically (red line), indicating a bad triplet.

a number of triplet points that satisfy several conditions. The first condition is related to the triplet stability, whereas the second and the third conditions are associated to the hand size and shape, respectively.

A. 3D Reconstruction and Sampling

We rely on stereo vision algorithms in order to retrieve 3D information. The *semi-global matching* algorithm [22] is applied to the images recorded by our robot to estimate the depth map, and project each visible pixel of the object in the 3D space. We then estimate the normal to the surface at each point of the cloud by running a least-squares fitting over the point neighborhood to determine the corresponding tangent plane [23]. We define the neighborhood P^k of a point $\boldsymbol{p_i}$ as the set of k points that lie within a circular area having radius equal to the radius of the robotic fingertip. Given a point $\boldsymbol{p_i}$ and its neighborhood P^k, the plane tangent to the surface can be defined as a couple $(\boldsymbol{x_i}, \bar{\boldsymbol{n}}_{\boldsymbol{i}})$, where $\boldsymbol{x_i}$ is a point of the plane and $\bar{\boldsymbol{n}}_{\boldsymbol{i}}$ is the normal to the plane. The distance d_i between a point $\boldsymbol{p_i} \in P^k$ and the fitting plane can be expressed as $d_i = (\boldsymbol{p_i} - \boldsymbol{x_i})\bar{\boldsymbol{n}}_{\boldsymbol{i}}$; therefore, the plane parameters are computed minimizing the distance d_i for each point. If we assume that $\boldsymbol{x_i}$ is the centroid of the neighborhood (i.e. $\boldsymbol{x_i} = \frac{1}{k}\sum_{j=1}^{k} \boldsymbol{p_j}$), then the solution for $\bar{\boldsymbol{n}}_{\boldsymbol{i}}$ can be calculated by analyzing the eigenvectors and eigenvalues of the following covariance matrix:

$$C = \frac{1}{k}\sum_{j=1}^{k}(\boldsymbol{x_i} - \boldsymbol{p_j})(\boldsymbol{x_i} - \boldsymbol{p_j})^T. \qquad (1)$$

We thus assemble a 3D voxel grid over the point cloud, which results in a set of small 3D boxes in the space. We downsample the point cloud selecting the center of each 3D boxes. Under the reasonable assumption that very close points have very similar surface normals, we obtain a good representation of the initial point cloud.

B. Triplet Desired Properties

We now aim to retrieve a sufficient number of good triplets. A triplet is defined as "good" based on three properties. A first, necessary property is the triplet stability in terms of force-closure. The other two properties are related, respectively, to the dimension and the shape of the robotic hand.

1) Force-closure Grasp: We will exploit the stability analysis to define if a triplet is acceptable or not acceptable. In case the set of contact points is not stable, it cannot be considered as a candidate triplet.

We adopt the hard finger contact model [5], which is the most commonly used. This model is characterized by a significant friction acting on small contact patches, and by no transmission of the angular velocity or moment components to the object. Assuming that only static friction holds, we can build a friction cone $\boldsymbol{F_i}$ with the vertex in contact point $\boldsymbol{p_i}$ and aligned along the normal $\boldsymbol{n_i}$ on the point. The friction cone at point $\boldsymbol{p_i}$ represents the set of forces that can be exerted on the point without producing a slippage of the finger along the surface of the object. The cone $\boldsymbol{F_i}$ is determined by the friction coefficient μ_i on the point $\boldsymbol{p_i}$, as follows:

$$\boldsymbol{F_i} = \{(f_{in}, f_{it}, f_{io}) | \sqrt{f_{it}^2 + f_{io}^2} < \mu_i f_{in}\}. \qquad (2)$$

To ensure that the finger will not slip along the surface of the object, the force applied by the fingertip on the point $\boldsymbol{p_i}$ must lie within the friction cone $\boldsymbol{F_i}$. In order to evaluate the stability of a grasp, we adopt the force closure analysis. Intuitively speaking, a grasp is considered force-closure if and only if we can exert arbitrary force and moment on the grasped object by pressing the fingertips against the object [24].

Most of the force-closure evaluation techniques rely on the examination of the contact wrenches [25] [13], which consists of the computation of a 6D convex hull and implies the approximation of the friction cone to a pyramid. On the one hand, approximating the cone with a pyramid having a large number of sides entails an increasing computational complexity. On the other hand, using a small number of sides makes the friction cone approximation coarser and the force-closure analysis less reliable. Differently, we based our evaluation on geometrical considerations. Specifically, we use the necessary and sufficient condition proposed by [7]: a three finger 3D grasp achieves force-closure if and only if

- the three points are not collinear and meet on a plane S;
- each friction cone $\boldsymbol{F_i}$ intersects the contact plane S on a plane, generating two unit vectors $\boldsymbol{n_{i1}}$ and $\boldsymbol{n_{i2}}$ that bound the projection of the cone on the plane;
- the contact unit vectors construct a 2D force-closure grasp in S.

This condition is fast to compute and does not require any approximation of the friction cones.

2) Hand Dimension: In order to choose the best candidate among the set of available force-closure triplets, we need a specific heuristic to extract the most suitable grasp in relation with the robotic hand. The use of such a heuristic will also increase the probability of obtaining a feasible grasp in the subsequent stage. A well-known quality measure to evaluate how good a triplet is, is the area covered by the

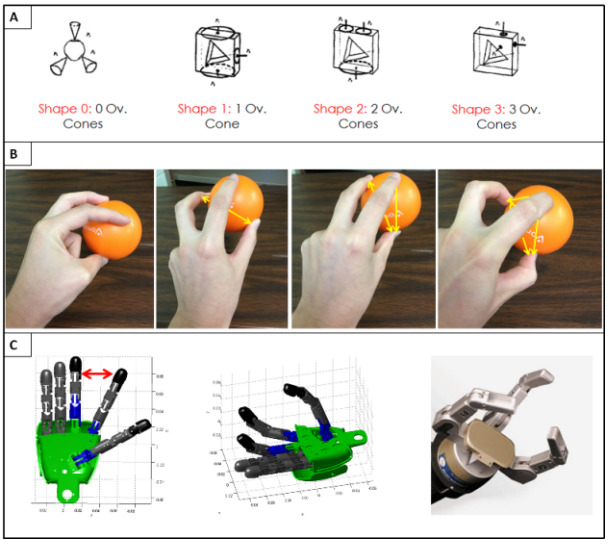

Fig. 3. A: the four classes of force-closure grasps are depicted. Each of them is characterized by a different number of pairwise counter-overlapping cones. B: the four classes are illustrated using a human hand grasping a ball. Yellow arrows indicate which pairs of friction cones are counter-overlapping. C: left figure, a 3D model of the iCub's hand is depicted. The red arrow represents the biggest distance between the index and the middle that can be realized. In the center, an image of the iCub's hand while closing is showed. This picture is meant to illustrate pictorially that iCub's hand is not suitable for $shape_3$ grasps. Differently, the Barrett hand, showed on the right, can move the fingers independently and therefore perform grasps belonging to $shape_3$ easily.

grasp polygon [26]. The bigger the area is, the better the grasp will be. We modify this quality measure in order to account for the robotic hand size (see Fig. 2). In particular, we privilege triplets that cover large areas, and at the same time do not exceed the physical limit imposed by the size of the robot's hand. Given a triplet T, the area of the grasp polygon a, and the maximum area a_{max} that the robotic hand's fingertips can cover, a good triplet must minimize the following discontinuous function:

$$d(T) = \begin{cases} k_1(a_{max} - a)^2, & \text{if } a \leq a_{max} \\ k_2 a, & \text{otherwise} \end{cases} \tag{3}$$

where $k_1 \geq 0$ and $k_2 \geq 0$ are parameters empirically found. Specifically, if the area covered by the triplet is smaller than a_{max}, the function has a paraboloid shape. Contrarily, when the area covered by the triplet is larger, the function increases linearly (see Fig. 2).

3) Hand Shape: Another crucial information that, as far as we know, has never been taken into account, is the specific shape of the hand. Indeed, different robotic hands will have different grasping capabilities. It has been demonstrated in [6] that there exists four type of force-closure grasps with three hard-finger contacts, and they are dependent on the number of friction cones that pairwise counter-overlap. Specifically, two friction cones $\boldsymbol{F_i}$ and $\boldsymbol{F_j}$ are said to counter-overlap if the angle between their principal axes $\boldsymbol{n_i}$ and $\boldsymbol{n_j}$ is smaller than the angle at the basis of the friction cones, which is equal to $2\arctan(\mu)$, where μ is the friction coefficient. The four classes are depicted in Fig. 3, row A; they represent, respectively, a force-closure grasp with 0 counter-overlapping cones ($shape_0$), a force-closure grasp with 1 counter-overlapping cone ($shape_1$), a third force-closure grasp with 2 counter-overlapping cones ($shape_2$) and a last force-closure grasp with 3 counter-overlapping cones ($shape_3$). Fig. 3 also illustrates the four grasps performed by a human hand, in row B. In Fig. 3 row C the iCub's hand and a Barrett hand are depicted. From the images it is possible to infer that a humanoid robotic hand will strive to realize a $shape_3$ grasp, because of the limited displacement between the index and the middle fingers, which cannot face each other. This shape can be achieved only when the friction coefficient takes very high values. The iCub's most comfortable force-closure grasp classes will probably be $shape_1$ and $shape_2$. Therefore, when a good triplet is selected, it should also have a shape that adapts to the particular robotic hand in use. A very good example is represented by a spherical object. It will certainly present stable triplets belonging to $shape_3$, but for a humanoid robotic hand a triplet belonging to $shape_2$ or $shape_1$ will be preferable in terms of feasibility.

Inspired by these considerations, we develop a novel measure incorporating the notion of desired shape. Given a triplet T, a number m of desired counter-overlapping cones, and the actual number $c(T)$ of counter-overlapping cones in the triplet T, we compute the quantity $f = m - c(T)$. If $f \neq 0$ we build two vectors: a vector $\boldsymbol{u}$ containing the angles between the pairs of friction cones that counter-overlap, sorted from the smallest to the biggest value, and a second vector $\boldsymbol{v}$ that contains the sorted angles between the pairs of friction cones that do not counter-overlap. We propose the following measure:

$$s(T) = \begin{cases} 0, & \text{if } f = 0 \\ \sum_{i=1}^{f} v_i, & \text{if } f > 0 \\ \sum_{i=1}^{f} u_i, & \text{if } f < 0. \end{cases} \tag{4}$$

The above-mentioned formulation represents how much the current triplet is far from belonging to the class of the desired shape. As we will minimize these quantities, the smallest the measure is, the better the triplet is evaluated.

C. Discrete Particle Swarm Optimization

Given a set of points P, we aim at finding a triplet $T = (\{\boldsymbol{p_i}, \boldsymbol{n_i}\}, \{\boldsymbol{p_j}, \boldsymbol{n_j}\}, \{\boldsymbol{p_k}, \boldsymbol{n_k}\}), \boldsymbol{p_i}, \boldsymbol{p_j}, \boldsymbol{p_k} \in P$ that satisfies the properties illustrated in Sec. III-B. In particular, we want to solve the following problem:

$$\begin{aligned} \min_{T} \quad & d(T) + ws(T) \\ \text{s. t.} \quad & T \text{ is force-closure,} \end{aligned} \tag{5}$$

which summarizes the characteristics that a candidate triplet should present. The stability is formulated as a constraint, as it is necessary to consider a triplet acceptable. At the same time, we seek to select candidate triplets of a specific shape; this is defined by $s(T)$, which corresponds to Eq. (4). Then, $d(T)$ is Eq. (3), and guarantees that the area covered by the

triplet is large, but does not exceed the maximum area that can be covered by the hand. Finally, w is a variable chosen empirically, which regulates the contribution of $d(T)$ and $s(T)$. Even though the number of possible triplets is finite, and thus it would be possible to analyze them all to find the optimal one, an exhaustive search cannot be performed if the number of point is too large, as it is likely to be in real scenarios. Furthermore, Borst [27] demonstrated that there is no need to find the globally optimal grasp, because most of the times an average grasp (in the force-analysis sense) is acceptable. Therefore we resort to a local optimization algorithm.

We have a discrete number of points P, thus a discrete, derivative-free optimization algorithm is requested. On the other hand, our downsampled point cloud is an approximation of a surface, and it holds some nice properties that we would like to exploit. For instance, the fact that close points are likely to have similar normals. This assumption would help the optimization algorithm to find a good solution fast. Among the discrete optimization algorithms, a technique that satisfies all our needs is the Discrete Particle Swarm Optimization (DPSO), in the variant presented in [21]. Particle swarm optimization (PSO), originally presented in [28], is a population-based optimization method. It initializes a set of particles that explore the space domain on the basis of historical information learnt from the swarm population. In our setting a particle $\boldsymbol{p}$ is represented by three points $\boldsymbol{p_1}$, $\boldsymbol{p_2}$ and $\boldsymbol{p_3}$, therefore $\boldsymbol{p} \in \mathbb{R}^9$. At each point the normal computed as in Sec. III-A is associated, therefore a triplet T can be represented as the couple $\{\boldsymbol{p}, \boldsymbol{n}\}$, where $\boldsymbol{n}$ is the set $\{\boldsymbol{n_1}, \boldsymbol{n_2}, \boldsymbol{n_3}\}$ of normals on the points in $\boldsymbol{p}$. Each particle has its position and its velocity; it also keeps track of the best global position $\boldsymbol{g^*}$ ever visited from the population, as well as the best position $\boldsymbol{l^*}$ it has visited in the past. For each particle a fitness function on its associated triplet is computed. In our case, the fitness function is in the form $d(T)+ws(T)+fc(T)$ (see Eq. (5)), where $fc(T)$ returns 1 if the triplet is force-closure, a very large number otherwise. At each iteration, all the particles update their positions modifying their velocities on the basis of their past velocity, $\boldsymbol{g^*}$ and $\boldsymbol{l^*}$. As in [21], we use the velocity formula that is used in the standard version of PSO:

$$\boldsymbol{v_{t+1}} = \alpha \boldsymbol{v_t} + c_1 r_1 (\boldsymbol{l^*} - \boldsymbol{p}) + c_2 r_2 (\boldsymbol{g^*} - \boldsymbol{p}), \qquad (6)$$

where α, c_1 and c_2 are constant values, and r_1 and r_2 are random values. It is worth noting that only the points explore the space, whereas the normals are associated to the points in a fixed way. Therefore the velocity corresponds to an actual velocity vector in the space. We can use this formulation under the assumption that close points have similar normals; therefore if the velocity is small, the particle computed in the next iteration along with its associated normals will be similar to the current one. Obviously the velocity is a continuous variable and may return a particle $\boldsymbol{p}$ whose points are not contained in our set P; therefore, as in [21], at each iteration we select the set of points in P that are closest to $\boldsymbol{p}$. When the algorithm terminates, the best particle found is returned.

An ideal situation would be to analyze every acceptable triplet through inverse kinematics, in order to check whether it is actually feasible for the robot. This is not viable for computational reasons; indeed, even though the inverse kinematics algorithm is fast and reliable, it takes 500 milliseconds to find a solution. Therefore, running it for every triplet found by DPSO – which could be a very large number – is not computationally feasible. We need to develop an algorithm that can be performed in real-time, thus we rely on computing a limited number of the best triplets that DPSO finds, and we let the kinematics pick the best one. In the current setting we run 4 parallel DPSO procedures, one for each hand shape, and we select the best one for each DPSO routine. If requested, it would be possible to keep track of a bigger number of the best ranked triplets for each DPSO routine. These triplets are analyzed by the kinematics of both the left and the right hand, thus the algorithm can choose among 8 different solutions in total.

IV. INVERSE KINEMATICS

Given a triplet T, we are asked to find a feasible configuration $(\boldsymbol{x}, \boldsymbol{o}, \boldsymbol{q})$ for the robot's hand. Specifically, the inverse kinematics algorithm should provide the end-effector position $\boldsymbol{x} \in \mathbb{R}^3$ and orientation $\boldsymbol{o} \in \mathbb{R}^3$ in the Cartesian space, and finger configuration $\boldsymbol{q} \in \mathbb{R}^8$ in the joint space. We use the thumb, the index and the middle finger of the iCub's hand, which are equipped respectively with 3, 3 and 2 joints. The end-effector position and orientation correspond to the region of the object towards which the hand will move. The finger configuration (or hand preshape) corresponds to the angles to which finger joints are set to reach the contact points.

We formulate the inverse kinematics (IK) problem as an optimization task, similarly to [18] and [11], and we use *IPOPT* [29] to solve it. However, our problem is slightly different with respect to [18] and [11], as we are dealing with discrete point clouds; therefore we cannot use a model of the object. In addition, we are only interested in the inverse kinematics part, whilst [18] and [11] take into account a larger number of variables. In particular, given the triplet $T = (\{\boldsymbol{p_1}, \boldsymbol{n_1}\}, \{\boldsymbol{p_2}, \boldsymbol{n_2}\}, \{\boldsymbol{p_3}, \boldsymbol{n_3}\})$, the object center $\boldsymbol{x_o}$ and its covariance matrix D containing information about its dimension, we minimize with respect to $\boldsymbol{x}$, $\boldsymbol{o}$ and $\boldsymbol{q}$ the following problem:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{o},\boldsymbol{q}} \quad & \sum_{i=1}^{3} \boldsymbol{z_i} \boldsymbol{n_i} \\ \text{s. t.} \quad & ||\boldsymbol{f_i} - \boldsymbol{p_i}|| < \epsilon, \text{ for } i = 1, \ldots, 3 \\ & (\boldsymbol{x} - \boldsymbol{x_o})^T D^{-1} (\boldsymbol{x} - \boldsymbol{x_o}) > 1 \\ & l_i < q_i < u_i, \text{ for } i = 1, \ldots, m, \end{aligned}$$

where l_i and u_i are the lower and upper limits for each joint, m is the total number of joints, $\boldsymbol{z_i}$ is the direction of the force exerted by the fingertip and $\boldsymbol{f_i}$ is the position of the $i-th$ fingertip, computed via forward kinematics of the hand. The cost function represents the fact that the force exerted by the

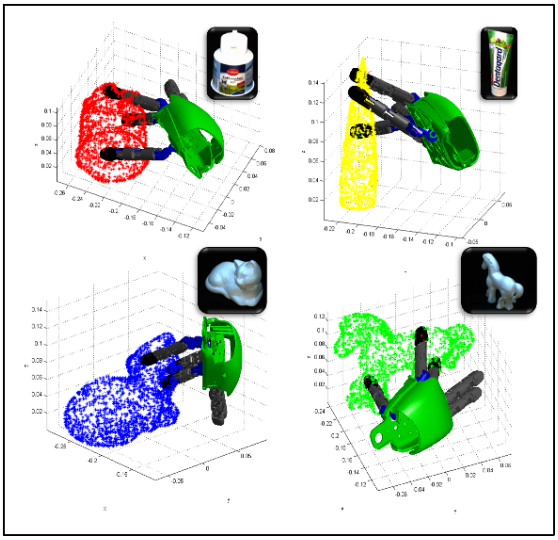

Fig. 4. The picture illustrates grasp results on some synthetic data taken from the KITObjectModels WebDatabase [32].

fingertip should be opposite with respect to the normal on the point. The first set of constraints takes care of minimizing the distance between the fingertip position and the contact point. The second constraint prevents the hand from colliding with the object, imposing that the end-effector should lie outside the object. The last set of constraints controls that the joint angles do not exceed their physical bounds.

This minimization problem is solved for all the triplets retrieved by the step described in Sec. III, and 8 solutions are found – 4 for the right hand and 4 for the left hand. Besides the value of the cost function, which already provides a good measure of a configuration's feasibility, we would need to consider also the whole robot's arm. Most of the methods dealing with precision grasp work in simulation, and they do not account for the robot's arm, assuming that the hand can move freely. Differently, we would like to use the proposed framework on real robots in the future; to this end, we take a preliminary step in considering the hand as being connected to the arm, which is bound to the rest of the robot's body. A good measure of how a certain configuration is suitable for a robot's joints configuration is provided by the standard manipulability [30]. To compute this quantity, we make use of the method in [31] to solve the inverse kinematics (IK) of the arm and the torso. It provides the joint configuration that satisfies the desired position and orientation of the hand using 10 degrees of freedom (7 for the arm and 3 for the torso). The solution that minimizes the cost function and maximizes the manipulability is finally selected.

V. EXPERIMENTAL RESULTS

We first qualitatively show that our method is suitable for dealing with complete 3D models of the objects. To this end, we present some results on the KITObjectModels WebDatabase [32] in Fig. 4, which contains complete 3D point clouds of several objects of different shapes.

The scope of our paper though, is showing that our framework is able to handle also incomplete, raw 3D point clouds. Therefore we assessed our method on the 8 real objects showed in Fig. 5. The objects have been reconstructed using the stereo vision of the iCub. As we are not provided with the model of the objects, we do not know their friction coefficients; we then employed a friction coefficient of 0.6, since smaller values (like in [33]) can prevent from finding stable triplets. This is due to two factors: first, we retrieve triplets only on the visible part of the object, therefore not all the possible force-closure triplets are evaluated. Second, the force-closure property is computed on the basis of the 3D reconstruction of the object, which usually does not reflect perfectly its surface. If no stable triplets are found, likely the visible part of the object does not present suitable contact points. We perform 10 different trials on each object, for a total of 80 trials. For each run we launched 4 parallel DPSO routines, resulting in 4 triplets. We used 20 particles for DPSO, which are initialized randomly. The inverse kinematics of the hand was computed for both left and right arms, obtaining 8 different solutions for each trial. Among these configurations, the one presenting the lowest value of the cost function and the highest manipulability was selected. The entire pipeline, from the object reconstruction to the retrieval of the grasp configuration, takes around $4-5$ seconds in total with a standard deviation of 1 second.

As grasping lacks of a standardized benchmark, we proceeded analyzing the resulting configurations as follows: if the fingertips are in contact with the object and there is no interpenetration among the fingers and the object, each finger is lying in a feasible position without crossing the other fingers, and, finally, the hand is in a suitable position for the robot, then the grasp is labelled as successful. We achieved an overall success rate of 85%. A detailed analysis of such accuracy is illustrated in Fig. 5. It is worth noting that the lowest accuracy is reported by the monkey, which is the biggest object we have used. In all the failure cases, the retrieved triplets covered a too big area for the hand; probably the DPSO algorithm was not able to find triplets covering a smaller area. The other object that registered a low accuracy is the watermelon, which is harder to reconstruct because of its flat faces; indeed, using stereo vision on flat, continuous surfaces may bring to inaccurate point projections due to the lack of matches between the left and the right image. Overall, most of the failure cases are due to the reconstruction of the object through stereo vision, which restrains DPSO from finding good triplets. Notably, thanks to the manipulability measure included in the second stage of our framework, all the selected grasps are feasible for the robot's position and do not entail singular or unnatural configuration.

To better discuss the reasons underlying the high success rate recorded in the simulations, we report hereinafter on further evaluations carried out on the two components of the proposed pipeline: the DPSO and the IK routines.

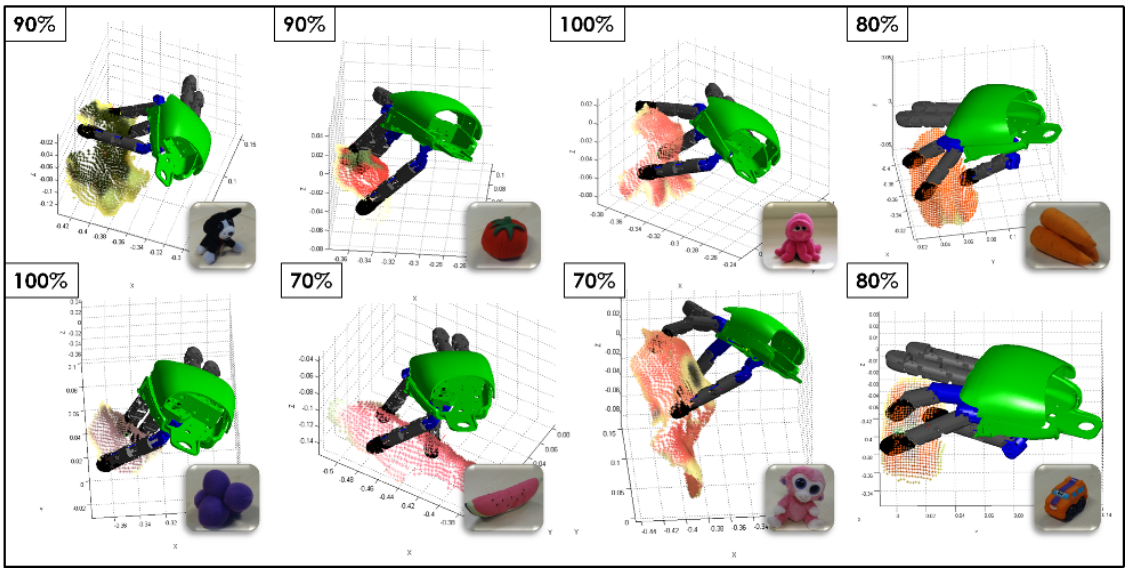

Fig. 5. The picture shows the 8 objects used to assess our algorithm. For each object is also depicted its 3D reconstruction, and the final hand configuration computed by our precision grasp pipeline. The ring and the little fingers are not used.

A. DPSO

The first stage of our algorithm is extremely accurate in retrieving only stable contact points. Indeed, on 80 trials there was only a single case where DPSO did not find any stable triplet. This was due to the reconstruction of the object, which provided a distorted point cloud. This leads to an inaccurate computation of the surface normals, and therefore to a failure in retrieving stable triplets. On the other hand, we aim at measuring how many triplets retrieved by DPSO were actually feasible for the robot's hand. In Table I we present, for each object, the percentage of feasible triplets provided by DPSO. In Table I it is also possible to review the number of times a specific grasp shape has been chosen. Notably, $shape_3$ is not reported in the table because it is never selected; indeed, as we mentioned, it is not suitable for the iCub's hand. Differently, $shape_2$ and $shape_1$ are the most commonly selected, as we expected.

TABLE I
EXPERIMENTAL RESULTS ON DPSO

Object	Feasible Triplets	shape_0	shape_1	shape_2
Car	0.9	0	5	5
Carrot	0.9	0	1	9
Dog	0.9	1	3	6
Grapefruit	1.0	0	5	5
Monkey	0.7	2	3	4
Octopus	1.0	0	3	7
Tomato	0.9	0	3	7
Watermelon	0.7	1	1	8

In order to demonstrate that DPSO performs effectively in our context, we compare it to a pure random search approach (RS), where triplets are chosen randomly and then evaluated. We perform two different experiments on the octopus. We first carry out an exhaustive search to obtain the triplet T^* that solves the problem (5), being c^* the corresponding minimum cost. We then measure the amount of time DPSO and RS take to find the best solution over 10 trials. RS takes $4.71s$, whereas DPSO only takes $0.65s$ on average. Our second experiment consists of setting up a time deadline for the search, which cannot last more than $1s$; when the deadline is reached we collect the best solution found by both the methods. Over 10 trials, DPSO always found the minimum cost solution T^* given by the exhaustive search. RS, instead, was not able to find any good triplet in one case, as $1s$ was not enough. In the other trials we obtained solutions that, on average, implied a cost 180% higher than c^*.

B. Inverse Kinematics

Finally, we assess the accuracy of the IK stage. If the reconstruction is actually accurate and the triplets are feasible, there is the possibility that the inverse kinematics algorithm does not converge to adequate hand configurations. In Table II we provide information in this regard: we compute the mean distance between the desired contact point locations and the fingertip positions, and the mean angle between the normal on the point and the direction of the force exerted by the fingertip on each object, when the grasp has been successful. We remind that the angle should be close to π. In the last column we also report how many times the inverse kinematics algorithm failed for each object, when feasible triplets were found. Overall, IK shows to be effective, achieving small reaching errors.

VI. CONCLUSIONS

We presented a complete, novel algorithm for three-finger precision grasp. The main contribution is the possibility to deal with raw incomplete 3D point clouds. Working on incomplete 3D point clouds instead of requiring complete mod-

TABLE II

EXPERIMENTAL RESULTS ON INVERSE KINEMATICS

Object	Norm [mm]	Angle [rad]	Failures
Car	4.4	2.13	1
Carrot	3.2	2.17	1
Dog	3.1	2.19	0
Grapefruit	3.5	2.13	0
Monkey	3.9	2.23	0
Octopus	3.5	2.36	0
Tomato	3.3	2.25	1
Watermelon	3.3	2.24	0

els is very important as it allows grasping unknown, partially perceived objects. Furthermore, we avoid the approximation of the objects to simpler shapes, meaningfully decreasing the uncertainty about the contact point locations on the object. We thus explore the incomplete 3D point cloud, looking for a number of stable triplets through a variant of the Discrete Particle Swarm Optimization algorithm. The candidate triplets should also satisfy specific properties that are related to the robotic hand that is performing the grasp, as different robotic hands have different grasping capabilities. Therefore, a second contribution is the inclusion of the hand size and shape in the triplet retrieval step. We also have proposed a new quality measure on the hand shape. We then employ *IPOPT* to solve the hand inverse kinematics. Finally, we account for the manipulability of the arm pose when selecting the best grasp configuration.

The main future direction is the refinement of our framework in order to be used safely and reliably on real robots.

REFERENCES

[1] J. Napier, "The prehensile movements of the human hand," *The Journal of bone and joint surgery*, pp. 902–913, 1956.

[2] R. Detry, C. Ek, M. Madry, J. Piater, and D. Kragic, "Generalizing grasps across partly similar objects," *IEEE International Conference on Robotics and Automation*, 2012.

[3] I. Gori, U. Pattacini, V. Tikhanoff, and G. Metta, "Ranking the good points: A comprehensive method for humanoid robots to grasp unknown objects," *IEEE International Conference on Advanced Robotics*, 2013.

[4] M. Roa, M. Argus, D. Leidner, C. Borst, and G. Hirzinger, "Power grasp planning for anthropomorphic robot hands," *IEEE International Conference on Robotics and Automation*, 2012.

[5] J. K. Salisbury and B. Roth, "Kinematic and force analysis of articulated mechanical hands," *Journal of Mechanisms, Transmissions and Automation in Design*, 1983.

[6] V.-D. Nguyen, "Constructing force-closure grasps in 3D," *IEEE International Conference on Robotics and Automation*, 1987.

[7] J. Li, H. Liu, and H. Cai, "On computing three-finger force-closure grasps of 2-D and 3-D objects," *IEEE Transactions on Robotics and Automation*, 2003.

[8] A. Erkan, O. Kroemer, R. Detry, Y. Altun, J. Piater, and J. Peters, "Learning probabilistic discriminative models of grasp affordances under limited supervision," *IEEE International Conference on Intelligent Robots and Systems*, 2010.

[9] D. Rao, Q. V. Le, T. Phoka, M. Quigley, A. Sudsang, and A. Y. Ng, "Grasping novel objects with depth segmentation," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.

[10] C. Borst, M. Fischer, and G. Hirzinger, "Calculating hand configurations for precision and pinch grasps," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2002.

[11] C. Rosales, R. Suárez, M. Gabiccini, and A. Bicchi, "On the synthesis of feasible and prehensile robotic grasps," *IEEE International Conference on Robotics and Automation*, 2012.

[12] A. Sahbani and S. El-Khouri, "An overview of 3D object grasp synthesis algorithms," *Robotics and Autonomous Systems*, 2011.

[13] S. El-Khoury and A. Sahbani, "A new strategy combining empirical and analytical approaches for grasping unknown 3D objects," *Robotics and Autonomous Systems*, 2010.

[14] M. A. Roa, K. Hertkorn, C. Borst, and G. Hirzinger, "Reachable independent contact regions for precision grasps," *IEEE International Conference on Robotics and Automation*, 2011.

[15] M. A. Roa and R. Suárez, "Independent contact regions for frictional grasps on 3d objects," *IEEE International Conference on Robotics and Automation*, 2008.

[16] S. Geidenstam, K. Huebner, D. Banksell, and D. Kragic, "Learning of 2D grasping strategies from box-based 3D object approximations," *Robotics: Science and Systems Conference*, 2009.

[17] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," *IEEE International Conference on Robotics and Automation*, 2003.

[18] S. El Khoury, M. Li, and A. Billarde, "Bridging the gap: One shot grasp synthesis approach," *International Conference on Intelligent Robots and Systems*, pp. 2027–2034, 2012.

[19] S. El-Khoury and A. Sahbani, "On computing robust n-finger force-closure grasps of 3d objects," *IEEE International Conference on Robotics and Automation*, 2009.

[20] G. Metta, G. Sandini, D. Vernon, L. Natale, and F. Nori, "The icub humanoid robot: an open platform for research in embodied cognition," in *8th Workshop on Performance Metrics for Intelligent Systems*, 2008.

[21] H. Nahvi and I. Mohagheghian, "A particle swarm optimization algorithm for mixed variable nonlinear problems," *International Journal of Engineering*, pp. 65–78, 2011.

[22] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *TPAMI*, 2008.

[23] B. R. Radu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universität München, Germany, October 2009.

[24] V.-D. Nguyen, "Constructing force-closure grasps," *International Journal of Robotics Research*, 1988.

[25] A. T. Miller and P. K. Allen, "Examples of 3d grasp quality computations," *IEEE International Conference on Robotics and Automation*, 1999.

[26] R. Suárez, M. Roa, and J. Cornella, "Grasp quality measures," *Technical report*, 2006.

[27] C. Borst, M. Fischer, and G. Hirzinger, "Grasping the dice by dicing the grasp," *IEEE International Conference on Intelligent Robots and Systems*, 2003.

[28] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," *IEEE international conference on neural networks*, 1995.

[29] A. Wätcher and L. Bielger, "On the implementation of a primaldual interior point filter line search algorithm for large-scale nonlinear programming," in *Mathematical Programming*, 2006, pp. 25–57.

[30] M.-J. Tsai, "Workspace geometric characterization and manipulability of industrial robots," Ph.D. dissertation, Ohio State University, 1986.

[31] U. Pattacini, F. Nori, L. Natale, G. Metta, and S. G., "An experimental evaluation of a novel minimum-jerk cartesian controller for humanoid robots," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 1668–1674, 2010.

[32] A. Kasper, Z. Xue, and R. Dillmann, "The kit object models web database: An object model database for object recognition, localizatino and manipulation in service robotics," *The International Journal of Robotics Research*, 2012.

[33] K. Matheus and A. M. Dollar, "Benchmarking grasping and manipulation: Properties of the objects of daily living," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010.