Towards Clothes Hanging via Cloth Simulation and Deep Convolutional Networks

David Estevez*, Juan G. Victores, Raul Fernandez-Fernandez, Carlos Balaguer

RoboticsLab, University Carlos III of Madrid, Calle Butarque 15, 28911 Madrid, Spain
**david.estevez@alumnos.uc3m.es*

SNE 31(3), 2021, 169-176, DOI: 10.11128/sne.31.tn.10578
Received: March 10, 2021 (Selected EUROSIM 2019 Postconf. Publ.), Revised: Sept. 1, 2021; Accepted: September 3, 2021
SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. People spend several hours a week doing laundry, with hanging clothes being one of the laundry tasks to be performed. Nevertheless, deformable object manipulation still proves to be a challenge for most robotic systems, due to the extremely large number of internal degrees of freedom of a piece of clothing and its chaotic nature. This work presents a step towards automated robot clothes hanging by modeling the dynamics of the hanging task via deep convolutional models. Two models are developed to address two different problems: determining if the garment will hang or not (classification), and estimating the future garment location in space (regression). Both models have been trained with a synthetic dataset formed by 15k examples generated though a dynamic simulation of a deformable object. Experiments show that the deep convolutional models presented perform better than a human expert, and that future predictions are largely influenced by time, with uncertainty influencing directly the accuracy of the predictions.

Introduction

Domestic tasks are very time-consuming. In general, every single human would benefit from a complete automation of domestic tasks, as they are not only a nuisance, but they can be considered a form of unpaid labor. In addition, certain collectives, such as the elder and disabled people, could specially take advantage of domestic task automation, due to their lack of mobility. For these collectives, domestic tasks can become a daily struggle that may even require assistance from other people. To automate some of these tasks, such as washing or cooking, specific appliances have been developed that help us save time and effort. However, due to the large number of disparate tasks to be automated, it becomes impossible to own a specific machine for each task, as this would be both expensive and demand an unfeasible amount of space for installation. For these reasons, robots are key in automating domestic tasks in a viable and maintainable way.

Laundry tasks constitute one of the most prominent subsets of domestic tasks and, at the same time, one of the hardest to automate, as they involve working with textiles. Textile articles, such as garments, are intrinsically hard to perceive and manipulate due to, amongst other facts, their almost infinite number of internal degrees of freedom and possible configurations, as well as their deformable and chaotic nature, which is extremely difficult to predict. While there is extensive existing literature on the typical laundry pipeline, defined as unfolding, ironing and folding, few works exist that focus on other garment-related tasks such as bed making, dressing assistance or hanging garments.

Hanging garments is a task performed after garments are washed, and before the garments are unfolded and ironed, to have a perfectly dry clothing article. As with all garment-related tasks, automation of garment hanging is difficult because of several reasons:

- Garment manipulation requires a robotic system with proficient manipulation skills and enough dexterity to deal with thin textile materials.
- It requires a deep understanding of deformable objects intrinsic properties and the physics that govern their movements, which make their movements chaotic and difficult to predict.
- The high number of internal degrees of freedom, possible poses, and occlusions present on a garment in a real world scenario makes perceiving and modelling garments a challenging problem.

In this paper, we aim to make a contribution towards the goal of an automated robotic system able to do the laundry, specifically the hanging garments task. For that purpose, we present two different deep convolutional models that are able to predict the behavior of the garment about to be hanged, addressing two different problems: determining if a garment dropped onto a hanger will hang or not from a depth image of the garment just before it is released, and estimating the future location in space of a dropped garment onto a hanger. To train these models, a synthetic dataset has been created, including examples of both dropped garments that hanged and garments that fell to the floor.

1 State of the Art

The typical laundry pipeline, as defined in the literature, is composed of three different tasks: unfolding, ironing and folding. For unfolding, existing approaches use depth images of the garment to determine the most suitable grasp sequence for unfolding, using active random forests [3], upper layer detection via Canny Edge detector [18] or a garment-agnostic model-less approach [5]. Once the garment is unfolded, it has to be ironed before it can be folded and stored, so that no wrinkles are present in the garment when the owner is about to wear it. If the wrinkles are large and soft, pulling the edges of the garment is enough to remove them [17]. If the wrinkles are small and marked, the robot has to iron them. While some approaches target individual wrinkles in very controlled illumination conditions [9], other use a human-inspired ironing method based on force control and a garment surface analysis [6]. When all the wrinkles have been removed, the only remaining step is to fold the garment. For folding, some approaches rely on manipulation sequences that take the garment from a random initial state to the desired folded state using Hidden Markov Models (HMMs) [2] while others follow a perception-based approach use polygonal models to estimate the garment shape and folding sequence [16].

For every garment-related task, prediction of garment movement and state estimation are key abilities, specially when involving garment manipulation. Miller et al. [13] used parametrized shape models that are able to fit 2D views of already flattened garments to find the grasp points required to apply a given folding sequence. Cusumano-Towner et al. [2] applied an approach heavily based on manipulation. Their approach used a series of manipulation operations to estimate the state a garment with a Hidden Markov Model. The initial state might be unknown, but a known state is reached through manipulation. Bersch et al. [1] used fiducial markers stamped on a t-shirt to obtain a 3D reconstruction of the garment. Since the markers were unique for each point, the state of the garment can easily be estimated from them, allowing them to compute suitable grasp poses on the cloth, for later manipulation with a PR2 robot. Kita el al. [7] introduced a method to estimate the state of a garment through a 3D reconstruction of the garment from different viewpoints. A cylindrical Z buffering algorithm is then applied to the reconstructed point cloud and expanded to obtain a flattened representation of the garment. Though matching with a database of garments grasped from different garment points, the actual state of the garment can be estimated.

Willimon et al. [19] used energy minimization and graph cuts to estimate the configuration of cloth surfaces from 2D color images, with an automatic mesh generation algorithm that provides a triangular mesh encapsulating the cloth surface without predefined values. Li et al. [10] proposed a real-time state estimation algorithm based on a cylindrical descriptor used to match a given real-world observation with a garment pose database generated through physical simulation. This descriptor obtains a binary string representation by fitting the garment 3D point cloud in a cylinder divided into different sections, that are then used as an occupancy grid and unrolled into a 1D binary string. Mariolis et al. [11] use two stacked deep convolutional neural networks to estimate the state of a garment from a depth image of the garment being hanged by one point. The first convolutional network determines the garment category, which is used to select which of the convolutional networks that have been trained on that particular category has to be used to compute the current pose.

Although the literature has traditionally focused on the tasks in the laundry pipeline, recent works exist that discuss other tasks, such as clothing assistance [8], and bed making [15].

2 Dataset Generation

A Deep Convolutional Network is used to model the behavior of a free falling garment and to predict its final location. Deep neural networks employ a large amount of parameters that need to be trained, and therefore they require a very large amount of data for training and validation. Ideally, the network has to be trained with data from the same domain as the application. That is, if the model is to be applied in a real world scenario, it should be trained with real world data. However, training data generation in the real world is very time demanding, as for each trial the robot has to be reset to its initial position, a garment has to be placed on its end-effector, and then the robot has to drop it so that the trajectory of the free falling garment can be recorded. Finally, the garment has to be picked up again from its final location to be set for the next trial. In addition, tracking the exact 3D position of the garment while falling is not trivial.

An alternative to using real world data is to train the model instead with simulated data. Since the characteristics of the simulated domain are very different from the real world domain (no noise, no lens distortion, different illumination conditions, uniform colors...), some modifications or special techniques are required to apply a model trained in the simulated domain to the real world. One of such techniques is domain randomization, that generates training data from a simulation using random colors, textures and illumination, so that the model is able to generalize and perform correctly independently of the domain.

In this work, simulation was used as the source of training examples to obtain a large training dataset. A piece of cloth is simulated in a virtual environment representing a simplified setup similar to the one in our lab, including a 1 m x 1 m hanger and the floor. (Figure 1). To simulate the garment dynamics and interaction with the lab environment, we use a spring-based model applied to the 3D mesh, as included in the Blender software package[`https://www.blender.org/`, last accessed: 08-06-2019].

For each of the trials, one vertex of the mesh is selected as hanging point, and then placed at a random initial location. The initial location is sampled from a normal distribution (with mean μ_{init} and standard deviation σ_{init}). To increase the chances of a garment being randomly hanged, the source distribution is centered around the hanger, and close to it. The simulation of the garment dynamics is then started so that the garment moves from its initial flat pose to the in-air hanging pose. The garment is left hanged a sufficient amount of time to reach a static state, as it will swing due to the inertia of this initial movement. Once the garment is static, a depth image showing the initial pose and location is stored. Using depth images instead of color images enables us to translate well between domains without the need for domain randomization, as depth images from the real sensor and virtual depth images are monochromatic and of very similar characteristics. Finally, the garment is dropped and the position of the center of mass of the falling garment is tracked for later analysis and to derive a ground truth binary label for classification.

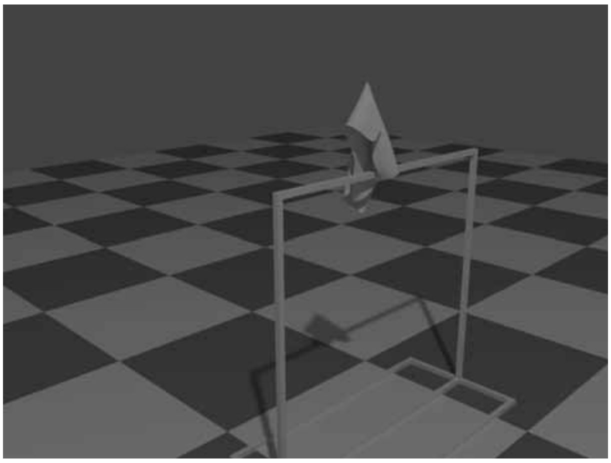

Figure 1: Simulated lab setup with hanging garment and hanger.

As garments are dynamic systems displaying a chaotic behavior, the result of each trial is not deterministic and depends heavily on small changes of the initial conditions (location and pose) of the piece of cloth. For that reason, the resulting dataset is unbalanced, including a larger amount of samples in which the cloth did not hang.

3 Hanging Prediction

Once enough training data has been generated, it is used to train a model to predict the behavior of the free falling garment. Two different models have been trained and evaluated. The first one tries to estimate the location of the garment at a given simulation time step, usually the last one in the trajectory (i.e. the final location of the garment, either hanged or in the floor). The other one classifies the final outcome into two different categories: hanged or floor, depending on the expected outcome of dropping the garment. This section will describe each of them.

3.1 Regression

The objective of the regression model is to predict the exact location of a garment at a given simulation time step from a depth image showing the initial location and pose of the garment. In most of the cases, the most interesting time step from the perspective of hanging clothes is the last one, as it will determine whether the garment will hang or not when dropped from a given initial location. Due to the chaotic nature of the cloth dynamics, as time steps advance the uncertainty of the garment location increases, with the initial location being the easiest to predict, and the final location the most difficult.

For the model, a Deep Convolutional Network is used. Figure 2 shows the architecture of the network. The network is composed of 4 sets of convolutional layers, and 2 fully connected layers to compute the output. Sets 1 and 3 are built from 2 convolutional layers with 16 filters of size (3x3) and an Exponential Linear Unit (ELU) as the activation function, followed by a max-pooling layer of size (2x2) and stride 2. Sets 2 and 4 are composed of a single convolutional layer with 32 filters of size (3x3) and ELU as the activation function, followed by a max-pooling layer of size (2x2) and stride 2. The fully connected layers have 300 and 3 neurons, respectively. For the regression problem, we use ELU as activation function for the last layer, representing the predicted 3D coordinates (X, Y and Z) of the garment at a given time step. The total number of learnable parameters of this model is 2 100 707.

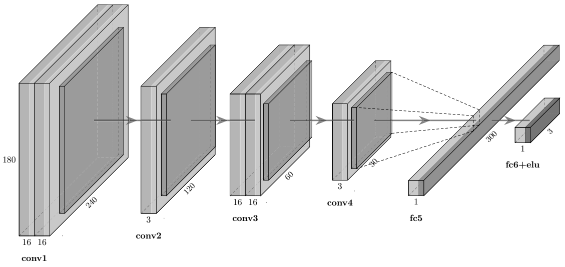

Figure 2: HANGnet architecture diagram.

The parameters were trained by optimizing a custom loss using an Adam stochastic optimizer. As the aim of the prediction is to hang garments, the most interesting information obtained from the prediction is the value of the Z coordinate. To emphasize in the importance of the Z coordinate, a custom loss function was developed:

$$Loss = \omega_x * (X - \hat{X})^2 + \omega_y * (Y - \hat{Y})^2 + \omega_z * (Z - \hat{Z})^2 \tag{1}$$

Where $\vec{X} = (X, Y, Z)$ are the actual coordinates of the point in the training example, $\vec{\hat{X}} = (\hat{X}, \hat{Y}, \hat{Z})$ are the predicted coordinates, and ω_x, ω_y and ω_z are hyperparameters expressing the relative importance of each of the coordinate components.

The performance of the network is reported in the Results section.

3.2 Classification

The classification problem is defined as predicting, from a depth image of its initial location and pose, whether a garment will hang or fall to the floor when dropped near a hanger.

The architecture of the Deep Convolutional Network used as model for the classification problem is very similar to the one used for the regression problem. Figure 3 depicts the architecture of the network. A total of 4 sets of convolutional layers, and 2 fully connected layers to compute the output are used. Sets 1 and 3 include 2 convolutional layers with 16 filters of size (3x3) and ELU as the activation function, followed by a max-pooling layer of size (2x2) and stride 2. Sets 2 and 4 are composed of a single convolutional layer with 32 filters of size (3x3) and ELU as the activation function, followed by a max-pooling layer of size (2x2) and stride 2. The fully connected layers have 300 and 1 neuron, respectively. As only one class is to be predicted, instead of a Softmax function, a Sigmoid activation function is applied to the output neuron to obtain the probability of the garment falling to the floor given a certain input depth image. When the output is above 0.5, the prediction is that the garment will fall, otherwise the prediction is that it will hang. The total number of learnable parameters of this model is 2 099 705.

The binary labels for each training sample can be obtained from the trajectory recorded at each simulated trial, by observing the Z coordinate of the last point of the trajectory (Z_{end}). Based on the final location of the center of mass, and considering a threshold T_{floor} one can compute the binary label $floor$ as:

$$floor = \begin{cases} 1, & \text{if } Z_{end} < T_{floor} \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

The resulting labels are used as ground truth to train

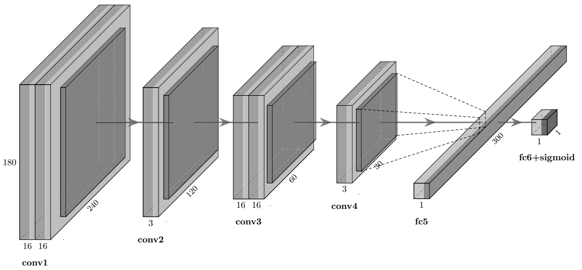

Figure 3: HANGnet_classify architecture diagram.

the classification network. For training, a Binary Cross Entropy loss is optimized by an Adam stochastic optimizer. This loss is weighted to account for the imbalance of the hanged/floor classes in the synthetic dataset.

4 Results

In this section we will discuss the training process of both models (regression and classification) as well as the results obtained with each of the models.

4.1 Regression

4.1.1 Setup

Following the simulation procedure described in section 2, a total of 15 000 training examples were obtained. Each of these training examples is composed by a depth image of the virtual setup at the initial time step (after the garment has reach a stable pose) and the trajectory of the center of mass of the garment once it has been dropped. The initial position of the garment is sampled from a normal distribution with $\mu_{init} = (0, 0, 1.5)$ m and $\sigma_{init} = (0.01, 0.4, 0.2)$ m. A duration of 51 simulation time steps has been selected as a good compromise between computational cost and reaching a stable hanged/floor state in simulation. Before they are fed to the network for training, the depth images are cropped at 2 m to remove the empty background and then normalized in the 0 m to 2 m range.

As the samples were obtained randomly, not every single sample results in the garment being hanged. In fact, the ratio of garments hanged/not hanged is near 1:3, making the dataset imbalanced. In order to deal with the imbalance in the dataset, stratification was used to make the training/validation/test splits, so that each of the three sets has the same proportion of examples of each of the two classes. The stratified split was the following: 20% of the samples (3000) were used for testing and, from the 80% remaining, 20% (2400) were used for validation and 80% (9600) for training.

For training the regression model, an Adam stochastic optimizer was used, with a learning rate of 0.0001. The custom loss introduced in section 3.1 (Eq. 1) was used, with weights $\omega_x = 0.033$, $\omega_y = 0.033$ and $\omega_z = 0.33$. The model was trained for 10 epochs, with a batch size of 32.

4.1.2 Results

To study the effect of the time on the uncertainty of the prediction and, therefore, on the accuracy of the model, the network was trained to predict the location at different time steps. Figure 4 shows the increase of the Mean Squared Error (MSE) as time advances.

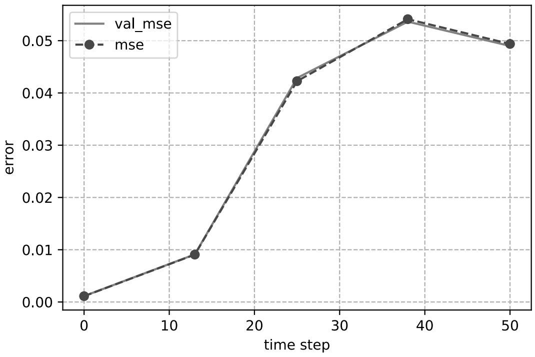

Figure 4: Mean Squared Error (MSE) and Validation MSE with respect to the time step.

In terms of coordinates, MSE was computed for each of the 3D components of the prediction (X, Y and Z). Figure 5 shows the MSE of each of the component, as well as the variation in error for each of them. It can be observed that the error increases from a few millimeters to several centimeters, and it is more dramatic in the case of the Z coordinate, which was an expected result, as the fact that the garment can remain randomly hanged or not hanged affects to the expected location in the Z axis.

4.2 Classification

4.2.1 Setup

For the classification model, the same dataset as the one described in section 4.1.1 was used, and the same

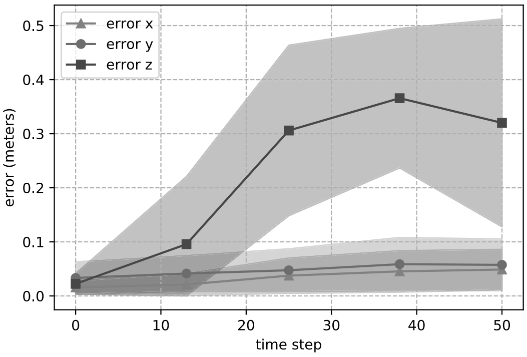

Figure 5: Mean Squared Error (MSE) of each of the 3 coordinates (X, Y, Z) with respect to the time step.

stratification techniques were use to deal with the imbalance of the dataset when performing the training/validation/test splits. As the output of the classification model is a single binary value representing the prediction of the network (0 if the garment with hang, or 0 if the garment will fall), the trajectory previously recorded in the dataset needs to be processed to obtain a binary label. For that purpose, eq. 2 was used, with a threshold value $T_{floor} = 0.81\,\mathrm{m}$.

For training the classification model, an Adam stochastic optimizer was used, with a learning rate of 0.0001. In addition, L2 regularization was added to the model, with a regularization strength of 0.01 to improve the generalization capabilities of the network.

4.2.2 Human baseline

To evaluate the relevance of the results achieved, a human baseline was obtained by labelling a subset of 200 elements from the training set by hand, based only on the input data. In other words, a depth image was presented to a human expert for him to predict whether it will hang or fall. To improve the human perception of depth in the input depth image, the 8-bit greyscale image was transformed using a GIST stern[`https://www.ncl.ucar.edu/Document/Graphics/ColorTables/MPL_gist_stern.shtml`, last accessed: 08-06-2019] colormap for display.

4.2.3 Results

After training, several metrics were computed based on the 3000 element test set, as reported in table 1. A subset of 200 elements were randomly selected and used to compute a human baseline and compare it with the performance of the classification model. Table 2 shows the results of this analysis. Although the model has a lower recall for the hanged class than the human expert, it improves the performance of the human in all remaining metrics considered.

Table 1: Classification model, 3000 items.

Class	Precision	Recall	F1-score	# items
Hanged	0.51	0.32	0.39	819
Floor	0.78	0.88	0.83	2181

Table 2: Classification model vs Human baseline, 200 items.

Class	Precision	Recall	F1-score	# items
Hanged	0.60	0.39	0.47	54
Floor	0.80	0.90	0.85	146
Hanged	0.38	0.56	0.45	54
Floor	0.80	0.66	0.72	146

In addition, confusion matrices were computed to compare the performance of the model and the human expert in terms of false positives / false negatives. As shown in Figure 6, the classification model outperforms the human expert when predicting garments that fell to the floor, while having a similar performance when predicting garments that remained hanged.

5 Conclusions

In this work we propose two different deep convolutional models to predict the behavior of a piece of clothing when dropped onto a hanger. One of models is able to predict whether the garment will hang or fall, and the other estimates the future location of the garment after it is dropped. A synthetic dataset composed of 15 000 examples obtained via deformable object simulation is used to train both models. Experiments performed with the regression model demonstrate an influence of time in the accuracy of the predictions, as uncertainty in the position estimation increases with time. The classification model performance was compared to the baseline performance of a human expert, obtaining a slightly better performance.

Our future work will focus on the integration of both models (regression and classification) to increase the individual accuracy of each of them, as well as in the implementation of these models on one or several robotic

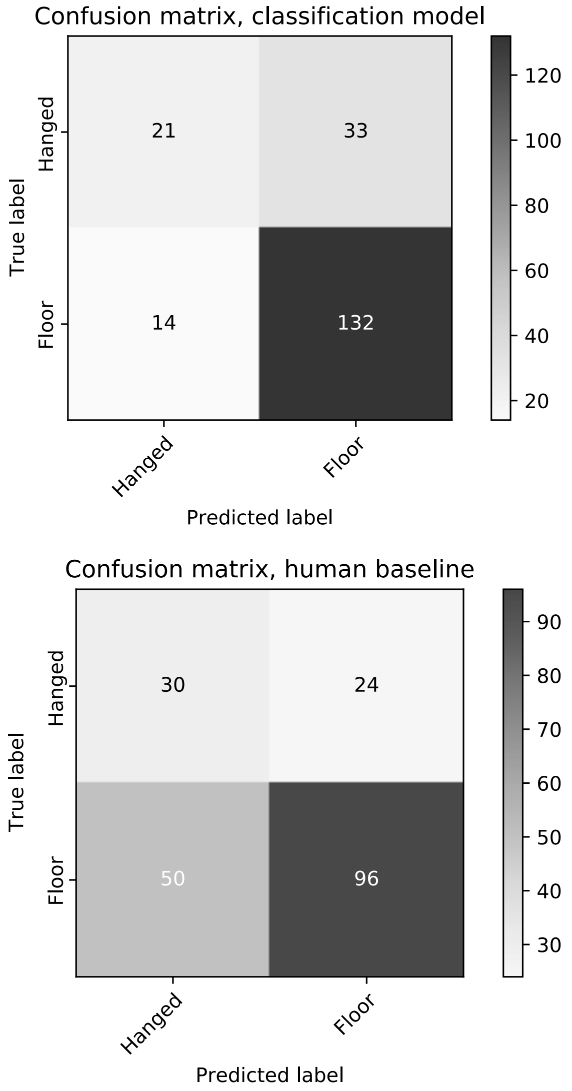

Figure 6: Confusion matrices for the classification model and human baseline.

platforms to obtain real world data and action, and further validate the idea behind this work.

Acknowledgement

This work was supported by RoboCity2030-III-CM project (S2013/MIT-2748), funded by Programas de Actividades I+D in Comunidad de Madrid and EU and by a FPU grant funded by Ministerio de Educación, Cultura y Deporte. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the NVIDIA Titan X GPU used for this research.

References

[1] Bersch C, Pitzer B, Kammel S. Bimanual robotic cloth manipulation for laundry folding. In: IEEE International Conference on Intelligent Robots and Systems. pp. 1413–1419 (2011).

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

[3] Doumanoglou A, Kim Tk, Zhao X, Malassiotis S. Active Random Forests: An Application to Autonomous Unfolding of Clothes. In: European Conference on Computer Vision (ECCV). pp. 644–658. Springer International Publishing (2014).

[4] Elbrechter C, Haschke R, Ritter H. Folding Paper with Anthropomorphic Robot Hands using Real-Time Physics-Based Modeling. 2012 12th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2012) pp. 210–215 (2012).

[5] Estevez D, Fernandez-Fernandez R, Victores JG, Balaguer C. Improving and Evaluating Robotic Garment Unfolding: A Garment-Agnostic Approach. In: IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC) (2017).

[6] Estevez D, Victores JG, Fernandez-Fernandez R, Balaguer C. Robotic Ironing with 3D Perception and Force/Torque Feedback in Household Environments. In: IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2017).

[7] Kita Y, Ueshiba T, Kanehiro F, Kita N. Recognizing clothing states using 3D data observed from multiple directions. In: International Conference on Humanoid Robots (Humanoids). pp. 227–233 (2013).

[8] Koganti N, Ngeo JG, Tomoya T, Ikeda K, Shibata T. Cloth dynamics modeling in latent spaces and its application to robotic clothing assistance. IEEE International Conference on Intelligent Robots and Systems **2015-Decem**, 3464–3469 (2015).

[9] Li Y, Hu X, Xu D, Yue Y, Grinspun E, Allen P. Multi-Sensor Surface Analysis for Robotic Ironing. In: IEEE International Conference on Robotics and Automation (ICRA). Stockholm (2016).

[10] Li Y, Wang Y, Case M, Chang Sf, Allen PK. Real-time Pose Estimation of Deformable Objects Using a Volumetric Approach. In: International Conference on Intelligent Robots and Systems (IROS). pp. 1046–1052. IEEE (2014).

[11] Mariolis I, Peleka G, Kargakos A, Malassiotis S. Pose and category recognition of highly deformable objects using deep learning. Proceedings of the 17th International Conference on Advanced Robotics, ICAR 2015 pp. 655–662 (2015).

[12] Matas J, James S, Davison AJ. Sim-to-Real Reinforcement Learning for Deformable Object Manipulation (CoRL) (2018).

[13] Miller S, Fritz M, Darrell T, Abbeel P. Parametrized shape models for clothing. In: International Conference on Robotics and Automation (ICRA). pp. 4861–4868 (2011).

[14] Schulman J, Lee A, Ho J, Abbeel P. Tracking deformable objects with point clouds. In: Proceedings - IEEE International Conference on Robotics and Automation. pp. 1130–1137. No. i, IEEE (2013).

[15] Seita D, Jamali N, Laskey M, Berenstein R, Tanwani AK, Baskaran P, Iba S, Canny J, Goldberg K. Robot Bed-Making: Deep Transfer Learning Using Depth Sensing of Deformable Fabric (2018).

[16] Stria J, Pruša D, Hlaváč V. Polygonal Models for Clothing. In: Advances in Autonomous Robotics Systems. vol. 8717, pp. 173–184 (2014).

[17] Sun L, Aragon-Camarasa G, Rogers S, Siebert JP. Accurate garment surface analysis using an active stereo robot head with application to dual-arm flattening. In: IEEE International Conference on Robotics and Automation (ICRA). vol. 2015-June, pp. 185–192. IEEE (2015).

[18] Triantafyllou D, Aspragathos NA. Upper layer extraction of a folded garment towards unfolding by a robot. In: Mechanisms and Machine Science, vol. 67, pp. 597–604 (2019).

[19] Willimon B, Walker I, Birchfield S. 3D Non-Rigid Deformable Surface Estimation Without Feature Correspondence. 2013 IEEE International Conference on Robotics and Automation pp. 646–651 (2013).