Sketch-based folded surface generating method with local remeshing and physical simulation technique

Yufei Zheng[1,a)] Suguru Saito[1,b)]

概要：This paper introduces an interactive method to creat folds with local remeshing and physical simulation methods. The user directly sketches on the surface of a cloth 3D model. A one-shot image of different kinds of folds will be generated in a short time by using our method, even for one who has little or no knowledge of the cloth apparel cutting structure and the physical properties of the fabric. The remeshing method to reconstruct the mesh structure of the folded surface locally while hardly changing the other parts of the cloth 3D model. To make the result more natural, we introduce a basic but comprehensive physical simulation system used for adjusting.

1. Introduction

In the process of character illustrations, cloth has always been an irreplaceable important part, among which the importance of the expression of the folds on cloth is more obvious. However, the expression of folds is influenced by many different factors, such as external forces including support force, wind force, and gravity, material of clothing, lighting condition and so on, which all have a great influence on the position and shape of folds in character illustrations.

In order to make it easy for more users to grasp the expressions of folds in various situations, we introduced a simple sketch-based modeling method[1] to provide users with 2D images of different styles of folds with different shapes and softness. This method provides with a very friendly user interface. The user only needs to draw some simple curves representing the position an direction of folds on the input cloth model to obtain the folded surface generated by it in real time, which can also be modified through some refinement methods including physical simulation, smoothing, and remeshing.

However, in our previous work, the shape and direction of the folds generated based on user input curves are not controllable and lack of customized options in the step of physical simulation. Therefore, we improve it by adding more customized options for physical simulation, and introducing local remeshing technology based on triangular mesh to realize the adjustment of fold's shape and the deletion of redundant folds.

2. Previous works

In this section, we will over view our previous work that allow the user to make the 2D image of folded surface using sketch-based method.

In fact, currently most of the systems for interactively generating folds are focusing on generating 3D models rather than just getting a single 2D cloth image. Such systems have a high demand for user expertise, and there are also many limitations on the location and shape of the generated folds. Like the systems [3], [5], [6] allow the user obtain a whole 3D cloth model by their input sketch. However, due to the need to ensure the sense of reality of the models obtained, the types of cloth and fold generated by such methods are very limited, and most of them lack subsequent customization options.

The method in [7] using a similar idea of the user input as our method, which allows the user directly draw the curves on the prepared cloth model, then get the 3D model of folded surface after physical simulation. But this method require the user with high expertise of the sewing cutting patterns of cloth, and the location, direction and shape of the folds are limited in a degree. The folds that

1 Tokyo Institute of Technology
a) yufei@img.cs.titech.ac.jp
b) suguru@c.titech.ac.jp

can be generated by [7] must located alonging the sewing lines of the cloth, and the direction of these curves has a strict limit. Because our goal is simply to get a single 2D folds reference image, our system has no particular limitations on the position and shape of the input curves. Some machine learning based method like [8] only can generate the model of cloth that the style is already existing in the database.

The basic idea of our system is based on [2], a sketch-based method which is focusing on making realistic cloth images. He provides a simple and practical idea for generating single folds reference 2D images which is easy to learn and use. Users can preprocess the input clothing model through the physical simulation system based on a Provot' s mass-spring system[4], which is modified in this papser, then directly draw curves on the cloth model to generate folded surface, and finally modify the folds through smoothing method and the physical simulation system again. Different from the the related works that focusing on generating cloth 3D model, Shi' s work only focuses on obtaining a single folds 2D image instead of a 3D model, so the height of the folds is not controllable, and the generated folded surface model will become unreal once rotated.

In response to the problems in [2], we propose an solution in [1]. In this improved version, we introduce the global remeshing operation in[10] that optimizes the mesh structure of clothing before the first physical simulation, which improves the quality of the results of physical simulation and mesh reconstruction based on user input curves to generate the folded surface, and the stability of the generated folded surface during physical simulation is also improved.

Based on [1], we will introduce the local remeshing method and new physical simulation condition of wind system and spring parameter's adjustment tools. With these new added customization options, the user has more freedom of their operation and get the desired folds 2D image.

3. Method and algorithm

In this chapter, we will introduce the two system upgrades, including the upgrade of the physical simulation system and the local remeshing method for processing user input curves to generate folded surfaces.

3.1 Physical simulation system improvement

In our previous work in [1], the user could not make any adjustments to the parameters of the physical simulation system, and the system only provided the external force of gravity and friction. Based on the same mass-spring structure system based on [2] and [4] with Verlet integral method[9], we have opened up the adjustment of parameters of the spring sturcture on the cloth model to accommodate of conditions with different types of fabric, and introduced a wind system that provides different options for preparing the initial state of cloth model.

3.1.1 Spring parameter's adjustment

In this improved method, we allow the user to modify the value of k-value and stretch rate from the spring-related parameters to change the fabric hardness and softness. K-value relates to the load on a spring when it is compressed. The larger the k-value, the less likely the spring to be deformed. Let the value of stretch rate be s, ff the spring overstretches too much during the physical simulation that its total length is greater than s times the original length, according to Provot's algorithm[4], the spring's two ends will be retracted in the opposite direction of the extension to avoid the overstretching of cloth model. Therefore when the k-value become bigger and the stretch rate become smaller, spring structures will be harder to deform, which means the cloth will have fewer folds, and the folded surface generated after physical simulation will look stiffer. We provide a friendly user interface which is shown in Fig. **??**, the user can easily change the parameters by dragging the slider, and can also select to use the default value we provided. We can see the results of the physical simulation under different parameter conditions in Fig. 1. Fig. 1(a) is the original cloth model, Fig. 1(b), Fig. 1(c) and Fig. 1(d) are the physical simulation results based on Fig. 1(a). Fig. 1(b) is the result using default spring parameters. Fig. 1(c) shows a result of harder cloth that the shape is rarely changed compared with Fig. 1(a), and Fig. 1(d) shows a result of softer cloth that will generate more folds than Fig. 1(b) and Fig. 1(c).

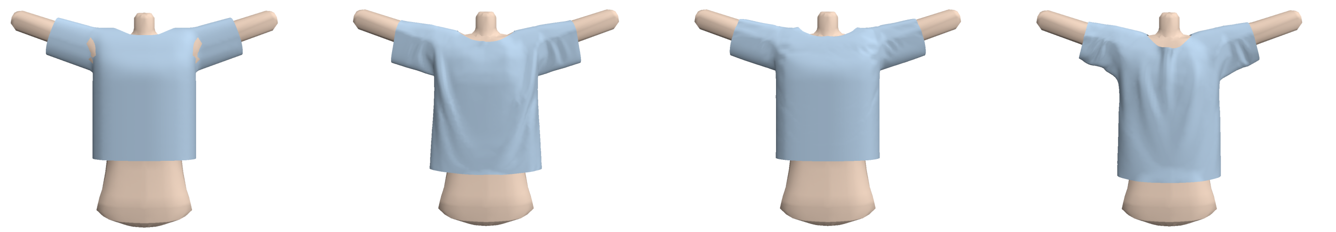
3.1.2 Wind system

In order to ensure operating efficiency, we do not use a real wind simulation calculated by fluid dynamics, but introduced a wind system based on [16], which can also produce a similar effect under wind force without using fluid dynamics.

The user can obtain different physical simulation results by adjusting the strength and direction of the wind.

3.2 Local remeshing

In this section , we will introduce two different local



(a) Original cloth model (b) Physical simulation result with default spring parameters (c) Physical simulation result with bigger k-value and smaller stretch rate (d) Physical simulation result with smaller k-value and bigger stretch rate

図 1 The comparison of the results of physical simulation under different spring parameter situations.

remeshing methods that refer to the fold eraser and fold shape customization.

3.2.1 Fold eraser

In our previous system, the user is only allowed to generate folds by sketching, but not able to remove the extra folds that generated either by user's choice or directly through the physical simulation system. Therefore, in this system upgrade, we introduce the operation of folded surface eraser. Users can use this eraser to remove the folds in the selected area by locally remeshing.

At the beginning, the user need to select an area where the folds need to be removed. Then the system will delete all the verteces and triangles in the area, obtain the boundary information in a certain order of the holes for hole filling operation on the 3D model, and finally refine the filled area through remeshing and Taubin smoothing[11] operation.

There are several hole filling algorithm for triangular mesh, like [12], [13], [14] and [15]. Most of these algorithms aim at large area holes with a large number of boundary verteces and are often used to optimize surface generated from point cloud, while in our system, the holes tend to be small in size and number. Therefore, we chose the traditional wavefront method to fill the holes.

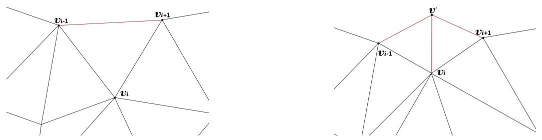
We simplify the algorithm in [15] and implement a recursive algorithm to iteratively repair the hole by finding the minimum internal angle generated by the adjacent boundary on the edge of current hole. The detailed steps of the fold eraser are shown in Algorithm 1.

Algorithm1 Local remeshing for fold eraser

Input: The original triangular mesh of the cloth model, selected verteces list V_{select} and selected triangles T_{select} inside the user selected area.

Output: Reconstructed triangular mesh;

Find the list of boundary points V_{edge} : $\boldsymbol{v_1}, \boldsymbol{v_2}, \ldots, \boldsymbol{v_n}$ on the edge of the selected area and sort them int counterclockwise using V_{select} and T_{select};
while The size of V_{edge} is bigger than 2 **do**
  Init a current smallest angle a_s as π;
  Init the index of the vertex corresponding to the smallest angle as k;
  for $\boldsymbol{v_i}, (i = 1, 2, \ldots, n)$ **do**
    Calculate the inner angle $a_{current}$ of $\angle \boldsymbol{v_{i-1}} \boldsymbol{v_i} \boldsymbol{v_{i+1}}$;
    if $a_{current} < a_s$ **then**
      $a_s = a_{current}$;
      $k = i$
    end if
  end for
  if $0 < a_s < \frac{\pi}{2}$ **then**
    Connect $\boldsymbol{v_{k-1}}, \boldsymbol{v_k}, \boldsymbol{v_{k+1}}$;
    Remove $\boldsymbol{v_k}$ in V_{edge};
  else
    Add a new vertex $\boldsymbol{v_{n+1}}$, where n is the current size of the verteces of the cloth model;
    Connect $\boldsymbol{v_{n+1}}, \boldsymbol{v_{k-1}}, \boldsymbol{v_k}$ and $\boldsymbol{v_{n+1}}, \boldsymbol{v_k}, \boldsymbol{v_{k+1}}$, where $\boldsymbol{v_{k-1}}$ is on the angle bisector of $\angle \boldsymbol{v_{k-1}}, \boldsymbol{v_i}, \boldsymbol{v_{k+1}}$, and the length of $|\boldsymbol{v_k} - \boldsymbol{v_{n+1}}|$ is the average of the lengths of $|\boldsymbol{v_k} - \boldsymbol{v_{k-1}}|$ and $|\boldsymbol{v_k} - \boldsymbol{v_{k+1}}|$; and the length of $|\boldsymbol{v_k} - \boldsymbol{v_{n+1}}|$ is the average of the lengths of $|\boldsymbol{v_k} - \boldsymbol{v_{k-1}}|$ and $|\boldsymbol{v_k} - \boldsymbol{v_{k+1}}|$;
    Replace $\boldsymbol{v_k}$ with $\boldsymbol{v_{n+1}}$ in V_{edge};
  end if
end while
Rebuilding the triangular mesh based on the new verteces and edges;
Do the remeshing operation twice time on the new mesh created by filling the hole;
Do the Taubin smoothing operation one time on the



(a) Join existing vertices (b) Add new vertices and join existing vertices

図 2 A simple example about the rule for adding triangles in each iteration.

new mesh created by filling the hole;
return Reconstructed triangular mesh;
end

The rule for adding triangles in each iteration is illustrated in Fig. 2. Fig. 2(a) shows that when $0 < \angle \boldsymbol{v_{i-1}v_iv_{i+1}} < \frac{2}{\pi}$, the system will connect $\boldsymbol{v_{i-1}}$ and $\boldsymbol{v_{i+1}}$ to get a new triangle. Fig. 2(b) shows that when $\angle \frac{2}{\pi} \leq \boldsymbol{v_{i-1}v_iv_{i+1}} \leq \pi$, a new vertex $\boldsymbol{v'}$ will be added on the angular bisector of $\angle \boldsymbol{v_{i-1}v_iv_{i+1}}$. The length of $\boldsymbol{v_iv'}$ is the average of the lengths of $\boldsymbol{v_iv_{i+1}}$ and $\boldsymbol{v_{i-1}v_i}$. The red lines in Fig. 2(a) and Fig. 2(b) are the new added lines. The example and result of using fold eraser to remove the folds generated by user input curves or physical simulation is shown in Fig. 3 and Fig. 4. Fig. 3(a) is the original cloth model. The red curve on Fig. 3(b) is the user input curve on Fig. 3(a), and Fig. 3(c) is the generated folded surface based on the curve in Fig. 3(b). The red area in Fig. 3(d) represents the user selected area for removing folds. Fig. 3(c) is the result that directly filling the hole in the selected red area in Fig. 3(d), and Fig. 3(f) is the local remeshing and Taubin smooting result based on Fig. 3(e). Fig. 4(a) is the back of the physical simulation result of a woman's shirt. The red area in Fig. 4(b) is the user selected area for removing folds. Fig. 4(c) is the result after hole filling, and Fig. 4(d) is the local smoothing and remeshing result based on Fig. 4(c). After the hole filling, the folds in the area selected by the user will be erased in a degree, and the result will be improved after remeshing and smoothing operation.

3.2.2 Fold shape customization

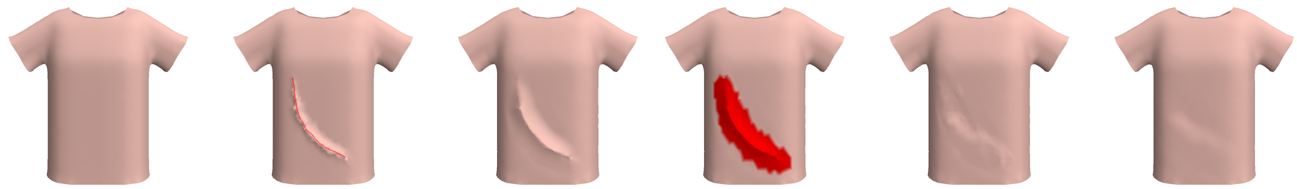
Our previous method in [1] can only generate one triangular shape folded surface, and there will have some problems when the user wants to generate different styles of folds. We will introduce a detailed local remeshing on the original surface reconstruction results. The process can be seen in Fig. 5. The user can change the relative position of the intermediate points shown in Fig. 5(f). The red curve in Fig. 5(a) is the user input curve, and the red points in Fig. 5(b) is the selected points based on the red curve in Fig. 5(a). Fig. 5(c) shows the original mesh reconstruction result, the blue lines are the new connected lines, and the blue points are the first neighbour points of the red points. Fig. 5(d) is the schematic of lifting the red points in Fig. 5(c) and Fig. 5(e) is the result after lifting. The green points in Fig. 5(f) are the new added intermediate points which are on the line between a red point and a blue point.

At present, we add two intermediate points on each segment connecting the lifting points and the corresponding adjacent points. By changing the control points that show the relative position of these intermediate points, we can obtain folds with different styles in the cross section, as shown in Fig. 6. Some examples of the different position of control points is shown in Fig. 7. The green points in Fig. 7 are the control points, red point is on the user input curve and the blue point is the first neighbour point of the red point. Fig. 7(a) shows the original position of the control points, which is like a triangle. Fig. 7(b) shows the position of control points that can make a round folds like in Fig. 6(d) , Fig. 7(c) shows the position for making slim folds like in Fig. 6(c). The detailed steps from generating Fig. 6(f) from Fig. 6(e) is in Algorithm 2. Fig. 6(a) shows the result of generated folded surface without adding intermediate points, and Fig. 6(b) is the result after adding intermediate points on Fig. 6(a). Fig. 6(c) and Fig. 6(d) are the results of moving the intermediate points on Fig. 6(b). Fig. 6(c) shows a thin fold and Fig. 6(d) shows a rounded fold. Fig. 8 shows the results of folded surface with two different kinds of fold's hardness or softness. Fig. 8(a) uses the same mesh reconstructure rule as Fig. 6(c), Fig. 8(b) uses the same mesh reconstructure rule as Fig. 6(d).

Algorithm2 Local remeshing for generating customization folded surface

Input: Triangular mesh of the cloth model after first step mesh reconstruction, Lifting verteces list V_{lift}, the list V_{origin} that stores the initial coordinates of the verteces in list V_{lift} a before lifted, list V_{around} which contains all the first neighbour verteces around lifting verteces, the number n_c of control points to insert, an array r_{2,n_c} to store the relative coordinates for determine the control point's position;

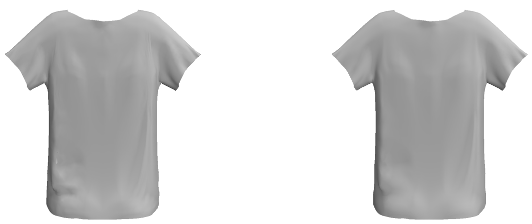
Output: Reconstructed triangular mesh;



(a) Original cloth model (b) User input curve (c) Generated folded surface (d) Area for eraser (e) Original hole filling result (f) Hole filling result after local remeshing and Taubin smoothing

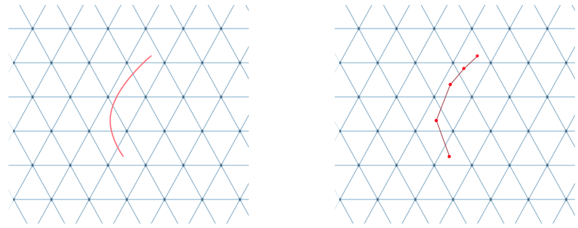
図 **3** The example and result of fold eraser.

(a) Original physical simulation result (b) User selected area for removing folds

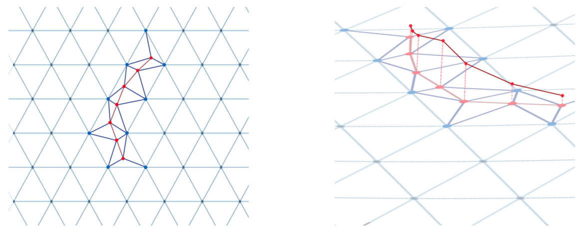


(c) Original hold filling result (d) Hole filling result after local remeshing and Taubin smoothing

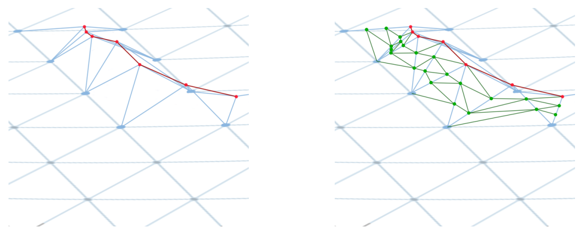
図 **4** The example and result for removing the folds directly generated after physical simulation.



(a) User input curve (b) Selected points based on user input curve



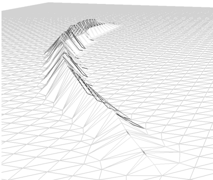
(c) Reconstruction result (d) Lifting the selected points



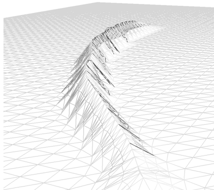
(e) Reconstruction result after lifting (f) Adding intermediate points on the relevant edges and making new triangles

図 **5** The process of current mesh reconstruction based on user input.

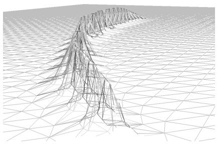
Create a list $V_{control}$ for storing the data of all the new added verteces on the edited edges;
for $\boldsymbol{v_i}, (i = 1, 2, \ldots, n), \boldsymbol{v_i} \in V_{lift}$ **do**
  Get list $V_{neighbour_i}$ of all the adjacent verteces of $\boldsymbol{v_i}$;
  for $\boldsymbol{v_j}, (j = 1, 2, \ldots, m), \boldsymbol{v_j} \in V_{neighbour_i}$ **do**
    if $\boldsymbol{v_j} \notin V_{lift}$ **then**
      for k from 1 to n_c **do**
        $\boldsymbol{v_{ij_k}} = r_{1,k}(\boldsymbol{v_i} - \boldsymbol{v_{i_{origin}}}) + r_{2,k}(\boldsymbol{v_j} - \boldsymbol{v_{j_{origin}}})$;
        Put $\boldsymbol{v_{ij_k}}$ into $V_{control}$;
      end for
    end if
  end for
end for
for $\boldsymbol{v_i}, (i = 1, 2, \ldots, n), \boldsymbol{v_i} \in V_{lift}$ **do**
  Get list $V_{neighbour_i}$ of all the adjacent verteces of $\boldsymbol{v_i}$;
  for All the pair of j_1, j_2 that $\boldsymbol{v_{j_1}}, \boldsymbol{v_{j_2}}$ is connected, $\boldsymbol{v_{j_1}}, \boldsymbol{v_{j_2}} \in V_{neighbour_i}$ **do**
    Connect $\boldsymbol{v_i}$, $\boldsymbol{v_{ij_{1_1}}}$ and $\boldsymbol{v_{ij_{2_1}}}$;
    Connect $\boldsymbol{v_{j_1}}$, $\boldsymbol{v_{j_2}}$ and $\boldsymbol{v_{ij_{1_{n_c-1}}}}$;
    Connect $\boldsymbol{v_{j_1}}$, $\boldsymbol{v_{j_2}}$ and $\boldsymbol{v_{ij_{2_{n_c-1}}}}$;
    for k from 1 to $n_c - 1$ **do**
      Connect $\boldsymbol{v_{ij_{1_k}}}$, $\boldsymbol{v_{ij_{1_{k+1}}}}$ and $\boldsymbol{v_{ij_{2_{k+1}}}}$;
      Connect $\boldsymbol{v_{ij_{2_k}}}$, $\boldsymbol{v_{ij_{1_{k+1}}}}$ and $\boldsymbol{v_{ij_{2_{k+1}}}}$;



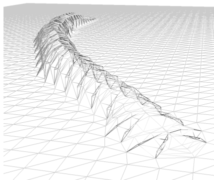
(a) Folded surface without adding intermediate points



(b) Folded surface with intermediate points

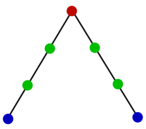


(c) Thin folded surface generated by moving intermediate points

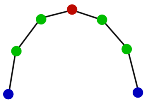


(d) Rounded folded surface generated by moving intermediate points

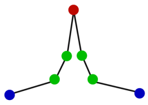
図 6 The comparison and example of different mesh reconstruction results on a same user input curve.



(a) Triangle shape

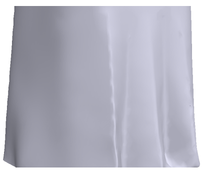


(b) Round shape



(c) Slim shape

図 7 Examples of control points.



(a) Hard folds

(b) Soft folds

図 8 Two different kinds of fold's hardness and softness made by fold's customization system.

```
        end for
    end for
end for
return Reconstructed triangular mesh;
end
```

4. Results

Our experiment environment is in Debian 9.9, with CPU: Intel(R) Core(TM) i9-9900K, memory: 32GiB, GPU: GeForce RTX 2060.

Fig. 9 shows the wind simulation results with folded surface generated by user input curves. Fig. 9(a) is the original physical simulation result of a cloth model of a woman's skirt in a windless environment. Fig. 9(b) shows the result of applying a wind from left to right. The red curves and blue curves in Fig. 9(c) represent the user input curves, which means to build concave or convex folded surface, and the smoothing and remeshing result is shown in Fig. 9(d).

Fig. 10 shows the physical simulation results of different folds generated by a same user input curve on the same woman shirt's 3D model with 11898 initial facet count and 6073 initial vertex count. Fig. 10(b) shows the physical simulation result based on Fig. 10(a) after smoothing. The fold in Fig. 10(a) is generated by our previous method without adding control points. The fold in Fig. 10(c) is based on Fig. 10(a) with control points, and the physical simulation result is in Fig. 10(d). The difference in structure can be obtained by comparing Fig.6(a) with Fig.6(b). Fig. 10(e) shows the rounded fold made by the same reconstruction rule in Fig.6(d). The corresponding physical simulation result is shown in Fig. 10(f). Fig. 10(g) shows the thin fold with the same mesh reconstruction rule as Fig.6(c), and its corresponding physical simulation result is shown in Fig. 10(h). The shape of the fold generated by our previous meshod and the thin fold are hard to be contained after physical simulation.

5. Conclusion and future work

We have introduced an improved sketch-based system for the user to generate folds interactively. In this improved system, we implement new local remeshing methods for fold eraser and fold's shape customization, and update more conditions for physical simulation.

At present, the newly added wind system is largely separated from the user input system. It is difficult to keep the shape of the folds generated based on user input after physical simulation with wind, and the customized folds with different shapes generated by user input is very unstable in the simulation process. In our future work, we will further improve the links in each parts in our system, enhance the stability of the system, and update the corresponding user interface.

参考文献

[1] Yufei Zheng, Hatsu Shi and Suguru Saito: Realistic Folded Surface Modeling from Sketching. Cyberworlds2019, pages 155–163(2019).

[2] Hatsu Shi, Suguru Saito,"An interactive method for adding wrinkles on cloth models to generate a convincing cloth CG",proceedings of the 81st National Convention

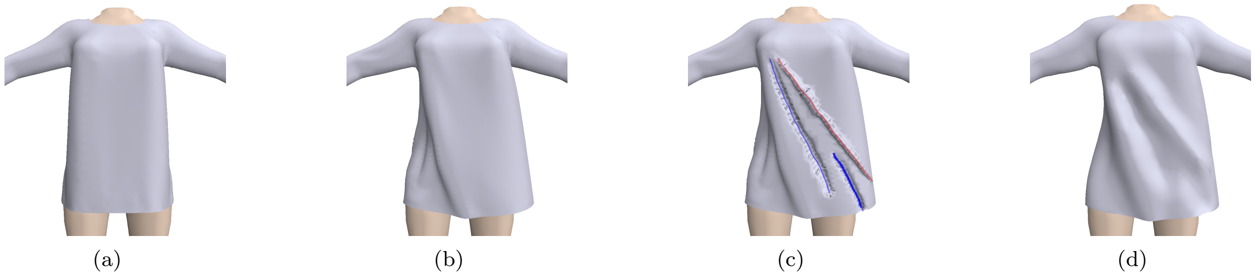


図 9　The wind simulation results with folded surface generated by user input curves.

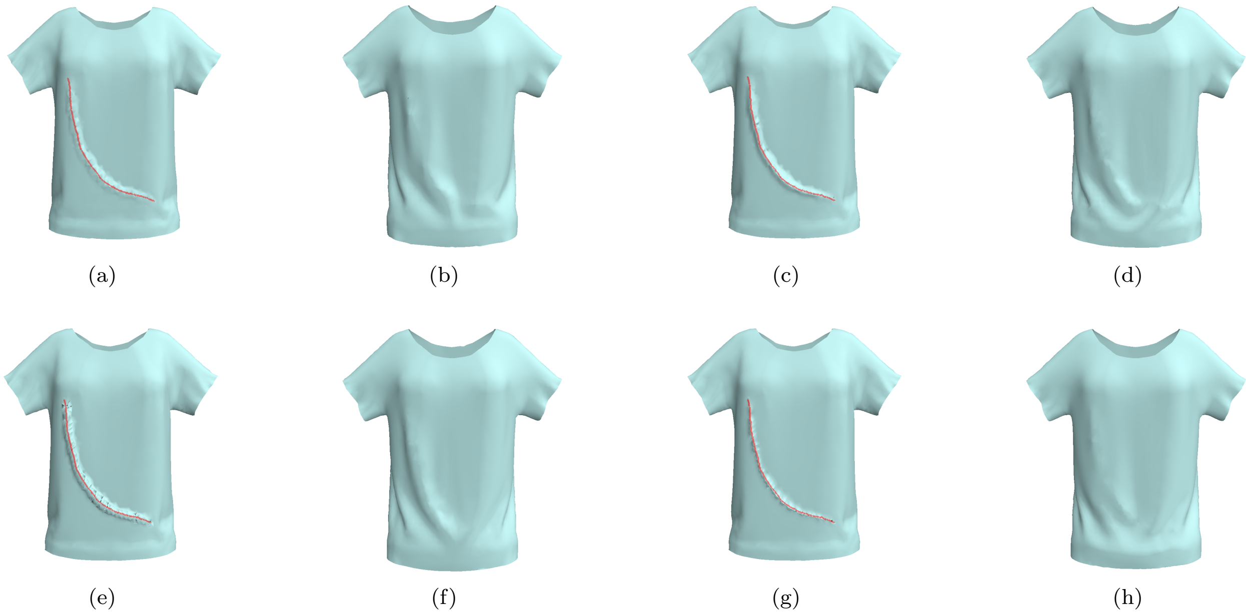


図 10　The physical simulation results of different types of folds on the same cloth model and same user input curve. The red curve in (a), (c), (e), (g) is the same user input curve.

of IPSJ, 2ZC-03, 2 pages, IPSJ, 2019.

[3] Takeo Igarashi, Satoshi Matsuoka, and Hidehiko Tanaka: Teddy: A sketching interface for 3d freeform design. pages 409–416, 1999.

[4] Xavier Provot: Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In Proceedings of Graphics Interface ' 95, pages 147–154, 1995.

[5] Takeo Igarashi and John F. Hughes: cloth manipulation. In Proc. UIST" 02, pages 91–100, 2002.

[6] Changjian Li, Hao Pan, Yang Liu, Xin Tong, Alla Sheffer, and Wenping Wang. Bendsketch: Modeling freeform surfaces through 2d sketching. ACMTrans. Graph., 36(4):125:1–125:14, July 2017.

[7] Minchen Li, Alla Sheffer, Eitan Grinspun, and Nicholas Vining. Foldsketch: Enriching garments with physically reproducible folds. ACM Transaction on Graphics, 37(4), 2018.

[8] Tuanfeng Y. Wang, Duygu Ceylan, Jovan Popovi ´ c, and Niloy J. Mitra. Learning a shared shape space for multimodal garment design. In SIGGRAPH Asia 2018 Technical Papers, SIGGRAPH Asia ' 18, pages 203:1–203:13, New York, NY, USA, 2018. ACM.

[9] Loup Verlet. Computer " experiments" on classical fluids. i. thermodynamical properties of lennard-jones molecules. Phys. Rev., 159:98–103, Jul 1967.

[10] Botsch M, Kobbelt L, Pauly M, et al. Polygon mesh processing[M]. CRC press 2010.

[11] Gabriel Taubin. A signal processing approach to fair surface design. In Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ' 95, pages 351–358, New York, NY, USA, 1995. ACM.

[12] Davis J, Marschner S R, Garr M, et al. Filling holes in complex surfaces using volumetric diffusion. Proceedings. First International Symposium on 3D Data Processing Visualization and Transmission. IEEE, 2002: 428-441.

[13] Brunton A, Wuhrer S, Shu C, et al. Filling holes in triangular meshes by curve unfolding. 2009 IEEE International Conference on Shape Modeling and Applications. IEEE, 2009: 66-72.

[14] Hu P, Wang C, Li B, et al. Filling Holes in Triangular Meshes in Engineering[J]. JSW, 2012, 7(1): 141-148.

[15] S. Liu, J. Liang, B. Dong. Rapid Mesh Hole Filling in Body Model. Hsi-An Chiao Tung Ta Hsueh/Journal of Xi'an Jiaotong University 52(8):37-42, August 2018.

[16] Krikienoid. FlagWaver: A web app for simulating a waving flag. GitHub, GitHub repository, `https://github.com/krikienoid/flagwaver`, 2018.