Finding enveloping grasps by matching continuous surfaces

Yi Li, Jean-Philippe Saut, Juan Cortés, Thierry Simeon, Daniel Sidobre

Finding Enveloping Grasps by Matching Continuous Surfaces

Yi Li[1,2], Jean-Philippe Saut[1,2], Juan Cortés[1,2], Thierry Siméon[1,2] and Daniel Sidobre[1,2]
[1]CNRS ; LAAS ; 7 avenue du colonel Roche, F-31077 Toulouse, France
[2]Université de Toulouse; UPS, INSA, INP, ISAE ; LAAS ; F-31077 Toulouse, France

Abstract— **This paper presents a new method to compute enveloping grasps with a multi-fingered robotic hand. The method is guided by the idea that a good grasp should maximize the contact surface between the held object and the hand's palmar surface.**

Starting from a given hand pregrasp configuration, the proposed method finds the hand poses that maximize this surface similarity. We use a surface descriptor that is based on a geodesic measure and on a continuous representation of the surfaces, unlike previous shape matching methods that rely on the Euclidean distance and/or discrete representation (*e.g.* random point set). Using geodesic contours to describe local surfaces enables us to detect details such as a handle or a thin part. Once the surface matching returns a set of hand poses, sorted by similarity, a second step is performed to adjust the hand configuration with the purpose of eliminating penetration of the object. Lastly, the grasp stability is tested in order to definitely validate the candidate grasps.

I. INTRODUCTION

Grasp planning can be simply stated as the problem of finding a penetration-free configuration such that the robot can hold the object stably in grasp. With this definition, many hypotheses concerning the object and the definition of the robot configuration can be proposed.

Bohg and Kragic [1] proposed three families of objects that lead to three different grasping approaches: Known objects, familiar objects, and unknown objects. In the first case, the robot already knows a set of grasps for the object. Once the object pose has been estimated, a valid grasp is selected depending on the context. Familiar objects can be grasped by adapting some grasps that were previously learnt for objects with similar shapes, whereas unknown objects require the computation of a new set of grasps.

The finality of grasp planning can also be classified by what has to be computed by the grasp planner: The hand configuration only (6D pose plus finger joint parameters), the whole robot (hand plus arm) configuration, or the robot configuration plus a valid approach motion.

The method proposed in this paper deals with unknown objects and the computation of valid hand configurations. It relies on the computation of the hand poses that give the best similarities between the object surfaces and the hand surfaces, where the similarity scores measure how well

the two surfaces fit together locally. We are interested in enveloping grasps (*i.e.* grasps with contacts on all the finger phalanges), not precision grasps (*i.e.* grasps with contacts only on fingertips). Enveloping grasps do not allow grasping of small parts as precision grasps do, but they may give more stable grasps while decrease the risk of damaging the object, because less pressure is required to hold it stably.

Some grasp planning techniques (*e.g.* [2], [3], [4]) explicitly compare the shapes of the object and the hand to find grasps. On the contrary, methods based on object segmentation [5], [6], [7], [8] can be seen as implicit shape matching methods, because the object's shape is not directly compared to the hand's shape, but compared to predefined primitive shapes (for which specific grasp planners have been implemented) in order to decompose the object into a collection of the primitive shapes.

The usual shape decompositions build a tree of convex primitive shapes, biasing the alignment of the grasp approach on the main axes of the convex parts of the object, not necessary leading to the best alignment if object or hand shapes are non-convex. Furthermore, the previously mentioned shape-matching-based methods rely on the Euclidean metric when computing distances between two surface points and/or discrete representation (*e.g.* representing local surfaces with randomly sampled surface points). Our method is based on contours around a surface point instead of randomly sampled surface points. Since these contours are clearly defined due to the employed geodesic measure, our method can better represent local surfaces. Consequently, the method presented in this paper offers the following contributions:

- The surface similarity computation is based on continuous surfaces, not on discrete points as in [2], [3], [4]. The use of the geodesic distance instead of the Euclidean distance allows us to correctly deal with object with small holes (*e.g.* a mug with handle).
- Surface matching and computation of the hand alignment are performed within the same algorithm unlike other techniques (*e.g.* [2], [3], [4]) that only use shape matching to find the best candidate in a grasp database.

The paper is organized as follows. Section II gives a brief overview of the related grasp planning methods. The overall principle of the method is then explained in Section III. Section IV details the surface similarity computation that is the base of the proposed method. Results of computed grasps for objects with various shapes are shown in Section VI.

II. RELATED WORK

Here is a brief overview of the grasp planners that are the most relevant to our method: Methods based on either implicit (object segmentation) or explicit (shape similarity computation) shape matching.

The idea behind the first group is to use techniques tailored for the simple shapes after the object segmentation has been performed, where the segmentation transforms the original shape of the object into a set of simpler and known shapes. Miller *et al.* [5] decompose the object into convex primitives (spheres, cylinders, cones, and boxes) and associate a pregrasp configuration of the hand to each primitive. Different approaches of the hand are then tested by closing the hand until contact, and the validity of the grasp is tested with the original shape using GraspIt! [9]. Other decompositions are possible (*e.g.* the ones based on superquadrics [6], simple boxes [8], or minimum volume bounding boxes [7]).

Instead of relying on primitives, Li *et al.* [2] start from a user-created database of human grasps that is representative of the different human grasp types. After a shape matching algorithm finds the hand shape that best matches the query object, another algorithm is used to find the alignment of the hand pose to the object shape. Kim's example-based method [3] does not perform object segmentation either. Instead, the object with the most similar shape to the query object is retrieved from the database. Secondly, a routine finds the point-to-point correspondence between the two objects. Finally, the correspondence is used to find the contact points on the query object from the contacts stored in the database. Goussous *et al.* [4] use a shape matching technique very similar to [2], albeit focus on virtual humans and also consider the effect of the upper body on the feasibility of the grasp.

III. METHOD OVERVIEW

The overview of the method is presented in Fig. 1. The input to the method is a pregrasp configuration of the hand. For now, it is defined manually. For instance, a set of hand pregrasps could be defined by an engineer for a given robotic hand during the design process.

As the hand model is composed of several bodies and not as a unique envelope, the first step of our method is to compute a valid mesh for the hand. A solution to this problem is given in the following subsection. After the triangulated mesh over the surface of the hand in the given configuration has been constructed, the hand mesh is compared to the object mesh in order to find the best alignments between them. If the hand penetrates the object at one of these top alignments, adjustments of both the finger configurations and the palm pose around the configuration computed in the previous step must be performed in order to find a penetration-free configuration. At last, the grasp stability is checked to validate the final configuration.

Although the method shares similar ideas as [2], there are two major differences between the two methods. First, the output of the surface matching in our method contains the grasp alignment unlike the one in [2] (*i.e.* the shape matching

in our method tells us not only the best hand preshape, but also how to grasp). The second major difference is in the employed surface matching method: Our method is based on continuous surface matching, not on discrete features.
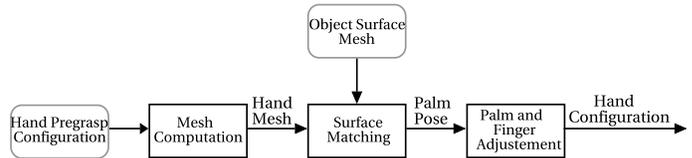

Fig. 1. Overview of the method.

A. Hand Mesh Computation

As a multibody system, the robot hand model is composed of several parts. Our goal is to obtain one manifold mesh for the whole hand or at least the parts of the hand that will be used to grasp the object.

Neither skeleton-based nor vertex-blending-based mesh deformation methods (*e.g.* [10]) are well-suited for a robotic hand as they will deform the mesh in the area close to the joints. Although the resulting meshes may still look realistic for character animation, they are not suitable for extracting the surface of the robotic hand. Instead, the marching cubes algorithm [11] was used to convert an isosurface to a triangulated mesh. The mesh is constructed as follows:

1) Build a grid of 3D points in the axis-aligned bounding box of the hand model.
2) For each point of the grid, set the value of the point to the distance between it and the closest point on the hand surface.
3) Apply the marching cubes algorithm [11] to get the surface of zero value as a set of triangles.

The original hand meshes we have are not well-defined, because they have multiple holes and consist of disjoint surface patches. Therefore, they can not be used directly to compute distances because the surface is not closed and hence their inside and outside are not clearly defined. Instead, a volumetric representation was chosen such that the bodies were modelled manually with superellipsoids. The radial Euclidean distance between a point and a superellipsoid, though not the exact distance, is easily computed from its algebraic expression [12]. Fig. 2 shows the result of the mesh computation for the Schunk Anthopomorphic Hand (SAH hand). Although the dorsal side of the surface differs from the original model, only the palmar side is meant to grasp the object.

Once the hand mesh was obtained, a mesh point, that is on the palmar surface and among the closest ones to the center of the middle finger's middle phalanx, was chosen manually as the center of the contact zone (Fig. 2).

IV. FINDING BEST HAND PLACEMENTS BY MATCHING CONTINUOUS SURFACES

Now that triangulated meshes of both the hand and the object have been obtained, multiple mesh points are sampled
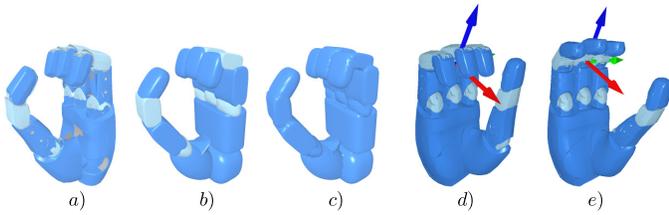
Fig. 2. Original model of the hand (a), remodeled with superellipsoids (b), and the final manifold mesh (c). The distance between two adjacent points of the grid was set to $2.2mm$. The first (d) and second (e) hand preshapes that were used for grasp computation in section VI. The frames of the hand surface patches — inputs to the surface matching algorithm — are also shown, where the red arrows represent the surface normals at the patch centers.
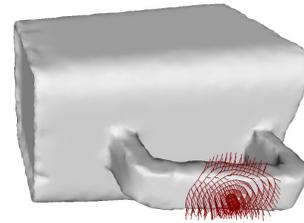

Fig. 3. 8 geodesic circles around one mesh point on a mug model, along with the normals of 50 sampled points on the geodesic circles. For clarity, the edges of the mesh are not shown here.

uniformly on the object and their local surface geometry characteristics are compared with the characteristics around a given mesh point on the hand (*i.e.* compare how similar the local surfaces around two mesh points are) in order to find the hand placements that maximize the contact surface between the hand and the object.

The following is the procedure for finding the best hand placements by matching continuous (local) surfaces. Firstly, geometry characteristics of the local surfaces around all sampled/given mesh points are encoded by geodesic circles (as done in [13]) in subsection IV-A. Secondly, a *quadratic shape descriptor* (QSD) [14] based filter (subsection IV-B) and a symmetry-based filter (subsection IV-C) are applied sequentially to extract the local surfaces on the object that are more likely to be the good grasping spots. Finally, the similarities (actually dissimilarities) between the object local surfaces and the hand local surface, which are used to rank the remaining local surfaces on the object, are computed in subsection IV-D using the normal variation encoded in the geodesic circles.

Note that the normal variation based dissimilarity measure can not distinguish between a concave local surface and a convex local surface, if these two surfaces have very similar shapes. Consequently, the QSD based filter has to be applied first to filter out concave/convex local surfaces from convex/concave ones. The QSD is also used to align the hand local surface to the object local surfaces in order to obtain hand placements.

A. Encoding Local Surface Geometry Characteristics

Geometry characteristics of the local surface around a point are encoded with M geodesic circles. On each geodesic circle, K points with equal distance between all neighboring points are sampled (Fig. 3). The geodesic circles are constructed by computing the geodesic distance from the point using the Fast Marching Method (FMM) applied to triangulated domains [15]. If the mesh has n mesh points, the computational complexity of the FMM is $O(n \log n)$.

B. The Quadratic Shape Descriptor Based Filter

The representation of a local surface is converted from geodesic circles to the QSD, which can easily distinguish

between concave local surfaces and convex ones. Given a mesh point $\mathbf{p}$ and the geodesic circles that represent the local surface around it, the first step of constructing the QSD is to transform all sampled points on the geodesic circles to the coordinates (u, v, w) on a second coordinate system spanned by three orthogonal unit vectors $\mathbf{i}$, $\mathbf{j}$, and $\mathbf{k}$ as follows:

$$\mathbf{r}(u,v) = u\mathbf{i} + v\mathbf{j} + w(u,v)\mathbf{k}, \qquad (1)$$

where $\mathbf{k}$ is aligned with the surface normal $\mathbf{n}$ at $\mathbf{p}$, while $\mathbf{i}$ and $\mathbf{j}$ span the plane tangent to $\mathbf{n}$ at $\mathbf{p}$. Note that when we refer to a surface normal in this section and the normal is on the hand surface, we have in mind the negated surface normal.

The local range curvatures and directions — represented by $(\kappa_{min}, \lambda_{min})$ and $(\kappa_{max}, \lambda_{max})$, where $\kappa_{min} \leq \kappa_{max}$ — of the local surface are simply the eigenvalues and the eigenvectors, respectively, of the local Hessian matrix $\mathbf{H}(u, v, w)$. The detail of calculating the elements of $\mathbf{H}$ can be found in [14]. The degree of concavity or convexity of a local surface patch is then given by the shape index Si, which is defined as

$$Si = -\frac{2}{\pi} \arctan \frac{\kappa_{max} + \kappa_{min}}{\kappa_{max} - \kappa_{min}}. \qquad (2)$$

Since we are looking for surface patches on the object that are similar to a certain surface patch on the hand, a surface patch on the object with shape index Si_o is pruned if $|Si_o - Si_h| > \Delta S$, where Si_h is the shape index of the surface patch on the hand and ΔS is a shape filtering threshold. The threshold ΔS is set empirically to 0.3 in [14].

For description of how to utilize the local range directions to superimpose a surface patch from the hand to a surface patch from the object, we refer to [14].

C. The Symmetry Based Filter

We propose the hypothesis that S_o — a surface patch on an object around point $\mathbf{p}_o$ — that is not symmetrical with respect to planes yz and zx of the right-handed orthogonal coordinate system, which is established by the rows of

$$\mathbf{R}_o = \begin{bmatrix} \lambda_{min,o} \\ \lambda_{max,o} \\ \mathbf{n}_o \end{bmatrix}, \qquad (3)$$

is not a good spot for grasping and hence should be discarded. In (3), $\mathbf{n}_o$ represents the surface normal at $\mathbf{p}_o$, while $\lambda_{min,o}$ and $\lambda_{max,o}$ represent the two local range directions on S_o.

The coordinate system is constructed such that: The origin is at the mesh point $\mathbf{p}_o$, its z axis is aligned with the surface normal $\mathbf{n}_o$ at $\mathbf{p}_o$, and its x and y axes are determined by the local range directions $\lambda_{min,o}$ and $\lambda_{max,o}$, respectively. The orthogonal coordinate system corresponds to (3) is shown in Fig. 4.
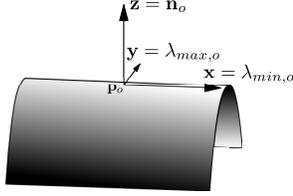

Fig. 4. The right-handed orthogonal coordinate system constructed from the surface normal $\mathbf{n}_o$ at the mesh point $\mathbf{p}_o$ and the local range directions $\lambda_{min,o}$ and $\lambda_{max,o}$.

The measure of a surface patch's reflective symmetry — the symmetry distance Sd — is defined as the L_2-difference between the geodesic circles over the surface patch and their reflections (inspired by [16]):

$$Sd(P, \gamma) = \sum_{\mathbf{p} \in P} \min_{\mathbf{q} \in \gamma_{yz}(P)} \|\mathbf{p} - \mathbf{q}\| + \min_{\mathbf{q} \in \gamma_{zx}(P)} \|\mathbf{p} - \mathbf{q}\|, \quad (4)$$

where P is a point set of all sampled points on the geodesic circles and γ is a reflection.

Once the symmetry distances of all surface patches on the object have been computed, the one-third surface patches that are the most symmetrical are kept.

D. The Normal Variation Based Dissimilarity Measure

In this final step, the similarities (actually dissimilarities) between the hand surface patch and the object surface patches that still remain after the first two steps are computed using normal variation encoded in the geodesic circles [13].

The dissimilarity measure Ds between a surface patch around the point $\mathbf{p}_h$ on the object and a surface patch around the point $\mathbf{p}_o$ on the hand is defined as

$$Ds = \sum_{m=1}^{M} \min_{i=1}^{K} \sum_{k=1}^{K} (\mathbf{n}_{h,m,k} \cdot \mathbf{n}_h - \mathbf{n}_{o,m,(k+i) \bmod K} \cdot \mathbf{n}_o)^2,$$
$$(5)$$

where M is the number of the geodesic circles, K is the number of all sampled points on each geodesic circle, $\mathbf{n}_h$ and $\mathbf{n}_o$ are the surface normals at the mesh points $\mathbf{p}_h$ and $\mathbf{p}_o$, respectively, $\mathbf{n}_{h,m,k}$ is the normal at the kth point on the mth geodesic circle from $\mathbf{p}_h$, $\mathbf{n}_{o,m,k}$ is the normal at the kth point on the mth geodesic circle from $\mathbf{p}_o$, and $\bmod$ is the modulo operator. Essentially, each geodesic circle on the object surface is "rotated" to find the best match with the corresponding geodesic circle on the hand

surface (*i.e.* to minimize their contribution to Ds). Observe that the "rotation" of each geodesic circle is independent of the "rotations" of other geodesic circles in (5) as compared to the similar equation in [13], which is more suitable for comparisons of local surface patches on a single object.

V. FINDING THE FINAL HAND CONFIGURATION

A. Finding a Penetration-Free Hand Configuration

The result of the computations in the previous section is a list of hand placements with respect to the object frame, where the list is ordered from the best placement (*i.e.* the one that gives the best surface match) to the worst one. If the hand is placed with the pregrasp configuration in one of the candidate placements, there might be penetrations between the hand and the object, and hence the finger joint parameters and possibly even the palm pose must be adjusted.

Some techniques have been proposed for finding a collision-free configuration starting from a configuration in collision [17], [18], but they were only used for free-flying robots and can not be easily extended to a hand with dozens of DOFs. Furthermore, a generic method is not necessary as our application is very specific, and in particular, collision (*i.e.* contacts) between the hand and the object is actually essential, although it must be penetration-free. We implemented a simple incremental approach to compute a penetration-free configuration. First, a set of configurations are sampled in the neighborhood (in the 6D space) of the initial palm configuration and the penetration-free configuration that is the closest to the initial one is selected. Then, from proximal to distal phalanges, the fingers are incrementally opened/closed until they are either in contact with the object but with zero-depth penetration or collision-free.

B. Checking Grasp Stability

With the final hand configuration in hand, the grasp stability must be checked as the last step. Among many grasp quality measures [19] that can be used to check and rank a grasp's stability, the radius of the largest ball, that is centered at the origin of the wrench space and fully contained in the convex hull of the primitive wrenches applied at the contacts [20], was chosen as the stability measure. The grasp is considered to be stable if the radius is non-null.

VI. RESULTS

To validate the proposed method, we implemented and tested it on three object shapes (Fig. 5) that were chosen for their complexities (important concavities, loops). The size of the mug was increased such that any finger of the robotic hand (except the thumb) can go through the handle's loop. The other two shapes were also roughly scaled to adapt to the dimensions of the hand (with a length of about $30cm$).

First, the surface matching algorithm was applied on two different hand configurations: An almost fully closed one and a more open one as shown in Fig. 2. 1000 points were sampled on the surfaces of the elephant and the mug, while 700 points were sampled on the surface of the horse. On a workstation with 3.0GHz Opteron processors, it took
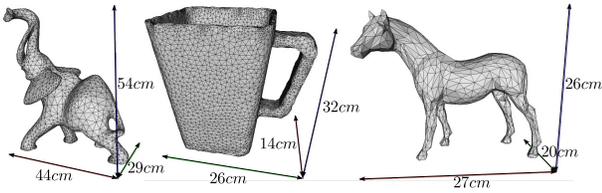
Fig. 5. The three shapes we used to validate our method. The elephant was taken from CGAL demos [21], the mug is the reconstruction of a real object of ours, and the horse was fetched from the *Large Geometric Models Archive* at Georgia Tech. Note that the scales of the three figures are different.

24, 142, and 512 seconds to compute the grasp candidates (*i.e.* the hand poses) for the horse, the elephant, and the mug, respectively. The surface matching algorithm runs in $O(mn \log n)$ time, where m is the number of the sampled mesh points and n is the total number of mesh points. Next, the computation of the final configuration (*i.e.* adjustments of the palm and the finger joints) for each hand pose takes about 1 second. The contact points are computed as the points on the object's surface (after a discretizing phase) that are closer than $6mm$ to the superquadric modelling of the hand surface. Again, the distance is the radial Euclidean distance, that is slightly greater than the exact distance and hence the contacts are actually closer than $6mm$.

A. The First Hand Preshape

The grasp candidates resulting from the surface matching algorithm are shown in Fig. 6.
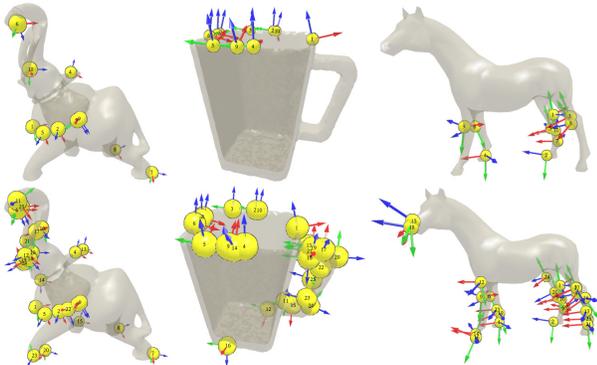

Fig. 6. The top 10 and top 25 grasp candidates for the first hand preshape are shown in the top three subfigures and the bottom three subfigures, respectively.

To grasp the elephant, the parts that were found to be the best for grasping are, in decreasing ranking, its ears, trunk, and legs. Moreover, valid grasps (*i.e.* penetration-free and stable) were found for all three parts as shown in Fig. 7, where the found contact points are drawn in yellow, while the contact normals are drawn in green. Some of the grasps (*e.g.* the top-right one in Fig. 7) may look like fingertip grasps, but some contacts are actually located on the middle or proximal phalanx surfaces. Furthermore, unlike the human

hands, the robotic hand consists of rigid parts, and hence the possibilities of grasping with the whole palmar surface are rather limited when the objects have complex shapes. For the mug, the best parts for grasping are, in decreasing ranking, its rim and its handle. Valid grasps were found all around the mug's rim and on its handle (Fig. 8). For the horse, the valid grasps were found on its legs as shown in Fig. 9.
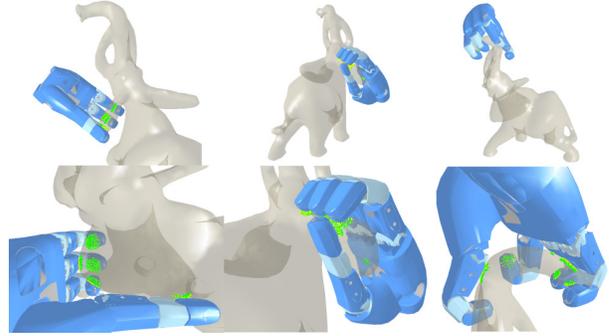

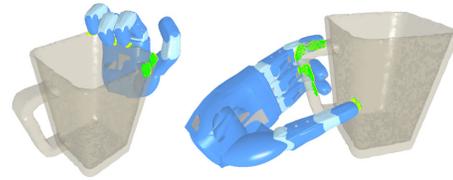Fig. 7. 6 stable grasps for the elephant obtained with the first hand preshape.

Fig. 8. 2 stable grasps for the mug obtained with the first hand preshape.
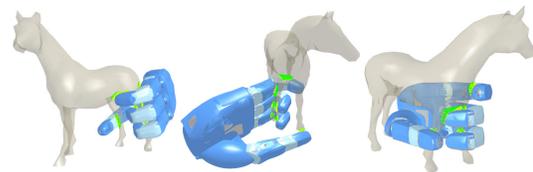

Fig. 9. 3 stable grasps for the horse obtained with the first hand preshape.

B. The Second Hand Preshape

The second hand preshape has a wider opening and its palmar surface is flatter. Consequently, the best candidates are the flatter parts of the objects (Fig. 10).

The surface matching does not favor anymore thin elements such as the legs of the animals or the mug's handle. However, grasping the flatter parts is not always possible due to the restricted opening of the SAH hand's thumb. As shown in Fig. 11, many of the best candidates give good surface matching but unstable grasps (*i.e.* with all non-thumb fingers around these flatter parts, the hand can not be opened wide enough to hold the objects in enveloping grasps, although this flatter hand preshape could be useful for finding dual-handed
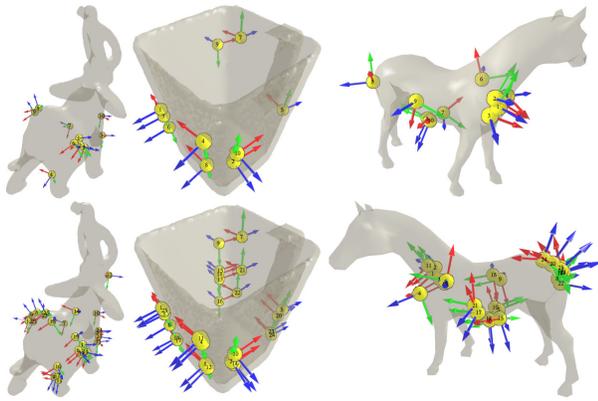
Fig. 10. The top 10 and top 25 grasp candidates for the second hand preshape are shown in the top three subfigures and the bottom three subfigures, respectively.

grasps). For instance, no valid grasp was found for the mug and the hand preshape, and only one valid grasp was found for the elephant (on its longer front leg). However, multiple valid grasps were found for the horse (Fig. 12) because it is not as wide as the mug and the elephant.
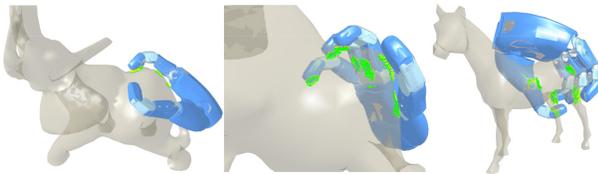

Fig. 11. With the second hand preshape, some of the best candidates did not lead to stable grasps and were discarded, because the hand can not be opened wide enough.
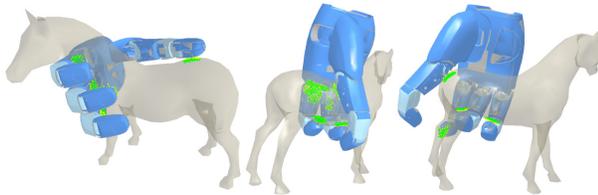

Fig. 12. 3 stable grasps for the horse obtained with the second hand preshape.

VII. CONCLUSIONS AND FUTURE WORKS

We have presented a new method for planning enveloping grasps with a robotic multi-fingered hand. Our method relies on a matching algorithm for continuous surfaces to find the best alignments between the surface of the hand in a given pregrasp configuration and the surface of the object. Experiments on several objects with complex shapes have shown that the method indeed finds good alignments.

There are several ways to improve the proposed method. The finger adjustment phase could be improved to find more

solutions, and it would be desirable to reason with multiple hand surface patches instead of a single one so that the distinct role of the thumb, with respect to the opposite fingers, can be taken into account.

REFERENCES

[1] J. Bohg and D. Kragic, "Learning grasping points with shape context," *Robotics and Autonomous Systems*, vol. 58, no. 4, pp. 362–377, 2010.

[2] Y. Li, J. L. Fu, and N. S. Pollard, "Data-driven grasp synthesis using shape matching and task-based pruning," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 4, pp. 732–747, 2007.

[3] J. Kim and T. Lozano-Pérez, "Example-based grasp adaptation," Master's thesis, Massachusetts Institute of Technology, 2007.

[4] F. Goussous, T. Marler, and K. Abdel-Malek, "A new methodology for human grasp prediction," *Trans. Sys. Man Cyber. Part A*, vol. 39, no. 2, pp. 369–380, 2009.

[5] A. T. Miller, S. Knoop, H. I. Christensen, and P. K. Allen, "Automatic grasp planning using shape primitives," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 2, Sept. 2003, pp. 1824–1829.

[6] C. Goldfeder, P. K. Allen, C. Lackner, and R. Pelossof, "Grasp planning via decomposition trees," in *Proceedings of IEEE International Conference on Robotics and Automation*, Apr. 2007, pp. 4679–4684.

[7] K. Huebner, S. Ruthotto, and D. Kragic, "Minimum volume bounding box decomposition for shape approximation in robot grasping," in *Proceedings of IEEE International Conference on Robotics and Automation*, May 2008, pp. 1628–1633.

[8] K. Harada, K. Kaneko, and F. Kanehiro, "Fast grasp planning for hand/arm systems based on convex model," in *Proceedings of IEEE International Conference on Robotics and Automation*, May 2008, pp. 1162–1168.

[9] A. T. Miller and P. K. Allen, "Graspit! a versatile simulator for robotic grasping," *IEEE Robotics & Automation Magazine*, vol. 11, no. 4, pp. 110–122, Dec. 2004.

[10] L. Kavan, S. Collins, J. Zara, and C. O'Sullivan, "Geometric skinning with approximate dual quaternion blending," *ACM Trans. Graph.*, vol. 27, no. 4, pp. 105:1–105:23, 2008.

[11] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, 1987.

[12] A. Jaklič, A. Leonardis, and F. Solina, *Segmentation and Recovery of Superquadrics*, ser. Computational imaging and vision. Kluwer, 2000, vol. 20.

[13] Y. Sun, J. Paik, A. Koschan, D. L. Page, and M. A. Abidi, "Point fingerprint: a new 3-D object representation scheme," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 33, no. 4, pp. 1083–4419, Aug. 2003.

[14] B. B. Goldman and W. T. Wipke, "Quadratic shape descriptors. 1. rapid superposition of dissimilar molecules using geometrically invariant surface descriptors," *J. Chem. Inf. Comput. Sci.*, vol. 40, no. 3, pp. 644–658, 2000.

[15] R. Kimmel and J. A. Sethian, "Computing geodesic paths on manifolds," in *Proceedings of Natl. Acad. Sci. USA*, 1998, pp. 8431–8435.

[16] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A reflective symmetry descriptor for 3D models," *Algorithmica*, vol. 38, no. 1, Oct. 2003.

[17] O. B. Bayazit, G. Song, and N. M. Amato, "Enhancing randomized motion planners: Exploring with haptic hints," *Autonomous Robots*, vol. 10, no. 2, pp. 163–174, 2001.

[18] E. Ferre and J.-P. Laumond, "An iterative diffusion algorithm for part disassembly," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3, 2004, pp. 3149–3154.

[19] R. Suarez, M. Roa, and J. Cornella, "Grasp quality measures," in *Universitat Politecnica de Catalunya (UPC), Technical Report*, 2006.

[20] C. Ferrari and J. Canny, "Planning optimal grasps," *Robotics and Automation, 1992. Proceedings., 1992 IEEE International Conference on*, pp. 2290–2295 vol.3, May 1992.

[21] CGAL, Computational Geometry Algorithms Library. [Online]. Available: http://www.cgal.org