4

Grasping Points Determination Using Visual Features

Madjid Boudaba[1] , Alicia Casals[2] and Heinz Woern[3]
[1] *Design Center, TES Electronic Solutions GmbH, Stuttgart*
[2] *GRINS: Research Group On Intelligent Robots and Systems,Technical University of Catalonia, Barcelona*
[3] *Institute of Process Control and Robotics (IPR), University of Karlsruhe*
[1,3]Germany, [2]Spain

1. Introduction

This paper discusses some issues for generating point of contact using visual features. To address these issues, the paper is divided into two sections: visual features extraction and grasp planning. In order to provide a suitable description of object contour, a method for grouping visual features is proposed. A very important aspect of this method is the way knowledge about grasping regions are represented in the extraction process, which is used also as filtering process to exclude all undesirable grasping point (unstable points) and all line segments that do not fit to the fingertip position. Fingertips are modelled as point contact with friction using the theory of polyhedral convex cones. Our approach uses three-finger contact for grasping planar objects. Each set of three candidate of grasping points is formu- lated as linear constraints and solved using linear programming solvers. Finally, we briefly describe some experiments on a humanoid robot with a stereo camera head and an anthropomorphic robot hand within the "Centre of excellence on Humanoid Robots: Learning and co-operating Systems" at the University of Karlsruhe and Forchungszentrum Karlsruhe.

2. Related work

Grasping by multi-fingered robot hands has been an active research area in the last years. Several important studies including grasp planning, manipulation and stability analysis have been done. Most of these researches assume that the geometry of the object to be grasped is known, the fingertip touches the object in a point contact without rolling, and the position of the contact points are estimated based on the geometrical constraints of the 2 Madjid Boudaba, Alicia Casals and Heinz Woern grasping system. These assumptions reduce the complexity of the mathematical model of the grasp (see [Park and Starr, 1992], [Ferrari and Canny, 1992], [Ponce and Faverjon, 1995], [Bicchi and Kumar, 2000], [J. W. Li and Liu, 2003]). A few work, however has been done in integrating vision-sensors for grasping and manipulation tasks. To place our approach in perspective, we review existence methods for sensor based grasp planning. The existing literature can be broadly classified in two categories; vision based and tactile based. For both categories, the extracted image

Source: Vision Systems: Applications, ISBN 978-3-902613-01-1
Edited by: Goro Obinata and Ashish Dutta, pp. 608, I-Tech, Vienna, Austria, June 2007

features are of concern which are vary from geometric primitives such as edges, lines, vertices, and circles to optical flow estimates. The first category uses visual features to estimate the robot's motion with respect to the object pose [Maekawa et al., 1995], [Smith and Papanikolopoulos, 1996], [Allen et al., 1999]. Once the robot hands is already aligned with object, then it needs only to know where the fingers are placed on the object. The second category of sensor uses tactile features to estimate the touch sensing area that in contact with the object [Berger and Khosla, 1991], [Chen et al., 1995], [Lee and Nicholls, 1999]. A practical drawback is that the grasp execution is hardly reactive to sensing errors such as finger positioning errors. A vision sensor, meanwhile, is unable to handle occlusions. Since an object is grasped according to its CAD model [Koller et al., 1993], [Wunsch et al., 1997], [Sanz et al., 1998], [N. Giordana and Spindler, 2000], [Kragic et al., 2001], an image also contains redundant information that could become a source of errors and ineffciency in the processing.

This paper is an extension of our previous works [Boudaba and Casals, 2005], [Boudaba et al., 2005], and [Boudaba and Casals, 2006] on grasp planning using visual features. In this work, we demonstrate its utility in the context of grasp (or fingers) positioning. Consider the problem of selecting and executing a grasp. In most tasks, one can expect various uncertainties. To grasp an object implies building a relationship between the robot hand and object model. The latter is often unavailable or poorly known. So selecting a grasp position from such model can be unprecise or unpracticable in real time applications. In our approach, we avoid to use any object model and instead it works directly from image features. In order to avoid fingers positioning error, a set of grasping regions is defined that represents the features of grasping contact point. This not only avoids detection/localization errors but also saves computations that could affect the reliability of the system. Our approach can play the critical role of forcing the fingers to a desired positions before the task of grasping is executed.

The proposed work can be highlighted in two major phases:

1. **Visual information phase**: In this phase, a set of visual features such as object size, center of mass, main axis for orientation, and object's boundary are extracted. For the purpose of grasping region determination, extracting straight segments are of concern using the basic results from contour based shape representation techniques. We will focus on the class techniques that attempt to represent object's contour into a model graph, which preserves the topological relationships between features.
2. **Grasp planning phase**: The grasping points are generated in the planning task taking as input these visual features extracted from the first phase. So a relationship between visual features and grasp planning is proposed. Then a set of geometrical functions is analysed to find a feasible solution for grasping. The result of grasp planning is a database contains a list of:
   - Valid grasps. all grasps that fulfill the condition of grasp.
   - Best Grasps. a criterion for measuring a grasp quality is used to evaluate the best grasps from a list of valid grasps.
   - Reject grasps. those grasps that do not fulfill the condition of grasp.

The remainder of this chapter is organized as follows: Section 3 gives some background for grasping in this direction. The friction cone modeling and condition of force-closure grasps are discussed. In section 4, a vision system framework is presented. The vision system is divided into two parts: the first part concerning to 2D grasping and the second part

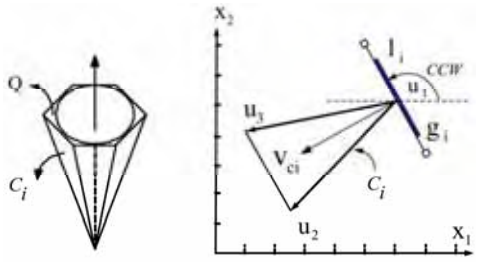
concerning 3D grasping. we first discuss the extracted visual information we have integrated in grasp planning, generation of grasping regions by using curves fitting and merging techniques, and discuss the method of selecting valid grasps using the condition of force-closure grasp. We then discuss the algorithm for computing feasible solutions for grasping in section 5. We verify our algorithm by presenting experimental results of 2D object grasping with three-fingers. Finally, we discuss the result of our approach, and future work in section 6.

3. Grasp Background

Our discussion is based on [Hirai, 2002]. Given a grasp which is characterized by a set of contact points and the associated contact models, determine if the grasp has a force-closure. For point contact, a commonly used model is point contact with friction (PCWF). In this model, fingers can exert any force pointing into friction cone at the edge of contacts (We use edge contact instead of point contact and can be described as the convex sum of proper point contacts). To fully analyze the grasp feasibility, we need to examine the full space of forces acting on the object. Forming the convex hull of this space is diffcult due to the nonlinear friction cone constraints imposed by the contact models. In this section, we only focus in precision grasps, where only the fingertips are in contact with the object. After discussing the friction cone modeling, a formalizme is used for analysing the force closure grasps using the theory of polyhedral convex cones.

3.1 Modeling the Point of Contact

A point of contact with friction (sometimes referred to as a hard-finger) im- poses non linear constraints on the force inside of its friction cones. For the analysis of the contact forces in planar grasps, we simplify the problem by modeling the friction cones as a convex polytopes using the theory of polyhedral convex cones attributed to [Goldman and Tucker, 1956]. In order to construct the convex polytope from the primitive contact forces, the following theorem states that a polyhedral convex cone (PCC) can be generated by a set of basic directional vectors.



(a) (b)

Figure 1. Point Contact Modelling

Theorem 1. A convex cone is a polyhedral if and only if it is finitely generated, that is, the cone is generated by a finite number of vectors $v_1, v_2, \cdots, v_m$:

$$\mathbf{C} = \left\{ \mathbf{u_i} \in R^n : \sum_{i=1}^{m} \alpha_i \mathbf{u_i}, \alpha_i \geq 0 \right\} \tag{1}$$

where the coeffcients α_i are all non negative. Since vectors u_i through u_m span the cone, we write 1 simply by $\boldsymbol{C}$=*span* $\{u_1, u_2, ..., u_m\}$. The cone spanned by a set of vectors is the set of all nonnegative linear combinations of its vectors. A proof of this theorem can be found in [Goldman and Tucker, 1956].
Given a polyhedral convex set $\boldsymbol{C}$, let *vert*($\boldsymbol{P}$)=$\{u_1, u_2, ..., u_m\}$ stand for vertices of a polytope $\boldsymbol{P}$, while *face*($\boldsymbol{P}$)=$\{\boldsymbol{F}_1, ..., \boldsymbol{F}_M\}$ denotes its faces. In the plane, a cone has the appearance as shown in Figure 1(b). This means that we can reduce the number of cone sides, m = 6 to one face, $\boldsymbol{C}_i$. Let's denote by $\boldsymbol{P}$, the convex polytopes of a modelled cone, and $\{u_1, u_2, u_3\}$ its three vertices. We can define such polytope as

$$\mathbf{P} = \left\{ \mathbf{x} \in R^n | x = \sum_{i=1}^{u_p} \delta_i \mathbf{u_i} : 0 \leq \delta_i \leq 1, \sum_{i=1}^{u_p} \delta_i = 1 \right\} \tag{2}$$

where u_i denotes the i-th vertex of $\boldsymbol{P}$, and up is the total number of vertices.
n=2 in the case of a 2D plane.

3.2 Force-Closure Grasps

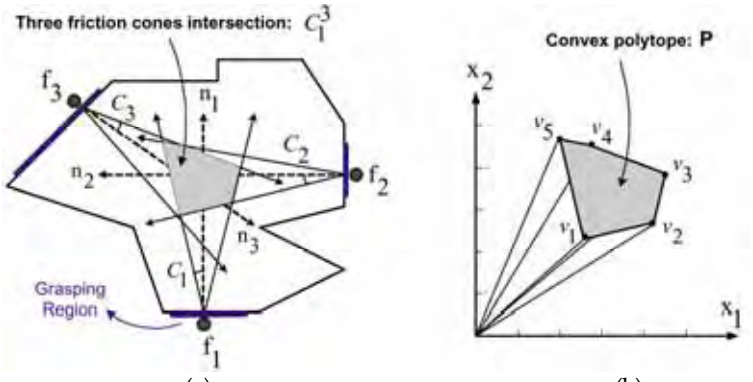
The force-closure of a grasp is evaluated by analysing its convex cone. For a set of friction cone intersection, the full space can be defined by

$$\mathbf{C_1^k} = C(\mathbf{P_1}) \cap C(\mathbf{P_2}) \cap \cdots \cap C(\mathbf{P_k}) \tag{3}$$

where k is the number of grasping contacts. Note that the result of C_1^k is a set of polytopes intersections and produces either an empty set or a bounded convex polytopes. Therefore, the solution of (3) can be expressed in terms of its extreme vertices

$$\Omega_1^{v_p}(U) = \left\{ \sum_{i=1}^{v_p} \alpha_i u_{ci}, \quad \sum_{i=1}^{v_p} \alpha_i = 1, \quad \alpha_i \geq 0 \right\} \tag{4}$$

where v_p is the total number of extreme vertices.



(a) (b)

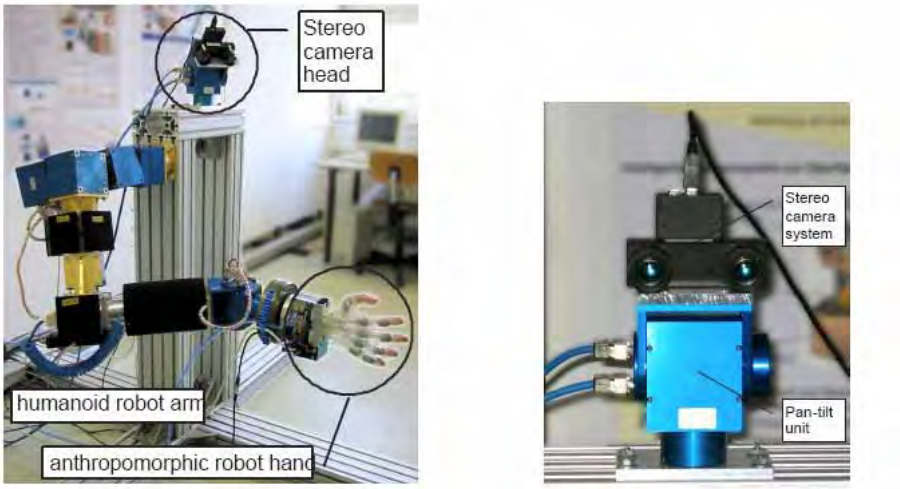
Figure 2. Feasible solution of a three-fingered grasp

Figure 2 illustrates an example of feasible solution of $\Omega_1^{v_p}(U)$ and its grasp space represented by its extreme vertices $\boldsymbol{P}$ = $\{v_1, v_2, \cdots, v_m\}$. From this figure, two observations can be

suggested: first, if the location of a fingertip is not a solution to the grasp, it is possible to move along its grasping region. Such displacement is defined by $u_i = u_{i0} + \beta_i t_i$ where β_i is constrained by $0 \le \beta_i \le l_i$ and u_i be a pointed vertex of $\boldsymbol{C}_i$. Second, we define a ray passing through the pointed vertex u_i, by a function $v_{ci}^T X \ (i=1,\ldots,k)$. The vector $v_{ci}=[v_{cix}, v_{ciy}] \in \boldsymbol{R}^2$ varies from the lower to the upper side of the spanned cone $\boldsymbol{C}_i$. This allows us to check whether the feasible solution remains for all v_{ci} in the cone spanned by u_2 and u_3 (see Figure 1(b)).
Testing the force-closure of a grasp now becomes the problem of finding the solutions to (4). In other words, finding the parameters of (3) that the (4) is a bounded convex polytopes.

4. System Description

We are currently developing a robotic system that can operate autonomously in an unknown environment. In this case, the main objective is the capability of the system to (1) locate and measure objects, (2) plan its own actions, and (3) self adaptable grasping execution. The architecture of the whole system is organized into several modules, which are embedded in a distributed object communication framework. There are mainly three modules which are concerned in this development: the extraction of visual information and its interpretation, grasp planning using the robot hand, the control and execution of grasps..



(a) Experimental setup (b) Stereo vision head

Figure 3. Robotic system framework. (a) An humanoid robot arm (7DOF) and an antropomorphic robot hand (10DOF). (b) Stereo vision system

4.1 The Robot Hand

The prototype of the anthropomorphic robot hands (see [Schulz et al. 2001]) has a 7 degrees of freedom (DOF) arm (see Fig. 3(a)). This first prototype is currently driven pneumatically and is able to control the 10 DOF separately, but the joints can only be fully opened or closed. The robot's task involve controlling the hand for collision-free grasping and manipulation of objects in the three dimensional space. The system is guided solely by visual information extracted by the vision system.

4.2 The Vision System

The vision system shown in Fig. 3(b) consists of a stereo camera (MEGA-D from Videre Design) mounted on pan-tilt heads equipped with a pair of 4.8 mm lenses and has a fixed baseline of about 9 cm. The pan-tilt head provides two additional degrees of freedom for the cameras, both of them rotational. The MEGA-D stereo head uses a IEEE 1394 firewire interface to connect to a workstation and has a SRI's Small Vision System (SVS) software for calibration and stereo correlation (see [Konolige, 1997]).

For its complexity, the flow diagram of visual information has been divided into two parts. The first part provides details of 2D visual features extraction. The second part is dedicated to 3D visual features retrieval. The image acquisition primarily aims at the conversion of visual information to electrical signals, suitable for computer interfacing. Then, the incoming image is subjected to processing having in mind two purposes: (1) removal of image noise via low-pass filtering by using Gaussian filters due to its computational simplicity and (2) extraction of prominent edges via high-pass filtering by using the Sobel operator. This information is finally used to group pixels into lines, or any other edge primitive (circles, contours, etc). This is the basis of the extensively used Canny's algorithm [Canny, 1986]. So, the basic step is to identify the main pixels that may preserve the object shape. As we are visually determining grasping points, the following sections provide some details of what we need for our approach.

Contour Based Shape Representation

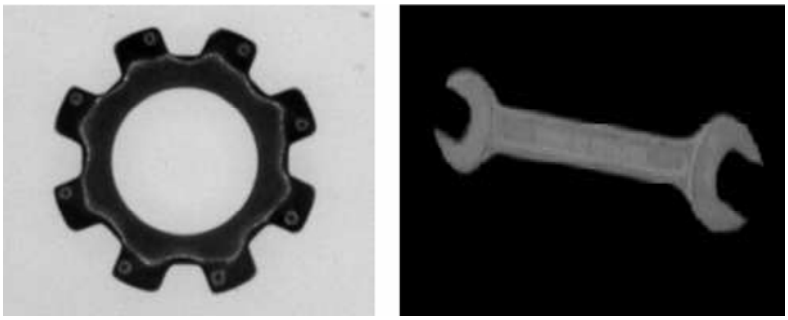
Due to their semantically rich nature, contours are one of the most commonly used shape descriptors, and various methods for representing the contours of 2D objects have been proposed in the literature [Costa and Cesar, 2001]. Extracting meaningful features from digital curves, finding lines or segments in an image is highly significant in grasping application. Most of the available methods are variations of the dominant point detection algorithms [M. Marji, 2003]. The advantage of using dominant points is that both, high data compression and feature extraction can be achieve. Other works prefer the method of polygonal approximation using linking and merging algorithms [Rosin, 1997] and curvature scale space (CSS) [Mokhtarian and Mackworth, 1986].

A function regrouping parameters of visual features together can be defined by

$$B = \{vlist, slist, llist, com\} \tag{5}$$

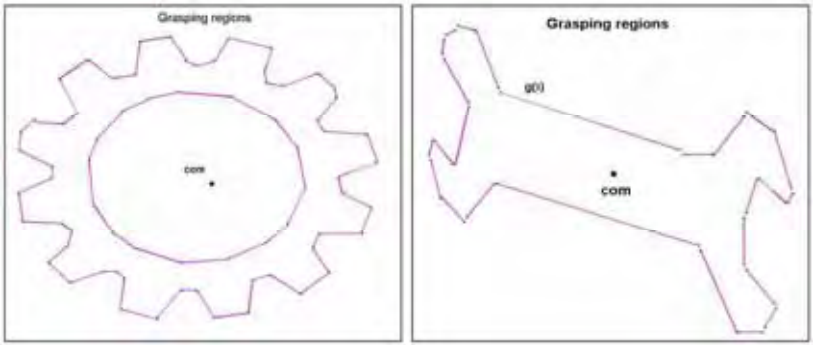
where $vlist = \{v_1, v_2, \cdots, v_m\}$ is a list of consecutive contour's vertices with $v_i = (x_i, y_i)$ that represents the location of v_i relative to the center of mass of the object, $com = (x_c, y_c)$. $slist = \{s_1, s_2, \cdots, s_m\}$ is a list of consecutive contour's segments. Both lists $vlist$ and $slist$ are labelled counter-clockwise (ccw) order about the center of mass. During the processing, the boundary of the object, B is maintained as a doubly linked list of vertices and intervening segments as $v_1 s_1 v_2, \cdots, v_m s_m v_1$. The first segment s_1, connecting vertices v_1 and v_2, the last segment s_m, connecting vertices v_m and v_1. A vertex v_i is called reflex if the internal angle at v_i is greater than 180 degrees, and convex otherwise. $llist$ is a list that contains the parameters of correspondent segments. Additional to the local features determined above, an algorithm for contour following is integrated. This algorithm follows the object's boundary from a starting point determined previously and goes counter-clockwise around the contour by ordering successively its vertices/edge points into a double linked list. The algorithm stops when the starting point is reached for the second time. The aim of this stage

is to determine that all vertices/segments belong to the object's boundary which we will need further for the determination of the grasping points position.



(a) (b)

(a) Binary object



(a) (b)

(b) Visual features extraction

Figure 4. Object shape representation. (a) Images from original industrial objects
(b) Extraction of grasping regions

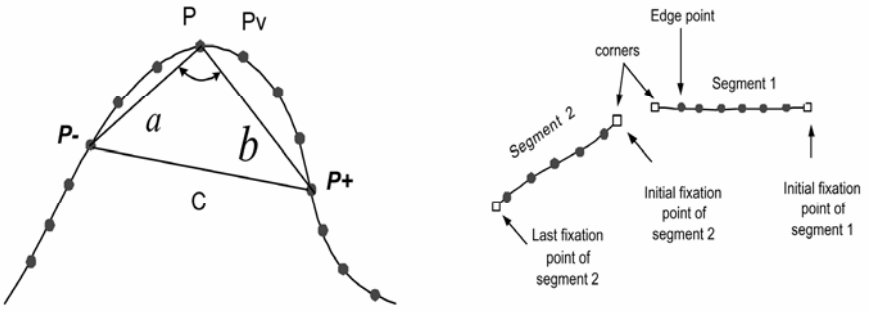
Extraction of Grasping Regions

Grasping regions are determined by grouping consecutive edge points from a binary edge image. This is usually a preliminary step before grasping takes place, and may not be as time critical as the task of grasping points determination. We deal with (5), the list $vlist=\{v_1, v_2, \cdots, v_m\}$ is the result that forms an ordered list of connected boundary vertices. We then need to store the parameters of these primitives instead of discrete points (or vertices) to fit a line segment to a set of vertices points that lie along a line segment. The aim of this step is to determine all salient segments that preserve the shape of the object contour. Figure 4(b) shows grasp regions on the object's contour. Afterwards, each grasping region is extracted as straight segment. The size of the grasping regions should be long enough for positioning the robot fingers. The curve fitting (as shown in Figure 5(a)) describes the process of finding a minimum set of curve segments to approximate the object's contour to a set of line segments with minimum distortion. Once the line segments have been approximated, the merging method (as shown in Figure 5(b)) is used to merge two lines segment that satisfied the merging threshold.

The final result of the algorithm is a list of consecutive line segments with a specified tolerance which preserve the object's contour. Briefly, merging methods, (1) use the first two vertices points to define a line segment (2) add a new vertex if it does not deviate too much from the current line segment (3) update the parameters of the line segment using least-squares measure (4) start a new line segment when edge points deviate too much from the line segment. The final result of the algorithm is a list of consecutive line segments with a specified tolerance which preserve the object's contour. We define such list by

$$slist = \{s_1, s_2, \cdots, s_m\} \tag{6}$$

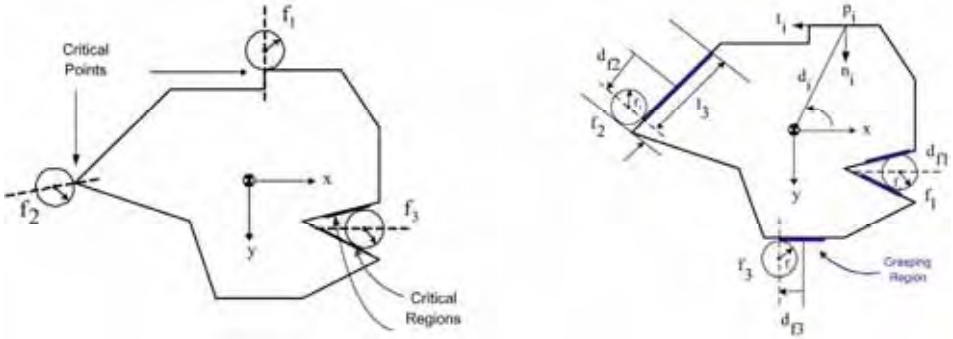
where a segment s_i is defined by its ending vertices $v_i=(x_i, y_i)$ and $v_{i+1}=(x_{i+1}, y_{i+1})$ that represent the location of a segment in the plane. m is the number of segments containing the list *slist*.



(a) Curve fitting (b) Segment merging

Figure 5. Curve fitting and merging methods. In each curve point p, a variable triangle (p^-,p,p^+) is defined. The admissible triangle is then checked by the following conditions: $d_{min} \leq |p - p^-|$, $d_{min} \leq |p - p^+|$, $\alpha \leq \alpha_{max}$, where $|p - p^-| = a$, $|p - p^+| = b$, and $\alpha = arccos\,(a^2 + b^2 - c^2)/2ab$ is the opening angle of the triangle

Critical Grasping Points



(a) Critical points (b) Finger displacement

Figure 6. (a) Critical grasping point. Possible displacement d_{fi} of a fingertip f_i on its corresponding grasping region (thicker region): Fingertip f_1 is placed at midpoint of its corresponding grasp region, $d_{f1}=0$. Fingertip f_2 is displaced at d_{f2} in positive direction from midpoint, and the fingertip f_3 is displaced at d_{f3} in negative direction from midpoint. We attach a left-handed frame (n_i, t_i) to each finger position p_i with a distance d_i to the center of mass. n_i and t_i) are normal and tangential direction of a finger f_i in the plane

To assure the robustness of contact placement, we make some assumptions in (6): **First assumption**, to avoid undesirable contacts at convex vertices and convex corners (see the position of finger f_1, f_2 in Figure 6(a)) which are not generally robust due to small uncertainty during the grasping phase. We also avoid the concave vertices having a size of concavity smaller than the size of the fingertip using the reachability conditions (see the position of finger f_3 in Figure 6(a)).
Second assumption, we estimate a fingertip as a sphere with radius f_r (see Figure 6). the grasping regions must be large enough for positioning the fingertip on it. Hence, a preprocessing (or prefiltering) is necessary in (6) to discard those segments with length less than the diameter of the sphere.
Based on both assumption, we define a small margin value at the endpoint of each segment by ε as shown in Figure 6 with $\varepsilon = f_r$. If a segment s_i contains all possible contact points from v_i to v_{i+1} then any grasping points must satisfy

$$\begin{cases} v_i + \varepsilon \le s_i \le v_{i+1} - \varepsilon \\ g_i = s_i - 2\varepsilon \\ g_i \ge 2f_r \end{cases} \tag{7}$$

Using the grasp criteria of (7) including the condition that the size of the grasp region must be large enough to place a finger on it, $g_i \le 2f_r$ (see Figure 6). Equation (6) becomes

$$glist = \{g_1, g_2, \cdots, g_m\} \tag{8}$$

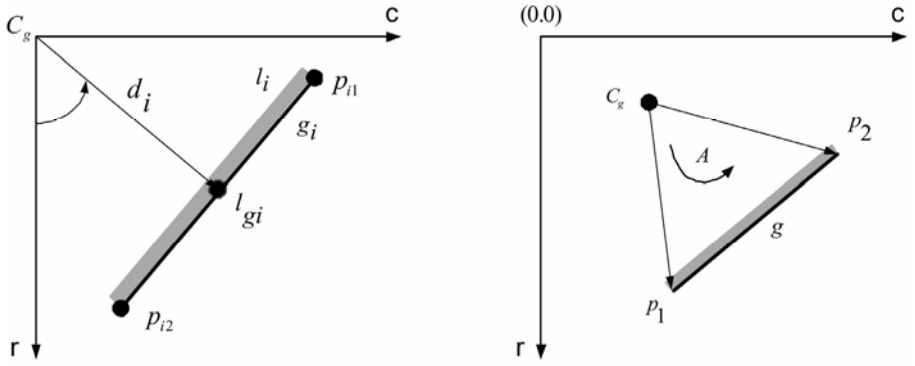
where *glist* is a linked list ordered in counterclockwise direction (see Figure 6(b)) and updated from the condition of (7).
Equation (8) is the result of a filtering test which excludes all grasping candidates that do not belong to the grasping regions and therefore reducing time consuming during grasp point generation.
Let

$$gparam_i = \{p_i, p_{i+1}, l_i, l_{gi}, d_i, \phi_i\} \tag{9}$$

be a function defining the parameter of a grasping region, g_i , where $p_i=(x_i,y_i)$ and $p_{i+1}=(x_{i+1},y_{i+1})$ represent its location in the plane, l_{gi} its length $\overline{p_i p_{i+1}}$, g_{ci} its center (midpoint), d_i is the perpendicular distance from $\overline{p_i p_{i+1}}$ to the object's center of mass, *com*. The relationship between the center of mass and grasping region is given in Figure 7(b). The sign of the area A defines the orientation of grasping region ϕ_i. The elements of (9) verify the following equations:

$$gparam_i : \begin{cases} l_i = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \\ l_{gi} = ((x_i + x_{i+1})/2, (x_i + x_{i+1})/2) \\ \phi_i = \arctan(y_{i+1} - y_i)/(x_{i+1} - x_i) \\ d_i = g_i / l_i, \quad g_i = a_i x + b_i y + c_i. \end{cases}$$

The last equation g_i is a linear equality constraint of a given grasping region in the plane (see Figure 7). An additional criteria should be added to avoid that two or three fingers are placed on the same contact point. In this paper we only assign one finger to each grasping region.



(a) (b)

Figure 7. Projection of grasping region g_i in the image plane (c, r)

5. Grasp Planning Algorithm

Grasp planning can be seen as constructing procedures for placing point contacts on the surface of a given object to achieve force-closure grasps. Taking as input the set of visual features extracted from the contour of the object, the output is a set of valid grasps. The relationship between visual features and grasp planning is given in next section.

5.1 Grasp Point Generation

Generating a number of valid grasps from a list of candidates and classifying the best among them is quite time consuming. Thus a preprocessing (or prefiltering) is necessary before the grasping points generation takes place. We first order the (8) in counterclockwise direction with a starting point from x-axis as shown in Figure 6(b). Second the initial contact of fingertips on grasping regions would be at the midpoint, which are considered as robust contacts and measured directly from the center of mass of the object. Then the displacement d_{fi} (see Figure 6(a)) of the fingertip on its corresponding grasping region (if necessary) should be first in the counterclockwise then in the clockwise direction.

The following equation describes the relationship between the visual features and grasp planning

$$G = f(glist, gparam, com) \tag{10}$$

where *glist*, *gparam* and *com* are the visual features observed on the image plane and G is a grasp map of outputs defined by the relationship between fingers and the location of contact points on its corresponding grasping regions. From the grasp map G three possible solutions are derived:

$$G: \begin{cases} G_s = \{G_{s_1}, G_{s_2}, \cdots, G_{s_{is}}\} \\ G_b = \{G_{b_1}, G_{b_2}, \cdots, G_{b_{ib}}\} \\ G_r = \{G_{r_1}, G_{r_2}, \cdots, G_{r_{ir}}\} \end{cases} \tag{11}$$

where G_s, G_b, and G_r are selected, best, and rejected grasp, respectively. The *is*, *ib*, and *ir* are the number of selected, best, and rejected grasps, respectively.

For a three-finger grasps, the selected grasps (G_s) is given in the following form:

$$G_s : \begin{cases} G_{s_1} = \{(f_1, g_1), (f_2, g_6), (f_3, g_9)\} \\ G_{s_2} = \{(f_1, g_2), (f_2, g_6), (f_3, g_{10})\} \\ \vdots \\ G_{s_{is}} = \{(f_1, g_1), (f_2, g_8), (f_3, g_{12})\} \end{cases}$$

A similar form can be given for representing the best grasps G_b and those rejected G_r.

5.2 The Algorithm

The algorithm is divided into three parts: Visual features part which are regrouped in (5) and (8); grasp planning part which is defined by (10) and (11); and Testing part that corresponds to (4). In the visual features part, the compact representation of the object's contour is obtained which includes the grasping regions and local parameters by using the standard image processing library. In the grasp planning, a relationship between visual features and the location of the contact points is obtained for selecting a valid grasp. In the testing part, the force-closure condition is based on determining the feasible solution of a grasps. We first model the friction cone as a convex polytopes. Then, we solve the problem of (3) and (4) for a given location of contact grasp using programming solvers as well as for computing the polytope convex cones, extreme vertices of polytopes, and calculating projections. One of the advantages of the proposed algorithm is that it does not require a geometrical model of the object and can grasp unknown objects.

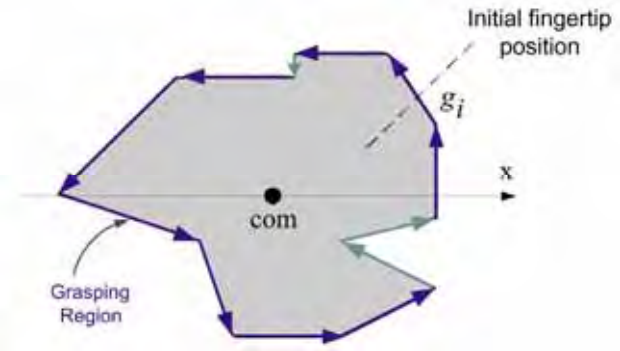


Figure 8. Grasp point generation. The fingertip range defined here is the range of its corresponding grasping regions and midpoint is its optimal contact positions, called initial pose

The whole algorithm is divided into several procedures and operates as follows:

1. *Visual features procedure*
   - *Function grouping visual features using (5)*
2. *Grasping point generation procedure*
   - *Pick three grasp regions from (8)*
   - *Determine the initial position of f_1, f_2 and f_3*
   - *Compute their friction cones using (2)*
   - *Compute the friction cones intersection of (3)*

3. *Grasping test procedure*
   - *Compute the feasible grasps using (4)*
   - *Check whether the polytopes given by (4) is bounded. If so, stop and save the selected grasps to G_s.*
   - *Else save the rejected grasps to G_r.*
4. *Quality test procedure*
   - *The last step of the algorithm consists of selecting the best grasps from a range of valid grasps from lower to upper acceptance measures by using the parameters measure given in table 1. Save to G_b.*

5.3 Implementation

We have implemented the visual features extraction and grasp planning algorithms in Matlab environment for computing feasible solution of a three-fingered grasp. We have experimented with two different kind of objects; a 3D object and a planar object. For both objects, the images extraction are saved in two jpeg files with a resolution of 320x240 and 160x220 pixels, respectively. Table 1 resumes the results of grasp planning algorithm. Three and four feasible grasp configurations have been selected from a total of 25 and 24 grasping regions generated on the object's boundary *obj*1 and *obj*2, respectively. d_1, d_2 and d_3 are distance measures of finger position f_1, f_2 and f_3 from the object's center of mass. x_1, x_2 are the coordinates of the focus point F in the plane. d is the measured distance between focus point and center of mass. R is the vector radius of the ball centered at F. The object's center of mass is located at *com* = 121.00098.000 and *com* = 115.00075.000, respectively. The angle of friction cone is fixed to α = 8.5 degrees for all grasp configurations. Figure 9 illustrates the grasp planing for object *obj*1. Three fingers are in contact with the object which is viewed from the top by the stereo vision head placed above the table. For the second object (*obj*2), the visual features are extracted from a single camera. The friction cone modelling and linear constraints programming have been implemented using [M. Kvasnica, 2005]. We further developed auxiliary function to compute various data such as extraction of visual features of the object, extraction of grasping regions, friction cone modelling, and grasp configurations.

6. Conclusions and Future Work

We have introduced an approach that combines vision and grasping. Based on the vision, visually determining grasping points is done by transforming the grasping regions into a geometrical optimization problem. The results shown in Figure 6 are obtained from applying the software packages in [20] to our Matlab 6.12 programming environment. In order to compute the feasible region of various grasps, we have integrated other linear programming solvers by providing a set of constraints for optimization procedure. Various grasps with three hard-fingers are tested on 2D original object and the feasible solution of grasps are determined by analysing the polytope region of grasps. The focus point inside the polytope convex and its distance from the object's center of mass are two measures used for selecting the best grasps.The most important aspects of our algorithm are how to select the grasping point set and to determine each one step of the grasping process. Three functions, pick(), insert(), and remove() are used. The initialization step picks a first grasping set. The while loop iterates by checking the feasible region of grasps and then by selecting a new candidate of grasp. A build library is used to store valid grasps by the insertion function

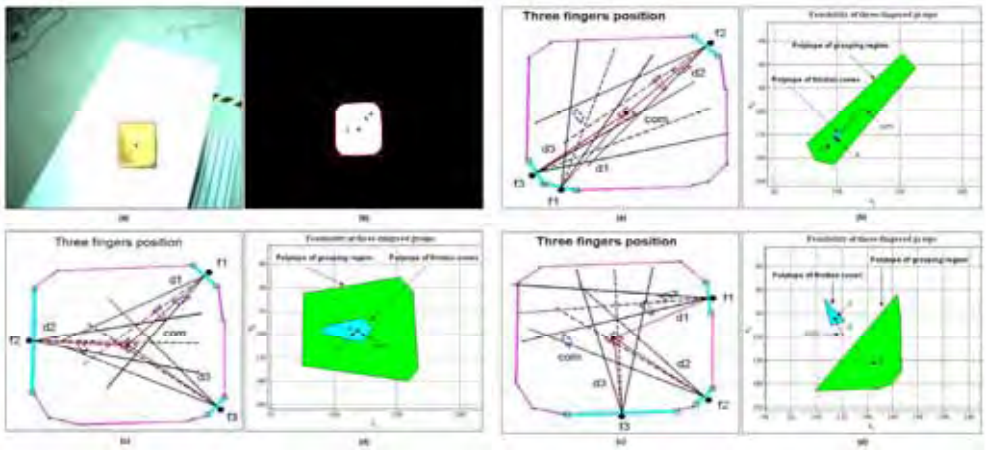
which inserts a valid candidate grasp into library, while the remove function deletes invalid grasp from the library. The results in this paper shows the potential to combine vision and grasping in a unified way to resemble the dexterity of human manipulation.
The second part of our visual processing: General flow diagram will be the future work for generating 3D grasps on unknown objects includes implementation on a humanoid robot with a stereo camera head and an anthropomorphic robot hand (as shown in Figure 3).

7. Acknowledgments

The authors would like to thank Prof. Dr. H. Woern and his co-workers from the IPR institute for their support in providing the facilities and the anthropomorphic robot hand for testing the proposed approach.

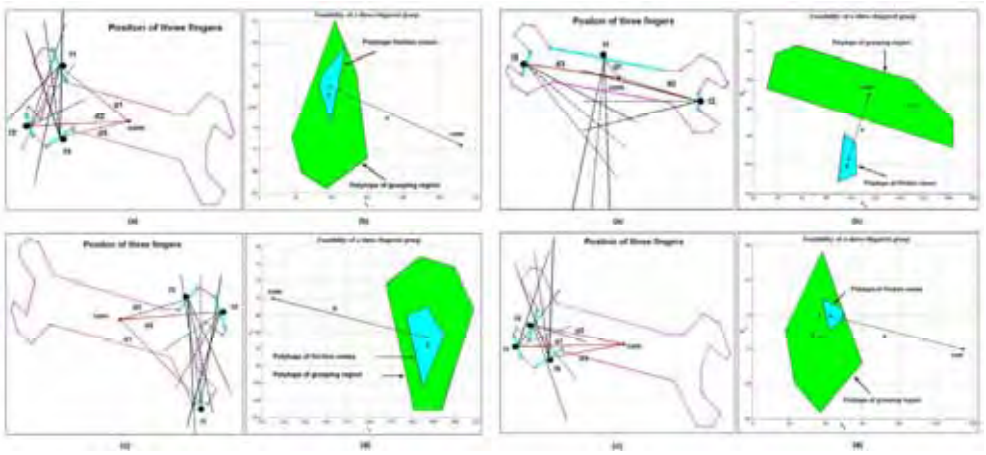
$\alpha = 8.5$ degrees for all configurations.					
obj	param.	d_1 d_2 d_3	$F(x_1,x_2)$	d	R
obj.1	GC.1	86.80 33.82 65.99	118.41 96.69	2.98	9.37
	GC.2	24.47 86.80 23.82	99.19 122.88	32.53	2.44
	GC.2	81.51 65.99 35.52	114.59 84.46	15.39	4.49
obj.2	GC.1	68.007 91.522 63.832	38.832 60.168	74.464	4.369
	GC.2	79.657 89.550 61.522	186.896 86.615	72.828	6.638
	GC.3	16.651 70.406 80.897	98.181 98.191	28.648	6.071
	GC.4	94.505 80.897 66.727	42.567 70.094	72.599	1.858

Table 1. Parameter measures of three fingered grasp configuration



(a) Grasp configuration (**GC**): 1 (b) Grasp configuration (**GC**): 2-3

Figure 9. (a) Grasp planning setup. (b) Result of three fingered grasp configuration 1*to*3 for object *obj*1



(a) Grasp configuration (**GC**): 1-2 (b) Grasp configuration (**GC**): 3-4

Figure 10. (a) Result of three fingered grasp configuration: 1–4 for object *obj*2

8. References

Allen, P., Miller, A., Oh, P., and Leibowitz, B. (1999). Integration vision, force and tactile sensing for grasping. Int. Journal of Intell. *Mechatronics*, 4(1):129–149. [Allen et al., 1999]

Berger, A. D. and Khosla, P. K. (1991). Using tactile data for real-time feedback. *Int. Journal of Robot. Res*. (IJR'91), 2(10):88–102. [Berger and Khosla, 1991]

Bicchi, A. and Kumar, V. (2000). Robotic grasping and contacts: A review. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 348–353. [Bicchi and Kumar, 2000]

Boudaba, M. and Casals, A. (2005). Polyhedral convex cones for computing feasible grasping regions from vision. In *Proc. IEEE Symposium on Computational Intelligence in Robotics and Automation* (CIRA'05), pages 607–613, Helsinki, Finland. [Boudaba and Casals, 2005]

Boudaba, M. and Casals, A. (2006). Grasping of planar obbjects using visual perception. In *Proc. IEEE 6th International Conference on Humanoid Robots* (HUMANOIDS'06), pages 605–611, Genova, Italy. [Boudaba and Casals, 2006]

Boudaba, M., Casals, A., Osswald, D., andWoern, H. (2005). Vision-based grasping point determination on objects grasping by multifingered hands. In *Proc. IEEE 6th International Conference on Field and Service Robotics* (FRS'05), pages 261–272, Australia. [Boudaba et al., 2005]

Canny, J. (1986). A computational approach to edge detection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 8(6):679–698. [Canny, 1986]

Chen, N., Rink, R. E., and Zhang, H. (1995). Edge tracking using tactile servo. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems* (IROS'95), pages 84–99. [Chen et al., 1995]

Costa, L. and Cesar, R. (2001). *Shape Analysis and Classification Theory and Practice*. CRC Press, Florida, USA, 1st edition. [Costa and Cesar, 2001]

Ferrari, C. and Canny, J. (1992). Planning optimal grasps. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2290–2295, Nice, France. [Ferrari and Canny, 1992]

Goldman, A. J. and Tucker, A. W. (1956). Polyhedral convex cones, in linear inequalities and related systems. *Annals of Mathematics Studies*, Princeton, 38:19–40. [Goldman and Tucker, 1956]

Hirai, S. (2002). Kinematics of manipulation using the theory of polyhedral convex cones and its application to grasping and assembly operations. *Trans. of the Society of Inst. and Control Eng*., 2:10–17. [Hirai, 2002]

J. W. Li, M. H. J. and Liu, H. (2003). A new algorithm for three-finger force-closure grasp of polygonal objects. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1800–1804. [J. W. Li and Liu, 2003]

Koller, D., Danilidis, K., and Nagel, H. H. (1993). Model-based object tracking in monocular image sequences of road traffc scenes. *Int. Journal of Comp. Vision* IJCV'93, 3(10):257–281. [Koller et al., 1993]

Kragic, D., Miller, A., and Allen, P. (2001). Real-time tracking meets online grasp planning. In *Proc. IEEE International Conference on Robotics and Automatation* (ICRA'2001), pages 2460–2465, Seoul, Korea. [Kragic et al., 2001]

Lee, M. H. and Nicholls, H. R. (1999). Tactile sensing for mechatronics - a state of the art servey. *Mechatronics*, 9:1–31. [Lee and Nicholls, 1999]

M. Kvasnica, P. Grieder, M. B. F. J. C. (2005). *Multiparametric toolbox, user's guide.* [M. Kvasnica, 2005]

M. Marji, P. S. (2003). A new algorithm for dominant points detection and polygonization of digital curves. *Journal of the Pattern Recognition Society*, 36:2239–2251. [M. Marji, 2003]

Maekawa, H., Tanie, K., and Komoriya, K. (1995). Tactile sensor based manipulation of an unknown object by a multifingered hand with rolling contact. In *Proc. IEEE International Conference on Robotics and Automatation* (ICRA'95), pages 743–750. [Maekawa et al., 1995]

Mokhtarian, F. and Mackworth, A. (1986). Scale-based description and recognition of planar curves and two-dimensional shapes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 8:34–43. [Mokhtarian and Mackworth, 1986]

N. Giordana, P. Bouthemy, F. C. and Spindler, F. (2000). Two-dimensional model-based tracking of complex shapes for visual servoing tasks. M. Vincze and G. Hager, editors, Robust vision for vision-based control of motion, pages 67–77. [N. Giordana and Spindler, 2000]

Park, Y. C. and Starr, J. P. (1992). Grasp synthesis of polygonal objects using a three-fingered robot hand. *IEEE International Journal of Robotics Research*, 11(3):163–184. [Park and Starr, 1992]

Ponce, J. and Faverjon, B. (1995). On computing three-finger force-closure grasps of polygonal objects. *Proceedings of the IEEE Transactions on Robotics and Automation*, 11(6):868–881. [Ponce and Faverjon, 1995]

Rosin, P. L. (1997). Techniques for assesing polygonal approximation of curves. *IEEE Trans. on Pattern Analysis and Machine Intell*., 19:659–666. [Rosin, 1997]

Sanz, P., del Pobil, A., Iesta, J., and Recatal, G. (1998). Vision-guided grasping of unknown objects for service robots. In *Proc. IEEE International Conference on Robotics and Automatation* (ICRA'98), page 30183025, Leuven, Belgium. [Sanz et al., 1998]

Smith, C. and Papanikolopoulos (1996). Vision-guided robotic grasping: Issues and experiments. In *Proc. IEEE International Conference on Robotics and Automatation* (ICRA'96), pages 3203–3208. [Smith and Papanikolopoulos, 1996]

Wunsch, P., Winkler, S., and Hirzinger, G. (1997). Real-time pose estimation of 3d objects from camera images using neural networks. In *Proc. IEEE International Conference on Robotics and Automatation* (ICRA'97), pages 3232–3237. [Wunsch et al., 1997]