

Optimization of Material Properties for Coherent Behavior across Multi-resolution Cloth Models

Nak-Jun Sung¹, Shane Transue², Minsang Kim¹, Yoo-Joo Choi³,
Min-Hyung Choi² and Min Hong⁴

¹ Department of Computer Science, Soonchunhyang University, Asan, Korea
e-mail: njsung@sch.ac.kr, ben399399@sch.ac.kr]

² Department of Computer Science and Engineering, University of Colorado Denver, Colorado, USA
[e-mail: shane.transue@ucdenver.edu, Min.Choi@ucdenver.edu]

³ Department of Newmedia Content, Seoul Media Institute of Technology, Seoul, South Korea
[e-mail: yjchoi@smit.ac.kr]

⁴ Department of Computer Software Engineering, Soonchunhyang University, Asan, Korea
[e-mail: mhong@sch.ac.kr]

*Corresponding author : Min Hong

*Received March 14, 2018; revised June 7, 2018; revised July 30, 2018; accepted August 6, 2018;
published August 31, 2018*

Abstract

This paper introduces a scheme for optimizing the material properties of mass-spring systems of different resolutions to provide coherent behavior for reduced level-of-detail in MSS(Mass-Spring System) meshes. The global optimal material coefficients are derived to match the behavior of provided reference mesh. The proposed method also gives us insight into levels of reduction that we can achieve in the systematic behavioral coherency among the different resolution of MSS meshes. We obtain visually acceptable coherent behaviors for cloth models based on our proposed error metric and identify that this method can significantly reduce the resolution levels of simulated objects. In addition, we have confirmed coherent behaviors with different resolutions through various experimental validation tests. We analyzed spring force estimations through triangular Barycentric coordinates based from the reference MSS that uses a Gaussian kernel based distribution. Experimental results show that the displacement difference ratio of the node positions is less than 10% even if the number of nodes of MSS^{sim} decreases by more than 50% compared with MSS^{ref}. Therefore, we believe that it can be applied to various fields that are requiring the real-time simulation technology such as VR, AR, surgical simulation, mobile game, and numerous other application domains.

Keywords: Mass-spring System, Estimation, Coherent behavior, Cloth Simulation, Dynamic Simulation

A preliminary version of this paper was presented at ICONI 2017 and was selected as an outstanding paper. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(Ministry of Science, ICT & Future Planning) (No. 2017R1A2B1005207) and was supported by the Soonchunhyang University Research Fund.

1. Introduction

Mass-spring systems (MSS) have been widely used in the dynamic simulations to represent various types of soft objects including cloth, hair, and deformable solids. Due to its simplicity and computational efficiency, MSS is still very popular for the real-time interactive applications and large-scale deformable object simulations. Levels-of-detail (LOD) of mesh structures have been widely researched for representing meshed geometry in various detail levels, however when we adopt similar concept to the mass-spring systems for deformable objects, it's highly unlikely to preserve behavioral coherency of objects when same elastic coefficients are used, as shown in the middle image of **Fig. 1**. This problem is caused by the anisotropic and geometric properties of mass-spring systems and different mesh structures and resolutions can introduce different connections within the solid mesh.

This paper describes a novel method for deriving the optimal material coefficients for the lower resolution of mass-spring systems based on a higher resolution of mass-spring system. This provides the ability to introduce the LOD of mass-spring system to account for visual appearance in behavior and performance. The proposed method includes four primary steps. In the first step, it finds coherent points between high resolution of MSS and low resolution of MSS. Second, it simulates the deformable object using a high resolution of MSS and estimate the position of the coherent points. Third, the proposed method computes the stiffness parameters by inverse procedure of Hooke's law, and finally solves an estimation problem based on the proposed error metric to obtain a global optimal stiffness and damping coefficient.

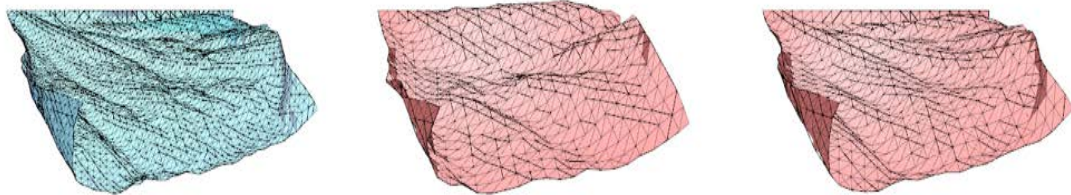


Fig. 1. The cloth models with different mesh resolutions. (a) the reference model with high resolution(60*60 nodes), (b) the low resolution(24*24 nodes) model without the optimal material properties and (c) the low resolution(24*24 nodes) model with the material properties optimized by the proposed method

Fig. 1 shows the difference without and with the proposed optimal material properties method that provides coherent behavior between the reference and simulation model. Vastly different mesh resolutions would result in irreconcilable behaviors, while limits of lower bounds of mesh resolution for real-time processing are different for each hardware platform. Therefore, the optimization of material properties between deformable objects with different mesh resolutions is required to provide the coherent behavior of the same deformable object in the various hardware platforms in real-time.

2. Related Work

Deformable object simulation refers to an object that changes its shape when a force is applied to the object [1-4] with various derivations including mass-spring systems, the Finite Element Method (FEM), and Position-based Dynamics [22]. MSS has been widely applied to real-time simulations because its computational performance is better than that of the finite element method [5-7] and simplicity over position-based dynamics. The most common MSS formulation consists of point-masses that make up a deformable object and a set of virtual springs connect these masses together in a self-contained system. While other techniques have grown in popularity, cloth simulation remains prominent as one of the primary uses for mass-spring systems. Conventional cloth deformable simulations use a formulation that defines three types of basic springs such as structural, shear, and bend springs as shown in Fig. 2.

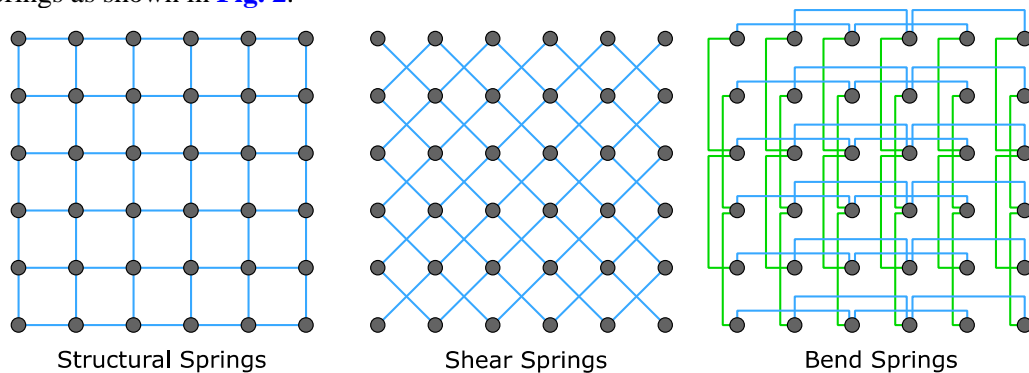


Fig. 2. Three types of basic springs for cloth simulation

Physically based cloth simulations have been extensively studied for decades [8-9] and have been widely adapted in games, animation, mobile computing, and virtual reality. Many approaches have been introduced to improve the realism of cloth simulation and volumetric models using mass-spring systems [10]. Provot [11] focused on iteratively adjusting the nodal position to avoid over-elongation and adopt an inverse dynamic procedure to resolve it and Baraff [12] provided a stable implicit numerical integration method to allow for large time steps when simulating mass-spring systems. Mathieu et al. [13] proposed a predictor-corrector approach to alleviate the computational burden of solving a linear system at each time step for animating mass-spring systems. Liu et al. [14] proposed a fast-implicit solver for mass spring systems while maintaining behavioral correctness when compared to a traditional Newton-based solver.

Reference behavior optimization techniques have also been introduced for modifying cloth simulation behaviors. Bhat et al. [15] used images of actual cloth movements to infer the parameters of the cloth simulation. The sequence of the cloth movement was compared, and the change amount was measured and the parameters of the cloth simulation were estimated to improve the behavioral coherence between the simulated and real-world cloth. In addition, researchers have examined the collision of mass-spring systems using Barycentric coordinates that make the motion of objects similar [16-19]. In this work, our objective is to derive the parameters of the cloth mesh based on the contributions of these studies to build an LOD-based behavioral coherence compatible with these existing simulation methods.

3. Coherent Behavior

The proposed method focuses on real-time cloth simulation and our objective is to optimize the material property coefficients of a low resolution deformable model to achieve a behavior similar to that of a high resolution model. We define an algorithmic optimization process for deriving consistent dynamics based on the selection of ideal material properties that generate this coherent behavior. We precisely define the coherent behaviors between two simulated mass-spring systems by two quantitative metrics: (1) global trajectory and (2) localized deformation behaviors. These metrics define the similarity between two simulated objects based on their global trajectory and the behaviors of the elastic material properties that defines how meshes deform over time. As these general cost metrics are minimized, the simulated MSSs should coincide in displacements, velocity, deformation, and trajectory which leads to coherent behavior between two MSSs with different resolutions.

3.1 Method Overview

The general cost metrics for global trajectory and local deformation are defined as functions of the isotropic elastic material properties of MSS: the uniform spring stiffness of $K_{structure}$, K_{shear} , K_{bend} and the damping coefficient $C_{structure}$, C_{shear} , C_{bend} . To define the differences in the discrete representation of simulated MSS, we define the high resolution of model as the reference model, MSS^{ref} , which has the behavior to be emulated, and get the low resolution of cloth model, MSS^{sim} . Through the estimation of these parameters between MSS^{ref} and MSS^{sim} , we provide a least-squares approximation of the optimal material properties that can be applied to MSS^{opt} to provide behaviors that are coherent with MSS^{ref} . The process of optimizing the material properties of the input MSS to obtain the coherent behavior provided by the MSS^{opt} is shown in Fig. 3.

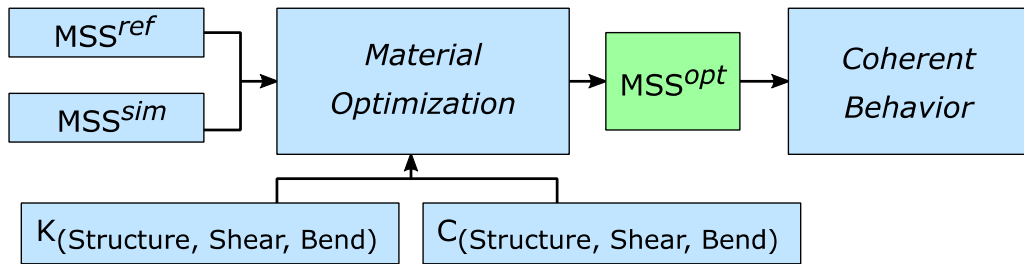


Fig. 3. Coherent MSS behavior generation using material property optimization

As the primary source for driving the estimation of the material parameters, we define a relational mapping that defines the discrete nodal position differences between MSS^{ref} and MSS^{sim} . This defines a method for both tracking the movement of the MSS^{ref} and the cost function for the material estimation. The objective of this cost function is to minimize the difference between the two meshes for the given set of recorded frames within the simulation.

3.2 Barycentric Correlation

In our approach, we utilize Barycentric coordinates to identify the corresponding points that

map the geometric structure of the MSS^{sim} to that of the MSS^{ref} to define the behavioral correlation. Between the two MSS cloth models, we define correlated Barycentric coordinates between each node P of the low-resolution model MSS^{sim} and a triangle ($\triangle ABC$) of the reference model MSS^{ref} with the high resolution containing the node P inside as shown in the left image of Fig. 4. Fig. 4 shows the correlated point within the high-resolution model to drive the behavior of the node from the low-resolution model.

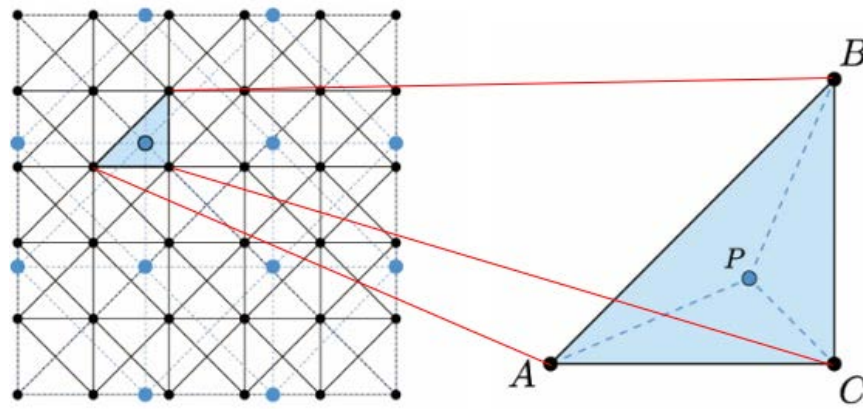


Fig. 4. Barycentric coordinate method for finding the coherent points between MSS^{ref} and MSS^{sim}

After we achieve the coherent relationship between reference model and simulated model from the barycentric correlation, we still need to translate back to Cartesian coordinate for each time step to achieve the actual correspondent positions. We use the standard equation to translate a barycentric coordinate to cartesian coordinate, which we can then use within our estimation algorithm to drive the behavior of the MSS^{sim} to behave like the MSS^{ref} . The node position of MSS^{sim} , which is corresponding to positions of three nodes in specific triangle of MSS^{ref} can be estimated using barycentric correlation as shown in the Fig. 5. Thus, the node positions for low resolution of MSS^{sim} are calculated through the simulation result of node positions for high resolution of MSS^{ref} .

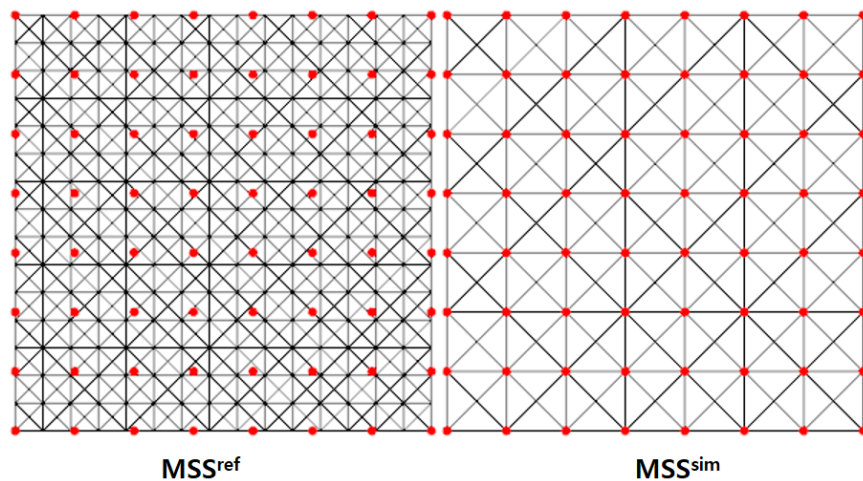


Fig. 5. The relation between the reference model MSS^{ref} and the simulated model MSS^{sim}

4. Various Resolutions in Cloth Simulation

Considering the mature state of the standard mass-spring system formulation, we build upon the existing techniques for improving the physical plausibility of the physical behaviors provided by spring-simulated meshes. The MSS cloth consists of $N * M$ nodes and three types of virtual springs indicating the relationship between them. Each node has the position, velocity, force, and mass as default attributes. Each virtual spring has three stiffness ($K_{Structure}$, K_{Shear} , K_{Bend}) and damping coefficients ($C_{Structure}$, C_{Shear} , C_{Bend}), the information of two connected nodes (**Index1**, **Index2**), and initial length for spring (**RestLength**). The cloth simulation is performed by calculating the spring forces, which cause the movement of node in cloth model. The spring forces are calculated by equation (1).

$$\begin{aligned}
 Stiffness &= (CurLength - RestLength) * K \\
 Damping &= \frac{VelocityDirection \cdot dot(PositionDirection)}{CurLength} * C \\
 SpringForce &= (Stiffness + Damping) * \frac{PositionDirection}{CurLength}
 \end{aligned} \tag{1}$$

Stiffness and damping coefficients are calculated by the change of PositionDirection and current length of spring (CurLength) due to the change of node position. The forces applied to the cloth object are determined through the sum of each spring forces. While MSS provides semi-plausible behaviors of deformable objects, our objective is to provide efficient and coherent behavior between low and high resolution of models for the large-scale simulations. Therefore, our implementation utilizes 4th order Runge-Kutta explicit integration method. Equation (2) represents the 4th order Runge-Kutta method. Although the 4th order Runge-Kutta integral method is complex than the simple Euler method, the error is less than the Euler integral method.

$$\begin{aligned}
 v(t + \Delta t) &= v(t) + \frac{1}{6} (F_1 + 2(F_2 + F_3) + F_4) \\
 F_1 &= \frac{\Delta t}{2} * a(t + \Delta t), F_2 = \frac{\Delta t}{2} * \frac{F_1}{m}, F_3 = \Delta t * \frac{F_2}{m}, F_4 = \Delta t * \frac{F_3}{m} \\
 p(t + \Delta t) &= p(t) + \frac{1}{6} (K1 + 2(K2 + K3) + K4) \\
 K_1 &= \frac{\Delta t}{2} * a(t + \Delta t), K_2 = \frac{\Delta t}{2} * K_1, K_3 = \Delta t * K_2, K_4 = \Delta t * K_3
 \end{aligned} \tag{2}$$

where Δt , m , $v(t)$ and $p(t)$ represent the unit time, the mass, velocity and position of a given node at time t , respectively. The generation of cloth models for MMS is well established as the generation of two-dimensional patch surfaces defined by (1) structural springs, (2) shear springs, and (3) bend springs. The model is generated through the number of nodes defined as $N * M$. As the number of N and M increases, the cloth model is created as a finer model, and as the number decreases, it becomes a simpler model. **Fig. 6** shows the result of the generation of various cloth models.

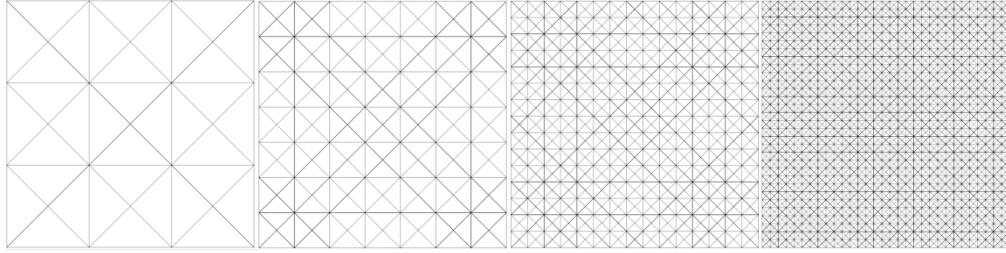


Fig. 6. The different cloth models generated by $N \times M$ nodes (4×4 , 8×8 , 16×16 , 32×32)

5. Material Property Estimation Method

This section describes how to estimate the material properties of MSS models. Our goal is to figure out the coherent relation between different resolution of models, thus, we need to derive the material property of low resolution of model based on the high resolution of reference model. The proposed material property estimation approach focuses on the parameter of stiffness and damping coefficient, and it has two steps of estimation for material properties. First, the MSS^{ref} is simulated using given stiffness coefficient ($K_{structure}$, K_{shear} , K_{bend}) and damping coefficient ($C_{structure}$, C_{shear} , C_{bend}), the position information of the node is maintained at each time step, and the corresponding position in the time of MSS^{sim} is calculated by applying barycentric coordinates. Next, we applied the inverse procedure to calculate the spring forces for MSS^{sim} in each time-step: this is our first estimation step. The second estimation step is finding the optimal K and C values among all frames from the displacement difference.

5.1 Estimation of Material Property

In the first step, the proposed method estimates the node position of MSS^{sim} using barycentric coordinates from the simulated result of node position for refined MSS^{ref} at each time step. Then we can estimate whole spring forces of each node at each time step because we already have the previous and current node position of MSS^{sim} by the inverse procedure manner. Since each node has three types of spring connections for the traditional cloth models, the estimated whole spring forces should be divided into structure, shear, and bend spring forces according to certain ratio.

In this paper, we propose two methods to distribute the whole spring force which is calculated based on the node for three types of each spring force. The BCD is BCD (Barycentric Coordinate based Distribution) which is applied to calculate the node location of MSS^{sim} with barycentric coordinates [Fig. 7](#) shows the BCD and the estimated whole spring forces should be divided into the red node ($Node^x$). Since the BCD used barycentric coordinate based approach, the ratio of spring force that is applied to the red node was calculated by applying the ratio of spring force that is applied to the three nodes of MSS^{ref} . Equation (3) shows the process of estimating structure, shear, and bend spring forces using BCD. R_{ST} , R_{SH} , R_B are representing the ratio of structure, shear, and bend spring forces and the estimated whole spring forces are divided with this ratio for spring forces of $Node^x$.

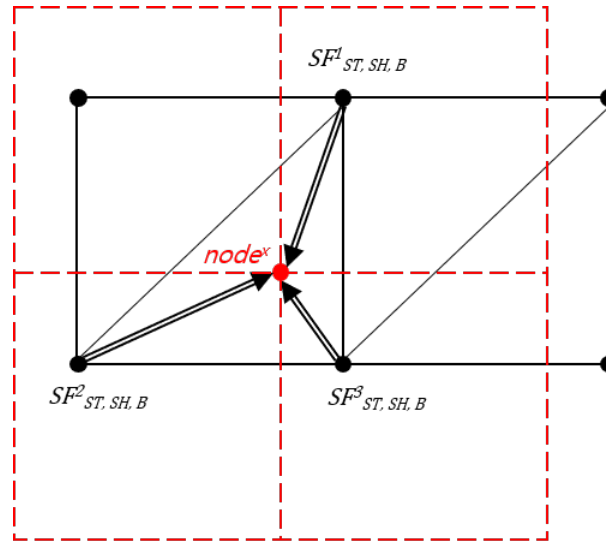


Fig. 7. Barycentric Coordinate-based Distribution (BCD) Method: Calculation of the spring force ratios for an MSS^{sim} node from MSS^{ref}

$$\begin{aligned}
 R_{ST} &= \frac{SF^1_{ST} + SF^2_{ST} + SF^3_{ST}}{SF^1 + SF^2 + SF^3} \\
 R_{SH} &= \frac{SF^1_{SH} + SF^2_{SH} + SF^3_{SH}}{SF^1 + SF^2 + SF^3} \\
 R_B &= \frac{SF^1_B + SF^2_B + SF^3_B}{SF^1 + SF^2 + SF^3}
 \end{aligned} \tag{3}$$

$SF^{1,2,3}$ mean the force of three nodes forming the barycentric triangle of MSS^{ref} . And ST, SH, and B mean the structural force, shear force, and bend force of the corresponding force. The three spring forces ratio determines the ratio of the force of the $node^x$ in **Fig 7**.

The second proposed method is the GKD (Gaussian Kernel-based Distribution) method which considers the ratio of spring force to the nodes of the MSS^{ref} in a wider range to estimate the ratio of the spring forces for a node in MSS^{sim} . In order to estimate the ratio of spring force for the red node ($Node^x$) in MSS^{sim} , we calculated the whole ratio of spring forces for some nodes in MSS^{ref} which are located slightly larger than the length of shear spring for MSS^{sim} as shown in **Fig. 8**. In addition, since the influence of spring forces on the $Node^x$ varies depending on distance, the ratio of spring forces is calculated using the Gaussian kernel.

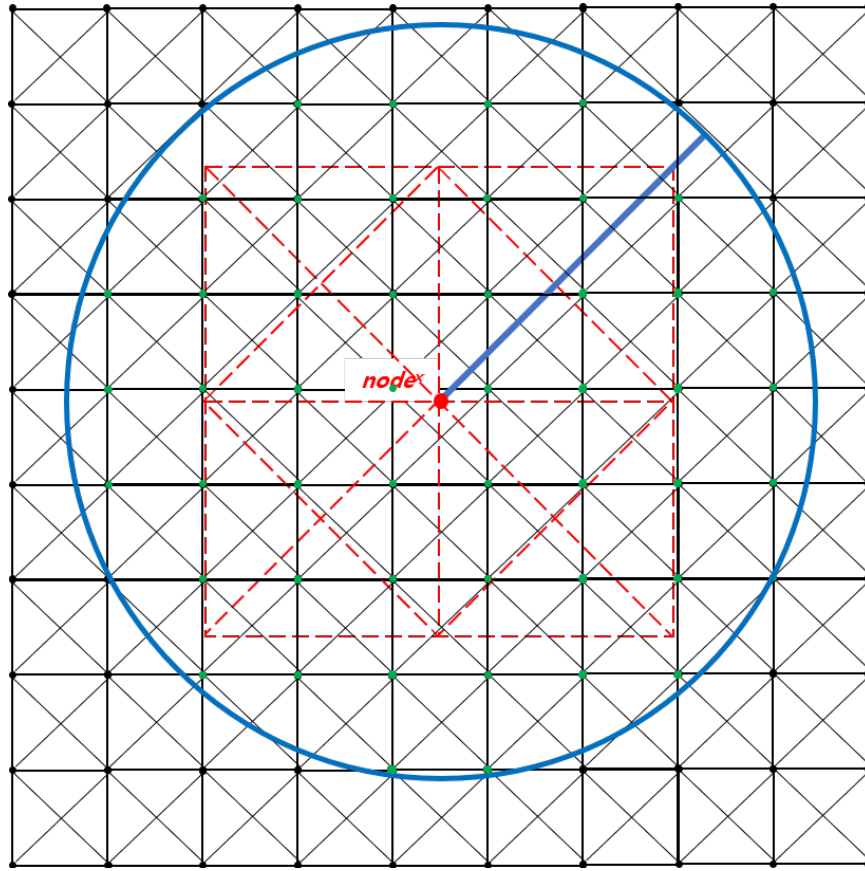


Fig. 8. Gaussian Kernel-based Distribution (GKD) Method: Calculation of spring force ratio for MSS^{sim} node from MSS^{ref}

To account for the spring forces applied to the nodes of MSS^{ref} at the shear spring distance of MSS^{sim} , the estimation of spring forces on the $Node^x$ was applied to a range of 25% greater distance than the length of shear spring like the blue circle in **Fig. 8**.

We apply the weight over the distance through the Gaussian kernel method, which is calculated by equation (4). σ means width of the Gaussian kernel and x means size of the Gaussian kernel.

$$G_{1D}(x; \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-x^2}{2\sigma^2}} \quad (4)$$

Fig. 9 shows the result of applying different values of σ and x and we set up the σ value to 3.0 and x to 7 for Gaussian kernel for our experimental tests. Each region of the Gaussian kernel has different weights, and we apply the corresponding weights to the distances from $Node^x$ in MSS^{sim} to nodes that are included within a length of shear spring of in MSS^{ref} . **Fig. 10** shows the weights of Gaussian kernel applied in this paper.

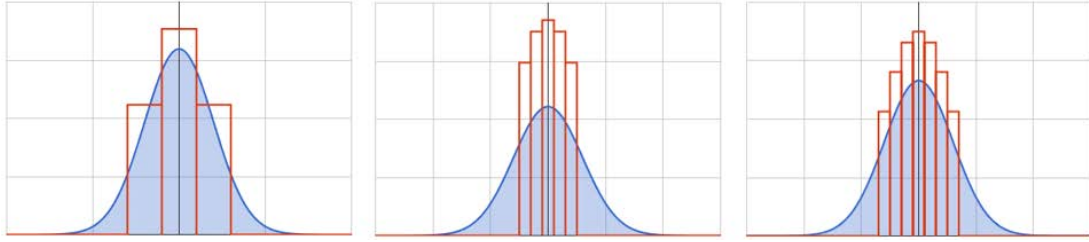


Fig. 9. Gaussian Kernel by parameter (σ , x)

left : $\sigma = 1.0$, $x = 3$ / mid : $\sigma = 3.0$, $x = 5$ / right : $\sigma = 3.0$, $x = 7$

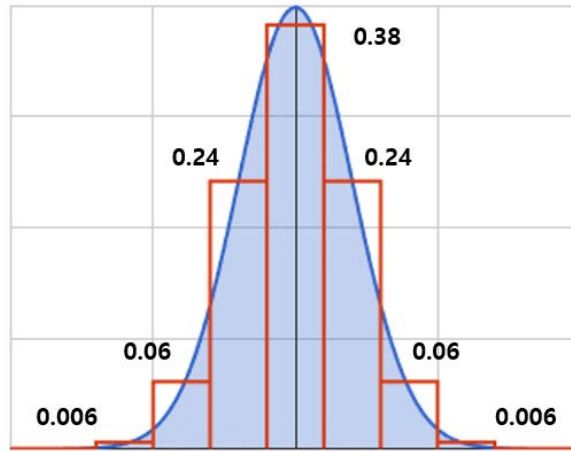


Fig. 10. Setting result of Gaussian Kernel and Gaussian Weight

First, the $gaussian_weight^i$ for each node ($node^i$) in MSS^{ref} is calculated as shown in equation (5). The proposed GKD compares the distance between $Node^x$ in MSS^{sim} and nodes in MSS^{ref} distance with the Gaussian kernel, and stores the weight for corresponding distance. To calculate the ratio of spring forces for R_{ST} , R_{SH} , R_B , the spring forces of n nodes which are inside the length of shear spring of MSS^{sim} in MSS^{ref} are multiplied by their respective Gaussian weights. Therefore, the ratio of spring forces according to each spring type are calculated through equations (3) and (6). Then, the calculated force ratio is reflected to each spring force for nodes in MSS^{sim} by dividing whole spring force into the respective ratio of spring forces. The K values are calculated by reversing the spring force equation (1) based on each calculated spring force. In this case, the K values of each spring are estimated by applying equation (7) without considering the damping coefficients which are C values. Equation (8) is derived from Equation (1) and the damping coefficients are calculated from the already estimated K values. In the proposed method, estimated K and C values that are calculated through equations (7) and (8) are stored at each time-step. After the simulation is completed, the average value of K and C coefficients is calculated and used for the material property of MSS^{opt} .

$$gaussian_weight^i = distance(MSS^{sim}node, MSS^{ref}node^i) \quad (5)$$

$$\begin{aligned}
R_{ST} &= \frac{\sum_{i=0}^n SF_{ST}^i * gaussian_weight^i}{\sum_{i=0}^n SF^i * gaussian_weight^i} \\
R_{SH} &= \frac{\sum_{i=0}^n SF_{SH}^i * gaussian_weight^i}{\sum_{i=0}^n SF^i * gaussian_weight^i} \\
R_B &= \frac{\sum_{i=0}^n SF_B^i * gaussian_weight^i}{\sum_{i=0}^n SF^i * gaussian_weight^i}
\end{aligned} \tag{6}$$

$$K_{ST,SH,B} = (SF_{ST,SH,B} / PosDirection * len) / (len - Restlen) \tag{7}$$

$$C_{ST,SH,B} = (SF_{ST,SH,B} / PosDirection * len) - K_{ST,SH,B} \tag{8}$$

5.2 Displacement Difference

In order to maintain the accuracy of simulation results between MSS^{ref} and MSS^{opt} , we used the displace difference as an important factor. In this paper, the displacement difference between MSS^{ref} and MSS^{opt} are calculated through simulation of 20,000 frames using average of K and C values calculated through the previous computation. We calculate the displacement Δt of each node of MSS^{ref} and MSS^{opt} for each simulation step. First, the error rate for Δt is calculated by equation (9).

$$E_d^{\Delta t} = \frac{1}{O_{max}} \sqrt{\frac{1}{n} \sum_{i=0}^n (\mathbf{x}_i^{ref} - \mathbf{x}_i^{sim})^2} \tag{9}$$

In the equation 8, O_{max} is the maximum Euclidean distance between the unmodified shape and the deformed shape of the reference model through simulation. x_i^{ref} and x_i^{sim} are the positions of coherent node i in MSS^{ref} and MSS^{sim} .

6. Experimental Results

This research aims the coherent behavior or motion between the low resolution of cloth simulation and the high resolution of cloth simulation by applying the estimation algorithm to MSS. Therefore, the K and C values calculated through the proposed estimation algorithm cause the cloth simulation composed of relatively few nodes to move similarly to the cloth simulation composed of much larger numbers of nodes. In this paper, we calculated the behavioral difference between MSS^{ref} (reference model) and MSS^{opt} (estimation model) by comparing the node positions at each time-step. For the experimental tests, we configured MSS^{ref} model as $64 * 64$ (4,096) nodes and set up the total number of nodes for MSS^{sim}

model to decrease by about 60% based on MSS^{ref} model. In addition, since the K and C Values of MSS^{sim} may vary depending on the simulation situations where the MSS^{ref} is subjected to given experimental tests situations, the following three scenes were constructed to estimate the K and C values under various situations.

For the first experimental test (scene 1) as shown in the row image of Fig. 11, the upper nodes marked in red are fixed, and the external forces are applied mainly to pull the structure springs to downward. For the second experimental test (scene 2) as shown in the row image of Fig. 11, the upper left node marked with red is fixed, and the external forces are applied mainly to pull the shear springs bottom-right corner. For the final experimental test (scene 3) as shown in the row image of Fig. 11, the nodes on the left marked with red are fixed, and the external force is applied mainly to pull the structure springs and bend springs to rightward. For MSS^{sim} model simulation, each K and C values are calculated for each types of springs during each time-step. The average values of obtained K and C values are applied for experimental tests.

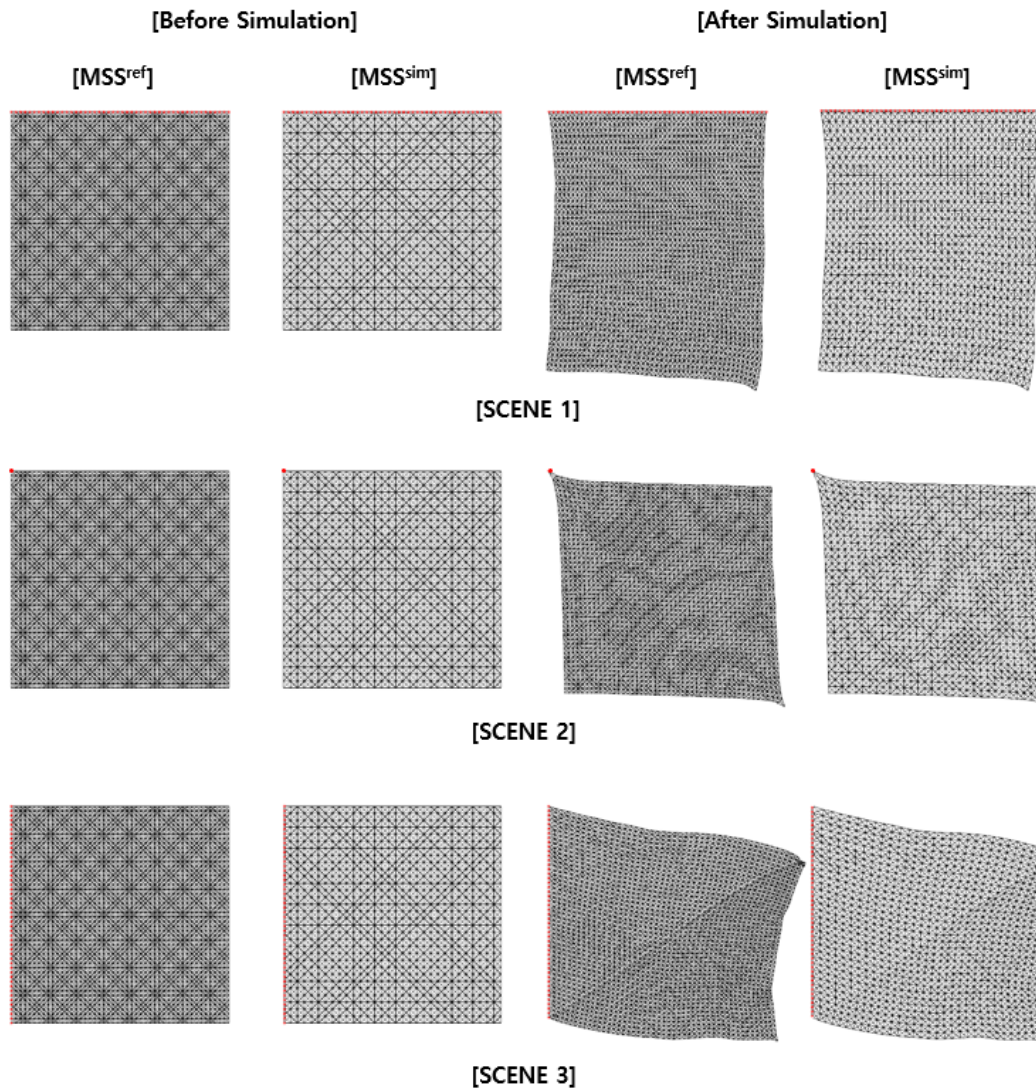



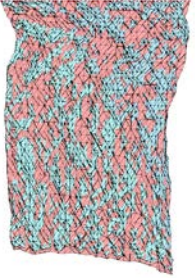
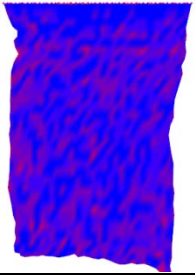
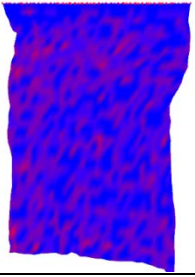
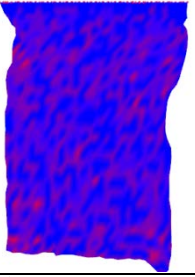
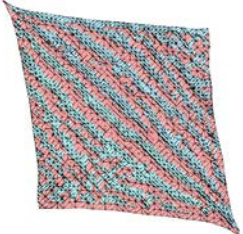
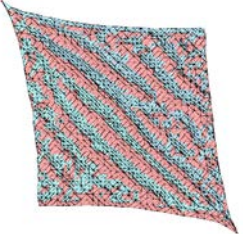
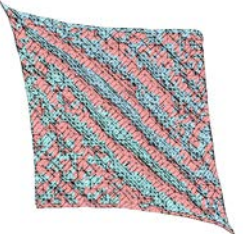
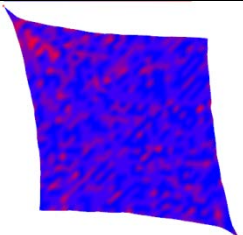
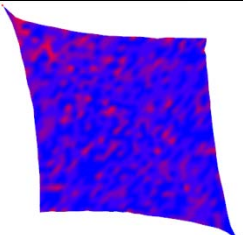
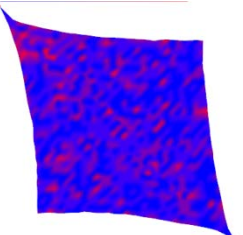


Fig. 11. Snapshot of experimental scenes.
Resolution of MSS^{ref} (64x64) and MSS^{sim} (48x48)

In the experimental tests, we calculate the optimal constant values of K and C values from MSS^{sim} and then calculate the position difference between the simulated MSS^{opt} and MSS^{ref} for performance comparison. We show the results of the experiment in two ways: face rendering with overlaying the result of reference and optimal cloth and color mapping rendering with showing error difference. The following figure is a snapshot showing the highest average displacement difference among 20,000 frames of simulation results for three experimental methods and three scenes. The light blue cloth is MSS^{ref} and the pink cloth is MSS^{opt} . The displacement difference of each node is mapped from min (blue color) to max (red color). The two cloth models have to behave similarly through our experiments.

<i>Applying same Property</i>	<i>Applying BCD method</i>	<i>Applying GKD method</i>
		
		
		
Fig. 12. Experimental Result : Scene 1 (MSS^{ref} :64x64, MSS^{opt} :48x48). Average of displacement difference(same material property:17.65%, BCD:12.51%, GKD:11.66%)		
		
		
Fig. 13. Experimental Result : Scene 2 (MSS^{ref} :64x64, MSS^{opt} :48x48). Average of displacement difference(same material property :13.12%, BCD:9.75%, GKD:8.86%)		

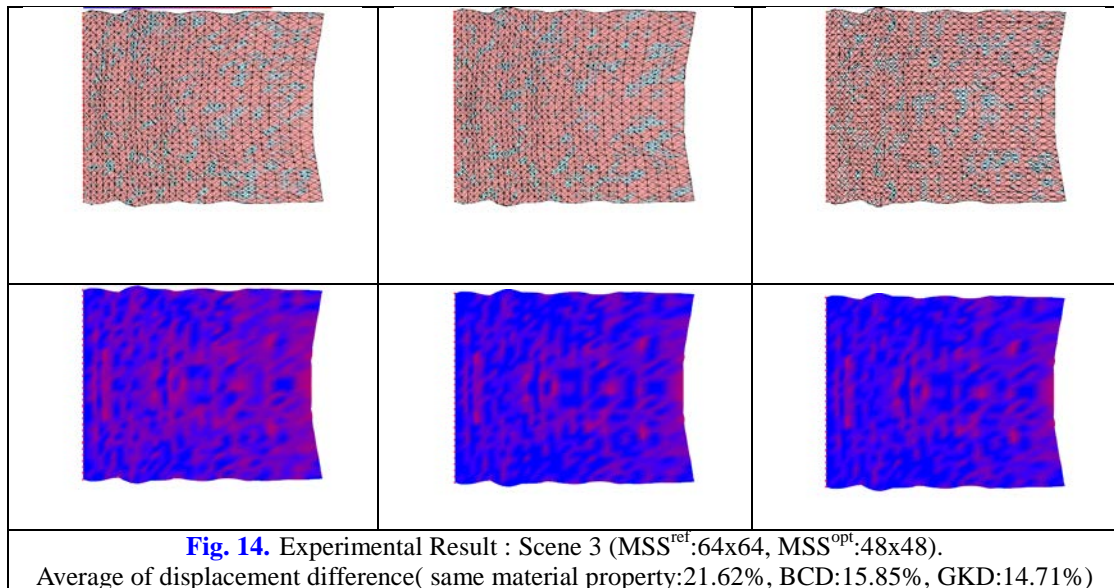


Fig. 12-Fig. 14 show the simulation results of relatively low resolution of MSS^{opt} using the material properties obtained from the experiment and the high resolution of MSS^{ref} . We performed blue to red color mapping through the minimum and maximum values of the displacement differences. Top of **Fig. 12-Fig. 14** are triangle rendering result and bottom row of **Fig. 12-Fig. 14** are color mapping rendering results. The behavior of MSS^{opt} model shows similar motion as MSS^{ref} model. However, we can check that the displacement difference of two cloth models in each simulation scene. The red part at the bottom row of **Fig. 12-Fig. 14** is a region with a relatively high displacement difference.

We performed experimental tests to compare mesh's behavior using BCD that use only three-node information of barycentric triangle and GKD that defined use gaussian distribution function. We also compared the mesh's behavior with the applying same material property. The results of the comparison between MSS^{ref} and MSS^{opt} displacement difference ratio are shown in the **Table 1**.

Table 1. Comparison of displacement difference with three methods

Simulation Node Count (N*M)	Reduction Rate(%)	Displacement error with applying same material property	Displacement error with applying BCD	Displacement error with applying GKD
60 * 60	12	6.56	4.47	4.36
56 * 56	23	9.68	5.35	5.16
52 * 52	34	12.93	6.11	5.89
48 * 48	44	16.81	6.93	6.65
44 * 44	53	20.25	7.43	7.12
40 * 40	61	26.93	11.22	10.48

Experimental results show displacement difference applying same material property is about 6.5% to 27%. however, displacement difference of applying BCD is about 4.5% to 11.2%, while that of applying GKD is about 4.3% to 10.5%. This indicates that GKD improved by about 2.5% to 7% over BCD and showed better performance than applying same material property. Better results are obtained when GKD is applied to approach more nodes than BCD. Experimental results show displacement difference applying same material property is about 6.5% to 27%. however, displacement difference of applying BCD is about 4.5% to 11.2%, while that of applying GKD is about 4.3% to 10.5%. This indicates that GKD improved by about 2.5% to 7% over BCD and showed better performance than applying same material property. Better results are obtained when GKD is applied to approach more nodes than BCD. Experimental results indicate the simulation can be performed within the maximum displacement difference ratio of 10% even if the resolution of model can be reduced by about 50% or more. This indicates that there is no significant difference in behavior when simulating a model with fewer nodes. It also indicates that the computation speed of the cloth model can be greatly reduced. It also shows that it can be widely used for contents requiring real-time performance such as medical system, VR, AR, and game.

7. Conclusions

Although the mass-spring system based cloth simulation shows high accuracy as the cloth resolution increases, the real-time simulation is difficult because of the large amount of computation. In this paper, we are dealing with the study of moving the exact motion of a high resolution model to a low resolution model. For this we have presented an estimation method of material properties for MSS to provide coherent behaviors between different resolutions of simulated cloth models. We used the Barycentric coordinate method to calculate the position of the MSS^{sim} and to estimate the optimal K and C values to ensure similar behavior. We implemented a low resolution cloth model that showed similar behavior to the high resolution cloth model in three scenes. The experimental tests compare the displacement difference between MSS^{opt} and MSS^{ref} using K and C values estimated by each method. For this, we implemented the 3 methods: setting the same material property, using three nodes of Barycentric coordinates information (BCD) and the applying Gaussian distribution function (GKD). Experimental results show the simulation can be performed within the maximum displacement difference ratio of 10% even if the resolution of model can be reduced by about 50% or more. Also, it indicates that the GKD is possible to provide a more suitable MSS for the real-time simulation contents (medical system, VR, AR, game, etc.). It is better to use the proposed BCD, GKD implemented in this paper than the applying same material property. Based on this, we will study on the method of automatically calculating the material properties of the cloth model in the future.

References

- [1] Goldenthal, Harmon, Fattal, Bercovier, Grinspun, "Efficient simulation of inextensible cloth," *ACM Transactions on Graphics (TOG)*, Vol. 26, No. 3, pp. 49, 2007. [Article \(CrossRef Link\)](#)
- [2] Nedel, Thalmann, "Real time muscle deformations using mass-spring systems," in *Proc. of Computer Graphics International (IEEE)*, pp. 156-165, 1998. [Article \(CrossRef Link\)](#)
- [3] Hong, Jung, Choi, Welch, "Fast volume preservation for a mass-spring system," *IEEE Computer Graphics and applications*, Vol. 26, No. 5, pp.83-91, 2006. [Article \(CrossRef Link\)](#)

- [4] V. Baudet M. Beuve F. Jaillet B. Shariat F. Zara "Integrating Tensile Parameters in Hexahedral Mass-Spring System for Simulation," in *Proc. of 29th Int'l Conference Computer Graphics Visualization and Computer Vision (WSCG '09)*, pp. 145-152, 2009. [Article \(CrossRef Link\)](#)
- [5] Rahul Narain, Armin Samii, James F.O'brien, "Adaptive Anisotropic Remeshing for Cloth Simulation," *ACM transactions on graphics (TOG)*, Vol. 31, No. 6, 2012. [Article \(CrossRef Link\)](#)
- [6] Zhang, D., Yuen, M. M, "Cloth simulation using multilevel meshes," *Computers & Graphics*, Vol. 25, No.3, pp.383-389, 2001. [Article \(CrossRef Link\)](#)
- [7] Oshita M., Makinouchi A., "Real-time cloth simulation with sparse particles and curved faces," in *Proc. of The Fourteenth Conference on Computer Animation of Computer Animation (IEEE)*, pp. 220-227, 2001. [Article \(CrossRef Link\)](#)
- [8] Lasseter John, "Principles of traditional animation applied to 3D computer animation," *ACM Siggraph Computer Graphics*, Vol 21, No.4, pp. 35-44, 1987. [Article \(CrossRef Link\)](#)
- [9] Terzopoulos, Platt, Barr, Fleischer, "Elastically deformable models," *ACM Siggraph Computer Graphics*, Vol 21, No.4, pp. 205-214, 1987. [Article \(CrossRef Link\)](#)
- [10] Nealen, Muller, Keiser, Boxeman, Carlson, "Physically based deformable models in computer graphics," *Computer graphics forum*, Vol 25, No.4, pp. 809-836, 2006. [Article \(CrossRef Link\)](#)
- [11] Xavier Provot, "Deformation constraints in a mass-spring model to describe rigid cloth behaviour," *Graphics interface*, pp. 147-147, 1995. [Article \(CrossRef Link\)](#)
- [12] David Baraff, Andrew Witkin, "Large Steps in Cloth Simulation," in *Proc. of the 25th annual conference on Computer graphics and interactive techniques*, pp. 43-54, 1998. [Article \(CrossRef Link\)](#)
- [13] Desbrun, Mathieu, Peter Schröder, and Alan Barr, "Interactive animation of structured deformable objects," *Graphics Interface*, Vol 99, No.5, pp. 10, 1999. [Article \(CrossRef Link\)](#)
- [14] Liu, Bargteil, Obrien, Kavan, "Fast simulation of mass-spring systems," *ACM Transactions on Graphics (TOG)*, Vol 32, No.6, pp.214, 2013. [Article \(CrossRef Link\)](#)
- [15] Debunne, Desbrun, Barr, Cani, "Interactive multiresolution animation of deformable models," *Computer Animation and Simulation 1999, Springer*, pp. 133-144, 1999. [Article \(CrossRef Link\)](#)
- [16] Bhat, Kiran, Twigg, Hodgins, Khosla, Popovic, Seitz, "Estimating cloth simulation parameters from video," in *Proc. of 2003 ACM SIGGRAPH - Eurographics symposium on Computer animation. Eurographics Association*, 2003. [Article \(CrossRef Link\)](#)
- [17] Bayraktar, Serkan. "Simulating cloth behavior by using mass-spring networks," *Diss. bilKent university*, 2002. [Article \(CrossRef Link\)](#)
- [18] Teschner, Heidelberger, Muller, Pomerantes, Gross. "Optimized Spatial Hashing for Collision Detection of Deformable Objects," *Vmv*. Vol. 3. 2003. [Article \(CrossRef Link\)](#)
- [19] Robert Bridson, Ronald Fedkiw, John Anderson, "Robust Treatment of Collisions, Contact and Friction for Cloth Animation," *ACM SIGGRAPH 2005 Courses*, 2005. [Article \(CrossRef Link\)](#)
- [20] Hossein Tehrani Niknejad, Akihiro Takeuchi, Seiichi Mita, "On-road multivehicle tracking using deformable object model and particle filter with improved likelihood estimation," *IEEE Transactions on Intelligent Transportation Systems*, Vol 13, No. 2, pp.748-758, 2012. [Article \(CrossRef Link\)](#)
- [21] Alex X. Lee, Abhishek Gupta, Henry Lu, "Learning from multiple demonstrations using trajectory-aware non-rigid registration with applications to deformable object manipulation," in *Proc. of Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on. IEEE*, pp.5265-5272, 2015. [Article \(CrossRef Link\)](#)



Nak-Jun Sung received the BS degrees in Computer Software Engineering from Soonchunhyaung University in 2016. He also received MS in Dept of Computer Science from Soonchunhyaung University in 2018. Now he is undertaking a Ph.D of computer science courses as a member of the computer graphics lab at Soonchunhyang University. His research interests are computer graphics, GPGPU, VR (Virtual Reality) and AR (Augmented Reality)

E-mail : njsung@sch.ac.kr



Shane Transue received his B.S. from the Metropolitan State University of Denver in Computer Science with a minor in Mathematics. He received his M.S. from the University of Colorado Denver in Computer Science with a research focus in computer graphics and physical simulation. His research focus includes physical simulation of deformable objects, animation, software engineering, interactive tool design, procedural geometry, and 3D scanning/printing. In addition to these core research topics, several of his projects explore the fields of artificial intelligence, programming language design, human computer interaction, game design, and scientific visualization.

E-mail : shane.transue@ucdenver.edu



Min Sang Kim received the BS degrees in Computer Software Engineering from Soonchunhyaung University in 2017. He is currently pursuing master's degree at the Department of Computer Science, Soonchunhyang University, in Asan, Korea. His current research interests include the areas of physically based simulation, etc.

E-mail : ben399399@sch.ac.kr



Yoo-Joo Choi is a professor in Department of Newmedia at Seoul Media Institute of Technology(SMIT), Korea. She received her M.S. and Ph.D. degrees in Computer Science from Ewha Womans University in 1991 and 2005, respectively. She was a researcher at R&D Department of KCI Co. and POSDATA Co. in Korea between 1991 and 1999. Her research interests include image processing, computer vision, computer graphics, augmented reality and human-computer interaction.

E-mail : yjchoi@smit.ac.kr



Min-Hyung Choi received his M.S. and Ph.D. from University of Iowa in 1996 and 1999 respectively. He joined the Computer Science and Engineering Department at the University of Colorado at Denver in 1999. Currently he is the Director of Computer Graphics and Virtual Environments Laboratory. His research interests are in Computer Graphics, Scientific Visualization and Human Computer Interaction with an emphasis on physically-based modeling and simulation.

E-mail: min.choi@ucdenver.edu



Min Hong received his B.S. from Soonchunhyang University. He received his M.S. from University of Colorado at Boulder, and Ph.D. received from University of Colorado at Denver and Health Sciences Center in 1995, 2001, and 2005 respectively. Currently he is the Department Chair of Computer Software Engineering and Director of Computer Graphics Laboratory in Soonchunhyang University. His research interests are in computer graphics, physically-based modeling and simulation, bioinformatics, and image processing related applications.

E-mail: mhong@sch.ac.kr