A Learning Method of Dual-arm Manipulation for Cloth Folding Using Physics Simulator

Daisuke Tanaka and Sho Tsuda
Dept. of Engineering
Grad. School of Sci. and Tech.
Shinshu University
4-17-1 Wakasato, Nagano, Japan
Email: 19hs203j@shinshu-u.ac.jp

Kimitoshi Yamazaki
Mech. Sys. Engineering
Faculty of Engineering
Shinshu University
4-17-1 Wakasato, Nagano, Japan
kyamazaki@shinshu-u.ac.jp

***Abstract*— This paper describes a learning method of cloth folding manipulation by a dual-arm robot. To reduce the burden on doing manipulation experiments using actual robots, cloth manipulation is virtually learned using physics simulator. The issue on using virtual environments is how to reproduce realistic movements of cloth in the simulator. We adopt methods of determining simulation parameters from three kinds of actual measurement. In addition, we propose a method using Bayesian optimization in order to obtain folding manipulation efficiently using the simulator. The method enables to select appropriate gripping positions and moving trajectories. The effectiveness of these methods were confirmed by means of simulation, and also by experiments using an actual robot.**

***Index Terms*— Cloth manipulation, Motion planning, Dual-arm robot.**

I. Introduction

In the work of folding a cloth product, it is necessary to grip appropriate positions and to follow an appropriate trajectory to perform a desired folding. In order to automate such task, it is desirable to grasp what kind of transformation happens to the cloth while folding and to generate a manipulation trajectory connecting the initial gripping positions and the goal positions.

The purpose of this study is to establish a method for acquiring appropriate folding trajectory required for the folding of cloth products by dual-arm robots. Since cloth products are a flexible object, its shape changes when manipulation is applied. Therefore, even if the start and target positions of manipulation does not change, the shape of the cloth after the manipulation might differ if the intermediate moving trajectory is different. Furthermore, some manipulation might be the cause of undesirable results, e.g. the whole body of the cloth product is moved. This means the target position of manipulation is also shift. Therefore, grasping positions and manipulation trajectory are important factors greatly related to the degree of accomplishment of the folding work.

Based on the above-mentioned fact, studies have been conducted to determine the method of folding. Among them, as one of the recent achievements, Petriík et al.[1] explained the appropriateness of manipulation trajectory combining multiple curves such as straight lines and circular arcs, and proposed a method for automatic generation of precise folding trajectories. They showed the effectiveness of the trajectory generation method by means of folding a rectangular cloth into two. Li et al.[2] showed a method of trajectory generation through physics simulation. The method includes the techniques to adjust the parameters of physical simulation from the behavior of actual cloth.

In the field of work automation, it is still a difficult task to predict and recognize the behavior of cloth. Due to the fact, in order to impose the manipulation of cloth product to the automated machine, more trial and error is required than the case of rigid objects. The studies described the above reduce its physical burden and are a promising approach. However, in the present form, they still consider a manipulation with a single arm, and the applicability in case of dual-arm manipulation is not discussed. However, folding of cloth products which humans normally do is often performed simultaneously on both arms. That is, in order to proceed folding tasks efficiently, it is desirable to be able to generate a trajectory considering being dual arms.

Therefore, we focus on the learning of manipulation method (gripping position and manipulation trajectory in folding) on the premise of using a dual-arm robot. Among others, in order to reduce the burden of discovering appropriate manipulation, we use a physics simulator. This approach is based on the method proposed by Li et al. and is then extended in the following points:

- We proposed a method for selecting gripping positions and a manipulation trajectory on the assumption of using dual-arm robots.
- We introduce Bayesian optimization to obtain an appropriate manipulation trajectory. Comparing to the previous approach[2], which uses a large amount of data for nonlinear optimization, our approach only needs a small number of trials.
- We investigated methods of physics simulation to mimic the behavior of actual cloth products, and found an additional criterion to reproduce the sagging of cloth in the vicinity of a gripping position. In this paper, we show how to adjust physical parameters based on actual measurement.

Based on the above, the contributions of this paper are to make it possible to learn the folding method carried out by a dual-arm robot.

The structure of this paper is as follows. The next section introduces the related works of cloth manipulation. Section III describes our issues and approach. Section IV explains the simulation of cloth using physics simulator, and describes the methods of parameter setting to mimic the behavior of simulated cloth into an actual one. Section V explains the optimization method using Bayesian optimization, and introduces how to use it in this study. Section VI reports our experimental results of learning, and we conclude this paper in Section VII.

II. Related work

Studies on cloth manipulation for making a cloth product into a desired shape state by an automatic machine has been conducted. These studies are roughly divided into two types, unfolding and folding. Representative issues on unfolding is how to determine gripping positions and where to move the gripped position [3]. Previous studies on unfolding are: unfolding of a rectangular cloth placed on a desk [4], flattening [5], ironing [6], and so on. Some studies premised to use dual-arm robots [7][8][9]. Also, there are studies such as an isolation task [10] that picks one out of multiple cloth products, and an extension that combines with classifying the types of the cloth picked up [11][12].

On the other hand, the issues on folding often include an unfolding task in its first step. That is, in order to create a shape that is easy to fold, the cloth product is first shifted to the unfolded state. Previous studies have been achieved a folding of rectangular fabric products even when the initial shape of the target cloth is in a unarranged shape[13][14]. Some studies achieved folding tasks with targeting cloth products with complicated form, such as T-shirts[2][15].

Among the above studies, Kita et al. [7] positively used physics simulation. In the step of the shape recognition of a cloth product, authors succeeded to improve the efficiency of the learning process as reducing the load of experiments on actual robots. However, there are remaining problems to use physics simulation. That is the difference in the behavior of cloth between the real world and the virtual world. Currently, simulation of flexible objects can be executed at relatively high speed by using a general physics simulator. However, the friction force between a flexible object and surrounding rigid bodies, or between two parts of cloth is particularly different. This is an essential problem because flexible objects in physics simulator are defined as a mesh structure, which is largely different from actual cloth structure. In order to alleviate this difference, a method of determining parameters of physics simulation from actual cloth behavior has also been proposed[2].

III. Issues and approach

A. Issues

We assume that a piece of cloth product with an unfolded state is placed on a table. The purpose is to fold the cloth product by a dual-arm robot. The shape of the cloth product after being folded is given beforehand. Therefore, it is sufficient that the shape after folding is close to the given shape. As cloth products, rectangular towels, T-shirts, trousers, are assumed. Gripping positions for them and releasing positions of the grasp are also given in advance.

B. Approach

Li et al.[2] proposed a method that finds a folding trajectory by a search of parameters for a Bezier curve. They introduced the Levenberg-Marquardt method to solve an optimization problem. In this method, it is necessary to repeat trials with a fine interval in order to obtain a good convergent solution. So, as introduced in Section II, they reduced the load of experiments using physical simulation. In relation to that, they showed how to determine material properties of cloth and friction with environments in physics simulation based on actual measured values.

The authors believe that this policy has high rationality in automating cloth manipulation. Therefore, we extend this approach to the case using a dual-arm robot. In addition, the following improvements are added in order to cope with more detailed cloth behavior and to improve the performance of manipulation learning.

- In the study by Li et al.[2], though the flexibility of the whole body of cloth was explicitly considered, it was not considered the local deformability of cloth in the vicinity of a point gripped by a robot hand. That is insufficient to predict cloth deformation when the cloth product is thin and soft, etc. Therefore, we make it possible to predict such deformation by physical simulation. This improvement enables to add gripping positions into search target in addition to manipulation trajectory.
- Instead of Levenverg-Marquardt method, which is one of the nonlinear minimization methods, we introduce Bayesian optimization. Compared to Levenverg-Marquardt method that requires a large amount of data at batch process, it enables to search appropriate manipulation with a small number of trials by sequential manner.

We use Blender[19], which is a 3D CG creation software, to simulate the movement of cloth products accompanying manipulation. Blender is a multi-platform open source software and has been used to implement various CGs. Since it has interactive programming by Python language, there is a possibility of useful for the development of robot applications. In addition, a physical engine software is embedded, and it enables to simulate flexible objects. However, since such object is composed of a three-dimensional mesh structure, there is an essential difference between virtual cloth and actual cloth. Therefore, the behavior when a manipulation is added to the virtual cloth is not the same as the actual one.

IV. Parameter settings for physics simulation

A. Adjustment of material characteristics[2]

While folding of cloth products, elongation might occur in the cloth by lifting, pulling or moving. When we reproduce it in the virtual world, cloth sometimes stretches unnecessarily

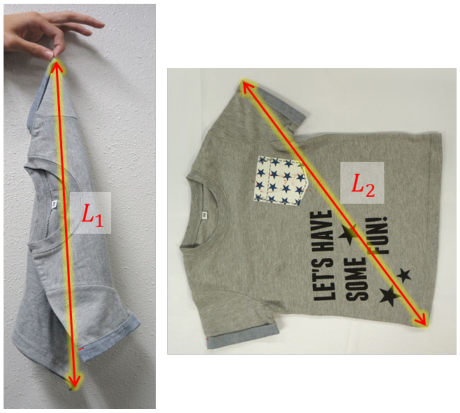


Fig. 1. Measurement of cloth elongation

large. Therefore, the characteristic values of the cloth should be set so as to obtain an appropriate elongation.

In the following, the procedure of folding shirt is explained as an example. First of all, prepare an actual shirt, grasp a part which can become the end part such as cuffs etc., and hang it. Then measure the distance from the gripping position to the lowest end of the shirt. Let the distance be L_1. Next, place the shirt on a horizontal plane so as not to disturb the suspended shape. Then, measure the distance between the same two parts as before. Let the distance be L_2. Fig.1 shows these situations.

On the other hand, for physics simulator, the procedure is as follows. First, generate a flexible mesh object of the same shape as the shirt, pick the same part and hang it, and measure the distance to the lowest end. Then place it on a flat plate and measure the distance again. After that, adjust the stiffness value for the vertex group of the mesh so that the difference of the distance is equivalent to $L_1 - L_2$.

B. Adjustment of friction force[2]

When a horizontal external force is applied to a cloth placed on a table, a part or the whole of the cloth slips. If we aim to fold the cloth, it is desirable to know the external force to the extent that the whole of the cloth is moved. For that purpose, it is desirable to run the simulation with appropriate friction parameters between the cloth product and the table.

The procedure to select the friction parameter is as follows. First, place an actual shirt in the unfolded state on a table. Let the length of the shirt be L_t. Then gradually tilt the table itself and record the lifting height H_s when the shirt starts to slide. Finally calculate $sin^{-1}(H_s/L_t)$ as the friction angle.

On the other hand, for physics simulator, the procedure is as follows. First, a high friction value is set on the table. Place a shirt on the table and tilt the table so that it will have the friction angle as the same above. After that, gradually decrease the friction parameter between the table and the cloth, and find out the value when the shirt starts to slide.

C. Parameter settings for locally sagging part

When a part of the cloth near the edge is grasped and is lifted up, cloth around the grasping position will hang down due to tension such as gravity. If it folds without considering this sagging, it will remain until the end of the manipulation and cause improper folding like the red circle part of Fig.2. Therefore, it is necessary to reproduce on the simulator what kind of sagging occurs when gripping which position.

Fig. 2. Unnecessary bending that occurs after folding

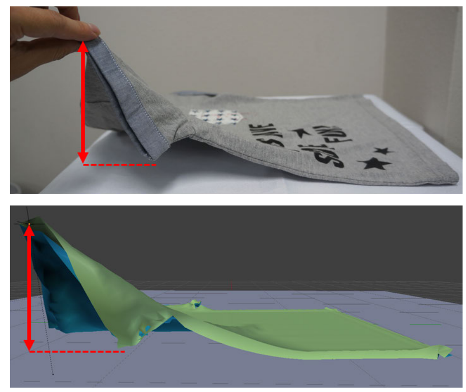


Fig. 3. Measurement of the bending state and adjustment of physics simulation

Therefore, the physical model is adjusted by setting the bending resistance parameter. The procedure is as follows. First, hold and lift the edge of the actual cloth product. Then, measure the vertical distance from the lowest point to the grasping part, as shown as a red arrow in the upper part of Fig. 3.

On the other hand, in the physical simulation environment as well, a cloth product is lifted up with grasping the same position as the actual cloth product. Then, as shown in the lower part of the figure, set the parameter values for the bending resistance so that the size of the sag is the same.

V. Manipulation Learning by Bayesian Optimization

A. Approach to manipulation learning

In this study, we regard the acquisition of cloth folding as a combination of the problem of determining the proper gripping positions and the problem of generating the moving trajectory of the hand. For this purpose, the coordinates of the gripping positions are searched. Also, the hand trajectory is regarded as a curve, and the position of the control point of the curve is properly selected.

The reason for setting the grasping position as a search target is to prevent the occurrence of unnecessary bending after folding as shown in Fig.2. In addition, about the folding trajectory, one criterion is to avoid a case that a cloth product is moved on a table during folding. The reason for this is that the positions to which the grasped parts should reach also changes if the position of the cloth product is changed. This makes trajectory generation be complicated.

We assume that an end-effector trajectory follows a Bezier curve. A Bezier curve $T(t)$ is an N-order curve defined by $N+1$ points. The equation is as follows:

$$T(t) = \sum_{k=0}^{N} B_k^n(t) P_k, \tag{1}$$

where $P_k = P_0, P_1, \cdots, P_N$ denotes control point and $B_k^n(t)$ denotes a Bernstein basis function. Under the condition that t changes from 0 to 1, a Bezier curve whose end points are P_0 and P_N are obtained. Notice that this curve does not path any points except the two end points. We set $N = 3$, therefore there are two control points between the end points. Folding trajectory is obtained by properly selecting the position of the two intermediate control points.

B. Bayesian Optimization

When a task is given and let a robot to obtain the required motion for the task, it is needed that the robot takes trial and error to accomplish the purpose of the task. In order to do it efficiently, there is an approach of setting some kind of objective function and optimizing it. If the function is differentiable to the input, and also convex function, the optimization can be accomplished by the gradient descent method and so on. However, it is difficult to define an appropriate objective function in deformable object manipulation, so that the function has to be treated as black-box. That is, it is difficult to apply common methods such as gradient descent.

In this study, we focus on Bayesian optimization [16]. Bayesian optimization can find the region where the optimal solution exists in a few trials even if the objective function is black-box. As a related work in the robotics field, Berkenkamp et al.[17] used safety Bayesian optimization to obtain control gain of a quadrotor. Nishimura et al. [18] searched for a gripping position for an underactuated gripper. On the other hand, we use Bayesian optimization for searching both grasping positions and folding trajectories of fabric products.

In Bayesian optimization, a Gaussian process is assumed as the prior distribution of the objective function $f(\boldsymbol{x})$. Such assumption makes it possible to obtain the posterior distribution of the objective function depending on input data $\boldsymbol{x}_{1:n} = [\boldsymbol{x_1}, \cdots, \boldsymbol{x_n}]^{\mathrm{T}}$ and output data $\boldsymbol{y}_{1:n} = [f(\boldsymbol{x_1}), \cdots, f(\boldsymbol{x_n})]^{\mathrm{T}}$. Then, the predicted distribution enables to obtain $f(\boldsymbol{x})$ from unknown input $\boldsymbol{x}$. It is represented by a Gaussian distribution with $\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})$, that is,

$$f(\boldsymbol{x}) \sim N\left(\mu(\boldsymbol{x}), \sigma^2(\boldsymbol{x})\right), \tag{2}$$

$$\begin{aligned}
\mu(\boldsymbol{x}) &= k_{\boldsymbol{\theta}}(\boldsymbol{x})^{\mathrm{T}}(K_{\boldsymbol{\theta}} + \beta^{-1}I)^{-1}\boldsymbol{y}_{1:n}, & (3)\\
\sigma^2(\boldsymbol{x}) &= c - k_{\boldsymbol{\theta}}(\boldsymbol{x})^{\mathrm{T}}(K_{\boldsymbol{\theta}} + \beta^{-1}I)^{-1}k_{\boldsymbol{\theta}}(\boldsymbol{x}), & (4)\\
k_{\boldsymbol{\theta}}(\boldsymbol{x}) &= (k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}_n))^{\mathrm{T}}, & (5)\\
K_{\boldsymbol{\theta}(i,j)} &= k_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j), & (6)\\
c &= k_{\boldsymbol{\theta}}(\boldsymbol{x}, \boldsymbol{x}) + \beta^{-1}, & (7)
\end{aligned}$$

where $k_{\boldsymbol{\theta}}(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a kernel function and $\boldsymbol{\theta}$ denotes parameters of the function. In this study, we use the Matern kernel which is commonly used in Bayesian optimization. β denotes a parameter that represents the uncertainty of the objective function. In this paper, β is set to 0 because the simulation is assumed to be used.

Bayesian optimization uses an acquisition function $a(\boldsymbol{x})$, which is a lower cost function than the objective function $f(\boldsymbol{x})$. The optimization algorithm is accomplished by successively generating $\boldsymbol{x}_{n+1}$ that maximizes $a(\boldsymbol{x})$. Therefore, the behavior of the optimization process changes depending on how to formalize $a(\boldsymbol{x})$. In this study, GP-EI represented by the following equation is introduced,

$$a(\boldsymbol{x}) = \mathrm{E_{GP}}[\max(f(\boldsymbol{x}) - f(\boldsymbol{x}^*_{1:n}), 0))]. \tag{8}$$

GP-EI uses the predicted distribution by Gaussian process, and calculates an expected value of the amount of improvement for the optimal value $f(\boldsymbol{x}^*_{1:n})$ in the data set. In the procedure of Bayesian optimization using GP-EI, the next observation point $\boldsymbol{x}_{n+1}$ is selected as the present input that maximizes the expected value.

The procedure for the optimization of the objective function is as follows. First, some data are randomly sampled for calculating a posterior distribution of the objective function. Next, a predicted distribution of the objective function is calculated using the data set $\boldsymbol{x}_{1:n}, \boldsymbol{y}_{1:n}$. Then, $a(\boldsymbol{x})$ is calculated from the predicted distribution, and the point with the largest value of $a(\boldsymbol{x})$ is taken as the next observation point $\boldsymbol{x}_{n+1}$. Next new data $\{\boldsymbol{x}_{n+1}, f(\boldsymbol{x}_{n+1})\}$ are sampled and the data set is updated. After the updating, the next sampling point is selected. As described above, repeat sampling and updating alternately to find $\boldsymbol{x}^*$ that optimizes the objective function.

VI. Proof experiments

A. Settings

A T-shirt, a rectangular towel, and a pair of pants were selected as cloth products to be folded. First, actual cloth products were prepared. Materials of these products were cotton. The dimensions of each are shown in Fig.4. Then, 3D mesh models for use in Blender were made. Since the shirt and the pants have the annular part which put a plurality of cloths together, technical devices were necessary for the modeling. Taking the T-shirt as an example, we explain the implementation method.

First, define a cloth object that simulates the shape of the front of the shirt. Then copy it and create another cloth object. Next, overlap the two objects, connect the edges of each other, and then register the connected vertices. Then, the connection edges are contracted and joined using the stitching function.

Fig. 4. The dimensions of target fabric products. Left: Towel, Center: T-Shirt, and Right: Pants.

TABLE I
PHYSICAL PARAMETERS FOR THREE TYPES OF CLOTH PRODUCT

	T-Shirt	Towel	Pants
Multiplier (value related to speed)	1.0	1.0	1.0
Weight (mass of material)	0.3	0.3	3.0
Structure (overall hardness)	15.0	1.5	10.0
Spring (value affecting vibration)	5.0	5.0	5.0

TABLE II
ADDITIONAL PHYSICAL PARAMETERS DETERMINED BY THE METHODS DESCRIBED IN SECTION IV

	T-Shirt	Towel	Pants
Structural rigidity	25.0	6.0	10.0
Friction	80.0	55.0	80.0
Bending of fabric	6.0	20.0	10.0
Bending resistance	15.0	30.0	10.0

This procedure makes it possible to construct a model suitable for folding simulation.

Table I shows physical parameter values on defining cloth object. These were manually adjusted. The parameters obtained for the three types of adjustment described in Section IV are also shown in Table II. The parameter for section IV-B is "Structural rigidity," the parameter for section IV-C is "Friction" and the parameters for section IV-D are "Bending of fabric" and "Bending resistance."

B. Aquisition of folding motion

Using a physics simulation, folding motions were acquired by the proposed method explained above. The parameters obtained by Bayesian optimization were defined as a 6-dimensional vector as follows:

$$\mathbf{x} = \left(x_1, z_1, x_2, z_2, g_1, g_2\right), \tag{9}$$

where $x_1 \sim z_2$ indicates the x and z coordinates of P_1 and P_2, respectively. P_1 and P_2 are the control points of Bezier curve. g_1 and g_2 describe the grasping points for folding, and they are also the positions of P_0. These grasping points were a discrete variable because they were chosen from M candidates of the mesh model edges. However, it is difficult to optimize the parameter mixed continuous and discrete variables. So We treat g_1 and g_2 as continuous variables like $0 \leq g1, g2 < M$ while paramter searching, and converted these variables into an integer by floor function when calculating the value of an objective function. P_3 which is a control point of the Bezier curve was set as the position of releasing after folding. It was determined by parallel translation of a fixed distance from the grasping point.

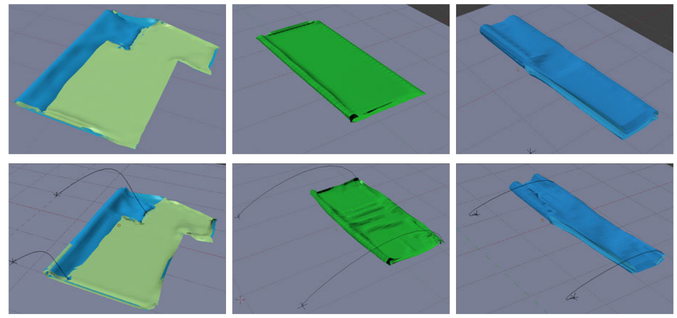


Fig. 5. Results of folding using an acquired manipulation method. Left: T-Shirt, Center: Towel, and Right: Pants. The upper row shows the target shapes, and the lower row shows the results of folding.

The objective function was formalized as follows:

$$C\left(x\right) = l_x + \alpha D\left(S_t, S_x\right), \tag{10}$$

$$D\left(S_t, S_x\right) = \Sigma_i w_i e_i, \tag{11}$$

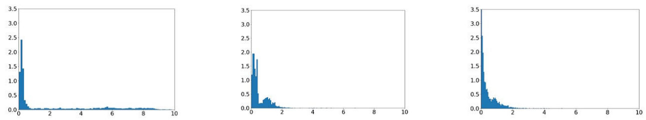
$$e_i = \|q_i - y_i\|, \tag{12}$$

$$w_i = max(e_i - (\mu_e + 1.28\sigma_e), 0) + 1, \tag{13}$$

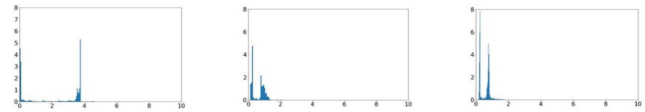
where l_x describes the length of the manipulation trajectory, α is the ratio, and α was empirically set to 0.01. $D(S_t, S_x)$ is the sum of the vertex distance e_i between the two mesh models of a target shape S_t and a manipulation outcome S_x. If the folding manipulation perfectly accomplished the target shape S_t, the value of $D(S_t, S_x)$ was 0 because each vertex of S_t and S_x are overlapped. If a cloth product is slipped overall during a folding manipulation, $D(S_t, S_x)$ become large because S_t is fixed at the initial position. On the other hand, considering the case where an unnecessary bending occurs after folding as shown in Fig. 2, most regions of S_x should match with S_t. In such a situation, if the distances e_i is simply summed between the two mesh models, the evaluation value will be small. Therefore, the coefficient w_i was added so that the evaluation value would be increased if unnecessary bending occurred in the local part. μ_e and σ_e shown in Eq. (13) are the mean and the standard deviation of e_i. Under the setting above, the objective function $C(x)$ was minimized by means of Bayesian optimization. The initial value was given one piece of data obtained randomly, and the search was performed from the situation.

C. Discussion

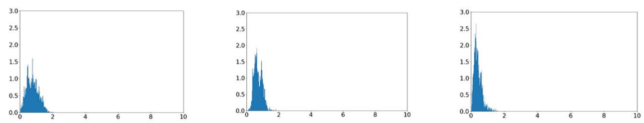
For each of the three types of T-shirts, towels, and trousers, 50 samplings were performed for Bayesian optimization, and the best result on Eq. (9) was selected. The upper panels of Fig. 5 show the goal state of the folding, and the bottom panels show the folded result according to the acquired manipulation. The black curve shown in each bottom panel is the acquired end-effector trajectory. Although the trajectories are different in each cloth product, a folded shape was obtained that was close to the goal state.



(a) T-Shirt



(b) Towel



(c) Pants

Fig. 6. Frequency histogram of the position error of the mesh model with respect to the result of folding obtained by means of Bayesian optimization. The first column shows the result of random sampling (initial phase). The second column is the result just after 10 times search, and the third column is the result with the best manipulation method in the data just after 50 times search.

Figure 6 shows frequency histograms that show the distance between each vertex of the mesh model after folding manipulation using Eq. (12). The horizontal axis means the distance, and the vertical axis means frequency. When the manipulation result is close to the target shape, many distance values e_i should be close to 0, therefore all the frequency close to the left end. The graphs in Fig.6 consist of three graphs per line. From the left, it shows the initial state, the result of 10 times sampling, and 50 times sampling. As the frequency value is on the left as the number of searches increases, therefore it can be confirmed that appropriate manipulation was found by Bayesian optimization

In Fig.6b, there are graphs showing two peak values. This is a case where the upper side of cloth can reproduce the arrangement close to the target shape, but the lower side of cloth slightly slips and the error with the target shape becomes large. Even in that case, it can be said that it was possible to search for a trajectory that would not slip the cloth since the two peaks approached 0 as the sampling progressed.

Finally, the folding of cloth products by an actual robot was executed using the obtained gripping positions and trajectories, and it was confirmed that folding was possible. The robot used was HIRONX [20] made by Kawada Robotics Inc. This robot is a dual-arm robot. The degrees of freedom (DoF) in one arm is six, and it has a movable range necessary for the purpose of folding the above three types of cloth products

The procedure to execute the manipulation by the robot is as follows. First, the obtained gripping positions were gripped. Next, the obtained trajectories were divided into 10 frames interval to create via-points, and the motion of end-effectors were calculated to pass them smoothly. During manipulation, the orientation of end-effectors was fixed from the state of gripping the cloth first. Then, after moving the end-effector to the final point of the trajectory, the robot slowly released the grip so that the shape of the cloth did not collapse.

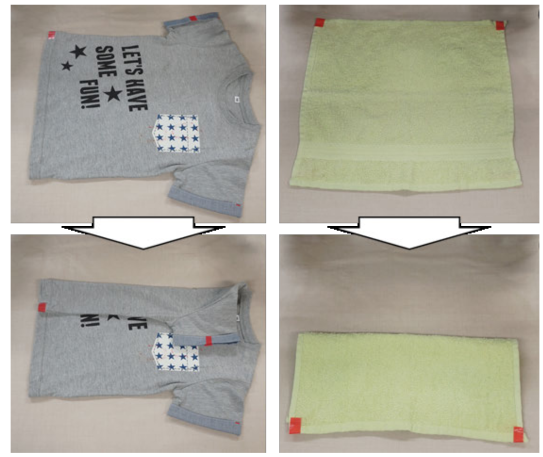


Fig. 7. T-shirt and towel folded by an actual robot

Fig. 8. Pants folded by an actual robot

Figure 7 and 8 shows the states of actual cloth products that are before and after folding. The red markers show the obtained for gripping positions. From these figures, it can be recognized that appropriate folding manipulations were obtained by the proposed method.

VII. Conclusions

In this paper, we described a learning method of cloth folding manipulation by a dual-arm robot. We adopted to introduce physics simulator to the learning process because trial and error using actual cloth products are significantly burdensome work. We extended a previous method that assumes to use single-arm robots. The proposed method enables to adjust physical parameters that can reproduce the behavior of an actual cloth, and to search for an appropriate manipulation more efficiently than the previous method. Using Blender as a physics simulator, it was confirmed that folding manipulation can be obtained by three kinds of cloth products. Furthermore, the obtained manipulation was executed by an actual robot, and it was confirmed that the folding could be realized.

Future work includes increasing the number of cloth products to verify the effectiveness of the proposed method. To reduce the load and time for preparing physics simulations is also important.

ACKNOWLEDGEMENT

This work was partly supported by NEDO and JSPS KAKENHI Grant Numbers JP18K19809.

REFERENCES

[1] V. Petriík, V. Smutný, P. Krsek, V. Hlaváč: Single arm robotic garment folding path generation. Advanced Robotics, Vol. 31, No.24, 1325 - 1337, 2017.

[2] Y. Li, Y. Yue, D. Xu, E. Grinspun, P. K. Allen: "Folding Deformable Objects using Predictive Simulation and Trajectory Optimization," in Proc. of IEEE/RSJ IROS, 2015.

[3] B. Willimon, I. Walker, and S. Birchfield: "A New Approach to Clothing Classification using Mid-Level Layers," in Proc. of the IEEE Int'l Conf. on Robotics and Automation, 2013

[4] E. Ono, H. Okabe, H. Ichijo, N. Aisaka and H. Akami: "Robot hand with sensor for handling cloth," in Proc. of IEEE Int'l Workshop on Intelligent Robots and Systems, vol. 2, pp. 995-1000, 1990.

[5] L. Sun, G. Aragon-Camarasa, S. Rogers, J. P. Siebert: "Accurate Garment Surface Analysis using an Active Stereo Robot Head with Application to Dual-Arm Flattening," in Proc. of IEEE ICRA, pp. 185-192, 2015.

[6] Y. Li, X. Hu, D. Xu, Y. Yue, E. Grinspun, P. K. Allen: "Multi-Sensor Surface Analysis for Robotic Ironing," in Proc. of IEEE ICRA, pp. 5670-5676, 2016.

[7] Y. Kita, F. Kanehiro, T. Ueshiba, N. Kita: "Clothes handling based on recognition by strategic observation," in Proc. of Int'l Conf. on Humanoid Robots, pp.53-58, 2011.

[8] Shibata M, Ohta T, Hirai S, et al.: "Robotic unfolding of hemmed fabric using pinching slip motion," in Proc. of Int'l Conf. on Advanced Mechatronics. pp. 392-397, 2010.

[9] H. Yuba, S. Arnold, K. Yamazaki: "Unfolding of a rectangular cloth from unarranged starting shapes by a Dual-Armed robot with a mechanism for managing recognition error and uncertainty," Advanced Robotics, http://dx.doi.org/10.1080/01691864.2017.1285722, 2017.

[10] K. Hamajima and M. Kakikura: "Planning Strategy for Unfolding Task of Clothes -Isolation of clothes from washed mass-," in Proc. of Int'l. Conf. on Robots and Systems, pp. 1237-1242, 2000.

[11] F. Osawa, H. Seki, and Y. Kamiya: "Unfolding of Massive Laundry and Classification Types by Dual Manipulator," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol.11 No.5, pp. 457-463, 2007.

[12] A. Doumanoglou, A. Kargakos, T-K. Kim, S. Malassiotis: "Autonomous Active Recognitionand Unfolding of Clothes using Random Decision Forests and Probabilistic Planning," in Proc. of IEEE ICRA, 2014.

[13] J. Maitin-Sphepard et al.: "Cloth Grasp Point Detection based on Multiple-View Geometric Cues with Application to Robotic Towel Folding," in Proc. of Int'l. Conf. on Robotics and Automation, pp.2308-2315, 2010.

[14] D. Tanaka, S. Arnold, K. Yamazaki: "EMD Net: An Encode-Manipulate-Decode Network for Cloth Manipulation," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1771-1778, 2018.

[15] C. Xiong, N. Shukla, W. Xiong, and S. Zhu: "Robot Learning with a Spatial, Temporal, and Causal And-Or Graph," in Proc. of IEEE ICRA, pp. 2144-2151, 2016.

[16] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms.", Advances in neural information processing systems. 2012.

[17] F. Berkenkamp, A. P. Schoellig, and A. Krause, "Safe controller optimization for quadrotors with Gaussian processes." in Proc. of Int'l. Conf. on Robotics and Automation, pp.491-496, 2016.

[18] T. Nishimura, K. Mizushima, Y. Suzuki, T. Tsuji and T. Watanabe: "Variable-Grasping-Mode Underactuated Soft Gripper With Environmental Contact-Based Operation," in Proc. of IEEE Robotics and Automation Letters, vol. 2, no. 2, pp. 1164-1171, 2017.

[19] https://www.blender.org/

[20] http://nextage.kawada.jp/hiro/