Robotica (2008) volume 26, pp. 331–344. © 2007 Cambridge University Press
doi:10.1017/S0263574707003955 Printed in the United Kingdom

Kinematic enveloping grasp planning method for robotic dexterous hands and three-dimensional objects

Shahram Salimi and Gary M. Bone*

Department of Mechanical Engineering, McMaster University, Hamilton, Ontario L8S 4L7, Canada

(Received in Final Form: September 4, 2007. First published online: November 16, 2007)

SUMMARY

Three-dimensional (3D) enveloping grasps for dexterous robotic hands possess several advantages over other types of grasps. This paper describes a new method for kinematic 3D enveloping grasp planning. A new idea for grading the 3D grasp search domain for a given object is proposed. The grading method analyzes the curvature pattern and effective diameter of the object, and grades object regions according to their suitability for grasping. A new approach is also proposed for modeling the fingers of the dexterous hand. The grasp planning method is demonstrated for a three-fingered, six degrees-of-freedom, dexterous hand and several 3D objects containing both convex and concave surface patches. Human-like high-quality grasps are generated in less than 20 s per object.

KEYWORDS: Robotic hands; Enveloping grasp; Grasp planning; Grasping quality.

1. Introduction

Grasp planning is a difficult problem in robotics that has occupied researchers since the 1980s. In general, identifying suitable contact locations, hand pose (both position and orientation), and force-exertion strategies require satisfying three main sets of constraints: (1) constraints due to limited capabilities of the gripper or the dexterous hand, (2) constraints due to object geometry and material characteristics, and (3) constraints due to the task requirements. In analyzing a grasp, it is hard to separate these constraints from each other. A successful grasp is typically accomplished by reasonably satisfying all of these constraints together, which is not always possible. Most of the grasping systems (i.e., planning software plus robotic hardware) are task based and are focused on the range and type of the objects the robot should grasp. To overcome some of these limitations, the researchers introduced dexterous robotic hands. The design of these hands is usually a compromise between the simplicity of single degrees-of-freedom (DOF) grippers and the many DOF of human hands. For an excellent review of the robotic hand design and control, see the work of Pons *et al.*[1] Two common types of grasps implemented with dexterous hands are fingertip grasps (also termed precision grasps) and enveloping grasps. Between these two types, from a dexterous manipulation perspective, a fingertip grasp is preferable. However, from a grasping perspective, an enveloping grasp is preferable for the following reasons: it is more robust to errors in positioning of the fingers on the object; it is more robust to force control errors; it relies less on friction to constrain the object; and it is compatible with simple robotic hands. The robotic hands designed by Ceccarelli *et al.*[2] and by Figliolini and Rea[3] are examples of simple and effective mechanisms designed to perform such enveloping grasps. However, enveloping grasps are particularly difficult to plan for 3D objects for the following reasons.

- Locating the palm and fingers of an N DOF hand for an enveloping grasp requires searching a $6 + N$ dimensional space.
- The physical limits such as the length of the finger phalanges, or the ranges of motion for the joints, add complex constraints to the search.
- The contact between the object surface and the surfaces of the palm and fingers is much more complex to model and analyze than a fingertip contact.

This paper is focused on the kinematic aspects of enveloping grasp planning for 3D objects with dexterous hands, and presents an algorithm for finding a feasible and human-like high-quality grasp.

2. Related Work on Enveloping Grasp Planning

In this section, we will focus on the recent enveloping grasp planning literature, with a particular emphasis on the kinematic aspects. For more extensive reviews of the grasp planning literature, see the works of Bicchi and Kumar[4] and Shimoga.[5] Based on the human enveloping grasping routine, Kaneto *et al.*[6] divided the procedure of grasping into three phases: approach, lifting, and grasping. The routine they proposed can be very helpful in estimating the relative placement of the hand and object before hand closure. However, it models the grasping as lifting a cylinder. This assumption can make the results quite inaccurate for noncylindrical objects. Hwang *et al.*[7] solved the kinematic contact problem between a robot hand and an object as a contact between a B-spline surfaced object and a finger modeled by cylinders and a half ellipsoid. They solve for all contact locations using a complex recursive numerical calculation. Their algorithm is not truly a grasp planner since it requires the location of the palm to be specified *a priori*. This algorithm could be quite useful in handling delicate

* Corresponding author. E-mail: gary@mcmaster.ca

objects, since in any step of the manipulation, an accurate estimate of the contact point is calculated, and as a result, the forces applied by the joints could be more precisely determined. Note that they did not address the force/torque mechanics in their work. Miller *et al.*[8] have developed a combination grasp planner and grasping simulator. They first simplify the object model into a union of shape primitives such as spheres, cylinders, etc. They then use heuristics to select candidates from a set of hand preshapes and hand approach positions/orientations related to each of the shape primitives. Each potential grasp is then evaluated using the grasping simulator. They demonstrate their program for a three-fingered robotic hand and a set of four 3D objects. Their planning method produced feasible grasps but rarely ones that were human-like. It required 478 s to generate 41 grasps for a conical flask and 120 s to generate 35 grasps for a cordless phone on a 1 GHz Pentium IV PC. Guan and Zhang[9] proposed a kinematic grasp feasibility analysis for a polyhedral object with triangular facets. They modeled the phalanges of the hand as cylinders. Similar to [7], they classified the contact points in different categories such as tip-face, link-edge, etc. Then they parametrically defined them by a set of equality and inequality constraints. The kinematic feasibility problem was then solved as a constrained nonlinear global optimization problem. They demonstrate their approach with four nonenveloping grasps of a rectangular prism object. Each grasp was generated in approximately 8 s on a 200 MHz Pentium II PC. An algorithm for synthesizing human-like enveloping grasps by shape matching with a database of grasp examples was presented by Li and Pollard.[10] The creation of a grasp database is necessary for their method. The algorithm can produce various hand poses relative to the object, but it cannot guarantee that the grasp will have a complete and proper kinematic contact between the hand and the object. They did not include timing results. Although it does not deal with enveloping grasps, the recent paper by Lopez-Damian *et al.*[11] is also relevant since they used the principle axes of the object to guide their grasp planner. Their algorithm plans grasps for a two-fingered, one DOF gripper, and polyhedral objects.

3. Algorithm Overview

3.1. Algorithm inputs

The required inputs are the model of the object, and the geometric model of the hand with its corresponding constraints. If a feasible grasp can be found, the outputs of the algorithm are the position and orientation of the palm and the angular joint positions of the fingers for grasping the object with an enveloping grasp. A relatively simple robotic dexterous hand design is adopted in this research. The hand consists of three fingers, with two phalanges and two revolute joints each, connected to a rigid palm. This design is shown in Fig. 1. It belongs to the class of hands termed "cylindrical" in [1], and it is intentionally kinematically similar to the robotic hands that exist in universities and companies. To demonstrate different aspects of enveloping grasp planning, the method is first implemented on a specially designed test object (see Fig. 2).

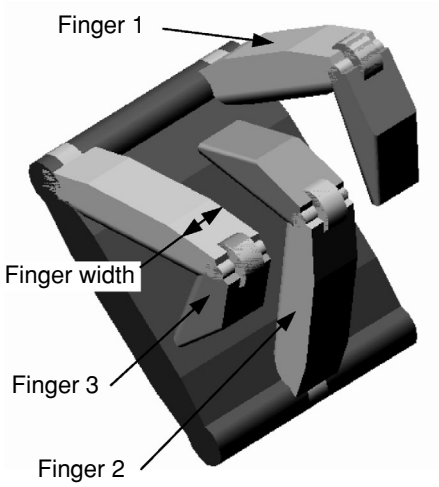

Fig. 1. Design of the dexterous hand.

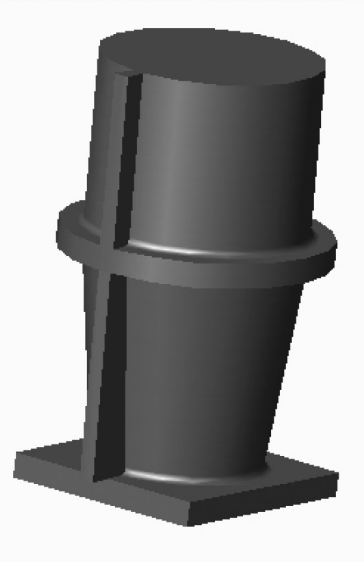

Fig. 2. Specially designed object for initially testing the algorithm.

The algorithm requires a discretized 3D model of the object. Converting the object's computer aided design (CAD) model to the stereolithography (STL) format is one way to accomplish this. This conversion creates a shell out of the outer surface of the object in the form of a triangular mesh. The STL file includes a set of coordinates for the vertices of these triangles and outward unit normal vectors for each of these triangles. Fig. 3 shows the test object in STL format.

3.2. Reduction of palm search dimensions

One of the great difficulties of planning enveloping grasps for 3D objects is the size of the search domain. Finding the position and orientation of the palm is a 6D search, and including the six angles of the finger joints increases the search dimension to 12. Conducting a 12D search is not efficient even with today's powerful computers. As a matter of fact, not all the object positions have good grasp possibilities, and based on the kinematic limitations of the given hand, many positions are not capable of any kind of grasp. Therefore, before conducting a detailed analysis, it would be better for the algorithm to have a preliminary

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

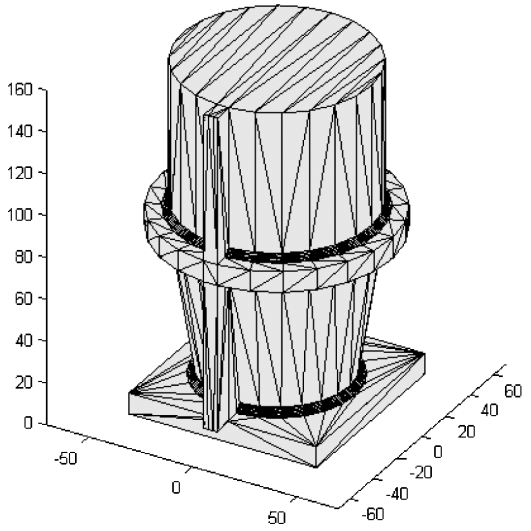

Fig. 3. Test object in STL format.

evaluation on which parts of the object are potentially graspable, eliminate regions which violate the kinematic restrictions and are not graspable, and rank the rest base on their potential suitability of a feasible grasp. Although this reduction task adds extra calculation time at the beginning of the procedure, it has valuable benefits as follows:

1. By using basic grasp principles, many ungraspable regions can be excluded from the search domain before conducting further grasp planning. This will eliminate redundant calculations considerably and reduces the execution time.
2. It produces a good estimate of where the algorithm should start placing the fingers onto the object to be reasonably close to a feasible grasp.
3. If the initial estimate does not lead to a feasible grasp, the grading system provides a set of ranked alternatives.

The first step in analyzing the potential grasp regions of the object is to decide how the palm should approach the specified region. The proposed technique considers the overall shape of the object and narrows the search down to a single initial approach of the palm. If a uniform density is assumed, it is easy to calculate the principal axes of the object. The principal axis with the smallest principal moment of inertia will be referred to here as simply the "principal axis." Conceptually, if the shape of the object is close to a cylinder, then the obvious choice for an enveloping grasp is to wrap the fingers of the hand around the principal axis. We extend this idea to any given object as a heuristic. This heuristic can be stated as follows: "*it is more probable to produce a stable enveloping grasp by having the palm approach the object parallel to its least inertia axis*." This reduces the dimensions of the palm position search space from three to one, namely the coordinate of the palm along the principal axis. The palm is then allowed to pitch and roll relative to the principal axis, but not to yaw. This reduces the orientation search space dimension of the palm from three to two. For convenience, we transform the object's coordinate system to make its Z axis equal to its principal axis. We also assume that the gravity force will be acting in the negative Z direction when the object is being carried by the robot.

4. Grading Method

4.1. Introduction

To direct the search and improve its efficiency, a new search domain grading method is introduced that is based on the following ideas:

1. Rather than simplifying the object and then considering this simplified object as a whole, find a way to remove the ungraspable sections of the object and only analyze the remaining sections. This will reduce the size of the search domain considerably without losing the object's geometric data.
2. Over the remaining object, grade each elevation or level of approach of the hand according to its suitability for grasping.

To implement the grading method, the 3D object is divided into a set of vertical 2D slices hinged on the object's principal axis. One slice is depicted in Fig. 4. The angle between successive slices is determined by the parameter *angular step*. Note that depending on the complexity of the object geometry, the *angular step* should be modified to balance the tradeoff between higher resolution and longer calculation time. In our implementation, we have found a value of 30° to be an effective choice.

The algorithm analyzes the contour of each slice and grades their elevation levels along the principal axis according to kinematic constraints, (i.e., finger thickness and palm dimensions), or grasp quality metrics (i.e., effective diameter and curvature of the object). Then it combines the grades from each slice together to create a 3D quality grade for the object. The elevation levels are created by discretizing the object in the Z dimension using a resolution equal to *elevation step*. As with the *angular step*, the value for *elevation step* is dependent on the level of detail with which the object should be analyzed. We have used a value of 2.5 mm in our implementation.

This grading method is straightforward to implement and frees the algorithm from analyzing the 3D object all at once. It simply cuts the object into many slices and analyzes each slice, extracts the data out of each slice and accumulates the results. This method can easily recognize the kinematically

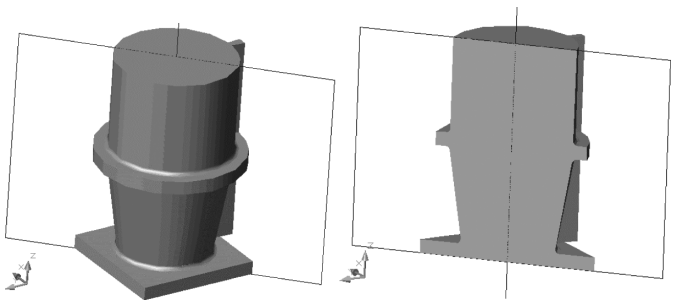

Fig. 4. Sample vertical slice of the test object.

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

ungraspable areas of the object (e.g., too narrow or too wide to grasp or with a curvature that is unreliable for positioning the finger (e.g., too sharply curved) and excludes them from the search domain. The methods for grading the object curvature and effective diameter are described below.

4.2. Grading the object curvature

The quality of a grasp is very dependent on the object surface curvature at, and near to, the contacts between the fingers and object. If the fingers are placed onto locations with better resting regions, suitable concavities, or where the side slope directs the grasping force in a way to better support the weight of the object, the grasp should be more stable and reliable.

The method begins by assuming that contacts between the fingers and object occur at both the left and right sides of the given vertical slice at the given discretized elevation level (whose resolution was set by the *elevation step* parameter). The right and left side contours near to the contact location are analyzed independently. The grades for each side are then averaged to determine the 2D curvature grade of the vertical slice at that elevation. The 2D grades from all vertical slices at a single elevation are then averaged to determine the 3D curvature grade for that elevation. Each 2D curvature grade is based on the angle of the normal vector at the point of contact, as well as the changes in the angle for the line segments that fall within a region of interest, H_{ROI}, above and below the contact point. A labeled diagram showing the right side of a vertical slice contour is given in Fig. 5. The curvature grading algorithm is as follows:

1. Set $i = 1$ and $j = 1$, where i is the index of the vertical slice through the object and j is the index defining the elevation of the contact between the finger and object.
2. Isolate the right side of the ith vertical slice to provide the contour for analysis.
3. Intersect the jth elevation with the contour to find the point v_{contact}.
4. Set $k = 0$, where k is the relative index of the line segment within the region of interest.
5. Compute the vertical distance H_{a+k}.
6. Compute $H_{\text{above}} = \sum_0^k H_{a+k}$.
7. If $H_{\text{above}} > H_{\text{ROI}}$ then set $\Delta\theta_{\text{above}} = 0$ and go to step 11.
8. Compute $\Delta\theta_k^{\text{above}} = \theta_{a+k} - \theta_{\text{contact}}$, where the θ angles are obtained using the normal vectors of the line segments.
9. If $\Delta\theta_k^{\text{above}} \geq \Delta\theta_{\text{aboveMax}}$ or $\Delta\theta_k^{\text{above}} \leq \Delta\theta_{\text{aboveMin}}$, then set $\Delta\theta_{\text{above}} = \Delta\theta_k^{\text{above}}$ and go to step 12; else set: $\Delta\theta_{\text{above}} = \frac{1}{k+1} \sum_{p=0}^{k} \Delta\theta_p^{\text{above}}$.
10. If $H_{\text{above}} < H_{\text{ROI}}$ then set $k = k + 1$ and return to step 5, else continue.
11. Repeat steps 5–10 to obtain$\Delta\theta_{\text{below}}$ by replacing the a subscripts with b.
12. Compute grades $G(\theta_{\text{contact}})$, $G(\Delta\theta_{\text{above}})$, and $G(\Delta\theta_{\text{below}})$ using the grading function presented in Fig. 6 and the parameters defined in Table 1.
13. If $G(\theta_{\text{contact}}) = -1$ or $G(\Delta\theta_{\text{above}}) = -1$ or $G(\Delta\theta_{\text{below}}) = -1$ then set $\text{SliceGrade}_{i,j} = -1$ and go to step 16; else, average the grades as follows:

$$\text{Grade}_{i,j}^{\text{Right}} = \tfrac{1}{3}(G(\Delta\theta_{\text{above}}) + G(\Delta\theta_{\text{below}}) + G(\theta_{\text{contact}}))$$

14. Repeat steps 2–13 for the left side of the ith vertical slice.
15. Average the right side and left side grade results to obtain a grade for the ith slice and the jth elevation step: $\text{SliceGrade}_{i,j} = \frac{1}{2}(\text{Grade}_{i,j}^{\text{Right}} + \text{Grade}_{i,j}^{\text{Left}})$
16. Set $j = j + 1$ for the next elevation step. If $j \leq m$, where m is the number of elevation steps, go to step 2; else set $j = 1$ and continue.
17. Set $i = i + 1$ for the next vertical slice. If $i \leq n$, where n is the number of vertical slices, go to step 2; else continue.
18. Average grades for all vertical slices corresponding to each jth elevation step to obtain the grades for the 3D object at each elevation step:

$$\text{CurvatureGrade}_j = \tfrac{1}{n} \sum_{i=1}^{n} \text{SliceGrade}_{i,j} \quad \text{for } j = 1, 2, \ldots, n.$$

19. Normalize the set of positive CurvatureGrades such that the maximum value is unity.

Further explanation of this algorithm and the required parameters is warranted. The size of the region of interest H_{ROI} should be proportional to the finger width (see Fig. 1). Since the surface in the vicinity of the finger, and not just the surface under it, is important to the quality of a grasp, we set H_{ROI} equal to 0.75 times the finger width in our implementation. This produces a 25% margin on either side of the finger. The vertical distances of the vertices above and below the contact point are checked to see whether they fall within the region of interest. If the vertex is outside the region of interest, as depicted below the point of contact in Fig. 5, then there is no change in angle through that region, and $\Delta\theta_{\text{below}}$ in this case is set to zero. If the vertical distance between the vertex and the contact point is less than the

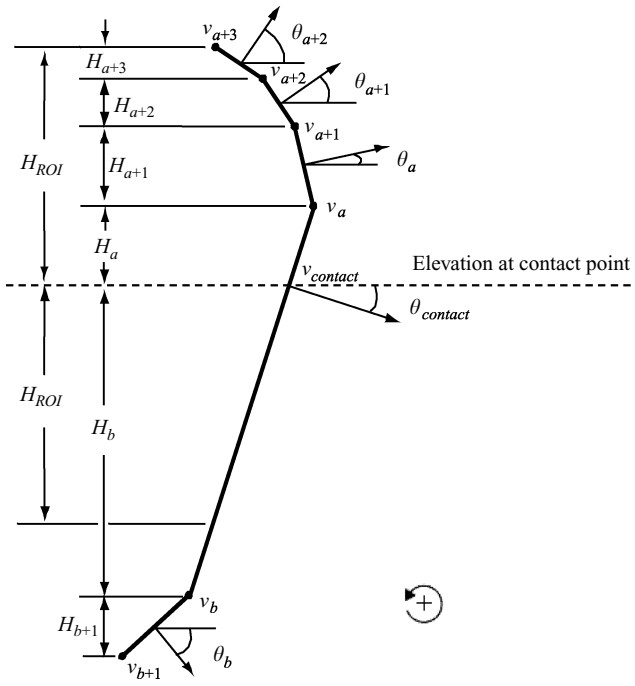

Fig. 5. Geometry used in the curvature grading algorithm (right-side case).

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

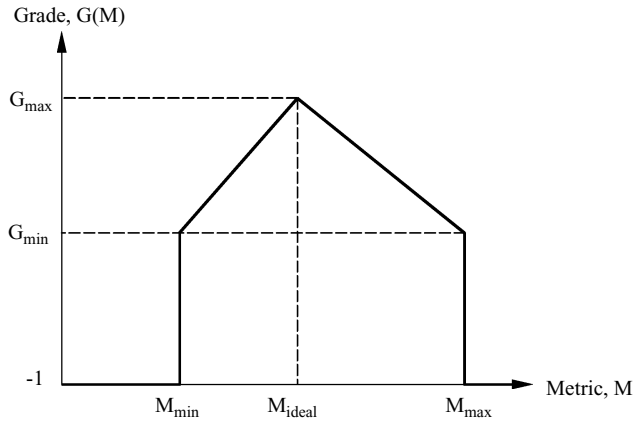

Fig. 6. Grading function used with the quality metrics.

region of interest (as shown above the contact point in Fig. 5), it becomes necessary to take into account the next line segment along the contour. This process iterates until the vertical distance between vertices covers the entire region interest. The change in angle of each normal vector relative to the contact normal vector is determined for the line segments within this region. The overall change in angle, either $\Delta\theta_{\text{above}}$ or $\Delta\theta_{\text{below}}$, is set equal to the average of these values.

Once θ_{contact} and the changes in angle are known, the grade is determined by using the function depicted in Fig. 6. This grading function is used for both the curvature grading and the effective diameter grading that is described further in Section 4.3. This function was developed based on the assumption of an ideal quality metric. When the metric, e.g., θ_{contact}, equals its ideal value, the highest grade is assigned. The grade decreases linearly as the metric shifts away from the ideal, until reaching the boundary conditions defined by a maximum or minimum value of the quality metric. Beyond these boundaries, the metric is deemed ungraspable and a grade of -1 is assigned. The values chosen for the maximum, minimum, and ideal quality metrics, along with their associated grades, are listed in Table I.

For θ_{contact}, it is assumed that a vertical contour ($\theta_{\text{contact}} = 0$) would result in an ideal grasp if gravity were not acting on the object since the finger force would be normal to the surface. It is well known that with gravity taken into account, a slightly negative θ_{contact} is ideal for lifting the object since this reduces the reliance on friction to prevent slipping. This is reflected in the design of household objects such as cups. With this in mind, the ideal value of θ_{contact} was chosen to be -10°. The maximum and minimum values of θ_{contact} were chosen to be 40° and -60°, respectively. Beyond these values, the finger contact is likely to slip along the surface, making for an undesirable grasping scenario.

Table I. Chosen grading parameters.

	Parameter				
Quality Metric	$M_{\max}$	M_{ideal}	$M_{\min}$	$G_{\max}$	$G_{\min}$
θ_{contact}	40°	-10°	-60°	27	19
$\Delta\theta_{\text{above}}$	20°	-20°	-60°	9	0
$\Delta\theta_{\text{below}}$	20°	-20°	-60°	9	0
Effective diameter	150 mm	30 mm	12 mm	9	0

The ideal angle change both above and below the contact was chosen to be when the curvature is slightly concave ($\Delta\theta_{\text{above}} = -20^\circ$ and $\Delta\theta_{\text{below}} = -20^\circ$), creating a niche for the finger to rest within. The grade assigned to this ideal value is 9. While a certain amount of concavity is beneficial, there is a limit to how sharp the angle change should be, since a tight contour change may indicate a location where the finger would be constricted. The minimum angle change was therefore designated at -60°. The maximum change in angle for the curvature above and below the contact has been designated as 20°, corresponding to the case where the curvature is convex. The magnitude of this upper bound was deliberately chosen to be smaller than the magnitude of the lower bound since a relatively small amount of convexity can produce an unstable finger placement. At the maximum and minimum bounds for the angle change both above and below the contact point, the grade is set to zero. Beyond these bounds, the contour is considered ungraspable, and a grade of -1 is assigned.

The grades assigned to θ_{contact} angles within the graspable range vary linearly from 19 to 27, the latter corresponding to the ideal case. These were chosen so that θ_{contact} dominates the curvature grading, since the slope at the contact surface has the greatest bearing on how easily graspable the object is at the given elevation. If, for example, $\Delta\theta_{\text{above}}$ and $\Delta\theta_{\text{below}}$ are ideal (i.e., producing grades of 9 and 9, giving a total of 18) but θ_{contact} is very close to one of its bounds, then its grade of 19 would still dominate the outcome.

Fig. 7 shows the result of curvature grading applied to the test object. An arbitrary vertical slice has been superimposed onto the grading results to make them easier to understand. To improve the clarity of the graph, the grades have been magnified by a factor of 50. Note that the portions of the object less than half a finger width from the top and bottom are removed from the search domain before grading begins. Starting from the top, the first two grades are negative because

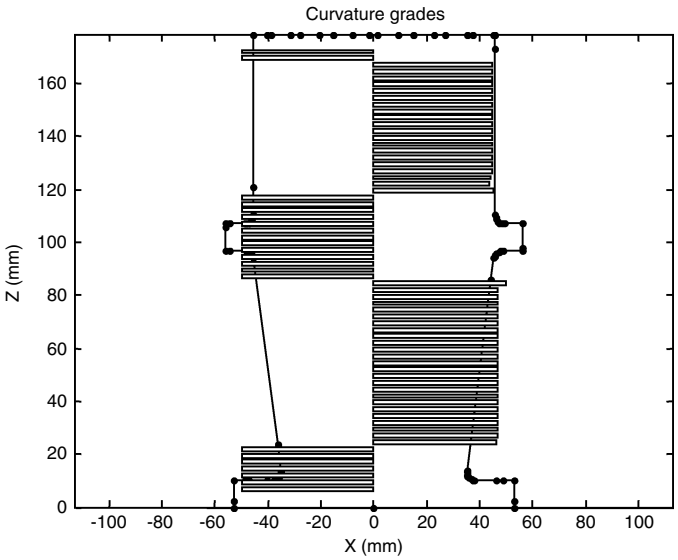

Fig. 7. Curvature grades for the test object (values magnified by 50).

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

their $\Delta\theta_{\text{above}}$ within H_{ROI} was outside its maximum bound. The grades along the cylindrical section are positive and constant because θ_{contact}, $\Delta\theta_{\text{above}}$, and $\Delta\theta_{\text{below}}$ are constant and within their acceptable ranges. The grades then decrease slightly when θ_{contact} becomes positive above the ridge. When the ridge is within H_{ROI}, the grades become negative as the sudden changes in angle categorize them as ungraspable. The first positive grade below the ridge is the highest value. This is the result of the favorable values of θ_{contact} due to the tapered cylindrical section and of $\Delta\theta_{\text{above}}$ due to the concave fillet above this elevation. The grades then drop slightly after the fillet is outside H_{ROI}, but they remain higher than along the straight cylinder. When the sudden angle changes due to the bottom ridge are within H_{ROI} the grades are again set as ungraspable.

4.3. Grading the effective diameter of the object

With an enveloping grasp, the fingers should wrap around the object; the tighter the fingers wrap, the more stable the grasp will be. The fact that the narrower the object is, the better the fingers can wrap around it, led us to another grading parameter: the *effective diameter* of the object at each elevation.

The *effective diameter* is obtained by averaging the widths of all the vertical slices at a particular elevation. To find the width:

1. The algorithm begins with the first vertical slice.
2. The left outermost and right outermost points crossing this elevation are found.
3. The distance between these points measures the width.
4. This procedure repeats for all of the vertical slices.

The *effective diameter grades* are obtained from the *effective diameter* metric using the grading function shown in Fig. 6, followed by normalization to unity. We denote the parameters corresponding to M_{min}, M_{max}, and M_{ideal} as D_{max}, D_{min} and D_{ideal}, respectively. The choice of $D_{\text{ideal}} = 30$ mm, as listed in Table I, was based on its prominent use in items requiring a strong grasp, such as hammers and screwdrivers. The values of D_{min} and D_{max} are derived from the kinematics of the hand.

D_{min} is set equal to the diameter of the smallest cylinder that can fit in the hand when one or more finger is at the lower limit of its joint angular range. Figure 8 shows the corresponding geometry for the given hand design. Note that the joint limits lead to a triangular space between the

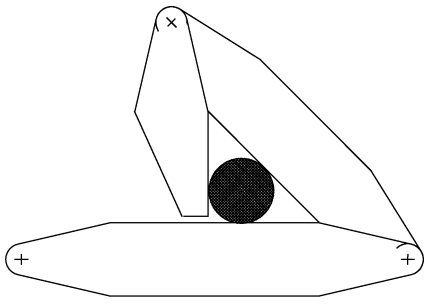

Fig. 8. Determination of the D_{min} from the joints positioned at their lower limits.

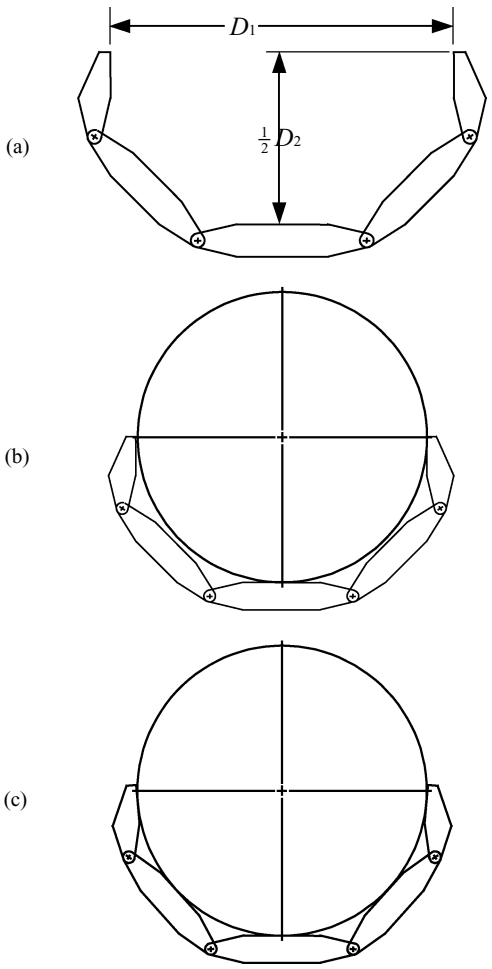

Fig. 9. (a) Geometry of the hand when first joint angles are set equal to their maxima and contact surfaces of second phalanges are made parallel. (b) Non-enveloping grasp of an object with diameter equal to D_{max}. (c) Enveloping grasp of an object with diameter equal to D_{max}.

phalanges. This parameter can therefore be calculated using the radius formula for an inscribed circle as follows:

$$D_{\text{min}} = \frac{2K}{S} \tag{1}$$

where K is the area of the triangle and S is the semiperimeter of the triangle. For our hand design, this parameter equals 12 mm.

For D_{max}, assuming the worst case of an object with a very small friction coefficient, the upper limit for grasping a cylinder will occur when the second phalanges are parallel and the contacts occur at the tips of the phalanges and the cylinder's diameter, as illustrated in Fig. 9. The first joint angles are set at their maximum values, and the second joint angles are set such that the second phalanges are parallel, as shown in Fig. 9(a). The distances D_1 and D_2 are then calculated using geometry, and the desired parameter is given by:

$$D_{\text{max}} = \min(D_1, D_2) \tag{2}$$

For our hand design, this parameter equals 150 mm. Note that in Fig. 9(b), the first phalanges are not in contact with the object. Since all phalanges must touch the object with our

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

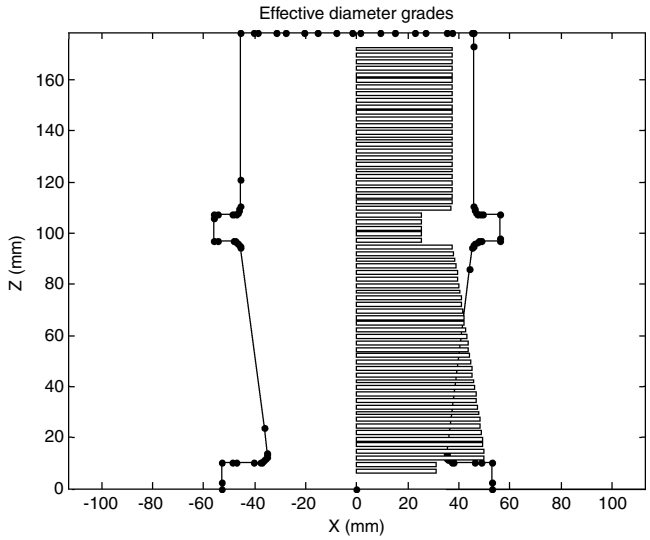

Fig. 10. *Effective diameter* grades for the test object (horizontally magnified by 50).

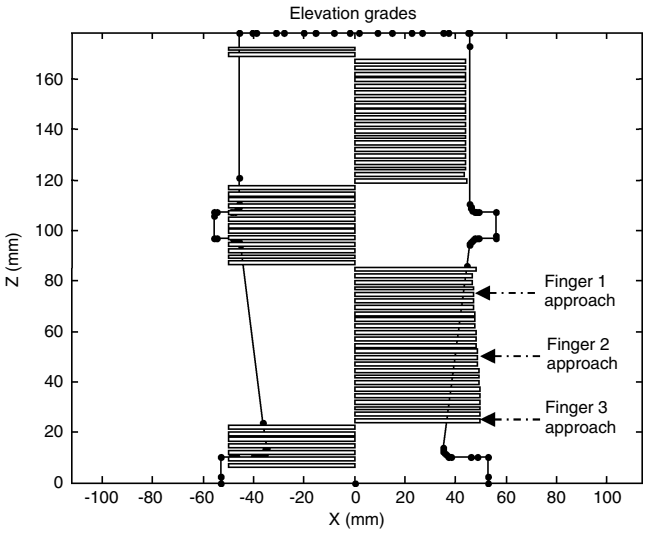

Fig. 11. Elevation grades for the test object (horizontally magnified by 50).

planning method, the actual contacts on the second phalanges may occur slightly below the diameter of the largest graspable cylinder as shown in Fig. 9(c). This grasp will be stable for nonslippery objects.

Figure 10 shows the *effective diameter grades* for the test object. Since all of the *effective diameter* values are greater than D_{ideal} and less than $D_{\max}$, the grades are all positive and larger grades have been assigned to the smaller diameters.

4.4. Combining the curvature and effective diameter grades

The last step in this stage is to combine the grades from both the curvature and effective diameter ratings. A weighted average will be used, and the weighting factors are termed *significance factors*. These factors should be set by the user according to importance of *curvature* or *effective diameter* for their application. The equations are as follows:

$$\text{Elevation Grade} = f_C \times \text{curvature grade} + f_D \times \text{effective diameter grade} \quad (3)$$

$$f_D = 1 - f_C \quad (4)$$

where f_C is the *curvature significance factor* and f_D is the *effective diameter significance factor*. In our implementation, we chose $f_C = 0.6$, and as a result of (4): $f_D = 0.4$, giving more importance to curvature relative to the effective diameter. Figure 11 shows the combined *elevation grades* obtained using these significance factors. The approach elevations corresponding to the highest hand-based grade, explained further in the next section, are also indicated.

4.5. Adding the palm dimensions to the grading

So far, the object has been analyzed over the set of discretized elevation layers. This can be interpreted as finding the quality of the grasp for only one finger of the hand. Since the goal is to place the entire hand (including the palm and three fingers) properly onto the object, it is necessary to combine the grades for the individual fingers based on the dimensions of the palm. These hand-based elevation grades will be used to guide the initial placement of the palm.

The *largest span height* is defined as the largest vertical distance between two consecutive ungraspable elevations. The *approach vector* of the palm is defined as the unit normal vector directed outwards from the center of the palm surface. To be able to place the palm onto the object, the *largest span height* should be greater than the palm width, or the distance between the first and third finger outlines. This condition is based on a conservative assumption that the palm approaches the object with its *approach vector* horizontal, perpendicular to the principal axis. For most objects, the palm will rotate to make multipoint contact with the object. Choosing the vertical palm alignment makes the algorithm conservative, since a rotated palm requires less vertical distance. If the *largest span height* does not comply with this condition, then the object is considered ungraspable, and no further calculations are done.

If the object passes the previous stage, it means that there is at least one height span that enables the palm to approach the object. The hand-based grading rules are:

- Position scanning is done from the top of the object to the bottom.
- The hand-based grade equals the average of its three finger elevation grades.
- When positioning the palm, the elevation span between Finger 1 (top finger) and Finger 3 (bottom finger) should be graspable.
- The palm cannot penetrate into the table, or into the object.
- If an ungraspable elevation appears between Finger 1 and Finger 3, then Finger 1 will move to the next graspable elevation below this ungraspable elevation. Additionally, all the grades between the elevations of the old position of Finger 1 and the new position of Finger 1 will be assigned as ungraspable. This step eliminates any future searching process over the locations where the palm cannot be placed to rest properly.

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

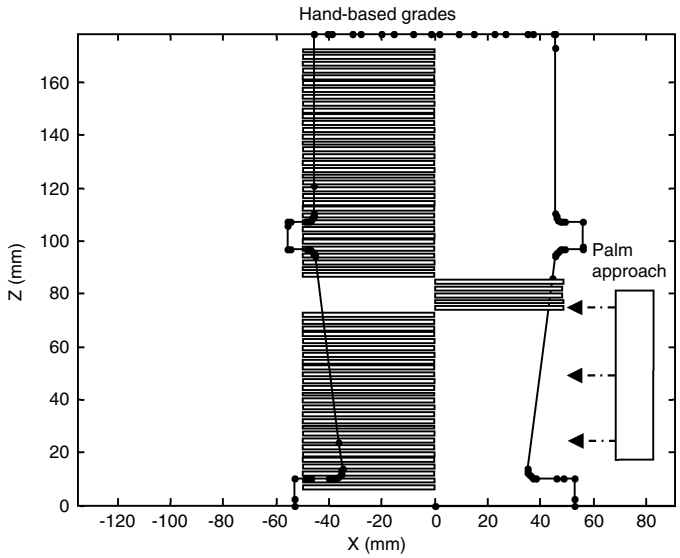

Fig. 12. Hand-based grades and approach elevation of the palm.

Figure 12 shows the hand-based grades and remaining search domain of five elevation levels. The approach elevations of the palm and of the three finger centerlines for the final grasp are also shown.

4.6. Sorting the grades

The vertical positioning of the palm is guided by the hand-based grades, where the best grade will have the first choice. In some situations, there are elevations having the same grade, so an additional quality factor is necessary to break the tie. Since physically it is more convenient and practical for a robot hand to reach the object from the top, in the case of similar grades the indices will be sorted from higher elevation to lower elevation. This guides the algorithm to search for feasible grasps at higher elevations before trying the lower ones in case of the same quality of potential grasps.

5. Finger Modeling

The algorithm deals with a minimal set of hand kinematic parameters as input. The input parameters used to describe the fingers are their thickness, width, phalange lengths, and angular ranges of motion of their joints. This minimal list has the advantage of being computationally efficient but also introduces algorithmic challenges due to the lack of detailed information.

Different types of robot fingers can have different cross sections that are not necessarily rectangular, square, or circular. Therefore, even for the same object, palm dimensions, finger thicknesses, hand elevation, and approach angle, with different finger cross sections, the position of the real contact point between the finger and the object may be different. Another difficulty is that the method has already approximated the smooth curved surface of the object with a series of triangular facets (from the conversion to STL format); this approximation will create an error between the modeled and real contact points. However, enveloping grasps have the benefit of being robust to variations in the contact locations, so determining them precisely is unnecessary. This allows us to develop a simple and efficient solution to the problem.

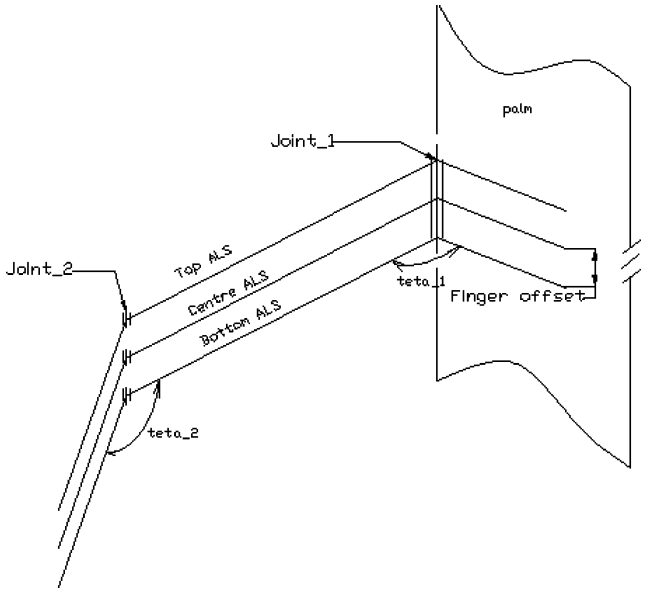

Fig. 13. ALS model of a finger.

With our new finger model, each finger is modeled by three articulated line segments (ALS). These ALS are used to model the top, bottom, and centerline of the contact surface of the phalanges respectively. Figure 13 shows the ALS model proposed for one finger. Figure 14 shows the top ALS in the top view of Finger 1. Each ALS will be treated as a virtual finger and will be separately placed onto the object and checked for kinematic feasibility.

The benefits of this model are:

- The real contact must occur vertically somewhere between the top ALS and the bottom ALS of each finger. By analyzing the joint angles related to these ALSs, their range can be found. If any part of this range violates one of the angular ranges of the joints then the finger placement candidate will be rejected. This is much simpler than finding the vertical location of the real contact point.
- To keep the three ALSs as a finger unit, they are kinematically constrained together. The *joint angular*

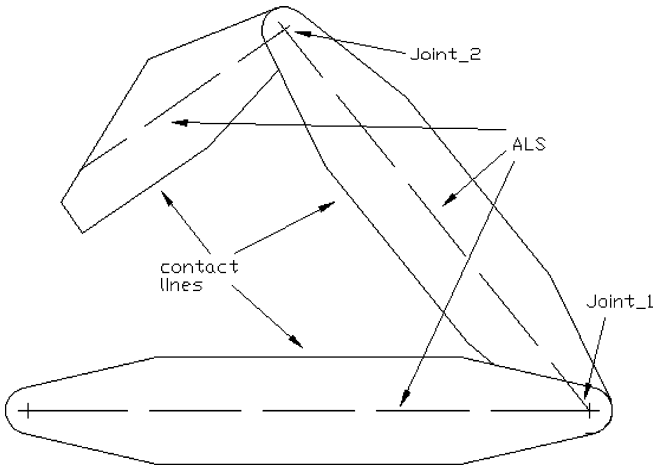

Fig. 14. Top ALS for Finger 1.

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

deviation is defined as the allowable change in the joint angles of the three ALS representing one finger. This parameter is used to verify the consistency of each set of ALSs before they are accepted. In our implementation, these parameters are set for the first and second joints of each finger to: $\Delta\theta_{1\text{Max}} = 5^\circ$ and $\Delta\theta_{2\text{Max}} = 5^\circ$.

For instance, the $\Delta\theta_{1\text{Max}} = 5^\circ$ means that in each finger the differences between the first joint angle created by placing the top or bottom ALS onto the object and the first joint angle created by placing the center ALS of the finger onto the object should not exceed 5°.

6. Final Palm and Finger Placement

The combination of the palm and finger phalanges form an articulated linkage set. The position and orientation of each link is dependent on the position and orientation of the preceding link. The process of grasp planning over the sorted search domain found in Section 5 is composed of two stages: (1) palm placement and (2) finger positioning.

6.1. Palm placement

The first stage of the grasp planning is to place the palm onto the surface of the object:

- The procedure begins by elevating the palm to the potentially feasible location with the best grade found during the grading procedure.
- The vertical slicing plane is set to be coplanar with the plane vertically passing through the middle of the palm.
- The approach vector is made to pass through the principal axis and to lie on the XZ plane.
- Any penetration of the object and hand will be checked, and if detected, the grasp attempt will be rejected.
- It is assumed that the object rests motionless while the palm contacts it.

The analysis is performed in the XZ plane. To put the palm into contact with the slice, the palm surface line segment is moved toward the object until it touches one of the line segment endpoints in its domain. After the first contact point is found, it is saved and will then be used as a rotation pivot point. The palm will pitch around it until it hits the second contact point.

Since the hand is three fingered and the second finger is located between and opposite to the first and third fingers, by spreading the palm contact points on both ends (i.e., top and bottom) of the hand, the resultant palm placement will be more stable. To achieve this desirable situation, the conditions applied to the rotation direction are as follows:

- If the first contact point is on the top half of the palm, then the palm will rotate to find the second contact point on the bottom half (i.e., the direction of rotation is chosen to move the lower half of the palm closer to the object).
- If the first contact point is on the bottom half of the palm, then the palm will rotate to find the second contact point on the top half.

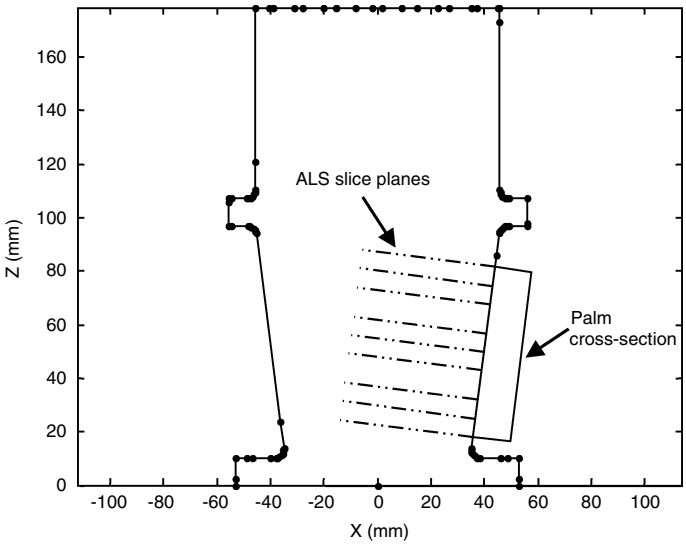

Fig. 15. Palm placement in the vertical slice plane.

- If the first contact point is located at the midpoint of the palm, then the algorithm will find the second outermost point in the range of palm rotation, either on the top half or bottom half, and set it as the second contact point.

When the algorithm is trying to place the palm onto the object, the palm is modeled by one line segment defining its contact surface in the XZ plane. After virtually placing the palm onto the object, the algorithm must check if the physical thickness of the palm makes this palm location infeasible due to collision. In other words, the palm and the object should not have any kind of intersection with each other. This check is performed by finding the outline of the palm in the XZ plane, and testing for any intersection of these lines with the vertical slice contour. Figure 15 shows the palm position in the vertical slice plane. This position passed the intersection check.

6.2. Finger positioning

Before continuing with the next stage, the finger slice planes should be defined. The slice planes are the planes created by rotating the ALSs of the finger about its first or second joint. These three planes (top, center, and bottom for each finger) are created for each of the hand's three fingers, such that there are nine planes in total. Cross sections of these slice planes are shown in Figure 15.
Note:

- All the finger positioning should be analyzed over the object contours created by cutting the STL-formatted object model with these planes.
- Before passing this phase, the algorithm tests for any penetration between the palm outline and the object contour in each of these planes. If any penetration is found then the grasp candidate is rejected.

By having the palm location fixed, the position of the first joint that connects the palm to Phalange 1 can easily be found. In order to find the resting locations of the first phalange, the

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

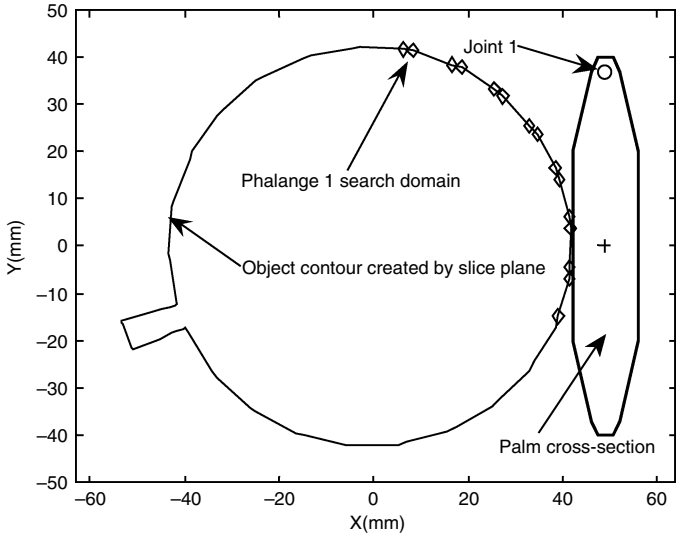

Fig. 16. Contact search domain for the first phalange.

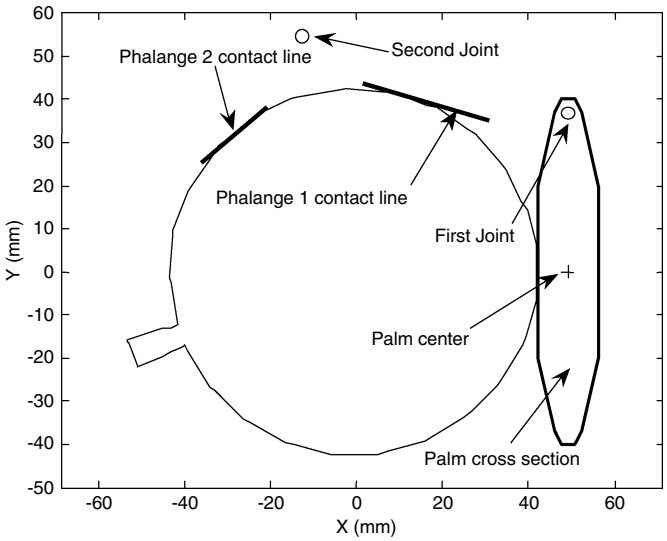

Fig. 17. Center ALS of Finger 1 placed onto the test object.

algorithm rotates this link to find its contact point with the object by executing the following sequence:

1. *Determining the phalange search domain*: In order to remove unnecessary calculations, the phalange contact will only be checked over the portion of the object that is physically reachable by the contact surface of the phalange. This search domain will be a subset of the object contour created by the finger slice plane as illustrated in Fig. 16.
2. *Finding the contact point*: The contact point of the phalange is found by first measuring the angles of joint 1 that occur when the phalange contacts the points in its search domain. The phalange contact point is the one corresponding to the widest joint 1 angle.
3. *Angular range check*: The angle of contact found should be within the angular range of the first joint, otherwise the contact is rejected, and as a consequence the grasp candidate is rejected.
4. *Phalange and object separation*: The algorithm checks for any penetration of the object and the physical body of Phalange 1, including the regions containing the joints. If any penetration is detected the grasp candidate is rejected.

Figure 17 shows a successful center ALS placement for Finger 1. This placement satisfied all of the kinematic constraints.

The results given so far suggest that the Finger 1 center ALS has passed the finger positioning check. However, each finger is modeled with three ALSs, and each ALS must be positioned over the object and checked. If any of them fail, then the grasp is rejected. For the same palm position, the top ALS and bottom ALS for Finger 1 were successfully positioned as well. Their corresponding joint angular deviations were within $\Delta\theta_{1\mathrm{Max}}$ and $\Delta\theta_{2\mathrm{Max}}$. Finally, since all tests were passed, the positioning of Finger 1 was accepted. The same routine must be repeated for the remaining two fingers. If at any stage the positioning failed, then the grasp for that position is rejected and

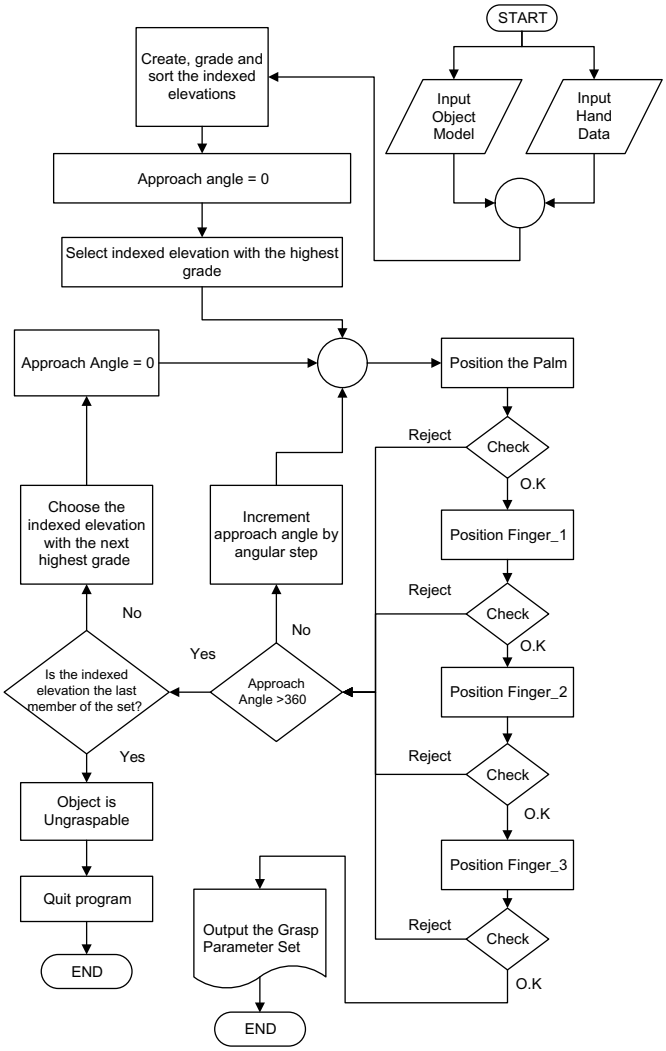

Fig. 18. Flowchart of the complete grasp planning algorithm.

the process is restarted for the next point in the search domain.

All tests are passed for the assigned palm elevation and approach angle, so the grasp is kinematically feasible and its

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

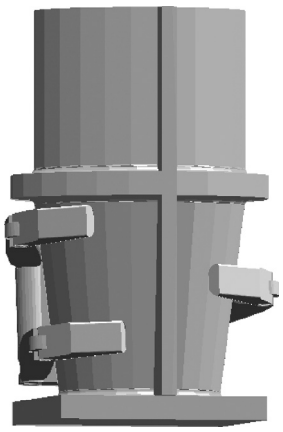

Fig. 19. Grasp planned for the test object.

kinematic parameters are as follows:

$$\text{Finger 1} \rightarrow \theta_1 = 107.7^\circ \quad \theta_2 = 124.8^\circ$$
$$\text{Finger 2} \rightarrow \theta_1 = 103.9^\circ \quad \theta_2 = 122.0^\circ$$
$$\text{Finger 3} \rightarrow \theta_1 = 99.6^\circ \quad \theta_2 = 118.9^\circ$$

$$\text{Approach angle} = 90^\circ$$
$$\text{Palm pitch angle} = 7.3^\circ$$
$$\text{Finger 1 approach elevation} = 74.5\,\text{mm}$$

The steps of the complete grasp planning algorithm are summarized in the flowchart shown in Fig. 18.

Figure 19 shows a 3D view of the grasp found for the test object. Note how the fingers have been placed under the horizontal ridge and have avoided unreliable contact with the vertical ridge.

7. Results for Other Objects

Ball (or sphere), pop bottle, cordless phone, cup, block (or rectangular prism), and spray bottle objects were used to further evaluate the performance of the grasp planning algorithm. Intermediate results are shown for the ball and pop bottle in Figs. 20 and 21. The curvature grades, effective diameter grades, combined elevation grades, and final palm placements are depicted. For the curvature grading of the sphere, starting from the top, the first four curvature grades are negative because the θ_{contact} values at those elevations are

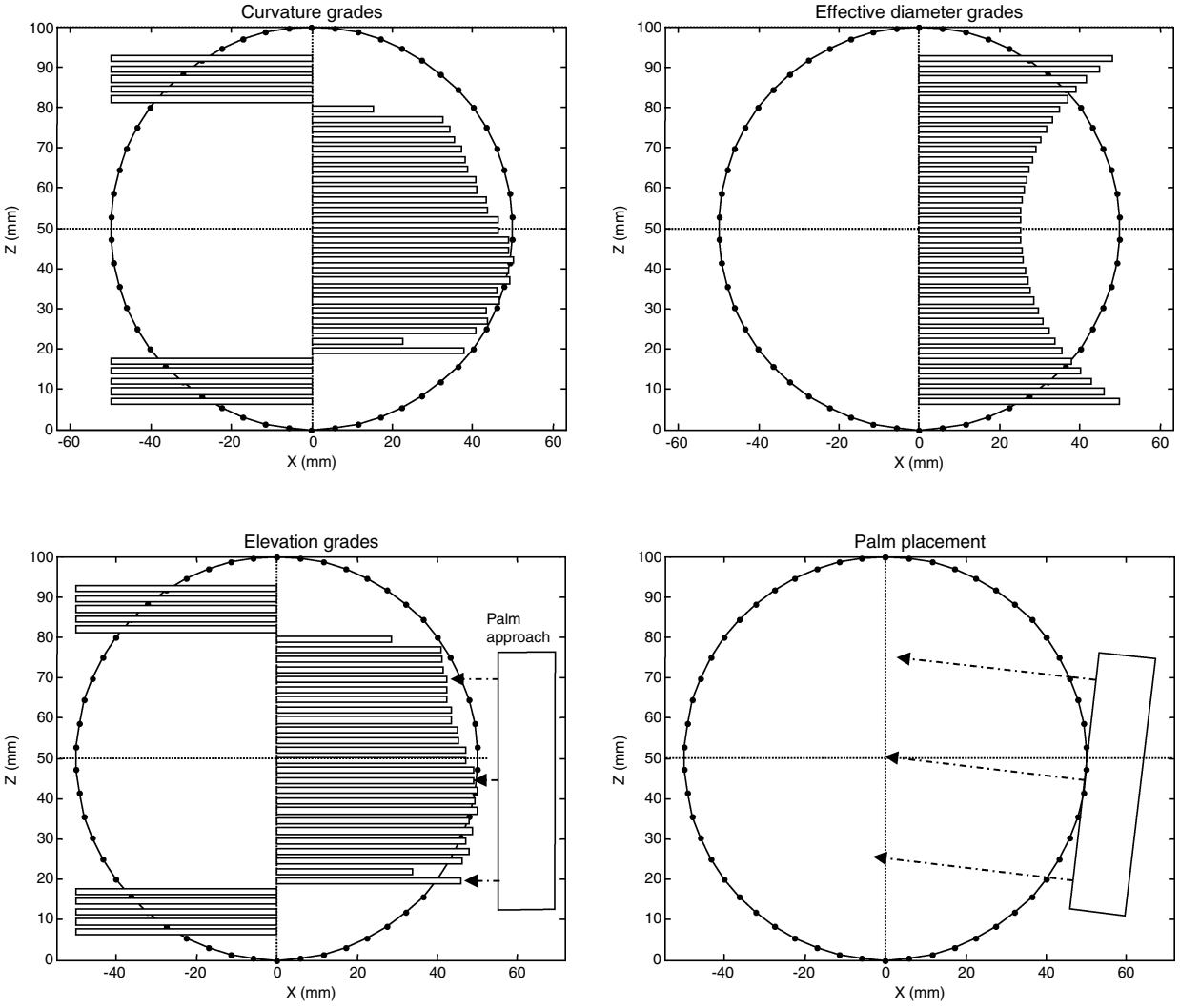

Fig. 20. Grades, palm approach, and palm placement for the ball.

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

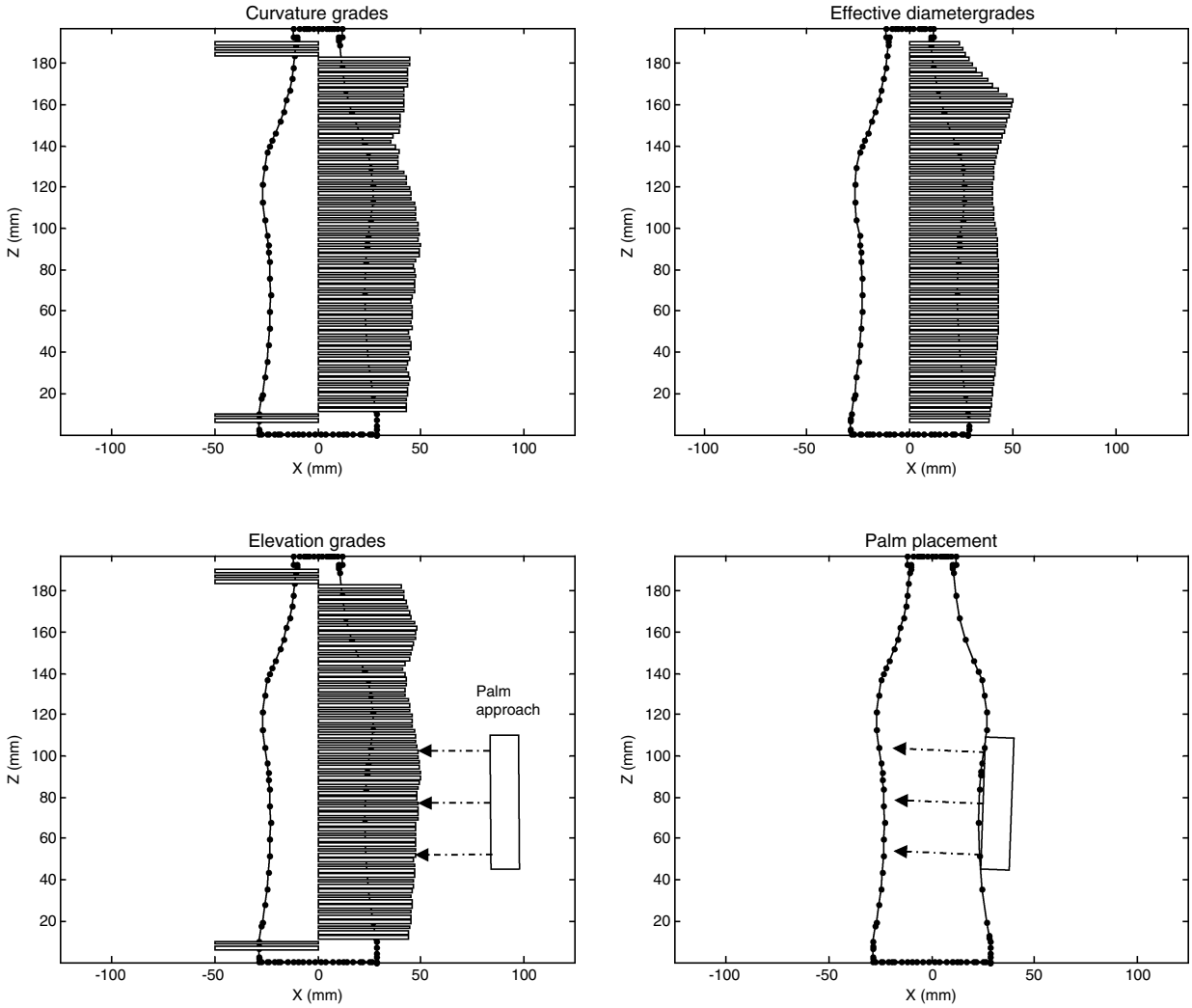

Fig. 21. Grades, palm approach, and palm placement for the pop bottle.

greater than the maximum bound. The grades then become increasingly larger as θ_{contact} approaches the ideal value of -10°, slightly below the centerline of the sphere. Recalling that the hand-based grade is the average of the elevation grades for the three fingers, the highest hand-based grade occurs when the palm approaches the sphere with Finger 2 slightly below the 50 mm centerline as shown. The final palm placement has been rotated to make proper contact as shown.

For the curvature grading of the pop bottle in Fig. 21, the top three grades are negative because the associated $\Delta\theta_{\text{above}}$ values are outside their maximum bound. Along the contour toward the bulge in the pop bottle, the curvature grades are positive and decrease as θ_{contact} increases. The highest curvature grade occurs around $z = 100$ mm, due to the combined benefit of a slightly negative θ_{contact} and a desirable $\Delta\theta_{\text{above}}$ due to the presence of the slight concavity. The curvature grades then decrease slightly along the lower portion of the bottle as θ_{contact} increases. When the bottom of the bottle is within H_{ROI}, the grades become negative as the sudden changes in angle exceed the graspable bounds. The effective diameter grades are largest at $z = 160$ mm, where the throat of the bottle reaches the ideal diameter. The grades decrease slightly at the bulge, and then increase again somewhat in the thinner section of the bottle. After combining the grades, the best results for the palm approach occur directly under the bulge in the bottle, where the fingers are supported by the gradual concavity of the contour.

The final grasps for the ball, pop bottle, phone, cup, block, and spray bottle are presented in Fig. 22. The human-like nature of all of these grasps demonstrates the effectiveness of our planning method. The grasps for the pop bottle, phone, and spray bottle demonstrate how our method takes advantage of concavities whenever possible. The grasps for the ball, cup, and block demonstrate that our method also works well when concavities are not present. Finally, the grasps for the block and phone illustrate that our method can employ contact at the distal end of the second phalange when it is necessary for grasping.

Timing results are presented for all seven objects in Table II. The times were obtained with the algorithm coded in Matlab 7.0 running on a 2.0 GHz Intel Centrino Duo CPU PC. In general, the times increased with the number of object model facets and with the number of elevation levels. A strict relationship between the execution time and these parameters cannot be obtained however because the complexity of the object shape plays a significant role. For

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

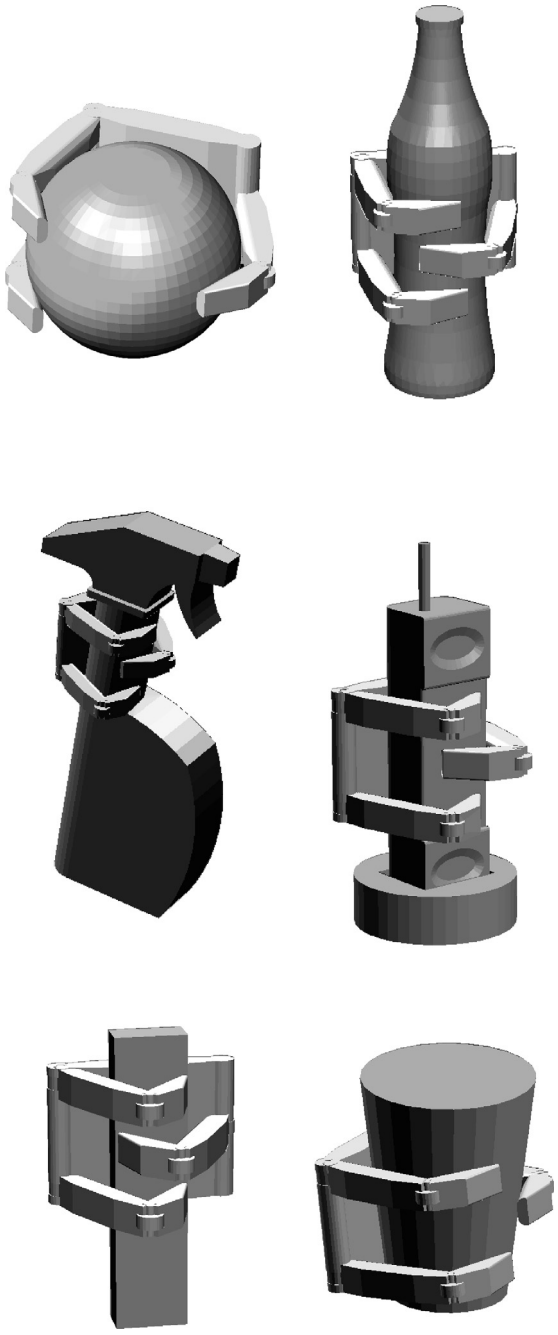

Fig. 22. Grasps obtained for household objects. Clockwise from top-left: ball (sphere), pop bottle, cordless phone, cup, block (rectangular prism), and spray bottle.

example, the test object has roughly one-third the number of facets as the pop bottle and roughly the same number of elevation levels, but its execution time is roughly two-thirds that of the pop bottle. The planning difficulty added by the test object's horizontal and vertical ridges explains its relatively slower execution. We also tested the influence of the number of vertical slices. This number is inversely proportional to the *angular step*. The presented results were produced using *angular step* = 30°. When this was decreased to 5°, similar grasps were generated, and the average execution time for the seven objects increased from 5.3 to 11.1 s. This demonstrates the influence of the number of vertical slices and also justifies our original choice for the *angular step*.

Table II. Execution times for grasp planning.

Object	Number of facets	Number of elevation levels	Execution time (s)
Prism	12	45	0.7
Cup	92	39	1.2
Phone	648	58	2.0
Spray bottle	384	94	2.8
Sphere	2808	37	4.3
Test object	1282	67	10.0
Pop bottle	3736	74	16.4

8. Conclusions

In this paper, a new method is presented for 3D enveloping grasp planning with a dexterous hand. The objective of this method is to find a high-quality kinematically feasible 3D enveloping grasp for an arbitrary object whose 3D model is available, by applying a novel grading approach along with the kinematic constraints of a three-fingered dexterous hand. The method has been successfully tested with several object shapes and sizes. The characteristics of this method are as follows:

1. A new approach for defining the search domain for grasp planning is proposed. The algorithm has three significant benefits:
   - By using the kinematic constraints (i.e. hand dimensions, curvature constraints, and effective diameter constraints), the method eliminates the ungraspable areas from the search domain. As a result, the remaining search domain has a better probability of a successful grasp.
   - Before beginning the steps for finger positioning, the method will consider the kinematic constraints of the palm geometry and eliminate all the areas of the search domain in conflict with these constraints.
   - In the process of allocating the search domain, if any of the object areas is potentially graspable, the method assigns a grade to it based on the kinematic preferences of palm and finger positioning for an enveloping grasp (specifically, the curvature and effective diameter). The method uses these grades to sort the potentially graspable members of the search domain. This sorting strategy directs the algorithm to test the areas of the object that should result in a better quality grasp first.

   The outcome of this approach is a reduced and graded search domain. It is important to have a reduced search domain since it requires less calculation time to analyze. It is also important for the search domain to be graded since it exposes the places with potentially better quality of the grasp earlier in the planning process.
2. A simple and practical method is proposed to model the 3D finger. The great benefit of the new model is that it frees the grasp planning process from the complexity of the calculations for finding the real finger contact points with the object. In the proposed method, each finger is modeled with three ALSs that are bound to each other by a kinematic constraint (the *joint angular deviation*). Each of these ALSs is tested with the object separately. In order

https://doi.org/10.1017/S0263574707003955 Published online by Cambridge University Press

to have a successful contact, they have to comply with the kinematic constraint of the hand (e.g., joint movement range) and the kinematic constraint that bounds them together.

3. In the proposed method, each of the grasp parameters (i.e. palm position, palm orientation, and joint angles) is calculated in separate stages. If at any stage, the algorithm finds the grasp unfeasible, it will stop the current analysis and will go to the next available member of the search domain. This considerably simplifies the numerical analysis and keeps the grasp planning procedure free of any redundant calculations.
4. A database of shape primitives, hand preshapes, or grasp examples are not required.

The proposed method has been successfully implemented and tested using seven 3D object models, ranging from a block with 12 facets to a pop bottle with 3736 facets. It has been shown to generate human-like high-quality grasps for each object in under 20 s.

Acknowledgements

The authors wish to thank Andrew Lambert for his assistance with the creation of the object models and the 3D graphical display of the grasps, and Heather Ker for her assistance with the manuscript preparation.

References

1. J. L. Pons, R. Ceres and F. Pfeiffer, "Multifingered dextrous robotics hand design and control: A review," Robotica **17**, 661–674 (1999).
2. M. Ceccarelli, N. E. N. Rodriguez and G. Carbone, "Design and tests of a three finger hand with 1-DOF articulated fingers," Robotica **24**, 183–196 (2006).
3. G. Figliolini and P. Rea, "Overall design of Ca.U.M.Ha. robotic hand for harvesting horticulture products," Robotica **24**, 329–331 (2006).
4. A. Bicchi and V. Kumar, "Robotic Grasping and Contact: A Review," *Proceedings of the 2000 IEEE International Conference on Robotics and Automation*, San Francisco, CA (2000) pp. 348–353.
5. K. B. Shimoga, "Robot grasp synthesis algorithms: a survey," *Int. J. Robot. Res.* **15**(3), 230–266 (1996).
6. M. Kaneko, Y. Hino and T. Tsuji, "On Three phase of Achieving Envelope Grasps," *Proceedings of the 1997 IEEE International Conference on Robotics and Automation*, Albuquerque, NM (1997) pp. 835–390.
7. C.-S. Hwang, M. Takano and K. Sasaki, "Kinematics of grasping and manipulation of a B-Spline surface object by a multifingered robot hand," *J. Robot. Syst.* **16**(8), 445–460 (1999).
8. T. Miller, S. Knoop, H. I. Christensen and P. K. Allen, "Automatic Grasp Planning Using Shape Primitives," *Proceedings of the 2003 IEEE International Conference of Robotics and Automation*, Taipei, Taiwan (2003) pp. 1824–1829.
9. Y. Guan and H. Zhang, "Kinematic feasibility analysis of 3-D multifunctional grasps," *IEEE Trans. Robot. Autom.* **19**(3), 507–513 (2003).
10. Y. Li and N. S. Pollard, "A Shape Matching Algorithm for Synthesizing Humanlike Enveloping Grasps," *Proceedings of the 2005 IEEE-RAS International Conference on Humanoid Robots*, Tsukuba, Japan (2005) pp. 442–449.
11. E. Lopez-Damian, D. Sidobre and R. Alami, "A Grasp Planner Based on Inertial Properties," *Proceedings of the 2005 IEEE Internationals Conference on Robotics and Automation*, Barcelona, Spain (2005) pp. 754–759.