

Model for Unfolding Laundry using Interactive Perception

Bryan Willimon, Stan Birchfield, and Ian Walker
Department of Electrical and Computer Engineering
Clemson University, Clemson, SC 29634

{rwillim, stb}@clemson.edu, ianw@ces.clemson.edu

Abstract—We present an algorithm for automatically unfolding a piece of clothing. A piece of laundry is pulled in different directions at various points of the cloth in order to flatten the laundry. The features of the cloth are extracted and calculated to determine a valid location and orientation in which to interact with it. The features include the peak region, corner locations, and continuity / discontinuity of the cloth. In this paper we present a two-stage algorithm, introducing a novel solution to the unfolding / flattening problem using interactive perception. Simulations using 3D simulation software, and experiments with robot hardware demonstrate the ability of the algorithm to flatten pieces of laundry using different starting configurations. These results show that, at most, the algorithm flattens out a piece of cloth from 11.1% to 95.6% of the canonical configuration.

I. INTRODUCTION

Laundry is a common household chore that is a difficult problem to automate. The process of “doing the laundry” consists of several steps: handling, washing, drying, separating/isolating, classifying, unfolding/flattening, folding, and putting it away into a predetermined drawer or storage unit. Figure 1 gives a high level flow chart on these various steps. Several researchers have worked in the past on handling [1] [2] [3] [4] [5] [6] [7] [8], separating/isolating [8] [9] [10] [11] [12], and classifying [6] [9] [11] [12] [13] clothes.

Folding, in particular, has received significant attention as of late [14] [15] [16]. Various researchers are developing folding algorithms and machines at the lower end of the laundry process (i.e. folding clothing). Keio University’s “Foldy” mobile robot has demonstrated the ability to fold a shirt based on high-level user input.¹ Similar research at other universities aimed at folding manipulation has made progress on folding T-shirts, towels [14], pants / shirts [16], and origami [17]. Others have developed cardboard machines to fold T-shirts.² These existing systems are invariably dependent on a starting point of having the items identified and laid out flat in a standard configuration prior to manipulation. Cusano-Towner et al. [18] were aimed at solving the unfolding/flattening problem, that is, to flatten a piece of crumpled clothing by implementing a disambiguation phase and a reconfiguration phase. In this paper we present algorithms, simulations, and experiments with robot hardware, introducing a novel solution to the unfolding/flattening problem. In this paper, we have designed a model/algorithm to unfold/flatten clothes after they have already been labeled as a shirt, shorts, etc. through isolation and classification as

¹http://inventorspot.com/articles/laundryfolding_robot_learns_job_34327

²http://www.metacafe.com/watch/1165247/clothes_folding_machine

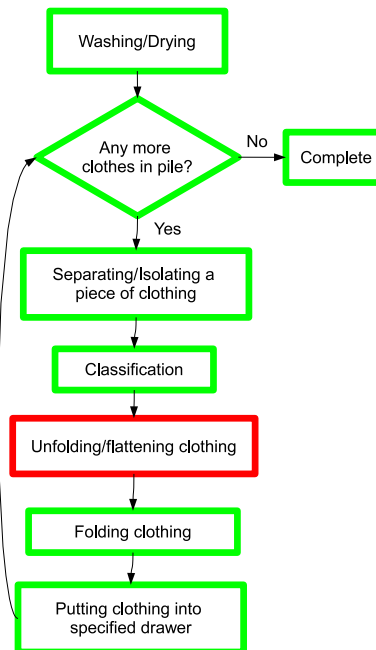


Fig. 1. Overview of the laundry process, adapted from [10]. GREEN areas represent parts of the process that have already been explored in previous work, while the RED area represents the part of the process that is the focus of this paper.

in [9]. Figure 2 shows the robot system that was used in testing our approach.

II. APPROACH

A. Overview

The overall goal of this paper is to define a model / algorithm that describes how to unfold laundry into a flat canonical position. The algorithm of the cloth model contains two phases. In the first phase, initial wrinkles and/or folds are removed without using any depth information (and hence can be accomplished with a single overhead camera). The second phase implements the proposed model, explained below, to remove more difficult folds using depth information. Each component of the model will be explained in further detail below. We use the corners of the article as initial locations for possible grasp points. Corners are used based on an assumption that if the corners are lying flat on the table in opposite orientation and/or are evenly spaced apart, then the article is closer to the canonical (desired) position. The



Fig. 2. The robot system used for flattening a piece of clothing, in this case a washcloth: One PUMA manipulator and one Logitech Quickcam 4000.

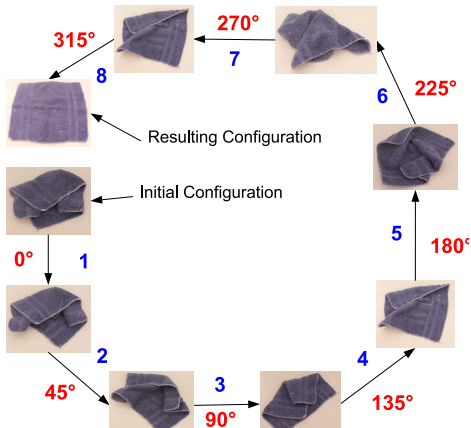


Fig. 3. Process of the first phase to unfolding / flattening laundry. Each step of the process is numbered along with the orientation that is used to transform one configuration into another. In each step the outer edge of the piece of clothing is grasped and pulled away from the center of the object. This is an illustrative example only.

idea behind obtaining a goal orientation/form is to remove all peaks (i.e. topologically high areas) and decrease the vertical size iteratively into a uniform layout. Several factors are considered to characterize a method in flattening an article of clothing. In this paper, we use peak ridges, continuity of a surface, and corner locations.

B. First Phase

The purpose of the first phase of flattening a piece of clothing is to remove any minor wrinkles and/or folds. This phase provides a better configuration for the second phase to start with instead of the initial configuration. In the first phase, the robot moves around the cloth counter-clockwise, pulling at individual corners every d degrees (we set d to be 45). The cloth is grasped at the edge of the clothing (determined by background subtraction) and pulled away from the centroid. Figure 3 illustrates the process of the first phase, which consists of the first eight steps of the algorithm.

C. Second Phase

The second phase uses depth information to locate possible folds in the cloth, and to find grasp points and directions to enable the folds to be removed. Each iteration of this phase involves six steps.

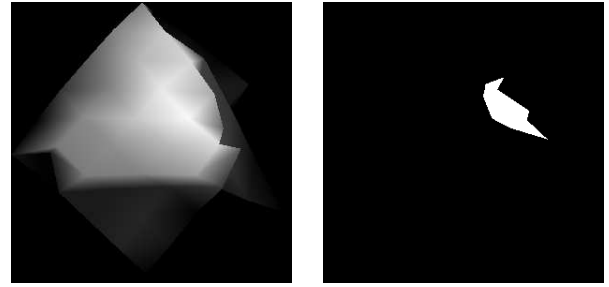


Fig. 4. Depth image (left) and peak ridge (right) of an unfolded washcloth. In the depth image, brighter points are closer to the sensor (higher above the table). The peak ridge contains points within 10%, and contiguous with, the peak.

1) *Peak Ridge*: The peak ridge is a binary map computed from the depth image as follows:

$$P_R(x, y) = \begin{cases} 1, & \text{if } E(x, y) \geq 0.9 \max_{x', y'} E(x', y') \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where $E(x, y)$ is the value of the depth image for pixel (x, y) , and $\max_{x', y'} E(x', y')$ is the maximum value in the depth image, which we call the peak. In the depth image, larger (brighter) points are farther from the table, so the peak is the highest point above the table. This equation locates the area(s) containing pixels whose depth is within 10% of the peak. The area that matches this criterion and also includes the maximum value is the peak ridge. The function computes an (x_C, y_C) point (centroid) and the orientation $\theta_{maj}/\theta_{min}$ (major/minor vectors) of the peak ridge. Figure 4 illustrates the original depth image of the object and the binary mask of the peak ridge of the object.

2) *Corner Locations*: To detect corners along the edge of the cloth, we run the Harris corner detector on the binary image that results from thresholding the depth image so that points on the table are zero while points on the cloth are one. The corner locations are then the locations of these Harris corners, so that $C_L(x, y) = 1$ if a Harris corner was found at location (x, y) , and 0 otherwise. The procedure finds locations of all detected corners and returns the locations in terms of position (x_N, y_N) and orientation θ_N of the corner. Some corners will be located on the peak ridge, while others will be located in the other non-peak regions, see Figure 5 (left).

3) *Discontinuity Function*: Discontinuities in the depth image are stored in a binary array computed as follows:

$$D_C(x, y) = \begin{cases} 1, & \text{if } B_{3 \times 3}(x, y) \cap B_{5 \times 5} = 1 \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

where $B_{3 \times 3}(x, y)$ tests for sharp increases/decreases in values of the depth image, $B_{5 \times 5}(x, y)$ tests for sharp increases/decreases in the slope of the depth image, and \cap is the logical AND operator. More specifically, $B_{3 \times 3}(x, y) = 1$ if $\max(|E(x+1, y) - E(x-1, y)|, |E(x, y+1) - E(x, y-1)|) > th$, where $th = 5$ is a threshold, and 0 otherwise. $B_{3 \times 3}(x, y)$ looks for large changes in the depth image using

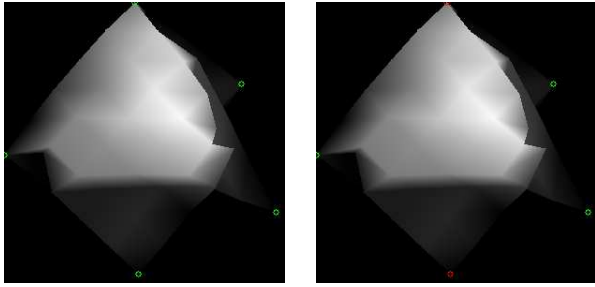


Fig. 5. LEFT: Corner locations of the object, indicated by green circles, found by applying the Harris detector to the binary image that segments the cloth from the background table. In this case there are five corners. Note that, where the cloth is flat, the depth image blends into the background in the figure. RIGHT: The same corners distinguished by whether they are connected with the peak region (green), or outside the peak region (red). In this case three of the five corners are connected with the peak region.

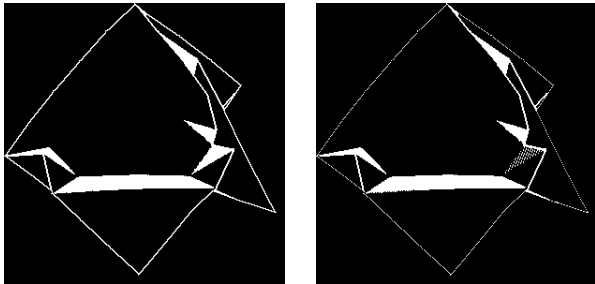


Fig. 6. Locations of discontinuity (white points) on the object before (left) and after (right) removing steep slopes. The left image displays $B_{3 \times 3}$, while the right image displays $D_C(x, y)$.

the 4-neighbors of the pixel in the surrounding 3×3 window. Similarly, $B_{5 \times 5}(x, y) = 1$ if, in either the up/down or left/right direction, the slopes of the depth image along successive columns/rows are either in different directions or vary by an amount more than th . Therefore, $B_{5 \times 5}(x, y)$ looks for large changes in the slope of the depth image using the 5×5 neighborhood of the pixel. Figure 6 illustrates the results of equation (2).

4) *Continuity Check of the Peak Ridge*: The continuity check combines the peak ridge with the discontinuity function. The value $C_P(x, y) = 1$ if (x, y) is contiguous with the peak ridge as determined by the discontinuity function, and 0 otherwise. To compute this function, a floodfill procedure is applied to the peak ridge image, successively incorporating adjacent pixels whose value in the discontinuity function is 0. When a discontinuity pixel is found whose value is 1, it is not included in the region. Figure 7 (right) shows the result of this procedure.

5) *Continuity Check for all Peak Corner Locations*: The continuity check on all peak corner locations determines which corners are connected to the peak region:

$$C_C(x, y) = C_L(x, y) \cap C_P(x, y). \quad (3)$$

This equation returns a subset of $C_L(x, y)$ that contains the locations of corners. The remaining subset, $F(x, y) = C_L(x, y) \cap \overline{C_C(x, y)}$, where the overline indicates binary complement, contains the locations of corners known to be

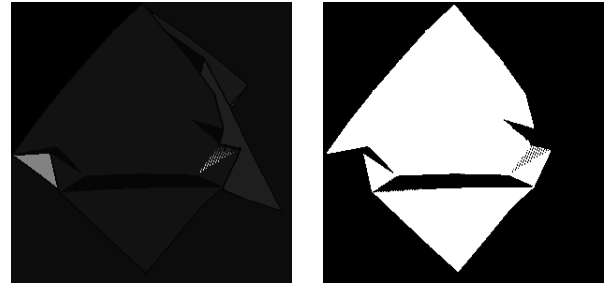


Fig. 7. LEFT: Different regions of the object found by applying connected components after finding discontinuities. RIGHT: The region connected to the peak ridge.

located on a different region than the one containing the peak ridge. Such corners are likely on a different fold of the cloth. Figure 5 (right) shows the former corners (for which $C_C(x, y) = 1$) as red circles, while the latter corners (which are located on a different fold from the peak ridge) are shown as green circles.

6) *Cloth Model Grasp Point and Direction*: The former computations are used to determine a grasp point for the cloth as follows. All of the points for which $C_C(x, y) = 1$ are candidate grasp points, and the final grasp point is selected arbitrarily from among these candidates. For the direction in which to pull the cloth, there are two possibilities. First, if $|\{(x, y) : C_C(x, y) = 1\}| = |\{(x, y) : C_L(x, y) = 1\}|$, that is, all of the detected corners are on the peak region surface, then the cloth is pulled away from the centroid of the cloth. Otherwise, the cloth is pulled toward the centroid in an attempt to unfold the fold. The entire six-step procedure is repeated until the cloth does not change shape.

III. EXPERIMENTAL RESULTS

The proposed approach was applied to a variety of initial configurations of cloths to test its ability to perform under various scenarios using Houdini 3D simulation software³. In each experiment, different initial configurations were used. Figure 11 illustrates the various configurations used during our experiments. We tested our approach on a single washcloth to demonstrate the process of our algorithm on a piece of laundry.

A. Differences Between Pulling in Different Directions

Figure 8 illustrates the initial cloth configuration, I_C , along with the eight different configurations that result from pulling the cloth from a single (x_G, y_G) coordinate in eight different orientations, θ_G . The eight different orientations proceed from 0 to 315 degrees in a counter-clockwise direction at 45-degree intervals, i.e., the sequence of angles is 0, 45, 90, 135, 180, 225, 270, 315, in degrees. We adopt the convention that 0 degrees points toward the bottom of the image.

Figure 9 displays the difference values from the initial configuration to the eight different configurations, in terms

³<http://www.sidefx.com/>

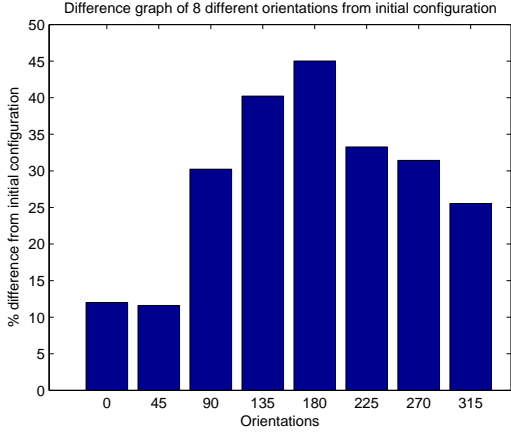


Fig. 9. Difference graph of the eight different configurations of Figure 8 to the initial configuration in the same figure, measured as a difference between binary images (with 1 indicating foreground and 0 indicating background).

of pixel values. The lower the difference value, the more in common the two configurations share in terms of shape. As can be seen from Figure 9, the 4th and 5th configurations are significantly different from the initial configuration, in terms of shape. This plot illustrates how much the cloth configurations can change from pulling on a single point.

In Figure 9, the reason the low and high orientations are correlated to the initial configuration is because those orientations are pulling the point away from the centroid of the cloth. The middle orientations (i.e., 135 and 180) have a very different shape due to the fact that they are pulling the point over the centroid and therefore completely change the configuration / topology of the cloth.

B. Experimental Test of Algorithm

This experiment tested the first phase of the proposed algorithm and monitored the process from eight iterations of pulling the cloth from point (x_G, y_G) in orientation θ_G starting with the same initial configuration, I_C , as in section III-A. Figure 10 illustrates the configurations throughout the entire process. As can be seen from Figure 10, the models continually change the configuration in a manner that flattens and unfolds larger areas of the cloth as the iterations increase. Eventually, the cloth is mostly flattened into a more recognizable shape in the final iteration. The following equation describes how the percentage of flatness is calculated:

$$PC_{Flat} = \frac{\sum_{(x,y)} E(x,y) < \mu}{\sum_{(x,y)} E(x,y)}, \quad (4)$$

where $E(x,y)$ is the value of the depth image, and $\mu = 20$ is a threshold indicating the maximum depth image value for which the cloth is considered to be lying flat on the table. The overall goal of the next step in the laundry process is to increase the flatness toward 100%.

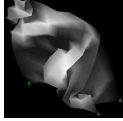

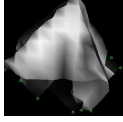
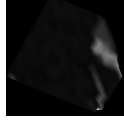
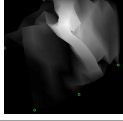
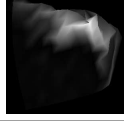
Cloth	Configuration		$PC_{Flat}(\%)$	
	Initial	Final	Initial	Final
Dropped			4.8	18.1
Folded			11.1	83.5
Placed			30.6	56.2

Fig. 11. Various initial and final configurations along with the percentage of cloth that is unfolded/flattened using the first phase only of the proposed algorithm. The initial configuration was obtained by dropping the cloth (top), folding the cloth across a table edge (middle), and placing the cloth by lowering it to the table (bottom). In all cases the algorithm increases the flattening percentage of the object. The final configurations are a result of the first phase only.

C. Taxonomy of Possible Starting Configurations

Figure 11 displays the initial and final configurations of three different starting configurations after the eight steps of the first phase of the proposed algorithm. The dropped cloth was created by dropping the cloth onto the table from a predefined height, the folded cloth was created by sliding the article across the corner of the table and allowing it to fold on top of itself, and the placed cloth was slowly placed on the table from the same position as the dropped cloth. For the most part, all of the final configurations contain a large amount of the cloth to be unfolded and/or flattened. Figure 11 also displays the percentage of cloth that is unfolded/flattened in the initial and final configurations. As observed in Figure 11, the difference in percentage between the initial and final position is always increasing.

D. Test to Fully Flatten the Cloth

This experiment tested the proposed algorithm in determining if this approach would completely flatten a piece of clothing. The test used the first and second phase of the algorithm to grasp the cloth at various locations, (x_G, y_G) , and move the cloth at various orientations, θ_G , until the cloth obtained a flattened percentage greater than 95%. Figure 12 illustrates the configurations at selected iterations of the entire algorithm. The percentages of flatness range from 0.01% \rightarrow 95.6%. Figure 13 shows the percentage of flatness against all iterations of the algorithm.

E. Experiment using PUMA 500

The goal of this experiment is to test the performance of our algorithm in a real world environment using a PUMA 500 manipulator. Figure 14 displays the results of using the peak region on an actual cloth to determine which corner position (x_G, y_G) to select and in which orientation θ_G to pull the object. We used a Logitech QuickCam 4000 for

Depth Image									
Cloth Model									
Orientation θ_G	I_C	0	45	90	135	180	225	270	315

Fig. 8. From Left to Right: The depth image of the initial configuration, I_C , with the grasp point (x_G, y_G) coordinate marked with a red dot; and the eight different configurations resulting from pulling the cloth from the initial configuration in eight different orientations, θ_G , using the same (x_G, y_G) coordinate. Pulling in the direction opposite the centroid of the object tends to improve flattening (e.g., $\theta_G = 0$).

Depth Image									
Cloth Model									
$PC_{Flat}(\%)$	11.08	17.60	16.30	8.16	33.78	44.12	41.47	68.31	83.51

Fig. 10. Experimental test of algorithm for one initial configuration (folded cloth, see section III-C). From Left to Right: Eight successive configurations resulting from pulling the initial configuration into eight successive orientations, θ_G , from eight different grasp points, (x_G, y_G) , using the first phase of the proposed algorithm. Pulling in directions opposite the centroid causes the object to be nearly flattened.

Iteration	1	7	13	19	25	31	37	43	49
Depth Image									
Cloth Model									
$PC_{Flat}(\%)$	3.21	22.61	45.27	72.32	46.36	4.54	23.93	0.01	95.57

Fig. 12. Flattened cloth test for one initial configuration (folded cloth, see section III-C). From Left to Right: Nine different configurations resulting from pulling the initial cloth 49 times in successive orientations and grasp points, using the second phase of the proposed algorithm. The nine configurations shown were selected by hand to be representative of the 49. The final iteration resulted in flattening the cloth with over 95% of the cloth flattened.

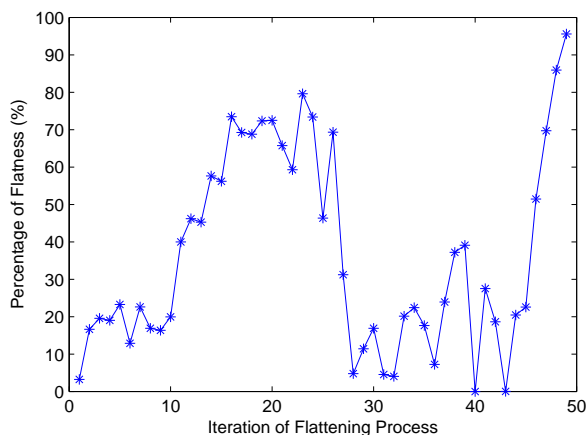


Fig. 13. Plot of the percentage of flatness against all iterations of the algorithm.

an overhead view to capture the configuration of the cloth. After running the image through our approach, the system calculated an output that was transmitted to the robot for extraction. Figure 15 shows an example of the movements of the robot after the location and orientation have been found.

IV. CONCLUSION

Laundry is a common household chore that is a difficult problem to automate. We have proposed an approach to interactive perception in which a piece of laundry is flattened into a canonical position by pulling at various locations of the cloth. The algorithm is shown to provide an initial step in the process of unfolding / flattening a piece of laundry by using features of the cloth. The features used in this paper are a handful of possible cues that could be used in the future to flatten a piece of laundry in fewer iterations. Other features that were considered, but not used, were a prior physical model of the cloth, the relationship between each corner and edge, and the physical features of the texture and material of the cloth.

Though this is a first step, future research in this novel

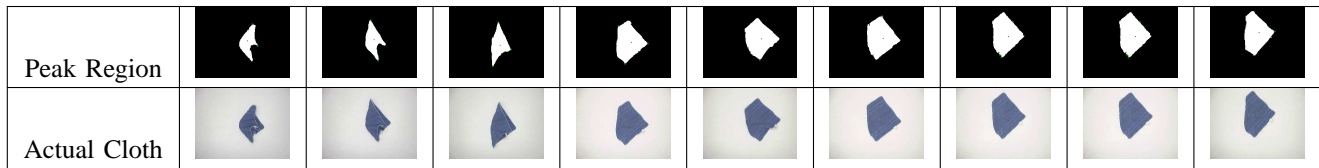


Fig. 14. Actual cloth test for one configuration. From Left to Right: The initial configuration and eight different configurations resulting from pulling the initial cloth into 8 different orientations, θ_G , using 8 different grasp points, (x_G, y_G) successively, using the first phase of the proposed algorithm (since depth information was not available).

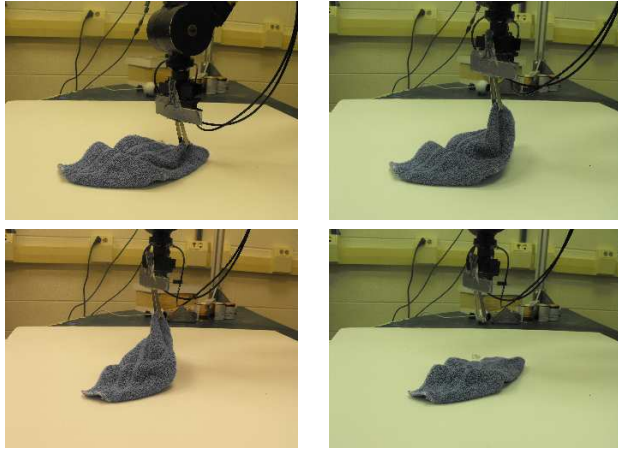


Fig. 15. The various steps of the PUMA manipulator pulling the cloth in a specified orientation, θ_G . From Left to Right, Top to Bottom: the manipulator grasps the cloth on the table, picks the cloth up to a predefined height, pulls the cloth in a precalculated orientation, and drops the cloth back onto the table.

approach of interactive perception would be directed towards using other types of laundry (i.e. shirts, pants, etc). Another direction could be to handle parts of the cloth that are folded inside out, like the arm of a shirt. We believe that these areas are fruitful extensions for future research.

V. ACKNOWLEDGMENTS

We wish to acknowledge Brian Peasley for helpful insight and feedback during our research discussions. This research was supported by the U.S. National Science Foundation under grants IIS-1017007, IIS-0844954, and IIS-0904116.

REFERENCES

- [1] S. Hata, T. Hiroyasu, J. Hayash, H. Hojoh, and T. Hamada, "Flexible handling robot system for cloth," in *International Conf. of Mechatronics and Automation*, 2009.
- [2] Y. Yoshida, J. Hayashi, S. Hata, H. Hojoh, and T. Hamada, "Status estimation of cloth handling robot using force sensor," in *International Symp. on Industrial Electronics (ISIE)*, 2009.
- [3] K. Salleh, H. Seki, Y. Kamiya, and M. Hikizu, "Tracing manipulation in clothes spreading by robot arms," *Journal of Robotics and Mechatronics*, vol. 18, no. 5, 2006.
- [4] Y. Kita and N. Kita, "A model-driven method of estimating the state of clothes for manipulating it," in *Proc. of 6th Workshop on Applications of Computer Vision*, 2002, pp. 63–69.
- [5] Y. Kita, F. Saito, and N. Kita, "A deformable model driven visual method for handling clothes," in *Inter. Conf. on Robotics and Automation*, 2004.
- [6] Y. Kita, T. Ueshiba, E. Neo, and N. Kita, "A method for handling a specific part of clothing by dual arms," in *Conf. on Intelligent Robots and Systems (IROS)*, 2009.

- [7] P. Gibbons, P. Culverhouse, and G. Bugmann, "Visual identification of grasp locations on clothing for a personal robot," in *Conf. Towards Autonomous Robotic Systems (TAROS)*, Aug. 2009, pp. 78–81.
- [8] H. Kobayashi, S. Hata, H. Hojoh, T. Hamada, and H. Kawai, "A study on handling system for cloth using 3-d vision sensor," in *Proceedings of IEEE IECON*, 2008.
- [9] B. Willimon, S. Birchfield, and I. Walker, "Classification of clothing using interactive perception," in *International Conf. on Robotics and Automation (ICRA)*, 2011.
- [10] M. Kaneko and M. Kakikura, "Planning strategy for unfolding task of clothes - isolation of clothes from washed mass -," in *13th Annual Conference of RSJ*, no. 1, 1996, pp. 455–456.
- [11] —, "Planning strategy for putting away laundry -isolating and unfolding task," in *Proc. of the 4th IEEE I.S. Assembly and Task Planning*, 2001, pp. 429–434.
- [12] F. Osawa, H. Seki, and Y. Kamiya, "Unfolding of massive laundry and classification types by dual manipulator," *J of Advanced Computational Intelligence and Intelligent Informatics*, vol. 11, no. 5, 2007.
- [13] Y. Kita, T. Ueshiba, E. Neo, and N. Kita, "Clothes state recognition using 3d observed data," in *Proceedings of the International Conference on Robotics and Automation*, May 2009.
- [14] 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 *International Conference on Robotics and Automation (ICRA)*, 2010.
- [15] F. Osawa, H. Seki, and Y. Kamiya, "Clothes folding task by tool-using robot," *Journal of Robotics and Mechatronics*, vol. 18, no. 5, pp. 618–625, 2006.
- [16] S. Miller, M. Fritz, T. Darrell, and P. Abbeel, "Parametrized shape models for clothing," in *Proceedings of the International Conference on Robotics and Automation*, May 2011.
- [17] D. J. Balkcom and M. T. Mason, "Robotic origami folding," *Inter. J of Robotics Research*, vol. 27, no. 5, pp. 613–627, May 2008.
- [18] M. Cusumano-Towner, A. Singh, S. Miller, J. O'Brien, and P. Abbeel, "Bringing clothing into desired configurations with limited perception," in *International Conf. on Robotics and Automation*, 2011.