Jan Stria, Vaclav Hlavac. Classification of Hanging Garments Using Learned Features Extracted from 3D Point Clouds. In IROS 2018: Proc. 31th International Conference on Intelligent Robots and Systems. Madrid, Spain, October 2018, IEEE, p. 5307-5312.

Classification of Hanging Garments Using Learned Features Extracted from 3D Point Clouds

Jan Stria, Václav Hlaváč

***Abstract*— The presented work deals with classification of garment categories including pants, shorts, shirts, T-shirts and towels. The knowledge of the garment category is crucial for its robotic manipulation. Our work focuses particularly on garments being held in a hanging state by a robotic arm. The input of our method is a set of depth maps taken from different viewpoints around the garment. The depths are fused into a single 3D point cloud. The cloud is fed into a convolutional neural network that transforms it into a single global feature vector. The network utilizes a generalized convolution operation defined over the local neighborhood of a point. It can deal with permutations of the input points. It was trained on a large dataset of common 3D objects. The extracted feature vector is classified with SVM trained on smaller datasets of garments. The proposed method was evaluated on publicly available data and compared to the original methods, achieving competitive performance and better generalization capability.**

I. Introduction

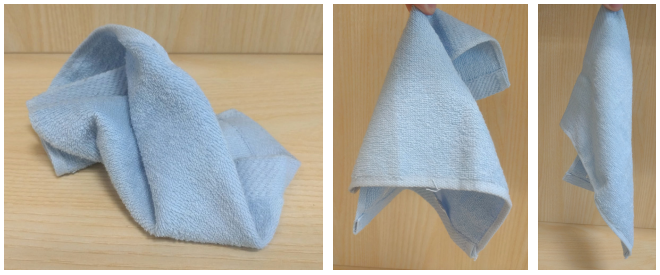
Visual perception of garments for their robotic manipulation is a challenging task. Since common garments are made mostly of *soft*, highly *deformable* materials, their appearance can vary significantly based on their actual state. This makes their recognition or pose estimation very difficult, as it is not easy to specify features invariant to the deformations.

The presented work deals with *classification* of an unknown deformed garment into several categories including pants, shorts, shirts, T-shirts and towels. Knowledge of the garment category is usually crucial for its robotic handling, because it determines the selected manipulation strategy.

There are basically two approaches to classify a randomly tossed garment (Fig. 1a). It can be perceived either *passively* or *actively* manipulated by a robot for better perception. The active approach usually consists of several steps [1]. A suitable *grasping point* is found on the surface of the crumpled garment at first, which can be e.g. a wrinkle or hemline. The garment is grasped and lifted by the robot (Fig. 1b). Optionally, the *lowest point* of the hanging garment is grasped with the other arm to reduce the space of its possible configurations (Fig. 1c). E.g. the lowest point of a hanging towel should always be its corner. The garment is then perceived with a camera, stereo rig or RGB-D sensor.

The input of our method is a set of depth maps captured from *many viewpoints* around the hanging garment (Fig. 2a). It is obtained by rotating a wrist of the robotic hand holding the garment around the vertical axis, while perceiving it with a stationary RGB-D sensor. The depth maps are fused into a single *point cloud* and the garment category is recognized.

The authors are with the Czech Institute of Informatics, Robotics and Cybernetics at the Czech Technical University in Prague, Czech Republic, {jan.stria, vaclav.hlavac}@cvut.cz



(a) Crumpled (b) Random (c) Lowest

Fig. 1: Various states of towel: a) crumpled and laying on a table, b) grasped at random point and hanging, c) regrasped for the lowest point.

The *key contributions* of the proposed work are:

- We introduce a novel convolutional neural network (CNN) architecture for classifying an unstructured 3D point cloud. It employs a generalized convolution over a spatial neighborhood of a point. The CNN is trained on a large dataset of 3D objects to learn extraction of distinctive features transferable to various domains.
- The intermediate outputs of the network are used as features for an SVM classifier of garment categories that was trained on smaller datasets.
- Contrary to the existing methods for garments classification that were only evaluated on their own datasets, we compare our classifier on existing data. We achieve competitive performance and better generalization.

II. Related work

A. Garments classification

Classification of a garment being held in a *hanging state* (Fig. 1b, 1c) was pioneered by Kita *et al.* [2]. They build a planar mass-spring model for each garment category that is virtually grasped at different points and hung up. Silhouettes of the garment perceived from two viewpoints are matched to the simulated models. The method was improved [3] by using more physically plausible cloth simulation included in Maya 3D modeling software. The model is matched to a 3D point cloud reconstructed by a trinocular camera system.

Willimon *et al.* [4] employ the active perception strategy. An unknown garment is grasped repeatedly at various random locations and perceived from two perpendicular viewpoints. The extracted silhouettes and edges are matched to annotated template images.

Li *et al.* [5] build an SVM classifier using quantized SIFT features extracted from a depth map. It is trained on the artificial dataset of hanging garments simulated in Maya

software. An unknown garment is rotated and perceived from 150–200 viewpoints, each of them classified independently. The final classification is obtained by majority voting. The method was sped up by merging depth maps from individual viewpoints to a volumetric representation [6], which is an alternative to the point cloud representation used by our method. The closest virtual model is found by minimizing the weighted Hamming distance.

Doumanoglou *et al.* [7] train a random forest on simple features extracted from a depth map, including depth difference of two points or curvature estimated in certain point. The classifier is used in partially observable Markov decision process (POMPD) framework that decides whether the hanging garment should be rotated and perceived from another viewpoint for better confidence. In their later work [8], the next best view is selected by the decision forest.

Instead of using handcrafted features, there have been attempts recently to learn the whole classification pipeline in the form of a CNN [9]–[12]. All proposed architectures share similar properties. The input is a depth map depicting the hanging garment. The networks are rather shallow, comprising 3 convolutional followed by 3 fully connected layers with hyperbolic tangent activations [9], [10], or 4 and 2 layers with ReLU activations [11], [12], respectively. The networks are trained with SGD on thousands real samples accompanied with significantly more depth maps generated by a simulator. It may be, however, problematic to ensure sufficient diversity of synthetic data. The accuracy is improved by aggregating classifications from multiple viewpoints.

Classification of a crumpled garment *laying on a table* (Fig. 1a) is even a more challenging task. Willimon *et al.* [13] propose a multi-level architecture. They use a combination of standard image features and depth features to train separate SVM classifiers for the selected mid-level characteristics, including presence of a collar or used material. These characteristics are then used for high-level category classification.

Sun *et al.* [14] train SVM directly on the combination of local and global depth features. In their related work [15], the garment is grasped, shaken or flipped, and tossed again after each perception stage, so that its configuration changes before the next perception. The classification confidences are modeled, tracked and updated by a Gaussian process.

Ramisa *et al.* [16] introduce their own descriptor called FINDDD that was developed specifically for textiles. It is based on quantized normals estimated from a depth map. They build a bag-of-words feature vector on top and classify it with an SVM model.

Yamazaki [17] filter the grayscale image of the crumpled garment with Gabor filters to detect overlaps and wrinkles at various scales and orientations. Their geometric properties are measured, discretized and accumulated into histograms to obtain descriptors that are classified with SVM.

B. Neural networks for depth and 3D data

There are basically three approaches how to employ CNNs for depth data classification. First, a standard 2D CNN known from image understanding can be applied on a *depth map* [18]. This is used by the existing garment classification methods [9]–[12], which further improve the accuracy by voting over multiple viewpoints. The network can eventually work with multiple views inherently [19], building its internal 3D representation. We rather reconstruct the 3D model explicitly by employing a well geometrically formulated algorithm (Sec. III-A).

There are CNN architectures available that work with the *volumetric representation* of an object [20], [21]. The main advantage is the neighborhood of cells in the volumetric grid is well defined and thus application of a discrete 3D convolution is straightforward. The disadvantage is a sparsity of the volumetric representation that makes the computation expensive and increases the number of the network free parameters. Moreover, the volumetric representation depends on 3D pose of the object and discretization of the grid.

There have been also attempts to apply the convolution on *unstructured 3D point clouds* [22], [23]. The point cloud representation of an object is very compact, yet powerful. Individual parts of the object can be, moreover, sampled with different densities. The pioneering work called PointNet [22] applies convolution on each point separately to deal with their unknown permutation in the cloud. The convolutions therefore serve as point-wise feature transformers only. The architecture was later improved [23] by max-pooling the features over neighboring points in a hierarchical manner. In contrast, the proposed work generalizes the convolution operation itself over the neighborhood of a point, enabling computation of richer local features.

III. Method description

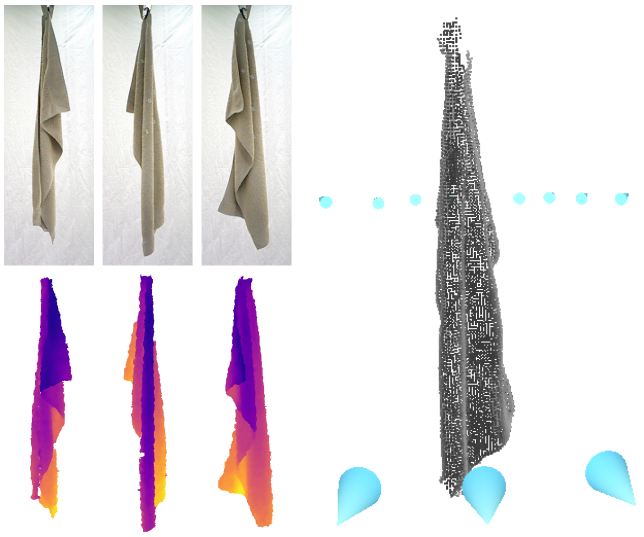
Our method for category classification of hanging garments consists of the following three stages:

1) Depth maps depicting the hanging garment from various viewpoints are fused into a single 3D point cloud.
2) The point cloud is processed by CNN that transforms it to a single distinctive global feature vector.
3) The garment category is classified with SVM.

A. Reconstruction of 3D point cloud

It is assumed that the unknown garment was already grasped by the robot, lifted up and it is now being held in a *hanging state* (Fig. 1b, 1c). The proposed work does not deal with the manipulation that was already described in our previous paper [1]. It is assumed that the garment is perceived from *many viewpoints* distributed around it (Fig. 2a). It can be achieved by rotating the robotic wrist holding the garment around the vertical axis, while perceiving it with a sensor attached to the other arm [1]. It is finally assumed that the wrist is rotated reasonably slowly, e.g. several seconds per rotation, so that the configuration of the hanging garment does not change by fluttering.

Our approach to garments classification is based purely on *depth data* coming from an RGB-D sensor. This ensures good generalization over all possible colors and textures. Assuming that the sensor and robot are properly calibrated, it is straightforward to *segment* the garment from its surroundings



(a) Input images and depths (b) Reconstructed point cloud

Fig. 2: a) The input depth maps acquired from multiple viewpoints around the garment are segmented and fused. b) The reconstructed 3D point cloud with the estimated sensor poses visualized as cones.

by keeping only the data from a properly sized cuboid or cylinder below the gripper holding the garment. The cuboid is defined manually in our experiments on existing datasets.

The segmented depth maps (Fig. 2a) are fed into Kinect Fusion algorithm [24]. It fuses the depth maps from multiple viewpoints into a single global *dense surface model*, while tracking the *sensor pose* over time (Fig. 2b). The model is represented volumetrically and refined incrementally. Each cell holds a truncated signed distance function (TSDF) value that is negative for the points inside the object and positive for those outside. The reconstructed cloud contains the points whose neighbors have an opposite sign of TSDF (Fig. 2b).

We use the optimized KinFu[1] implementation of the fusion algorithm in CUDA. It delivers real-time 3D reconstructions on the low-level GPU Nvidia GT 730M. We use 256^3 volumetric grid to represent the reconstructed cubical space of edge length 200 cm. Density of the reconstructed 3D point cloud is therefore 7.8 mm, which corresponds approximately to the depth quantization step used by commonly available RGB-D sensors. We have found out that it suffices to rotate the garment once with 10° step between the individual viewpoints to obtain a precise enough reconstruction.

B. Network architecture

The reconstructed 3D point cloud usually contains thousands to tens thousands points. It is downsampled by selecting $n = 1024$ point randomly, which is enough for a suitable representation of its original shape. The sampled cloud is translated to zero mean and scaled to a unit ball. It is then processed by the CNN.

The proposed network architecture (Fig. 3) is based on the idea introduced by PointNet network [22]. In order to achieve the *invariance to permutations* of the input points, the local

[1]KinFu library: `github.com/Nerei/kinfu_remake`

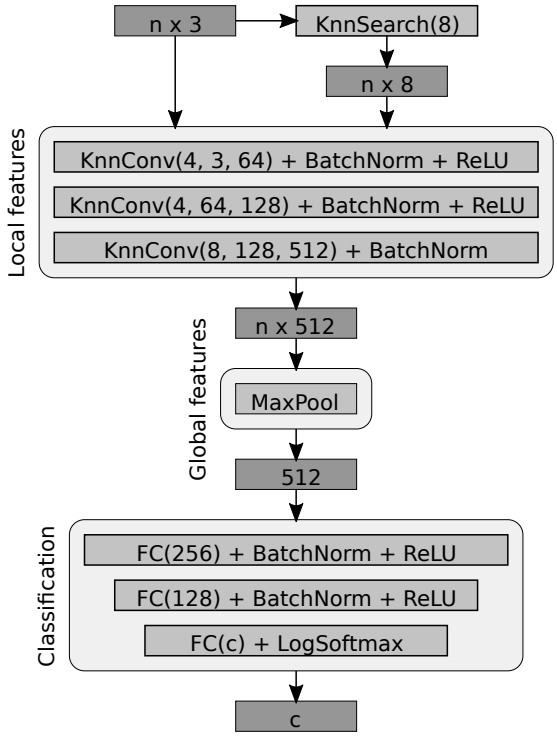


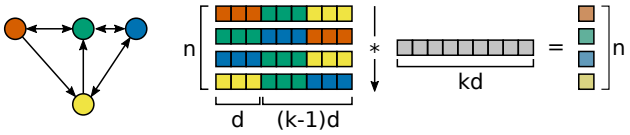
Fig. 3: Proposed network architecture. The input 3D cloud consisting of n points is searched for k-NN of each point. The convolutional layers extract 512D feature vector from the local neighborhood of each point. The local features are max-pooled to a single 512D global feature vector that is classified by the fully connected layers.

features are computed for each point separately by the network. Local features for all points are then aggregated to a single global vector by applying a function symmetric in its arguments, e.g. max pooling, average pooling or summation.

On contrary, our network computes the local features on k *nearest neighbors* of each point. The k-NN relation and the ordering of k-NN according to their distance is not only invariant to permutations of points in the cloud, but also to its translation, rotation and scaling. Knowing k-NN of each 3D point, the network is able to learn features that could not be computed for isolated points, e.g. estimation of surface normals or curvatures.

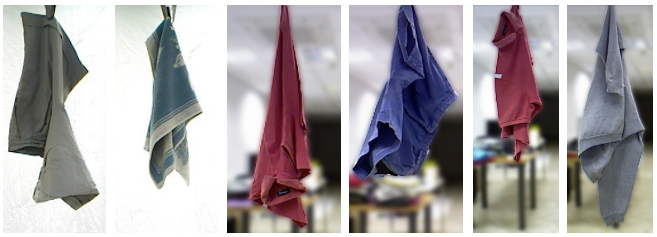
Fig. 3 shows the proposed network. Its input is $n \times 3$ tensor of points. They are fed into KnnSearch(k) layer that builds $n \times k$ tensor, containing for each input point the indices of its k-NN sorted in ascending order according to their distance. We use the convention that each point is included in its own neighborhood, i.e. only $k - 1$ neighbors are found in fact. The parameter $k = 8$ is chosen as the maximum size over the neighborhoods used by all feature extraction layers. The search computes all $O(n^2)$ point to point distances, which is a sufficient solution, given that the distances are evaluated and sorted in parallel on GPU.

The local features are extracted by a stack of *convolutional layers* KnnConv(k_l, d_l, m_l). The parameter k_l denotes neighborhood size for l-th layer, d_l is dimension of input feature vectors and m_l is convolution kernels count. It holds $d_1 = 3$



(a) Neighbors (b) Convolution on concatenated features

Fig. 4: Schematic view of convolution over the nearest neighbors. a) Each of $n = 4$ points is found its $k = 3$ spatially closest neighbors including itself (arrows pointing to the neighbors). b) For each point, k feature vectors of dimension $d = 3$ are concatenated and convolved with $1 \times kd = 1 \times 9$ kernel (along the arrow).



(a) Random point (b) Lowest point (c) Unfolding point

Fig. 5: Datasets of hanging garments used in the evaluation: a) randomly grasped garments [10]; b) garments regrasped for the lowest point followed by c) the first unfolding point [8].

and $d_{l+1} = m_l$. The convolution over the nearest neighbors is implemented as 1D convolution of $n \times k_l d_l$ tensor with $1 \times k_l d_l$ kernel (Fig. 4). The product $k_l d_l$ denotes the number of channels in 1D convolution. The channels are formed by concatenating d_l-dimensional vectors corresponding to k_l-NN of the point in ascending order according to their distance. The convolution outputs are batch normalized. ReLU activation is applied at first two layers.

The output of the feature extraction module is $n \times 512$ tensor, i.e. 512D feature vector for each of n input points. The local feature vectors are max pooled over all n points to obtain a single 512D global vector. It is classified by a stack of three *fully connected layers* FC(c_l), each containing c_l neurons. The first two layers are batch normalized and use ReLU activation. The output layer uses the logarithm of softmax function to predict the classification log probabilities over c output classes.

The network contains 732k trainable parameters, of which 560k are weights in the convolutional layers and 172k in the fully connected layers. The feature extraction part of the network is able to learn *efficient and robust global representation* of the input 3D point cloud (Sec. III-C, III-D).

C. Network training

The existing neural networks for garments classification [9]–[12] were trained from scratch on datasets of hanging garments. The issue of not having enough training data is overcome partially by generating them with a cloth simulation engine included in Blender [9] or Maya [11]. However, it is difficult to ensure sufficient diversity of the synthetic data to avoid overfitting.

We rather train the network on ShapeNet [25], which is arguably the largest publicly available dataset of annotated 3D shapes. It contains more than 42k 3D mesh models over $c = 55$ common categories, including vehicles, household equipment, electronics etc. We believe that by training our network on such a variety of objects, the convolutional layers learn to extract general enough features that can be transfered to a new domain of garments. The dataset is split to 90% training and 10% validation subset, which was used to develop the network architecture and optimize its hyperparameters.

Each 3D mesh model from ShapeNet is converted to a 3D point cloud by sampling 2048 uniformly distributed points on its surface. The clouds are normalized to be zero meaned and located inside a unit ball. Following augmentations are employed in each training epoch. A subset of $n = 1024$ points is sampled randomly, which ensures that the same cloud is almost never seen twice. Then it is rotated randomly around the vertical axis in range $[0, 2\pi)$.

The network is implemented in PyTorch[2]. It is trained with Adam [26] algorithm, using the learning rate $\alpha = 0.001$ and the recommended values of hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The categorical cross entropy loss is minimized. The network is trained for 100 epochs in batches of 32. The training takes less than 10 hours on the single GPU Nvidia Tesla K40m. The classification accuracy on the validation subset of ShapeNet is 75%.

D. Classification of garment category

While *transfering* a pretrained network to a new domain, a common approach is to freeze the weights in the feature extraction layers and retrain the classification layers. Since the available datasets of garments are rather small, we only use the network to compute a global 512D feature vector (Fig. 3) describing the garment. The vector is then classified with *linear SVM* that has significantly less parameters than the stack of fully connected layers.

We use the LIBSVM [27] implementation of SVM. The multiclass classifier is trained with one-vs-all strategy, optimizing a squared hinge loss. The regularization hyperparameter C, which weights the error term in standard formulation of SVM [27], is found by cross-validation on the training set.

IV. Experimental evaluation

The vast majority of existing works on garments classification evaluate their methods on own data only. We rather use the publicly available datasets to compare performance of the proposed method to the original works. We acquired two datasets of hanging garments, grasped either *randomly* [10] or for the *lowest point* [8]. Both datasets contain images and depth maps taken from multiple viewpoints around the garment, which is needed to reconstruct the 3D point clouds (Sec. III-A). All data were acquired by a low-cost RGB-D sensor ASUS Xtion.

The datasets were only used for training in the original works. The evaluation was performed on additional testing

[2]Implementation of the proposed network in PyTorch and weights trained on ShapeNet: `cmp.felk.cvut.cz/~striajan/iros2018`

Method	Pants	Shorts	Shirts	T-shirts	Towels	Overall
RF [7]	0.47	0.73	0.82	0.80	0.72	0.75
dCNN [9]	0.60	0.53	0.84	0.76	0.48	0.70
RF+dCNN [10]	0.56	0.67	0.88	0.90	0.78	0.82
Ours	0.89	0.70	0.91	0.84	0.39	0.82

TABLE I: Comparison of garment category classification accuracies evaluated on the dataset of randomly grasped hanging garments [10].

True\Pred.	Shirts	Shorts	Pants	T-shirts
Shirts	0.88	0.01	0.02	0.09
Shorts	0.00	0.90	0.00	0.10
Pants	0.03	0.03	0.94	0.00
T-shirts	0.02	0.03	0.00	0.95

TABLE II: Confusion matrix for the proposed classifier evaluated on the dataset of garments regrasped for the lowest point [8]. Rows correspond to true garment categories, columns to predictions.

True\Pred.	Shirts	Shorts	Pants	T-shirts
Shirts	0.83	0.00	0.03	0.14
Shorts	0.00	0.85	0.02	0.13
Pants	0.06	0.02	0.89	0.03
T-shirts	0.07	0.03	0.03	0.87

TABLE III: Confusion matrix for the proposed classifier evaluated on the dataset of garments being held for the lowest or unfolding point [8]. Rows correspond to true garment categories, columns to predictions.

datasets that are unfortunately no longer available [8] or do not contain full sequences of viewpoints [10]. We therefore use the original training datasets both for training and evaluation of our method and compare it to performance of the competing methods reported on the original testing data.

The evaluation of our method is performed in *leave-one-out* manner. One clothing item per each category is put into the testing set and the SVM classifier (Sec. III-D) is trained on the remaining items in each iteration. The process is repeated as many times as there are unique garments per category. The classification results are averaged. Cross-validation of SVM hyperparameters is performed again in leave-one-out manner on each training set independently.

A. Hanging garments grasped randomly

The dataset of randomly grasped garments [10] contains 16 unique items of 5 categories (Fig. 5a). Each garment was grasped by a robot at various points on its surface and hung up. The dataset comprises 3 pants $\times$ 59 grasping points, 3×74 shirts, 3×20 shorts, 4×56 T-shirts and 3×25 towels. That is 758 combinations in total, each perceived from 180 viewpoints. We use only 90 views for 3D point clouds reconstruction.

As described in Sec. II-A, the original work [9] applies a CNN directly on the depth maps. To improve the classification accuracy, it was later combined [10] with the approach based on random forests [7]. Predictions from multiple viewpoints are aggregated, which can be considered an alternative to our fusion of depths.

Tab. I shows the comparison of the classification results. Overall accuracy of our method is 82% which is on par with the current state of the art approach combining CNN and random forest. The main source of our failures are towels that have no distinguishable parts like collars or sleeves. Their shape can, moreover, resemble shorts (Fig. 5a).

B. Hanging garments regrasped for the lowest point

The dataset [8] contains 4 clothing categories: pants, shirts, shorts and T-shirts. Each category is represented by 6 unique items. Each garment was grasped by a robot, lifted up and regrasped for its lowest point (Fig. 5b). There is 1 possible lowest point for pants and shirts, 2 lowest points for shorts and T-shirts. Each combination of item and the lowest point occurs 20 times. This is 720 sequences in total, each comprehending data from 40 viewpoints.

As stated in Sec. II-A, the original work [7] applies random forest (RF) classifier on simple features computed from a depth map. Information from more views is aggregated by POMPD. The improved method [8] aggregates directly in the RF, which also selects the next best viewpoint. Only several views are needed for a decision, compared to tens views distributed uniformly around the garment required by our method. On the other hand, time spent by rotating the wrist holding the garment is negligible compared to other manipulation steps [1].

We achieve 92% overall classification accuracy, consistent over all classes (Tab. II), while both original works [7], [8] claim perfect 100% accuracy. The RF approach, however, has inferior performance on randomly grasped garments (Tab. I). Since our method normalizes the input point cloud to a unit ball, it distinguishes the garment category purely on its shape. On contrary, the RF can learn to classify the garments based on their size, since e.g. shorts usually occupy smaller portion of the depth map than pants, which may not be desired behavior. Moreover, due to normalization, our method is arguably easier transferable to different robotic setups, where a sensor with different resolution is used or its distance to the garment varies over time.

The dataset [8] contains additional 360 sequences of garments that have been already unfolded partially by recognizing and grasping a certain point, which is a shoulder for shirts and T-shirts, and a waist corner for shorts and pants (Fig. 5c). These data are not used for classification in the original work [8], because the garment category must be known before unfolding. We mixed these 360 sequences with 720 sequences for the lowest point. The resulting dataset should therefore be more complex than the lowest point one, but simpler than the one with random grasping points. The achieved overall recognition accuracy 86% proves that. Tab. III shows a confusion matrix for individual categories.

V. Conclusion

We proposed a method for classification of garments being held in a hanging state (Fig. 1b, 1c). We showed that instead of using handcrafted features, as suggested by the majority of works, they can be learned efficiently by the CNN. Our network is one of the first architectures taking an unstructured 3D point cloud on its input and the first such network used for garments classification. The network

is trained on the large dataset of general 3D objects to learn distinctive features, instead of training it from scratch on simulated garments as in the existing works.

The presented work is the first that evaluates its performance on existing datasets and compares it to the original methods. It achieves the accuracy comparable with the current state-of-the-art. Moreover, it is more robust to variations in sizes of the garments and arguably better transferable to different robotic setups.

In future, we would like to extend our method to predict which point on the surface of the garment needs to be grasped next to unfold it. We believe that it is possible to reuse the feature extraction part of the network, replacing only the SVM category classifier with regression of the grasping point. Another possible extension is modification of the network to perform a semantic segmentation of the garment parts to classes, e.g. sleeves, collars or pockets, that could be used for better informed manipulation.

ACKNOWLEDGMENT

This work was supported by the European Regional Development Fund under the projects Robotics for Industry 4.0 (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000470) and IMPACT (reg. no. CZ.02.1.01/0.0/0.0/15_003/0000468), and by the Grant Agency of the Czech Technical University in Prague under the projects SGS15/204/OHK3/3T/13 and SGS18/205/OHK3/3T/37.

REFERENCES

[1] A. Doumanoglou, J. Stria, G. Peleka, I. Mariolis, V. Petrík, A. Kargakos, L. Wagner, V. Hlaváč, T.-K. Kim, and S. Malassiotis, "Folding clothes autonomously: A complete pipeline," *IEEE Trans. on Robotics*, vol. 32, no. 6, pp. 1461–1478, 2016.

[2] Y. Kita, F. Saito, and N. Kita, "A deformable model driven visual method for handling clothes," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2004, pp. 3889–3895.

[3] Y. Kita, T. Ueshiba, E. S. Neo, and N. Kita, "Clothes state recognition using 3D observed data," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2009, pp. 1220–1225.

[4] B. Willimon, S. Birchfield, and I. Walker, "Classification of clothing using interactive perception," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 1862–1868.

[5] Y. Li, C.-F. Chen, and P. K. Allen, "Recognition of deformable object category and pose," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 5558–5564.

[6] Y. Li, Y. Wang, M. Case, S.-F. Chang, and P. K. Allen, "Real-time pose estimation of deformable objects using a volumetric approach," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2014, pp. 1046–1052.

[7] A. Doumanoglou, A. Kargakos, T.-K. Kim, and S. Malassiotis, "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2014, pp. 987–993.

[8] A. Doumanoglou, T.-K. Kim, X. Zhao, and S. Malassiotis, "Active random forests: An application to autonomous unfolding of clothes," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014, pp. 644–658.

[9] I. Mariolis, G. Peleka, A. Kargakos, and S. Malassiotis, "Pose and category recognition of highly deformable objects using deep learning," in *Proc. Int. Conf. on Advanced Robotics (ICAR)*, 2015, pp. 655–662.

[10] C. Kampouris, I. Mariolis, G. Peleka, E. Skartados, A. Kargakos, D. Triantafyllou, and S. Malassiotis, "Multi-sensorial and explorative recognition of garments and their material properties in unconstrained environment," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016, pp. 1656–1663.

[11] A. Gabas, E. Corona, G. Alenyà, and C. Torras, "Robot-aided cloth classification using depth information and CNNs," in *Proc. Articulated Motion and Deformable Objects (AMDO)*, 2016, pp. 16–23.

[12] E. Corona, G. Alenyà, A. Gabas, and C. Torras, "Active garment recognition and target grasping point detection using deep learning," *Pattern Recognition*, vol. 74, pp. 629–641, 2018.

[13] B. Willimon, I. Walker, and S. Birchfield, "A new approach to clothing classification using mid-level layers," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 4271–4278.

[14] L. Sun, G. Aragon-Camarasa, S. Rogers, R. Stolkin, and J. P. Siebert, "Single-shot clothing category recognition in free-configurations with application to autonomous clothes sorting," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2017, pp. 6699–6706.

[15] L. Sun, S. Rogers, G. Aragon-Camarasa, and J. P. Siebert, "Recognising the clothing categories from free-configuration using Gaussian-process-based interactive perception," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2016, pp. 2464–2470.

[16] A. Ramisa, G. Alenyà, F. Moreno-Noguer, and C. Torras, "A 3D descriptor to detect task-oriented grasping points in clothing," *Pattern Recognition*, vol. 60, pp. 936–948, 2016.

[17] K. Yamazaki, "A method of classifying crumpled clothing based on image features derived from clothing fabrics and wrinkles," *Autonomous Robots*, vol. 41, no. 4, pp. 865–879, 2017.

[18] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proc. European Conf. on Computer Vision (ECCV)*, 2014, pp. 345–360.

[19] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3D shape recognition," in *Proc. IEEE Int. Conf. on Computer Vision (ICCV)*, 2015, pp. 945–953.

[20] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2015, pp. 922–928.

[21] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1912–1920.

[22] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 652–660.

[23] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5105–5114.

[24] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "Kinectfusion: Real-time dense surface mapping and tracking," in *Proc. IEEE Int. Symp. on Mixed and Augmented Reality (ISMAR)*, 2011, pp. 127–136.

[25] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, "ShapeNet: An information-rich 3D model repository," *arXiv:1512.03012*, 2015.

[26] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[27] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. on Intelligent Systems and Technology*, vol. 2, no. 3, p. 27, 2011.