

RESEARCH ARTICLE

Open Access



# Dual edge classifier for robust cloth unfolding

Antonio Gabas<sup>1,2,3\*</sup> , Yasuyo Kita<sup>2,4</sup> and Eiichi Yoshida<sup>1,2</sup>

## Abstract

Compared with more rigid objects, clothing items are inherently difficult for robots to recognize and manipulate. We propose a method for detecting how cloth is folded, to facilitate choosing a manipulative action that corresponds to a garment's shape and position. The proposed method involves classifying the edges and corners of a garment by distinguishing between edges formed by folds and the hem or ragged edge of the cloth. Identifying the type of edges in a corner helps to determine how the object is folded. This bottom-up approach, together with an active perception system, allows us to select strategies for robotic manipulation. We corroborate the method using a two-armed robot to manipulate towels of different shapes, textures, and sizes.

**Keywords:** Cloth manipulation, Deformable objects, Unfolding, Robot cloth unfolding

## Introduction

In recent years, robots have contributed to a significant increase in the automation of industrial tasks. However, the level of automation of household tasks has yet to become commonplace. The demand for robots capable of assisting with household tasks is likely to increase in parallel with aging global populations, a demographic phenomenon caused by improved life expectancies and dropping birth rates. One operation that is central to many household tasks, including laundry, assisted dressing, and bed making, is the manipulation of cloth items. This skill, which is simple for most humans, is actually very difficult for robots to perform. The difficulty of cloth manipulation lies in the deformability, nonlinearity, and low predictability of the behavior of the materials. Because of their deformable nature, compared with rigid objects, cloth objects are also inherently difficult for robots to recognize. This is why it is often necessary to completely unfold cloth items prior to starting a task. An unfolded garment is easier to recognize and manipulate

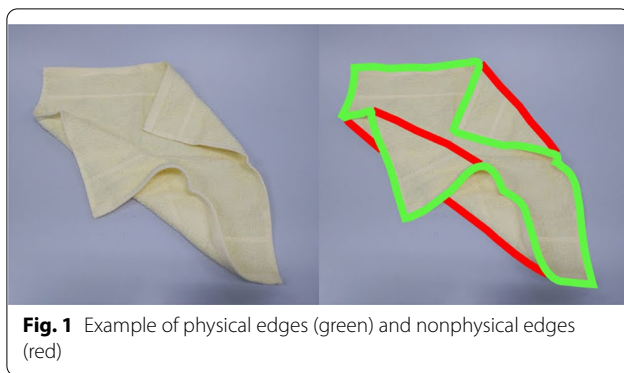
because a robot can then approximate the shape to a model or locate interest points like corners.

A common method of cloth unfolding is to lay the garment flat on a surface and unfold it, as in a pick-and-place problem [1–4]. In [3], similar to our method, the authors present an analysis of the types of corners in order to find strategies for unfolding. By contrast, our approach does not require a table or any flat surface, and involves simply grasping one point of the garment, lifting it into the air, and letting it hang from that point by the effect of gravity.

In this paper, we deal with a rectangular piece of cloth as a basic problem to investigate. Typical methods used to open such garments while hanging require locating predefined points and grasping them [5, 6]. However, because there are often hidden folds, we analyze the depth of the garment's edges instead of searching for specific points, which allows us to extract information for forming a manipulation strategy. We distinguish between two types of edges: those that belong to the hem of the garment, which we call physical edges, and the remaining nonphysical edges, often formed by folds. Figure 1 shows an example of this edge classification. In the image on the right, the physical edges are marked in green and the nonphysical edges are marked in red. Locating physical edges is very useful for finding grasping points and to better

\*Correspondence: a.gabas@aist.go.jp

<sup>1</sup> CNRS-AIST JRL (Joint Robotics Laboratory), IRL3218, Tsukuba, Japan  
Full list of author information is available at the end of the article



**Fig. 1** Example of physical edges (green) and nonphysical edges (red)

understand the shape of the garment. Opening the garment requires locating two corners formed by physical edges, which we call physical corners. These two corners should be consecutive i.e., connected by the same physical edge. Once located, grasping each corner with one hand leads to unfold the garment.

The configuration of edge types in the whole garment reveals some patterns. However, the high dimensionality of clothing items makes it very difficult to find global features that could identify an edge as physical or not. On the other hand, local features around edges tend to show slight differences between physical and nonphysical edges.

Therefore, to classify the edges, we propose a system that combines the results from two classifiers: a local one that selects a small patch around a pixel as an input and a global one whose input is the whole image. Finally, we present a categorization of the types of corners found in the image of the garment and use this categorization in an algorithm to actively choose the best robot action for opening the garment.

The main contributions of this work are the following:

- A combined local and global classifier capable of determining edge types.
- An algorithm that chooses the best course of action towards unfolding a garment according to its state, which is inferred from the types of edges. The algorithm is capable of locating physical corners even when they are occluded.

We apply this algorithm to the case of unfolding different towels and show how this skill can be applied to other garments.

In the “[Related work](#)” section, we present several related approaches to cloth manipulation. The “[System overview](#)” section describes the categorization of folding patterns for a cloth held in the air. In the “[Cloth edge classification](#)” section, we provide a detailed description

of the edge type classifier. The “[Action planning](#)” section describes the algorithm that chooses the best action according to each folding pattern. Finally, in the “[Experiments](#)” section, we validate our system using different examples of rectangular cloth items.

## Related work

### Semantic edge detection

The use of learning techniques to detect edge information allows to perform edge segmentation with respect to more subjective criteria than classical methods such as Canny [7]. In [8], they use random forests to learn a mid-level representation based on object contours called sketch tokens. Similarly, in [9], they use boosted decision trees to extract depth maps.

Semantic edge detection goes one step further by turning this binary classification into a multiclass problem. CASEnet [10] proposes a network that classifies each pixel in the edge to one or more semantic labels. They demonstrate the results using Semantic Boundaries Dataset and Cityscapes datasets. The work in [11] improves the results of CASEnet by doing full deep supervision.

This paper expands on previous work [12], which was, to our knowledge, the first attempt to teach machines semantic edge segmentation for the perception of deformable objects. In [12], edge detection is successful in finding the corner to be grasped to unfold a towel. In cases where the corner is hidden, however, the unfolding of the garment cannot be completed. In this work, we present a detection and manipulation technique that allows us to identify and grasp a corner that is hidden behind curled up cloth, and then bring the garment to a complete unfolded state.

### Cloth manipulation

Feature detection is the approach most commonly used to locate a point to be grasped for unfolding. In [2] they detect the hem and propose grasping points that are later manually selected. If the garment lies on a surface and only presents some wrinkles as in [13], topology analysis can be used to generate a strategy for flattening. Yuba et al. [14] uses a “pinch and slide” action that involves locating a corner, grasping it, and then pinching the edge close to it before finally sliding toward the next corner.

With the advent of deep learning, several studies have tried to solve the cloth manipulation problem. Triantafyllou et al. [15] uses horizontal edges and junctions found in the depth images as grasping points. This approach considers all of the depth edges without distinguishing whether they really belong to a physical edge or are produced by folds or noise. This can lead to selecting incorrect grasping points.

Doumanoglou et al. [5] uses random decision forests to learn to find specific points of garments (e.g., the shoulders in a t-shirt or corners in a cloth). To solve a problem where the points are not visible, the authors use a probabilistic action planner to acquire new views of the object by rotating it. However, soft garments, tend to wrinkle in a way that can hide big parts of the object, including these specific points (see Fig. 2). In those cases, such points cannot be found, even by rotating the garment 360 degrees.

Similarly to Doumanoglou's method, Corona et al. [16, 17] detect specific points for each garment using deep convolutional neural networks to find the grasping points on a garment after a neural network identifies the garment type.

In the work by Hu et al. [18], the authors hold the unknown garment to form one shape from a small set of limited shapes and match it with ones in a database prepared in advance. For bringing the item to such a limited shape, they first grasp the garment by the lowest hanging point and then by the farthest point from the vertical axis through the holding position, considering that the farthest point should be a characteristic point such as a shoulder. This second grasping strategy may not be applicable to all kinds of garments especially in the case of soft garments.

## System overview

### Cloth shape observations

The main problem with working with deformable objects is that the number of configurations they can take is infinite. In order to limit the possible configurations of the garment, we leverage a simple observation to grasp the garment by one of its corners. If the garment is grasped



**Fig. 2** Some methods that try to locate specific points like corners; however, cloth tends to curl over, which can hide these points. The green lines are the painted physical edges. Physical edges are detected in the RGB image using color segmentation and are used to generate labels. During training, we only use the depth image, and the neural network never receives this color information

by any random point, then the lowest point of the garment from a frontal view corresponds to one of the corners (see Fig. 3). The same observation was used in [5, 17].

Regrasping by that lowest points ensures that the garment is grasped by one of its corners. We thereafter assumed this to be the initial position for all of the experiments. After grasping one corner, we gained insight by looking at how humans manipulate cloth before unfolding it. We found that the first action is often to look for any other contiguous corner and grab it. If the corner is not visible, humans tend to grasp one of the edges and slide the hand towards the corner.

### Analysis and categorization of cloth folding patterns

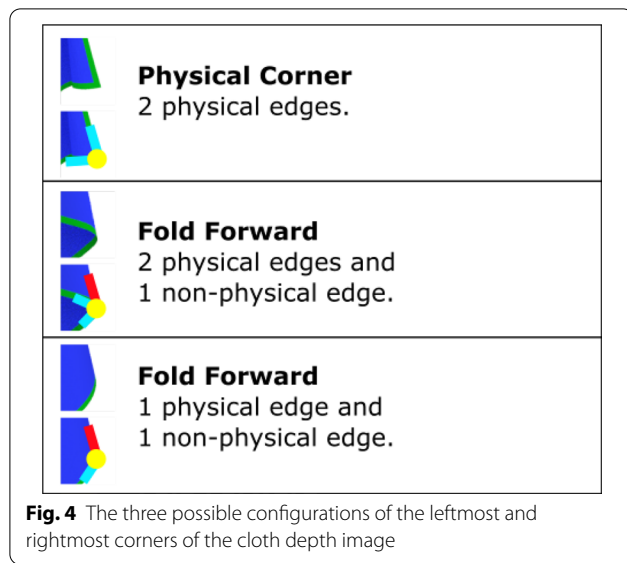
We present a categorization of the possible configurations of a cloth item. Next, we use the result to reveal and grasp the hidden corner. To understand how the garment is folded, expanding on the work in [12], we focus on distinguishing between physical and nonphysical edges, as mentioned earlier in the “Introduction” section.

Based on the edge types, the type of corner made by the edges can be classified. In the method proposed in this paper, we focus on the lateral (leftmost and rightmost) corners of a cloth held in the air and identify its type, as shown in Fig. 4.

With one physical corner being held, the bottom point always corresponds to the opposite corner. For the other two corners, there are three possible states: visible, curled forward, and curled backward. To evaluate the state of the corners of a garment, we observe the leftmost and rightmost corners of the perimeter as shown in Fig. 5. If two physical edges are coming out of that corner, it is a real corner (e.g., the right corner in Fig. 5a). If one or more edges are nonphysical, then it is a pseudo corner. In the case of two edges coming out of the corner, the real corner is folded backward (e.g., left corner in Fig. 5a–c).



**Fig. 3** When grasping the garment from a random point, the lowest point from a frontal view corresponds to one of its corners



If three edges are coming out of the corner, the real corner is folded forward (e.g., right corner in Fig. 5b–e). In the case of a corner folding forward, the actual physical

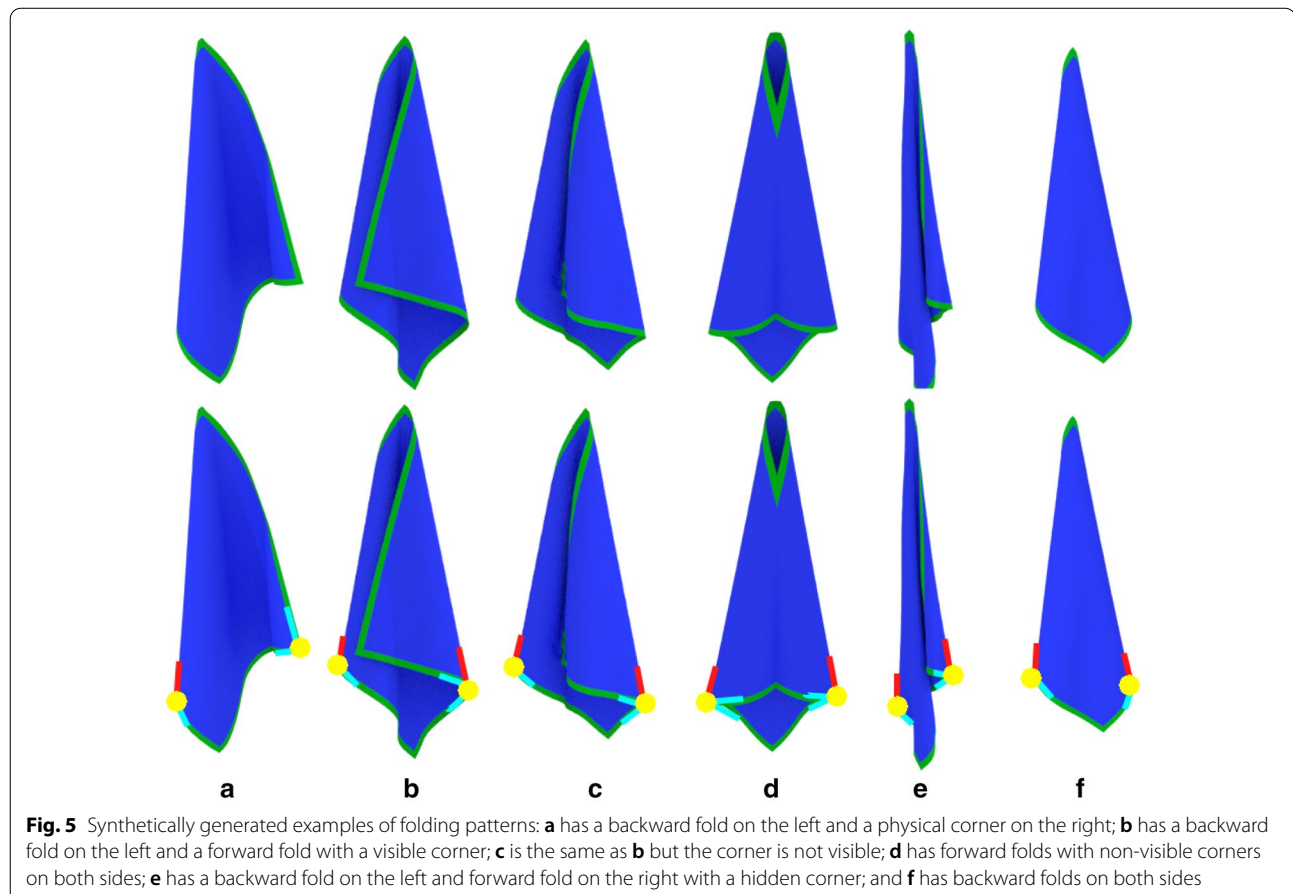
corners are either visible (e.g., right corner in Fig. 5b) or hidden (e.g., right corner in Figures 5c–e). In cases where it is hidden, further manipulation is needed to reveal it before grasping. Figure 6 shows the process that needs to be followed to identify the pattern in the leftmost and rightmost corners.

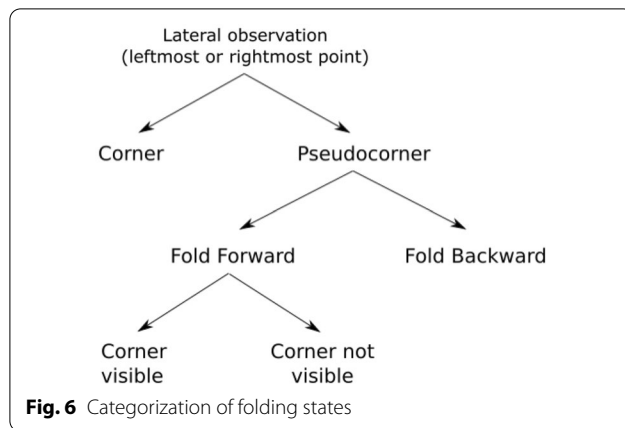
From this observation, we can see that it is possible to obtain crucial information about how the garment is folded simply by identifying the types of edges leading to the corners in these two points.

### Pipeline

Figure 7 shows the whole pipeline of the system. First, the robot takes the cloth to the initial position and then, from the depth image, the edges are extracted. Next, the leftmost and rightmost points are located and their folding pattern is classified according to the type and number of edges at each point. Finally, the robot executes an action according to the observation.

In the next sections, we explain the details of each stage.





### Cloth edge classification

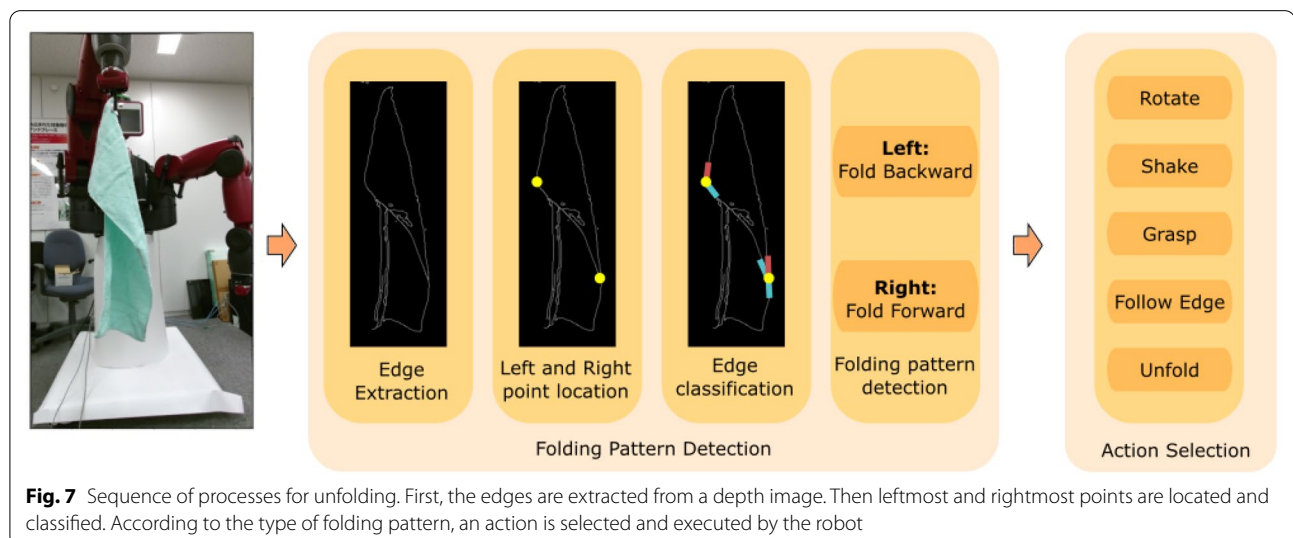
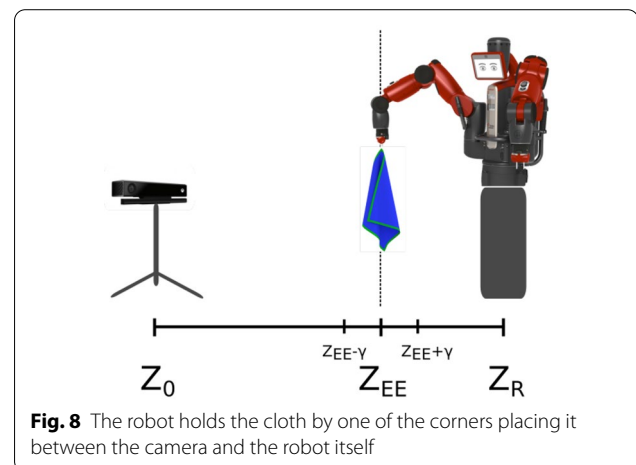
The vision system takes a depth image of a garment as input data and classifies its edges as physical or non-physical. It consists of two detectors: a local one and a global one. The local one only considers small patches in the image, around the point that it classifies. This is useful for the generalization of other garments, but it lacks the ability to consider the global structure in the current item. For this purpose, we introduce a global detector that takes into account the whole image as it classifies the pixels.

Training a neural network requires large quantities of labeled data. Manually labeling the physical edges in thousands of images is not feasible owing to time constraints. To overcome this, we use a semi-automatic dataset generation method. We paint the physical edges of a cloth item (as seen in Fig.2) and then with an RGB camera, we detect and automatically label these edges. It should be noted that we use the RGB images only to

generate the labels; this color information is never seen by the neural network, as it only uses depth information. Using this method, we are able to obtain hundreds of labeled images with minimal human intervention. The garment is hung from the robot end effector and rotated while the images are captured. After a full rotation of the garment, the shape of the garment is modified and another round of images is captured.

### Image acquisition and preprocessing

We use a Kinect One sensor placed as shown in Fig. 8. The sensor provides an RGB image matrix  $I(p)$  and a depth image matrix  $D(p)$ . Both cameras are calibrated so that each pixel  $p = (x, y)$  in the images corresponds to the same location in the real scenario. The camera is also





calibrated with the robot so that its position relative to the robot is known.

To remove the pixels that do not correspond to the cloth, we filter by depth, keeping only the pixels that are at a distance  $Z_{EE} \pm \gamma$  near the end effector (as shown in Fig. 8). Next, we extract the edges from the filtered image using the Canny algorithm [7]. We denote  $V_d$  as the set of pixels in the resulting binary image.

The RGB image is only used during training to generate label images  $\{\hat{Y}(p)_0 \dots \hat{Y}(p)_N\}$ . When we train using a cloth with painted edges, we segment each image by color to extract a binary image label in which  $\hat{Y}(p) = 1$  if the pixel  $p$  corresponds to a physical edge. Otherwise, it is zero.

### Local detector

As a local detector, we use the same structure as we did in our previous work [12]. Figure 9 shows the way the inputs and outputs to the network are arranged. For each pixel in  $V_d$ , a patch  $h(p)$  of size  $50 \times 50$  is extracted around that point from  $D(p)$ . The patch size was determined empirically by visually analyzing the images. It corresponds to a size that is big enough to contain some context surrounding the point and small enough to avoid capturing other nearby edges that could affect the classification. Batches of patches are fed into the neural network. After the input layer, we set a convolutional layer (Fig. 9a) with 32 convolutional kernels of size  $3 \times 3$  and stride 1. The next layer (Fig. 9b) is a batch normalization layer followed by a max pool layer of size 2 and rectifying linear unit (ReLU). This structure is repeated in the subsequent layers (see Fig. 9c–d), with a 64-kernel convolution of the same size. The last set of convolution layers (Fig. 9e–f) consists of 128 kernels of the same size as the previous ones. The

output of (f) is linearly rearranged, forming a vector of length 2048 (g), which is then passed to a fully connected layer of 500 neurons (h). Finally, the output layer (i) has two neurons that activate, indicating the probability of the pixel belonging to a physical or nonphysical edge.

For each batch of  $N$  samples  $X = \{h(p_0), \dots, h(p_N)\}$  (with  $p$  from the set  $V_d$ ), the neural network returns  $\{y(p_0), \dots, y(p_N)\}$  with  $y(p)$  being the probability of pixel  $p$  belonging to a physical edge. We then evaluate the binary cross entropy loss:

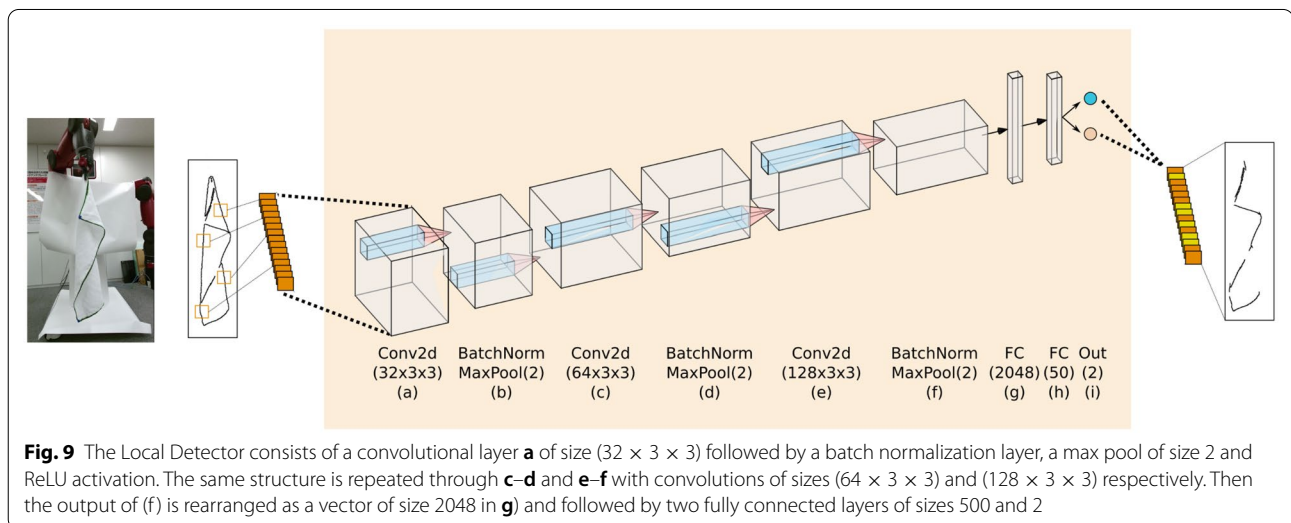
$$BCE_{loss} = \frac{1}{N} \sum_{p=0}^N -\hat{Y}(p) \log y(p) - (1 - \hat{Y}(p) \log (1 - y(p))) \quad (1)$$

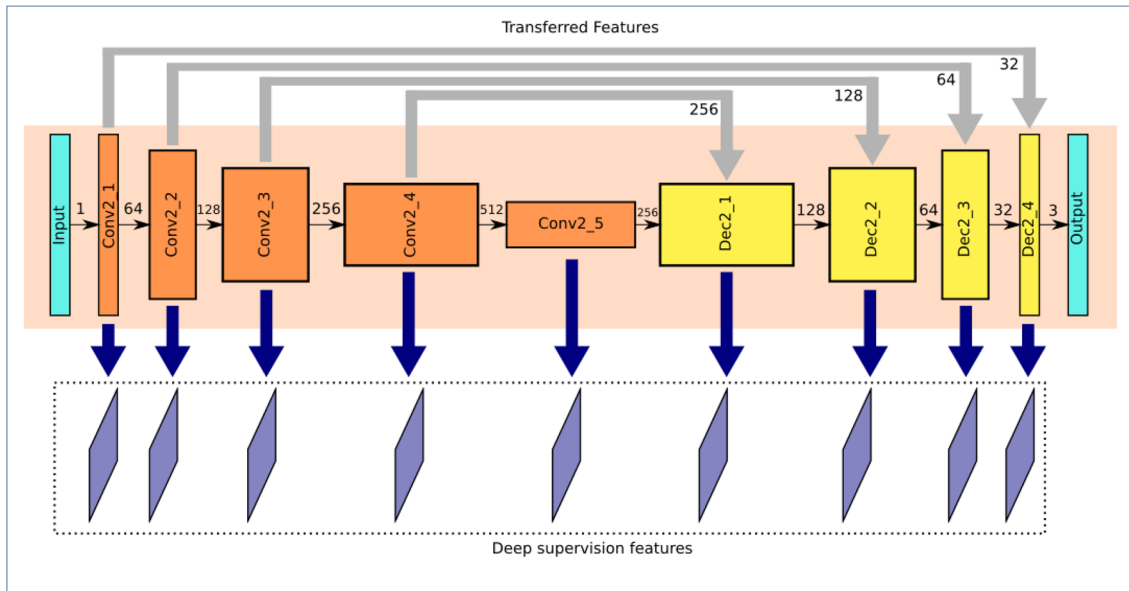
### Global detector

Since the local detector classifies pixels individually without taking into account the full cloth, it is susceptible of presenting discontinuities in an edge. To compensate this effect, we use a global detector that takes into account the whole image and classifies every pixel in the image by using a fully convolutional neural network.

Figure 10 shows the structure of the network. The orange boxes represent the feature maps at each convolution layer. The yellow boxes are the feature maps at each deconvolution layer merged with the features from early stages of the neural network (represented by the gray arrows). Each box follows the ResNet architecture [19] and is followed by a batch normalization layer and ReLU activation.

In this case, we formulate the problem as a multi-label problem. Each of the  $N$ -label images  $\bar{Y}_N^{(k)}$  contains  $K$  binary images, one for each of the  $K$  categories. We use





**Fig. 10** The global detector is a fully convolutional neural network with deep supervision. The orange boxes represent the feature maps at each convolution layer. The yellow boxes are the feature maps at each deconvolution layer merged with the transferred features from early stages (grey arrows). The blue arrows represent the feature extraction for deep supervision at each layer

$K = 3$  with one category corresponding to the physical edges ( $\hat{Y}$ ), one for the nonphysical edges ( $V_d - \hat{Y}$ ), and the rest of the pixels corresponding to the background.

The multi-label loss ( $ML_{loss}$ ) is defined as

$$ML_{loss} = \sum_{k=0}^3 \sum_p -\epsilon \bar{Y}(p)^{(k)} \log Y(p)^{(k)} - (1 - \epsilon)(1 - Y(p)) \log (1 - Y(p)^{(k)}) \quad (2)$$

To compensate for the skewness in the dataset, we use  $\epsilon$  and  $(1 - \epsilon)$ , which represent the percentage of non-edge and edge pixels respectively.

Similar to other works [10, 11] we perform supervision at each stage. Supervision layers (represented by blue lines in Fig. 10) extract feature layers at each stage. We denote the weights as  $W = \{w^0, \dots, w^n\}$  for each of the  $n = 9$  layers. The supervised loss is evaluated as the sum of the multi-label loss of each of the individual layers:

$$L_{supervision}(W) = \sum_{i=0}^n ML_{loss}(Y_{w_i}) \quad (3)$$

The final loss  $\mathcal{L}$  consists of the loss at the output layer and the supervision loss:

$$\mathcal{L} = ML_{loss}(Y_{out}) + \lambda L_{supervision} \quad (4)$$

where  $\lambda$  is a parameter between 0 and 1 that defines the weight of the supervision in the final loss.

### Output

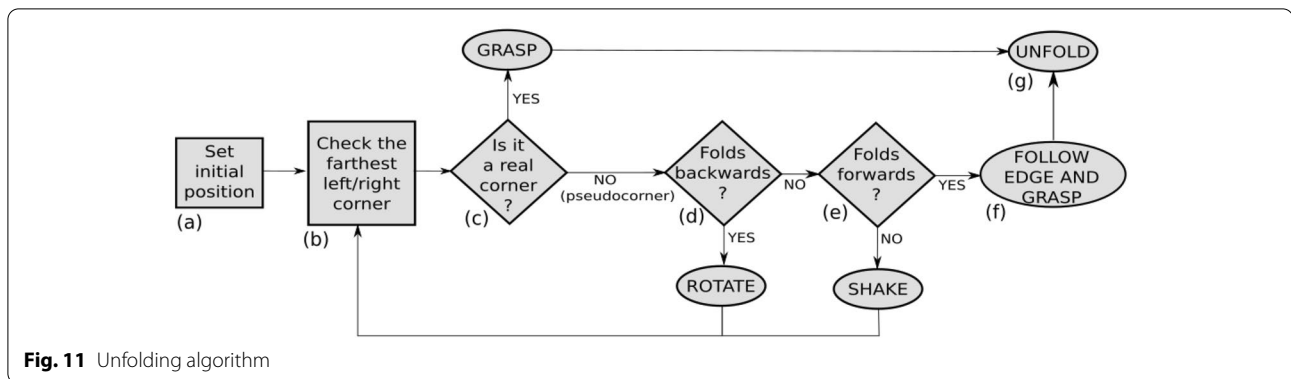
For each pixel, we have two classification results, one coming from the local detector and the other from the global one. We can ponder the outputs to give more importance to generalization or global structure by tuning  $\beta$ .

$$Y = \beta Y_{local} + (1 - \beta) Y_{global} \quad (5)$$

### Action planning

We introduce five actions the robot can take to accomplish the goal of unfolding the garment: Grasp, Rotate, Shake, Follow-Edge, and Unfold. The action Unfold, is the last action (as shown in Fig. 11g) and after that the garment should be in an unfolded state. Otherwise, the process starts again from the beginning. The Rotate action performs a rotation of the garment around the vertical axis by rotating the end effector of the robot arm that holds the cloth. The Grasp action is performed with the free hand by grasping a point on the garment, usually a corner. In the Shake action, the arm that is holding the garment allows it to spread vertically by the effect of gravity. Finally, Follow-Edge, moves the right hand's end effector along one of the physical edges.

The algorithm starts from the initial position i.e., the robot holding one of the garment's corners (Fig. 11a). We assume this position can be reached following the observation in the "Cloth edge classification" section. In other words, the robot first grabs the garment by any point and



then, with the other arm, grasps the lowest point, which corresponds to a corner.

Next, the farthest horizontal point is examined (Fig. 11b). A hanging garment will typically take the shape of a rough triangle, with its hypotenuse along the vertical axis. We showed in the “[Analysis and categorization of cloth folding patterns](#)” subsection that this outer corner is crucial to understanding how the garment is shaped.

We then analyze the edges that are connected to the farthest corner. If there are two physical edges (Fig. 11c), the corner in question is a real corner and we can proceed to grasp and then unfold it by extending it.

If it is a pseudo corner, we look more closely at the edge types and determine the type of folding, as shown in the “[Analysis and categorization of cloth folding patterns](#)” subsection. If the edge folds backward (Fig. 11d), the corner is probably behind the garment and the appropriate action is to rotate the garment to reveal the corner.

If it folds forward (Fig. 11e), we will move the end effector of the free arm along the trajectory defined by the physical edge to reveal the corner, grasp it, and then unfold the garment.

If the detected edges do not correspond to any of the defined categories, we will perform an action to shake the garment to loosen any folds and extend it by the effect of gravity. Then, we start the process again.

## Experiments

### Experimental setup

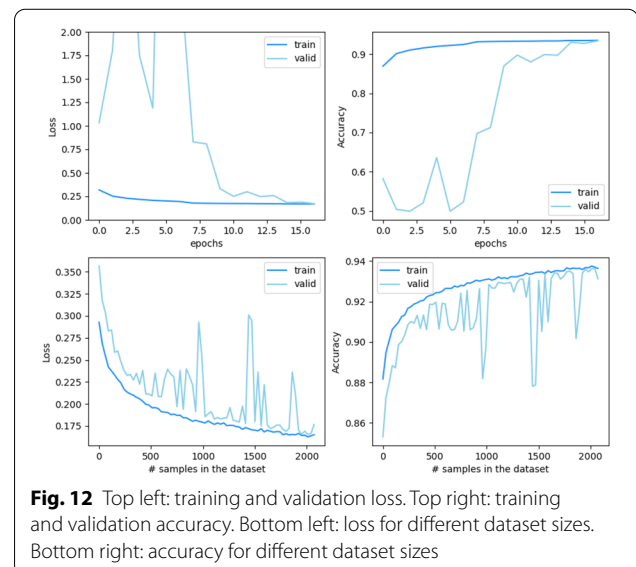
In all of the experiments, we use a Baxter robot with a Kinect One camera facing each other, as seen in Fig. 8. The neural networks are implemented using the open software Pytorch [20]. The GPU is an NVIDIA GTX1080 with 8 GB of memory, and the CUDA edition is 10.0. In all of the experiments, unless stated otherwise,  $\lambda = 1$  in Eq. 4 and  $\beta = 0.6$  in Eq. 5. Training was done with a garment with painted edges (see Figs. 2 and 3), from which

we extract more than 1600 images. This amounted to more than 3.2 million patches.

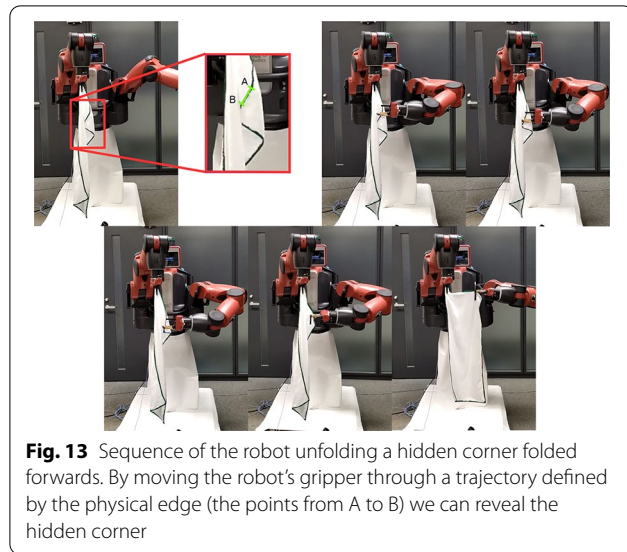
Each experiment begins with the robot holding a cloth with its right arm as an initial state, then taking actions to unfold it with the left arm (Fig. 13). The camera is calibrated and its position with respect to the robot is known. We conducted three types of experiments. First we analyzed the robot’s performance in edge classification and grasping for 20 attempts using the same garment (see Figs. 2 and 3). Then we validated the results of our method by having the robot unfold several previously unseen garments. Finally, to demonstrate the effectiveness of the global detector, we show an ablation study comparing the local + global detector with the local only detector from previous work [17].

### Training

The training progress is shown in Fig. 12. The top row shows the loss and accuracy during training and validation. We train for 15 epochs, stopping before any signs







**Fig. 13** Sequence of the robot unfolding a hidden corner folded forwards. By moving the robot's gripper through a trajectory defined by the physical edge (the points from A to B) we can reveal the hidden corner

of overfitting. To demonstrate that the amount of data gathered is enough, we trained the system with increasing amounts of samples in the dataset. The left graph in the bottom row of Fig. 12 shows that the loss quickly decreases as we increase the size of the dataset. After around 750 samples, the change in the loss relative to the dataset size decreases more slowly, indicating less significance of adding more data. The graph on the bottom right shows the accuracy, which inversely grows at a similar rate.

### Classification and grasping

In order to show the effectiveness of the method and determine the stage at which possible errors might occur, we performed 20 attempts to unfold a single garment (seen in Figs. 2 and 3). We studied these attempts to ascertain whether the edge classification had been produced correctly and the grasping and unfolding were successful. The results are summarized in Table 1. Figure 13 shows an example of unfolding when the inner corner is hidden. The robot followed the trajectory of the physical

edge to reveal the corner, grasp it, and successfully unfold the cloth. The video in Additional file 1 contains examples of robot cloth unfolding.

Table 1 shows the four possible outcome cases depending on the success or failure in corner classification and unfolding. A circle in corner classification column indicates that the corners were correctly classified in all the steps. The first row indicates that 75% of the times the unfolding was successful with correct corner classification in every step. The second row indicates that in the 10% of the cases in which the Edge classification was not successful, the Grasping was. This result is produced in cases where, somewhere in the process, there are errors in the classification, but after some action (like Rotate) the next step led to a correct classification and grasping. In this case, the corner classification was correct in 85.72% of the steps

Note that we do not differentiate between success in grasping and success in unfolding because a successful grasp led to successful unfolding in every attempt.

### Generalization

We tested the results of the system through experiments using four cloths of different sizes and textures that were not seen during training (shown in Fig. 14). The robot attempts to grasp each garment 20 times and the success rate of edge classification and unfolding are shown in Table 2. For each attempt, we consider that the corner classification is successful if it was correct in all the steps, and the unfolding is considered successful if the physical edges form a square. The success rate represents the percentage of success in the 20 attempts. The cloth A (seen in Fig. 14) reaches 100% in corner-type classification. This garment is, in fact, the most similar to the one used during training. Cloths B and D are smaller and have different folding patterns. Cloth B is the most different in terms of color texture and it has the lowest classification accuracy. That said, the cause of the lower corner classification success ratio is not the color texture, but the tendency of this cloth to curl up and hide its edges more often than others. Images with a physical edge present are correctly classified most of the time regardless of the

**Table 1** Outcomes of 20 grasping attempts

Corner classif.	Grasping unfold	Ratio
O	O	75%
X <sub>(1)</sub>	O	10%
O	X	15%
X	X	0%

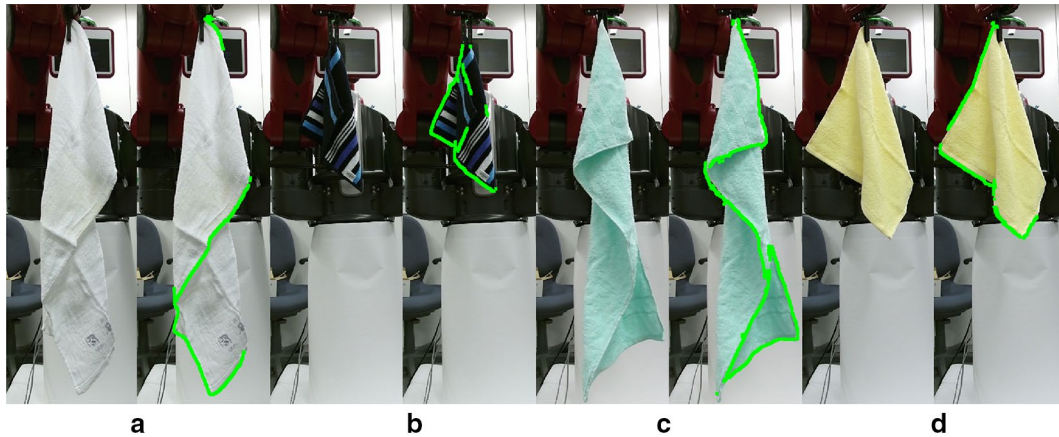
(1) Corner classification was correct in 85.72% of the steps

An "X" in a cell means the operation in that column was not successful, whereas a circle indicates success. The ratio indicates the percentage of attempts with that particular corner classification and unfolding outcome

**Table 2** Success ratios for different unseen cloths during training

Cloths	Corner Classif.	Unfolding
Cloth A	100%	85%
Cloth B	85%	80%
Cloth C	95%	95%
Cloth D	95%	85%

The middle column shows the success ratio in edge classification, and the right column the success ratio in grasping and unfolding



**Fig. 14** Examples of edge detection in unseen garments (Cloths **a** to **d** respectively)

**Table 3** Success ratios for different unseen cloths during training

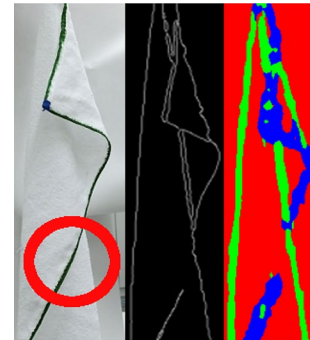
Cloths	Local	Global + Local
Cloth A	81%	88%
Cloth B	78%	86%
Cloth C	83%	89%
Cloth D	74%	89%

The middle column shows the percentage of correctly classified pixels using only the local classifier and the right column shows the result after the refinement with the global classifier

cloth's texture. Error cases tend to appear in cases with hidden edges. These are more difficult to classify, and an increased tendency of a cloth to curl is the main factor affecting the classification success ratios.

#### Ablation study

To show the benefits of using both global and local classifiers, we compare the percentage of correctly classified pixels in the edges when taking into account both local and global classifiers with the ablated version using only the local classifier as in [12]. For this experiment, not only the classification of the edges in the corner is considered, but the classification of all the edges in the image. The success ratio represents the ratio of pixels in the image's edges that are correctly classified to the total number of pixels in the image's edges. Table 3 shows the results for each cloth. Again, cloths A and C, which are the biggest in shape (the length is similar and they are rectangular) and more similar to the one used during training in terms of cloth texture, are the ones with the best accuracy. The global detector does not add a big increase in the accuracy, since the results of the local detector are already



**Fig. 15** When the edges are very close to another layer of cloth, this can lead to errors in the edge detection

high. Cloths B and D are shorter and more squared benefit from knowing the whole structure of the garment and significantly improve their results when taking into account the global classifier.

#### Failure cases

Figure 15 shows the most common example of failure in edge detection. Because we are exclusively using the depth image to detect the edges, having the edges very close to another layer of cloth can lead to failure in their detection. This kind of error, however, only tends to happen around the center of the cloth. The proposed method for analyzing the leftmost and rightmost corners generally avoids this kind of error because there is usually a background behind those points and not another layer of the cloth. There are two main possibilities to increase the accuracy in pixel detection. First, by improving the inputs, that is improving the resolution of the sensor or adding more channels (color RGB). The other possibility

is to focus in the design and optimization of a new model of neural network.

The most common cause of failure in manipulation and, also the main general cause of failure, is the incapability of finding a solution in the inverse kinematics for the grasping point. This is more common when trying to reveal a hidden corner. To solve this we relaxed the tolerance of the goal position and orientation, and tried to find other configurations within a short distance and angle from the desired configuration. We also use an L-shaped gripper to grasp the edge at an angle, making it easier to find a solution for the inverse kinematics. In order to further improve the results, a more task-specific robot could be designed to better satisfy the task. However, we chose to use a two-arm robot with 7 degrees of freedom in each arm which is a common design.

## Discussion

We presented a comparison with our previous method that shows an increase in the accuracy of pixel classification, and most importantly solves a previously unsolved problem: revealing a hidden corner. The method in [5] is similar to ours in that it unfolds the garment while hanging. However, when the corner or feature to grasp is hidden and not found after a full rotation, they restart the whole process by regrasping the garment. We chose to take a different approach to find a strategy to reveal the hidden corner. Other methods like [15, 21] use a table to assist in the unfolding operation and are not directly comparable to ours.

## Conclusions

We have presented a method for manipulating cloth items that is based on reliable identification of the types of edges in a depth image. Using only depth information makes the algorithm robust to changes in color and texture. This also makes it possible to use the color information to generate a large number of labeled examples for further network training.

Our method recognizes how a cloth is folded by analyzing the types of edges connecting to the leftmost and rightmost corners in the depth image, which facilitates choosing the next appropriate action.

We employed both local and global classifiers to benefit from generalization of the former and the ability of the latter to take into account the whole structure.

The experiments demonstrated that, with a high success ratio (85%), the robot was able to grasp a corner of the cloth in order to unfold it even when the corner was not visible in the image. We also showed how the method can be expanded to include other types of cloth not seen during the training.

Contrary to methods that try to model the whole cloth item in order to manipulate it, we showed that finding and analyzing the edges is a promising way to understand how to manipulate an object with a robot. With further research, this method could be extended to other types of garments.

The main limitation is the restriction to rectangular cloths. Future work can solve this limitation by studying other common patterns found in the edges of other types of folded cloth. For example, a t-shirt often presents similar patterns, but more analysis and strategies are needed to deal with the sleeves.

## Supplementary Information

The online version contains supplementary material available at <https://doi.org/10.1186/s40648-021-00202-8>.

**Additional file 1.** Video explanation with examples of cloth unfolding.

## Acknowledgements

We thank dr. Yusuke Yoshiyasu for his expertise and assistance through this work and dr. Natsuki Yamanobe for providing the robot and hardware necessary for the experiments.

## Authors' contributions

AG and YK conceived and designed the study and experiments. AG conducted the experiments and wrote the paper. YK and EY provided advice during writing and critically revised the paper. All authors read and approved the final manuscript.

## Funding

This work was supported by a Grant-in-Aid for Scientific Research, KAKENHI (16H02885).

## Availability of data and materials

The datasets used and/or analyzed during the current study are available from the corresponding author upon request.

## Declaration

## Competing interests

The authors declare that they have no competing interests.

## Author details

<sup>1</sup>CNRS-AIST JRL (Joint Robotics Laboratory), IRL3218, Tsukuba, Japan. <sup>2</sup>National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Japan. <sup>3</sup>Department of Intelligent Interaction Technologies, University of Tsukuba, Tsukuba, Japan. <sup>4</sup>Department of Electrical Engineering, Faculty of Science and Technology, Tokyo University of Science, Noda, Japan.

Received: 18 January 2021 Accepted: 29 March 2021

Published online: 26 April 2021

## References

1. Sun L, Aragon-Camarasa G, Rogers S, Siebert JP (2015) Robot vision architecture for autonomous clothes manipulation 14(8):1–15 [arXiv:1610.05824](https://arxiv.org/abs/1610.05824)
2. Yamazaki K (2014) Grasping point selection on an item of crumpled clothing based on relational shape description. In: IEEE international conference on intelligent robots and systems, pp 3123–3128. <https://doi.org/10.1109/IROS.2014.6942994>

3. Triantafyllou D, Aspragathos NA (2011) A vision system for the unfolding of highly non-rigid objects on a table by one manipulator. *Lecture Notes in Computer Science*, pp 509–519
4. Yang PC, Sasaki K, Suzuki K, Kase K, Sugano S, Ogata T (2017) Repeatable folding task by humanoid robot worker using deep learning. *IEEE Robot Autom Lett* 2(2):397–403. <https://doi.org/10.1109/LRA.2016.2633383>
5. Doumanoglou A, Kargakos A, Kim T-K, Malassiotis S (2014) Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning. In: *Proc. IEEE international conference on robotics and automation (ICRA14)*, pp 987–993. <https://doi.org/10.1109/ICRA.2014.6906974>
6. Li Y, Xu D, Yue Y, Wang Y, Chang S-F, Grinspun E, Allen Peter K (2015) Abstract—deformable: regrasping and unfolding of garments using predictive thin shell modeling. In: *International conference on robotics and automation*
7. Canny J (1986) A Computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 8(6):679–698. <https://doi.org/10.1109/TPAMI.1986.4767851>
8. Lim JJ, Zitnick CL, Dollar P (2013) Sketch tokens: a learned mid-level representation for contour and object detection. *IEEE Conf Comput Vis Pattern Recogn.* <https://doi.org/10.1109/CVPR.2013.406>
9. Dollar P, Zhuowen Tu, Belongie S (2006) Supervised learning of edges and object boundaries. *IEEE Comput Soc Conf Comput Vis Pattern Recogn* 2:1964–1971. <https://doi.org/10.1109/CVPR.2006.298>
10. Yu Z, Feng C, Liu M-Y, Ramalingam S (2017) CASENet: deep category-aware semantic edge detection. [arXiv:1705.09759](https://arxiv.org/abs/1705.09759)
11. Liu Y, Cheng M-M, Fan D-P, Zhang L, Bian J-W, Tao D (2018) Semantic edge detection with diverse deep supervision. [arXiv:1804.02864](https://arxiv.org/abs/1804.02864)
12. Gabas A, Kita Y (2017) Physical edge detection in clothing items for robotic manipulation. In: *18th international conference on advanced robotics (ICAR)*, pp 524–529. <https://doi.org/10.1109/ICAR.2017.8023660>
13. Sun L, Aragon-Camarasa G, Rogers S, Siebert JP (2015) Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In: *Proceedings—IEEE international conference on robotics and automation (June)*, pp 185–192. <https://doi.org/10.1109/ICRA.2015.7138998>
14. Yuba H, Yamazaki K (2014) Unfolding an item of rectangular clothing using a single arm and an assistant instrument. *IEEE/SICE Int Sympos Syst Integr SII* 2014:571–576. <https://doi.org/10.1109/SII.2014.7028102>
15. Triantafyllou D, Mariolis I, Kargakos A, Malassiotis S, Aspragathos N (2016) A geometric approach to robotic unfolding of garments. *Robot Autonom Syst* 75:233–243. <https://doi.org/10.1016/j.robot.2015.09.025>
16. Corona E, Alenyà G, Gabas A, Torras C (2018) Active garment recognition and target grasping point detection using deep learning. *Pattern Recogn* 74:629–641. <https://doi.org/10.1016/j.patcog.2017.09.042>
17. Gabas A, Corona E, Alenyà G, Torras C (2016) Robot-aided cloth classification using depth information and CNNs. In: *Articulated motion and deformable objects*, pp 16–23
18. Hu J, Kita Y (2015) Classification of the category of clothing item after bringing it into limited shapes. vol 588–594. <https://doi.org/10.1109/HUMANOIDS.2015.7363422>
19. Targ S, Almeida D, Lyman K (2016) Resnet in resnet: generalizing residual architectures. *CoRR*. [arXiv:1603.08029](https://arxiv.org/abs/1603.08029)
20. Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Kopf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) PyTorch: an imperative style, high-performance deep learning library. *Adv Neural Inform Process Syst* 32:8024–8035
21. Tanaka D, Arnold S, Yamazaki K (2018) EMD Net : an encode-manipulate-decode network for cloth manipulation. *IEEE Robot Autom Lett* 3(3):1771–1778

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

---

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)