Identification of the highest wrinkle grasping point of a folded hospital gown*

Olivia Nocentini ^{1,2,*}, Jaeseok Kim¹, Julia Borras³, Guillem Alenyà³ and Filippo Cavallo^{1,2}

¹University of Florence, Department of Industrial Engineering, Via di Santa Marta 3, 50139, Florence ²Sant'Anna School of Advanced Studies, Department of BioRobotics, Viale Rinaldo Piaggio 34, 56025, Pontedera (PI) ³Institut de Robotica i Informatica Industrial CSIC-UPC, Carrer de Llorens i Artigas, 4, 08028 Barcelona, Spain

Abstract

There are already more than one billion people over the age of 60, and the World Health Organization predicts that number will increase to 1.4 billion by the year 2030. As a result, the need for caretakers is increasing, which could make society in the future unable to provide it. In this scenario, the need for automated assistance increases as the global population ages. One area of robotics where robots have demonstrated tremendous promise in closely collaborating with people is service robotics. Hospitals, residences, and facilities for the elderly will all require the deployment of intelligent robotic agents to carry out regular tasks. Cloth manipulation is one such daily activity and represents a challenging area for a robot. The research goal of this paper focused on finding the grasping points of the highest wrinkle (from a later point of view) of a folded hospital gown to then unfold it and help dressing a patient. The wrinkle is detected using the Generative Grasping Convolutional Neural Network (GG-CNN2), while the approach to the cloth by a manipulator is obtained by designing a visual servoing algorithm that considers the input of the GG-CNN2. In conclusion, the results described in this paper tend to study by deep some AI-based approaches for cloth manipulation capabilities; in particular, we concentrated on studying how to identify the first wrinkle of a cloth by combining the visual servoing approach with a neural network.

Keywords

cloth manipulation, convolutional neural networks, visual servoing, social robots

1. Introduction

In both home and industrial contexts, detecting and manipulating cloth is a common activity, however, due to the deformability of cloth, such tasks continue to be difficult for robots. Furthermore, in many cloth-related tasks like laundry folding and bed making or in dressing a person it is crucial to manipulate specific regions like edges, corners and wrinkles. Concerning the detection of the edges and corners, in [1], the authors concentrated on the challenge of seg-

International Conference on Social Robotics

^{*}You can use this document as the template for preparing your publication. We recommend using the latest version of the ceurart style.

^{*}Corresponding author.

[†]These authors contributed equally.

[☆] olivia.nocentini@santannapisa.it (O.N.); jaeseok.kim@unifi.it (J. Kim); jborras@iri.upc.edu (J. Borras); galenya@iri.upc.edu (G. Alenyà); filippo.cavallo@unifi.it (F. Cavallo)

^{© 02022} Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

menting and understanding these crucial sections. Their method taught a network to separate folds and creases from the edges and corners of a piece of clothing in a depth image. The grip location, direction, and directional uncertainty from the segmentation are also estimated using a novel approach that they provided. There are many different ways to detect wrinkles and grasping points in a cloth. The easiest one, as shown in [2], is to compute the binary image of the apparel and some filters to the image to find the wrinkles as the darkest regions. In [3], the authors used a four-step approach to analyse the input from a 3D camera and the challenge of selecting the best grabbing postures for cloth-like deformable objects is tackled in this study. The source point cloud is divided into the first stage, and a wrinkledness measure that can reliably identify areas of the cloth that can be grasped is implemented in the second step. The final stage involved fitting a piecewise curve to each individual wrinkle to identify it. The fourth and final stage estimated a target clutching stance for each observed wrinkle. In another work, [4], the same authors showed a wrinkledness measure to identify wrinkles in the cloth surface, to robustly assign spline curves to the detected wrinkle-like structures and to estimate grasping frames. In [5], the proposed visual perception architecture is able to parse the various garment configurations by detecting and quantifying structures i.e. grasping triplets and wrinkles from unfolded cloth with a dual-arm robot.

The main lack of previous works is finding wrinkles in folded clothes with a robot. Moreover, detecting wrinkles and grasping points from a lateral point of view instead of detecting these regions of the garment from a top point of view is a topic that has not been investigated. Detecting wrinkles from a lateral point of view is challenging since sometimes it is more "natural" for a person to take the garment from a lateral perspective. Our work tries to solve these issues in detecting the highest wrinkle grasping point of a folded hospital gown from a lateral point of view.

The contributions of our work are the following:

- detecting the highest wrinkle grasping point of a folded hospital gown from a lateral point of view
- combining the Generative Grasping Convoulutional Neural Network (GG-CNN2) with a visual servoing approach to move the robot near the highest wrinkle

2. Methods

The main idea of this work is to implement a method that finds the grasping point in the highest wrinkle of the hospital gown. In the following subsection, the procedure to detect the grasping point is shown.

2.1. Detecting the grasping points of the wrinkle of a folded hospital gown

This section's major goal is to demonstrate how to locate the ideal grabbing rectangle in the top wrinkle of a hospital gown, which is also the region that makes it easiest to unfold the fabric. We employed a neural network dubbed the Generative Grasping Convolutional Neural Network to do this (GG-CNN2) [6]



Figure 1: The architecture of the Generative Grasping Convolutional Neural Network 2: it is composed by 6 convolutional layers, 2 MaxPooling layers and two Transpose Convolution layers.

2.1.1. Generative Grasping Convolutional Neural Network

The network GG-CNN2 used in our work derives from the GG-CNN network [6]. The advantages of GG-CNN over other state-of-the-art grasp synthesis CNNs are twofold. First, the authors do not rely on sampling of grasp candidates, but rather directly generate grasp poses on a pixel-wise basis, analogous to advances in object detection where fully convolutional networks are commonly used to perform pixel-wise semantic segmentation rather than relying on sliding windows or bounding boxes [7]. Second, the GG-CNN has orders of magnitude fewer parameters than other grasp synthesis networks, allowing our grasp detection pipeline to execute fast enough for closed-loop grasping.

In [8], [9], [10], [11] the grasp representation proposed by [8], and then simplified by [12], was used to generate antipodal robotic grasps using RGB-D images of objects. The grasp representation is defined by the following formula:

$$g = \{x, y, \phi, h, w\},\tag{1}$$

The function *g* represents a five dimensions rectangle that includes the center of the rectangle (x, y), the orientation of the rectangle relative to the horizontal axis of the image ϕ , its width (h), and height (w). Morrison et al. [6] proposed a new representation of robotic grasps that changes the function *g* into this representation:

$$g = \{p, \phi, w, q\},\tag{2}$$

where p = (x, y, z) is the center position of the gripper, ϕ is the rotation angle relative to the horizontal axis of the image plane, *w* is the gripper width, and *q* is the grasp quality. Concerning the computation of *q*, which represents the chances of grasp success, each grasping rectangle has set the corresponding area of *q* to a value of 1. All other pixels are 0.

Robotic grasps are detected in the depth image $I = \mathbb{R}^{H \times W}$ with height *h*, and width *w*. In the image space *I*, the grasping point is represented by

$$\tilde{g} = \{s, \phi, \tilde{w}, \tilde{q}\},\tag{3}$$

where s = (u, v) represents the center point in pixels coordinates, $\tilde{\phi}$ denotes the rotation relative to the camera frame around the z-axis, \tilde{w} denotes the gripper width in pixels, and \tilde{q} the grasp quality.

The grasp map G proposed by Morrison et al. [6] is :

$$G = \{\Phi, W, Q\} \in \mathbb{R}^{3 \times H \times W}$$

$$\tag{4}$$

 Φ is an image which describes the angle of a grasp to be executed at each point, W an image which expresses the gripper width of a grasp to be executed at each point, Q an image which defines the quality of a grasp executed at each point (u,v). Φ , W, Q are each in $\mathbb{R}^{1 \times H \times W}$ and each pixel contains the values $\tilde{\phi}$, \tilde{w} , and q respectively at each pixel s. Following [6], the authors used this network to generate a grasp g for each pixel in depth image I, which denotes the pixel-wise representation:

$$M(I) = G, (5)$$

where the map function *M* is a deep NN, and then the best grasp can be found by :

$$g = \max_{Q} G. \tag{6}$$

The Mean Squared Error loss is applied to predict grasp pose, grasp quality \tilde{q} , gripper width \tilde{w} and rotation $\tilde{\phi}$.

2.2. Image Visual Servoing

The term *visual servo* (VS) control describes the use of computer vision data to regulate a robot's movements. The camera that collected the visual data might be directly attached to a robot manipulator or on a moving robot, in which case the camera moves along with the robot, or the camera can be fixed in the area so it can see the robot's movements from a position of stillness. Other arrangements might be imagined, for example, many cameras installed on pan-tilt heads watching the robot's motion.

The aim of all vision-based control schemes is to minimize an error e(t), which is typically defined by

$$e(t) = s(m(t), a) - s^{*}$$
(7)

The vector m(t) is a set of image measurements (e.g., the image coordinates of interest points, or the parameters of a set of image lines or segments). These image measurements are used to compute a vector of k visual features, s(m(t), a), in which a is a set of parameters that represent potential additional knowledge about the system (e.g., true or approximate camera intrinsic parameters or a model of the object to be tracked). The vector s^* contains the desired values of the features. Note that the order of the desired and actual values in Equation (7) is reversed with respect to the common convention for feedback control systems. Visual servoing schemes mainly differ in the way that s is designed.

There are several visual servoing approaches including image-based visual servo control (IBVS), in which s consists of a set of features that are immediately available in the image, and pose-based visual servo control (PBVS), in which s consists of a pose, which must be

estimated from image measurements. In this paper, we consider the IBVS and we will call it as VS approach.

As concerns IBVS, the image measurements *m* are usually the pixel coordinates of the set of image points (although this is not the only possible choice), and the parameters *a* in the definition of s = s(m, a) in Equation 7 are nothing but the camera intrinsic parameters to go from image measurements expressed in pixels to the features.

A three-dimensional world point with coordinates $\mathbf{X} = (X, Y, Z)$ in the camera frame projects into the image plane of a conventional perspective camera as a two-dimensional point with normalised coordinates $\mathbf{x} = (x, y)$. More precisely we have:

$$\begin{cases} x = \frac{X}{Z} = \frac{u - c_u}{f\alpha} \\ y = \frac{Y}{Z} = \frac{v - c_v}{f} \end{cases}$$
(8)

where m = (u, v) gives the coordinates of the image point expressed in pixel units, and a = (c_u, c_v, f, α) is the set of camera intrinsic parameters, c_u and c_v are the coordinates of the principal point, *f* is the focal length, and α is the ratio of the pixel dimensions. In this case, we take $s = \mathbf{x} = (x, y)$, the image plane coordinates of the point.

Taking the time derivative of the projection Equation (8), we obtain

$$\begin{cases} \dot{x} = \frac{\dot{X}}{Z} - \frac{X\dot{Z}}{Z^2} = \frac{\dot{X} - x\dot{Z}}{Z} \\ \dot{y} = \frac{\dot{Y}}{Z} - \frac{Y\dot{Z}}{Z^2} = \frac{\dot{Y} - y\dot{Z}}{Z} \end{cases}$$
(9)

We can relate the velocity of the 3-D point to the camera spatial velocity using the well-known equation

$$\dot{\mathbf{X}} = -v_c - w_c \times \mathbf{X} = \begin{cases} \dot{X} = -v_x - w_y Z + w_z Y \\ \dot{Y} = -v_y - w_z X + w_x Z \\ \dot{Z} = -v_z - w_x Y + w_y X \end{cases}$$
(10)

where $v_c = (v_x, v_c y, v_z)$ and $\omega_c = (\omega_x, \omega_y, \omega_z)$. Inserting Equation 10 into 9, grouping terms, and using Equation 7 we obtain

$$\begin{cases} \dot{x} = -\frac{v_x}{Z} + \frac{xv_z}{Z} + xyw_x - (1+x^2)w_y + yw_z \\ \dot{y} = -\frac{v_y}{Z} + \frac{yv_z}{Z} + (1+y^2)w_x - xyw_y - xw_z \end{cases}$$
(11)

which can be written:

$$\dot{\mathbf{x}} = L_x v_c \tag{12}$$

where the interaction matrix L_x is given by

$$L_{x} = \begin{pmatrix} -\frac{1}{Z} & 0 & \frac{x}{Z} & xy & -(1+x^{2}) & y \\ 0 & -\frac{1}{Z} & \frac{y}{Z} & (1+y^{2}) & -xy & -x \end{pmatrix}$$
(13)

In the matrix L_x , the value Z is the depth of the point relative to the camera frame. Therefore, any control scheme that uses this form of the interaction matrix must estimate or approximate the value of Z. Similarly, the camera intrinsic parameters are involved in the computation of x and y. We discuss this in more detail below. To control the six degrees of freedom, at least three points are necessary. If we use the feature vector x = (x1, x2, x3), by merely stacking interaction matrices for three points we obtain

$$L_x = \begin{pmatrix} L_{x1} \\ L_{x2} \\ L_{x3} \end{pmatrix} \tag{14}$$

In this case, there will exist some configurations for which L_x is singular [13]. Furthermore, there exist four distinct camera poses for which e = 0, i. e., four global minima exist for the error function ||e||, and it is impossible to differentiate them [14]. For these reasons, more than three points are usually considered.

In our work, the GG-CNN2 network outputs the pixels (u, v) of the grasping rectangle of the highest wrinkle but this rectangle is not centered in the image of the camera. The main purpose of this part of work (concerning visual servoing) is to "translate" the grasping rectangle in the image center to then move the manipulator forward (on the z-axis) to grasp the hospital gown.

3. Experimental Set-Up

An RGB endoscopic camera that is mounted in the gripper of a Kinova Gen3 arm was used to capture photographs of the hospital gown (see Figure 2). The camera is mounted on the bottom part of the gripper and it is parallel to the wrinkle. The hospital gown position was changed during the trials. The dataset, collecting the images of the hospital gown, was trained on a PC running Ubuntu 18.04 LTS. The code was written in Python and it will be publicly available. The library mainly used for the implementation is Pytorch and the experiments are in real-time.

3.1. Dataset

We added jitter and white noises, salt and pepper noise, and median blur to our dataset because it is quite small compared to datasets like the Cornell dataset and the Jacquard dataset used to train the GG-CNN2. We obtained a dataset of roughly 3000 photos following the augmentation (the original dataset consists of 121 RGB images). The hospital gown is displayed in isolation on a soft tabletop scenario with a neutral background in the photographs that make up the dataset. Rectangles with clutching hands are used to manually label the figures. Then, we divided the training and testing of the GG-CNN2 by 80/20. We trained the network by altering several parameters in order to find the optimal network design. (Table 1).

4. Results

In this section, the results related to detecting the grasping point of the highest wrinkle of a folded hospital gown are shown.



Figure 2: The experimental set-up: it is composed by a Kinova Gen3 arm, a costumized gripper with an endoscopic camera mounted on it, and a hospital gown.

4.1. Qualitative Results

To test the method that combines GG-CNN2 with visual servoing approaches, 20 trials were carried out to see if the manipulator could identify the first wrinkle of the folded hospital gown. From the tests performed, we achieved 90% of accuracy.

In Figure 3, the four steps related to the identification of the grasping point on the highest wrinkle of the hospital gown are shown. At the beginning the robot is in its initial position; then the robot, conbining the ouptut of visual servoing and of the GG-CNN2, approaches the hospital gown until it arrives in detecting the highest wrinkle.

5. Discussions

In this section, the discussions related to the detection of the highest layer grasping point of the hospital gown are pointed out.

Concerning the experimental trials, issues appeared when the robot did not move correctly due to some mechanical problems or when the GG-CNN2 does not predict good grasping rectangles (on the highest wrinkle of the folded cloth) or when the visual servoing ouput is wrong. Future works should study in depth different approaches that involve newer networks compared to

Table 1

Internal parameters of the GG-CNN2.

batch-size	18
epochs	100
batches-per epochs	1000
val-batches	250



Figure 3: Example of identification of the highest wrinkle of the folded hospital gown.

our approach to obtain higher accuracy and a better performance.

6. Conclusions

The study and development of AI-based strategies for fabric manipulation capabilities were provided in this paper; specifically, the primary subject covered in this work was the identification of gripping spots for manipulating clothing.

The goal of this study was to identify the location where the highest wrinkle of a folded hospital gown could be observed (from a later perspective). The wrinkle grabbing point is located using a Generative Grasping Convolutional Neural Network (GGCNN2), and the manipulator's approach to the fabric is determined using a visual servoing algorithm that considers the GGCNN2's input. During the experimental set-up, we benchmarked this method on the Kinova Gen3 arm and achieved a validation accuracy of 90%. The trained accuracy is 98%.

Future research should be done in-depth on the long-term goal of examining how well AI-based tactics can manipulate fabrics.

References

- J. Qian, T. Weng, L. Zhang, B. Okorn, D. Held, Cloth region segmentation for robust grasp selection, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2020, pp. 9553–9560.
- [2] Z. Zhao, Y. Zhou, Clothing fabric automatic recognition, in: 2018 10th International

Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), volume 1, IEEE, 2018, pp. 24–27.

- [3] A. Caporali, G. Palli, Pointcloud-based identification of optimal grasping poses for cloth-like deformable objects, in: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), volume 1, IEEE, 2020, pp. 581–586.
- [4] A. Caporali, W. B. Bedada, G. Palli, A cyber-physical system for clothes detection, manipulation and washing machine loading, in: 2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS), IEEE, 2021, pp. 519–524.
- [5] L. Sun, G. Aragon-Camarasa, S. Rogers, J. P. Siebert, Autonomous clothes manipulation using a hierarchical vision architecture, IEEE Access 6 (2018) 76646–76662.
- [6] D. Morrison, P. Corke, J. Leitner, Closing the loop for robotic grasping: A real-time, generative grasp synthesis approach, arXiv preprint arXiv:1804.05172 (2018).
- [7] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 3431–3440.
- [8] Y. Jiang, S. Moseson, A. Saxena, Efficient grasping from rgbd images: Learning using a new rectangle representation, in: 2011 IEEE International conference on robotics and automation, IEEE, 2011, pp. 3304–3311.
- [9] J. Redmon, A. Angelova, Real-time grasp detection using convolutional neural networks, in: 2015 IEEE international conference on robotics and automation (ICRA), IEEE, 2015, pp. 1316–1322.
- [10] S. Kumra, C. Kanan, Robotic grasp detection using deep convolutional neural networks, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2017, pp. 769–776.
- [11] D. Park, S. Y. Chun, Classification based grasp detection using spatial transformer network, arXiv preprint arXiv:1803.01356 (2018).
- [12] I. Lenz, H. Lee, A. Saxena, Deep learning for detecting robotic grasps, The International Journal of Robotics Research 34 (2015) 705–724.
- [13] H. Michel, P. Rives, Singularities in the determination of the situation of a robot effector from the perspective view of 3 points, Ph.D. thesis, INRIA, 1993.
- [14] M. A. Fischler, R. C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (1981) 381–395.