

# Garment Perception and its Folding Using a Dual-arm Robot

Jan Stria, Daniel Průša, Václav Hlaváč, Libor Wagner, Vladimír Petrík, Pavel Krsek, Vladimír Smutný<sup>1</sup>

**Abstract**—The work addresses the problem of clothing perception and manipulation by a two armed industrial robot aiming at a real-time automated folding of a piece of garment spread out on a flat surface. A complete solution combining vision sensing, garment segmentation and understanding, planning of the manipulation and its real execution on a robot is proposed. A new polygonal model of a garment is introduced. Fitting the model into a segmented garment contour is used to detect garment landmark points. It is shown how folded variants of the unfolded model can be derived automatically. Universality and usefulness of the model is demonstrated by its favorable performance within the whole folding procedure which is applicable to a variety of garments categories (towel, pants, shirt, etc.) and evaluated experimentally using the two armed robot. The principal novelty with respect to the state of the art is in the new garment polygonal model and its manipulation planning algorithm which leads to the speed up by two orders of magnitude.

## I. INTRODUCTION

The reported research contributes to the garment sensing and its manipulation (sorting, folding, etc.) which is performed on a dual arm robot in our case within the European Commission funded project CloPeMa [1]. The project advances the state of the art in the autonomous perception and manipulation of limp materials like fabrics, textiles and garments. The emphasis is put at universality and robustness.

We propose a method for an autonomous folding of a piece of garment spread out on a flat surface. Our aim is to provide a real-time procedure which is applicable to an extended collection of garments of various shapes and to give satisfactory results when employed to a real robot. The method involves vision sensing, garment understanding, planning and manipulation tasks.

The similar objective has already been approached by Miller et al. [2], [3] and applied to a garment folding on Willow Garage PR2 robot. The authors consider a garment fairly spread on an uniform green surface which extremely simplifies the segmentation task. A contour is obtained by segmenting a single image taken by the robot camera and it is fitted to a parametric polygonal model specific for a particular category of garment. The fitting procedure is an iterative estimation of numeric parameters of the model. Quite accurate fitting results are reported. However, the main drawback is the slow performance as it takes 30–150 seconds to fit a single contour, depending on a complexity of the particular model. The same parametric polygonal model was

used for socks configuration recognition and for pairing corresponding socks [4]. The previously used global model matching approach is combined with local fitting of texture and shape descriptors in this work.

An automated folding of towels based on a robust visual detection of their corner points is presented in [5]. The PR2 robot starts with a towel dropped on a table and folds it in a sequence of manipulations performed both on the table and in the air. This problem is also partially solved by Hata et al. [6] who are interested in lifting a single towel from a pile of highly wrinkled towels. The towel is then regrasped to be held for its corner in order to simplify future manipulation. The solution is based on detecting the highest point of the pile followed by corner detections in stereo data. Ramisa et al. [7], [8] combine features computed from both color and range images to define a measure of clothes wrinkledness. The identified highly wrinkled regions are good candidates for automated grasping.

State estimation of the hanging clothes being held by a single gripper is approached by Kita et al. They utilize both single-view [9] and stereo [10] images. The method is based on matching the observed data to many precomputed deformable 3D models, each one for an individual grasping location, and selecting the best fitting model. The recognition accuracy can be improved by pushing the hanging garment with the second robotic hand in order to bring it into a more distinguishable position [11]. Doumanoglou et al. [12] are interested in category and pose recognition of a hanging garment in order to bring it into the desired position. They work with depth images acquired by a range sensor. The recognition algorithm utilizes decision forests trained on very simple local features. The manipulation planning is formulated in a probabilistic Markov framework. Willimon et al. [13] estimate configuration of moderately wrinkled clothes by fitting a triangulated surface to the observed range image. The fitting procedure is based on iterative minimization of the energy function which expresses surface smoothness and both visual and spatial similarity of the data.

We already explained the computer vision basics of our approach in [14] where the former method detected unfolded garments which allowed us to launch a scripted sensing-less sequence of robot actions to fold a piece of garment. Here we present a more accurate polygonal model incorporating relative lengths of contour segments. In addition, we propose a generic procedure allowing to derive folded models from the basic, unfolded ones. A new dynamic programming-based algorithm for matching the model to observed data is proposed. This work also documents its robotics side, i.e. introducing our dual arm testbed and performing proposed

<sup>1</sup>All authors are with Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic. Contacts to all authors are to be found at our web page <http://cmp.felk.cvut.cz>.

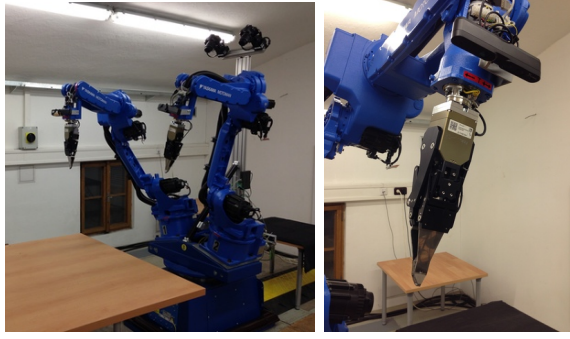


Fig. 1: Testbed and a detail of the arm with a mounted gripper and Xtion on the wrist.

methods on it. The testbed used within CloPeMa is documented in Sec. II. All vision related models and algorithms are described in Sec. III. Sec. IV gives brief overview of the automated manipulation planning and execution. Experiments, accuracy of recognition and the performance of the folding by the robot are evaluated in Sec. V.

## II. TESTBED DESCRIPTION

CloPeMa testbed consists of two Motoman MA1400 robotic arms, the R750 turn-table and two DX100 controllers. The robotic arms are mounted on the turn-table tilted from the vertical axis and the angle between them is  $30^\circ$ . The arms and the turn-table are driven by two controllers (master and slave) at a low-level.

The high-level control system is built on the Robot Operating System [15] (ROS Hydro). The system runs on a PC connected to the master controller by a local Ethernet. Basic software allows to move the robotic arms to required positions as well as to read actual positions of the arms joints. This functionality is supported by MotoROS package, distributed by the robot manufacturer, and extended to support the dual arm robot.

The testbed is fitted with sensors and grippers. Xtion range finder sensors are attached on the wrists of both arms. Additional Xtion sensor is mounted to a camera-head holder. The binocular head [16] is placed on the top of the holder. Two grippers specialized to garment manipulation [17] are attached to robot wrists. Each arm is also equipped with a force and torque sensor in the wrists. Fig. 1 shows the testbed and a detail of the gripper and Xtion sensor on the wrist. The reported work uses only Xtion vision sensors.

## III. VISION SENSING AND UNDERSTANDING

The crucial task in our system for automated folding is to recognize the configuration of a piece of garment to be manipulated by the robot. This is performed as a computer vision task utilizing a single color image captured by the Xtion camera mounted on the robotic arm. The algorithm precedes the folding sequence to determine the initial configuration of the spread garment. Vision sensing is also repeated after performing each single fold, utilizing results from the previous runs.

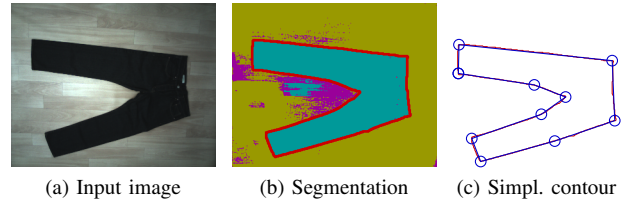


Fig. 2: Pixels of the input image are used to initialize trimap for grabcut algorithm. The trimap consists of foreground (plotted in cyan), background (yellow) and unknown (magenta) pixels. The resulting segmentation gives a garment contour (red) which is approximated by a polygon (blue).

The vision procedure can be split into several steps:

- The input is formed by a color image of a piece of garment placed on a table. The color of the table with a wooden-like surface differs from the garment color.
- The garment location in the image is determined automatically by a segmentation method trained from data.
- The contour of the garment is extracted from the segmentation mask and approximated by a polygon, which reduces the number of contour points (now polygon vertices) significantly.
- The approximated polygonal contour is matched to a polygonal model specific for a particular category of clothing. Once the polygonal model is matched, positions of the contour landmark points are known.

### A. Segmentation

Segmentation is the first phase of the recognition pipeline. Since the table beneath the garment is unchanged, its color can be learned from training data. We model the table color probabilistically as a Gaussian mixture model (GMM) of RGB triples. Components of the GMM are initialized by a binary tree algorithm for palette design [18] which repeatedly splits RGB vectors to subsets in a direction with the greatest variance. The number of GMM components is determined empirically to model variability of the table color sufficiently. We use 3 components in our case of a wooden table. The prior probabilities, mean vectors and covariance matrices for individual components are learned according to the maximum likelihood principle [19].

The color of the garment lying on the table is unknown. However, we suppose that it is sufficiently different from the color of the table. Thus pixels visualizing the table should have higher probability in the trained GMM than pixels visualizing an unknown garment. Based on this assumption, we label the  $n$ -th pixel by label  $t_n$  according to probability  $p(z_n)$  of its color  $z_n$  in the trained GMM model:

$$t_n = \begin{cases} \text{foreground,} & p(z_n) < P_F \\ \text{unknown,} & P_F \leq p(z_n) \leq P_B \\ \text{background,} & P_B < p(z_n) \end{cases} \quad (1)$$

The probability thresholds  $P_F$  and  $P_B$  are chosen so that 3% training pixels have probability lower than  $P_F$  and 80%

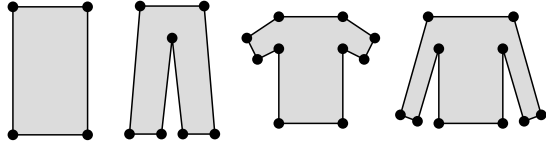


Fig. 3: Polygonal models for all currently supported categories of clothing (towel, pants, short-sleeved shirt and long-sleeved shirt).

of them have probability higher than  $P_B$ . Fig. 2b shows example of such a labeling.

The described labeling is used to automatically initialize color models of the grabcut segmentation algorithm [20]. This is done instead of requiring the user to draw a stroke over foreground (garment) and background (table) pixels to initialize the color models. The grabcut algorithm is based on the iterative reestimation of the GMM color models and relabeling of pixels by minimizing a certain energy function. In our implementation, we use GMM with 3 components and we interrupt the optimization after performing 3 iterations.

### B. Contour processing

A contour of the garment is extracted from the obtained segmentation mask by Moore's algorithm [21] for border tracking. The contour is formed by pixels at the garment's border as seen in Fig. 2b. Thus it can be considered to be a polygon with several hundreds or thousands vertices. Their exact amount depends mainly on size of the garment and on resolution of the input image. However, shape of the contour is much simpler and thus it can be closely approximated by a polygon which has at most tens of vertices, depending on the shape complexity and on the desired precision. Example of a simplified contour can be seen in Fig. 2c.

The simplification applies iteratively the algorithm for optimal approximation of an open curve by a polyline [22]. The algorithm utilizes dynamic programming approach to minimize the overall distance of the original contour points to the edges of the approximating polygon. To find the global optimum, the inner algorithm would have to be run for each point of the original cyclic contour, breaking it to an open curve. However, in practice it is possible to stop it after several iterations to obtain a sufficient approximation.

### C. Polygonal models of clothing

The shapes of possible contours for a particular category of clothing are described by polygonal models. We distinguish the following categories of clothing: towel, pants, short-sleeved shirt and long-sleeved shirt. Fig. 3 visualizes all the models and Fig. 4 shows a more detailed polygonal model for a short-sleeved shirt.

Each polygonal model is determined by its vertices and their relative mutual positions. The mutual positions are described by inner angles adjacent to vertices as well as by relative lengths of polygon edges with respect to its perimeter. The inner angles and relative lengths are learned from training data. They are modeled probabilistically by

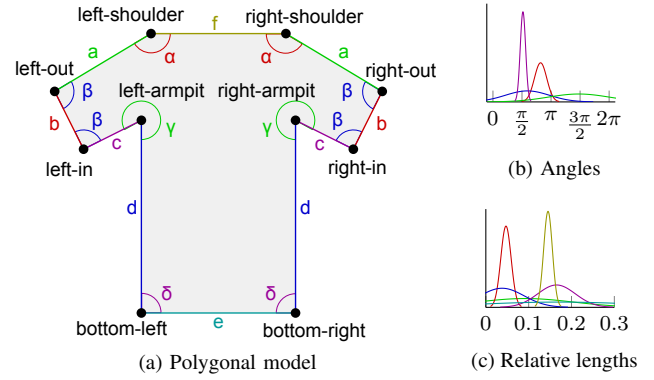


Fig. 4: Polygonal model for a short-sleeved shirt. Inner angles sharing the same distribution are denoted by the same Greek letter  $\alpha \dots \delta$ . Edges sharing the same distribution of relative lengths are denoted by the same Latin letter  $a \dots f$ . Colors of the probability distributions correspond to colors of angles and edges in the polygonal model.

independent normal distributions as seen in Fig. 4b and Fig. 4c. Some distributions are shared by more vertices and edges of the same model, e.g. the distribution of inner angles adjacent to left and right armpit of a shirt or the distribution of relative lengths of towel top and bottom edges. The distributions sharing is allowed by the obvious left-right and top-bottom symmetries of clothing shapes as in Fig. 4.

### D. Matching contours to polygonal models

Now, we have a contour of a garment approximated by a polygon having  $N$  vertices  $p_1 \dots p_N$ , which we will call points from now on. The polygonal model is determined by vertices  $v_1 \dots v_M$ . The number of vertices  $M$  is specific for a particular category of clothing. It always holds  $N > M$ .

We describe how to match points of the polygonal contour to vertices and segments of the polygonal model, i.e. how to find a mapping  $f$  such that  $\forall i \in \{1 \dots N\}$ :

$$f(p_i) = \begin{cases} v_m, & \text{point } p_i \text{ is mapped to vertex } v_m, \\ s, & \text{point } p_i \text{ is aligned to a segment.} \end{cases} \quad (2)$$

The mapping has to satisfy several conditions:

- There exists a point  $p_i$  mapped to it for each vertex  $v_m$ .
- No two points  $p_i$  and  $p_j$  are mapped to the same vertex  $v_m$ . However, many points can be mapped to segments represented by a symbol  $s$ .
- The mapping preserves the ordering of points on the polygonal contour and the ordering of vertices of the polygonal model in the clockwise direction. See Fig. 5.

The number of all possible mappings  $f$  satisfying the conditions can be enumerated easily. To do it, select one of  $N$  points to be mapped to vertex  $v_1$ . Then select an arbitrary subset of  $M - 1$  points from remaining  $N - 1$  points to be mapped to vertices  $v_2 \dots v_M$ . Thus the amount of all possible mappings is:

$$N \binom{N-1}{M-1} \quad (3)$$

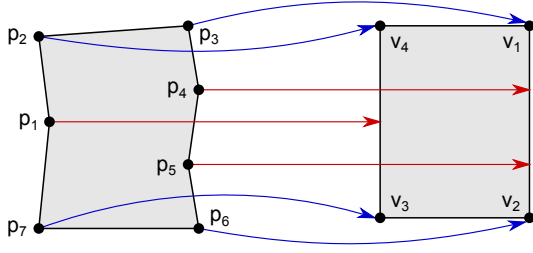


Fig. 5: Visualization of function  $f$  which maps points  $p_1 \dots p_N$  to vertices  $v_1 \dots v_M$  (blue arrows). Some points are mapped to segments (red arrows). The mapping preserves clockwise ordering of both points and vertices.

To compare the quality of mappings, we define a cost function  $C(f)$  associated with a mapping  $f$ . The overall cost is the summation of local costs which express local qualities of a particular mapping. Let assume that indices in all following equations iterate in closed cycles, namely  $i \oplus 1 = (i \bmod N) + 1$ ,  $i \ominus 1 = (i - 2 \bmod N) + 1$ ,  $m \boxplus 1 = (m \bmod M) + 1$  and  $m \boxminus 1 = (m - 2 \bmod M) + 1$ .

*Vertex matching cost*  $V_{i,j,k}^m$  is defined for each triple of contour points  $p_i, p_j, p_k$  and each model vertex  $v_m$ :

$$V_{i,j,k}^m = -\lambda_V \log \mathcal{N}(|\angle p_i p_j p_k|; \mu_m, \sigma_m^2) \quad (4)$$

It expresses how the size of the oriented angle  $|\angle p_i p_j p_k|$  fits the normal distribution  $\mathcal{N}(\cdot; \mu_m, \sigma_m^2)$  of inner angles adjacent to the vertex  $v_m$ . Mean  $\mu_m$  and variance  $\sigma_m^2$  of the distribution are learned from data as in Fig. 4b. Symbol  $\lambda_V$  denotes weight of the vertex matching cost.

*Edge matching cost*  $E_{j,k}^m$  is defined for each pair of points  $p_j, p_k$  and each polygonal model vertex  $v_m$ :

$$E_{j,k}^m = -\lambda_E \log \mathcal{N}\left(\frac{\|\overline{p_j p_k}\|}{\sum_{i=1}^n \|\overline{p_i p_{i \oplus 1}}\|}; \nu_m, \tau_m^2\right) \quad (5)$$

It expresses how the relative length of the line segment  $\overline{p_j p_k}$  (with respect to overall length of the contour) fits the distribution of relative lengths of the model edge  $\overline{v_m v_{m \boxplus 1}}$ . Mean  $\nu_m$  and variance  $\tau_m^2$  of the distribution are learned from data as in Fig. 4c. Symbol  $\lambda_E$  denotes the weight of the edge matching cost.

*Segment matching cost*  $S_{j,k}$  is defined for each pair of simplified contour points  $p_j, p_k$  in following way:

$$S_{j,k} = -\lambda_S \sum_{i \in I_{j,k}} \log \mathcal{N}(|\angle p_{i \ominus 1} p_i p_{i \oplus 1}|; \xi, \phi^2) \quad (6)$$

The range  $I_{j,k}$  passed by index  $i$  is defined as:

$$I_{j,k} = \begin{cases} \{j+1 \dots k-1\}, & j \leq k \\ \{j+1 \dots N, 1 \dots k-1\}, & j > k \end{cases} \quad (7)$$

The segment matching cost expresses the penalty paid for points not matched to any vertex. These points together with neighboring segments should resemble straight lines as seen in Fig. 5. This is why the mean and the variance are set empirically as  $\xi = \pi$  and  $\phi^2 = \pi^2/16$ . Symbol  $\lambda_S$  denotes the weight of the segment matching cost.

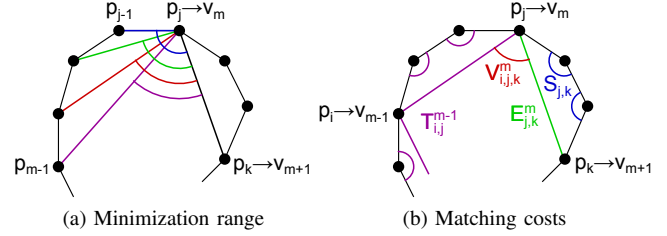


Fig. 6: (a) Minimization of the total cost  $T_{j,k}^m$  goes over all  $p_i$ , where  $i \in \{m-1 \dots j-1\}$ . Various choices of  $p_i$  are visualized in different colors. (b) Total matching cost  $T_{j,k}^m$  is given by summing previous total cost  $T_{i,j}^{m-1}$  (plotted in magenta), vertex cost  $V_{i,j,k}^m$  (red), edge cost  $E_{j,k}^m$  (green) and segment cost  $S_{j,k}$  (blue).

Weights of matching costs were set empirically as  $\lambda_V = 1$ ,  $\lambda_E = 1/3$  and  $\lambda_S = 1$  to balance typical values of the costs. Note that both vertex and segment matching cost evaluate angles and so their weights are equal, whereas the edge matching cost evaluate relative lengths.

All three types of costs are visualized in Fig. 6b by different colors. The overall cost is given by summing costs for all vertices  $v_m$  and points  $p_i, p_j, p_k$  such that  $f(p_i) = v_{m \boxminus 1}$ ,  $f(p_j) = v_m$ ,  $f(p_k) = v_{m \boxplus 1}$ . The goal is to find the mapping having the minimal overall cost  $f^* = \arg \min_f C(f)$ .

#### E. Dynamic programming algorithm for contour matching

The number of all possible mappings (see Eq. 3) is exponential in the number of vertices. It would be very infeasible to compute costs for all such mappings. Instead we propose an efficient dynamic programming algorithm.

The main part of the algorithm is listed in Alg. 1. It assumes that  $f(p_1) = v_1$  and  $f(p_r) = v_m$ , where  $r \in \{M \dots N\}$ . It finds cost of the optimal mapping to the remaining vertices  $v_2 \dots v_{M-1}$ . The optimal mapping itself can be constructed by remembering also index of the point  $p_i$  minimizing the cost  $T_{j,k}^m$ , followed by backward tracing as usual in dynamic programming algorithms. The global optimum can be found by calling Alg. 1 for each combination of  $N$  possible  $n$ -shifts of the contour points  $(p'_1 \dots p'_N) \leftarrow (p_n \dots p_N, p_1 \dots p_{n-1})$  with  $N - M + 1$  options of selecting  $p_r$ , i.e. Alg. 1 is called  $O(N^2)$  times in total.

Alg. 1 is based on an iterative evaluation of cost  $T_{j,k}^m$ , for  $m$  increasing. The cost is a summation of local costs defined in Eq. 4, Eq. 5, Eq. 6 for points  $p_1 \dots p_j$  optimally mapped to vertices  $v_1 \dots v_m$  so that  $f(p_r) = v_M$ ,  $f(p_1) = v_1$ ,  $f(p_j) = v_m$ ,  $f(p_k) = v_{m+1}$ . The main step of the algorithm is the minimization searching for a point  $p_i$  mapped to the previous vertex  $v_{m-1}$ . The minimization is visualized in Fig. 6a. The purpose of individual costs is summarized in Fig. 6b.

The most time complex part of Alg. 1 are three nested loops computing  $O(N^2 M)$  costs  $T_{j,k}^m$ , each of them obtained as minimization over  $O(N)$  elements. Thus the overall complexity of Alg. 1 is  $O(N^3 M)$ . Because Alg. 1 is called  $O(N^2)$  times, the overall time complexity of the proposed contour matching algorithm is  $O(N^5 M)$ . Although that the



**Algorithm 1** Contour matching algorithm

---

**In:**  $r = \text{index s.t. } f(p_r) = v_M$   
 $V_{i,j,k}^m = \text{cost of matching } \angle p_i p_j p_k \text{ to } v_m$   
 $E_{j,k}^m = \text{cost of matching } \overline{p_j p_k} \text{ to } \overline{v_m v_{m+1}}$   
 $S_{j,k} = \text{cost of approximating } p_{j+1} \dots p_{k-1} \text{ by } \overline{p_j p_k}$

**Out:**  $T_{j,k}^m = \text{cost of matching } p_1 \dots p_{k-1} \text{ to } v_1 \dots v_m$   
s.t.  $f(p_j) = v_m, f(p_k) = v_{m+1}$

**for all**  $j \in \{2 \dots r - M + 2\}$  **do**  
  **for all**  $k \in \{j + 1 \dots r - M + 3\}$  **do**  
     $T_{j,k}^2 \leftarrow (V_{r,1,j}^1 + V_{1,j,k}^2) + (E_{r,1}^M + E_{1,j}^1 + E_{j,k}^2)$   
     $+ (S_{r,1} + S_{1,j} + S_{j,k})$   
  **end for**  
**end for**

**for all**  $m \in \{3 \dots M - 1\}$  **do**  
  **for all**  $j \in \{m \dots r - M + m\}$  **do**  
    **for all**  $k \in \{j + 1 \dots r - M + m + 1\}$  **do**  
       $T_{j,k}^m \leftarrow \min_{i \in \{m-1 \dots j-1\}} (T_{i,j}^{m-1} + V_{i,j,k}^m)$   
       $+ E_{j,k}^m + S_{j,k}$   
    **end for**  
  **end for**  
**end for**

**return**  $T_{r,1}^M \leftarrow \min_{i \in \{M-1 \dots r-1\}} (T_{i,r}^{M-1} + V_{i,r,1}^M)$

---

degree of the polynomial is fairly high, the real performance is very good as we show in Sec. V. It is because we choose number of simplified contour points  $N$  between 10 and 20, depending on complexity of the matched model, which is enough for a precise approximation of the original contour.

### F. Generating folded models

The proposed pipeline for landmark points recognition is not used only for a spread garment. After revealing its initial configuration, the garment is folded by the robot and a new image is taken. The contour is extracted and simplified in the same way as already described. Also the matching algorithm is unaltered, however, we have to use a modified polygonal model which reflects the performed fold.

The incremental creation of folded models is shown in Fig. 7. The original vertices are being replaced by vertices denoting endpoints of individual folds. The  $s$ -th fold is performed in the clockwise direction along the contour from position  $F_s$  to  $T_s$ . All the original vertices positioned either between or near  $F_s$  and  $T_s$  are removed and two new vertices  $F_s$  and  $T_s$  connected by an edge are added.

The distributions of inner angles and relative lengths, which are used to evaluate penalty  $V_{i,j,k}^m$  in Eq. 4 and penalty  $E_{j,k}^m$  in Eq. 5, are adjusted to correspond to the observed image and planned fold. The means  $\mu_m$  and  $\nu_m$  for the next folded model are set to the angles and relative lengths measured in the actual image, considering line of the planned fold. The variances  $\sigma_m^2$  and  $\tau_m^2$  adjacent to the original vertices are all set to the smallest variance learned for the

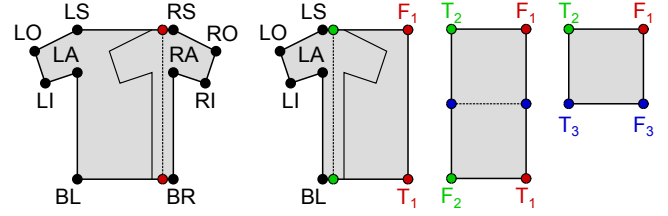


Fig. 7: Incremental creation of folded models for a short-sleeved shirt. The original vertices are being replaced by new vertices denoting endpoints of individual folds (plotted in various colors).

original model, as all following manipulation is performed with that particular piece of garment. The variances adjacent to the newly added vertices are set to twice that value because of the uncertainty in the performed fold.

## IV. PLANNING, GRASPING AND MANIPULATION

ROS provides packages to perform various robotics tasks. We utilize MoveIt package [23] which is included to support the motion planning. It implements interfaces for common robotics libraries. We use two approaches to generate robot trajectories for our purposes – planning and interpolation. The planning uses Open Motion Planning Library (OMPL) [24] to schedule collision free trajectories from one joint state to another one. We have tested several planning algorithms from OMPL and found out that RRT-Connect [25] suits best our needs since it successfully finds a plan in most of the cases and in a reasonable time. The interpolation generates points evenly distributed on a line in Cartesian coordinates. Then it computes the inverse kinematic for each point to produce the final trajectory which is sent to the robot controller.

We adopted the scheme proposed in [26] for the folding. The robot moves the grasped corners along a triangular path and utilizes the gravitational force acting on the garment. Since our gripper is not suitable for grasping flat garment from above, our lower finger slides under the garment and grasps it. The motion near to the garment and with the garment in grippers uses interpolation to have a full control over the actual trajectory and to prevent tearing the garment. The rest of the motion utilizes the planning discussed above.

## V. EXPERIMENTS

We have performed two sets of experiments to test the proposed methods. Sec. V-A shows performance of the computer vision pipeline on a dataset of static images. Sec. V-B describes folding experiments performed on a robot.

### A. Experiments on the dataset of images

We have tested the proposed computer vision pipeline on a dataset of spread garments collected by our team [27]. The dataset contains  $1280 \times 1024$  color images taken from a bird's eye perspective. One pixel roughly corresponds to 0.09 cm in world coordinates. All images were manually annotated by specifying locations of vertices of the described

Error	Towel	Pants	Short-sleeved	Long-sleeved
Median [cm]	0.31	0.46	0.49	0.50
Mean [cm]	0.35	0.52	0.88	1.03
Std. dev. [cm]	0.03	0.11	1.17	2.14

TABLE I: Displacements of the vertices found by the polygonal model matching for various categories of clothing.



Fig. 8: Displacements of the vertices found by model matching (plotted in green) and the manually annotated landmarks (red). The displacements were computed for various configurations of garments and then they were projected to the canonical image.

polygonal models. We used 170 testing images (41 towels, 45 pants, 45 short-sleeved shirts and 39 long-sleeved shirts). The algorithms were implemented in MATLAB and C++. The performance was evaluated on a notebook with Intel M430 2.27 GHz processor and 8 GB memory.

Segmentation was performed on images downsampled to resolution  $320 \times 256$  to achieve a better time performance. Totally 165 of 170 input images were correctly segmented. The incorrectly segmented images were excluded from the further evaluation. The time spent by segmenting one image is on average 0.83 seconds. The contour simplification algorithm is the most time consuming operation which takes between 0.5 and 3.5 seconds, depending mainly on the contour complexity. The subsequent model matching procedure works with the already simplified contour. Its runtime is 0.14 seconds on average. The complete computer vision pipeline runs almost every time under 5 seconds which is a significant improvement compared to 30–150 seconds of Miller et al. [2]

Tab. I summarizes displacements of vertices found by the proposed algorithm compared to the manual annotations. They are similar to those achieved by Miller et al. [2] on their own dataset. Moreover, the reported displacements are approximately 20% lower than displacements achieved by our former algorithm [14] on a subset of the current dataset. Fig. 8 visualizes displacements for various configurations of the selected pieces of garments. The experiments showed that determining exact locations of shoulders is the most problematic task for our model. They are sometimes confused with the neckline. However, since positions of shoulders are not used for the automated folding, these errors cause no problems.

### B. Experiments on the CloPeMa testbed

We performed several experiments on CloPeMa testbed to test the proposed algorithms. A piece of garment was spread manually on the table next to the robot in each

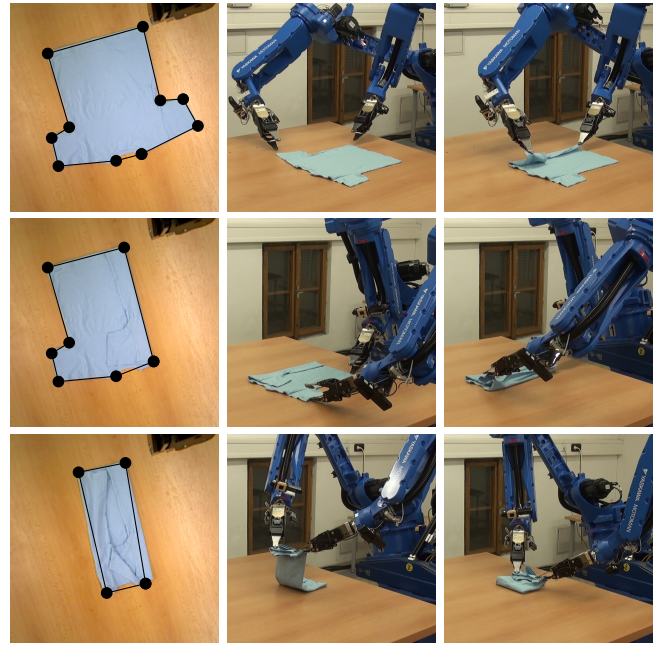


Fig. 9: Robot performs series of folds with a short-sleeved shirt. Images shown in the left column were taken by the robot's camera in order to fit the plotted polygonal model.

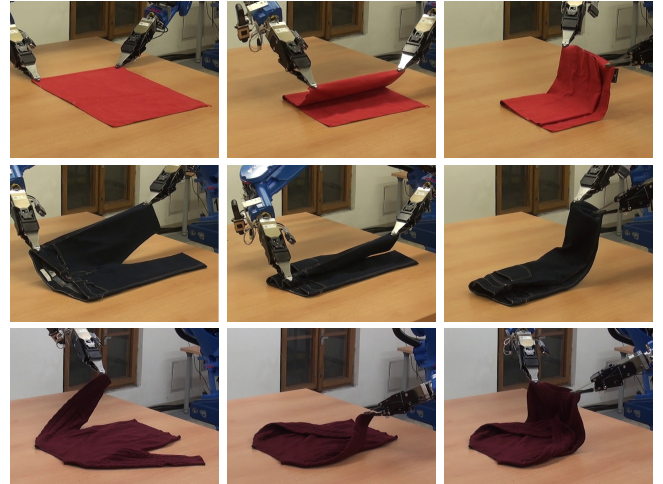


Fig. 10: Detailed view of the robot successfully folding a red towel, black jeans and a violet long-sleeved sweater.

experiment. ROS application following predefined folding steps was started next. In each step, the robot was moved into the start configuration and the scene was perceived using the Asus Xtion RGB camera. The captured image was sent to the vision pipeline to find the polygonal model of the garment. The matched model was used to determine the position of the next fold, i.e. grasping points and points where the garment had to be placed. Several gripper approach directions were generated for each grasp or each place point to increase the chance of a successful planning. If the planning was successful then the resulting trajectory was executed by the robot. Stages of the folding process are captured in Fig. 9. Fig. 10 gives more detailed examples.

The whole process takes up to 4 minutes from which roughly 10 seconds is the actual computation and the rest is manipulation, as measured for a short-sleeved shirt which requires performing three folding moves. The folding procedure succeeds approximately in 70% attempts. However, the possible failures are seldom caused by the proposed vision sensing. We have observed another failure factors instead. The first and most frequent failure is an unsuccessful inverse kinematic computation or motion planning either due to the manipulation range limitation or due to the possible collision. We plan to solve the limited manipulation range problem by pulling the garment to a suitable position prior to folding. The collision avoidance is a very challenging task as we perform very complex moves with two independent arms. We hope that it can be solved by involving more advanced planning algorithms. The second common factor is that the garment moves while folding. The position of corners to grasp changes during the manipulation, resulting in an incorrectly placed fold. In order to prevent a garment from moving, the table surface was roughen.

## VI. CONCLUSION AND FUTURE WORK

We have proposed, implemented and tested the method for folding a piece of garment by a two armed robot which benefits from a new polygonal model extending significantly our previous work [14]. Additional constraints on the relative size of segments led to a more accurate detection of garment landmarks. The matching algorithm is significantly faster than [2]. We have also shown how to automatically derive models of folded garments. This approach has proved to be very useful in the considered scenario.

In future, we would like to be able to detect folding failures in order to recover from them. We also work on more complicated scenarios dealing with garment which is not spread on the table. We would like to develop advanced models of clothing considering also its physical properties. Moreover, we would like to be able to detect and model special parts of clothing like buttons, pockets or collars which provide additional information about the garment configuration.

## ACKNOWLEDGMENT

The reported research has been financed by the European Commission under the project FP7-ICT-288553 CloPeMa (J. Stria, L. Wagner, V. Petrík, V. Smutný), by the Grant Agency of the Czech Technical University in Prague under the project SGS13/205/OHK3/3T/13 (J. Stria, D. Průša) and by the Technology Agency of the Czech Republic under the project TE01020197 Center Applied Cybernetics (V. Hlaváč, P. Krsek).

## REFERENCES

- [1] "CloPeMa (Clothes Perception and Manipulation) research project," <http://www.clopema.eu>.
- [2] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Parametrized shape models for clothing," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011, pp. 4861–4868.
- [3] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The Int. Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.
- [4] P. C. Wang, S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Perception for the manipulation of socks," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2011, pp. 4877–4884.
- [5] J. Maitin-Shepard, M. Cusumano-Towner, J. Lei, and P. Abbeel, "Cloth grasp point detection based on multiple-view geometric cues with application to robotic towel folding," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2010, pp. 2308–2315.
- [6] S. Hata, T. Hiroyasu, J. Hayashi, H. Hojoh, and T. Hamada, "Robot system for cloth handling," in *Proc. Annual Conf. of IEEE Industrial Electronics Society (IECON)*, 2008, pp. 3449–3454.
- [7] A. Ramisa, G. Alenyà, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2012, pp. 1703–1708.
- [8] —, "FINDDD: A fast 3d descriptor to characterize textiles for robot manipulation," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2013, pp. 824–830.
- [9] Y. Kita and N. Kita, "A model-driven method of estimating the state of clothes for manipulating it," in *Proc. IEEE Workshop on Applications of Computer Vision (WACV)*, 2002, pp. 63–69.
- [10] 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.
- [11] Y. Kita, E. S. Neo, T. Ueshiba, and N. Kita, "Clothes handling using visual recognition in cooperation with actions," in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems (IROS)*, 2010, pp. 2710–2715.
- [12] A. Doumanoglou, A. Kargakos, and T.-K. K. 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.
- [13] B. Willimon, I. Walker, and S. Birchfield, "3D non-rigid deformable surface estimation without feature correspondence," in *Proc. IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2013, pp. 646–651.
- [14] J. Stria, D. Průša, and V. Hlaváč, "Polygonal models for clothing," in *Proc. Towards Autonomous Robotic Systems (TAROS)*, 2014.
- [15] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: An open-source robot operating system," in *Proc. ICRA Workshop on Open Source Software*, 2009.
- [16] G. Aragon-Camarasa, S. Oehler, W. P. Cockshott, and J. P. Siebert, "A binocular robot head for clothes perception and manipulation tasks," SICSA DEMOfest 12, University of Edinburgh, Tech. Rep., 2012.
- [17] T.-H.-L. Le, M. Jilich, A. Landini, M. Zoppi, D. Zlatanov, and R. Molfino, "On the development of a specialized flexible gripper for garment handling," *Journal of Automation and Control Engineering*, vol. 1, no. 2, pp. 255–259, 2013.
- [18] M. Orchard and C. Bouman, "Color quantization of images," *IEEE Trans. on Signal Processing*, vol. 39, no. 12, pp. 2677–2690, 1991.
- [19] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley, 2000.
- [20] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut – interactive foreground extraction using iterated graph cuts," *ACM Trans. on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.
- [21] R. C. Gonzalez, R. E. Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*, 2nd ed. Gatesmark, 2009.
- [22] J.-C. Perez and E. Vidal, "Optimum polygonal approximation of digitized curves," *Pattern Recognition Letters*, vol. 15, no. 8, pp. 743–750, 1994.
- [23] "MoveIt package for ROS," <http://moveit.ros.org>.
- [24] I. A. Şucan, M. Moll, and L. E. Kavraki, "The open motion planning library," *IEEE Robotics & Automation Magazine*, vol. 19, no. 4, pp. 72–82, 2012.
- [25] J. J. Kuffner and S. M. LaValle, "RRT-connect: An efficient approach to single-query path planning," in *Proc. Int. Conf. on Robotics and Automation (ICRA)*, 2000, pp. 995–1001.
- [26] J. van den Berg, S. Miller, K. Y. Goldberg, and P. Abbeel, "Gravity-based robotic cloth folding," in *Proc. Int. Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2011, pp. 409–424.
- [27] L. Wagner, D. Krejčová, and V. Smutný, "CTU color and depth image dataset of spread garments," Center for Machine Perception, Czech Technical University, Tech. Rep. CTU-CMP-2013-25, 2013.