Active garment recognition and target grasping point detection using deep learning

Enric Corona[a], Guillem Alenyà[a], Antonio Gabas[a,b], Carme Torras[a]

[a] *Institut de Robòtica i Informàtica Industrial CSIC-UPC*
C/ Llorens i Artigas 4-6, 08028 Barcelona, Spain.

[b] *National Institute of Advanced Industrial Science and Technology (AIST)*

Abstract

Identification and bi-manual handling of deformable objects, like textiles, is one of the most challenging tasks in the field of industrial and service robotics. Their unpredictable shape and pose makes it very difficult to identify the type of garment and locate the most relevant parts that can be used for grasping. In this paper, we propose an algorithm that first, identifies the type of garment and second, performs a search of the two grasping points that allow a robot to bring the garment to a known pose. We show that using an active search strategy it is possible to grasp a garment directly from predefined grasping points, as opposed to the usual approach based on multiple re-graspings of the lowest hanging parts. Our approach uses a hierarchy of three Convolutional Neural Networks (CNNs) with different levels of specialization, trained both with synthetic and real images. The results obtained in the three steps (recognition, first grasping point, second grasping point) are promising. Experiments with real robots show that most of the errors are due to unsuccessful grasps and not to the localization of the grasping points, thus a more robust grasping strategy is required.

Keywords: garment classification, garment grasping, deep learning, depth images

1. Introduction

Robots are expected to continue gaining autonomy so as to be increasingly helpful in our daily lives in areas as diverse as elderly care, housework or maintenance. Currently, their ability to perform

repetitive tasks very precisely is essential for process automation, which combined with their high throughput makes them the optimal option for industry. The use of robots has been extended to places inherently unsafe for humans, such as space exploration or military tasks. Other types of robots can be found in the medical field because of their high precision and reliability.

Nevertheless, the diversity of tasks robots can be found working on contrasts with their still low self-sufficiency. They can be accurately programmed to achieve a list of goals in a predefined setting, but their performance becomes uncertain when confronted with new situations. Thus, for robots to be useful in service and assistive contexts, they need to be endowed with quite different abilities than their industrial counterparts. They should be made intrinsically safe to people, easy to teach by non-experts, able to manipulate not only rigid but also deformable objects, and they must be highly adaptable to non-predefined and dynamic environments [**?**].

In particular, handling deformable objects requires dealing with high-dimensional configuration spaces, not just the 6D pose space of rigid objects. Devising robust procedures that are capable of adapting to new situations constitutes an important challenge for robotics. For example, bringing a garment from an unknown configuration to a reference one from which a given task can be performed, such as folding it, is a very demanding objective. It does not only requires complex perceptions, but also proper procedures to interpret the information captured and abstract the relevant parts. Some recent machine learning algorithms –such as deep learning– hold promise to tackle the above-mentioned challenge, and in this paper we explore a possible approach along this line.

In this paper we propose a hierarchy of three Convolutional Neural Networks (CNN) to carry out garment identification and garment grasping. The envisaged process, explained in more detail in section 3, is as follows. First, one robot arm grasps a garment from any point and shows it to an RGB-D camera. We recognize the garment using the *first* CNN described in section 4. Then, the visibility and location of two reference grasping points are identified using a *second* CNN.

At this point, we formulate a loss function to train a CNN that learns to be invariant respect to the order of the points predicted. The exact formulation is described in section 5. This allows to decide to turn the garment when the points are occluded, or to grasp the garment from one of those points with the other robot arm. Observe that the set of possible configurations is still

huge, as the garment can be grasped from any point. Next, we locate the second predefined point with a *third*, more specialized, CNN and grasp the garment by that point. In this case, the set of possible configurations is reduced as the garment has been grasped from a known grasping point. To train the different networks we propose to use a different combination of simulated and real images. Observe that the ground truth for the grasping points, specially the non-visible ones, is hard to obtain with real images and straightforward in simulated ones.

Differing from the usual approaches that repeatedly re-grasp the piece of clothing until ensuring that it is in a small range of poses, our goal is to perform a one-shot grasping that mimics a more natural human manipulation. To achieve this goal we propose using two CNNs per cloth after recognition. We finally show in section 6 our garment identification rate compared to other works, and the results of the garment manipulation pipeline.

2. Related Work

A significant amount of research has focused on detecting and identifying deformable objects. The process of isolating a piece of clothing from a pile was identified as one of the first tasks to be solved for laundry manipulation in a pioneer work by Kaneko and Kakikura [? ?]. Their initial paper recognized three categories: shirt, pants, and towels. Previously to recognition, works such as [?] by Colome et al. study the grasping process from a pile of clothing, implementing a method to determine whether the robot gripper is holding just one garment or several of them together. Other methods consider 3D shape recovery of deformable surfaces from images, to understand and track deformable structures [?].

Recent works usually handle image or pointcloud databases taken by a kinect sensor or a pair of stereo cameras. The main difference between them lays in the identification method. Willimon et al. [? ?] used interactive perception to recognize and classify different small garment types such as socks and short pants, but their approach relies on a color-based image segmentation that may fail when seeing fully textured clothes. Other works [? ?] demonstrate the ability to recognize poses of different garments by matching them to a precomputed database. Li et al. [?] use a SIFT descriptor that only needs an input image and, therefore, they achieve a high-speed recognition of several pieces of clothing: sweaters 85%, jeans 70%, and shorts 90%.

Another common approach is based on comparing volumetric features, such as Li et al. [?] proposing to reconstruct a 3D model by using several images of the garment. They extract volumetric features and match them to an offline database. However, their method requires rotating the garment 360º, which slows the recognition process. In the same manner, KinectFusion [?] could be used on different deformable objects to directly compare the 3D model to the pre-recorded database, which also requires powerful computational resources.

CNNs have significantly improved the state-of-art in a wide range of topics, including vision[? ?] and natural language processing [?]. In this paper, we take advantage of their perception power to make progress in manipulation of deformable objects. Many works have used CNNs to classify[? ?], detect[?] and estimate pose[? ?] from objects. However, few of them consider deformable objects.

Mariolis et al. [?] used CNNs on depth images to obtain an accuracy rate of 89.38 % in the identification of shirts, pants and towels. Using an analogous approach, in our preliminary work [?] we achieved an accuracy rate of 92 % when classifying four pieces of clothing. Our approach was to use CNNs on several depth images of real garments obtained at a close distance, turning the garment to make the approach robust to occlusions.

Manipulation of deformable objects appeared in the literature more recently, when Osawa et al. [?] first proposed using two robotic arms to unfold a garment. In order to find grasping points, they applied color-based segmentation with a clean background, which only works for garments with a unique color. Maitin-Shepard et al. [?] proposed to detect the corners of a piece of clothing using geometric cues, and then apply a sequence of grasps ending with the garment always in the same position. Their approach works for towels, whose known configuration follows from being grasped by their corners. They show that the lowest point of a towel, after some re-grasps, is always a corner. This allows to fold previously unseen towels without the need to compile a database. Assuming this as a valid hypothesis, our method aims to determine where the grasp points are located for more complex garments.

Similarly, Triantafyllou et al. [?] depicts a method to unfold garments grasping them from two points on their outline. That makes the garment be in approximately flat surface. Then they match the cloth to a set of foldable templates using shape analysis techniques. Using the established

correspondences with the template's landmark points the garment is re-grasped by such two points that it will naturally unfold in a spread out configuration.

Ramisa et al. [**?**] present a system that uses a very efficient shape descriptor, named FINDDD, combining depth and color to find good grasp candidates in a cloth lying on a table. They cannot disambiguate between several clothing poses using more than one view nor choose known grasping points that lead to reference poses. Nevertheless, their work is a good starting point for ours, to identify and manipulate the garment in a desired way. In a posterior work[**?**], they show that appearance is a very informative feature. Combining it with 3D information they achieve better performance to detect suitable grasping points in clothes.

Other approaches to manipulate garments involve some kind of machine learning algorithm to decide where the desired grasp point is. Doumanoglou et al. [**?**] proposed the first method to unfold regular-size clothing. They build Hough forests using the garments' depth images, trained with a set of images manually taken and labelled, which may be a very time-consuming task method to implement in practice. It may also lead to worse results when seeing new fabrics, although it achieves impressive rate of success within their database. More recently, they propose a complete pipeline to fold a pile of clothes [**?**].

Similarly to us, Li et al. [**?**] make use of a physics engine to create a database to train an algorithm that recognizes the garment's pose, matches it to a pose achieved with the synthetic 3D model of the same garment, and relates the grasping point of the synthetic image to the real one. After some re-grasps, the process stops when the pose is matched to the reference configuration. Our proposal differs from theirs in that we aim at avoiding costly re-grasping.

3. System overview

Our objective is to develop the required perceptions enabling a robot to execute specific tasks, such as folding, starting from a pile of clothing laying on a table. To do so we need the robot to grasp one garment from predefined grasping points, selected depending on the task. In our scenario, we assume that the task of isolating one garment from a pile can be solved using well-known methods [**?** **?** **?** **?**] and simplify the task by having a single garment on the table. Once it has been grasped, the robot places it in front of a camera, where it will be identified.

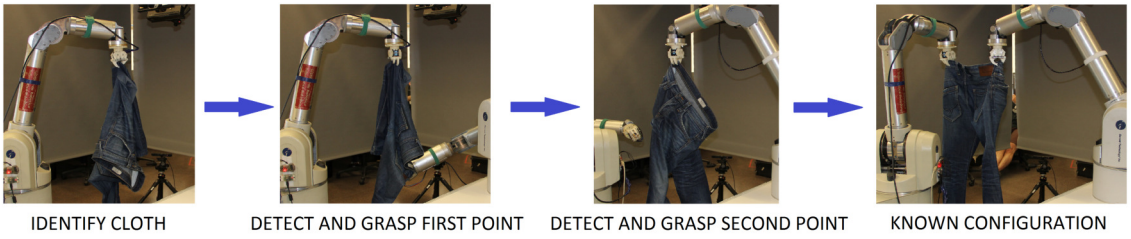


Figure 1: Process to move from holding an unidentified grasped garment to holding it from the predefined points. At each detection step the robot rotates the garment until the grasping points are visible.

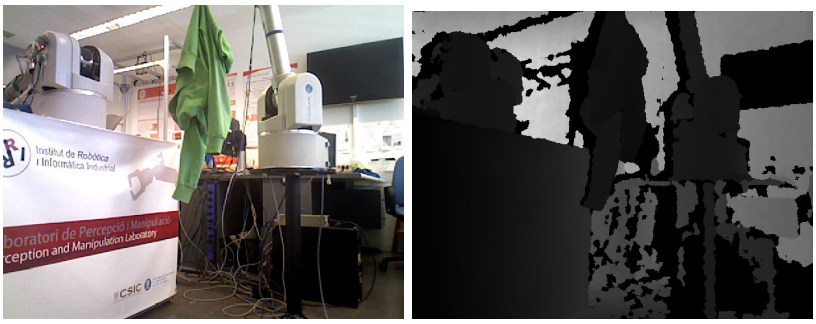


Figure 2: Color and depth views of our setup to capture depth images of the garment, in this case a jumper.

The whole method to bring a garment to a known configuration is depicted in Figure 1 with a pair of jeans being manipulated by two Barrett WAM arms. Figure 2 shows the color and depth appearance of our setup to obtain depth images and later grasp the garment, in this case a jumper, using the two robot arms.

Figure 3 depicts our proposal, based on three hierarchical steps: garment classification, first grasping point discovery, and second grasping point discovery. Our method needs training two CNNs per complex garment and a classifier CNN. Section 4 presents our proposal to classify the garment using a CNN trained on depth images from real and simulated garments. The garment texture provides a significant amount of information, but we need a vast amount of images to avoid overfitting on the garment colors. Hence, we train the system using only depth images, which encode the information about the garment pose and shape. We achieve good results without having to simulate the garments

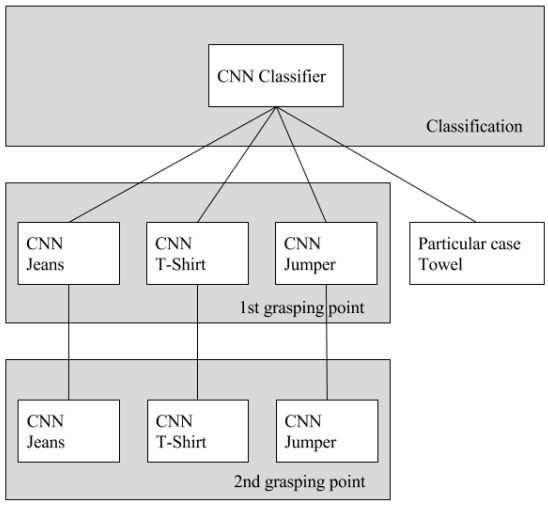


Figure 3: Garment manipulation pipeline: Initially, the image of a hanging garment is used by a classification CNN to identify the garment class: jeans, T-shirt, jumper or towel. A towel can be brought to a known configuration by grasping the vertex, which is the lowest point [**?**]. For more complex garments, a specialized CNN uses the same image to discover the first grasping point. Once the robot has used it to grasp the garment its shape has changed, and a second specialized CNN discovers the second grasping point using the new image.

in many different textures.

We obtained them using an Xtion depth camera located at 1,5 meters from the garment. At this distance the camera can enclose whole views of the pieces of clothing and their poses become identifiable. However, the images are slightly noisier than in previous works [**?**] where the camera was located nearer. Finally, the gripper is removed from the image and the remaining objects are depth-segmented before training the CNN. To increase the amount of data to train the CNNs, we generate depth images from simulated garments using a physics engine [**?**].

Section 5 is devoted to the process for finding the two target grasping points. We design a second CNN that analyzes the image trying to discover any of the two predefined grasping points. When one of the predefined grasping points is discovered the second robot arm grasps the garment from it. Otherwise, the garment is rotated to obtain new views of the garment until one of the points is discovered. Having grasped one of the two target points, a third more specialized CNN is used to find the visibility and Cartesian position of the other predefined point. Repeating the rotate-or-grasp process we end up having the garment in a known configuration. Results are detailed in Section 6.

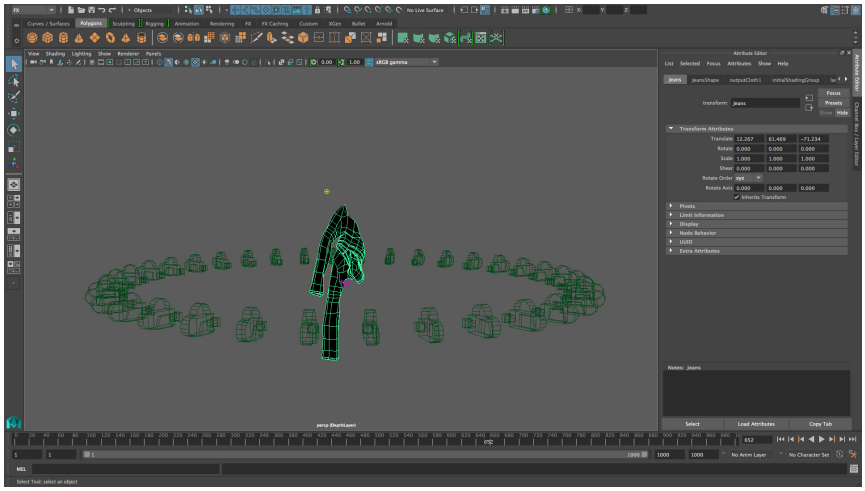


Figure 4: Physics engine interface when capturing depth images of jeans. Cameras are located at a fixed distance to capture different views of the same pose.

4. Garment recognition

This section presents our CNN-based approach to classify a hanging garment when it is being grasped by a robot in a reference position. Before detailing its architecture, we describe the process of obtaining all the training samples and some previous results. Our first attempt [**?**] was to use only real images. Using a robot arm, we continuously grasped a piece of cloth from a table and showed it to a camera. Doing it for the four garments taken into account, we obtained 2530 depth images from real garments. Since we consider the garment fixed in the center of the image, no additional translations or rotations could be used to increase the size of the dataset apart from applying symmetry over the x-axis. However, the number of images to train the CNN was quite limited and the classification success, around 80%, raised to 90% when using multiple views to disambiguate. In this first approach, we placed the camera at 70 centimeters from the grasping point, in order to take advantage from the more certain measurements in lower distances from depth cameras such as the Kinect, and the texture information on the cloth. However, some images do not show a complete view of the garment. In the current work, we move the camera back to 150 centimeters from the grasping point to show how we can obtain better results can be obtained when having a larger perspective of the garment.

Convolutional Neural Networks need a huge amount of data and training with the real images

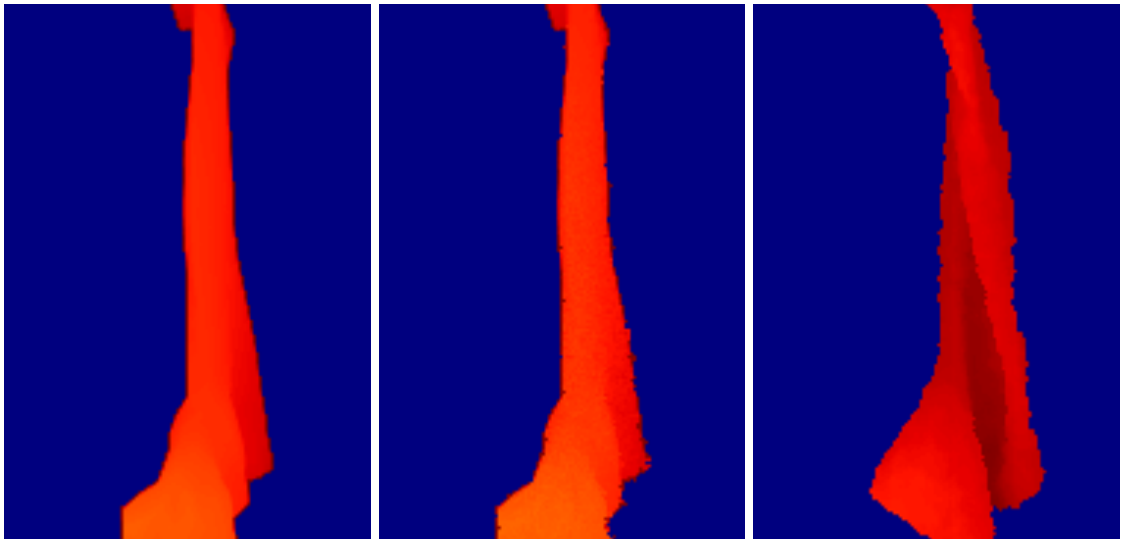


Figure 5: From left to right: Synthetic image of a pair of jeans with no noise, synthetic image with added kinect noise [**?**] and depth image taken from a real pair of jeans in a similar configuration.

obtained does not achieve enough reliability. Therefore, we used a physics engine to simulate pieces of cloth hanging as in Figure 4, imitating the real environment. We have used publicly available[1] 3D models of jeans, jumpers and T-shirts, and automatically created towels as rectangular surfaces of very different sizes.

The process hangs clothes from points all over the garment, simulates the piece of clothing falling and captures 36 depth images of the hanging pose from different points of view. By changing cloth properties and sizes, this automated process allowed us to capture up to 60,000 depth images to train a classifier[2].

Both, the real and synthetic images have a size of 240 pixels height and 320 pixels width. The center column of the image is the most variant region, as the garments hang from the top middle point, while the sides are less informative. Accordingly, the sides were cropped to have a width of 160 pixels, which still contains most of the information, reducing significantly the training time and the amount of memory needed.

[1]The models of pieces of cloth are available at `https://www.turbosquid.com/`

[2]Code to create towels and generate the database is available at `https://gitlab.iri.upc.edu/perception/MEL-database`

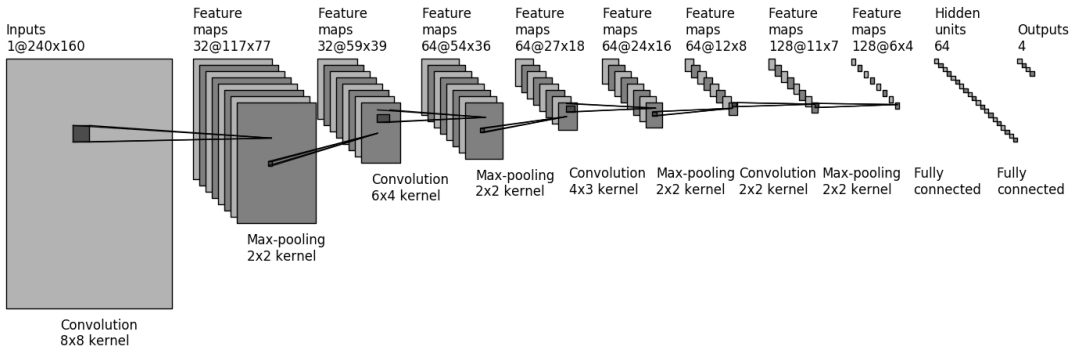


Figure 6: CNN structure: four convolutional layers with non-square size, four max-pooling layers and two fully connected layers.

On the simulator, the synthetic images are saved in 16-bit variables, which makes the quantization error extremely low, and the borders of the garment appear very definite. Instead, images from real cameras have several types of errors [**?**]. First, the uncertainty on the captured depth can be characterized as Gaussian noise correlated to the depth of each point and the pixel position. Also, horizontal and vertical error appears, though it is only recognizable on object borders, where the depth from a pixel to its neighbors is most different. We apply Choo et al. [**?**] model to simulate noise in images, to give them a more natural appearance. Figure 5 shows the aspect of the noisy real images, clean synthetic images and synthetic images with this noise model added. After adding this noise in the synthetic training images our model becomes more reliable and invariant to noise.

We designed a CNN whose structure is depicted in Figure 6. It consists of four convolutional and two fully connected layers. Between each convolutional layer, a max-pooling operation provides the network invariance to position. Nonlinearity is introduced in the form of ReLU functions [**?**] in the convolutional and fully connected layers, which proved to train faster and produce more accurate results than the sigmoid. The last fully connected layer has a softmax activation to convert the results to probabilities.

On the input layer we used Batch Normalization [**?**], which has recently shown better performance and considerably shorter training times than other normalization methods.

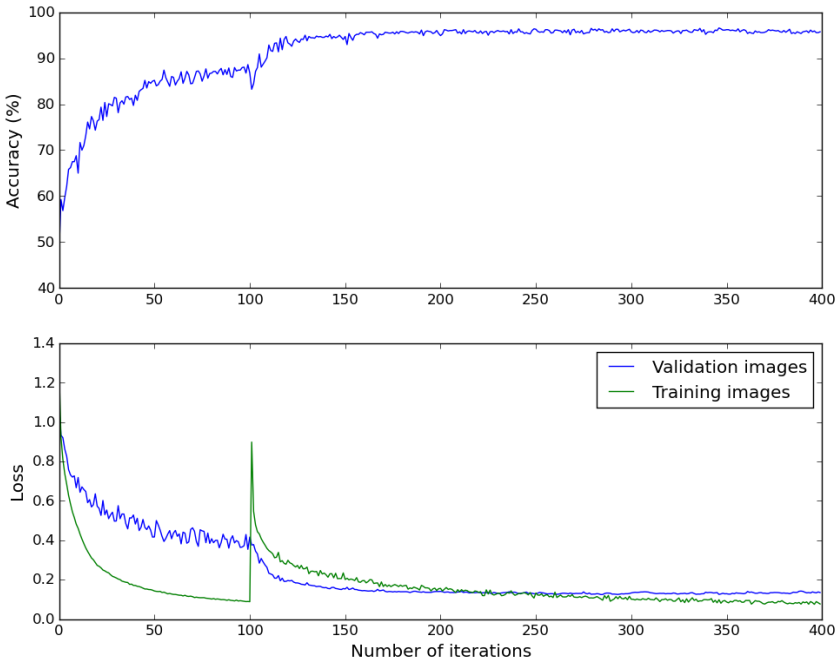


Figure 7: Accuracy and loss evolution during training for the training set (Green) and the validation set (Blue). From epoch 0 to 100, the training set is formed by the simulated images. Refinement starts at epoch 100 and generates a training loss peak, due to the change from the simulated training images to the real ones. The validation set is always formed by real images and shows how both sets contribute to the correct training.

The CNN was trained using 50 % of dropout [**?**] in the fully connected layers to prevent co-dependencies between nodes. Still, dropout did not improve the image filtering when used in convolutional layers. In addition, we applied L2 regularization [**?**] on parameters to increase the network generalization. The training optimization method was Adam [**?**], which adaptively modifies the learning rate in each layer.

The whole dataset of real images, consisting of 5060 images, was divided into training (60 %), validation (20 %) and test (20 %) sets, while 60,000 simulated images were also used for training. Due to the small amount of real images, we trained the CNN first on the simulated images and then refined it using the real data. The initial process helps the network in recognizing the parts of the garment and in learning how to identify it, while the second improves the network capabilities in processing the noisy real images. Figure 7 shows accuracy and loss during training for the

training and validation sets. The validation dataset is always composed of the same real garments, but the training dataset change can be seen around epoch 100, after the first training loss had stabilized. From then on, the network was refined using the dataset of real training images, which is significantly smaller and thus epochs pass much more rapidly. Thereby, in this part of the training we set a much smaller learning rate and higher regularization. The validation set gives a measure of how the algorithm is performing at every step for our real aim, to classify real clothes. Although the validation images do not modify directly the network parameters, we trained several networks and finally designed the structure in function of the performances in the validation set.

The global loss minimum is achieved in the epoch 284 and, after that, the network starts to slightly overfit the training images. This is shown by the validation curve not being reduced while the training loss is still decreasing, even though it includes added difficulties such as dropout and regularization. The parameters achieving the minimum validation loss are used to evaluate the performance of the algorithm in the test set, presented in section 6.

5. Garment manipulation

The following step consists of moving from an identified garment in any possible configuration to the garment in a known pose. Most tasks performed with clothing, such as folding garments or dressing a person, require manipulating a piece of cloth by grasping it from at least two points. Accordingly, we defined standard configurations as garments being grasped from two reference points.

Our approach to manipulate clothing consists of an informed one-shot grasping to try to obtain directly the garment in a desired position. Among the four garments being previously classified, the towel is a special case since its vertexes are easily identified in images: When a towel is grasped and held, the lowest point seen in the images is a vertex, as showed by Maitin Shepard et al. [**?**]. However, manipulation of complex garments such as jeans, jumpers and T-shirts remains a big challenge.

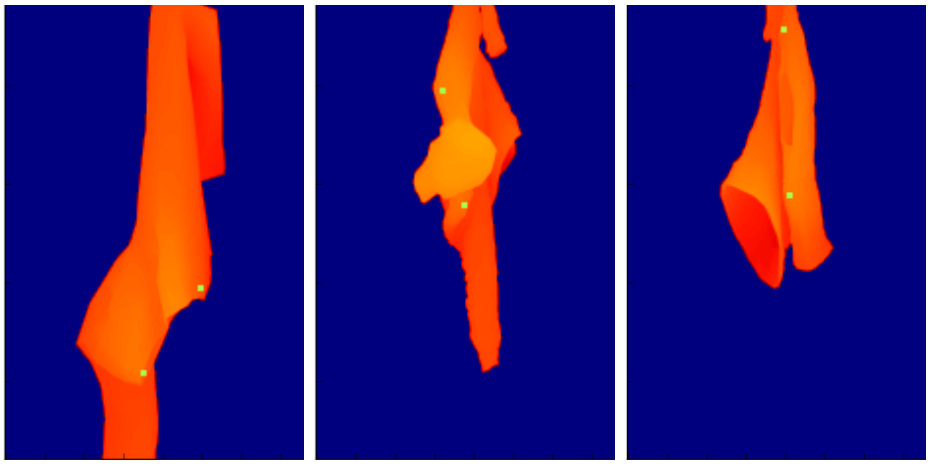
Apart from being a very time-consuming task, hand-labelling the reference grasping point position in each image brings significant noise. Also, the position of the points is remarkably difficult to label in some cases, for example with occluded points or wrinkled T-shirts. Therefore, in this case

we trained the CNNs using only simulated depth images. To do so, we first manually label the two reference grasp points in the physics simulator introduced in Section 4. Then, we repeat the process of simulating the cloth falling and saving depth images, while retrieving the Cartesian location of the grasp points for each pose. With this relation, we project the two points on the depth image in each view. Comparing the depth on the projection point to the labelled depth, we determine if the points are being occluded. We repeat the process with several variations on the garments sizes and physical properties to obtain 60,000 depth images per each garment type, with the labelled Cartesian position and visibility of the points. These were divided into training (60 %), validation (20 %) and test (20 %) sets. Some examples from these databases are shown in Figure 8. The first row shows images of garments considering any possible position and the projections of the reference points over the depth images. Our method involves localizing and grasping one of those visible points. By doing so, we obtain the type of images shown in the second row, with the second point to be retrieved and grasped.

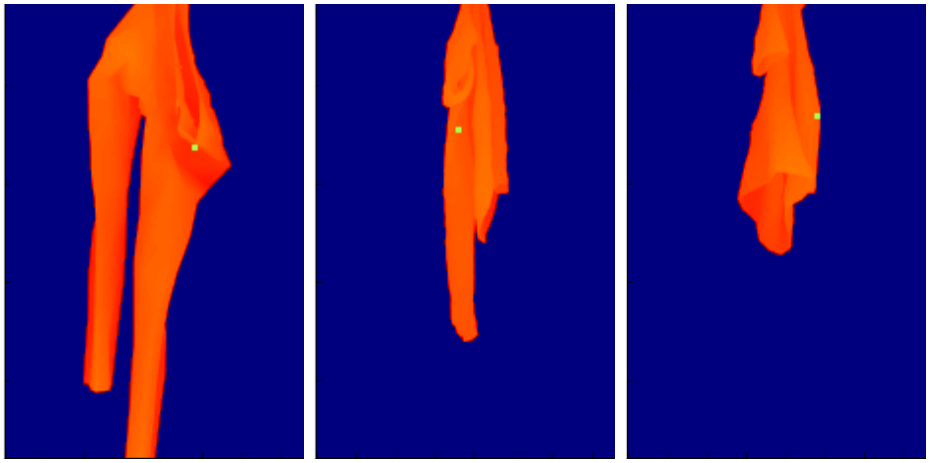
The first step deals with any possible configuration the garments may have, considering the full cloth surface as uniformly distributed possible grasping points. In the second step, though, the cloth should be grasped from one of the pre-defined points. Therefore the inspected poses follow a normal distribution, centered on the garment being grasped from the correct point and with a variance depending on the accuracy of the first step.

We predict the visibility and the Cartesian position of the points separately in the same network. To this extent, the output layer of the networks has been modified as shown in Figures 9a and 9b. The remaining structure is the one presented by Krizhevsky et al. [**?**], trained on a single GPU. Needless to say, predicting the position of occluded points supposes a significantly more complex problem, that results in decreasing the accuracy of the predictions even when they are visible. Instead, we are interested in having the maximum accuracy in the visible points, that is when we are going to grasp. The networks predict the location of the points at every stage, but, to maximize grasping success probability, mistakes over position do not penalize when the points are not visible.

Apart from the different range of poses they handle, the network on the first step has to output the position and visibility of the two points while the network on the second is simpler. As seen in Figure 9, the first network's output layer has 10 neurons, of which 6 are linear and predict the

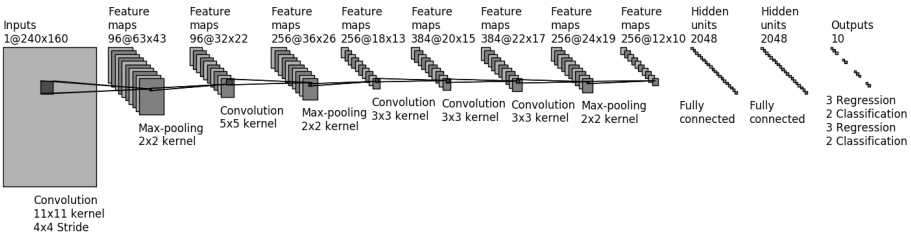


(a) Synthetic ground truth for localizing the first grasp point, in yellow points. From left to right: Jeans, jumper and T-shirt.

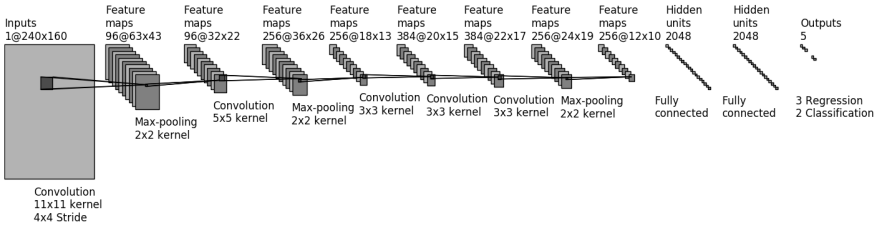


(b) Synthetic ground truth for localizing the second grasp point, in yellow points. From left to right: Jeans, jumper and T-shirt.

Figure 8



(a) CNN structure to detect one of the two pre-defined grasping points from an unknown initial grasping.



(b) CNN structure to detect the second grasping point after the first grasping.

Figure 9: Network structures, varying from [**?**] in the output layer.

location of each point. The other four predict the probability of the points being visible or not.

One problem with this situation is the order of the points predicted matching the ground truth. In a usual network, the error is computed from the difference - squared in regression and cross entropy in classification - between the labels and the predictions. In this case, we aim invariance on the prediction order, and requiring that the points are predicted in a particular order is an added complexity for the network to tackle. In the best case, the point order would follow a certain logic i.e. the first is the nearest one. However, we gain invariance on the points order by varying the loss function in the following way. We express P as a point having P_x,P_y and P_z to represent its Cartesian location and P_v, being 1 when the point is visible and 0 otherwise. First, we match the order of the points predicted to the ground truth for every image as

$$Order(P1, P2, T1, T2) = argmin(SE(P1, T1) + SE(P2, T2), SE(P1, T2) + SE(P2, T1)) \quad (1)$$

where P and T are the predicted and ground truth points, and SE is the squared error of a point and is defined as

$$SE(P, T) = (P_x - T_x)^2 + (P_y - T_y)^2 + (P_z - T_z)^2 . \quad (2)$$

Then, the order of the points retrieved determines the correspondence of the errors to the ground truth. We are considering the prediction of the position as a regression problem, where the location error in occluded points does not penalize the network. This can be expressed mathematically as

$$Loss_{regression}(P1, P2, T1, T2) = \begin{cases} P1_v \times SE(P1, T1) + P2_v \times SE(P2, T2) & \text{if } Order = 0 \\ P1_v \times SE(P1, T2) + P2_v \times SE(P2, T1) & \text{if } Order = 1 \end{cases} \quad (3)$$

given that the visibility of the two points, $P1_v$ and $P2_v$, adopt the values 0 or 1. The visibility is a classification problem whose error depends on the order of the points in the same manner than previously:

$$Loss_{classification}(P1, P2, T1, T2) = \begin{cases} CCE(P1, T1) + CCE(P2, T2) & \text{if } Order = 0 \\ CCE(P2, T1) + CCE(P1, T2) & \text{if } Order = 1 \end{cases} \quad (4)$$

where CCE is the binary cross entropy defined, as is standard, as

$$CCE(P,T) = -T_v log(P_v) - (1 - T_v) log(1 - P_v)\,. \tag{5}$$

Notice that T_v assumes the values 0 or 1 for occluded and non-visible points, respectively. So, CCE ends being the negative logarithm of the probability that has mistakenly been assigned to the incorrect class. The final error equation is the weighted sum of the squared error on the Cartesian location, the binary cross entropy on the visibility and the L2 regularization loss:

$$Loss = \sum Loss_{Regression} + k \sum Loss_{Classification} + Regularization \tag{6}$$

where k is a meta-parameter that puts all the gradients at the same order of magnitude. At training time, the back-propagating gradients sum up to modify the weights.

6. Results

This section reviews the results of the classification CNN and the rate success of the whole method to bring garments to known configurations. In addition, it presents some final details of how the networks have been trained. We have made available the source code and the documentation about the results obtained[3].

6.1. Garment classification

We evaluate the garment classification CNN depicted in Section 4. We use different combinations of real and simulated images, obtained using the physics simulator.

Figure 10 presents some examples of correct and incorrect classifications (first and second row, respectively). Mistakes show that different garments may have very similar appearance in depth images. When classifying an image, one of the options usually stands out from the others. Nevertheless, when the image does not contain features characteristic from one garment, such as sleeves,

[3] `http://www.iri.upc.edu/groups/perception/CNNgarments`

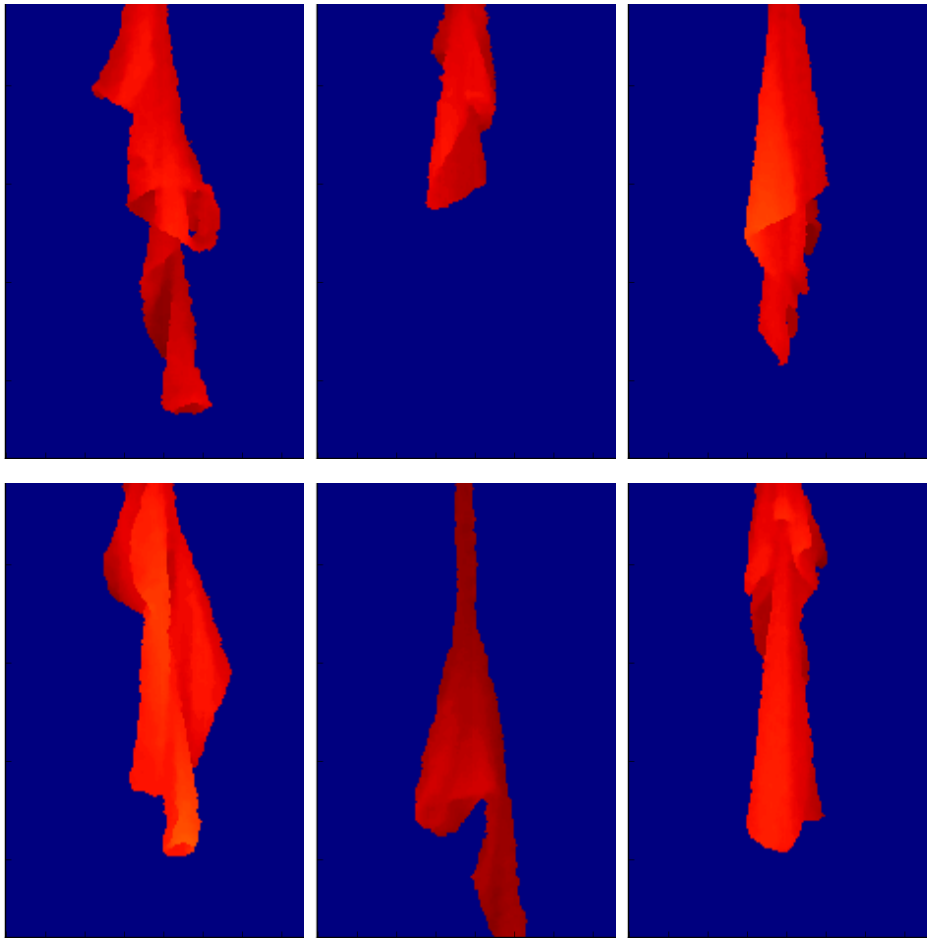


Figure 10: First row: Images of jeans, T-shirt and towel correctly identified, respectively. Second row: The same garments in misleading poses that made the classifier be mistaken.

a jumper's hood or a T-shirt's collar, the network is more uncertain and the probabilities for the different garments are more balanced.

In other occasions, a high probability is attributed to a mistaken garment due to the similarity of the poses adopted, as in the second row of Figure 10. The long sleeves of the jumper are very similar to the jeans legs, which is a common source of error for the network. The confusion between the jeans and the towel could be due to the similarity of towel views with some poses of the jeans where the legs are not distinguishable.

This topic has been studied very differently in other works. Every one takes into account different

Table 1: Comparison of the proposed approach with other methods.

Approach	Number of garments	Accuracy %
Li et al. [?]	3	81.67
Mariolis et al. [?]	3	89.38
Gabas et al. [?]	4	92
Our approach using only real images	4	88.61
Our approach using synthetic and real images	**4**	**96.85**

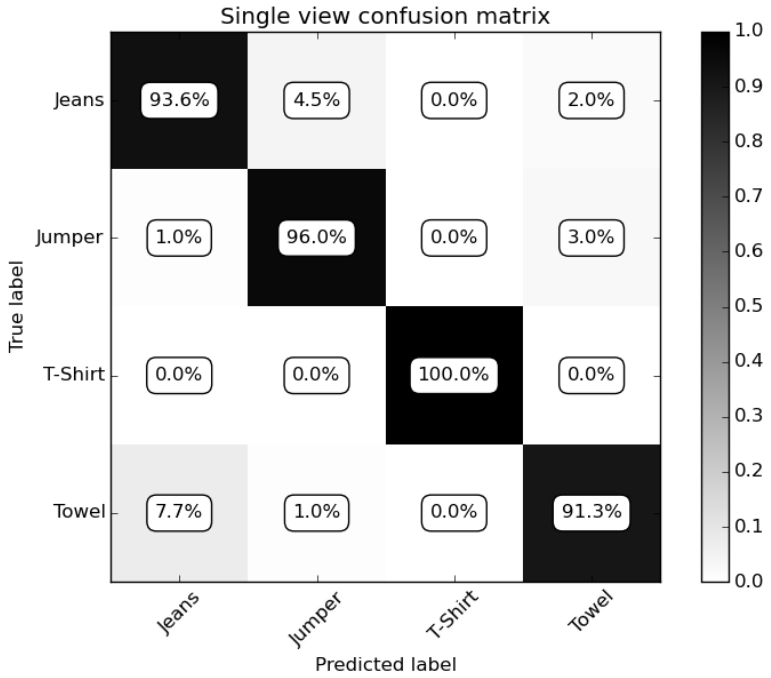


Figure 11: Confusion matrix. The diagonal shows correct identification rates per each type of garment. The most relevant missclassification occurs between jeans and towels.

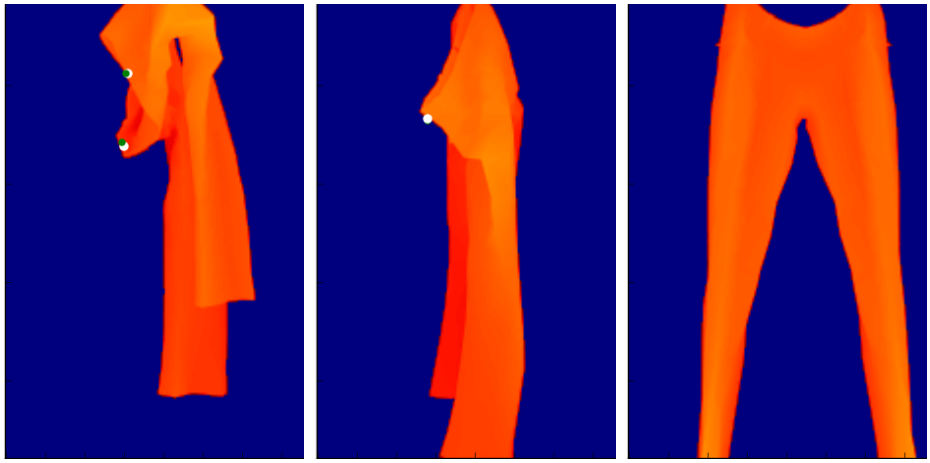
numbers of pieces of clothing and points of view, which difficulties the comparison. Gabas et al. [?], Mariolis et al. [?] and us have considered a fixed relative position of the garment with respect to the camera while other approaches such as Li et al. [?] consider different views from other angles but still maintaining a fixed distance. Additionally, we are the ones dealing with more different garments, together with Gabas et al. [?]. Using only the real images available, we obtain a one-shot classification accuracy similar to the state of the art, only less accurate than Mariolis et al. [?], who uses one garment less, and Gabas et al. [?], that disambiguates on several views of the same pose. When we combine synthetic and real images, we obtain a considerable improvement over all other recent approaches. The comparison to these works is summarized in Table 1.

The confusion matrix in Figure 11 shows accuracy and miss-classification rates for the four garments considered in this work. Paying attention to the nature of the images, we observe that T-shirts adopt a unique shape, very different to the other three types of clothes. As a consequence, T-shirts are correctly classified in the nearly two hundred T-shirt real test images in the dataset. On the contrary, the other three types may have poses that look very similar, thus leading to identification mistakes. Significantly, the towel is the most disorienting garment, often confounded with jeans.

6.2. Bringing garments to a known configuration

In this section we validate the discovery of the first and the second grasping points, using the methodology presented in Section 5. As explained before, we use only simulated garments where ground truth is easy to label.

We use the example of a pair of jeans to illustrate the whole process (see Figure 12) and evaluate the performance at each of the two stages. The jeans are initially grasped from a random point (Figure 12a). The image contains the ground truth (white points) and the grasp predictions (green points). Observe that, in this step the garment can be in an infinite range of positions. If no points are predicted, the robot rotates the garment until at least one point is visible. Then, the robot grasps using the point whose coordinates are more accurately localized in the point cloud. After grasping by one point, Figure 12b shows the second configuration with the ground truth and the second point location prediction. As in this case the second point is visible, the robot can grasp it. Otherwise, the robot again rotates the garment and provides new views. Finally, Figure 12c shows



(a) (b) (c)

Figure 12: Simulation of the complete pipeline grasping a pair of jeans. (a) Image after grasping from a random position with the projection of the predicted grasping points, in white, and the ground truth, in green. (b) Image of the jeans grasped from one of the two predicted points in the first image. This step is significantly more accurate due to dealing with a narrower range of poses. The ground truth (green point) and the prediction (white point) are superimposed. (c) The jeans grasped from the two points predicted previously.

the final configuration. More grasping experiences can be found in the article webpage[4].

After training, the whole process was evaluated simulating two different sizes of jeans that had never been seen before by the network. For each one of them, 93 different initial poses were used, randomly choosing the grasping point. To simulate that the garment is being rotated, we make use of the images obtained by cameras on neighbouring points of view. Then, we predict the visibility and Cartesian position of the points with the second hierarchical network. If no point is predicted visible, we choose the neighbor image until at least one point becomes visible. If both points are visible, we compute their Cartesian projections on the image and compare their depths in the images to the depths predicted by the network. Then, we use the point more accurately predicted.

To advance to the next step, we simulate the same garment being grasped by the nearest point to the prediction one in the 3D model. Similarly to the previous step, then we retrieve the visibility and position of the remaining point using the third network. This ends by grasping the nearest point to the prediction, ideally ending with the garment in a reference position.

[4] `http://www.iri.upc.edu/groups/perception/CNNgarments`

Table 2: Percentage of grasping points correctly obtained in simulation and their distance to the ground truth. Possible grasping points were discretised and labelled in the synthetic model.

	Correct vertex	Between 2 - 4 cm	Between 4 and 8 cm	Between 8 and 12 cm	Between 12 and 16 cm	Between 16 and 20 cm
1st point grasped	81.18 %	4.3 %	5.38 %	5.38 %	2.15 %	1.07 %
2nd point grasped	96.77 %	2.69 %	0.64 %	0.00 %	0.00 %	0.00 %

Table 3: Mean distance from predicted point to ground truth in centimetres.

Garment	Jeans	Jumper	T-Shirt
Average error distance on the first grasping point	1.59	2.22	2.76
Average error distance on the second grasping point	1.52	1.18	2.16

The discretized distance between the reference grasping points and the simulated ones are quantified in Table 2. As expected, the second grasping point discovery is significantly more accurate than the first one. As explained before, the third CNN has to detect only one point and deals with a reduced set of possible deformations of the garment as the garment is being hanged from one of the pre-defined points. In comparison, the second CNN has to discover two points and the deformation set includes all possible deformations as the grasping is performed randomly at any point.

After going in detail with the jeans example, the results for the three types of garments considered are presented. Table 3 presents the average absolute distance from the predicted points to ground truth in centimetres. As can be observed, jeans and jumpers achieve similar results but T-shirts are slightly more difficult to deal with, because of the wrinkles they form and their small characteristic features - only short sleeves.

6.3. Real setup experiments

We performed several experiments of the whole process of bringing a grasped garment to a known configuration. Our setup includes two Barret's WAM robot arms and a Xtion camera. The whole manipulation process can be qualitatively appreciated in a video where we manipulated a real

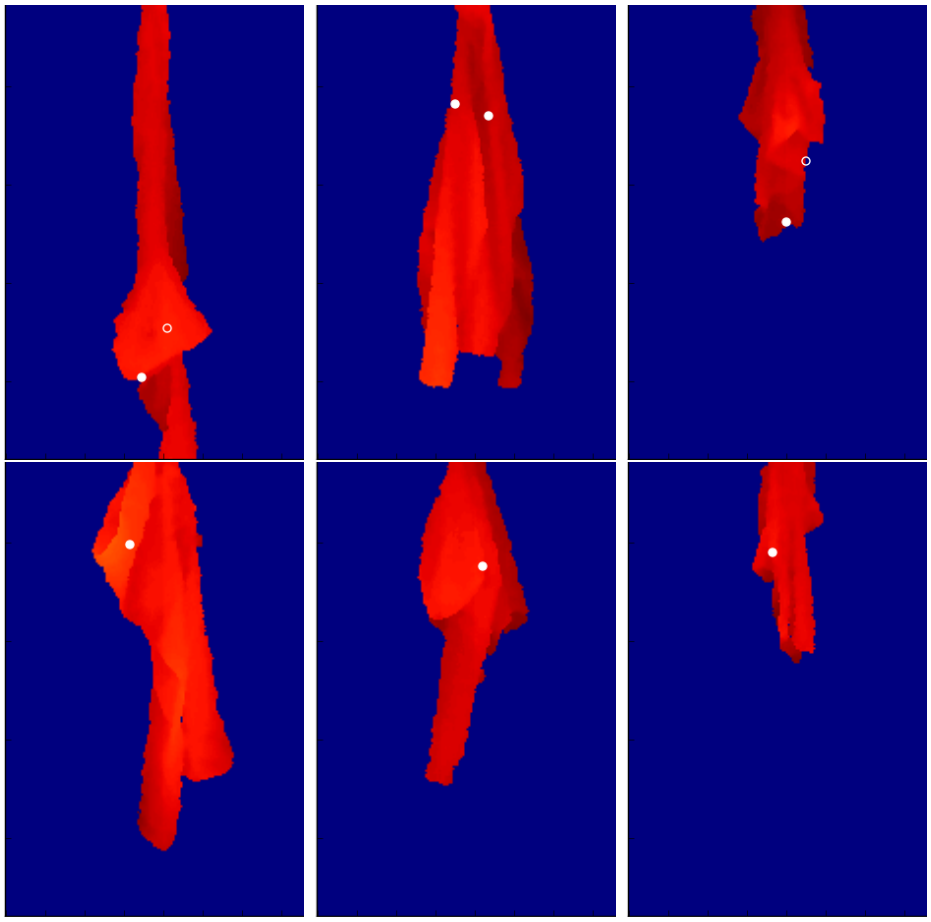


Figure 13: Grasping point location predictions. From left to right columns: Grasping points prediction for jeans, jumper and T-shirt. The upper row shows the predicted position of the two first known points and the second row shows the predicted second grasp point location. A filled white circle means the grasp point is visible, and therefore the robot could approach and grasp it, while an empty circle suggests another view is needed to see the point.

jumper[5]. The most relevant grasping actions can be seen in Figure 14, where the robot arms hold a jumper by the shoulders. The predictions are not as accurate as in simulation but, still, the process leads to a similar pose to the reference one, for each garment.

As expected, the prediction of the grasping point positions was less accurate when predicting from real garment images not used for training the convolutional networks. The error in real images could not be measured as we do not have a labelled dataset. However, we qualitatively observe that training the CNNs with additional noise helps the network to increase its success. We believe that a refinement would still improve its performance with real images, as it did in Section 4 with the garment classification task.

Regarding the robot execution, we have observed that the grasping action is a critical aspect. Most of the failures were caused by defective graspings, mainly because the robot gripper sometimes collides with the garment in the approach trajectory changing the grasping point position. We think that a more elaborated grasping strategy will help, for example using a specific grasping orientation for each point. This orientation could be either predicted with the CNN or computed from the garment pointcloud. Moreover, our gripper is generic and a specialized gripper for garment manipulation may help.

7. Conclusions and future work

We established a three-level hierarchy to recognize garments and grasp each of them appropriately. Every level contains a specialized CNN, whose output defines how to move to the next stage.

The first step consists of recognizing the clothing type so as to permit manipulating it in a specialized way. To do so, we train a CNN on depth images from real and simulated garments. We achieve comparable results to the state of the art when using only one type of images, and outperform the current best classification accuracy when combining real and simulation images. Moreover, the possibility of using only synthetic images opens up the way to completely automate the process, reducing significantly the amount of work to be done when adding new pieces of clothing. Observing the nature of the images, we realize that T-shirts appear to be very different from the other clothing

[5] `http://www.iri.upc.edu/groups/perception/CNNgarments`

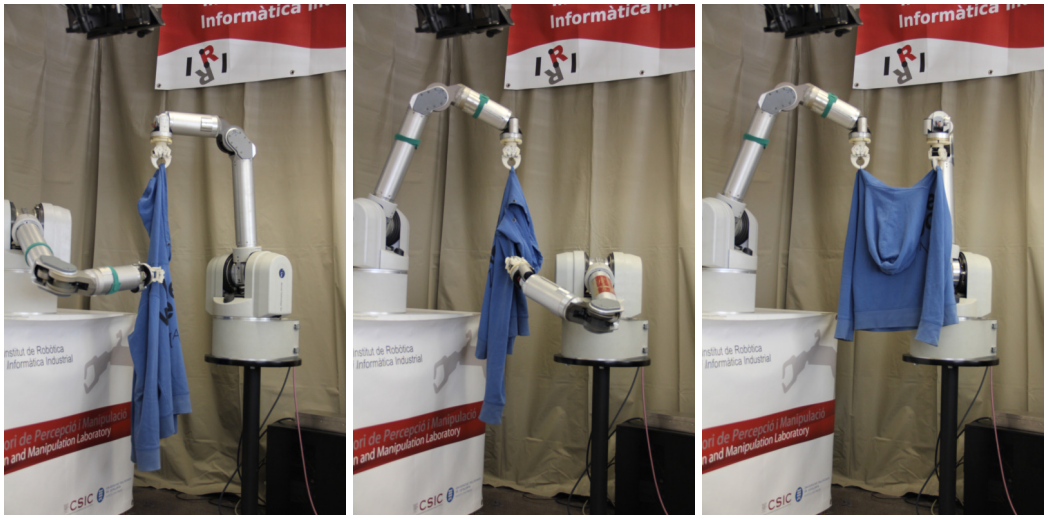


Figure 14: Process of bringing a garment to a target configuration. Initially, one robot arm grasps a jumper that is identified using the first CNN of the hierarchy. Then the garment is rotated until one of the pre-defined grasp points is predicted to be visible by the second CNN. Then a second robot arm approaches and grasps the jumper and moves it in front of the camera while the first arm moves away. Again, the garment is rotated until the first arm approaches the visible second pre-defined grasp point and moves the garment to the target configuration on the third image.

items considered, they being correctly identified in the nearly two hundred T-shirt test depth images available. On the contrary, the other three garments may adopt similar poses, which lead the network to make mistakes. Not surprisingly, the towel is the most difficult item to recognize, often confounded with jeans. This work considers only 4 different garment categories, as is common in literature. In the future, it should be generalized to a larger number adding new garment types.

In this paper, we have assumed that the camera is able to view almost the complete garment held by the robot. This imposes some limitations on the physical placement of both the camera and the robot arms. As future research, we would like to investigate if our algorithm could be able to detect grasping points using partial views of the garments. Moreover, we plan to increase the difficulty of the identification problem by considering the possibility of simultaneously having more than one garment grasped. When several clothing garments form a pile, experience shows that more than one can be caught at the same time. This could result in either two garments side by side or one over the other. In both cases, the process of bringing the garments to reference configurations would be more complex.

Our pipeline continues by retrieving the reference 3D grasping points for the identified garment. Our results from Section 5 show this problem can also be tackled with CNNs and, using synthetic images, achieve a good success rate when bringing a garment to a familiar pose. In this case, we use only simulated images due to the difficulty of gathering enough real labelled data. In the future we would like to compute not only the position but also the orientation of the reference 3D grasp.

The experiments using a real robot with two WAM arms are promising. The learned lesson in this case is that the robot trajectories and the gripper should be improved, as most of the failures are due to collisions and lack of performance of the used gripper.

Acknowledgments

This work was partially supported by the EU CHIST-ERA I-DRESS project PCIN-2015-147, by the Spanish Ministry of Economy and Competitiveness under project Robinstruct TIN2014-58178-R, and by the CSIC project TextilRob 201550E028.

8. References

[1] C. Torras, Service robots for citizens of the future, European Review 24 (1) (2016) 17–30.

[2] M. Kaneko, Y. Tanaka, T. Tsuji, Scale-dependent grasp-a case study, in: IEEE International Conference on Robotics and Automation (ICRA), Vol. 3, 1996, pp. 2131–2136.

[3] M. Kaneko, M. Kakikura, Planning strategy for putting away laundry-isolating and unfolding task, in: Proceedings of the IEEE International Symposium on Assembly and Task Planning, 2001, pp. 429–434.

[4] A. Colomé Figueras, D. E. Pardo Ayala, G. Alenyà Ribas, C. Torras, External force estimation for textile grasp detection, in: Proceedings of the 2012 IROS Workshop Beyond Robot Grasping: Modern Approaches for Learning Dynamic Manipulation, 2012, pp. 1–1.

[5] C. Wang, S. Shen, Y. Liu, A fast approach to deformable surface 3d tracking, Pattern Recognition 44 (12) (2011) 2915–2925.

[6] B. Willimon, S. Birchfield, I. Walker, Classification of clothing using interactive perception, in: IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011, pp. 1862–1868.

[7] B. Willimon, I. Walker, S. Birchfield, A new approach to clothing classification using mid-level layers, in: IEEE International Conference on Robotics and Automation (ICRA), 2013, pp. 4271–4278.

[8] Y. Kita, F. Kanehiro, T. Ueshiba, N. Kita, Strategy for folding clothing on the basis of deformable models, in: International Conference Image Analysis and Recognition, Springer, 2014, pp. 442–452.

[9] M. Cusumano-Towner, A. Singh, S. Miller, J. F. O'Brien, P. Abbeel, Bringing clothing into desired configurations with limited perception, in: Robotics and Automation (ICRA), 2011 IEEE International Conference on, 2011, pp. 3893–3900.

[10] Y. Li, C.-F. Chen, P. K. Allen, Recognition of deformable object category and pose, in: IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 5558–5564.

[11] Y. Li, Y. Yue, D. Xu, E. Grinspun, P. K. Allen, Folding deformable objects using predictive simulation and trajectory optimization, in: Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, 2015, pp. 6000–6006.

[12] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, A. Fitzgibbon, Kinectfusion: Real-time dense surface mapping and tracking, in: 10th IEEE international symposium on Mixed and augmented reality (ISMAR), 2011, pp. 127–136.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[14] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.

[15] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015.

[16] S. Ren, K. He, R. Girshick, J. Sun, Faster r-cnn: Towards real-time object detection with region proposal networks, in: Advances in neural information processing systems, 2015, pp. 91–99.

[17] M. Schwarz, H. Schulz, S. Behnke, Rgb-d object recognition and pose estimation based on pre-trained convolutional neural network features, in: Robotics and Automation (ICRA), 2015 IEEE International Conference on, IEEE, 2015, pp. 1329–1335.

[18] H. Su, C. R. Qi, Y. Li, L. J. Guibas, Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views, in: The IEEE International Conference on Computer Vision (ICCV), 2015.

[19] I. Mariolis, G. Peleka, A. Kargakos, S. Malassiotis, Pose and category recognition of highly deformable objects using deep learning, in: Advanced Robotics (ICAR), 2015 International Conference on, 2015, pp. 655–662.

[20] A. Gabas, E. Corona, G. Alenyà, C. Torras, Robot-aided cloth classification using depth information and cnns, in: International Conference on Articulated Motion and Deformable Objects, Springer, 2016, pp. 16–23.

[21] F. Osawa, H. Seki, Y. Kamiya, Unfolding of massive laundry and classification types by dual manipulator., JACIII 11 (5) (2007) 457–463.

[22] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, P. Abbeel, Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding, in: IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 2308–2315.

[23] D. Triantafyllou, I. Mariolis, A. Kargakos, S. Malassiotis, N. Aspragathos, A geometric approach to robotic unfolding of garments, Robotics and Autonomous Systems 75 (2016) 233–243.

[24] A. Ramisa, G. Alenya, F. Moreno-Noguer, C. Torras, Finddd: A fast 3d descriptor to characterize textiles for robot manipulation, in: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2013, pp. 824–830.

[25] A. Ramisa, G. Alenyà, F. Moreno-Noguer, C. Torras, A 3d descriptor to detect task-oriented grasping points in clothing, Pattern Recognition 60 (2016) 936–948.

[26] A. Doumanoglou, T.-K. Kim, X. Zhao, S. Malassiotis, Active random forests: An application to autonomous unfolding of clothes, in: European Conference on Computer Vision, Springer, 2014, pp. 644–658.

[27] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrik, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, S. Malassiotis, Folding clothes autonomously: a complete pipeline, IEEE Transactions on Robotics 32 (6) (2016) 1461–1478.

[28] Y. Li, D. Xu, Y. Yue, Y. Wang, S.-F. Chang, E. Grinspun, P. K. Allen, Regrasping and unfolding of garments using predictive thin shell modeling, in: IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 1382–1388.

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

[30] Maya, `http://www.autodesk.com/products/autodesk-maya`.

[31] B. Choo, M. Landau, M. DeVore, P. A. Beling, Statistical Analysis-Based Error Models for the Microsoft Kinect$^{\text{TM}}$ Depth Sensor, Sensors 14 (9) (2014) 17430–17450.

[32] S. Foix, G. Alenya, C. Torras, Lock-in time-of-flight (tof) cameras: A survey, IEEE Sensors Journal 11 (9) (2011) 1917–1926.

[33] B. Xu, N. Wang, T. Chen, M. Li, Empirical evaluation of rectified activations in convolutional network, arXiv preprint arXiv:1505.00853.

[34] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167.

[35] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting., Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[36] F. Girosi, M. Jones, T. Poggio, Regularization theory and neural networks architectures, Neural computation 7 (2) (1995) 219–269.

[37] T. Chilimbi, Y. Suzue, J. Apacible, K. Kalyanaraman, Project adam: Building an efficient and scalable deep learning training system, in: 11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14), 2014, pp. 571–582.