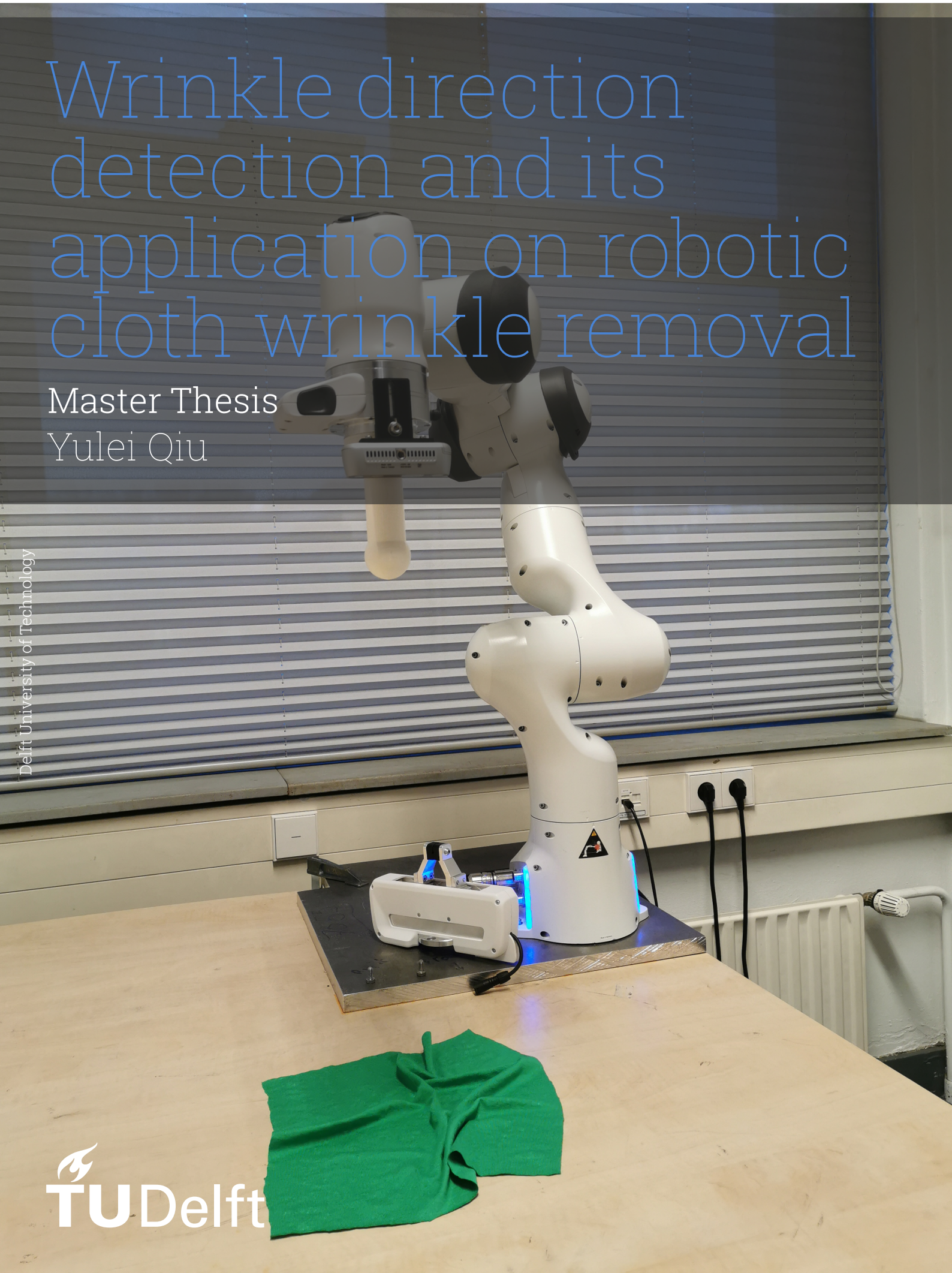
Wrinkle direction detection and its application on robotic cloth wrinkle removal

Master Thesis

Yulei Qiu

Delft University of Technology



TU Delft

Wrinkle direction detection and its application on robotic cloth wrinkle removal

by

Yulei Qiu

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday December 22, 2022 at 9:00 AM.

Student number:	5233178	
Degree:	MSc Robotics	
Thesis committee:	Dr. Jihong Zhu,	University of York, daily supervisor
	Dr. Jens Kober,	TU Delft, supervisor, chair
	Dr. Michael Gienger,	Honda Research Institute Europe, supervisor
	Dr. Michaël Wiertlewski,	TU Delft, external committee member

An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

Acknowledgement

I would like to thank Dr. Jihong Zhu, my daily supervisor for his suggestions and patient guidance during my master thesis. I am also grateful to Dr. Jens Kober and Dr. Michael Gienger, my supervisors at TU Delft and Honda Research Institute Europe, whose regular advice and insightful feedback inspired me a lot. The experience of working with you is very great. Finally, I would like to express my gratitude to my family and friends for their constant support. They are always by my side no matter what happen. That's one of the reasons why I am here.

Yulei Qiu
Delft, December 2022

Wrinkle direction detection and its application on robotic cloth wrinkle removal

Yulei Qiu

***Abstract*— Deformable Object Manipulation (DOM) is an important field of research as it contributes to practical tasks such as cloth handling, cable routing, surgical operation etc. The sensing in DOM is now considered as one of the major challenges in robotics due to the complex dynamics and high degree of freedom of deformable objects. One challenge is to find a suitable representation with low dimensionality and reliable accuracy. The aim of this thesis to develop an algorithm to represent the state of the deformable objects like cloth in low-dimensional vectors, together with a framework based on visual servoing to flatten cloth-like objects. We present a novel pipeline for cloth flattening, which determines a stretching direction (in 2D vector) and an operation point for the robot to removes the wrinkles. The performance of the perception algorithm are validated in simulation and real-world experiment. The whole framework is evaluated in the real-world experiment, which is compared with a human operator. The results show that our framework efficiently determines the direction of wrinkles on the cloth in the simulation as well as the real robot experiment. Besides, the proposed framework has a good performance close to that of a human operator in terms of cloth flattening tasks.**

I. INTRODUCTION

Despite significant progress made in recent years in object manipulation, Deformable Object Manipulation (DOM) is still considered as one of the major challenges in robotics due to the complex dynamics and high degree of freedom of deformable objects. The needs of DOM is also increasing, since many manipulation tasks in our daily life involve deformable objects, such as picking up fruits and folding clothes.

Deformable objects manipulation (DOM) usually breaks down into perception and control. Perception in robotics means to make sense of the unstructured real world. For this specific task, it is to obtain states to describe the object, while control uses this information to guide robot motion. Humans can handle the perception and control in an integrated way. They can therefore efficiently infer the complex configuration of a deformable object, determine an decent policy for manipulation and perform a suitable manipulation. Althouhg there exists end-to-end learning methods such as [1] that model the entire process for a robot, we can still divide the process into two modules and study them respectively.

Among various tasks in DOM, cloth manipulation is one of the most common tasks in our daily life. And clothing is also the most challenging deformable object for perception and manipulation tasks [2]. With the development of computer vision, we are beginning to see some results in both simulations and robot experiments on how to perceive clothes. Early methods, for example, rely on extra markers [3] that requires the cloth to be completely covered to infer the state, or predefined visual features [4], [5] which approximates the cloth with predefined polygon models. The use of markers is usually not possible in practical cases, and predefined geometric features are often not robust and introduce errors in state estimation [6]. Actions, policies or strategies to manipulate are usually learned from human demonstrations since the human behavior during cloth manipulation process is difficult to model. Common used methods in Learning from Demonstration (LfD) include Dynamic Movement Primitives (DMPs) [7] and Gaussian Processes (GPs) [8]. Recent data-driven methods usually combine perception and control stages, which does not infer the explicit states of the cloth and output the policy directly. They learn the mapping from the raw RGB(D) images to the robot actions [1], [9], [10]. Like most learning-based methods, they requires a large number of training data and additional requirements of computational resources.

We would like to develop a sensing algorithm that does not rely on markers or geometry shapes, following by a control module to make use of the sensing information. To this end, we present a pipeline to complete the cloth flattening task. For perception, our method uses a 2D vector to represent the direction and magnitude of wrinkles on the cloth. The directions are then used to determine the stretching direction that removes the wrinkles. The magnitude reflects how "wrinkled" the area is, which helps determine the operation point. For control, the proposed framework determines the stretching direction and operation point based on the magnitude of the wrinkle. An Image-Based Visual Servoing (IBVS) system is designed to control the manipulator using the direction and point.

The main contributions of this thesis are:

1) A framework for cloth flattening tasks combining computer vision and IBVS system.
2) An algorithm adapted from Wrinkle cOntaction Detection (WORD) [11] in cloth flattening tasks that successfully applies on cloth flattening tasks
3) Experiments demonstrating that the proposed vision-based architectures are close to the baseline. We evaluate the performance of this framework and compare it with a human operator that flattens the same cloth. The result shows that our framework can reach a similar level for this task with respect to a human operator.

The whole pipeline is tested using Intel RealSense D435 camera and Franka Emika Robot. We also discuss the shortcomings of the system, which points to interesting areas for future research.

II. Related work

Besides aforementioned perception methods [3]–[5], other work on cloth perception use wrinkle as a feature [12], [13]. It is intuitive that wrinkle is an obvious feature for clothes placed on a flat surface like table. These two methods both compute a heat map from visual information and determine the state of the wrinkles or the grasping policies based on the heat map. Prior to them, Gabor filter is applied on images to extract wrinkle features [14]. The wrinkled area of the cloth can be detected by the Gabor filter with caerful design of the kernel. Inspired by [14], Word cOntraction Detection (WORD) is designed to represent the state of the cloth that infers the main direction of all the wrinkles on the cloth using Gabor filter [11]. We notice that WORD uses only a 2D vector to represent the state of the cloth, which shows potential to be applied to other cloth manipulation tasks. Our algorithm is therefore developed based on WORD.

Robot actions can be either determined from the perception information or learned directly from the raw image input [15], [16]. Learning-based methods that learn the trajectory of robot motion [17]–[19] have a better performance in learning the movement policies. These methods encode skills by extracting trajectory patterns from demonstrations. Visual servoing, a combination of robot vision and robot control, is also a promising method directly making use of perception information. The main advantage of visual servo system is its strong robustness [20] but requires calibration. We would like to make good use of the result from perception to investigate the possibility to generalize WORD. Hence, we choose to design a visual servo system for the cloth flattening task.

III. Methods

In perception part, our method can determine the stretching direction and the operation point of a cloth on the table from the top-down visual observation. The wrinkle direction represents the state of the cloth, indicating toward which direction the robot shall pull. The magnitude shows how wrinkled the area is, which is a criteria when determining the operation point. Next, we determine the operation point based on the magnitude distribution of. Finally, the visual servo system executes the corresponding action with the calculated direction and operation point. The algorithm will work until the cloth is flatten. The main components of our proposed method – visual feature extraction and visual servo system – are described respectively in the two subsections below.

A. Visual Feature Extraction

WORD has been validated in both simulation and real robot experiment by its author [11]. Figure 1 shows the results of WORD in simulation and real world. Although WORD looks promising that it can calculate the main wrinkle direction on the cloth, it has some limitations. For example, it is designed for a cloth placing task, where the cloth is firstly picked at the two corners handing in the air vertical to the table. Therefore, the original WORD can only output direction towards right. Therefore, we develop an variant of WORD for our scenario. The modified WORD has three outputs:

1) Stretching direction. The direction is first calculated in a bit different way from original WORD, see below.
2) Operation point. The algorithm determines a point in the image where the end-effector should move, i.e. the pixel coordinate will be given.
3) Stretching distance. This is the distance the end-effector should move along the given direction.

With these three outputs, the robot should be able to move to complete the cloth flattening task.

Figure 3 shows an overview of the modified WORD. The raw RGB image from the camera is first processed by HSV thresholding to remove the background and leave the cloth part only. Then the image is split into small blocks. Algorithm 1 shows the pre-process steps. After that, WORD is applied on those blocks and now we have one direction and one magnitude for each block. All the magnitude forms a heat map over the image. The block with the highest magnitude should be the block to start with, since it is the most wrinkled part of the cloth. And the magnitude of this block is the highest magnitude among all the blocks. For direction, the output should be either the direction in the block with the highest magnitude or its opposite one, which is determined by considering the position of the current block with respect to the center of mass (CoM) of the cloth. The direction that is pointing outward with respect to the center should be the direction of this block. We dot-product a direction with the vector pointing from CoM to the center of the current block to determine whether this direction is point outward. By doing so, all the vector output by WORD are pointing outwards, which is reasonable for a cloth flattening task. The algorithm of the modified WORD is shown in Algorithm 2.

For simplicity, the operation point in our method is one of the center of the blocks. Although now we know the block with highest magnitude (denoted by *center block*), it is not suitable to make its center as the operation point since we don't want the robot operate right on the wrinkle, which may create more wrinkles. The selection of the operation point follows the steps below:

1) The candidate blocks are the eight blocks around the *center block*.
2) Blocks that are closer to CoM are first sifted out of the candidate blocks. Here we denote the vector pointing

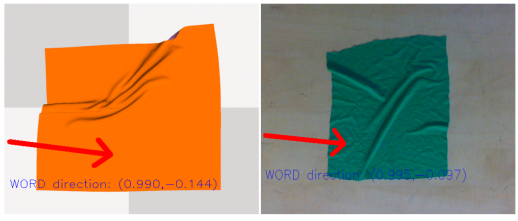


Fig. 1: Original WORD output

Algorithm 1 Pre-process of the image

Input: image $\boldsymbol{I}$
Output: small image list $\boldsymbol{I}_{list}$, the center of mass CoM
Apply HSV thresholding on the image to get an image without background $\boldsymbol{I}_{cloth}$
Calculate the center of mass of the remaining cloth in the image, CoM
Divide the image $\boldsymbol{I}_{cloth}$ into small blocks and store in $\boldsymbol{I}_{list}$
return $\boldsymbol{I}_{list}$, CoM

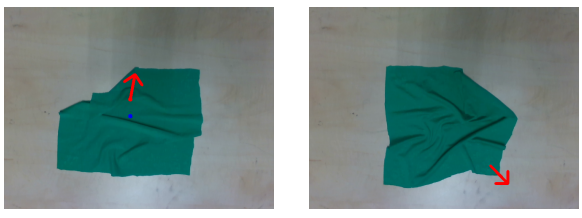
from CoM to *center block* by $\boldsymbol{a}$. The criterion is the dot product between $\boldsymbol{a}$ and the vector pointing from *center block* to the rest candidates. Blocks with negative dot product are sifted out of the candidate blocks. This step filters out the blocks which are on the inner side (with respect to *center block*) to avoid creating more wrinkles when stretching.

3) The chosen block among the rest candidate blocks is the one with lowest magnitude (denoted by *operation block*). The idea is to select a block with less wrinkles from the rest candidate blocks. The position of center of the *operation block* (in pixel) is therefore the operation point.

Figure 2 shows the visualization of the output of the Visual Feature Extraction part. Note that the operation point is the center of the *operation block*. To summarize, the stretching direction $\boldsymbol{w}$ in Algorithm 2 and the operation point are calculated by the perception algorithm discussed above.

B. Visual Servo System

Visual servoing is a well-known approach to guide robots using visual information [21]. Particularly, Image-Based Visual Servoing (IBVS) systems calculate the control law using the visual information directly. In our task, the visual information is the image captured by the camera mounted on the end-effector of the manipulator, and the control law is the information required for robot motion, which is stretching distance, direction and the operation point here. Therefore, we design an IBVS system to make good use of the visual information from perception part. This system solves two problems in the whole pipeline:



(a) (b)

Fig. 2: The red point is the operation point. Note that blue point in the the left figure is the center of the block with the highest magnitude.

Algorithm 2 Modified WORD (mWORD)

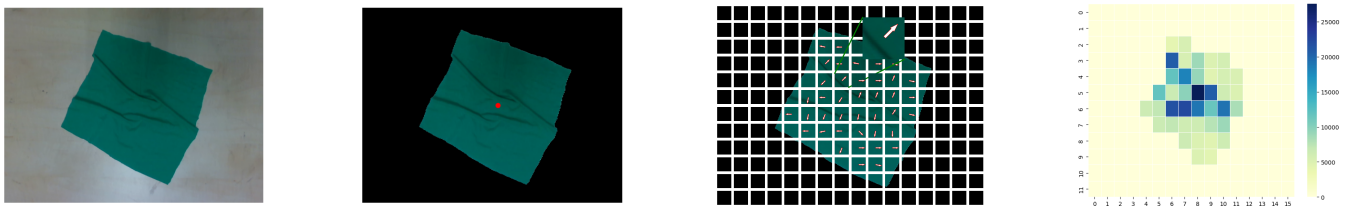
Input: image list $\boldsymbol{I}_{list}$, center of mass CoM, number of orientations n
Output: WORD vector $\boldsymbol{w}$
for $\boldsymbol{I}$ in image $\boldsymbol{I}_{list}$ **do**
for $i = 0, 1, ..., n-1$ **do**
$\theta_i \leftarrow i \times \pi / n$
Get Gabor kernel in direction θ_i
Apply the Gabor kernel on the image $\boldsymbol{I}$ to get wrinkles in direction θ_i
Stack the summation of pixel values in an n-dimensional vector $\boldsymbol{L}$
end for
$m \leftarrow \max_{i \in \{0,1,...,n-1\}} \boldsymbol{L}(i)$
▷ m: the magnitude of the current block
$\boldsymbol{d} \leftarrow \boldsymbol{v}(\theta_i)$, where $i = \arg\max_{i \in \{0,1,...,n-1\}} \boldsymbol{L}(i)$
▷ $\boldsymbol{d}$: the direction of the current block
▷ $\boldsymbol{v}$: unit vector perpendicular to θ_i direction
Calculate the pixel coordinate $\boldsymbol{p}$ of the center of the current block $\boldsymbol{I}$
Calculate the vector pointing from center of mass to the current block, $\boldsymbol{a} \leftarrow (\boldsymbol{p} - CoM)$
if $\boldsymbol{a} \cdot \boldsymbol{d} \geq 0$ **then**
$\boldsymbol{d} \leftarrow \boldsymbol{d}$
else
$\boldsymbol{d} \leftarrow -\boldsymbol{d}$
end if ▷ Make $\boldsymbol{d}$ point outwards the center
Add the direction and magnitude of the current image block to two lists, $\boldsymbol{d}_{list}$ and m_{list}
end for
$\boldsymbol{w} \leftarrow \boldsymbol{d}_{list}(i)$, $i = \arg\max_{\boldsymbol{I} \in \boldsymbol{I}_{list}} m_{list}(\boldsymbol{I})$
return $\boldsymbol{w}$

1) How to calculate the world coordinate given the pixel coordinate?
2) How to execute the robot given the stretching distance, direction and the operation point?

The conversion of image pixel coordinates to world coordinates is a multi-step process, as illustrated in Figure 5. We calculate the transformation from image coordinates to camera coordinates first. We consider a pinhole camera model here since most of commodity cameras are based on this model. As shown in Figure 4, the pinhole camera model gives us the relationship between the location of a pixel in 2D image and the corresponding points in 3D space. The camera coordinates and the pixel coordinates have the following relationship:

$$Z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = KP_c \tag{1}$$

where u and v are the pixel coordinates, f_x and f_y are focal length of the image along pixel width and height, u_0 and v_0 are the principal point coordinates, X_c, Y_c and Z_c are the camera coordinates, and K is called *intrinsic matrix*.



(a) Input image (b) CoM (c) Direction (d) Heat map

Fig. 3: An overview of important steps in the perception stage. Note that the zoomed-in part in Figure 3c indicates the stretching direction of this block. The heatmap in Figure 3d shows the wrinkle distribution in Figure 3c.

The position in world coordinate system can be calculated by the following equation:

$$^{base}P = ^{base}T_{tool} \times ^{tool}T_{cam} \times ^{cam}P \tag{2}$$

where $^{base}T_{tool}$ is the for end-effector (tool) coordinate system relative to world (robot base) coordinate system, $^{tool}T_{cam}$ is the for camera coordinate system relative to end-effector coordinate system, and ^{cam}P is the position in camera coordinate system. Equation 1 gives the camera coordinates, which means now we need to find the transformation from world coordinate system to camera coordinate system. Back to Equation 2, the transformation between world (robot base) coordinate system and end-effector coordinate system can be easily computed by directly reading the end-effector pose from the robot. Therefore, we only need to determine the transformation between camera and end-effector coordinate system, which can be calculated by hand-eye calibration.

As mentioned before, the camera in our IBVS system is mounted on the end-effector, where the configuration of the robot end-effector (hand) and the camera is eye-in-hand. The camera is attached to the moving hand and observing the relative position of the target. Figure 6 shows the matrix transformations among these four coordinate systems in hand-eye calibration. For any two poses of manipulator while keeping the calibration board at the same location, the following relationship holds:

$$^{base}T_{tool2} \times ^{tool2}T_{cam2} \times ^{cam2}T_{cal} = ^{base}T_{tool1} \times ^{tool1}T_{cam1} \times ^{cam1}T_{cal} \tag{3}$$

Move some terms from one side to the other:

$$^{base}T_{tool}{}^{-1} \times ^{base}T_{tool2} \times ^{tool2}T_{cam2} = ^{tool1}T_{cam1} \times ^{cam1}T_{cal} \times ^{cam2}T_{tool}{}^{-1} \tag{4}$$

Finally we have:

$$AX = XB \tag{5}$$

where $A = ^{base}T_{tool}{}^{-1} \times ^{base}T_{tool2}$, $B = ^{cam1}T_{cal} \times ^{cam2}T_{tool}{}^{-1}$ and $X = ^{tool1}T_{cam1} = ^{tool2}T_{cam2}$ since the relative position between robot base and the calibration remains unchanged. $^{tool}T_{cam}$ can therefore be computed by solving Equation 5, which is a typical problem in 3D robot hand/eye calibration. There are already solutions for this problem such as [24], [25].

C. Summary

In this section, we describe our algorithm – modified WORD – that calculates the stretching direction and operation point (in pixel coordinate). The IBVS system can further transform the pixel coordinate to world coordinate, which can be used to guide the robot motion.

IV. Experiment

We first validate our perception method on cloth with wrinkles in an open-source simulation environment, SoftGym, and in the customized cloth flattening environment. Besides the validation in a simulation environment, we also perform robot experiments to test the performance of our proposed methods.

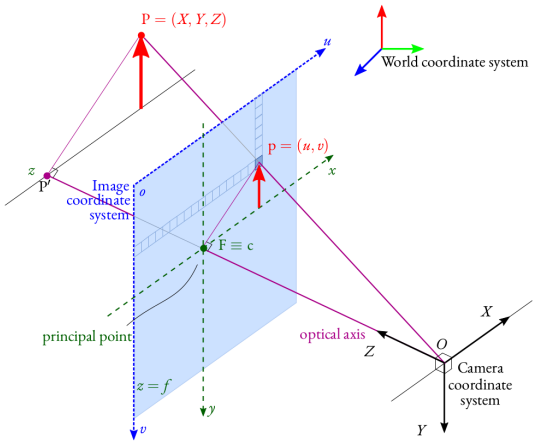


Fig. 4: A pinhole camera model [22].

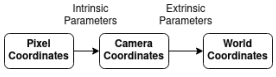


Fig. 5: A schematic view of converting pixel coordinates to world coordinates

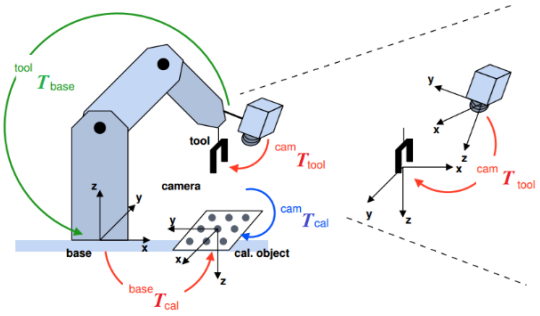


Fig. 6: The transformation relationship between different coordinate systems for eye-in-hand configuration [23].

A. Simulation environment

We setup the simulation environment for cloth flattening task in a a soft-body simulator, Softgym [26]. It is the state-of-the-art (SOTA) particle-based simulator for non-rigid objects like rope, cloth and fluid. Example results are shown in Figure 7. The observation is an RGB image of size of 720×720, taken from a top-down view. We can see from the left part of Figure 7 that the directions are perpendicular to the wrinkles in each block, which is the expected output of our algorithm. The heatmap in Figure 7 indicates higher value for blocks with more wrinkles, which matches the wrinkle distribution on the left.

B. Real Robot Experiment

To make use of the direction and operation point, we design and 3D print an end-effector for this task. Figure 9a shows the self-made end-effector mounted on the end of the manipulator. The end-effector has a finger-like shape, which is a combination of a cylinder and a ball. The ball part will contact the cloth during the manipulation. The other side is a flange connected to the robot. Note that this end-effector also limits the manipulation in a 2D surface.

We evaluate our method on a Franka Emika Robot. An Intel RealSense D435 camera is mounted on the end-effector of the robot looking down to the cloth on the table to get the RGB image inputs from a top-down view. We also design a finger-like end-effector for this task. For extrinsic calibration, we use the chessboard downloaded from OpenCV [27] and Visual Servoing Platform (ViSP) [28] to calculate the transformation matrix ${}^{tool}T_{cam}$. The workflow of the experiment is:

1) Before the start of the experiment, put the cloth under the camera in the middle of its field of view and flatten the cloth manually.
2) Run the script to activate the camera. The camera will take a snapshot of the current cloth and calculate the initial coverage.
3) The algorithm computes the operation point and stretching direction.
4) The robot execute the action by first moving the end-effector to the operation point and pull towards the stretching direction for a certain distance.
5) The robot moves back to the starting position, which completes a step.
6) Return to 2 and execute until the coverage meets the stopping criterion.

Baseline: We compare the performance of our method with a human operator. During the task, the human operator can see the observation from the camera from the screen. They will determine the operation point and stretching direction by clicking twice on the observation window. The position of the first click will be the operation point, and the stretching direction will be the direction from first click to second click (see Figure 2b). To better compare the performance, we fix the moving distance for both method. The human operator is the author himself.

Evaluation metric: We evaluate the performance by the relative coverage of the cloth. For the cloth flattening task, we compute the coverage of the cloth, the number of pixels of cloth divided by the total number of pixels in the image. The relative coverage is computed by

$$rC = \frac{FC}{IC}$$

where FC means final coverage, the coverage at the last step and IC means initial coverage, the coverage at the beginning of the task.

The stopping criterion is either the relative coverage is greater than 99%, or the number of steps is greater than 8.

Tasks: We design two tasks with different difficulty. For easy tasks, we create wrinkles on the cloth so that they can be completely removed by 2D manipulation. This type of tasks is to evaluate whether the direction and operation point given by our algorithm works in a simple case. For hard tasks, one corner of the cloth is folded up before making wrinkles. We would like to see whether our algorithm can perform well under this disturbance.

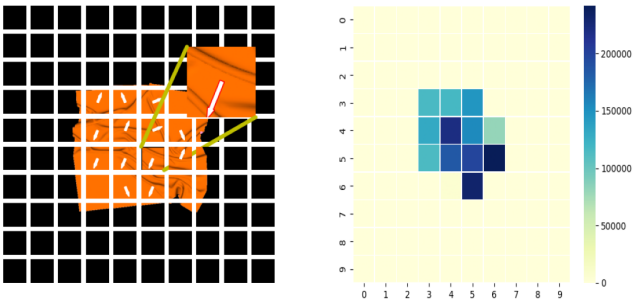


Fig. 7: The output direction and heatmap by applying our algorithm.

TABLE I: Real Robot Experiment Results

Method \ Task	Easy		Hard	
	Step	Final Coverage	Step	Final Coverage
WORD	5.33	0.335		0.286
Human	4.33	0.332		0.297

Results: The quantitative results of the real robot experiments are given in Table I and the visualization of one experiment is provided in Fig 9. Other results are shown in Figure 10 and Figure 11, for easy and hard tasks respectively. Note the the data in Table I is averaged for three experiments.

V. Discussion

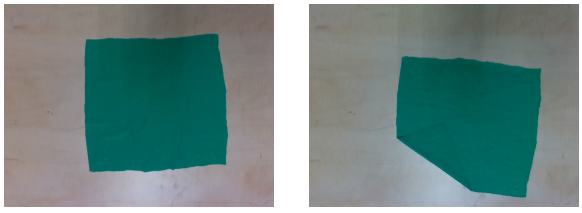
In this study, the aim is to develop an algorithm for cloth flattening tasks. We propose a pipeline containing perception and control parts, where we design the perception algorithm (based on WORD) and the vision-based controller. The perception algorithm is tested in the simulation environment. The output is shown in 7, which gives correct directions perpendicular to wrinkles in most blocks and in the heatmap clearly reflects the distribution of wrinkles. The performance of the perception algorithm in real robot experiment shown in Figure 3c and 3d is also satisfying, indicating both the direction and magnitude.

A. Easy Tasks

To evaluate the performance in manipulation, we conduct experiments with different difficulty to compare the performance of our method and that of a human operator. Both methods can complete this task in no more than 5 steps. Although we can observe that human generally outperforms our algorithm, the direction and operation point given by it can be used as a manipulation policy successfully. Human operator can complete the tasks within 4 steps with their decent policy, while our method tasks up to 5 steps to flatten the cloth. Another point worth noting is that the policy from human operator can always result in an increasing coverage. Our method sometimes gives a sub-optimal policy which slightly decreases the coverage, such as step 2-3 in Figure 10b and 10c. But generally we can see that the modified WORD indicates a correct direction perpendicular to the biggest wrinkle on the cloth and a proper operation point that avoids operation on other wrinkles, and that it finally leads to a successful completion of the task. Besides, the final coverage of the two methods are very close, only with a difference of 0.9%.

B. Hard Tasks

Since it is impossible for both methods to totally flatten the cloth in hard tasks, the number of steps is not compared here. Figure 11 and 12 shows again that both methods perform well in cloth flattening tasks. Figure 11 shows that the human operator ends the manipulation earlier, since he can observe the convergence of the coverage. Besides, human operator always flatten the cloth more "quickly" than our proposed algorithm. Human operator usually reaches a relative coverage over 0.7 at step 2 or step 3 because human can make correct decisions in cloth flattening task in most cases. Our proposed method focuses on removing big wrinkles in the first few stages, which results in a slower convergence in coverage. This can also be seen in Figure 12 that the first few operations point are selected around the most wrinkled area due to our rules. The selected operation points by the human operator at the beginning of the task are always at the corners no matter how the wrinkles are distributed. Although the performance of our algorithm cannot exceed the human operator for hard tasks, the proposed method shows a good performance in flattening the cloth in finite steps, the relative coverage of which can reach up to 0.9 and the average relative coverage is only 3.7% less than that of human operator.



(a) (b)

Fig. 8: Initial configuration of the cloth for easy and hard task.

C. Limitations

The first limitation is that the perception algorithm is not so robust. WORD relies heavily on Gabor filter to identify the wrinkles. The output of Gabor filter is easily influenced by the environment such as illumination and the material (color) of the cloth. Although this limitation can be alleviated by fine-tuned the hyperparameters for new environments and new clothes, it is not a complete solution to this problem.

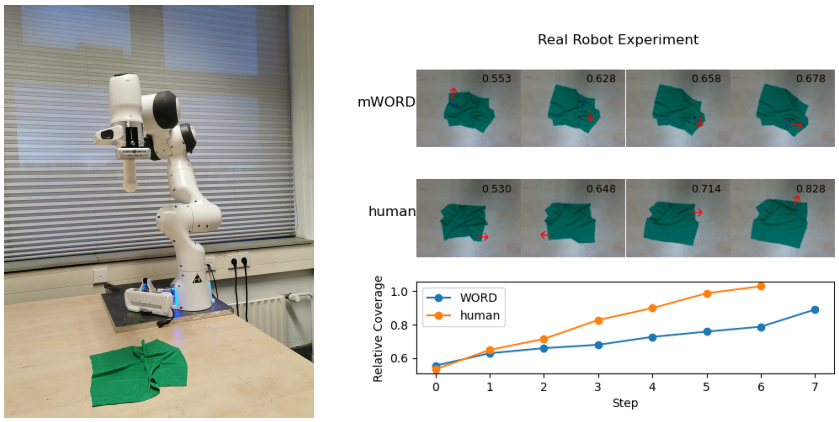
Another limitation is that our method is not "flexible". Compared with human operators, the strategy of using magnitude to determine policy is a bit far from the optimal one. Besides the current method, it is also promising to use learning-based methods for the policy. We can either learn from human operator by providing demonstrations, or the visual feedback. Other works such as [1] combine visual feedback and learning method for deformable object manipulation, which makes good use of the perception information.

D. Future work

In the future, we plan to use learning-based method in the control module. The fact that human operator outperforms the current methods means that a better-perform model can be learned from human demonstration. We also want to try to implement other learning frameworks utilizing the visual feedback given the perception module. The current method will be compared as the baseline method. We expect the learning-based method to outperform the current method and get closer to human operator. Besides, it is possible to combine the current method with other cloth representation methods like [6] create a more generalized and robust framework for more types of DOM.

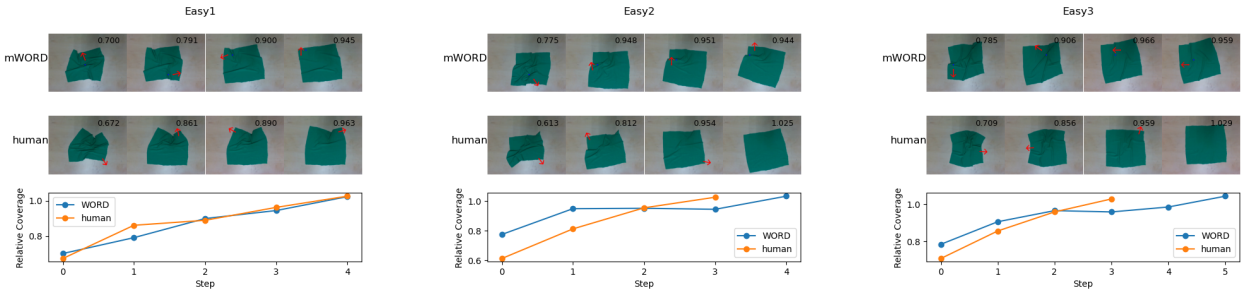
VI. CONCLUSIONS

This thesis presents promising pipeline for cloth flattening task. We have shown that our perception algorithm can successfully calculate a stretching direction of the most wrinkle area, and and a corresponding operation point on



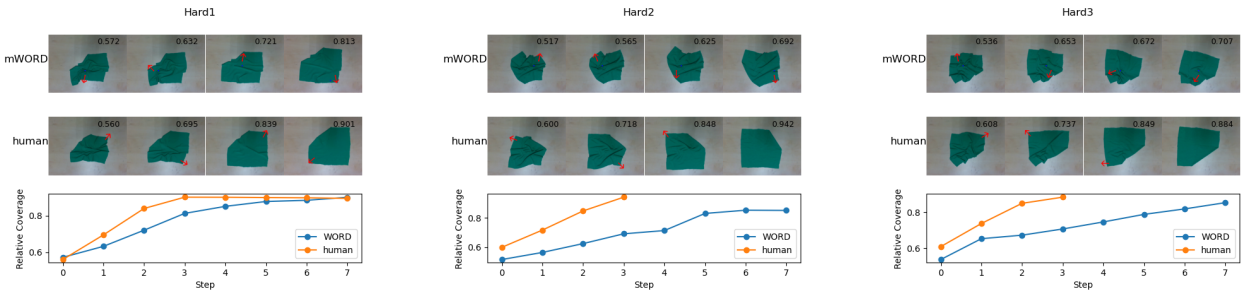
(a) (b)

Fig. 9: Robot experiment on cloth flattening. (a) We use a Franka Panda robot with the end-effector for manipulation and an Intel RealSense D435 camera with a top-down view for sensing. (b) We show the execution of first four steps for our algorithm and the human operator, and the change of the coverage with respect step.



(a) (b) (c)

Fig. 10: Experiments for easy task



(a) (b) (c)

Fig. 11: Experiments for hard task

the cloth. We have also presented the setup of an IBVS system that can execute the policy using the information from the perception part. Validation experiments in SoftGym have shown that our algorithm works for simulated cloth. The experiments on real robots have shown that the pipeline performs well in removing wrinkles of crumpled cloth, with a little gap between the performance of human operator. With the validation in simulation and real robot experiments, we have provided a novel pipeline for cloth flattening tasks including perception and control module.

Fig. 12: All steps of WORD in hard task.

REFERENCES

[1] Y. Wu, W. Yan, T. Kurutach, L. Pinto, and P. Abbeel, "Learning to manipulate deformable objects without demonstrations," 2019.

[2] L. Sun, G. A. Camarasa, A. Khan, S. Rogers, and P. Siebert, "A precise method for cloth configuration parsing applied to single-arm flattening," *International Journal of Advanced Robotic Systems*, vol. 13, no. 2, p. 70, mar 2016.

[3] C. Bersch, B. Pitzer, and S. Kammel, "Bimanual robotic cloth manipulation for laundry folding," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2011, pp. 1413–1419.

[4] S. Miller, J. van den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, dec 2011.

[5] J. Stria, D. Prusa, V. Hlavac, L. Wagner, V. Petrik, P. Krsek, and V. Smutny, "Garment perception and its folding using a dual-arm robot," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, sep 2014, pp. 61–67.

[6] X. Ma, D. Hsu, and W. S. Lee, "Learning latent graph dynamics for visual manipulation of deformable objects," 2021.

[7] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," 2021.

[8] C. E. Rasmussen, "Gaussian processes in machine learning," in *Advanced Lectures on Machine Learning*. Springer Berlin Heidelberg, 2004, pp. 63–71.

[9] R. Jangir, G. Alenya, and C. Torras, "Dynamic cloth manipulation with deep reinforcement learning," 2019.

[10] Y. Tsurumine, Y. Cui, E. Uchibe, and T. Matsubara, "Deep reinforcement learning with smooth policy update: Application to robotic cloth manipulation," *Robotics and Autonomous Systems*, vol. 112, pp. 72–83, feb 2019.

[11] C.-Y. Tsai, "Wrinkle contraction direction: a usefulfeature for learning robotic fabricmanipulation from demonstration," mathesis, Delft University of Technology, Oct. 2021.

[12] A. Ramisa, G. Alenya, F. Moreno-Noguer, and C. Torras, "Using depth and appearance features for informed robot grasping of highly wrinkled clothes," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, may 2012.

[13] A. Caporali and G. Palli, "Pointcloud-based identification of optimal grasping poses for cloth-like deformable objects," in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, sep 2020, pp. 581–586.

[14] K. Yamazaki and M. Inaba, "A cloth detection method based on image wrinkle feature for daily assistive robots," in *MVA*, 2009, pp. 366–369.

[15] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," 2018.

[16] A. Singh, L. Yang, K. Hartikainen, C. Finn, and S. Levine, "End-to-end robotic reinforcement learning without reward engineering," 2019.

[17] M. Arduengo, A. Colomé, J. Lobo-Prat, L. Sentis, and C. Torras, "Gaussian-process-based robot learning from demonstration," 2020.

[18] N. Jaquier, D. Ginsbourger, and S. Calinon, "Learning from demonstration with model-based gaussian process," 2019.

[19] R. P. Joshi, N. Koganti, and T. Shibata, "Robotic cloth manipulation for clothing assistance task using dynamic movement primitives," in *Proceedings of the Advances in Robotics on - AIR '17*. ACM Press, 2017.

[20] X. Sun, X. Zhu, P. Wang, and H. Chen, "A review of robot control with visual servoing," in *2018 IEEE 8th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, jul 2018.

[21] Pomares, "Visual servoing in robotics," *Electronics*, vol. 8, no. 11, p. 1298, nov 2019.

[22] H. Lê, "Camera model: intrinsic parameters," Jul. 2018, [Online; accessed December 6, 2022]. [Online]. Available: https://lhoangan.github.io/camera-params/

[23] K. Wei, Y. Chu, and H. Gan, "A novel method of automatic hand-eye calibration for robotic manipulator," in *2021 2nd International Conference on Control, Robotics and Intelligent System*. ACM, aug 2021.

[24] R. Tsai and R. Lenz, "A new technique for fully autonomous and efficient 3d robotics hand/eye calibration," *IEEE Transactions on Robotics and Automation*, vol. 5, no. 3, pp. 345–358, jun 1989.

[25] F. Park and B. Martin, "Robot sensor calibration: solving AX=XB on the euclidean group," *IEEE Transactions on Robotics and Automation*, vol. 10, no. 5, pp. 717–721, 1994.

[26] X. Lin, Y. Wang, J. Olkin, and D. Held, "Softgym: Benchmarking deep reinforcement learning for deformable object manipulation," 2020.

[27] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[28] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Magazine*, vol. 12, no. 4, pp. 40–52, December 2005.