Pointcloud-based Identification of Optimal Grasping Poses for Cloth-like Deformable Objects

Alessio Caporali, Gianluca Palli

DEI - Department of Electrical, Electronic and Information Engineering
University of Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

***Abstract*— In this paper, the problem of identifying optimal grasping poses for cloth-like deformable objects is addressed by means of a four-steps algorithm performing the processing of the data coming from a 3D camera. The first step segments the source pointcloud, while the second step implements a wrinkledness measure able to robustly detect graspable regions of a cloth. In the third step the identification of each individual wrinkle is accomplished by fitting a piecewise curve. Finally, in the fourth step, a target grasping pose for each detected wrinkle is estimated. Compared to deep learning approaches where the availability of a good quality dataset or trained model is necessary, our general algorithm can find employment in very different scenarios with minor parameters tweaking. Results showing the application of our method to the clothes bin picking task are presented.**

***Index Terms*— Deformable Object Manipulation, Point Clouds, Computer Vision, Robotic Applications.**

I. Introduction

The manipulation of clothes and fabrics, in general, constitutes a very active research topic other than a broad field of interest for industrial manufacturing. Clothes are highly deformable objects which means that their mechanical and visual properties (shape, appearance) change during time due to previous handling or external effects as the gravitational force. This result in their manipulation being more challenging compared to rigid objects, in particular concerning the development of perception-related capabilities.

A recent survey covering the literature of robotic sensing and manipulation of deformable objects can be found in [1]. Among the main contributions of the study, the authors propose a classification scheme for deformable objects based on their geometric shape and physical properties. In this study, clothes are classified as biparametric objects not possessing any compression strength. Biparametric means that they have one dimension considerably smaller that its other two, i.e. the thickness of the fabric. In the surveys, the authors recognize a tendency of approaches to develop task-specific techniques or to exploit peculiar assumptions on object type and characteristic. The authors conclude that the lack of general solutions remains a major open issue.

This work was supported by the European Commission's Horizon 2020 Framework Programme with the project REMODEL - Robotic technologies for the manipulation of complex deformable linear objects - under grant agreement No 870133.

Corresponding author: *alessio.caporali2@unibo.it*

(a) Bin picking (b) Washing Machine picking

Fig. 1: Scenarios in which our proposed algorithm can find employment: robot executing grasp operations in a) bin picking task and b) washing machine picking task.

Extensive work has emerged in the garment manipulation literature, as noted by [1]. In the following, major contributions are presented.

In [2] the authors use 3D data to recognize the shape of a cloth while being held in the air by a humanoid robot. From a crumble configuration, the cloth is re-grasped along the rim, and its configuration is compared to simulated data obtained based on the object size and softness.

In [3] the authors describe a three layer architecture for the detection of informed initial grasping points using a RGB-D image. A Bag of Features descriptor is constructed using texture and geometric information and a detection probability map is built based on the descriptor. The map expresses the probability of a polo collar to be present in a given region. Local peaks of the probability map are refined. A graspability measure (shown in details in [4]) is applied to the depth image. It assigns more importance to the regions of the image showing wrinkle-like structures. Combining the graspability measure results and the probability map, it is possible to obtain grasping points corresponding to the polo collar.

In [5], the goal is to bring a cloth in a planar state with the least amount of manipulations. A method is developed for obtaining two grasp point for a cloth randomly placed on a flat surface. Utilizing a dual-arm robot and the two computed grasp points, the cloth is spread out directly. Shape description and grasp point representation are based on hem elements. Grasp points are selected comparing a candidate point to a training dataset where ideal points were assigned manually. Similarly, in [6], a stereo robot head gathers high-quality RGB-D images of clothes from which a preprocessing stage extracts and ranks wrinkle structure. The largest wrinkle is targeted and an heuristic cloth-flattering strategy employing a dual-arm robot is applied.

In [7] the authors address the problem of grasping a cloth lying flat on a surface. The solution proposed is based on the generation of a wrinkle and its utilization for the grasping. The approach uses a single arm robot with a parallel gripper.

More related to garment manipulation efficiency, in [8] an approach for separating clothes on a table with the least amount of manipulations is discussed. The task is defined as a partially observable Markov decision problem (POMDP).

Finally, addressing the problem of garment folding, we can cite [9], [10] and [11]. In [9] the authors present a vision-based algorithm for the reliable detection of corners in a piece of cloth. Only geometric cues are used making the algorithm robust to variation in texture. Depth-discontinuity edges in the image are key properties used by the algorithm to identify borders. In addition, a procedure for the folding of the cloth is discussed, it is based on the employment of a two arm robot performing a sequence of vision-based re-grasps and manipulation both in the air and on a table. In this work, only clothes with a rectangular shape are addressed. Differently, in [10], the folding of T-shirts with a dual-arm robot is presented. The cloth, covered with fiducial markers, is rotated by 360 degrees and a pointcloud representation generated. The current grasp point is estimated and the next grasp point is selected and evaluated (using machine learning techniques). The goal is to reach a configuration where the robot holds the T-shirt by the shoulders in order to perform the folding sequence. In [11] different types of clothes are addressed. Starting from a spread out configuration on a table, a system composed by a dual arm robot is capable of folding the garment. Key points are detected by fitting the contour mask to a polygonal model. The latter allows the algorithm to work with various types of objects.

In this paper, we propose an algorithm able to identify optimal grasping poses for cloth-like deformable objects. Our approach does not require any prior knowledge of the clothes under exams nor any validation/comparison stage employing a machine learning technique. Hence, it is very general and can be applied to completely new scenarios without performing any early step in preparation (e.g. creation of a dataset and training). Additionally, our proposed algorithm addresses the drawbacks highlighted by the authors in [3] and [4], specifically the absence of any concavity measure and the missing usage of points height information in defining the graspability measure. This work aims at showing how the exploitation of these auxiliary cues combined with a wrinkles detection algorithm increases the robustness in the estimation of the grasping poses.

The paper is structured as follow. In Sec. II the proposed algorithm is discussed, while in Sec. III experimental results are presented to verify the algorithm effectiveness with different samples of pointcloud data obtained from a ToF 3D sensor. Finally, Sec. IV draws some conclusions on the achieved results and mentions future works.

II. Proposed Method

The method here proposed aims at detecting a reliable grasping pose where the robot can attempt a robust grasp by using a parallel gripper. For this task, performing the grasp on a sleeve, neck or collar is not relevant. Hence, the identification of these peculiar regions is not addressed. Moreover, although a wrinkle generation strategy [7] removes the problem of identify the grasping pose, it increases the number of manipulations and execution time required. Thus we focus on providing an efficient algorithm where the grasp is executed in one motion only.

The algorithm is organized in four steps:

- Step 1: performs the segmentation of the source pointcloud in order to provide, to the next step, a new pointcloud composed by only cloth-related points;
- Step 2: implements the detection of the graspable regions by exploiting an entropy measure combined with convexity and depth maps to increase the detection robustness;
- Step 3: performs the fitting of piecewise curves to identify the wrinkles;
- Step 4: estimates the grasping pose for each detected wrinkle.

In Fig. 2, a graphical representation of the algorithm workflow is provided. The algorithm is developed in C++ and takes advantage of the PointCloud Library (PCL) [12] and the Eigen Library. The four steps of the algorithm are analyzed below.

A. PointCloud Segmentation

This step is intrinsically scenarios dependent. Geometrical considerations and tools available inside the PointCloud library are used to perform the segmentation. Even if not mandatory, the segmentation of the source pointcloud brings beneficial effects, as: a) ambiguous points that do not belong to a cloth are removed; b) the overall size of the pointcloud is reduced and, consequently, the computation efficiency increased; c) the bin segmentation enables resizing the considered interior region of the bin, i.e. removing areas too close to the bin wall to add a safe space for the gripper operations.

In more details, given the organized source pointcloud P_S, the edges are detected as shown in [13]. The plane with edges inliers that is both parallel and most distant from the ground plane is found and its points are extracted. The four corners of the bin are found in the extracted points by fitting four lines and by verifying the orthogonality for each pair of lines composing a corner. Fig 2(b) shows the detected bin contour with corners depicted in red. The RANSAC algorithm [14] is used to fit both plane and line models to the pointcloud. From the corners, a convex hull is built and used to crop the internal volume of the bin, as shown in Fig. 2(c) by the yellow points inside the bin. The eight red dots display the corners of the volume considered which is reduced from the original one to introduce a safe space nearby the bin walls.

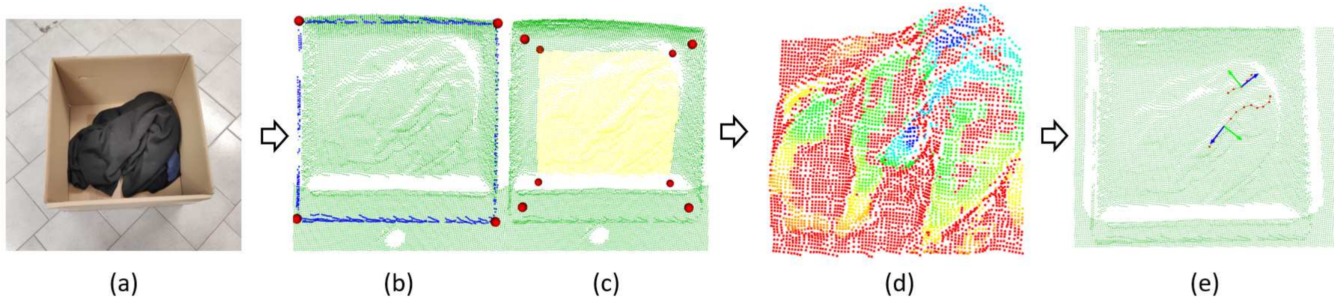


Fig. 2: Schema of the proposed algorithm. a) Image showing the input clothes inside a bin; b-c) Pointcloud segmentation applied to the bin (step 1); d) Entropy map related to the graspable regions detection (step 2); e) Source pointcloud with estimated grasp poses (steps 3-4).

B. Graspable Regions Detection

The segmented pointcloud is processed in the second step in order to detect graspable regions. A graspability measure is employed to gain an understanding of the location of highly wrinkled areas in the cloth. This information is encoded into an *entropy map*. A *depth map* and *convexity map* are used as auxiliary cues to robustify the detection of the wrinkled areas.

1) Normals Estimation: The first step in the detection of the regions is based on low-level features as the surface normals of the pointcloud. They are computed relying on the Moving Least Squares algorithm [15] which smooth out the pointcloud surface by fitting a polynomial curve before estimating the surface normals. By using this approach, a reduction in the noise in the estimation process is obtained.

The normal vectors obtained are expressed in Cartesian coordinates as (n_x, n_y, n_z). They are transformed in spherical coordinates since only two components are relevant. The transformation involved is documented in [16]. The pair of angles (ϕ, θ) are calculated as:

$$\phi = \text{atan}\left(\frac{n_z}{n_y}\right), \quad \theta = \text{atan}\left(\frac{\sqrt{n_z^2 + n_y^2}}{n_x}\right)$$

The angle ϕ is denoted as azimuth angle while θ as inclination angle.

2) Convexity Map: Detecting the concavity or convexity of a local area is important to remove from the highly wrinkled areas the regions that are not easily graspable. The method presented here is capable of providing such understanding with a very small computation footprint. This procedure can determine if a point's neighborhood is convex or concave [17]. Given the input and normal vectors pointclouds in Cartesian coordinates, for each point a local neighborhood is found by choosing local patches composed by 9 points. Focusing on a given patch, lets denote by $\vec{p}_1$ the (x, y, z) coordinates of the considered point and by $\vec{n}_1$ its normal. Lets $\vec{p}_2$ and $\vec{n}_2$ be in turn one of its neighborhood points. The distance vector between $\vec{p}_1$ and $\vec{p}_2$ is computed as $\vec{d} = \vec{p}_1 - \vec{p}_2$. Then, the angle α_1 between $\vec{n}_1$ and $\vec{d}$ is compared with the one α_2 between $\vec{n}_2$ and $\vec{d}$. A convex connection between $\vec{p}_1$ and $\vec{p}_2$ is defined if α_1 is smaller than α_2. The condition can be expressed as:

$$\alpha_1 < \alpha_2 \Rightarrow \cos(\alpha_1) - \cos(\alpha_2) > 0 \iff \vec{n}_1 \cdot \hat{d} - \vec{n}_2 \cdot \hat{d} > 0 \tag{1}$$

$$\hat{d} = \frac{\vec{p}_1 - \vec{p}_2}{||\vec{p}_1 - \vec{p}_2||}$$

If the condition (1) is not satisfied, the two points will exhibit a concave connection between them. The computation is performed for all the neighborhood points that satisfy a check based on the normal vectors difference angle: the convex connectivity is calculated only if the two normal vectors n_1 and n_2 have a significant angle difference between them. The original point is set to be convex if all of its neighborhood exhibit a convex connection with him. Figure 3 displays the convex and concave conditions. The results show that this simple approach is able to detect convex regions as wrinkles and edges in the clothes, at least in an approximated way. The combination of the entropy filter with the convexity check allows the robust detection of convex wrinkles only.

3) Depth Map: The depth map importance is twofold. First, it is used for the calculation of the weight factor in case the weighted version of the entropy formula is employed (see Sec. II-B.4). Second, it allows to correctly detect situations in which the target cloth lies completely flat on a planar surface with not detectable wrinkles. Indeed, the entropy filter in absence of wrinkles identifies as "highly wrinkled areas" the corners along the edge of the cloth although this type of points is not easily graspable. The depth map is build

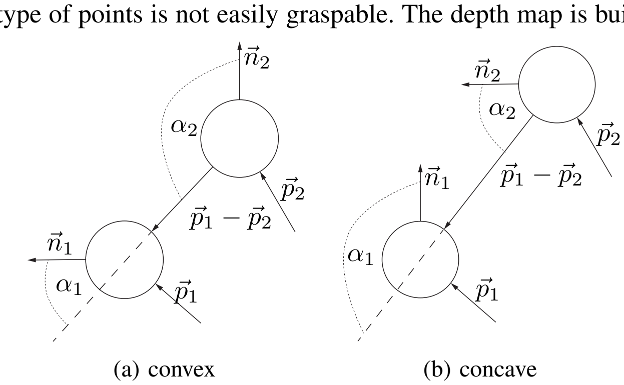
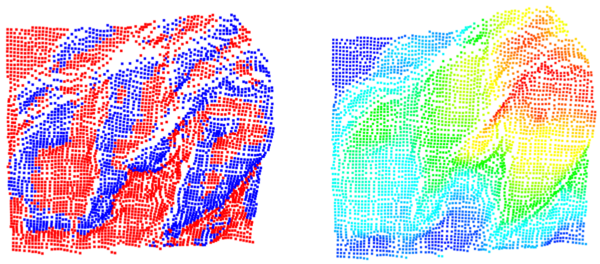


Fig. 3: Drawing showing the convex and concave conditions.



(a) Convexity map (b) Depth map

Fig. 4: Convexity and depth maps. The convexity map can be interpreted as binary map, where red points correspond to concave or flat regions and blue points to convex areas. In the depth map blue points correspond to low depth values whereas red points signify high depth values in that regions.

by using a reference plane and by evaluating the distance between each point in the input pointcloud and this plane. The choice of the reference plane depends on the scenario. In the bin picking, the knowledge of the four top vertices location is used for the computation of the bin-top plane. The point to plane distance is computed for all the points of the segmented pointcloud. At the height of the point corresponding to the largest distance is fit a plane parallel to the bin-top one using RANSAC. This plane is used as reference level for the calculation of the map.

4) Entropy Filter: The entropy filter is employed in order to quantify how much information exists in a given local region. In particular, the goal is to discover regions of the clothes with a sparse distribution of normals. They will result in a high value in the entropy measure. Instead, regions with normals mostly aligned with each other will be characterized by a low value in the entropy measure. For each point in the input pointcloud a local region is considered and a two-dimensional histogram is constructed. The histogram is built with the two spherical components of the surrounding normal vectors. Hence, the histogram is used to model the spherical coordinate angles distributions. The entropy measure is defined as:

$$H(x) = -w_x \sum_{i=1}^{n} p(x_i) \log p(x_i) \quad (2)$$

where x is the point considered to which a two-dimensional histogram of orientation angles in spherical coordinates (azimuth and inclination) is associated. The histogram is made of n bins for each dimension. The parameter w_x is the weight related to point x. With x_i we are denoting the i-th bin of the histogram and with $p(x_i)$ its associated value. The weight factor comes from the depth map (Sec. II-B.3). In particular, w_x represent the intensity value in the depth map of the point considered. As the point is far away from the reference plane, its associated intensity value is larger and the weight factor increases. As result, the points that are more distant from the plane are preferred. If the not weighted version of the formula is needed, w_x can be set equal to one.

5) Entropy Map Segmentation: The last operation performed in the second step is the extraction of the graspable areas. The input pointcloud is segmented based on an entropy threshold. All the points having entropy normalized values above a threshold are kept. This threshold value is a user-defined parameter.

C. Wrinkles Curve Fitting

Ideally, the wrinkles of a cloth represent the best areas to perform the grasp, especially when working with a robot equipped with a parallel gripper. For this reason, we try to discover all the wrinkles present. The detection is performed on the segmented entropy map pointcloud of the previous step. Each wrinkle is detected individually by the process described in Algorithm 1.

Algorithm 1: Curves Fitting Algorithm

Result: Set of piecewise curves
initialization: p_s = global maximum entropy point;
for *all possible high entropy areas* **do**
    for $it \leq it_{max}$ **do**
        extraction of P_{NN} given p_s;
        calculation of p_C and M_C given P_{NN};
        estimation of propagation direction;
        step in the estimated direction;
        update of p_s;
        if $\|p_s^{NN} - p_s\| \leq$ *threshold* **then**
            it++;
        else
            break;
        end
    end
    curve = concatenation of p_C points;
    extraction of curve area from entropy map;
    p_s = new global maximum entropy point;
end

In Alg. 1, with P_{NN} is denoted the pointcloud composed by the neighborhood points of p_s (search point). The symbols p_C and M_C are referred to the centroid and covariance matrix respectively. The matrix M_C is normalized and its eigenvalues and eigenvector are used to estimate the direction of propagation. An increasing motion is enforced by checking the angle between the new direction and the previous one. The step motion is performed from p_C. The symbols p_s^{NN} indicates the closest neighborhood point of p_s belonging to the segmented entropy map pointcloud.

D. Poses Estimation

Each extracted wrinkle is stored as a set of sequence of points B. By connecting each point one after the other, a piecewise representation of a curve denoting the wrinkle's path is obtained, see Fig. 5. The grasp pose is computed by considering a picking motion orthogonal to the ground plane. Hence, the curve is projected on the ground plane and the RANSAC algorithm applied to identify the most linear

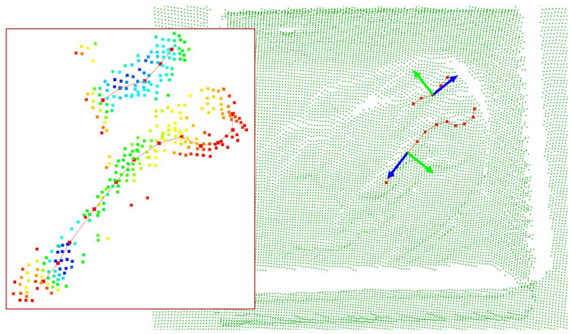


Fig. 5: Wrinkles estimated with grasping frames shown on source pointcloud. A zoom highlighting the piecewise curves on the segmented entropy map is shown as well: the propagation is initialized from the bluish regions corresponding to the local maxima of entropy values.

section of the curve. In such spot the target grasping frame is attached. The origin of the frame is translated back from the projected curve to the original one. The first axis is aligned to the wrinkle direction while the second one is set orthogonal to the first one and, at the same time, parallel to the ground plane due to the projection. The last axis is orthogonal to both the first and second one. The procedure described is performed for all the wrinkles detected in step 3. The target frames are expressed in the camera reference frame, therefore they need to be transformed in the world frame to properly make use of them. Moreover, the grasp poses can be further ordered according to their distance from the bin border in order to select the safer for the robot to execute the grasp.

III. Experimental Results

The setup used during the experiments is composed of a PC running Ubuntu 16.04 LTS and the PointCloud Library in version 1.9.1. The processing power consists of a dual core CPU with a maximum speed of 2.90 GHz.

We have performed tests taking pointcloud samples from a CamBoard Pico Flexx 3D ToF device. The samples have a resolution of 224 x 171 (38k) pixels, the maximum possible. They consist of one or more clothes randomly placed inside a bin. In particular, the clothes were dropped from above.

In Fig. 6, the use of the weighted version of the entropy formula (2) is compared to the not weighted one. Moreover, this figure provides a comparison between the calculation of the entropy for all the pointcloud points or for only the convex points obtained from the convexity map described in Sec. II-B.2 and reported in Fig 4a.

A comparison of the average execution time of the algorithm is provided in Fig. 7, where the execution time of step 2 only is reported in blue and the one of the entire algorithm is reported in orange. In Convexity Map, the entropy calculation has been performed only for convex points instead of considering all the pointcloud points.

Figure 6(A) shows how for some concave areas, an high entropy value is detected. The usage of the weighted version of eq. (2) allows to mitigate this effect since areas with a greater depth value are probably convex. This can be appreciated by comparing the ”bluish” areas of Fig. 6(C)

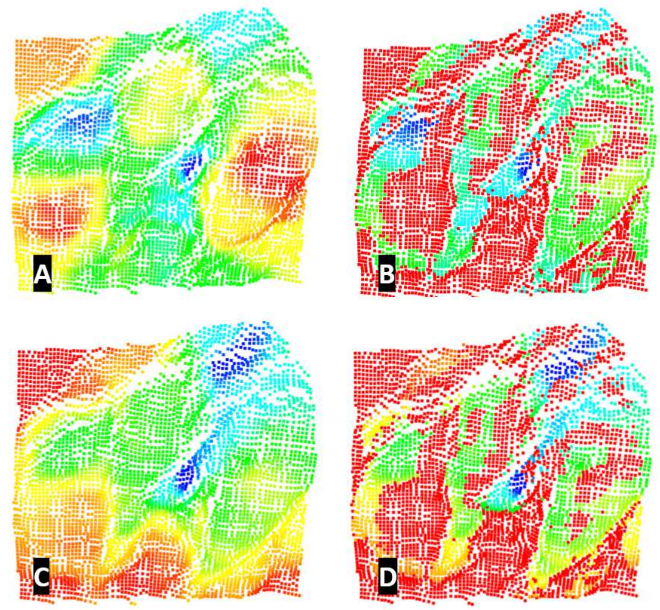


Fig. 6: Comparison involving the entropy formula application: A) not weighted on all points; B) not weighted on convex points; C) weighted on all points; D) weighted on convex points

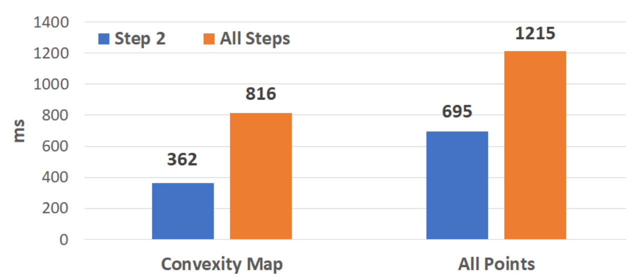


Fig. 7: Average execution time of the algorithm.

with the convexity map shown in Fig. 4a. By exploiting the convexity map, other than an increase in the robustness of the algorithm in computing graspable poses, a reduction in the execution time due to the decrease in the total number of points to evaluate is obtained, see Fig. 7 which shows a reduction in time by 1/3.

Extensive tests were performed on clothes with different configurations. Some results are presented in Fig. 8. From these results, it is possible to observe that the algorithm is effective also in case of very crumbled configurations and with clothes in an almost flat configuration having only small wrinkles. The user-defined parameters used to obtain the results shown are noted in Table I. Figures 8-(C, D, and E) present highly wrinkled clothes, resulting into more than one possible grasping poses. On the contrary, in Figures 8-(A, B, and F) the possible grasping pose is just one. This difference is due to the utilization of the weighted version of the entropy formula (2) where wrinkled areas with greater depth value are prioritized. In order to detect the other ”shallow” wrinkles, an iterative approach can be utilized where the area of the previously detected wrinkle is removed. Alternatively,

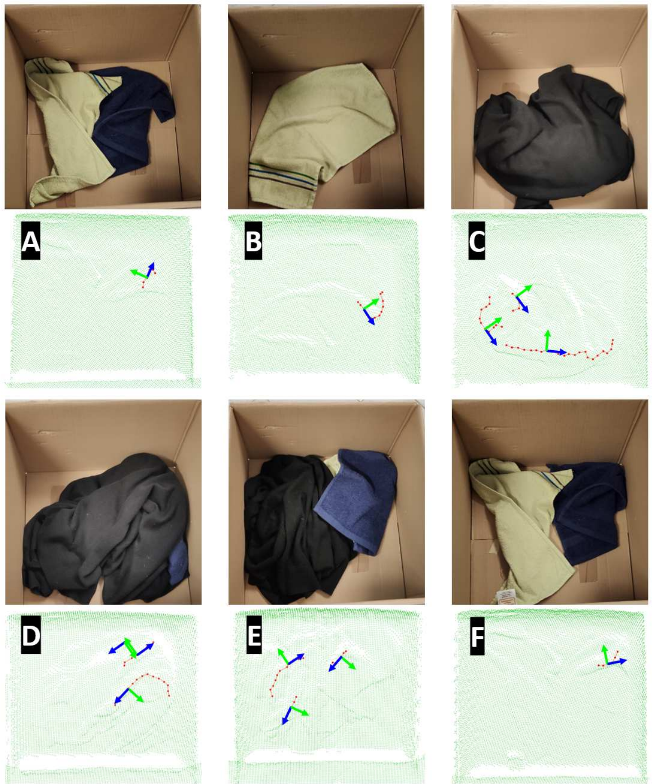


Fig. 8: Results of tests performed on clothes with different crumbled configurations.

the entropy threshold parameter can be decreased to include a larger set of points.

IV. Conclusions

In this paper, an algorithm for the identification of grasping poses of deformable objects like clothes given an input pointcloud is proposed. For tasks involving the development of perception capabilities for robotic manipulation of deformable objects, it is usually the case that the scenario analyzed is constrained by several restrictive assumptions. In the work here presented, a general and modular vision-based procedure where restrictive assumptions were reduced as much as possible has been developed and demonstrated through experiments by the application of the algorithm to the clothes bin picking task. The solution proposed does not require any prior knowledge of the type of clothes under exams, nor any initial training phase.

TABLE I: Parameters used during the experimental tests.

Parameters	Values
Entropy threshold	0.5
Depth threshold [m]	0.03
Convexity angle threshold [deg]	5
Curvature threshold	0.1
Local search neighborhood [pixels]	500
Downsampling leaf-size [m]	0.005
Histogram dimensions [n]	64x64

Future work includes the generalization of the algorithm and its experimental evaluation with different types and loading conditions of the clothes' containers. Moreover, tests involving the application of the algorithm to other tasks, as the washing machine picking of Fig. 1(b), can be explored. Finally, a more computationally efficient implementation can be investigated as well as a reduction in the number of user-defined parameters.

References

[1] J. Sanchez, J.-A. Corrales, B.-C. Bouzgarrou, and Y. Mezouar, "Robotic manipulation and sensing of deformable objects in domestic and industrial applications: a survey," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 688–716, 2018.

[2] Y. Kita, F. Kanehiro, T. Ueshiba, and N. Kita, "Clothes handling based on recognition by strategic observation," in *2011 11th IEEE-RAS International Conference on Humanoid Robots*. IEEE, 2011, pp. 53–58.

[3] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 1703–1708.

[4] ——, "Determining where to grasp cloth using depth information," 2011.

[5] K. Yamazaki, "Grasping point selection on an item of crumpled clothing based on relational shape description," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 3123–3128.

[6] L. Sun, G. Aragon-Camarasa, S. Rogers, and J. P. Siebert, "Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2015, pp. 185–192.

[7] M. Shibata, T. Ota, and S. Hirai, "Wiping motion for deformable object handling," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 134–139.

[8] P. Monsó, G. Alenyà, and C. Torras, "Pomdp approach to robotized clothes separation," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2012, pp. 1324–1329.

[9] 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 *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 2308–2315.

[10] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1413–1419.

[11] J. Stria, D. Průša, V. Hlaváč, L. Wagner, V. Petrík, P. Krsek, and V. Smutnỳ, "Garment perception and its folding using a dual-arm robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 61–67.

[12] R. B. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *2011 IEEE international conference on robotics and automation*. IEEE, 2011, pp. 1–4.

[13] C. Choi, A. J. Trevor, and H. I. Christensen, "Rgb-d edge detection and edge-based registration," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1568–1575.

[14] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[15] M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C. T. Silva, "Computing and rendering point set surfaces," *IEEE Transactions on visualization and computer graphics*, vol. 9, no. 1, pp. 3–15, 2003.

[16] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," *KI-Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010.

[17] S. Christoph Stein, M. Schoeler, J. Papon, and F. Worgotter, "Object partitioning using local convexity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 304–311.