

Interaction skills for a coat-check robot: identifying and handling the boundary components of clothes

Lukas Twardon and Helge Ritter

Abstract—Identifying the relevant functional degrees of freedom is a key prerequisite for the proper handling of everyday objects. Recognizing and exploiting these degrees of freedom in the context of non-rigid objects poses challenges that are significantly different from the rigid case. As a major generic subtask, we consider the identification and exploitation of boundary components during clothes manipulation, combining RGBD vision with uni- and bi-manual handling through a robot. Specifically, we present a novel graph-based approach to detecting boundary components by extracting closed contours from depth images. Based on that, we suggest a planner minimizing a heuristic energy function for an optimal grasp pose of a robot hand around the boundary of a garment. We demonstrate the effectiveness of the approach in interactive perception and regrasping experiments with a dual arm and two attached anthropomorphic hands. Furthermore, we show how to make use of these capabilities to implement a basic skill for a coat-check robot: hanging up a knit cap on a hat-stand.

I. INTRODUCTION

Future robots promise to take a lot of tedious work out of humans' hands. However, present-day robots still struggle with tasks humans accomplish quite naturally. For example, coat-check attendants are able to deal with hundreds of different garments everyday. They grasp the items, identify the important parts and hang them up without any problem. Although robots are getting better and better at rigid object grasping and manipulation, robotic handling of such complex deformable objects still poses a major challenge.

We believe that this reflects the circumstance that current robots excel at extracting and exploiting precise metric information, but are still very limited when it comes to recognizing and interacting with more qualitative structures often aptly characterized at the level of *topology* instead of metrics. For highly deformable objects, metric features cease to be good invariants, and more abstract topological features must be used to guide interaction reliably. This is particularly evident in the case of clothes, whose handling typically requires the identification of boundaries along with their topological patterns in order to be able to recognize or change the orientation of a garment, or to properly interact with structures such as openings.

This outlines the motivation of the current paper which is to develop methods that allow "affordances" (i.e., the perceivable action possibilities [1]) to be grounded in everyday situations centered around the handling of clothes. Our focus of interest is in shaping perception and manual actions for basic interaction primitives such as folding, unfolding,

The authors are with the Neuroinformatics Group, EXC Cognitive Interaction Technology (CITEC), Bielefeld University, Germany. {ltwardon, helge}@techfak.uni-bielefeld.de

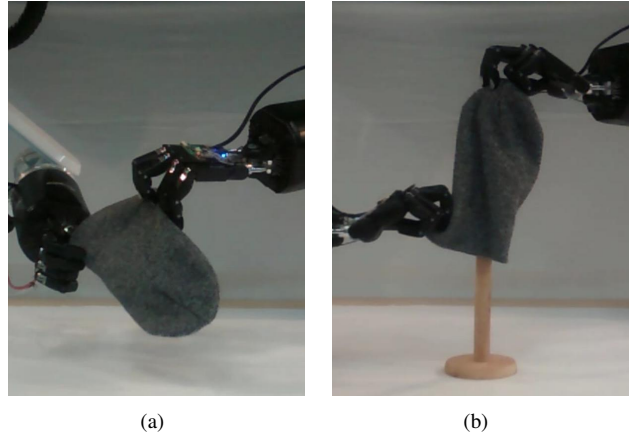


Fig. 1. A robot with two dexterous hands manipulating a piece of clothing. (a) Regrasping a knit cap around its boundary. (b) Exploiting the boundary grasp for hanging up the cap.

reaching into, hanging up, or pulling over (a destination object or a human body part) for clothes or similar objects. From a basic research perspective, we view this as an entry point to more general topology-based representations and to elucidating the connection between affordances and topology. We aim to endow a robot with some higher level cognition about interaction patterns with deformable objects for which metric geometry is burdened with way too much detail and lacks the necessary invariance.

Within the scope of the present paper, we propose algorithms which can provide robots with some basic skills of a coat-check attendant. We focus on the openings of garments which can be described topologically by their boundary components. The boundary components appear as closed contours and can be used to define a suitable grasp pose (Fig. 1(a)). As can be seen in Fig. 1(b), our robot is able to exploit the boundary grasp for such a complex task as hanging up a piece of clothing on a hat-stand. Although we restrict ourselves to a garment with just one boundary component, we suppose that the general approach is applicable to various types of clothing.

The key contribution of our current work is a method for explicitly extracting topological structures, namely boundary components, from depth images. This is opposed to the widely used approach of detecting fragmentary metric features. Another contribution is that we show how the algorithm can be implemented efficiently, and applied to an interactive clothes perception and regrasping task in a single-view setup in the presence of outliers and noise.

II. RELATED WORK

Katz and Brock [2] have suggested that complex manipulation tasks can be accomplished by factorization of high-dimensional state spaces into lower-dimensional subspaces. They make use of this approach in the context of articulated objects. Pokorny et al. [3] share with us the idea that finding holes or openings is an important topological factor for grasping. However, deformable objects usually have a much higher-dimensional configuration space. Therefore, Lui and Saxena [4] have presented a learning approach to perceiving the knot structure of a rope for untying it.

A couple of research projects have aimed to enable robots to handle clothes. As an example, the reader is referred to the CloPeMa (Clothes Perception and Manipulation) project which has resulted in methods for solving complex manipulation tasks such as heuristic-based flattening [5], data-driven folding [6], and unfolding of clothes using a polygonal model [7].

Researchers from UC Berkeley intended to have an entire laundry cycle done by a robot. For this purpose, they studied gravity-based cloth folding [8], recognizing the configuration of a garment using parametrized shape models [9] as well as grasp point selection based on geometric cues like borders and corners [10]. Finding optimal grasp points has also been addressed by Ramisa et al. [11] who proposed a measure of wrinkledness computed from the distribution of normal directions in a point cloud, and by Yamazaki [12] who presented an approach to learning graspable hem elements.

Another common approach is to create a mesh model of the garment. Besides depth data, fiducial markers [13] as well as prior knowledge and cues from strategic observation [14] can be helpful. Furthermore, physics-based simulation allows the dynamic properties of deformable objects like a piece of paper [15] or clothing [16] to be modeled.

Willimon et al. characterize their work on robotic laundry handling as an application of the interactive perception paradigm [17]. They made use of this idea for unfolding [18] and classification [19] of clothing. Moreover, they proposed an energy minimization approach to markerless pose estimation of deformable surfaces [20]. A hierarchical method for classifying and estimating the pose of deformable objects has been presented by Li et al. [21].

Tamei et al. [22] have intended to develop a clothing assistance robot. Contrary to most of the works on clothes handling mentioned so far, their approach is explicitly topology-based. A reinforcement learning algorithm permits the robot to put a mannequin’s head into a T-shirt. The reward function is defined using so-called topology coordinates. However, no robot vision is involved. Instead, the T-shirt’s neck is equipped with markers and tracked using a motion capturing system.

III. CONCEPTS

A. Topological characterization of clothes

Euclidean geometry is about distances between points, angles between lines as well as the shape, size and position

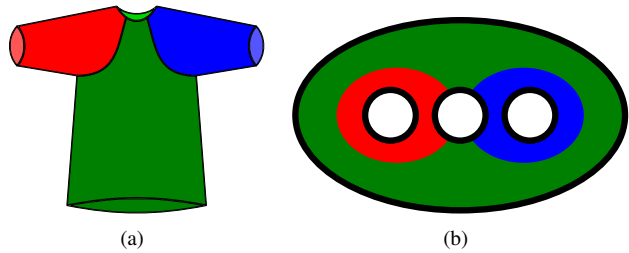


Fig. 2. (a) A T-shirt consisting of three sleeve components. (b) Top view on a flattened T-shirt considered as a surface with four boundary components depicted in black.

of objects. All these parameters change continuously when clothes are deformed in a typical way. Topology, in contrast, is concerned with object properties that remain invariant under continuous deformation. It is, therefore, well-suited for describing the spatial properties of clothes in a simple but correct way.

Consider the T-shirt depicted in Fig. 2(a). It topologically consists of a sleeve (green) with two smaller sleeves (red and blue) attached to it. The entire T-shirt comprising both the inside and the outside can be regarded as one locally euclidean surface (2-manifold) without boundary. Considering the sleeves as stretched 1-handles or tori, the number of sleeves is a topological invariant referred to as genus.

An alternative way to think about a (flattened) T-shirt is as a genus zero manifold with three interior boundary components, each corresponding to one of the sleeves, and one exterior boundary component (Fig. 2(b)). Most everyday articles of clothing can be topologically described by their openings in this manner. Note, however, that exceptions exist. A breast pocket, considered as a surface sewed on another surface, is not a manifold because it has edges with more than two neighboring surfaces and, hence, is not locally euclidean. Apart from that, assuming an orientable genus zero 2-manifold, the boundary components completely define the topology of a piece of clothing.

B. Interactive perception

One major reason why many robot vision researchers have preferred methods based on sparse metric features over topological representations is the assumption that image processing is always done before manipulation. In fact, identifying topological structures can be difficult in static environments. Detecting the boundary components of a wrinkled item of clothing lying on top of a table is almost impossible because of misleading wrinkles and occlusion by other parts of the garment.

The interactive perception paradigm [17] [18] [19] suggests that the weaknesses of traditional sense-plan-act architectures should be overcome by closely coupling manipulation and perception. Changing the configuration of an object may reveal important information which would be inaccessible otherwise. In section V-B.2, we propose a heuristic-based approach to picking up a knit cap from the tabletop in such a way that the robot can easily observe the cap.



Fig. 3. Intermediate results of the boundary detection algorithm. (a) Color image of a knit cap held by a human hand. (b) Depth image. (c) Edge image. (d) Skeletonized edge image. (e) Contour graph and selected cycle (depicted in red).

C. The boundary grasp

We believe that the topology of a garment is closely related to its affordances. The openings provide a link between the appearance of a piece of clothing and the meaningful manual actions. One affordance of most garments is grasping around the boundary of an opening. Humans perform boundary grasps during many daily tasks such as dressing, making the bed or doing the laundry. We argue that an optimal grasp pose is chosen taking into account the shape of the boundary contour, perceptual reliability and stability, as well as grasp comfort. A heuristic energy minimization approach to finding a suitable boundary grasp pose is presented in section IV-D.

IV. ALGORITHMS

A. Building a graph representation from a depth image

In section III-A, boundary components have been identified as the most important topological features of clothes. Therefore, we propose a graph-based approach to finding those closed contours which represent the boundary components. We start with the depth image of a garment (Fig. 3(b)), one opening being visible to the sensor. The color image is not very helpful since clothing with sparse texture may appear as a homogeneous single-colored area providing no additional information (Fig. 3(a)). We convert the depth image to a graph representation in three steps as follows:

- 1) Normal-based edge detection
- 2) Thinning/Skeletonization
- 3) Contour graph creation

For normal-based edge detection, we use the highly optimized implementation from Ückermann et al. [23]. The basic objective of the method is to separate edges from smoothly curved areas. We begin by temporally and spatially smoothing the depth image. This noise reduction step is crucial for the topological analysis, which we will conduct later on, to be stable. Then, for each pixel, a surface normal is estimated from the plane spanned by three neighboring points. Finally, the scalar products with the normals in an 8-neighborhood are averaged before binarizing the obtained angle image employing a threshold. The edge detection result is depicted in Fig. 3(c).

Topology-preserving thinning of shapes in a binary image is often referred to as skeletonization. Our skeletonization procedure is based on Zhang and Suen’s parallel algorithm described in [24] which we reimplemented using OpenCL for parallelization. The idea of the algorithm is to iteratively remove boundary points preserving the end points and pixel

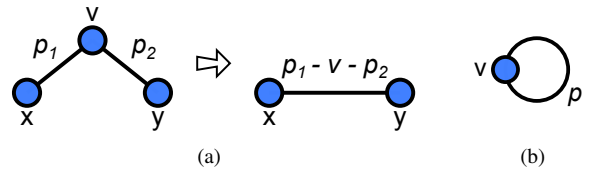


Fig. 4. Basic idea of the path graph reduction algorithm from [25]. (a) Removing node v and adding a new edge concatenating the paths p_1 and p_2 of two adjacent edges. (b) A simple cycle has been found if there is a loop in the path.

connectivity. To this end, two subiterations are conducted. Southeast points are deleted in a first pass before deleting northwest points in a second pass. The result is an 8-connected "skeleton" (Fig. 3(d)).

Converting the thinned edge image to a contour graph is done in a two-step process. First, we find nodes, i.e., points with at least three neighbors in the image, and insert them into the graph. Points with exactly one neighbor are free ends of a contour and, hence, cannot be part of a simple cycle (see section IV-B). Also, we completely miss closed contours without any junction. But in reality, boundary components of clothes never occur in this isolated form. In a second step, we find the contour edges linking the nodes. For this purpose, we create an edge for each neighbor of a node. Then, we follow the contour line until another node is found. If no edge representing the same contour exists, we insert the edge into the graph. Strictly speaking, the resulting representation (Fig. 3(e)) is not a graph but a multigraph, i.e., two nodes may be linked by multiple edges which is forbidden in conventional graphs.

B. Finding simple cycles in the graph

The visible boundary components are represented as cycles in the contour graph, but not every cycle is a boundary component. Some definitions are needed for a graph-theoretic specification of the following algorithm: A *cycle* is a walk through the graph starting and ending at the same node. *Simple cycles* are defined as those closed walks which do not contain any repetitions of edges or nodes, except the starting and ending node being the same. *Cycle chords* are edges which are not part of the cycle but connect nodes which are part of the cycle. Cycles without any chords are called *chordless cycles*.

There are efficient methods for detecting all chordless simple cycles. Unfortunately, the requested cycles are not necessarily chordless. For example, the red cycle in Fig. 3(e) has a chord resulting from a wrinkle in the inner part of the cap. The bottom line is that all simple cycles in the graph must be detected because they are possible boundary component candidates.

The maximum number of simple cycles in a graph grows exponentially with the number of edges. However, many of the simple cycles do not reflect any boundary component but result from background edges or wrinkles. We describe how we deal with these false positives in section IV-C. Nevertheless, all simple cycles can be found by applying

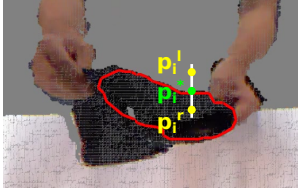


Fig. 5. Intuition of the convexity criterion: Contour points stick out compared to points to the left and to the right of the boundary contour.

a path graph reduction method. We adapted Hanser et al.’s algorithm [25] which returns the simple cycles as a side product of iteratively reducing the graph. The procedure is as follows: While there are nodes in the graph, the node v of minimum degree is selected. For each pair of edges $x - p_1 - v$ and $v - p_2 - y$ with disjoint paths p_1 and p_2 , a new edge $x - p_1 - v - p_2 - y$ is created (Fig. 4(a)). If an edge’s starting node is the same as its ending node ($v - p - v$), we have found a simple cycle (Fig. 4(b)). Then, node v and all its edges are removed from the graph and the next node is selected. The algorithm terminates when there are no nodes left in the graph. We give up if v has too many edges, i.e., more than a threshold $\Theta = 250$. In practice, having reduced the search space beforehand (see section IV-C), this only occurs when the garment is quickly moved which may lead to artificial edges caused by temporal smoothing.

C. From simple cycles to boundary components

The graph-theoretic approach raises three issues: how to avoid a drop in performance caused by exponential growth of the number of simple cycles, how to eliminate false positives (i.e., simple cycles that do not originate from a real boundary component), and how to transform 2D cycles to 3D closed contours. We tackle the first two issues by drastically reducing the search space before applying the cycle finding algorithm. Contrary to the algorithms described so far, the following steps require calibrated point clouds rather than raw depth images.

For one thing, we only consider a region of interest around the estimated object position in the point cloud, omitting cycles that occur in the background of the scene. Estimating the current position of the garment can be done by employing any traditional color or texture based tracking technique on a depth registered color image. Alternatively, in interactive perception scenarios like ours, it can be derived from the position of the robot end effector.

For another, we add edges to the contour graph only if they match *at least one* out of a list of criteria. The most important one is the *convexity criterion*: We find the pixel p_i^* with the minimum depth value in a small range around each contour point p_i accounting for minor deviations from the real contour caused by skeletonization. Then, after having computed the orthogonal vector of the contour in p_i , we compare the depth value of p_i^* with those of two pixels p_i^l and p_i^r to the left and to the right of the contour (Fig. 5). The contour point p_i fulfills the criterion if it is closer to the depth sensor than p_i^l and p_i^r . The edge matches the criterion

if the ratio of such points exceeds a threshold. Additionally, we introduce the *contour length criterion* in order to prevent very short edges from breaking the whole cycle. A contour edge matches the criterion if its length is below a threshold. Further criteria helping to distinguish boundary edges from non-boundary edges can be easily added.

All remaining simple cycles are evaluated by the criteria matching quality of their edges. Here, we make use of our prior knowledge of the type of clothing. For example, a knit cap has only one boundary component of a roughly known length. Consequently, we select the simple cycle with a contour length in a given range whose contour points match the convexity criterion better than a threshold and better than any other simple cycle. Thus, we obtain no more than one closed contour represented by a sequence of depth image pixels.

We reduce redundancy by approximating the contour by a closed polygonal chain using the curvature-based corner detector described in [26]. Afterwards, we convert the polygon to a 3D representation as follows: We, again, find the minimum depth pixel p_i^* in a small range around each polygon corner p_i . Then, we intersect the view ray of the depth camera at p_i with a plane parallel to the image plane and going through the point corresponding to p_i^* in the structured point cloud. The intersection point is added to the 3D contour.

D. From perceiving boundaries to regrasping garments

A regrasping pose can be determined from the boundary contour of a garment. However, some parts of a boundary component are more suitable for grasping than others. Good segments to grasp (depicted as blue curve in Fig. 6(b)) satisfy the following criteria:

- **Low curvature:** Segments with high curvature are likely to be corners or creased parts of the boundary. Low curvature segments generally provide more space for reaching into the opening.
- **Stability:** Segment hypotheses should not jump too much between successive frames. Instead, segments whose position remains stable over time are preferred to unstable segments.
- **Reachability:** The left hand should not try to grasp around a segment on the right side of the boundary and vice versa. Therefore, segments which match a desired direction well are preferred.
- **Good segment length:** Graspable segments should be of a given length L greater than the breadth of the fingers reaching into the opening. In our case, $L = 5\text{cm}$ has been a good choice.

Given a boundary component represented by a sequence of 3D points, we iterate through all segments of length L and find the segment with points p_1, p_2, \dots, p_N which minimizes an overall energy term E . The procedure is repeated for several consecutive frames. E is a weighted sum of three terms E_{curv} , E_{dist} and E_{dir} :

$$E = \alpha E_{curv} + \beta E_{dist} + \gamma E_{dir} \quad (1)$$

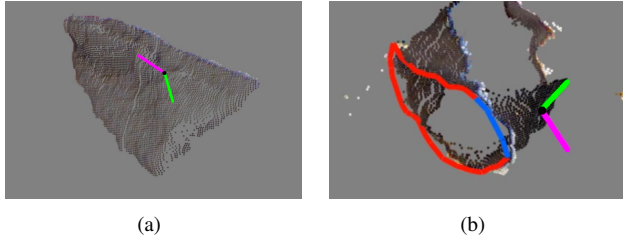


Fig. 6. Point clouds of a knit cap and two different grasp poses. Grasp points are depicted as black dots. Approach and orientation vectors are depicted in green and pink, respectively. (a) Pick-up grasp pose suitable for visual observation of the garment. (b) Boundary component detection technique with different garments. The results are depicted in red. (a) Three examples of successful detections. (b) Three examples of detection failures.

The parameters α , β and γ control the relative influence of the terms. *Curvature* is approximated as follows:

$$E_{curv} = \frac{L}{|p_1 - p_N|}. \quad (2)$$

E_{dist} penalizes sudden changes of position (enforces *stability*):

$$E_{dist} = \frac{1}{NL} \sum_{i=1}^N |p_i(t) - p_i(t-1)|. \quad (3)$$

E_{dir} (enforces *reachability*) is defined as

$$E_{dir} = - \left\langle a \frac{p_1 - p_N}{|p_1 - p_N|}, v_{des} \right\rangle, \quad (4)$$

where $v_{des} = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$ if the left hand is used or

$v_{des} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix}$ if the right hand is used,

and $a = -1$ if the points are given in clockwise order or $a = 1$ if they are given in counterclockwise order. The direction of rotation a of the boundary points with coordinates $(x_1, y_1, z_1), \dots, (x_M, y_M, z_M)$ is determined by employing the so-called shoelace formula:

$$a = \text{sgn} \left(\sum_{i=1}^{M-1} x_i z_{i+1} + x_M z_1 - \sum_{i=1}^{M-1} x_{i+1} z_i - x_1 z_M \right) \quad (5)$$

We compute the actual regrasping pose based on the optimal segment. However, all contour points are used to fit a plane to the boundary employing a planar RANSAC algorithm [27]. The approach direction is set to be the normal vector of the plane. We limit the approach vector to deviate from the tabletop plane by at most 20° in order to obtain a relaxed elbow pose. For the orientation vector, we project both end points of the optimal segment onto the RANSAC plane and subtract the vectors from each other. The grasping position is halfway between the two end points of the segment plus 7 cm in approach direction. Thus, the thumb reaches into the opening of the garment while the other fingers grasp it from the outside.

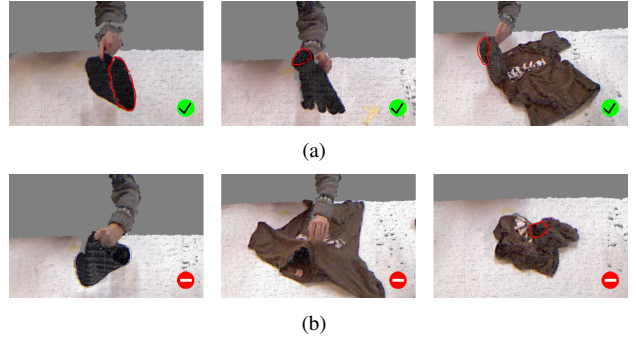


Fig. 7. Testing the boundary component detection technique with different garments. The results are depicted in red. (a) Three examples of successful detections. (b) Three examples of detection failures.

V. EXPERIMENTS

A. Evaluation of the vision algorithms

In order to test our computer vision algorithms, we implemented them using the Image Component Library (ICL)¹ which allows for efficient 2D image and 3D point cloud processing. The performance of our boundary detection method was found to be near real-time (10 Hz on an Intel XEON 3.6 GHz CPU using a NVIDIA GeForce GTX 660 Ti graphics card).

In addition to speed tests, we performed a qualitative evaluation of the algorithms. To this end, different items of clothing were held by a human hand in such a way that one opening was visible to the depth sensor. As shown in Fig. 7(a), boundary detection was successful for garments of various colors and sizes, e.g., a knit cap, a glove and a T-shirt. As opposed to curve detection methods based on parametrization or template matching such as Hough transform [28], our algorithm successfully found closed curves of arbitrary shape in a model-free manner. Furthermore, our approach had the advantage over active contour tracking [29] that no contour initialization was needed. The shortcomings of our method can be seen in Fig. 7(b). Firstly, we found that it was not robust against occlusion, e.g. by the human hand, because the occluding object interrupted the cycle in the graph. Secondly, detection failed when parts of the boundary between the garment and the table were too smooth to be found by the edge detector. Thirdly, dents which fulfilled the criteria of a boundary component led to false positives. In the future, these problems might be addressed by making use of prior knowledge of the scene.

B. Regrasping a knit cap with interactive perception

1) *System setup*: We tested our regrasping approach with a knit cap using two redundant 7-DOF Mitsubishi PA-10 robot arms. Both arms were equipped with a 20-DOF electric-actuated Shadow Dexterous Hand. For robot vision, we used a Microsoft Kinect including both a color camera and a depth sensor viewing from above toward the tabletop at an angle of about 45° . We employed a distributed system consisting of three computers, one for each group of tasks:

¹<http://www.iclcv.org/>

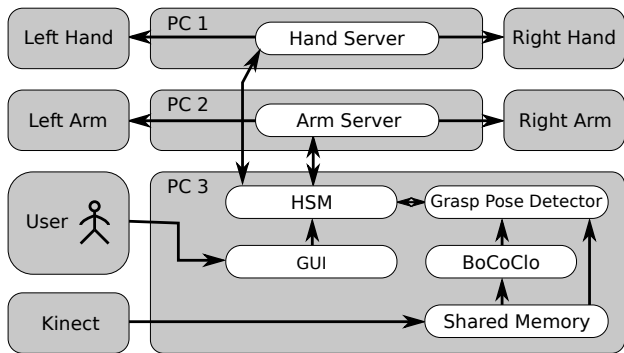


Fig. 8. Inter-process communication in the distributed system. A robotic server component controls the arms and simulates the robot kinematics in order to avoid obstacles. A shared memory provides the Kinect sensor data to the vision components. The BoCoClo (Boundary Components of Clothes) component is in charge of boundary detection, while the Grasp Pose Detector tries to find suitable pick-up and boundary grasp poses. Users can send commands to the robot through a graphical user interface. Overall task control is carried out by a hierarchical state machine (HSM).

hand control, arm control and vision. The inter-process communication flow is depicted in Fig. 8.

2) *Heuristic-based pick-up grasp*: Several sophisticated approaches to finding an optimal pick-up grasp pose for a piece of clothing being placed on top of a flat surface have been proposed [11] [12]. However, in our scenario, the grasp has to satisfy an essential restriction: The robotic hand must hold the knit cap in such a way that the opening is visible to the sensor. Therefore, we present a heuristic-based method exploiting gravity and the shape of the cap.

In order to keep it simple, we assume the biggest connected region of points above the tabletop to be the knit cap. Even if the cap may be crumpled, it usually retains an elongated shape. A binary decision has to be made whether the opening of the cap is pointing in direction of the robot or the opposite direction. We leave the decision to the user as this is not the focus of our current work. Picking up the cap at a high point in the front area of the item makes it hang in a good pose to be visually observed.

The grasp pose is again specified by a position vector, an approach vector and an orientation vector (defined as pointing backwards through the hole between thumb and forefinger). We perform a 2-dimensional principal component analysis (PCA) of the knit cap points, omitting the vertical coordinates. Let μ be the mean of the points, let λ_1, λ_2 be the eigenvalues of the covariance matrix, and let v_1, v_2 be the corresponding eigenvectors, v_1 pointing roughly toward the opening of the cap. Then, a line r can be defined as

$$r = \mu + \frac{1}{2} \sqrt{\lambda_1} v_1 + t \cdot v_2 \quad (6)$$

for $t \in \mathbb{R}$. The highest point within a distance of 1 cm to this line is selected as the grasping position (Fig. 6(a)). The approach direction is straight downward and the orientation vector is set to $-v_1$. In order to cover the whole range of possible angles, the grasping hand is chosen depending on the orientation vector. The robot picks up the cap and uses

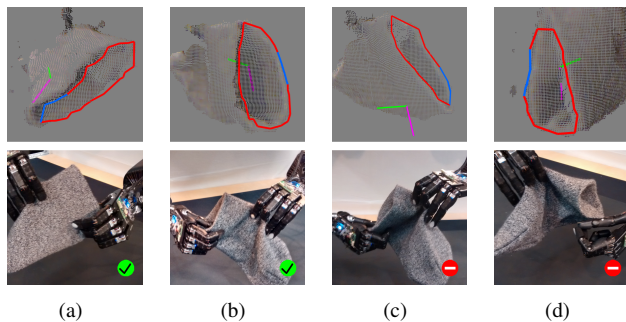


Fig. 9. Experimental trials. The robot regrasps a knit cap using interactive perception. (a+b) Two examples of successful runs. (c+d) Two regrasping failures.

TABLE I
EXPERIMENTAL RESULTS

	pick-up grasp	perception	regrasp
success rate	11/12	10/11	8/10
percentage	91.7	90.9	80.0

the arm and two fingers to bring it into a desirable position for visual observation.

3) *Experimental procedure*: We conducted twelve experimental runs. The human experimenter held the knit cap in the middle at different orientations and dropped it from a height of about 50 cm above the tabletop. The robot's task was to pick up the cap, detect the opening and regrasp it around a suitable segment of the boundary component. We aborted the trial if one of the subtasks failed.

4) *Results*: The robot successfully picked up the cap in eleven out of twelve runs. Boundary perception was successful in ten out of eleven remaining trials. The single detection failure was due to the cap being turned down in such a way that the opening wasn't visible to the sensor. The robot succeeded in regrasping the cap around its boundary in eight out of ten runs (e.g., Fig. 9(a) and 9(b)). One failure was caused by trying to grasp the cap too close to a corner and unintentionally also squeezing the opposite part of the boundary (Fig. 9(c)). The second failure occurred because the hand folded the cap to the side when trying to grasp it and, hence, missed the boundary (Fig. 9(d)). The overall success rate of the whole sequence was 66.7 percent. The results are summarized in Table I.

C. The coat-check scenario: exploiting the boundary grasp

We present an attempt to make use of the suggested regrasping technique in a simplified robotic coat-check scenario. The robot performs a sequence of eight actions in order to change the configuration of a knit cap from lying on the table to hanging on a hat-stand. While the first three steps have been discussed in detail, we basically regard the subsequent actions as an application of the boundary grasp. The overall procedure (Fig. 10) is as follows:

- 1) **Picking up**: The knit cap is picked up from the tabletop according to the heuristic-based grasp pose detection.
- 2) **Visually observing**: One hand holds the cap in such a way that the boundary can be detected.

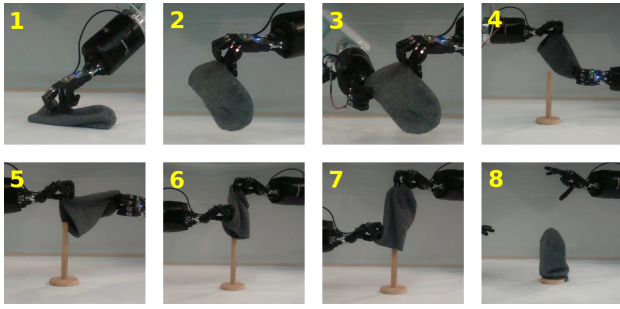


Fig. 10. The task of hanging up a knit cap on a hat-stand divided into a sequence of actions.

- 3) **Regrasping around the boundary:** Having determined a good grasping segment and pose, the robot regrasps the cap with the other hand.
- 4) **Regrasping at the tail:** The robot performs another grasp at a point close to the tail of the cap.
- 5) **Aligning:** The top of the hat-stand is detected in the point cloud and the cap is aligned accordingly.
- 6) **Lifting up:** One hand lifts the tail above the hat-stand.
- 7) **Pulling down:** The other hand pulls the cap over it.
- 8) **Releasing:** The robot drops the knit cap.

This basic clothes manipulation skill shows the usefulness of our affordance-driven and topology-based approach for an exemplary task of a robotic coat-check attendant. The procedure was successfully executed by a dual-arm robot with two anthropomorphic hands, although we observed occasional failures due to the lack of a tight coupling between perception and action in our prototypical implementation.

VI. CONCLUSION

We have proposed a graph-based method for perceiving the boundary components of a piece of clothing in a point cloud using interactive perception. We make a few assumptions with respect to the garment. These should be eliminated in order to bring the system closer to a real robotic coat-check attendant. Nevertheless, it is already possible to define a good regrasping pose based on the detected boundary components.

We have demonstrated the capabilities of the boundary grasp in a coat-check experiment with a knit cap. However, hanging up the cap is currently not as robust as regrasping it. Visual or tactile servoing would probably improve the performance drastically. In the future, the boundary grasp and other concepts inspired by topology might be useful for even more complex tasks such as putting on a pillow case or assisting people with dressing.

ACKNOWLEDGMENTS

This work was partially funded by the German Research Council (DFG), grant EXC 277. We would like to thank Guillaume Walck, Robert Haschke and everyone from the AGNI Grasp Lab for their helpful input and support.

REFERENCES

- [1] J. J. Gibson, "The theory of affordances," *Perceiving, Acting, and Knowing: Toward an Ecological Psychology*, 1977.
- [2] D. Katz and O. Brock, "A factorization approach to manipulation in unstructured environments," in *Robotics Research*, vol. 70, 2011, pp. 285–300.
- [3] F. T. Pokorny, J. A. Stork, and D. Kragic, "Grasping objects with holes: A topological approach," in *ICRA*, 2013, pp. 1100–1107.
- [4] W. H. Lui and A. Saxena, "Tangled: Learning to untangle ropes with rgb-d perception," in *IROS*, 2013, pp. 837–844.
- [5] L. Sun, G. Aragon-Camarasa, P. Cockshott, S. Rogers, and J. P. Siebert, "A heuristic-based approach for flattening wrinkled clothes," in *Towards Autonomous Robotic Systems*, 2013.
- [6] A. Doumanoglou, A. Kargakos, T.-K. Kim, and S. Malassiotis, "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning," in *ICRA*, 2014, pp. 987–993.
- [7] J. Stria, D. Prusa, V. Hlavac, L. Wagner, V. Petrik, P. Krsek, and V. Smutny, "Garment perception and its folding using a dual-arm robot," in *IROS*, 2014, pp. 61–67.
- [8] J. van den Berg, S. Miller, K. Goldberg, and P. Abbeel, "Gravity-based robotic cloth folding," *Springer Tracts in Advanced Robotics*, vol. 68, pp. 409–424, 2011.
- [9] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Parametrized shape models for clothing," in *ICRA*, 2011, pp. 4861–4868.
- [10] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *ICRA*, 2010, pp. 2308–2315.
- [11] A. Ramisa, G. Alenyà, F. Moreno-Noguer, and C. Torras, "Determining where to grasp cloth using depth information," in *Int. Conf. of the Catalan Association for Artificial Intelligence*, 2011, pp. 199–207.
- [12] K. Yamazaki, "Grasping point selection on an item of crumpled clothing based on relational shape description," in *ICRA*, 2014, pp. 3123–3128.
- [13] S. K. Christian Bersch, Benjamin Pitzer, "Bimanual robotic cloth manipulation for laundry folding," in *IROS*, 2011, pp. 1413–1419.
- [14] Y. Kita, F. Kanehiro, T. Ueshiba, and N. Kita, "Clothes handling based on recognition by strategic observation," in *Humanoids*, 2011, pp. 53–58.
- [15] C. Elbrechter, R. Haschke, and H. Ritter, "Folding paper with anthropomorphic robot hands using real-time physics-based modeling," in *Humanoids*, 2012.
- [16] J. Schulman, A. Lee, J. Ho, and P. Abbeel, "Tracking deformable objects with point clouds," in *ICRA*, 2013, pp. 1130–1137.
- [17] D. Katz and O. Brock, "Manipulating articulated objects with interactive perception," in *ICRA*, 2008, pp. 272–277.
- [18] B. Willimon, S. Birchfield, and I. Walker, "Model for unfolding laundry using interactive perception," in *IROS*, 2011, pp. 4871–4876.
- [19] B. Willimon, I. Walker, and S. Birchfield, "A new approach to clothing classification using mid-level layers," in *ICRA*, 2013, pp. 4271–4278.
- [20] —, "3d non-rigid deformable surface estimation without feature correspondence," in *ICRA*, 2013, pp. 646–651.
- [21] Y. Li, C.-F. Chen, and P. K. Allen, "Recognition of deformable object category and pose," in *ICRA*, 2014, pp. 5558–5564.
- [22] T. Tamei, T. Matsubara, A. Rai, and T. Shibata, "Reinforcement learning of clothing assistance with a dual-arm robot," in *Humanoids*, 2011, pp. 733–738.
- [23] A. Ückermann, R. Haschke, and H. Ritter, "Realtime 3d segmentation for human-robot interaction," in *IROS*, 2013, pp. 2136–2143.
- [24] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *CACM*, vol. 27, no. 3, pp. 236–239, 1984.
- [25] T. Hanser, P. Jauffret, and G. Kaufmann, "A new algorithm for exhaustive ring perception in a molecular graph," *Journal of Chemical Information and Computer Sciences*, vol. 36, no. 6, pp. 1146–1152, 1996.
- [26] X. C. He and N. H. C. Yung, "Corner detector based on global and local curvature properties," *Optical Engineering*, vol. 47, no. 5, 2008.
- [27] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *CACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [28] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
- [29] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: active contour models," *Int. Journal for Computer Vision*, vol. 1, no. 4, pp. 321–331, 1988.