Visual Grasp Planning for Unknown Objects Using a Multi-Fingered Robotic Hand

Vincenzo Lippiello, Fabio Ruggiero, *Member, IEEE,* Bruno Siciliano, *Fellow, IEEE, ASME,* and Luigi Villani, *Senior Member, IEEE*

***Abstract*—A method for fast visual grasping of unknown objects with a multi-fingered robotic hand is presented in this paper. The algorithm is composed of an object surface reconstruction algorithm and a local grasp planner, evolving in parallel. The reconstruction algorithm makes use of images taken by a camera carried by the robot arm. A virtual elastic reconstruction surface is placed around the object. The surface shrinks toward the object until some points intercept the object visual hull. Then, attractive forces with respect to the border of the visual hull are generated so as to compensate for the elastic forces: when an equilibrium between those forces is reached, the surface takes the form of the object shape. Running in parallel to the reconstruction algorithm, the grasp planner moves the fingertips on the current available reconstruction surface, towards points which are optimal (in a local sense) with respect to a number of indices weighting both the grasp quality and the kinematics configuration of the hand. This method, referred to as parallel visual grasp, may represent a valid candidate for applications where online grasp planning is required. A number of experiments are presented, showing the effectiveness of the proposed approach.**

***Index Terms*—Robotics, Vision, Surface Reconstruction, Grasping, Multi-Fingered Hands.**

I. Introduction

THE execution of robot grasping tasks, in general, requires a priori knowledge about the characteristics of the objects to grasp. The adoption of vision can be useful to reduce the need of a priori information [1], [2].

Two main operations have to be performed for grasping unknown objects [3]: recognition/reconstruction of the object geometry and grasp planning.

Different methods have been proposed in the literature to cope with 3D geometric model reconstruction based on visual data. A certain number of algorithms can be classified under the so called *volumetric scene reconstruction* approach [4], while other methods are referred to as *surface scene reconstruction* algorithms: these latter are the most suitable for grasp planning. In [5], the model of the object is obtained starting from a surface that moves towards the object under the influence of internal forces, given by the surface itself, and external forces, given by the image data. Typically, the starting surface is a sphere: this approach may be considered as a generalization of snakes used in 2D. A finite-element method is adopted in [6] to reconstruct both 2D and 3D boundaries of the object. Using an active contour model, data extracted from images are employed to generate a pressure force on the active contour which inflates or deflates the curve, making its behavior like a balloon. A technique for computing a polyhedral representation of the so called visual hull [7] is studied in [8]: in such approach, only the contours of the silhouettes in the images have to be visited, and the computed visual hull is quickly represented. Furthermore, other methods rely on the use of apparent contours such as in [9]: in these cases, the reconstruction is based on a spatio-temporal analysis of deformable silhouettes. Other methods for object reconstruction rely on the use of several kinds of sensors, like in the case of vision-based structured light systems [10], [11]. Finally, a method to grasp an unknown object using information provided by a deformable contour model algorithm is proposed in [12].

As for grasp planning techniques, they rely upon the choice of grasp quality measures used to select suitable grasp points on the object surface. Several quality measures proposed in the literature depend on the position of the contact points (algebraic properties of the grasp matrix, geometry of the grasp area of the polygon created by the contact points and so on), while others depend on the finger forces. Two optimal criteria are introduced in [13], where the total finger force and the maximum finger force are considered, while measures based on algebraic properties of the grasp matrix and a measure based on the task to accomplish are presented in [14]. A number of quality measures is based on the evaluation of the capability of the hand to realize the optimal grasp [15]. A rich survey of grasp quality measures can be found in [16]. Only a few papers address the problem of grasp planning by taking into account quality measures depending on both object geometry and hand kinematics [17]–[19].

Also the preshaping of a robotic hand –the preparation of the hand to grasp the object– is a non-trivial prior step to grasping [20]. In the literature, several methods deal with this problem. Most of them rely on a previous knowledge learned from humans [21]; other methods rely on the use of vision [22], fuzzy logic [23], neural networks [24], and are based on rough approximations of the geometry of the object to grasp [25], and generally are task-dependent [26].

In this paper, a method for fast visual grasping of unknown

The authors are with PRISMA Lab, Dipartimento di Informatica e Sistemistica, Università degli Studi di Napoli Federico II, via Claudio 21, 80125, Naples, Italy, {vincenzo.lippiello, fabio.ruggiero, bruno.siciliano, luigi.villani}@unina.it.

The research leading to these results has been supported by the DEXMART Large-scale integrating project, which has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement ICT-216239. The authors are solely responsible for its content. It does not represent the opinion of the European Community and the Community is not responsible for any use that might be made of the information contained therein.

Manuscript received on September 26, 2011; revised on January 24, 2012.

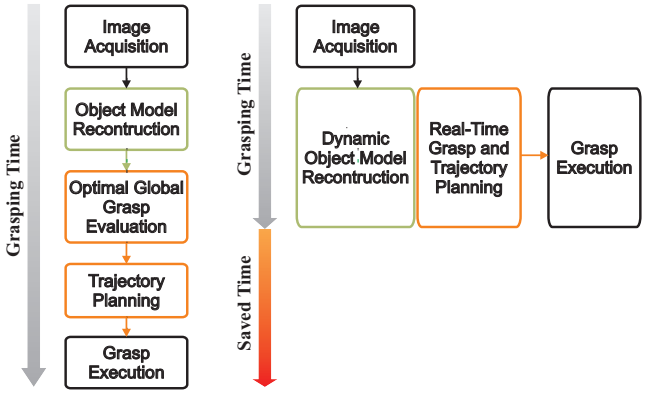

Fig. 1. Classical serial algorithm vs. proposed parallel visual grasp algorithm.

objects using a camera mounted on a robot in an eye-in-hand configuration is presented. The method is composed of an *object surface reconstruction algorithm* and a *grasp planner*, which evolve in a synchronized parallel way. The reconstruction algorithm makes use of a virtual elastic surface of ellipsoidal initial shape, with axes and dimensions computed by a suitable object preshaping process, placed around the object. The surface shrinks toward the object until some points intercept the object visual hull. Then, elastic forces are compensated by repulsive forces at the border of the visual hull. At the equilibrium, the surface assumes the shape of the object. Running in parallel to the reconstruction algorithm, the grasp planner moves the fingertips, starting from a suitable preshape configuration, towards points of the current available reconstruction surface, which are optimal (in a local sense) with respect to a number of indices weighting both the grasp quality and the kinematics configuration of the hand.

The proposed approach, based on preliminary results presented in [27], is referred to here as a "parallel visual grasp". It may represent a valid candidate for applications where online grasp planning is required, as confirmed by a number of case studies presented in the paper.

II. Parallel Visual Grasp Algorithm

The typical approach to grasping unknown objects, here called serial visual grasp algorithm, consists of two stages, as shown in Fig. 1: in the first stage, the geometry of external surface of the object is completely reconstructed from the acquired images; in the second stage, the grasping points on the object, optimal under a selected global criterion, are computed first, and then the corresponding trajectory for the robotic hand is planned and executed. This approach gives the best results in terms of grasp quality, since the evaluation of the optimal grasp is made in a global way. However, the total execution time, given by the sum of the time required for the reconstruction of the object geometric model and that required for the synthesis, planning and execution of the grasp, may be considerable, if powerful hardware is unavailable. Obviously, this is irrelevant for off-line applications, but it could be a serious drawback for online grasp planning.

The method proposed here, referred to as the parallel visual grasp algorithm, may represent a valid candidate for real-time

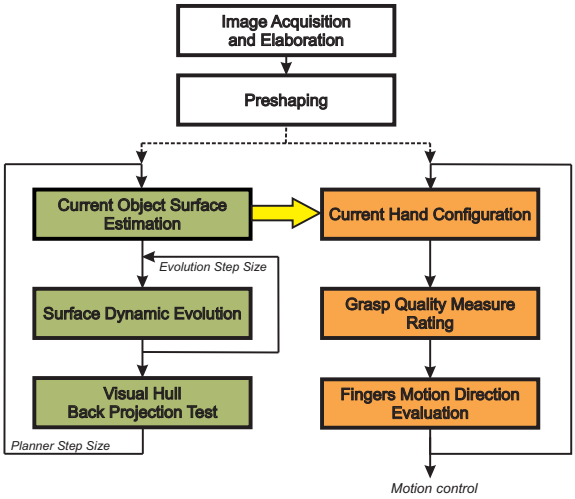

Fig. 2. Block diagram of the parallel visual grasp algorithm.

applications, because the total computational time is given by the slower between the reconstruction and the planning stage, while the grasp execution can start in parallel with the previous processes (see Fig. 1). As a matter of fact, object model reconstruction and grasp planning are independent processes and can also be allocated to different computational resources.The drawback is that the achieved final grasp is optimal only in the local sense.

Notice that, in the classical serial approach, any algorithm can be used for the reconstruction of the geometry of the external surface of the object starting from images. The idea here is that of using a method based on the continuous deformation of a virtual elastic surface, sampled by points and enclosing the object, which moves toward the object under the influence of the elastic forces and of repulsive forces. The last ones arise when parts of the surface penetrate in the object visual hull. The intermediate configurations of the surface are used by the grasp planning algorithm to compute the intermediate configurations of the hand toward the final grasp. Having a surface which shrinks under the influences of the elastic forces can be seen as a particular case of the level-set method proposed in [28], [29] for object reconstruction: in such works, one or more surfaces evolve in the direction of the steepest descent provided by the variation calculation of a given functional to be minimized. In this paper, instead, the functional is given by the potential elastic force which acts as an external force at each sampled point of the considered surface. Moreover, different from point-based reconstruction algorithms [30], the presence of the virtual elastic fibers connecting the points of the surface allows achieving a more uniform distribution of the points on the object visual hull, which is important for the computation of the optimal grasp.

The block diagram in Fig. 2 shows more details of the proposed visual grasp algorithm.

The procedure begins with some *preparatory steps*, consisting in the acquisition of a number of images using a camera mounted on the robot arm, from which a rough shape of the object, in the form of an ellipsoid, is computed (object preshaping); moreover, the initial grasp configuration of the

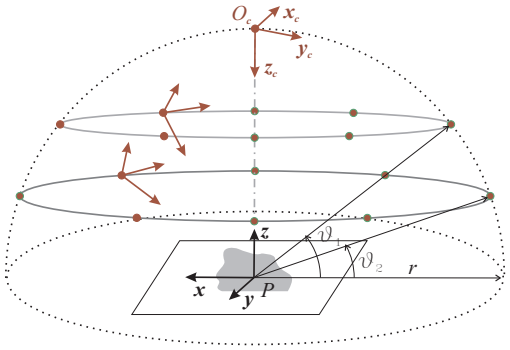

Fig. 3. Camera stations (bullets) and trajectories of the camera during the acquisition of the images.

hand is selected (hand preshaping).

At this point, both the *object model reconstruction* algorithm (green blocks) and the *grasp planning* algorithm (orange blocks) start in parallel and incrementally concur to the final goal. In particular, the reconstruction algorithm updates the estimation of the current reconstructed surface of the object, while the grasp planner, on the basis of the current estimation, computes the trajectories of the fingertips toward a (local) optimal grasp configuration; these incremental trajectories are executed by the motion control of the hand.

The single stages of the algorithm: preparatory steps, object model reconstruction, grasp planning, trajectory planning and motion control, are described in the following sections.

The assumptions made throughout this work are:

- An eye-in-hand configuration with a calibrated camera is available for image acquisition.
- The observed rigid object is fixed in the space and distinguishable w.r.t. the background and other objects.
- From a topological point of view, the object is a connected orientable surface with genus 0, i.e., without holes or handles.
- Any multi-fingered robotic hand mounted on a robot manipulator can be considered.

III. Preparatory Steps

A detection algorithm, based on a classic blob analysis, allows recognizing the presence of the object on a plane in the field of view of the camera, mounted on the robot arm. Then, by keeping the optical axis perpendicular to the plane, the camera is moved until the optical axis intercepts the estimated centroid P of the object. Hence, the process of acquisition of the images of the object can be started.

A. Image acquisition and elaboration

The images are taken from the stations (bullets) illustrated in Fig. 3, where frame O_c-$x_c y_c z_c$ is the cartesian frame attached to the camera (camera frame) in the initial configuration, with the z_c axis aligned to the optical axis and the origin O_c at an estimated distance r from P.

The acquisition process is carried out by moving the camera on a sphere of radius r centered at P, in order to have a constant image resolution, and by stopping the robot at each station. Namely, a first image is acquired from the initial configuration and a set of n_1 images is taken from camera stations equally distributed over a circular path on the sphere, at an elevation angle ϑ_1, with the optical axis of the camera pointing to P. In general, further acquisition steps can be considered, at different elevation angles (see Fig. 3), depending on the complexity of the object shape. It is obvious that both the distribution of the acquisition stations and the number n_{img} of acquired images affect the accuracy of the reconstruction process. The value of r, instead, is set so as to keep the object in the camera field of view from all the stations.

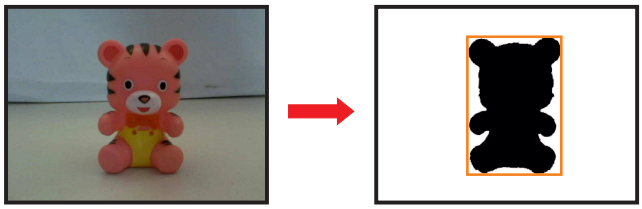

Fig. 4. Silhouette and bounding box extracted from an image of the object.

The n_{img} images are elaborated to extract the object silhouettes: to this aim, a simple binarization process, with a self-tuned threshold, is employed. A process of a binary dilation and erosion may be required to reduce the effects of noise. Additional filtering of the images, both in the spatial and frequency domains, can be performed to reduce noise and disturbances such as the presence of shadows in the views.

Once the silhouette for a given image is obtained, it is straightforward to determine the corresponding bounding box just considering the smallest rectangle which contains the whole silhouette. Obviously, this process must be performed for all the n_{img} acquired images.

Fig. 4 shows an image with the resulting silhouette and bounding box.

B. Object preshaping

The proposed method for object preshaping starts from a concept presented in [31], where a rough estimation of the object shape is computed by using a linear programming technique.

For each image, the four planes of the Cartesian space containing the origin of the camera frame and two adjacent vertices of the corresponding bounding box in the image plane are considered, resulting in $4n_{img}$ Cartesian planes. Each plane splits the Cartesian space into two regions, one of which contains the object visual hull $\mathcal{V}$. The intersections of all these regions create a polyhedron $\mathcal{P}$ containing $\mathcal{V}$, which is a polyhedric overestimation of the visual hull.

The vertices $\boldsymbol{x}$ of this polyhedron can be quickly computed as follows. Since each side of each bounding box is associated to a plane, if the normal unit vector to the plane is chosen pointing outwards with respect to the interior side of the bounding box, the inner space created by the considered planes for a given image is represented by the following set of inequalities:

$$\boldsymbol{A}_i \boldsymbol{x} \leq \boldsymbol{d}_i,$$

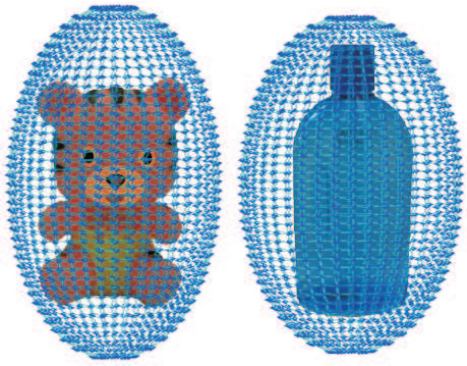

Fig. 5. Examples of ellipsoids used for object preshaping.

where subscript i denotes the i-th image, with $i = 1, \ldots, n_{img}$, $\boldsymbol{A}_i$ is a (4×3) matrix whose rows are the transpose of the normal unit vectors, and $\boldsymbol{d}_i$ is a (4×1) vector whose elements define uniquely the positions of the planes in the space. Stacking all the $\boldsymbol{A}_i$ and $\boldsymbol{d}_i$ in the matrices $\boldsymbol{A}$ and $\boldsymbol{d}$, the inner space of the polyhedron $\mathcal{P}$ is represented by the following inequality:

$$\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{d}.$$

The vertices of the polyhedron are the so called *basic feasible solutions*, whose computation is well known in the literature. The required computational time is very low and depends only on the number n_{img} of images.

Once all the n_v vertices $\boldsymbol{x}_v = \begin{bmatrix} x_{v_x} & x_{v_y} & x_{v_z} \end{bmatrix}^T$ of the polyhedron $\mathcal{P}$ have been computed, the central moments can be evaluated as:

$$\mu_{i,j,k} = \sum_{\boldsymbol{x}_v \in \mathcal{P}} (x_{v_x} - \bar{x}_{v_x})^i (x_{v_y} - \bar{x}_{v_y})^j (x_{v_z} - \bar{x}_{v_z})^k,$$

where

$$\bar{\boldsymbol{x}}_v = \begin{bmatrix} \bar{x}_{v_x} & \bar{x}_{v_y} & \bar{x}_{v_z} \end{bmatrix}^T = \frac{1}{n_v} \sum_{i=1}^{n_v} \boldsymbol{x}_{v_i}$$

is the centroid of the polyhedron.

Finally, a pseudo-inertia tensor of the polyhedron can be defined as:

$$\boldsymbol{I} = \begin{bmatrix} \mu_{2,0,0} & \mu_{1,1,0} & \mu_{1,0,1} \\ \mu_{1,1,0} & \mu_{0,2,0} & \mu_{0,1,1} \\ \mu_{1,0,1} & \mu_{0,1,1} & \mu_{0,0,2} \end{bmatrix},$$

where its eigenvalues and eigenvectors define the principal axes of inertia of an ellipsoid, suitably enlarged to ensure object wrapping (see Fig. 5), which is employed as the initial shape of the reconstruction surface.

C. Hand preshaping

Depending on the object shape, the ellipsoid may have one axis bigger/smaller than others, or all axes of similar dimension. For all these cases, a good choice of the grasp points on the object and of the initial grasp configuration of the hand depends also on the task to accomplish (e.g. pick-and-place, manipulation, assembling, etc.), on the type of grasp to perform (firm or fine), on the environmental constraints (e.g. the ground plane), on the hand kinematics and number of fingers. When firm grasp is considered, the axis of approach of the hand is typically chosen parallel to the major axis of the ellipsoid. On the other hand, for the fine manipulation case, the choice of the axis of approach depends on several factors.

In this paper, without loss of generality, the initial grasp points on the ellipsoid are chosen to form an equilateral grasp [32] in a plane parallel to the two minor axes of the ellipsoid, assuming that the corresponding grasp configuration for the hand is reachable for the given hand kinematics and environmental constraints.

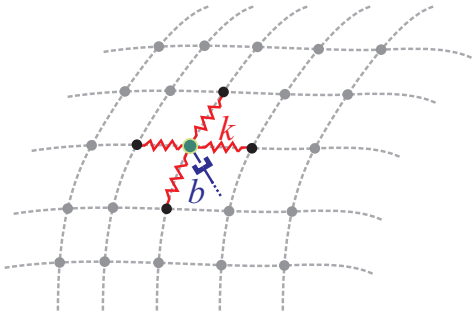

Fig. 6. Piece of the sampled reconstruction surface with the virtual mass, stiffness and damping of the i-th sample points.

IV. Object Model Reconstruction

The 3D ellipsoid is virtually placed around the object, centered at $\bar{\boldsymbol{x}}_v$ and sampled with n_s points. These points are set at the intersections of ideal meridian and parallel lines drawn on the external surface of the ellipsoid. Without loss of generality, the number of parallels n_p is chosen equal to the number of meridians n_m.

Four virtual elastic links of stiffness k connect each sample point with the four closest points. The two poles of the ellipsoid are connected with all the points of the nearest parallel of the grid. Hence, the sampled reconstruction surface behaves like a virtual membrane composed by a network of ideal elastic fibers connecting the points (see Fig. 6). A virtual viscous damping b, with respect to the ground, is also considered for each point. A damping effect between point cloud could also be defined, but it is not considered here for simplicity.

In order to derive the dynamics of each sample point of the ellipsoid, the balance of forces at a given point can be written as follows:

$$\begin{aligned} b\dot{\boldsymbol{x}}_{i,j} \quad &+ \quad k(\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i-1,j}) + k(\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i+1,j}) \\ &+ \quad k(\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i,j+1}) + k(\boldsymbol{x}_{i,j} - \boldsymbol{x}_{i,j-1}) = \boldsymbol{f}_{i,j}(\boldsymbol{x}_{i,j}) \end{aligned}$$

for $i = 1, \ldots, n_m$ and $j = 1, \ldots, n_p$, where $\boldsymbol{x}_{i,j}$ is the position in the workspace of the sampling point at the intersection of the i-th meridian with the j-th parallel. Vector $\boldsymbol{f}_{i,j}$ is instead the external force acting on point $\boldsymbol{x}_{i,j}$, which is repulsive with respect to the border of the visual hull $\mathcal{V}$ and is different from zero only when $\boldsymbol{x}_{i,j}$ comes into $\mathcal{V}$:

$$\boldsymbol{f}_{i,j}(\boldsymbol{x}_{i,j}) = \begin{cases} \alpha_{i,j} F_a \boldsymbol{n}_{i,j}, & \boldsymbol{x}_{i,j} \in \mathcal{V} \\ 0, & \boldsymbol{x}_{i,j} \notin \mathcal{V} \end{cases}$$

where $\boldsymbol{n}_{i,j}$ is the unit vector normal to the surface at $\boldsymbol{x}_{i,j}$, pointing out from the object, and $\alpha_{i,j} F_a$ is the amplitude of

the force. In detail, F_a is the maximum force module and $\alpha_{i,j} \in (0,1]$ is a strictly decreasing sequence of scale factors defined as:

$$\alpha_{i,j}(k+1) = \epsilon \alpha_{i,j}(k), \quad \alpha_{i,j}(0) = 1$$

where $\epsilon \in (0,1)$ and a new value of the sequence is computed every time point $\boldsymbol{x}_{i,j}$ comes out from $\mathcal{V}$.

Collecting some terms in the previous dynamic equation of the system, a more compact expression is

$$b\dot{\boldsymbol{x}}_{i,j} + k\left(4\boldsymbol{x}_{i,j} - \boldsymbol{c}(\boldsymbol{x}_{i,j})\right) = \boldsymbol{f}_{i,j}(\boldsymbol{x}_{i,j}),$$

where $\boldsymbol{c}(i,j) = \boldsymbol{x}_{i-1,j} + \boldsymbol{x}_{i,j+1} + \boldsymbol{x}_{i+1,j} + \boldsymbol{x}_{i,j-1}$. In these last expressions, the points with subscript $i-1 = j-1 = 0$ ($i+1 = n_m+1$ and $j+1 = n_p+1$) coincides with those with subscripts $i-1 = n_m$ and $j-1 = n_p$ ($i+1 = j+1 = 1$), respectively. The two poles have to be treated separately, due to their topological peculiarity, namely:

$$b\dot{\boldsymbol{x}}_{np} + k\Big(n_m\boldsymbol{x}_{np} - \sum_{j=1}^{n_m}\boldsymbol{x}_{1,j}\Big) = \boldsymbol{f}_{np}(\boldsymbol{x}_{np})$$

for the north pole, and

$$b\dot{\boldsymbol{x}}_{sp} + k\Big(n_m\boldsymbol{x}_{sp} - \sum_{j=1}^{n_m}\boldsymbol{x}_{n_p,j}\Big) = \boldsymbol{f}_{sp}(\boldsymbol{x}_{sp})$$

for the south pole, where the subscripts np and sp indicate quantities referred to the north and south pole, respectively.

The dynamics of the system, for any non-trivial initial condition of the ellipsoid, leads the elastic surface to shrink toward its center until the visual hull is intersected. The equilibrium is reached when the elastic forces are compensated by the repulsive forces at the border of the visual hull. When this happens, the elastic reconstruction surface wraps the object assuming the shape of the visual hull.

The accuracy of the reconstruction process depends on the distribution of the observation stations and increases with the number of views n_{img} and the density of the points n_s of the reconstruction ellipsoid. On the other hand, the computational time of the algorithm increases if n_{img} and/or n_s are increased. Hence, a compromise between performances and accuracy should be thus decided in order to choose such parameters. By considering that the final goal of the process is that of grasping the object and not the model reconstruction, which can be considered as a secondary outcome of the proposed method, the accuracy of the reconstruction process needs only to be adequate for the requirements of the grasp planning algorithm, as observed also in [33].

V. Grasp Planning

At each time step, the evolution of the virtual elastic surface is frozen and the coordinates of its points are stored in a memory buffer. Then, the grasp planner computes a new set of points on the surface, through a local search algorithm which locally maximizes a suitable grasp quality measure, keeping a fixed safety distance δ_f between the fingertips and the surface. The new set of points is taken as the initial grasp configuration for the next time step, when an update of the current estimation of the object's surface is available. The process ends when the object reconstruction algorithm reaches the equilibrium and the planner computes the final grasp configuration. The safety distance δ_f is employed to avoid undesired collisions between the fingers and the object before the final grasp is reached.

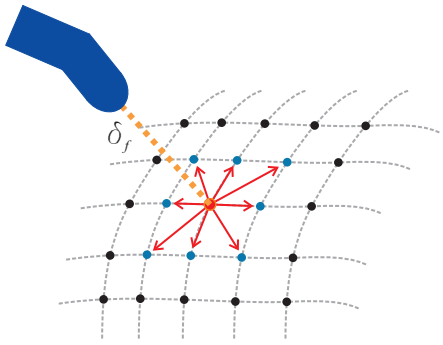

Fig. 7. Possible displacements from the current grasp point.

A. Local search algorithm

The local search algorithm is based on a discretized version of the gradient projection method. In detail, for each grasp point $\boldsymbol{p}_i$, a (virtual) force vector $\boldsymbol{f}_i$, aimed at selecting the direction along which the finger should be moved toward the (sub)optimal grasp configuration, is then computed. The force $\boldsymbol{f}'_i$ can be defined as the projection of the force $\boldsymbol{f}_i$ onto the tangential plane to the surface at point $\boldsymbol{p}_i$:

$$\boldsymbol{f}'_i = \boldsymbol{f}_i - (\boldsymbol{f}_i^T\boldsymbol{v}_i)\boldsymbol{v}_i,$$

where $\boldsymbol{v}_i$ is the normal unit vector to the surface at point $\boldsymbol{p}_i$. If $||\boldsymbol{f}'_i||$ is higher than a given threshold ϵ_f, a new grasp point is set as the neighbor point closest to the direction of $\boldsymbol{f}'_i$ (see Fig. 7).

Obviously, the choice of ϵ_f affects both the accuracy of the grasp solution and the computational time, determining the number of iterations required to converge to the (local) optimum. Therefore, ϵ_f must be carefully tuned on the basis of this trade-off.

B. Grasp quality measure

It is worth noticing that any quality index for the grasp can be used in principle. In this paper, the quality measure is assumed as the sum of a number of quality indices, well recognized in the literature, which allow to improve grasp reliability, namely:

- *Coplanarity*: The contact points on the object belong to the same plane; this condition simplifies the computation of good grasps but, obviously, may exclude a number of grasp configurations that can be more effective than planar ones.
- *Uniform distribution:* In 2D cases, grasp stability– the property of a grasp to resist to external wrenches applied on the object– is improved if the contact points are distributed in an uniform way on the surface [16], [34].
- *Maximum area:* The larger the area of the polygon formed by the contact points, the larger the external torque that the same finger contact forces can resist [34].

- *Center of mass on the grasp plane:* Gravity and inertial effects on contact forces are minimized when the object center of mass belongs to the grasp plane [35], [36].

The above quality indices depend only on the position of the contact points; however, the quality of the grasp is also related to the configuration of the particular robotic hand. This can be accounted for by considering quality indices depending on hand kinematics, namely: distance from finger joint limits, distance from finger kinematic singularities, distance of the fingers from each other and from the palm. Moreover, other quality indices taking into account the real force directions [37] can be considered.

During grasp execution, the contact points do not necessarily belong to the same plane. Therefore, the plane Π which minimizes the distance from all the contact points is considered and it is useful to compute the projection $\boldsymbol{p}_i^{\Pi}$ of contact point $\boldsymbol{p}_i$ on Π (see Fig. 8), with $i = 1, \ldots, n_f$, being n_f the number of fingers of the hand. Moreover, let $\boldsymbol{c}_m$ denote the estimated center of mass of the current shape of the object (assuming uniform mass distribution) and let $\boldsymbol{c}_m^{\Pi}$ be the projection of $\boldsymbol{c}_m$ on Π.

To account for the different quality indices, the (virtual) force vector $\boldsymbol{f}_i$ at contact point $\boldsymbol{p}_i$ is computed as the sum of a number of (virtual) force contributions:

$$\boldsymbol{f}_i = \boldsymbol{f}_{\Pi i} + \boldsymbol{f}_{ei} + \boldsymbol{f}_{ai} + \boldsymbol{f}_{c_m} + \boldsymbol{f}_{bi},$$

where:

- $\boldsymbol{f}_{\Pi i} = k_{\Pi}\left(\boldsymbol{p}_i^{\Pi} - \boldsymbol{p}_i\right)$ is the (virtual) force which moves $\boldsymbol{p}_i$ to $\boldsymbol{p}_i^{\Pi}$, so that all the contact points belong to the same grasp plane.
- $\boldsymbol{f}_{ei} = k_e(\theta_i - 2\pi/n_f)\boldsymbol{t}_i$ is the (virtual) tangential force in charge of producing an equilateral grasp configuration, where θ_i is the angle between vectors $\boldsymbol{p}_i^{\Pi} - \boldsymbol{c}_m^{\Pi}$ and $\boldsymbol{p}_j^{\Pi} - \boldsymbol{c}_m^{\Pi}$, with $j = i + 1$ for $i = 1, \ldots, n_f - 1$, and $j = 1$ for $i = n_f$, and $\boldsymbol{t}_i$ is the tangential unit vector normal to $\boldsymbol{c}_m^{\Pi} - \boldsymbol{p}_i^{\Pi}$ and pointing toward $\boldsymbol{p}_j^{\Pi}$.
- $\boldsymbol{f}_{ai} = k_a\left(\boldsymbol{p}_i^{\Pi} - \boldsymbol{c}_m^{\Pi}\right)/||\boldsymbol{p}_i^{\Pi} - \boldsymbol{c}_m^{\Pi}||$ is the (virtual) force which tends to enlarge the area of the grasp polygon.
- $\boldsymbol{f}_{c_m} = k_{c_m}\left(\boldsymbol{c}_m - \boldsymbol{c}_m^{\Pi}\right)$ is the (virtual) force, equal for all the contact points, which attracts the grasp plane Π to the center of mass $\boldsymbol{c}_m$.
- $\boldsymbol{f}_{bi}$ is a (virtual) barrier force, aimed at avoiding the motion of the fingers along directions that cause the reaching of joint limits, joint or hand singularities, and the collision between fingers or with the palm.

Parameters $k_{\Pi}, k_{c_m}, k_e, k_a$ are all positive constant coefficients, suitably chosen so as to weigh the single force contributions. Notice that the barrier forces can be also employed to cope with environmental constraints, e.g. object ground plane or other surrounding objects, avoiding in such a way that some parts of the surface points, e.g. the lowest ones, can be reached.

VI. Trajectory planning and control

The grasp planner produces a sequence of intermediate target grasp configurations at each iteration of the object's reconstruction algorithm which ends with the optimal grasp configuration (in a local sense). The intermediate configurations are used to generate the fingertips trajectories.

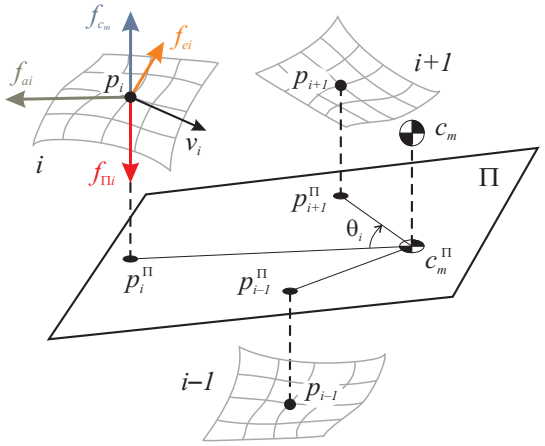

Fig. 8. Forces at contact point $\boldsymbol{p}_i$.

Namely, the sequence of intermediate configurations is suitably filtered by a spatial low-pass filter and interpolated in order to achieve a smooth path for the fingers. In fact, only the final configuration needs to be reached exactly, while the intermediate configurations can be considered as via points for the generation of the trajectories of the fingers, and can be computed online with a one step delay.

Moreover, the actual paths of the fingers generated by the trajectory planner is offset by a safety distance δ_f along the normal to the surface (see Fig. 7). When the final configuration is reached, the offset is progressively reduced to zero, producing the desired grasp action.

Any Cartesian motion control can be used, in principle, for tracking the tip trajectories computed by the planner until the final grasp configuration is reached. However, to control the contact forces required to grasp the object, an interaction control strategy (e.g., impedance control [38]) combined with a real-time force-optimization method [39] can be adopted. Also, the availability of tactile sensing for control is important to allow rapid adjustments of the grip [40].

VII. Experiments

A. Technical details

The experimental set-up implementing the proposed method is composed of a 4-fingered robotic hand made up of 16 Bioloid Dynamixel AX-12 servomotors (see Fig. 9). An industrial USB iDS UEYE UI-1220SE-C camera has been employed in an eye-in-hand configuration, and it has been mounted directly in the center of the palm of the hand. Several calibration algorithms can be adopted, e.g. the *Matlab Calibration Toolbox*[1]. A Windows OS process commands the actuators by providing position signals, while a high-priority multi-thread programming has been required in order to implement the proposed parallel method.

For the image acquisition process, a Comau Smart-Six robot manipulator has been employed to carry around the camera in all the acquisition stations. A number $n_{img} = 13$ images

[1]http://www.vision.caltech.edu/bouguetj/calib_doc/

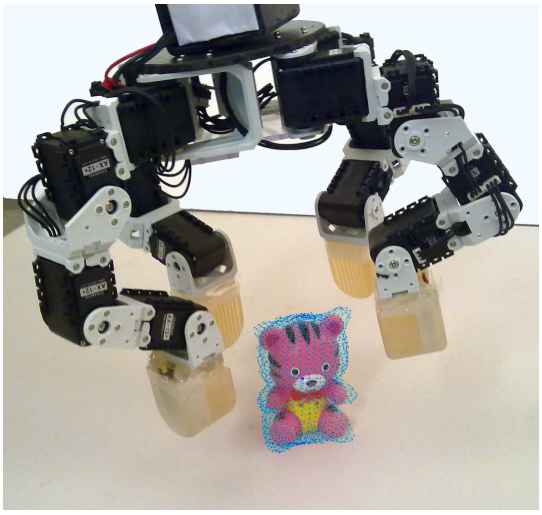

Fig. 9. Hand used for the experiments.

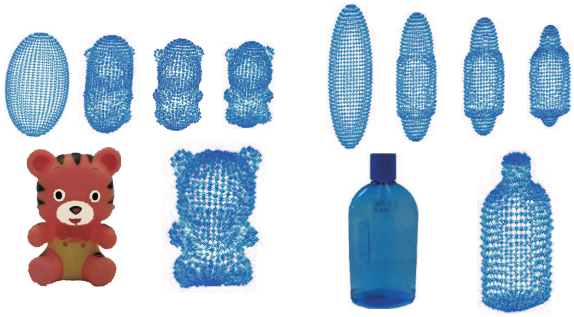

Fig. 10. Steps of the object model reconstruction algorithm: teddy bear (left), bottle (right). The height of both the objects is about 11 cm.

with a resolution of 1280×960 pixels has been acquired for each object considered in the experiments. With this number of images, it is possible to achieve a good accuracy in the surface reconstruction of objects with complex shapes. For simpler objects, a lower number of images can be considered.

The reconstruction surface is sampled with $n_s = 1500$ points, while the surface parameters have been chosen as: $k = 1$ N/m, $b = 1$ Ns/m, $F_a = 0.1$ N and $\epsilon = 0.9$. The parameters k_Π, k_{c_m}, k_e, k_a of the grasp planner have been chosen all equal to 1, in order to have the same weight for all the contributions to the grasp quality measure, while the threshold ϵ_f has been tuned to a value of 0.002 N. The floating security distance δ_f has been set to 2 cm, which is intentionally large to achieve a better visualization of the trajectories. The computational time for the whole process is about 1.5 s on a Pentium 1.7 GHz. In particular, the stage for the object model reconstruction employs only 80 ms to reach the final equilibrium, while the grasp planning stage is the slower one. Of course, the execution time for grasp planning depends on the choice of the quality indices.

B. Results

In the remainder, the results of the experiments performed with the objects shown in Fig. 5, namely a teddy-bear and a little bottle, are presented.

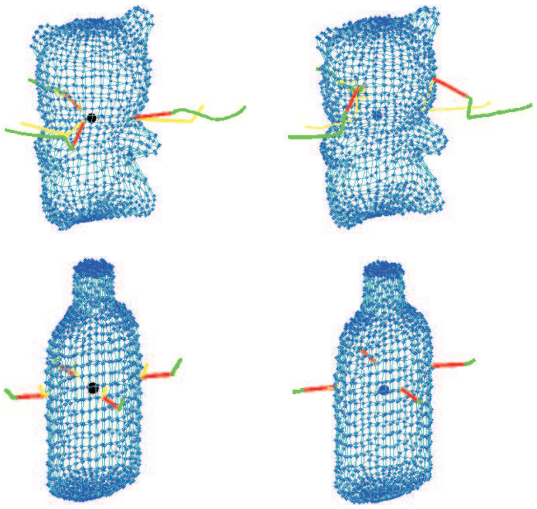

Fig. 11. Trajectories of the fingers (green: approach with safety distance δ_f, red: grasp) and corresponding sequence of points computed by the grasp planner (yellow) for the two objects, evaluated with $k_{c_m} = 1$ (left) and $k_{c_m} = 0$ (right).

In Fig. 10 some intermediate steps of the reconstruction algorithm are shown, while the finger trajectories and the final grasp configurations, respectively for the teddy-bear (using three fingers) and for the little bottle (using four fingers), are presented in Fig. 11. Both the cases $k_{c_m} = 1$ (left) and $k_{c_m} = 0$ (right) are considered (the bold points represent the positions of the objects' center of mass). In particular, in the case $k_{c_m} = 1$, it is evident that, for both objects, the grasp plane contains the center of mass; on the other hand, for $k_{c_m} = 0$, the grasp plane is far from the center of mass while the grasp polygon's area, for both objects, is maximum.

Notice that the teddy-bear is grasped with the three fingers in a planar equilateral grasp (120^o apart) for both cases $k_{c_m} = 1$ and $k_{c_m} = 0$. The yellow lines represent the sequence of points computed by the grasp planner. The green lines represent the trajectories that the planner generates for the fingertips after spatial filtering and considering the safety distance. Finally, the red lines show the last part of the grasp trajectory, when the safety distance is progressively reduced.

For the case of the little bottle, the final grasp configuration is planar and equilateral as well, with the four contact points 90^o apart. Moreover, the finger trajectories are very regular due to the good choice of the initial grasp configuration evaluated by the hand preshaping module. This result is common when the object is symmetric w.r.t. one or more axes, and so it is well represented by an ellipsoidal surface. Of course, for the particular bottle's shape, the results do not change significantly when $k_{c_m} = 0$.

C. Comparison

To further validate the proposed method, a comparison between the results obtained with the proposed parallel approach with local optimization and those obtained using the classical serial approach with global optimization has been performed. In detail, after the whole reconstruction of the unknown object model, a global search of the optimal grasp has been

TABLE I
COMPARISON BETWEEN THE PARALLEL (LOCAL) APPROACH AND THE SERIAL (GLOBAL) APPROACH WITH DIFFERENT GRASP QUALITY MEASURES.

$k_{c_m} = 1$	Q_1		Q_2		Q_3	
	Local	*Global*	*Local*	*Global*	*Local*	*Global*
Bottle	0.016	0.016	0.091	0.101	0.438	0.438
	$d_M = 0.0$ mm		$d_M = 4.1$ mm		$d_M = 4.7$ mm	
Teddy Bear	0.292	0.320	0.536	0.690	0.426	0.482
	$d_M = 4.7$ mm		$d_M = 4.5$ mm		$d_M = 5.3$ mm	

$k_{c_m} = 0$	*Local*	*Global*	*Local*	*Global*	*Local*	*Global*
Bottle	0.019	0.020	0.099	0.112	0.574	0.590
	$d_M = 5.9$ mm		$d_M = 4.0$ mm		$d_M = 7.8$ mm	
Teddy Bear	0.267	0.297	0.369	0.413	0.707	0.780
	$d_M = 4.5$ mm		$d_M = 4.5$ mm		$d_M = 3.5$ mm	

done according to three well-known quality measures, namely: (Q_1) the max-min singular value of the grasp matrix [14], (Q_2) the maximum volume of the ellipsoid in the wrench space [14], and (Q_3) the largest perturbation wrench that the grasp can resist [13]. Where requested, frictionless contact has been assumed. Two further constraints have been included in the global search: the grasp configurations violating hand physical constraints and those with the center of grip far from the object's center of mass have been discarded. This latter constraint has been considered only in the case $k_{c_m} = 1$.

For each index Q_i, the value computed in the globally optimal grasp configuration has been compared with that computed in the locally optimal grasp configuration obtained using the proposed parallel approach.

The results of this comparison are shown in Table I, where it is evident that the values of the indices are very similar in the case of the bottle, while there are small differences in the case of the teddy-bear. Moreover, the final contact points in all cases are very close to each other, as one can observe from the value d_M in the same table. This parameter has been defined by computing the distance between each contact point of the final grasp configuration reached in the parallel approach and the closest point of the final configuration reached in the serial approach: d_M is the maximum among these values. Hence, d_M can be considered as a measure of the distance between grasping configurations. It can be observed that this quantity remains quite small in all the situations, considering that the maximum dimension of both the objects is about 11 cm.

Finally, it is worth noticing that the time employed to run the serial method is, on average, ten times slower than the time employed by the parallel method.

VIII. CONCLUSION

A new method for online grasp planning of unknown objects for a multi-fingered robotic hand was presented in this paper. The proposed approach is composed of an iterative object's surface reconstruction algorithm and of a local optimal grasp planner, evolving in a synchronized parallel way. The reconstruction algorithm is based on vision and employs a virtual elastic reconstruction surface shrinking towards the object to grasp and, at the equilibrium, assuming the geometrical shape of the object. The grasp planner moves the fingertips of the robotic hand on the current available reconstruction surface towards points which are optimal (in a local sense) w.r.t. a certain number of indices weighting both the quality of the grasp and the kinematic configuration of the robotic hand. The total computational time is given by the slower between the reconstruction and the planning stage, while the grasp execution can start in parallel with these processes. This feature makes the proposed approach suitable for applications where fast or online grasp planning is required. For the planning stage, a simple quality measure has been tested in a number of experiments with satisfactory results. It is worth observing that, however, different grasp quality measures can be used in the same framework.

REFERENCES

[1] B. Yoshimi and P. Allen, "Visual control of grasping and manipulation tasks," in *IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, Las Vegas, 1994.

[2] Y. Wang, H. Lang, and C. de Silva, "A hybrid visual servo controller for robust grasping by wheeled mobile robots," *IEEE/ASME Transactions on Mechatronics*, vol. 15, no. 5, pp. 757–769, 2010.

[3] F. Bley, V. Schmirgel, and K. Kraiss, "Mobile manipulation based on generic object-knowledge," in *IEEE International Symposium on Robot and Human Interactive Communication*, Hatfield, 2006.

[4] C. Dyer, "Volumetric scene reconstruction from multiple views," *Foundations of Image Understanding*, pp. 469–489, 2001.

[5] C. Xu and J. Prince, "Snakes, shapes, and gradient vector flow," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 359–369, 1998.

[6] L. Cohen, "On active contour models and balloons," *Computer Vision, Graphics, and Image Processing: Image Understanding*, vol. 53, no. 2, pp. 512–522, 2001.

[7] A. Laurentini, "How far 3d shapes can be understood from 2d silhouettes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 2, pp. 188–195, 1995.

[8] J. Franco and E. Boyer, "Exact polyhedral visual hulls," in *British Machine Vision Conference*, 2003.

[9] S. Prakoonwit and R. Benjamin, "3d surface point and wireframe reconstruction from multiview photographic images," *Image and Vision Computing*, vol. 25, 2007.

[10] Z. Hu, Q. Guan, S. Liu, and S. Y. Chen, "Robust 3d shape reconstruction from a single image based on color structured light," in *International Conference on Artificial Intelligence and Computational Intelligence*, Shanghai, 2009.

[11] M. Ribo and M. Brandner, "State of the art on vision-based structured light systems for 3d measurements," in *IEEE International Workshop on Robotic and Sensors Environments*, Ottawa, 2005.

[12] D. Perrin, C. Smith, O. Masoud, and N. Papanikolopoulos, "Unknown object grasping using statistical pressure models," in *IEEE International Conference on Robotics and Automation*, San Francisco, 2000.

[13] C. Ferrari and J. Canny, "Planning optimal grasps," in *IEEE International Conference on Robotics and Automation*, Nice, 1992.

[14] Z. Li and S. Sastry, "Task-oriented optimal grasping by multi-fingered robot hands," *IEEE Journal of Robotics and Automation*, vol. 4, no. 1, pp. 32–44, 1988.

[15] K. Shimoga, "Robot grasp synthesis algorithms: a survey," *The International Journal of Robotic Research*, vol. 15, no. 3, pp. 230–266, 1996.

[16] R. Suarez, M. Roa, and J. Cornella, "Grasp quality measures," Universitat Politecnica de Catalunya, Institut d'Organitzacio i Control de Sistemes Industrials, Technical Report IOC-DT-P-2006-10, 2006.

[17] Y. Guan and H. Zhang, "Kinematic feasibility analysis of 3d grasps," in *IEEE International Conference on Robotics and Automation*, Seoul, 2001.

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

[19] E. Chinellato, R. Fisher, A. Morales, and P. del Pobil, "Ranking planar grasp configurations for a three-fingered hand," in *IEEE International Conference on Robotics and Automation*, Taipei, 2003.

[20] T. Nguyen and H. Stephanou, "A continuous model of robot hand pre-shaping," in *IEEE International Conference on Systems, Man and Cybernetics*, Cambridge, 1989.

[21] T. Supuk, T. Kodek, and T. Bajd, "Estimation of hand preshaping during human grasping," *Medical Engineering and Physics*, vol. 27, 2005.

[22] S. Winges, D. Weber, and M. Santello, "The role of vision on hand preshaping during reach to grasp," *Experimental Brain Research*, vol. 152, no. 4, pp. 489–498, 2003.
[23] K. Aydin, "Fuzzy logic, grasp preshaping for robot hands," in *IEEE Annual Conference of the North America Fuzzy Information Processing Society*, College Park, 1995.
[24] J. Vilaplana and J. Coronado, "A neural network model for coordination of hand gesture during reach to grasp," *Neural networks*, vol. 19, no. 1, pp. 12–30, 2006.
[25] K. Huebner and D. Kragic, "Selection of robot pre-grasps using box-based shape approximation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, 2008.
[26] M. Prats, P. Sanz, and A. del Pobil, "Task-oriented grasping using hand preshapes and tasks frames," in *IEEE International Conference on Robotics and Automation*, Rome, 2007.
[27] V. Lippiello, F. Ruggiero, B. Siciliano, and L. Villani, "Preshaped visual grasp of unknown objects with a multi-fingered hand," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, 2010.
[28] O. Faugeras and R. Keriven, "Complete dense stereovision using level set methods," in *Proceedings of the 5th European Conference on Computer Vision*, Freiburg, 1998.
[29] M. Lhuillier and L. Quan, "Surface reconstruction by integrating 3d and 2d data of multiple views," in *IEEE International Conference on Computer Vision*, Nice, 2003.
[30] V. Lippiello and F. Ruggiero, "Surface model reconstruction of 3d objects from multiple views," in *IEEE International Conference on Robotics and Automation*, Kobe, 2009.
[31] C. Dune, E. Marchand, C. Collowet, and C. Leroux, "Active rough shape estimation of unknown objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Nice, 2008.
[32] M. Cutkosky, "On grasp choice, grasp models, and the design of hands for manufacturing tasks," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 269–279, 1989.
[33] I. Biederman, "Recognition-by-components: a theory of human image understanding," *Psychological Review*, 1987.
[34] B. Mirtich and J. Canny, "Easily computable optimum grasps in 2-d and 3-d," in *IEEE International Conference on Robotics and Automation*, San Diego, 1994.
[35] D. Ding, Y. Liu, and S. Wang, "Computation of 3-d form-closure grasps," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 4, pp. 512–522, 2001.
[36] J. Ponce and B. Faverjon, "On computing three-finger force-closure grasps of polygonal objects," *IEEE Transactions on Robotics and Automation*, vol. 11, no. 6, pp. 868–881, 1995.
[37] R. F. E. Chinellato, A. Morales and A. del Pobil, "Visual quality measures for characterizing planar robot grasps," *IEEE/ASME Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol. 35, no. 1, pp. 30–41, 2005.
[38] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani, "Six-dof impedance control of dual-arm cooperative manipulators," *IEEE/ASME Transactions on Mechatronics*, vol. 13, no. 5, pp. 576–586, 2008.
[39] G. Liu and Z. Li, "Real-time grasping-force optimization for multifingered manipulation: theory and experiments," *IEEE/ASME Transactions on Mechatronics*, vol. 9, no. 1, pp. 65–77, 2004.
[40] N. Wettels, A. Parnandi, J.-H. Moon, G. Loeb, and G. Sukhatme, "Grip control using biomimetic tactile sensing systems," *IEEE/ASME Transactions on Mechatronics*, vol. 1, no. 6, pp. 718 –723, 2009.