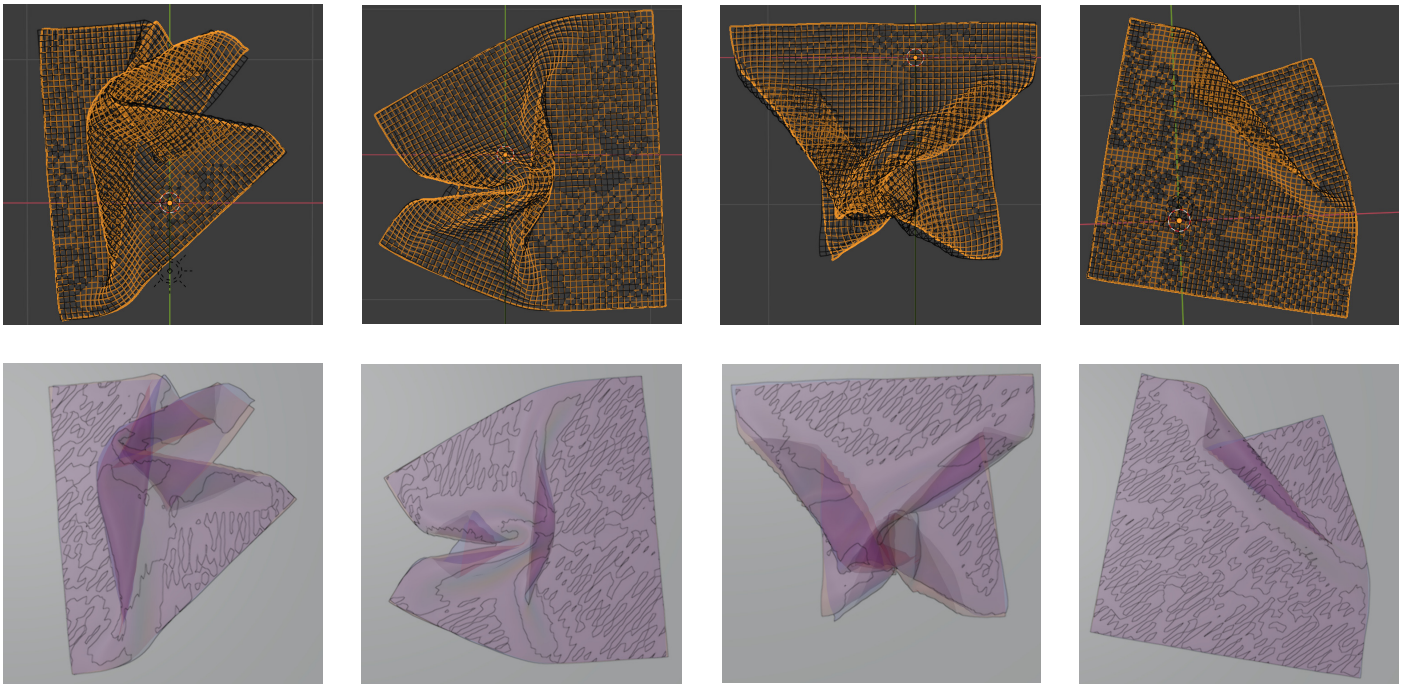
Learning of Geometry Estimation for Wrinkled Garments



Gen Li

Semester Thesis

Computational Robotics Lab

ETH Zürich

Supervisors:

Prof. Dr. Bernhard Thomaszewski

Prof. Dr. Stelian Coros

January 7, 2022

Eidgenössische Technische Hochschule Zürich

Swiss Federal Institute of Technology Zurich

Acknowledgements

I got a lot of valuable advice from Wenbo Wang about the simulation in the blender, including tuning the cloth parameters. I thank him for his dedication. I am grateful to Prof. Thomaszewski and Prof. Coros for their supervision. In my previous research experience related to deep learning, I rarely tried to interpret the network but stayed at the intuitive level. They always pushed me to find the closest mathematical meaning of the network, which made me think more deeply. Whilst deep learning is not interpretable most of the time, interpretability is an important area to focus on.

Abstract

We demonstrate the estimation of the geometry of wrinkled garments from single-view input depth images. Estimating the geometry of non-rigid objects is challenging because of the high-dimensional degree of freedom. Our approach is a vision-based, fully end-to-end Graph Attention Network. This choice has several advantages, including the ease of explicitly encoding geometric priors of the template mesh into the network, and more accurate and realistic predictions. From a high-level perspective, our network deforms the template mesh into the target geometry based on input depth images. In addition, our model can be generalized to visually different garments, since only depth images are used. We also demonstrate that our model can be trained to be robust to partial occlusions. We outperform the baseline without geometric priors quantitatively, while we also produce better predictions qualitatively.

Contents

Introduction

Chapter 1

Deformable object manipulation is a challenging task in robotics. There are many applications, such as assistive dressing [1], folding clothes [2], and unfolding clothes [3]. These algorithms must estimate the garment's geometry first before planning. However, estimating the geometry of deformable objects is generally difficult because: 1) deformable objects have a very high degree of freedom, and 2) partial observability is an intrinsic attribute of these objects.

In this work, we propose a vision-based model that enables a robot to estimate the geometry of a wrinkled garment in real-time. Previous learning-based solutions learn models in latent space and need additional depth sensors, which is not data-efficient [4]. Previous geometry-based approaches only detect key points rather than reconstruct the geometry completely [5], or need multi-viewing for reconstruction [6]. Handcrafted geometric features are often low-dimensional and don't generalize well [7]. In this work, we directly estimate the 3D mesh vertex positions of the cloth model using a single-view depth image. This method has several strengths. First, we can incorporate geometric priors of the cloth model into the network. As shown in the experiments, our model can generate more realistic predictions of the estimated cloth than models learned without geometric priors. Second, using depth images allows the model to learn an appearance-invariant representation. The learned model can be applied to visually different environments and garments. Then, the model is trained using simulated data in Blender [8], which eases the burden of collecting data in the real world. Finally, the explicit mesh representation makes it easier to extract accurate position information of the garment for subsequent planning algorithms.

The main challenge of this work is to predict a high dimensional mesh structure. Recently, Ma et al. [9] approximate a deformable object as a sparse set of key points, which only manipulate the object roughly. We solve this problem by incorporating geometric priors and using Graph Attention Network [10] to mimic the cloth model due to the rough equivalency between the graph neural network architecture and the spring-mass model [11]. Another challenge is that the cloth may be partially visible during the manipulation due to the occlusion of other objects, such as the robot arm. We overcome this problem by using the cutout technique [12] when training the neural network.

To summarize, the main contributions of this work are:

1. We propose a fully end-to-end deep neural network model for estimating the geometry of wrinkled garments from a single-view depth image with no handcrafted constraints. The predictions of the proposed model are more accurate and realistic compared to that of the baseline model.

2. We propose a large synthetic dataset containing a sample of 40,000 depth images of wrinkled garments with different complexity configurations, annotated with the corresponding 3D vertex positions.

Related Work

Chapter 2

Estimating the geometry of wrinkled garments is difficult due to the high-dimensional state space and self-occlusion of non-rigid objects. Many previous approaches do not reconstruct the entire geometry, but rather use key points to approximate or serve as a proxy [5, 9], which introduce more inaccuracies in subsequent manipulations. Many other works need to resort to additional sources of information such as attaching depth sensors to cloth [4], multi-viewing [6], and ungeneralizable handcrafted geometric features [7].

2.1 Reconstruction of the 3D shape

Reconstructing 3D shapes from monocular images has been a severely ill-posed but important problem in computer vision. Different assumptions must be introduced to solve this problem.

The traditional method of reconstructing 3D shapes is through an optimization-based approach such as superquadrics [13] and finite-elements [14]. Although the results are reliable due to physical constraints, it is very sensitive to initialization and quite time-consuming.

When multi-view or video data are available, for rigid objects, they can be uniquely recovered by rigid Structure-from-Motion (SfM) [15], with some assumptions on the camera projection function. However, when an object is deformable, its shape is not well constrained over time. The Non-Rigid SfM (NRSfM) problem requires various assumptions [16, 17].

Nonetheless, viewing objects from different angles during robot manipulation is time-consuming and often impractical. Therefore, estimating the geometry directly from one input image is of vital importance. In Shape-from-Template (SfT) [18] method, the 3D geometry of the reference template and the mapping of the 3D-to-2D points between the template and the image should be known. It is not robust to the appearance of objects and not applicable to deformable objects without rich details. Pumarola *et al.* proposed a geometry-aware network [19] in an end-to-end manner. It estimates a depth map from the input image. Although it improves state-of-the-art solutions with a lower computational time, it can only handle non-rigid objects with relatively flat surfaces. In the unfolding setting, wrinkled garments often have large self-occlusions. Moreover, in our work, we use depth images instead of RGB images as input to reduce the depth ambiguity of the problem.

Graph Network	*Analogy to Newtonian Mechanics*
Nodes	Particles
Pairs of nodes	Two interacting particles (i, j)
Edge model	Compute force F_{ij}
Pooling	Sum into net force $F_{net,i}$
Node model	Acceleration $a_i = F_{net,i}/m_i$
Updated nodes	Compute next timestep

Table 2.1: Explanatory comparison between GNN and Newtonian mechanics

Many other fields also face the problem of estimating 3D information from a single image. Take 3D human pose and shape estimation as an example, Kolotouros *et al.* [20] directly regressed the 3D position of mesh vertices using a Graph-CNN. The template mesh structure can be encoded within the network and the spatial locality helps to regularize the output of the network. Lin *et al.* [21] further improved the performance by incorporating a transformer encoder to jointly model vertex-vertex and vertex-joint interactions using its self-attention mechanism. The main idea of our work is inspired by these works: using the network to process image features on the mesh structure.

2.2 Graph Neural Network

Graph Neural Networks have been successfully applied to learning physical simulation recently. Interpretable graph networks become an important research topic.

Cranmer *et al.* [11] proposed a framework to extract symbolic representations of a learned graph neural network by introducing strong inductive biases. The regressed symbolic expressions can produce an overall algebraic equation equivalent to the trained Graph Neural Network. They provided a rough equivalency between GNN and Newtonian Mechanics in Table 2.1. They demonstrated that correct known equations, i.e. force laws, can be extracted from the model's message function. In summary, the Graph Neural Network is capable of modelling the spring-mass model in an interpretable way.

Pfaff *et al.* [22] proposed MeshGraphNets to learn mesh-based simulations using GNN. The network can predict the dynamics of various physical systems including cloth. Our work is inspired by this line of works. Instead of regressing the acceleration of particles at each time step, we directly regress the position of each particle of the cloth model.

Veličković *et al.* [10] proposed Graph Attention Network. The GAT has additional attention layers for nodes to attend over their neighbouring features, which allows (implicitly) assigning different weights to different nodes in the neighbourhood. GAT achieved or matched state-of-the-art performance in several benchmarks.

The method

Chapter 3

The brief architecture of our network is shown in Figure 3.1. The task is to estimate the geometry of a wrinkled garment using its depth image. A *Convolutional Neural Network* (CNN) extracts the depth image feature. The structure of this extractor is described in Section 3.1. In Section 3.2, we describe the core part of our network, the Graph Attention Network, which regresses the 3D vertex coordinates of the deformed mesh to reconstruct the wrinkled garment. Then, we describe model training details and methods to solve the occlusion problem in Section 3.3. Finally, the synthetic dataset we generated is described in Section 3.4.

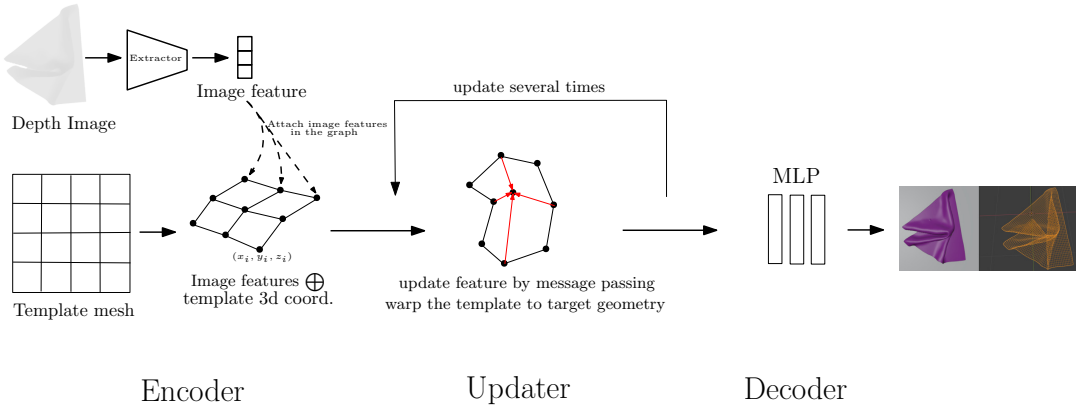


Figure 3.1: Overview of our cloth reconstruction system. Given an input depth image, a CNN encodes it as a low dimensional feature vector. The feature vector is then embedded into a graph defined by the template cloth mesh, which is attached to the 3D coordinates (x_i, y_i, z_i) of each vertex i. After several message passing updates, the geometry of the cloth is deformed in the latent space. Then we use a decoder to regress the 3D vertex coordinates of the deformed mesh.

3.1 Feature extractor

We choose a single-view depth image as input to minimize depth ambiguity while making the problem setup as simple as possible. In order to extract sufficient information from the depth image, we use a Residual Neural Network [23] to extract image features to ensure generalization capability, instead of handcrafting low-dimensional features.

The state-of-the-art CNN architecture is going deeper. AlexNet [24], VGG [25], and GoogleNet [26] has 5, 19, and 22 convolutional layers respectively. Stacking more layers should not degrade the network performance because the stacking layers can simply be identity mappings. However, deep networks are difficult to train due to the vanishing gradient problem: repeated multiplication in backpropagation will make the gradient infinitesimal. ResNet tackled this problem using the "shortcut connection" mechanism. He *et al.* [27] proposed a 1001-layer ResNet which improves generalization.

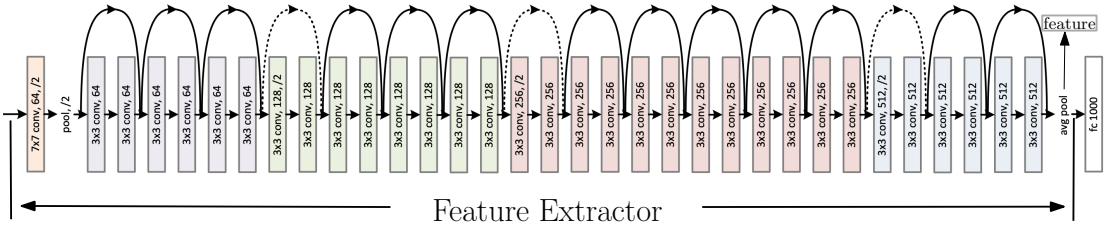


Figure 3.2: Image feature extractor architecture. The figure is adapted from the original ResNet paper [23]. The image feature vector is obtained before the last layer. All layers except the last fully connected layer act as the feature extractor. The obtained feature dimension is 512.

In our work, we modified the 34-layer ResNet as the generic image feature extractor. The architecture is shown in Figure 3.2. The extractor is not pre-trained and its parameters are initialized randomly.

3.2 Graph Attention Network

From [11, 22] and Table 2.1, we can see that the Graph Network has the ability to model the spring-mass model in an interpretable way, thus we propose to use a Graph Network to directly regress 3D vertex coordinates of the cloth mesh vertices. The architecture allows us to encode the mesh structure into the network explicitly, so the network can leverage geometric priors to produce more realistic predictions. First, we need to clarify some definitions of the network.

3.2.1 Definitions

Following the symbol system of [11], we use the template Graph $G = (V, E)$ to predict an updated Graph $G' = (V', E)$. The Graph $G = (V, E)$ consists of N^v nodes with L^v features each: $V = \{\mathbf{v}_i\}_{i=1:N^v}$, with each $\mathbf{v}_i \in \mathbb{R}^{L^v}$. There are N^e edges in Graph G: $E = \{(r_k, s_k)\}_{k=1:N^e}$, where $r_k, s_k \in \{1 : N^v\}$ represent receiving and sending node indices respectively. Each edge has L^e features: $\mathbf{e}_k \in \mathbb{R}^{L^e} (k = 1 : N^e)$. In the updated Graph G', $V' = \{\mathbf{v}'_i\}_{i=1:N^v}$, where $\mathbf{v}'_i$ is the updated node feature corresponding to $\mathbf{v}_i$, and $\mathbf{e}'_k$ is the updated edge feature corresponding to $\mathbf{e}_k$.

Edge function. The edge function (message function) is used to compute messages from one node to another for every edge: $\phi_e : \mathbb{R}^{L^v} \times \mathbb{R}^{L^v} \rightarrow \mathbb{R}^{L^{e'}}$, where $L^{e'}$ is the number of message features. For all edges indexed by k, the edge feature is calculated as: $\mathbf{e}'_k = \phi_e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k}) := \mathcal{NN}_e(\mathbf{e}_k, \mathbf{v}_{r_k}, \mathbf{v}_{s_k})$. The edge function models the pairwise interaction between two nodes.

Message pooling. For every receiving node i, neighbouring messages are pooled via element-wise weighted summation: $\mathbf{m}_{\mathcal{N}(i)} = \bar{\mathbf{e}}'_i = \sum_{k \in \{1:N^e | r_k = i\}} \mathbf{w}_{ik} \mathbf{e}'_k$, where $\mathbf{w}_{ik}$ is the weight of neighbouring node in the summation. The weights are learned by the network.

Node function. The node function is used to predict the output node $\mathbf{v}'_i$ from pooled messages for every node: $\phi_v : \mathbb{R}^{L^v} \times \mathbb{R}^{L^{e'}} \rightarrow \mathbb{R}^{L^{v'}}$. The node feature is calculated as: $\mathbf{v}'_i = \phi_v(\bar{\mathbf{e}}'_i, \mathbf{v}_i) := \mathcal{NN}_v(\bar{\mathbf{e}}'_i, \mathbf{v}_i)$.

3.2.2 Encoder

First, we transform the input into a graph. We can obtain the 3D coordinates of the template cloth mesh from a flat garment in Blender [8]. The encoder encodes the template mesh into Graph $G = (V, E)$. Mesh vertices are encoded into graph nodes V and mesh edges are encoded into bidirectional graph edges E. The encoded template mesh serves as the geometric prior and offers the spatial locality for the network.

As in Figure 3.1, first, we attach the extracted 512-D image feature to 3D coordinates of vertices in the template mesh. The image feature serves as goal information for deforming the template mesh. For every vertex $u_i = (x_i, y_i, z_i)$, its 3D coordinates and per-vertex image feature f are mapped into latent space as the node feature $\mathbf{v}_i$ using an MLP:

$$\mathbf{v}_i = \mathcal{MLP}([x_i, y_i, z_i, f]) \tag{3.1}$$

To achieve spatial equivariance, the relative displacement vector in world-space $\mathbf{x}_{ij} = u_j - u_i$ and its norm $|\mathbf{x}_{ij}|$ are encoded into the edge feature:

$$\mathbf{e}_{ij} = \mathcal{MLP}([u_j - u_i, |u_j - u_i|]) \tag{3.2}$$

The connectivity is added based on the triangular mesh topology in the simulation.

In summary, the node features embed the cloth vertex positions and the target geometry of the deformation. The edge features embed the original length of the template mesh edges. These are sufficient information for a spring-mass model when keeping constants fixed.

The input of the Graph Network is the 3D coordinates of each vertex in the template mesh and the depth image feature. Through message passing and feature updates described in Section 3.2.3, the network predicts the 3D coordinates of each vertex in the deformed mesh.

3.2.3 Updater

The defining feature of GNN is the message passing mechanism. The messages are aggregated from neighbouring nodes and updated iteratively using neural networks. After several iterations, the update can propagate through the entire graph, i.e. after k iterations, every node embedding $\mathbf{h}_u^{(k)}$ encodes all the node information in the k-hop neighbourhood of node u. In other words, apart from feature-based information, the node embeddings also encode structural information about the graph.

The overall function of our proposed network is to deform the template mesh into the target geometry. In each iteration of message passing, the communication range of each node increases. The entire message-passing update models the deformation process, and in every iteration, the template mesh is deformed further according to new adjacent information obtained from a larger communication range.

In a vanilla GNN, neighbouring messages are summed directly. In our work, neighbouring messages are pooled and updated based on the attention mechanism [28]. The network will learn on its own which neighbours are more important. The intuition is that neighbours in flat regions have less influence than those in bumpy regions.

In iteration t, we first update the edge feature using its endpoints' node embeddings $\mathbf{h}$, for node u and its neighbour w:

$$\mathbf{e}_{uw}^{(t+1)} = \phi_e^{(t)}(\mathbf{e}_{uw}^{(t)}, \mathbf{h}_u^{(t)}, \mathbf{h}_w^{(t)}) \tag{3.3}$$

Normalized attention weights are computed as a differentiable function of updated edge features $\mathbf{e}$:

$$\alpha_{uw}^{(t)} = \mathcal{MLP}(\mathbf{e}^{(t+1)})_{uw} \tag{3.4}$$

$$\mathbf{w}_{uw}^{(t)} = \frac{\exp\left(\alpha_{uw}^{(t)}\right)}{\sum_{v \in \mathcal{N}(u)} \exp\left(\alpha_{uv}^{(t)}\right)} \tag{3.5}$$

Then neighbouring messages are aggregated with learned weights $\mathbf{w}$. The embedding $\mathbf{h}_u^{(t)}$ corresponding to node u is updated using its neighbours $\mathcal{N}(u)$:

$$\mathbf{m}_{\mathcal{N}(u)}^{(t)} = \sum_{v \in \mathcal{N}(u)} \mathbf{w}_{uv}^{(t)} \mathbf{e}_{uv}^{(t+1)} \tag{3.6}$$

$$\mathbf{h}_u^{(t+1)} = \phi_v^{(t)}\left(\mathbf{h}_u^{(t)}, \mathbf{m}_{\mathcal{N}(u)}^{(t)}\right) \tag{3.7}$$

Neighbourhood aggregation ensures that neighbouring features are similar, so the output is smooth and more realistic in shape.

The edge function ϕ_e and the node function ϕ_v are parametrized using MLPs. The initial embedding $\mathbf{h}^{(0)}$ and $\mathbf{e}^{(0)}$ are set to the node feature and the edge feature encoded by the Encoder, respectively. After L iterations of message passing, the final output becomes the node embedding, i.e.,

$$\mathbf{v}_u = \mathbf{h}_u^{(L)}, \forall u \in V \tag{3.8}$$

3.2.4 Decoder

After L message passing updates, each node has information about all nodes in the L-hop neighbourhood. The decoder uses an MLP to decode the latent node features $\mathbf{v}_u$ to 3D coordinates Y_u of the mesh vertex u.

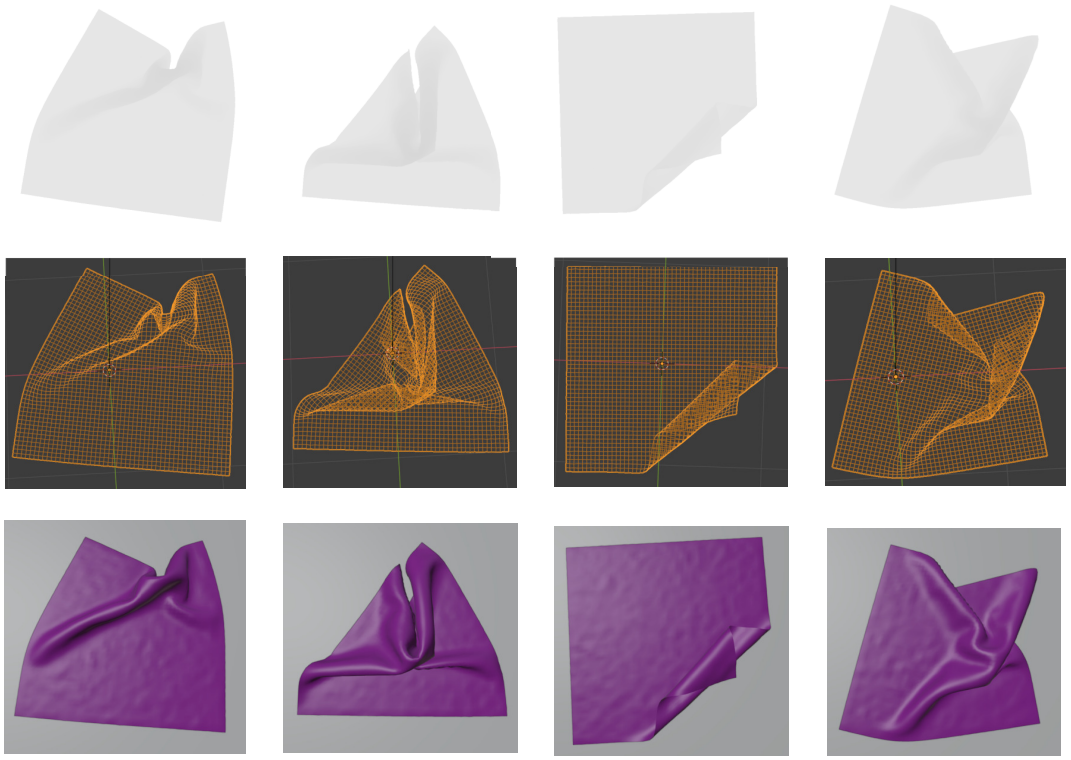


Figure 3.3: Examples of the synthetic dataset. **First row:** Input depth image. **Second row:** Rendered mesh structure corresponding to 3D coordinates of vertices in the annotation. **Third row:** Rendered cloth image corresponding to the annotation. The cloth images are not in the dataset and are for demonstration purposes only.

3.3 Model training

We train our GAT by supervising on the per-vertex output coordinates decoded by the decoder using the L_1 loss:

$$\mathcal{L} = \frac{1}{|V|} \sum_{i=1}^{|V|} \|\hat{Y}_i - Y_i\|_1 \tag{3.9}$$

where $\hat{Y} \in \mathbb{R}^{|V|\times 3}$ is the predicted 3D coordinates of mesh vertices, and Y is the ground truth shape.

Training Details. The model is trained using the synthetic dataset described in Section 3.4. From the generic image feature extractor to MLPs in our GAT, all parts of the network are trained from scratch. We use the Adam optimizer [29] with a batch size of 8 and a learning rate of 1e-4 for the first 100 epochs and 1e-5 for the last 100 epochs.

Data Augmentation. To solve the occlusion problem, we perform data augmentation during the training process. We simulate occlusions by randomly erasing pixels in the depth image using randomly generated rectangles. The trained network is robust to partial occlusions of input depth images.

3.4 Dataset

Varol *et al.* [30] proposed an annotated non-rigid surface dataset with 505 images, which is far from enough for training a neural network. Pumarola *et al.* [19] proposed a large photo-realistic dataset with 128,000 samples. However, the non-rigid surfaces are rather flat, which is trivial in the unfolding setting.

To reduce depth ambiguity while keeping problems as simple as possible, we generate a large-scale dataset with rendered depth images annotated with corresponding 3D coordinates. Each sample consists of a 1080×1080 depth image and a 2601×3 3D shape annotation. Some examples are shown in Figure 3.3.

We model the cloth as a spring-mass model in simulation when generating the dataset. Specific simulation parameters are detailed in Section A.1. The square mesh is divided into $50 \times 50 \times 2$ triangles. In each simulation, we randomly select a vertex, lift it with a random height, and move it in a random direction and distance. After each manipulation, the rendered depth image and 3D coordinates of mesh vertices are saved.

Results

Chapter 4

First, we present qualitative examples of our model. Then, we compare our proposed network with a baseline model without incorporating any prior information and demonstrate that our model can produce more accurate and realistic predictions. We also visualize the deformation process after each message passing iteration to show that our model iteratively deforms the template mesh to the target geometry. Finally, we demonstrate that our model is robust to occlusions.

4.1 Qualitative evaluation

Figure 4.1 presents qualitative examples of our model. This demonstrates that our model is able to accurately and realistically estimate the geometry of the wrinkled garment from the input depth image.

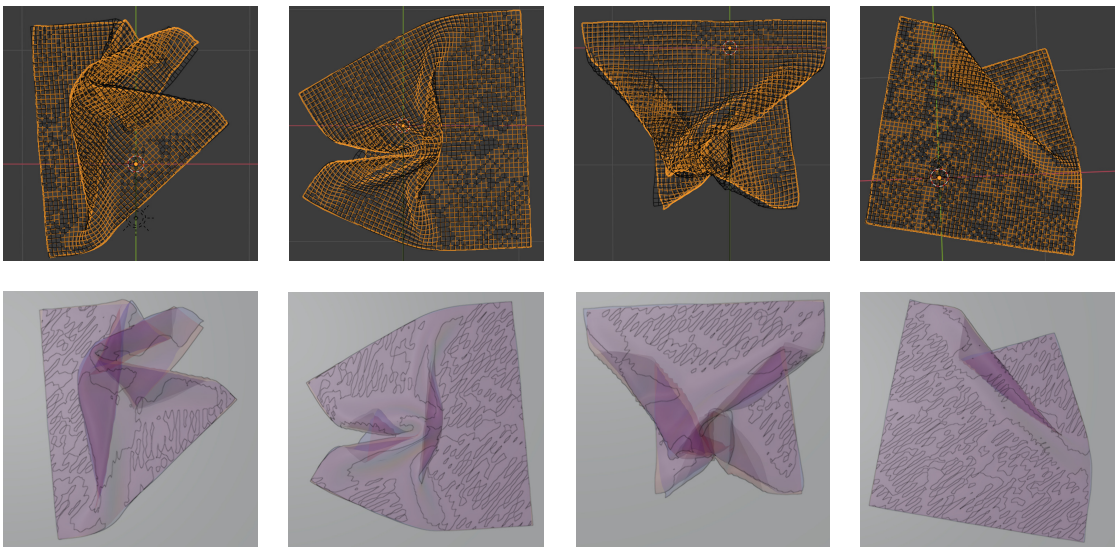


Figure 4.1: Qualitative results of our model. **First row:** Predicted mesh vertices. Yellow meshes are predictions and black meshes are ground truths. **Second row:** Rendered predicted garments. Blue ones are predictions and red ones are ground truths.

4.2 Network comparison

We also implement a baseline model which does not incorporate any geometric prior. The baseline model regresses 3D coordinates of mesh vertices with an MLP on top of the generic image extractor. The architecture of the baseline is shown in Figure 4.2. This baseline model fails to produce a realistic and smooth prediction. The comparison in Table 4.1 and Figure 4.3 demonstrates the advantage of incorporating geometric priors of the template mesh into our proposed network. Our network achieves 38% less error on the test set.

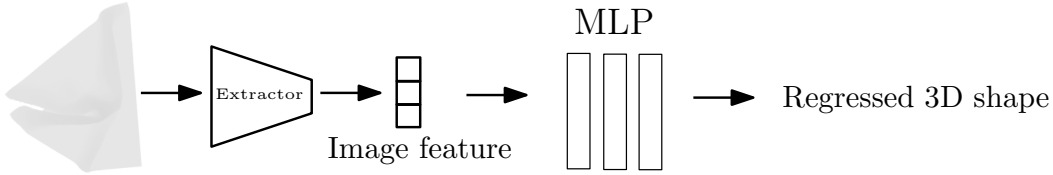


Figure 4.2: Overview of the baseline model architecture with no geometric prior.

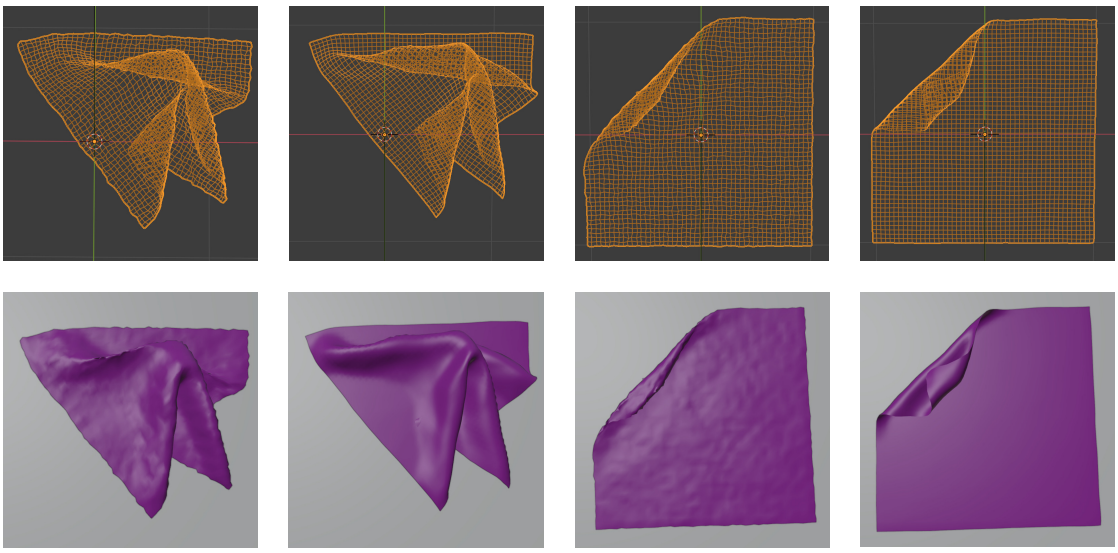


Figure 4.3: Qualitative comparison between our model and the baseline. The predictions of baseline are unrealistic, non-smooth and jagged. Because our model can leverage the geometric prior and encoded mesh structure information, the predictions are more realistic. Neighbouring node features are similar because of message aggregation in each update. So the predictions are more smooth.

Method	*Train Error*	*Test Error*
Baseline: Direct Regression	0.0093	0.0133
Ours: Graph Attention Network	0.0049	**0.0082**

Table 4.1: Comparison between the baseline model and our method. We report the L_1 loss of the 3D coordinates between estimated and ground-truth vertices on the mesh. With geometric priors, our model has a better fitting ability. It demonstrates that our model can generalize better leveraging mesh structure information.

4.3 Message passing visualization

From a high-level perspective, our model deforms the template mesh into target geometry using a depth image. During the iterative message passing process, each node knows more and more information about its neighbours. The deformation process is gradual with the increase of information and each iteration deforms the mesh a little in the latent space.

In our implementation, the network performs 15 message passing updates. To dig deeper into the network, we visualize the cloth mesh after each update, as shown in Figure 4.4. Node features are decoded with the trained decoder. We demonstrate that the template cloth mesh is deformed in 15 successive deformations until the target geometry.

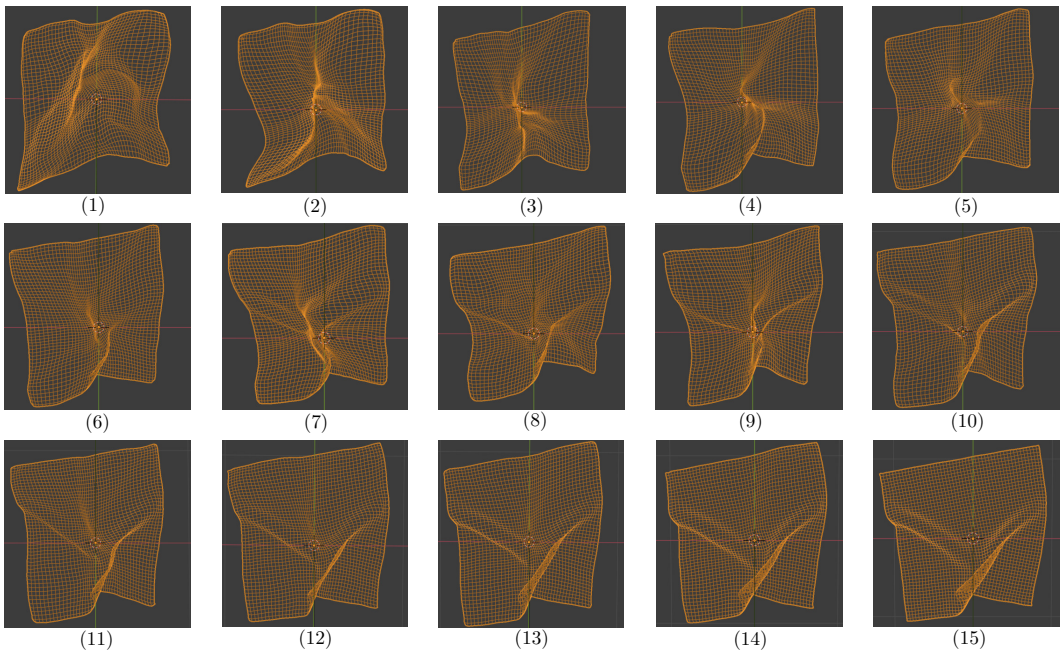


Figure 4.4: Visualization of the cloth mesh after each message passing iteration. The pictures are labelled in order. Picture (15) is the final prediction of the network. (1) - (14) are decoded using the same decoder as (15).

4.4 Prediction under occlusion

In a real-world setting, the view can be occluded by various obstacles including robot arms. Traditional geometry-based methods are sure to fail without enough information. However, in our method, we train the network using data augmentation and achieve an occlusion-robust performance.

The data augmentation is implemented by randomly erasing pixels from the input depth image. We generate a set of random distributed occluding rectangles on the input image with some probability. The trained network is robust to occlusions as shown in Figure 4.5.

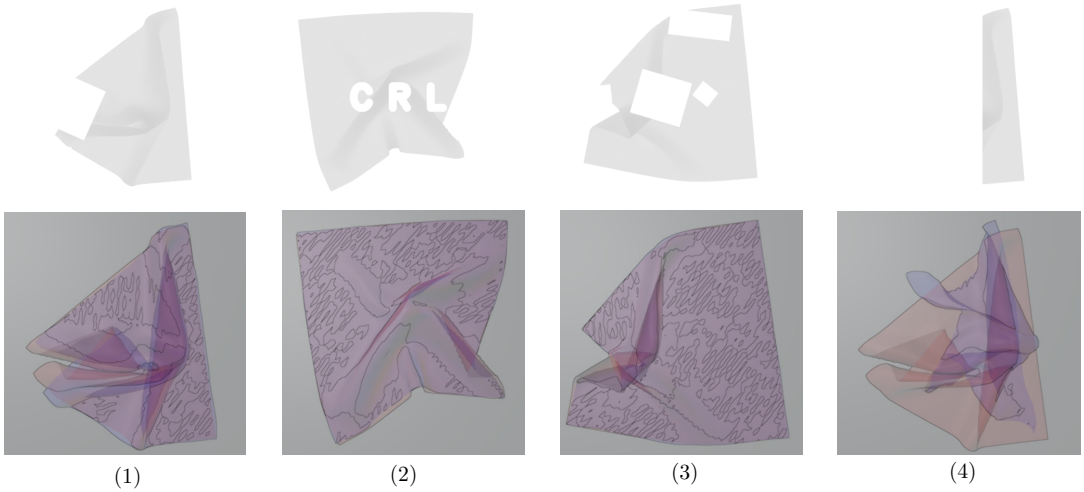


Figure 4.5: Model performance when input images have occlusions. In the second row, the red garment is the ground truth and the blue garment is the prediction. Random rectangles generated in data augmentation are similar to (1) and (3). (2) demonstrates that our model can also deal with non-rectangular occlusions. (4) shows one failure case when the occlusion can not be overlooked.

Conclusion

Chapter 5

We demonstrated our fully end-to-end Graph Attention Network model that is able to estimate the geometry of wrinkled garments from single-view input depth images in real-time. Our model explicitly estimates the 3D coordinates of cloth mesh vertices. By leveraging the structural information that the template mesh offers, we are able to incorporate geometric priors into the network. As a result, our model is able to produce accurate and realistic predictions of the geometry of wrinkled garments. Our model can be generalized to visually different garments since only depth images are used. Our model is also robust to partial occlusions, which solves part of the sim2real problem. We demonstrated that our model using Graph Attention Network has reduced prediction errors by 38% compared to a baseline model without geometric priors and that our method achieves more realistic predictions. From a high-level perspective, our model deforms the template mesh into the target geometry. Although we only demonstrated wrinkled garments with the same size and shape, our method can be extended to a different template, even some complex template, as long as all garments have the same geometry in the flat state.

For possible future directions, it would be important to explore when different garments have different geometries in the flat state, i.e. different templates. It would be interesting to train a network that can predict the geometry of both the jacket and the pants. Our model works well using simulated data, so it is worth exploring if the model can generalize well in the real-world setting. Finally, when the geometry of the garment is too complex and the garment has too much self-occlusion, our model will fail. It would be worth exploring in this ill-posed setting.

Bibliography

[1] A. Clegg, W. Yu, J. Tan, C. K. Liu, and G. Turk, "Learning to dress: Synthesizing human dressing motion via deep reinforcement learning," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–10, 2018.

[2] B. Balaguer and S. Carpin, "Combining imitation and reinforcement learning to fold deformable planar objects," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1405–1412.

[3] A. Doumanoglou, A. Kargakos, T.-K. Kim, and S. Malassiotis, "Autonomous active recognition and unfolding of clothes using random decision forests and probabilistic planning," in *2014 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2014, pp. 987–993.

[4] N. Koganti, T. Tamei, K. Ikeda, and T. Shibata, "Bayesian nonparametric learning of cloth models for real-time state estimation," *IEEE Transactions on Robotics*, vol. 33, no. 4, pp. 916–931, 2017.

[5] P. Jiménez and C. Torras, "Perception of cloth in assistive robotic manipulation tasks," *Natural Computing*, pp. 1–23, 2020.

[6] N. Ukita and T. Kanade, "Reference consistent reconstruction of 3d cloth surface," *Computer Vision and Image Understanding*, vol. 116, no. 8, pp. 869–881, 2012.

[7] S. Miller, J. Van Den Berg, M. Fritz, T. Darrell, K. Goldberg, and P. Abbeel, "A geometric approach to robotic laundry folding," *The International Journal of Robotics Research*, vol. 31, no. 2, pp. 249–267, 2012.

[8] B. O. Community, *Blender - a 3D modelling and rendering package*, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. [Online]. Available: http://www.blender.org

[9] X. Ma, D. Hsu, and W. S. Lee, "Learning latent graph dynamics for deformable object manipulation," *arXiv preprint arXiv:2104.12149*, 2021.

[10] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[11] M. Cranmer, A. Sanchez-Gonzalez, P. Battaglia, R. Xu, K. Cranmer, D. Spergel, and S. Ho, "Discovering symbolic models from deep learning with inductive biases," *arXiv preprint arXiv:2006.11287*, 2020.

[12] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[13] D. Metaxas and D. Terzopoulos, "Constrained deformable superquadrics and nonrigid motion tracking," in *Proceedings. 1991 IEEE Computer Society Conference on Computer Vision and Pattern Recognition.* IEEE Computer Society, 1991, pp. 337–338.

[14] T. McInerney and D. Terzopoulos, "A dynamic finite element surface model for segmentation and tracking in multidimensional medical images with application to cardiac 4d image analysis," *Computerized medical imaging and graphics*, vol. 19, no. 1, pp. 69–83, 1995.

[15] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] L. Torresani, A. Hertzmann, and C. Bregler, "Nonrigid structure-from-motion: Estimating shape and motion with hierarchical priors," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 5, pp. 878–892, 2008.

[17] C. Bregler, A. Hertzmann, and H. Biermann, "Recovering non-rigid 3d shape from image streams," in *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, vol. 2. IEEE, 2000, pp. 690–696.

[18] A. Bartoli, Y. Gérard, F. Chadebecq, T. Collins, and D. Pizarro, "Shape-from-template," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 10, pp. 2099–2118, 2015.

[19] A. Pumarola, A. Agudo, L. Porzi, A. Sanfeliu, V. Lepetit, and F. Moreno-Noguer, "Geometry-aware network for non-rigid shape prediction from a single view," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4681–4690.

[20] N. Kolotouros, G. Pavlakos, and K. Daniilidis, "Convolutional mesh regression for single-image human shape reconstruction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4501–4510.

[21] K. Lin, L. Wang, and Z. Liu, "End-to-end human pose and mesh reconstruction with transformers," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1954–1963.

[22] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, "Learning mesh-based simulation with graph networks," *arXiv preprint arXiv:2010.03409*, 2020.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[24] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, pp. 1097–1105, 2012.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1–9.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European conference on computer vision*. Springer, 2016, pp. 630–645.

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[30] A. Varol, M. Salzmann, P. Fua, and R. Urtasun, "A constrained latent variable model," in *2012 IEEE conference on computer vision and pattern recognition*. Ieee, 2012, pp. 2248–2255.

More Stuff

Appendix A

A.1 Simulation parameters

Parameters	*Values*
Method for splitting the quads into triangles	'FIXED' (Split the quads on the first and third vertices)
Cloth vertex mass	0.5
Air damping	1.0
Tension stiffness	10.0
Compression stiffness	10.0
Shear stiffness	40.0
Bending stiffness	40.0
Tension damping	10.0
Compression damping	10.0
Shear damping	2.0
Bending damping	2.0

Table A.1: Simulation parameters