Interactive Exploration of Drapes by Simulation Parameters

Eunjung Ju
CLO Virtual Fashion Inc.
kate@clo3d.org

Kwang-yun Kim
CLO Virtual Fashion Inc.
justin@foxxing.com

Jaehoon Lee
CLO Virtual Fashion Inc.
jack@foxxing.com

Sungjin Yoon
CLO Virtual Fashion Inc.
mason@clo3d.com

Myung Geol Choi
The Catholic University of Korea
mgchoi@catholic.ac.kr

Abstract

How similar a virtual product is to a real product is one of the most important issues when using virtual simulation to develop real apparel designs. The first step to achieve high similarity is finding optimal simulation parameters for the desired fabrics. However, it is notoriously difficult to find an optimal parameter set that reproduces the physical properties of a specific fabric as closely as possible. It is because the relationship between the changes of simulation parameters and drape shapes is highly non-linear, not intuitive, and hard to be predicted even by experts. Therefore, users have to repeat trial and error based on personal experience until they find satisfactory results, which is time consuming due to the simulation time required for each trial. To handle this problem, we proposed a neural network model that learns the relationship between the parameter space and the drape space, then we presented a user interface that allows users to quickly explore the extensive drape space through simulation parameters. To validate our method, we provided our UI with experts in the fashion design industry and conducted user studies with them for qualitative evaluation.

Keywords: User Interface, Parameter Tuning, Cloth Simulation

1 Introduction

In the industry, virtual manufacturing is widely used to develop new usable apparel designs. The first step of virtual manufacturing is finding optimal simulation parameters for the desired fabrics. These parameters should be able to simulate the drape of the fabric as close to the real fabric in imitation as possible. However, finding optimal parameters for a specific fabric is notoriously difficult, because the relationship between the changes of drape shapes and simulation parameters is highly non-linear, not intuitive, and hard to be predicted even by experts. It can be more difficult for the end users of virtual fashion software, who are fashion designers not engineers.

The general procedure for determining simulation parameters is as follows: First, the initial value is obtained by measuring the physical properties of the target fabric. With this application in mind, most virtual fashion design software come with their own simulator-specific measuring device. However, the parameters obtained from these devices are often not sufficiently accurate for use in the development of real products. It means that the simulated drape with the parameters will not be sufficiently similar to the actual drape. Therefore, second, manual parameter tuning is required. This step basically repeats the process of adjusting the parameters based on the user's intuition, re-simulation and confirmation of the final drape. One draping simulation takes at least tens of seconds. Therefore, completing one parameter tuning task takes from tens of minutes to an hour or more for one specific fabric.

In this study, we introduce a UI that allows users to interactively explore simulated drape shapes by changing simulation parameters. We trained a deep neural network with a large amount of training data, so simulation results (drapes) can be instantly inferred from a given set of simulation parameters. A simulated drape is a three-dimensional object represented as a mesh model. Generally, a highly complex neural network model is necessary, such as MeshNet [1], to handle mesh data. In addition, the possible inconsistency of structures (with regard to the number of vertices and connectivity between vertices) between meshes makes it more difficult to use them as training data. To avoid these complexities, we developed a novel training method specialized for simulated drape shapes. We discovered that a full drape shape can be reconstructed from the edge curve of the drape with small errors. Instead of full drape meshes, we trained it to learn the relationship between simulation parameters and the shapes of the edge curves of the drape. As a result, we were able to train it with a simple multilayer, fully connected neural network.

To validate our method, we first measured the errors between the full drape meshes and reconstructed drape meshes from the edge curves. Then, we proved that our deep neural network model can learn the relationship between the simulation parameters and the edge curves of the drapes within an acceptable error. Finally, we provided our UI with experts in the fashion design industry and conducted user studies with them for qualitative evaluation.

2 Background

Virtual fashion design software are being widely used in the fashion industry to develop prototypes of real products. The virtual cloth products should be as close to the real products as possible, but before that can be achieved, the physical properties of the virtual fabric should be as close to the real fabric as possible. The former is determined by simulation parameters. Therefore, finding optimal simulation parameters for real fabrics is one of the most important issues in virtual cloth manufacturing.

2.1 Measuring Simulation Parameters

The Kawabata Estimating System (KES) [2] is the most representative industrial device for measuring various physical properties of fabrics. Industrial measuring equipments guarantee high accuracy, but they are difficult, complex, and expensive for ordinary people to use. Also, because the physical dynamics of cloth simulation are different from those of the real world, using more physically accurate values will not produce more realistic simulation results. For this reason, some virtual fashion software provide their own method and device to measure the simulation parameters for a given fabric sample. Generally, the user first measures several physical states of the fabric sample, which physical states are something that can be easily measured by anyone but are not the simulation parameters themselves. The measured values are then converted into actual simulation parameters by simulator-specific algorithms. For example, the user measures the bending angle of the fabric sample placed in a specific experimental environment. Then, the bending stiffness, a simulation parameter, can be computed from the angle. However, the accuracy of the simulation parameters obtained in this way is often not satisfactory for use in the development of real-world fabric products. This is due to a number of reasons such as the limitations of the expression range of the measurement method, the condition of the fabric sample at the time of measurement, and human measurement errors. We recommend readers to refer to [3] for more information on the pros and cons of different methods of measuring simulation parameters.

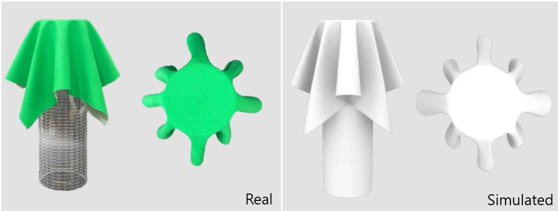


Figure 1: An actual example of manual parameter tuning

2.2 *Estimating Simulation Parameters*

There have been studies using machine learning to estimate simulation parameters that best reproduce the target drape shape. These studies can be divided into two categories: optimization and supervised learning. The former basically is the iterative process of adjusting parameters and simulating the drape until it finds parameters that reproduce a drape sufficiently similar to the target [4, 5, 6, 7]. This approach requires a long computation time the simulation to be repeated. Estimating a parameter for a single target fabric takes from a few to tens of hours.

The supervised learning approaches concern learning the relationship between the drape shapes and simulation parameters from a large amount of training data [8, 9, 10, 11] in order to estimate the simulation parameters from a given drape shape. With this method, users need to somehow measure the shape of the actual drape; thus, the problem of the human measurement error still remains. The other important issue with this approach is the underlying assumption that there is only one unique set of parameters that gives rise to a particular drape shape, but it is almost impossible to find a draping method that satisfies this assumption. Two different sets of parameters can result in almost identical drape shapes. However, our learning model is trained for the relationship that progresses in the opposite direction. That means the model is designed to estimate the drape shape from a parameter set. If the simulator is deterministic, one parameter set always produces the same drape shape.

2.3 *Simulation Parameter Tuning*

Manual parameter tuning is essential to compensate for the lack of measured or estimated parameters. However, to the best of our knowledge, there are no studies of UIs for tuning cloth simulation parameters. The commonly used method is to set the same drape condition in the real and virtual world and then to repeat adjusting the parameters and simulating until the shape of the virtual drape is close enough to that of the real one. This process is difficult and time consuming even for experts, because the relationship between the parameter change and drape shape change is not intuitive and difficult to predict and the simulation takes time. For example, Figure 1 shows an actual example of manual parameter tuning, which is provided by a professional virtual fashion designer. The left image shows the drape of the real fabric, and the right image shows a simulated drape observed during the parameter tuning task. In this case, the shape of the fabric specimen was 55 x 55 cm square, and the virtual specimen had 22,005 vertices and 43,564 triangle faces. The draping simulation was repeated dozens of times to complete the parameter tuning task, which took about $30\,\mathrm{min}$.

2.4 *Draping methods*

Various draping methods have been developed to analyze the properties of different fabrics. Cusick's drape [12] is the most representative method being used in the textile industry. In this method, a circular sample of fabric with a diameter of $30\,\mathrm{cm}$ is placed on the upper surface of a cylinder with a diameter of $18\,\mathrm{cm}$. The unsupported area of the sample flows down and forms a shape of drape. Various meaningful features of Cusick's drape have been investigated [13], and many variations of Cusick's drape have been studied [14, 15, 16]. We adopted a variation of Cusick's method proposed in [11]. In this method, squared fabric samples were used instead of circular ones, and a thinner cylinder was used for the supporting object. As opposed to the original Cusick's method, it is advantageous to distinguish differences in stretch stiffness because of the longer portion flowing down and because, through this, the distinction between the weft and warp directions also becomes clearer.

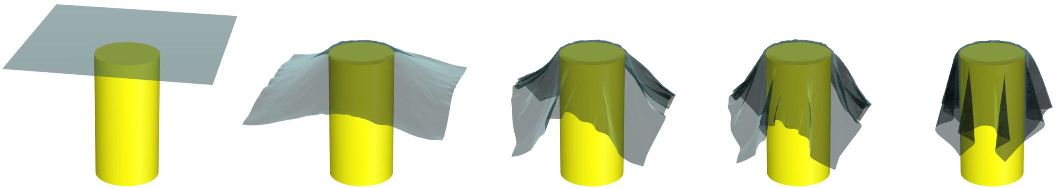


Figure 2: An example of draping simulation

Table 1: Simulation parameters

Symbol	Parameter
SU	Stretch force coefficient in the weft
SV	Stretch force coefficient in the warp
SH	Shear force coefficient
BU	Bending force coefficient in the weft
BV	Bending force coefficient in the warp
BD	Bending force coefficient in the diagonal
D	Density

3 Draping Simulation

3.1 Cloth simulation model

We deliberately used the simulation model of CLO3D in version 5.1 for our experiments. The first reason is that its practicality has been verified since it is one of the most widely used software products in the virtual fashion industry. The second reason is that it offers a parameter measurement device specialized for its simulator [17]. We used this device to obtain initial values for the parameter tuning tasks in the final user study (see Section 6.2). The simulation model of CLO3D is an implementation of Baraff's method [18], which uses the implicit integration method to maintain the stability of simulations for large-time steps and high-resolution fabric models. The seven simulation parameters that have the greatest influence on the drape result are listed in Table 1. These are also the input variables in our learning model (see Section 5) and controllable parameters in the UI (see Figure 4(b)). Our method was independent of the internal mechanisms of the simulation model, and the simulation model was not a contribution of our work. Therefore, we omit a more detailed description of the simulation algorithm and recommend readers refer to [18] for more details.

3.2 Draping Method

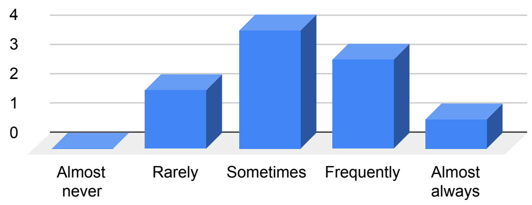
We used the draping introduced in [11]. This method involves dropping a 30 x 30 cm square fabric sample spread out on a $10\,\mathrm{cm}$ diameter cylinder. To build a high-resolution fabric model, the vertices were randomly sampled at intervals of $5\,\mathrm{mm}$, and the fabric mesh has 6,554 vertices and 12,862 triangle faces. Figure 2 shows an example of the draping simulation. The leftmost figure is the initial state, and the rightmost figure is the final state. The end condition is that all vertex speeds are under a certain threshold. The simulation time step was set to $0.033\,\mathrm{s}$ for all the experiments. The time that it takes to satisfy the end condition depends on the simulation parameters. The stiffer parameters tend to delay the time for the simulation to satisfy the end condition. Using a computer with an Intel i7-8700 (3.2 GHz) processor and 16 GB RAM, we spent between of seconds to minutes to complete one draping simulation.

4 User interfaces

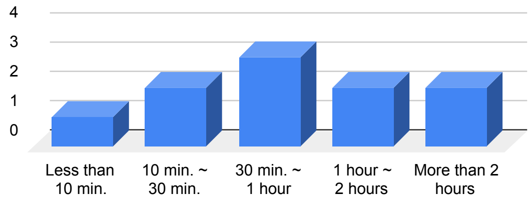
To verify the necessity of our research, we first conducted a preliminary user study with ten experts currently working as professional designers in the virtual fashion design industry. We asked them the following three questions. Figure 3 shows the five choices for each question and the number of times each option was chosen as a response.

Q1: How often did you have to perform additional manual parameter tuning instead of using the instrumentally measured parameters as they are?

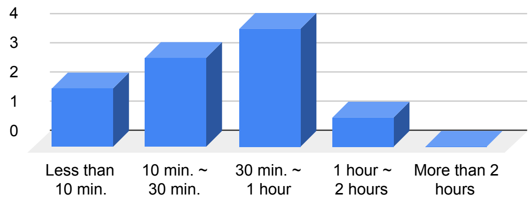
Q2: What was the longest time taken to complete parameter tuning for one fabric in your experience?



(a) Q1. How often manual parameter tuning were required



(b) Q2. Longest parameter tuning time



(c) Q3. Average parameter tuning time

Figure 3: Results of preliminary user study

Q3: What is the average time taken to complete parameter tuning for one fabric in your experience?

Most participants answered that manual parameter tuning is at least occasionally necessary. In their experience, the average parameter tuning time for one fabric was about 30 min, and in the worst case, it took more than two hours. In addition, we conducted face-to-face interviews with the ten experts, investigating their experience in parameter tuning works. Our UI was developed based on their experience. Figure 4 shows a screenshot of the UI. The main function is to interactively explore the result of the draping simulation as the user changes the simulation parameters. In Figure 4, (a) is the 3D viewport for displaying the estimated drapes and (b) is the set of sliders for adjusting the parameters. We used logarithmic sliders based on the previous observation; the correlation between the changes in the drape shape and the changes in physical properties are in a log-linear relationship [8, 10].

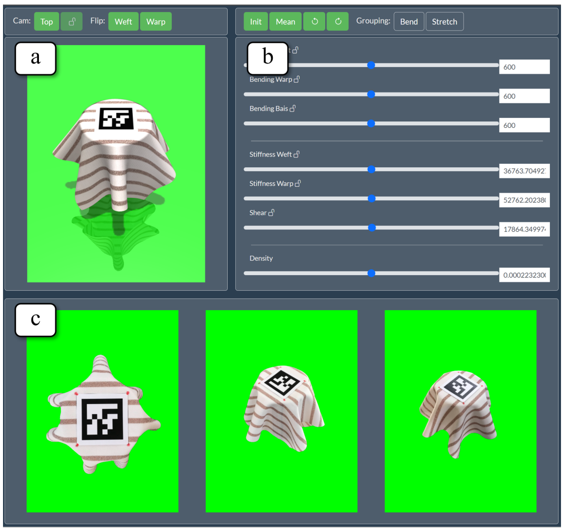


Figure 4: A screenshot of the user interface

Through the interviews, we found that when comparing real and simulated drapes, the flipped shape of the drape in the weft or warp direction is considered the same. This is because most commercial simulators assume that physical properties are independent and uniform for each of the three axes: weft, warp, and diagonal. Therefore, theoretically, there is a 50% chance that the simulation result will be flipped in the weft or warp direction. Reflecting this user experience, we implemented buttons to flip the drape shape in the weft and warp directions.

In addition, many of the interviewees wanted to see the simulated drape in the same angles shown by the photograph of the target drape. To implement this feature, we used an ARUCO [19] marker. When taking a photograph of the real drape, we placed an ARUCO marker on the top of the drape. By detecting the marker and analyzing its shape, we could infer the intrinsic camera matrix of the photograph. Figure 4(c) is where the photographs of the real drape are displayed in our UI. In this example, the three photographs taken from different angles are displayed. The background of each photograph is colored green to make the marker detection easier and the shape of the drape clearly visible. When one of them is clicked, its intrinsic camera matrix is computed and applied to the virtual camera of the 3D viewport.

5 Learning

Our goal is to find the function, $f : P \rightarrow D$, where P is the simulation parameter space and D is the set of all possible drape shapes. We trained the function f through a supervised learning method.

5.1 Training Data Generation

To collect large-size training data, we could randomly sample the simulation parameter spaces and then run the draping simulation with each of the sampled parameters. In the sampling, we wanted to exclude invalid parameters that are not physically possible or cannot be regarded as those belonging to fabric. These invalid parameters could cause the divergence problem in the simulation, which could unnecessarily broaden the domain of the parameter space, making learning difficult. To avoid this risk, we sampled the simulation parameters according to the probability distribution of a set of validated parameters. Specifically, we were provided with a set of simulation parameters validated for 400 different types of fabrics from CLO Virtual Fashion Inc. [20]. A Gaussian mixed model (GMM) with five components was fitted to the 400 parameters. Then, we sampled new parameter sets according to the probability distribution.

5.2 Data Representation

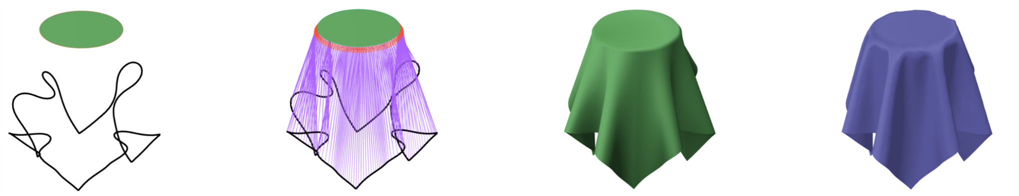
Each input data, $p \in P$, is normalized by log transformation and then scaled in the range $[0, 1]$. The output data of the training can be the drape meshes themselves. However, using raw 3D mesh data as training data will overcomplicate the training model and consequently negatively affect training accuracy. Instead of using the entire mesh data, we used only the edge curves of the drape meshes as the output data. The black curve in Figure 5(a) is an example of the edge curve. We assumed that the full shape of drape could be approximated from its edge curve. This assumption was deemed reasonable because creased fabrics were excluded from our consideration and there was no external force other than gravity, such as wind force. An edge curve is represented as a sequence of 244 uniformly sampled 3D points, which makes it a 732 dimensional vector.

5.3 Reconstruction

Figure 5 shows an example of reconstruction: (d) is the original mesh and (c) is the reconstructed result from the edge curve of (d). The reconstruction procedure is as follows. First, a circular mesh is added for the portion of the drape supported by the cylinder (see Figure 5(a)). The shape of the circle is the same as the top face of the cylinder, but the position is raised slightly to reflect the thickness of the fabric. Then, we attached a ring-shaped triangular strip mesh of $4\,\mathrm{mm}$ width around the circular mesh (the red mesh in Figure 5(b)). This ring strip was used to smoothen the flow of the drape near the edge of the cylinder. The heights of vertices on the outer circle of the ring are set $4\,\mathrm{mm}$ lower than the circular mesh. Finally, we filled another triangular strip between the edge curves of the drape and the ring strip (the violet mesh in Figure 5(b)). In order to prove that the reconstructed drape is sufficiently similar to the original drape, the following experiment was performed. First, we made 400 drapes by simulation with the parameter sets used to fit the generative model, reconstructed them again from their edge curves, and investigated the differences. We computed the average Hausdorff distance (AHD) and Hausdorff distance (HD) between the original and reconstructed drapes. Figure 6 shows the histograms of AHD and HD. In most cases, AHD was less than $2\,\mathrm{mm}$, and HD was less than $10\,\mathrm{mm}$. Figure 7 shows the case of maximum HD in a top view, and the color on the original drape (right one) means distance per vertex. As shown in the figure, the error is usually caused by the difference in the width of the folds. Folds of the reconstructed drape tend to be narrower than those of the original. However, even in the worst case, there is no significant difference in the overall shape.

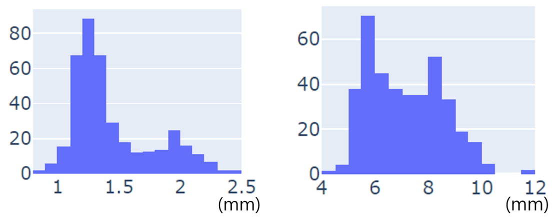
5.4 Deep Neural Network Model

Thanks to the use of edge curves instead of full meshes, we can train with a simple multilayer fully connected neural network. The input layer and output layer have seven (the size of the parameter vector) and 732 (the size of the edge



(a) Edge curve and top (b) Two triangle strips (c) Reconstructed drape (d) Original drape

Figure 5: An example of drape reconstruction



(a) AHD. (b) HD.

Figure 6: Histograms of AHD and HD between original and reconstructed drapes.

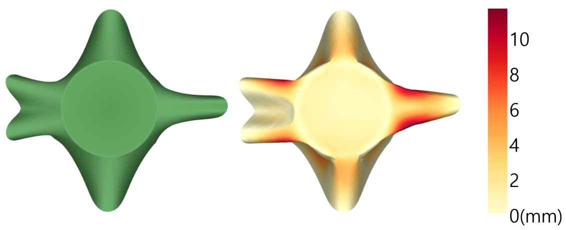


Figure 7: The case of maximum HD in a top view. The left one is the reconstructed drape, and the right one is the original drape. The color on the original drape means per-vertex distance.

curve vector) nodes, respectively. Our model has five hidden layers, and the numbers of nodes at each layer are 512, 4096, 4096, 4096, and 8192, respectively. Our network uses ReLU activation for each layer except the output layer, which uses linear activation.

6 Evaluation

6.1 *Training Accuracy*

We sampled 100,000 parameter sets using the GMM model (see Section 5.1) and then ran the draping simulation with each sampled parameter set. Then, we removed cases where the simulation did not converge after a certain period of time or where the final drape shape was judged

Table 2: Means and standard deviations of E_m

	Training	Test	Validation
Mean	$5.07\,\mathrm{mm}$	$5.46\,\mathrm{mm}$	$5.42\,\mathrm{mm}$
STD	$2.54\,\mathrm{mm}$	$2.87\,\mathrm{mm}$	$2.82\,\mathrm{mm}$

not to be a typical fabric shape (e.g., where it was too sagging or falling to the ground). The size of the final training data was 92,600. We used 80% of the total data as training data and split the remaining data in half to use as test data and validation data, respectively. The model was trained for 300 epochs with the mean square error loss function and the Adam optimizer. In order to intuitively understand the prediction error, we recalculated the error of each prediction in millimeters as follows.

$$E_m(y, \bar{y}) = \frac{1}{n} \sum_{i=1}^{n} \|y_i - \bar{y}_i\|, \qquad (1)$$

where y_i and $\bar{y}_i$ are the 244 ($= n$) sampling points of the original edge curve and the predicted edge curve, respectively. Table 2 shows the means and standard deviations for total E_m of the training, test, and validation data.

As reported in [11], if the mean difference of two contours is less than $5\,\mathrm{mm}$, there is little visual difference in the shape of the drape. Even at $10\,\mathrm{mm}$, which is about twice the mean, the overall drape shapes of the original and the predicted are quite similar. Since the purpose of parameter tuning is not to reproduce a perfectly identical drape shape but instead to find parameters that show similar drapability, this level of accuracy can be acceptable.

6.2 *User Study*

For the qualitative evaluation, ten experts who have participated in the preliminary user study

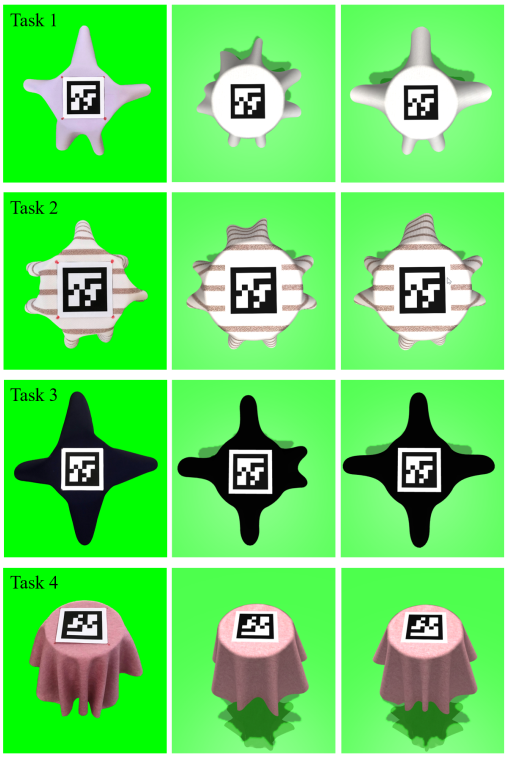
Table 3: The target fabrics for the user study

Task	Fabric type	Composition
1	Double Knit/Interlock	Cotton (39%), Lyocell (7%), Polyester (51%), Polyurethane (3%)
2	Double Knit/Interlock	Polyester (83%), Nylon (5%), Metallic (12%)
3	Dobby/Jacquard	Polyester (92%), Polyurethane (8%)
4	Rib	TENCEL™ Lyocell (66%), Wool (28%), Polyurethane (6%)

were asked to use and evaluate our UI. They were asked to perform parameter tuning tasks to find the optimal parameters for given read fabric drapes. For each task, the photographs of the target drape were presented, which had been taken from three different angles. And the initial values obtained from the measuring device designed for CLO3D 5.1 [17] were provided. All participants understood the meaning of the initial value as experts, and the degree to which they have confidence in the initial values depended on individual experience. The judgment on the completion of each task was subjectively determined according to their own experience.

Each participant performed a total of six tasks, and the first two were practice steps, regardless of whether tuning was completed or not. During this process, they were allowed to freely use the interface as much as they wanted. For the remaining four real tasks, the time taken to complete was recorded, but we asked the participants not to be constrained by time and to work until satisfactory results were achieved. The four target fabric samples had different fabric types and compositions as listed in Table 3. In the real tasks, we fixed the density parameter according to the advice of experts. According to experts, since density can be measured relatively accurately compared to other physical properties, they do not change the density unless there is a special reason.

Figure 8 shows a selected result for each task: (a) photographs of the target drape, (b) the drapes with the initial values, and (c) the final drapes after the tuning task was completed. It can be seen that the shape of the drapes predicted by the initial values were remarkably dif-



(a) Target photo (b) Initial drape (c) Tuned drape

Figure 8: Examples of each task

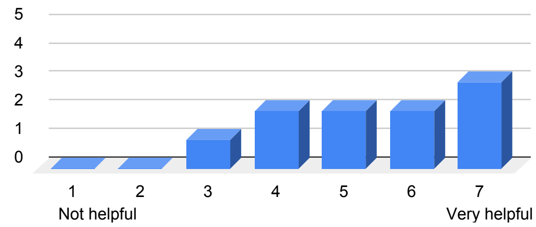
Table 4: Average completion time and standard deviations for each task.

Task	Average time (s)	Standard deviation					
		SU	SV	SH	BU	BV	BD
1	150.5	0.32	0.40	0.28	0.40	0.38	0.31
2	285.3	0.24	0.22	0.25	0.23	0.37	0.22
3	132.8	0.65	0.57	0.67	0.30	0.36	0.29
4	176.9	0.29	0.32	0.31	0.13	0.16	0.14

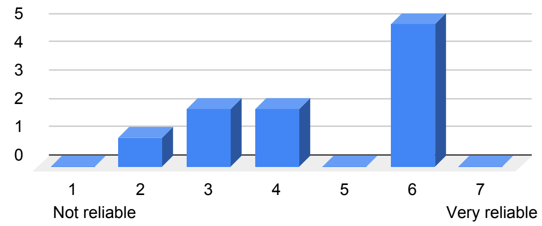
ferent from the target drapes. After the tuning tasks were completed, the results approached close to the targets. Table 4 shows the average completion time for each task and the standard deviations of log-scaled values chosen by participants for each parameter. The low standard deviations indicate that different participants made similar results. This means that we can expect consistent results with our method. The average completion time for each task was between 2 and 5 minutes. For the specific details and results of each task, please refer to the supplemental materials.

After completing the task, we asked each participant a qualitative evaluation about the UI and to answer following two questions:

Q1: How helpful do you think a UI like this can



(a) Q1. Helpfulness to understand the relationship.



(b) Q2. Reliability of the draping method.

Figure 9: Results of preliminary user study.

be for understanding the relationship between parameters and drape shapes?

Q2: How reliable do you think it is to do parameter tuning using the presented draping method?

Figure 9 shows the results. Their common opinion was that the interactivity of the UI allows users observe the continuous change of draping results with respect to the change of the parameters, which can be a great help in understanding the relationship between parameters and drape shapes. One participant also mentioned that he could see countless different draping results with different parameters in a short time, and that it was the most differentiated experience from the traditional way. On the other hand, several participants concerned that checking just one draping shape might not be reliable. In practice, more than one draping shapes of the target fabric are sometimes considered for better accuracy in the tuning. It is possible to add additional draping shapes in our UI. If our method is put to practical use, it will be an important consideration to use multiple draping methods.

In addition, participants were asked to give a descriptive evaluation of the UI. The most mentioned positive aspect was that they could see the draping result right away without waiting for the simulation. Most of the participants wanted such a high interactive UI to be included in the actual virtual fashion design software to improve the efficiency of tuning tasks. On the other hand, most participants noted that using only one draping method was a major drawback. In practice, tuning is often done with a single draping method, but in some cases, with additional samples in the form of t-shirts or skirts to cover more general situations. It is not difficult to add one or more draping methods in our method. If our method is put to practical use, it will be an important consideration to use multiple draping methods.

7 Discussion

In this study, the relationship between the simulation parameters and the final result of the draping simulation was learned from a large amount of training data. Using this, the user could interactively explore the change of the drape according to the change of simulation parameters. We also developed the UI for simulation parameter tuning, and it was evaluated for its usefulness by experts.

As they pointed out, using two different draping methods would be more helpful to users. In theory, adding more draping methods to our system is not difficult. However, in practice, because generating training data for a new draping method will incur high cost, verifying various draping methods is not a simple problem. In our experiments, we used hundreds of CPUs and completed 100,000 draping simulations over about two days. We determined that one way to reduce the cost is to optimize the distribution of the training data and the neural network structure, which makes for interesting future research work.

In terms of the draping method, it is necessary to develop one that is highly expressive and applicable to more diverse types of fabric. In our experiments, we considered only fabrics that, in their natural state, would not wrinkle due to their internal forces. For example, cursed and curled fabrics are excluded. Developing a suitable draping method to express the drapability of these fabrics will broaden the scope of application of our proposed method.

Our training model is designed to use an interactive UI. However, it can also be used to speed up autonomous optimization techniques. Because existing optimization techniques run simulations for every evaluation of the objective function, it takes from a few to tens of hours to optimize the simulation parameter for one target fabric. Replacing simulations with predictions from the training model greatly speeds up the optimization process. This will be also an exciting prospect for future research work.

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2021R1F1A1048002) and also supported by the Research Fund, 2020 of The Catholic University of Korea.

References

[1] Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter W Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations (ICLR)*, 2021.

[2] Sueo Kawabata. The standardization and analysis of hand evaluation. In *Effect of mechanical and physical properties on fabric hand*, pages 389–442. Elsevier, 2005.

[3] Sandra Kuijpers, Christiane Luible-Bär, and Hugh Gong. The measurement of fabric properties for virtual simulation—a critical review. *IEEE SA INDUSTRY CONNECTIONS*, pages 1–43, 2020.

[4] Kiran S Bhat, Christopher D Twigg, Jessica K Hodgins, Pradeep K Khosla, Zoran Popović, and Steven M Seitz. Estimating cloth simulation parameters from video. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 37–51, 2003.

[5] Huamin Wang, James F O'Brien, and Ravi Ramamoorthi. Data-driven elastic models for cloth: modeling and measurement. *ACM transactions on graphics (TOG)*, 30(4):1–12, 2011.

[6] Eder Miguel, Derek Bradley, Bernhard Thomaszewski, Bernd Bickel, Wojciech Matusik, Miguel A Otaduy, and Steve Marschner. Data-driven estimation of cloth simulation models. In *Computer Graphics Forum*, volume 31, pages 519–528. Wiley Online Library, 2012.

[7] Shan Yang, Zherong Pan, Tanya Amert, Ke Wang, Licheng Yu, Tamara Berg, and Ming C Lin. Physics-inspired garment recovery from a single-view image. *ACM Transactions on Graphics (TOG)*, 37(5):1–14, 2018.

[8] Katherine L Bouman, Bei Xiao, Peter Battaglia, and William T Freeman. Estimating the material properties of fabric from video. In *Proceedings of the IEEE international conference on computer vision*, pages 1984–1991, 2013.

[9] Shan Yang, Junbang Liang, and Ming C Lin. Learning-based cloth material recovery from video. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4383–4393, 2017.

[10] Wenyan Bi, Peiran Jin, Hendrikje Nienborg, and Bei Xiao. Estimating mechanical properties of cloth from videos using dense motion trajectories: Human psychophysics and machine learning. *Journal of vision*, 18(5):12–12, 2018.

[11] Eunjung Ju and Myung Geol Choi. Estimating cloth simulation parameters from a static drape using neural networks. *IEEE Access*, 8:195113–195121, 2020.

[12] G. E. Cusick. The dependence of fabric drape on bending and shear stiffness. *Journal of the Textile Institute Transactions*, 56(11):T596–T606, 1965.

[13] Enric Carrera-Gallissà, X Capdevila, and J Valldeperas. Evaluating drape shape in woven fabrics. *The Journal of The Textile Institute*, 108(3):325–336, 2017.

[14] Narahari Kenkare, Traci AM Lamar, Pradeep Pandurangan, and Jeffrey Eischen. Enhancing accuracy of drape simulation. part i: Investigation of drape variability via 3d scanning. *Journal of the Textile Institute*, 99(3):211–218, 2008.

[15] Viera Glombikova and KUS Zdenek. Drape evaluation by the 3d drape scanner. *Tekstil ve Konfeksiyon*, 24(3):279–285, 2014.

[16] Azmat Hussain, Yueqi Zhong, Tayyab Naveed, Zhicai Yu, Zhang Xi, and Wu Ge. A new approach to evaluate fabric hand based on three-dimensional drape model. *Autex Research Journal*, 20(2):155–167, 2020.

[17] CLO Virtual Fashion Inc. Fabric kit manual. `https://support.clo3d.com/hc/en-us/articles/360041074334-Fabric-Kit-Manual`, January 2020.

[18] David Baraff and Andrew Witkin. Large steps in cloth simulation. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 43–54, 1998.

[19] Sergio Garrido-Jurado, Rafael Muñoz-Salinas, Francisco José Madrid-Cuevas, and Manuel Jesús Marín-Jiménez. Automatic generation and detection of highly reliable fiducial markers under occlusion. *Pattern Recognition*, 47(6):2280–2292, 2014.

[20] CLO Virtual Fashion Inc. CLO. `https://www.clo3d.com/`, 2009–2021.