

國立交通大學

網路工程研究所

碩士論文



研究 生：劉政旻

指 導 教 授：黃 世 強 教 授

中 華 民 國 一 百 年 七 月

鑲嵌固體且具不可延展特性的布料模擬
Simulation of Inextensible Cloth Embedded with Rigid Bodies

研究 生：劉政旻

Student : Cheng-Min Liu

指 導 教 授：黃 世 強

Advisor : Sai-Keung Wong



Computer Science

July 2011

Hsinchu, Taiwan, Republic of China

中華民國一百年七月

Simulation of Inextensible Cloth Embedded with Rigid Bodies

Student: Cheng Min Liu

Advisor: Dr. Sai-Keung Wong

Institute of Computer Science and Engineering
College of Computer Science
National Chiao Tung University



Most textiles do not stretch noticeably under their own weight. Simulating inextensible cloth is consequently attracted much attention. The styles of cloth are not monotonous any more. Cloth designers can design to embed some objects onto the textiles. In this thesis, we develop a novel hybrid framework for simulating inextensible cloth with embedded objects. We iteratively perform fast projection and shape matching so that the strain of cloth is not larger than a predefined limit and the embedded regions of cloth are maintained their initial shapes.

Acknowledgments

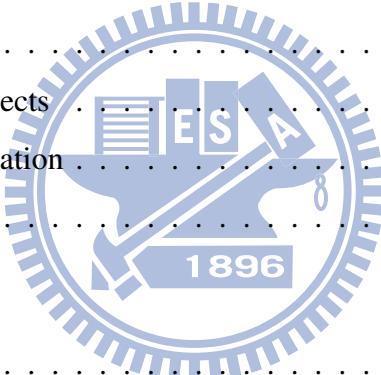
I would like to thank my advisor, Professor Sai-Keung Wong, for his guidances, inspirations, and encouragements. I thank to all my colleagues : Jau-An Yang, Hong-Xuan Ji, Tsung-Shian Huang and Ta-En Chen for their helps and discuss with me on research issues, my senior colleagues : Yueh-Tse Chen, Jyh-Shyang Chang, Ying-Tsung Li, Hsin-Hsiao Lin and Kuang-Wei Fu for their technical supports and useful information, and my junior colleagues: Yu-Chun Cheng, Wei-En Chen, Chiao-Chin Yeh, Kuan-Chen Lin, Yi-Chen Chen, Shih-Wei Fang and Shing-Yeu Lii for their kind assistances and encouragements. Lastly, I would like to thank my parents for their love, and support. Without their love, I could not pass all the frustrations and pain in my life.



Contents

1	Introduction	1
1.1	Contributions	2
1.2	Organization of the thesis	2
2	Related Work	4
2.1	Cloth Simulation	4
2.2	Collision handler	6
3	Introduction to fast projection	7
3.1	Fast projection	7
3.2	Lagrange multiplier	8
4	Introduction to shape matching	10
4.1	Notation	10
4.2	Algorithm of shape matching	11
5	Method	12
5.1	Preprocessing Stage	12
5.1.1	Cluster Constraints	13
5.1.2	Constraints for Fast Projection	14
5.2	Simulation Stage	15

5.2.1	Algorithm Overview	16
5.2.2	Hybird Fast Projection With Shape Matching	16
5.2.2.1	Fast projection (FP)	17
5.2.2.2	Iterative shape mathcing (Iter-SM)	17
5.2.2.3	Output of Hybird Fast Projection With Shape Matching . . .	19
6	Experiments and Results	22
6.1	Experiments	22
6.2	Results	26
6.2.1	Permission Strain:	26
6.2.2	Additional constraints:	27
6.2.3	Object size	28
6.2.4	Number of objects	36
6.2.5	Timing information	51
6.3	Limitations	54
7	Conclusions	55
7.1	Summary	55
7.2	Future Work	55
Bibliography		57



List of Figures

2.1	Difference between [GHF ⁺ 07] is applied and not applied. In (a), the strains at two pinned corners are noticeable when we don't apply fast projection. In (b), the cloth does not visible stretch.	5
3.1	Find x and y to maximize $f(x,y)$ subject to a constraint (shown in red) $g(x,y) = c$	8
4.1	The origin shape of x_i^0 is matched to the deformed shape of x_i . Goal positions g_i is computed. Then the deformed points are moved towards the goal positions.	11
5.1	We remove springs attached to any interior vertex.	14
5.2	Additional constraints are involved at the four surrounding edges e_1, e_2, e_3, e_4 and three types of hypotenuse constraints in the cluster are additional constraints to fix the length of these six edges. The shape of the cluster is approximately maintained by these constraints.	15
5.3	The flow chart of our hybird fast projection with shape matching method.	17
5.4	Two conditions for determining if the whole computation results from the shape matching parts are adopted or not.	20
5.5	The overview of our hybird fast projection with the shape matching method.	21

6.1	2 and 4 additional constraints for strain limit. In (a), there are two constraints fitting the diagonals of the cluster. In (b), we add four constraints on every edge of the cluster.	23
6.2	Three types of 6 additional constraints for strain limit. 6-constraint TYPE I is a combination of 2-constraint and 4-constraint. In 6-constraint TYPE II, we add two constraints connecting the each edge's midpoint to the midpoint of the edge's subtense. In 6-constraint TYPE III, we choose one pair of subtenses in the cluster. The constraints connect from the quarter of one edge to quarter of another.	24
6.3	8 and 10 additional constraints for strain limit. 8-constraint TYPE I is the combination of 2-constraint and 6-constraint TYPE II and 8-constraint TYPE II is combined by 2-constraint and 6-constraint TYPE III. 10-constraint contains all constraints in 8-constraint TYPE I and TYPE II.	24
6.4	Three types of cluster corresponded by different sizes of objects.	25
6.5	Different permission strain results in the middle of the simulation.	26
6.6	Different permission strain results at the last of the simulation.	27
6.7	The cloth is pinned at two corners and embedded by nine single-radius objects.	30
6.8	The cloth is pinned at two corners and embedded by nine double-radius objects.	31
6.9	The cloth is pinned at two corners and embedded by nine triple-radius objects.	32
6.10	The cloth allowed to relax under gravity is embedded by nine single-radius objects and its four corners are catched.	33
6.11	The cloth allowed to relax under gravity is embedded by nine double-radius objects and its four corners are catched.	34
6.12	The cloth allowed to relax under gravity is embedded by nine triple-radius objects and its four corners are catched.	35
6.13	2 rigid objects are embedded on the cloth which is pinned at two corners.	37
6.14	6 rigid objects are embedded on the cloth which is pinned at two corners.	38
6.15	9 rigid objects are embedded on the cloth which is pinned at two corners.	39

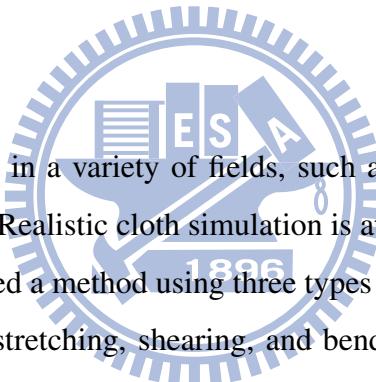
6.16	12 rigid objects are embedded on the cloth which is pinned at two corners.	40
6.17	16 triple-radius rigid objects are embedded on the cloth which is pinned at two corners.	41
6.18	16 double-radius rigid objects are embedded on the cloth which is pinned at two corners.	42
6.19	25 rigid objects are embedded on the cloth which is pinned at two corners.	43
6.20	The cloth is embedded with 2 objects and its four corners are catched.	44
6.21	The cloth is embedded with 6 objects and its four corners are catched.	45
6.22	The cloth is embedded with 9 objects and its four corners are catched.	46
6.23	The cloth is embedded with 12 objects and its four corners are catched.	47
6.24	The cloth is embedded with 16 triple-radius objects and its four corners are catched.	48
6.25	The cloth is embedded with 16 double-radius objects and its four corners are catched.	49
6.26	The cloth is embedded with 25 double-radius objects and its four corners are catched.	50
7.1	The unstable state of our simulation if the strain limit equals to 10%.	56

List of Tables

6.1	The error for each vertex to maintain its original shape when we choose different type of additional constraints.	28
6.2	The average error for each vertex needed to maintain the original shape when we choose different sizes.	29
6.3	Average error for each vertex sufficient to maintain the original shape when we choose different numbers of objects.	51
6.4	The computation time of the results using different experiment setting (1). . . .	53
6.5	The computation time of the results using different experiment setting (2). . . .	54

CHAPTER 1

Introduction



Cloth simulation has been used in a variety of fields, such as video games, films, animation production and textile industry. Realistic cloth simulation is attracted much attention.

Provot et al. [Pro95] proposed a method using three types of springs to form a cloth model. The three types of springs are stretching, shearing, and bending. These springs are used for modeling the internal forces of the cloth model. However, we observe that most fabrics do not stretch noticeably under their own weight. However, for many cloth simulators, enforcing inextensibility of the cloth always degrade the performance. Thus we demand further features to confirm the inextensibility of the cloth. In order to take into account the authenticity and acceptable effectiveness, Goldenthal et al. [GHF⁺07] proposed an iterative fast projection as a global velocity filter to alleviate this issue. We adopt this method as it can model the inextensibility of cloth and it is proven for convergence. However, the cloth models are embedded with solid objects in our system. It is unclear how the fast projection method alone can be applied for simulating such cloth models.

Garment designer usually embed rigid bodies onto textiles to increase the variety of the clothing style. Realistic of interactions between embedded objects and the fabrics attract our

attentions. In general, for parts of the cloth attached with embedded objects, have different movements from other parts. Geometries of the embedded objects should be allowed only to be translated or rotated while the cloth model deforms, but they are not able to change their shape and structure. We are motivated by the approach called “Shape Matching” in Müller et al. [MHTG05]. In [MHTG05], a new mass center and an optimal rotation matrix are computed to recover the original shape of a region.

We propose a method using fast projection [GHF⁺07] to enforce nearly inextensible but the fast projection method cannot maintain shape. Consequently, we add different types of constraints other than those defined in [GHF⁺07]. On the other hand, the shape matching method by Müller et al. [MHTG05] reserves the original shapes of regions in cloth embedded with rigid bodies, and computes the goal positions of these regions in minimum costs. We combine these two approaches and propose a novel hybrid approach for simulating inextensible cloth embedded with rigid bodies.

1.1 Contributions

Our major contributions are stated as follows:

1. Flexibility: different sizes and number of objects can be embedded onto the cloth.
2. A modified fast projection for simulating cloth with small strain: a simple, implicit and fast method for filtering velocities that decouples timestepping and inextensibility.
3. A modified progressive shape matching scheme: A scheme to maintain the shapes of patches on the cloth corresponding to the embedded bodies.

1.2 Organization of the thesis

The following chapters are organized as follows. In the context of the rest of our thesis, we briefly describe the related work in Chapter 2. In Chapter 3, we describe the basis of methods

and symbols in this thesis. The system architecture and ideas are illustrated in detail in Chapter 4. Experimental results are given in Chapter 5. In Chapter 6 we conclude this thesis with a discussion.



CHAPTER 2

Related Work



We review the related work in the areas of cloth simulation and collision handler.

2.1 Cloth Simulation

General Cloth Simulation A deformable cloth surface was simulated using a network of mass particles and springs [BW98, Pro95]. Baraff [BW98] proposed the implicit integration method and generated a large sparse system at each time step and kept the simulation stable. However, a damping force is required for solving the problem of the post-buckling instability. Choi et al. [CK05] used a semi-implicit integration method to solve the problem. Moreover, Bridson and Harmon et al. [BFA02, HVTG08] proposed methods to robustly handle collision response.

Inextensible cloth simulation Adjusting vertex information iteratively could be applied to enforce the inextensibility property of cloth. However, this kind of approaches might not be convergent [BFA02, Pro95]. Ye et al. [Ye08] used strain limit to filter the velocity of the vertex,

but the solution might not stable when using very large time step. Linearized implicit method was proposed to improve the stability of constrained dynamics [GHF⁺07, HCJ⁺05]. Goldenthal et al. [GHF⁺07] adopted constrained Lagrange mechanics to prevent the cloth from stretching too much. In their approach, linearization and efficient calculations can both be performed. We adapt the method by Goldenthal et al. [GHF⁺07] in our approach. Figure 2.1 shows difference between [GHF⁺07] is applied and not applied.

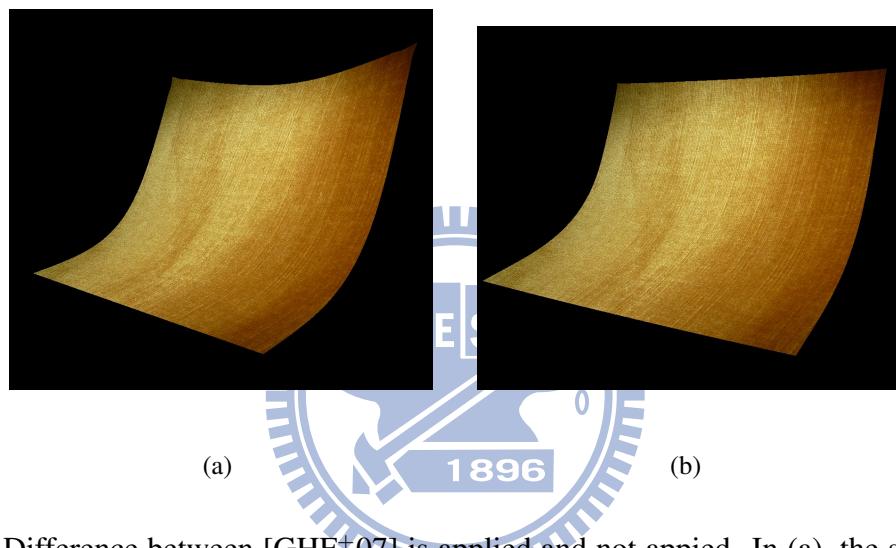


Figure 2.1: Difference between [GHF⁺07] is applied and not applied. In (a), the strains at two pinned corners are noticeable when we don't apply fast projection. In (b), the cloth does not visible stretch.

Shape correspondence Shape matching was presented for simulating non-deformable objects [MHTG05]. The main idea of this method is to simulate the non-deformable objects based on the geometric constraints [MHTG05, RJ07, SOG08, OKN09]. This kind of methods are unconditional stable. Stumpf et al. [SSBT08] proposed an adapted shape matching approach for performing efficient and robust cloth simulation. A combination of two different cluster types is employed to account for high stretching, high shearing and low bending resistance. We adapt Müller's approach in our system and maintain the shape of the regions with embedded objects during simulation.

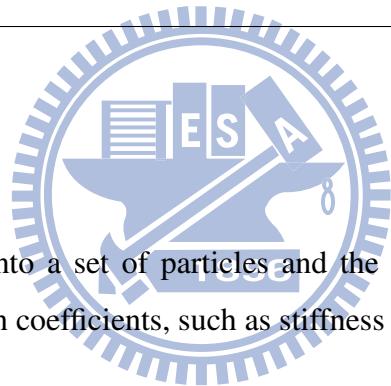
2.2 Collision handler

Collision detection is widely applied in the simulation of numerous deformable objects, such as cloth simulation or rubber material [BHW94, VMT00, GLM96, BW03, CK05, HVTG08, Ye08, VMTF09]. Many kinds of methods which are based on space partition and BVH algorithms are applied to accelerate collision detection [GLM96, KHM⁺98, SKTK95, Ber99]. Nowadays, the hierarchy of k-DOPs[KHM⁺98] is widely used because of its high culling effectiveness. But it should be noted that there is no conclusion about which kinds of k-DOPs is better for different simulation systems.

Continuous collision detection has been applied in the simulation of rigid bodies [RLMK05]. The main breakthrough for deformable objects is that this method can be used to compute the precise information of collision time and collision position. Liu [LKC96] proposed an approach for computing the contact time between two triangles by checking each pair of potentially colliding triangle pair (PCTP). Each test for a feature pair involves solving a cubic equation for the times when the feature pair is coplanar. Subsequently, this method was adopted in [Pro97, BFA02, WB05, HF07].

CHAPTER 3

Introduction to fast projection



A cloth model is discretized into a set of particles and the particles are inter-connected by springs. Each spring has its own coefficients, such as stiffness and initial length.

3.1 Fast projection

Some kind of cloth does not stretch much under the effect of their own weights. We want to simulate this kind of cloth. The approach which is developed based on spring mass system requires a high stiffness of springs. However, using springs of higher stiffness often require smaller time step. An alternative approach is to adopt fast projection [GHF⁺07]. Fast projection(called FP for short in our thesis) is proposed to restrict the textiles so that they can have a small strain, e.g. 0.1%. FP is a method for filtering velocities of particles iteratively for enforcing the elongation constraints between particles. FP computes the implicit constraint directions and then projects velocities of particles approximately onto the constraint manifold. Since the Lagrange multiplier is used for forming the linear systems, we introduce Lagrange multiplier in the following

section.

3.2 Lagrange multiplier

In mathematical optimization, the method of Lagrange multipliers provides a strategy for finding the maxima and minima of a function subject to constraints [Lar]. For instance (see figure 3.1), consider the optimization problem: maximize $f(x, y)$, subject to $g(x, y) = c$.

A new variable λ called a Lagrange multiplier is introduced, and we study the Lagrange function defined by

$$\Lambda(x, y, \lambda) = f(x, y) + \lambda \cdot (g(x, y) - c), \quad (3.1)$$

If $f(x, y)$ is a maximum for the original constrained problem, then there exists a λ such that (x, y, λ) is a stationary point for the Lagrange function.

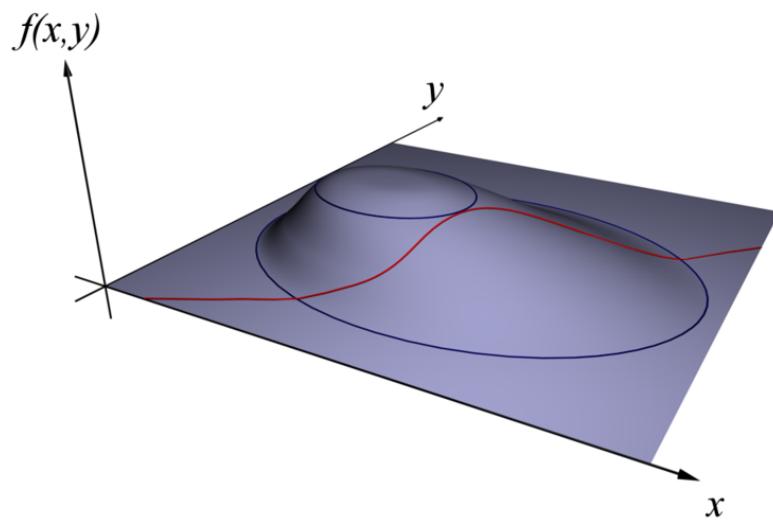
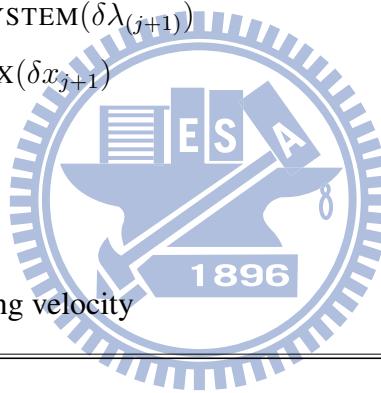


Figure 3.1: Find x and y to maximize $f(x, y)$ subject to a constraint (shown in red) $g(x, y) = c$.

After the introduction to Lagrange multiplier, we briefly introduce the algorithm of FP [GHF⁺07] is shown in Algorithm 3.2.1.

Algorithm 3.2.1: FAST PROJECTION()*Input* : \tilde{v} **comment:** candidate velocity*Input* : \tilde{x} **comment:** known start-of-step position $j \leftarrow 0$ $x_0 \leftarrow \tilde{x} + h\tilde{v}$ **while** EXCEED STRAIN THRESHOLD(x_j)

do $\left\{ \begin{array}{l} (1) \text{SOLVE LINEAR SYSTEM}(\delta\lambda_{(j+1)}) \\ (2) \text{EVALUATE NEXT X}(\delta x_{j+1}) \\ x_{j+1} \leftarrow x_j + \delta x_{j+1} \\ j \leftarrow j + 1 \end{array} \right.$

Output : $\frac{1}{h}(x_j - \tilde{x})$ **comment:** constraint-enforcing velocity

The Equations 3.2 and 3.3 in Algorithm 3.2.1 are stated as follows.

$$h^2(\bigtriangledown C(x_j)M^{-1} \bigtriangledown C(x_j)^T)\delta\lambda_{j+1} = C(x_j) \quad (3.2)$$

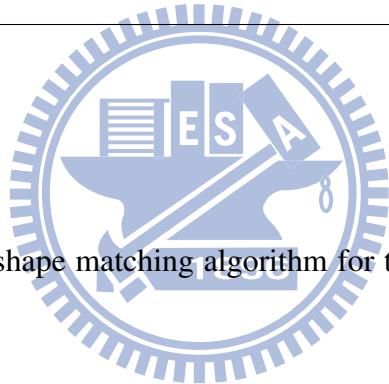
$$\delta x_{j+1} = -h^2 M^{-1} \bigtriangledown C(x_j)^T \delta\lambda_{j+1} \quad (3.3)$$

In equation 3.2, a numerical solver Constrained Lagrangian Mechanics is used for computing the constraint gradient $C(x_j)$. Furthermore, Equation 3.3 is used to find the positions of the modified implicit vertices.

The algorithm computes the vertex positions x_j iteratively if the constraint is not satisfied. The method is treated as a velocity filter and it is used to restrict the stretch of cloth in a permissible range.

CHAPTER 4

Introduction to shape matching



In this chapter, we present the shape matching algorithm for the current state and the original state of the target object.

4.1 Notation

We observe that a rigid body structure is not deformable even when it is embedded on a deformable object, such as cloth. In our thesis, we call the regions which are embedded with rigid bodies as “clusters” and denote them as C_i . We need to map each C_i to its original shape during the simulation while allowing other portion of cloth to be deformed. We introduce Müller et. al [MHTG05] in the following section (see figure 4.1).

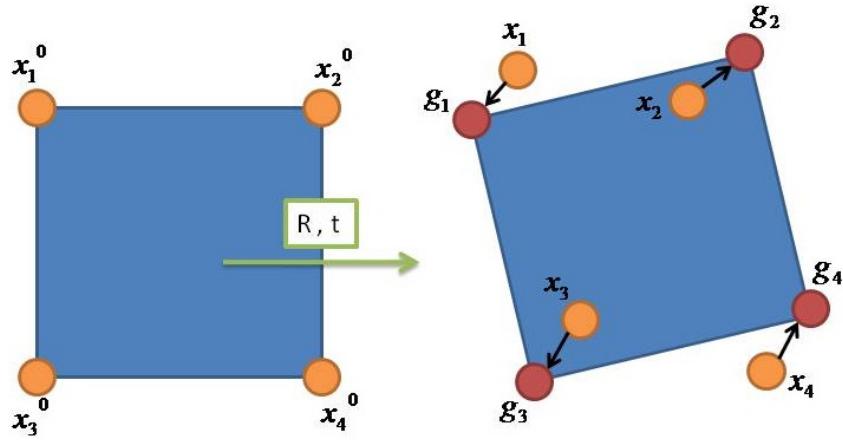


Figure 4.1: The origin shape of x_i^0 is matched to the deformed shape of x_i . Goal positions g_i is computed. Then the deformed points are moved towards the goal positions.

4.2 Algotithm of shape matching

A cluster is composed of n points. The origin positions of these points are denoted as x_i^0 , $i = 1 \dots n$. The mass center of x_i^0 is x_{cm}^0 . After the deformation, the points move to the new positions x_i and the center becomes x_{cm} .

We compute an optimal rotation matrix R and a translation vector t , as shown in figure 4.1. Then we can restore the clusters back to the original shapes and ensure the sum of the distances between the new positions and old positions of points is minimum.

$$g_i = R(x_i^0 - x_{cm}^0) + x_{cm} \quad (4.1)$$

The corrected position g_i of the point i is computed by using equation 4.1. After that, we move each vertex to its goal position. Thereby, C_i is tranformed to its original shape.

CHAPTER 5

Method



In this chapter, we illustrate our hybrid method of fast projection and shape matching. In our approach, two important goals needed to be achieved:

1. Inextensibility of the cloth.
2. Fixed shapes of regions embedded with rigid bodies.

We adopt Goldenthal et al. [GHF⁺07] method to achieve the inextensibility of the cloth. A modified method of [MHTG05] is applied to maintain the fixed shapes of regions embedded with rigid bodies.

There are two stages in our simulation process. First, we will illustrate the constraints which are needed to be enforced during the simulation in section 5.1. In section 5.2, we describe our algorithm during the run time stage.

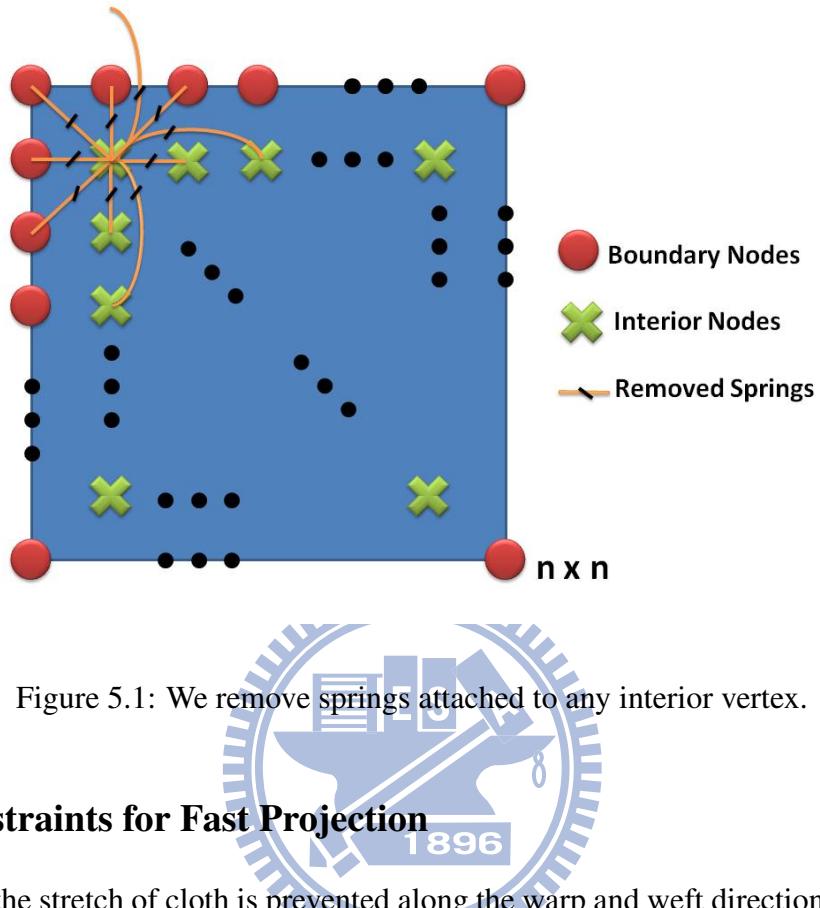
5.1 Preprocessing Stage

Based on observation, we find out not only the regions embedded with rigid bodies but also neighbouring regions around these embedded bodies cannot be deformed. The properties of

these regions should be different from other parts of the cloth. Furthermore, we need to develop a model which is capable of maintaining these undeformed regions. We use a two-dimensional square to approximate these special areas, and define the boundary part of the square as a cluster C_i . Each C_i consists of k vertices horizontally and vertically, respectively. Furthermore, we add several kinds of constraints for each cluster in order to fix the shape. The constraints are described in the following section.

5.1.1 Cluster Constraints

Given a triangular mesh with vertices and edges, constrained dynamics can be developed by a spring mass system. In the system, springs produce stretching forces, shearing forces, and bending forces during the simulation. However, since we embed rigid bodies on the cloth, the cluster corresponds to each object should also be rigid. Thus when the spring forces are computed for interior of the cloth, the vertices inside a cluster should not be affected by any internal forces. So we remove the springs attached to every vertex inside the cluster (see figure 5.1).



5.1.2 Constraints for Fast Projection

In [GHF⁺07], the stretch of cloth is prevented along the warp and weft directions. Hence, inextensible constraints are only added on edges in weft and warp directions. But above-mentioned constraints cannot help us to maintain the original shape of each cluster. Thus, in addition, we add more inextensible constraints which correspond to the edges of each cluster so that the cluster does not change its shape when fast projection is employed. Several types of additional constraints are given in figure 5.2. In next chapter, we will illustrate every type of additional constraint involved in the system in detail.

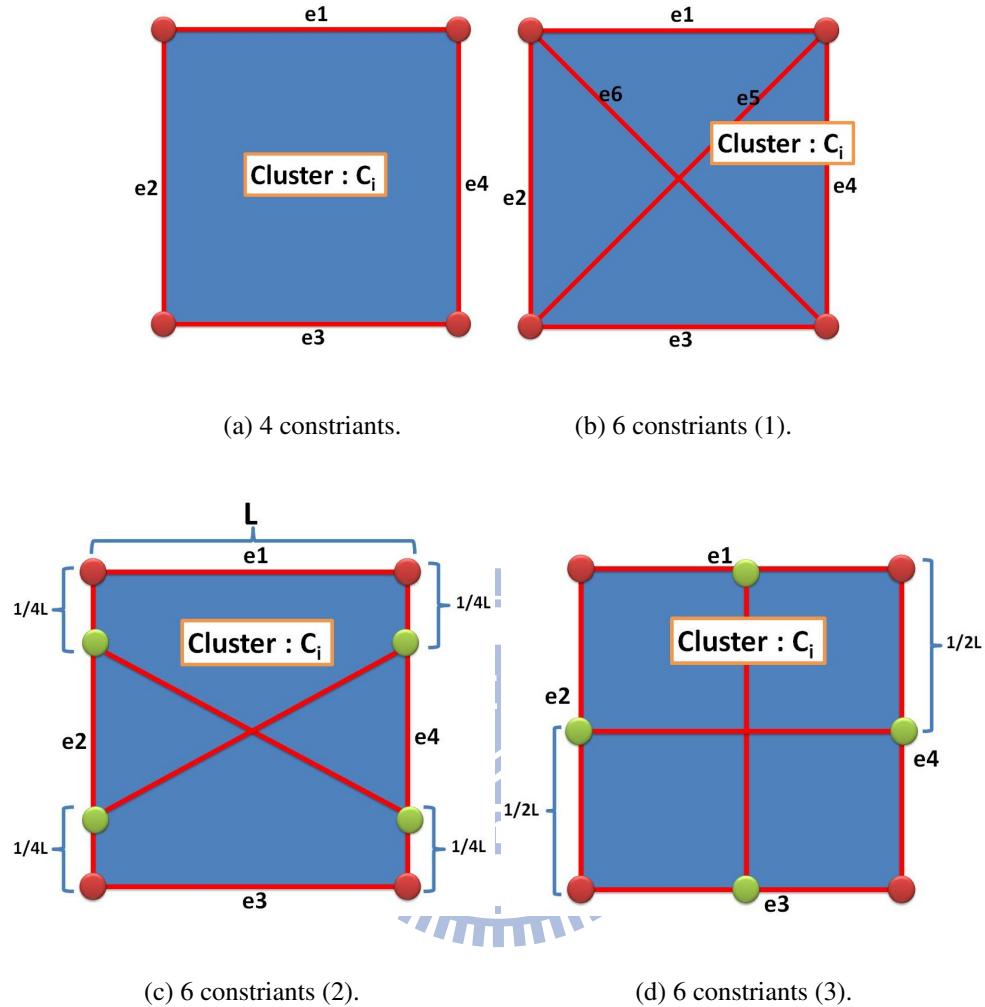


Figure 5.2: Additional constraints are involved at the four surrounding edges e_1, e_2, e_3, e_4 and three types of hypotenuse constraints in the cluster are additional constraints to fix the length of these six edges. The shape of the cluster is approximately maintained by these constraints.

5.2 Simulation Stage

In this section, we describe the simulation algorithm. The important steps in our algorithm are Goldenthal's fast projection scheme and progressive shape matching. We describe them in details in the following sections.

5.2.1 Algorithm Overview

In each frame, we need to finish all the computation in the Algorithm 3.2.1.

Algorithm 5.2.1: ALGORITHM OVERVIEW()

Step1 : COMPUTE EXTERNAL FORCES()

Step2 : COMPUTE INTERNAL FORCE()

Step3 : HYBIRD FAST PROJECTION WITH SHAPE MATCHING()

Step4 : UPDATE PRIMITIVE INFORMATION()

The influence from external forces (e.g. gravity, air drag force, and user defined forces) are computed in step1. In step2, we compute internal forces (i.e. spring forces and viscous drag forces), including stretching, shearing, and bending forces. We choose explicit schemes with small time step for the computations in step1 and step2 to simplify the computation complexity. After that, the velocity of each vertex is adjusted to obtain a nearly inextensible status and each cluster is adjusted to its original shape iteratively in step3. Finally, the primitive information is updated in step4.

5.2.2 Hybird Fast Projection With Shape Matching

In this section, we introduce the details of our hybird fast projection with shape matching (called HFP in following parts of our thesis). Figure 5.3 shows the flow chart of our idea:

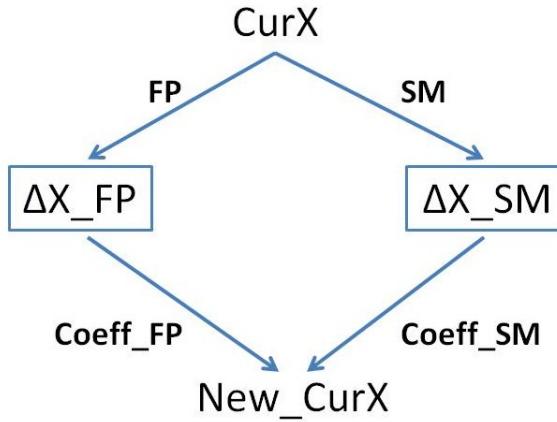


Figure 5.3: The flow chart of our hybrid fast projection with shape matching method.

In each step of HFP, we input the current position of each constrained vertex separately to the fast projection (FP) part and the shape matching (SM) part. Fast projection and shape matching are performed independently (see figure 5.3). Finally we use two coefficients $coeff_{FP}$ and $coeff_{SM}$, $coeff_{FP} + coeff_{SM} = 1$ to sum up the output these two parts.

5.2.2.1 Fast projection (FP)

Following Goldenthal's approach, we perform a velocity filter to enforce the strain limit of all edges of the cloth. We define a threshold ε as the maximum strain of the cloth model. In each iteration, we measure the strain of each edge e_i . The velocity filter is then applied for changing the velocities of all the vertices if the strain of an edge exceeds ε . Finally, the strain constraints of all the edges are satisfied. Since the Lagrange multiplier is used to form the linear systems, the structure of the entire cloth should be considered.

5.2.2.2 Iterative shape matching (Iter-SM)

The shape of each cluster should be maintained. In our thesis, we apply Müller's shape matching scheme [MHTG05]. According to Müller's approach, a new mass center x_{cm} is computed as the translation vector for each cluster C_i . As mentioned in Chapter 4, an optimal rotation matrix

R can be derived by using information of deformed state and undeformed state with minimum cost. Then the goal positions gi of each vertex of the cluster C_i can be computed by using the matrix R and the vector t .

Map to 2D coordinate We find there exists a special situation causing the wrong extraction of the rotation matrix R . In order to explain clearly, here we define a notation “shape vector”. For each vertex of the cluster, the shape vector is equal to the relative vector from the cluster’s barycenter to its current position. We name the shape vector at the first frame the “original shape vector”. The special situation is: for each cluster, if all its shape vectors are parallel to its original shape vectors, then the rotation matrix could not be derived correctly. In this case, we change the coordinate from 3D to 2D and perform a 2D shape matching. We use the method “principal component analysis” (PCA) to find out the fitting plane with bases \vec{u} and \vec{v} of each cluster. Then we map the cluster onto the fitting plane and follow [SD92]’s method to extract the rotation matrix and avoid Jacobi rotation. Finally, we map the goal positions in 2D back to 3D coordinate.

However, adjusting the positions without using impulse is equivalent to adding extra external forces into the system. Since we do the position adjustment movement at every frame, a large number of external force may lead the simulating system to instability. Thus, we use an modified version of shape matching scheme to avoid the situation. Algorithm 5.2.2 is our algorithm for the modified shape matching and we called it the “progressive shape matching (Prog-SM)” in our thesis.

Algorithm 5.2.2: PROGRESSIVE SHAPE MATCHING()

comment: n:number of vertices of each cluster

Input : x_i^0

comment: i=1...n

$x_i \leftarrow x_i^0$

$g_i \leftarrow \text{SHAPEMATCHING}(x_i)$

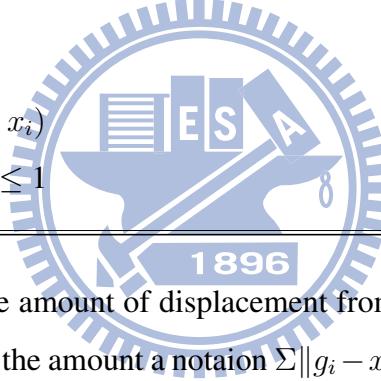
$distance \leftarrow \sum \|g_i - x_i\|^2$

if $distance \leq threshold$

then $\begin{cases} x_i \leftarrow g_i \\ break; \end{cases}$

else

then $\begin{cases} x_i \leftarrow x_i + \alpha(g_i - x_i) \\ \text{comment: } 0 \leq \alpha \leq 1 \end{cases}$



For each cluster, we sum the amount of displacement from the original position to its goal position of each vertex and give the amount a notaion $\sum \|g_i - x_i\|^2$. We define another notation ε as a threshold to bound the value of $\sum \|g_i - x_i\|^2$. If $\sum \|g_i - x_i\|^2 \leq \varepsilon$, we scale the displacement by a scalar α to adjust the vertex position. After iteratively testing the condition and position adjusting, we can have the original shapes at the end of this pass.

5.2.2.3 Output of Hybird Fast Projection With Shape Matching

In figure 5.5, for each vertex, in order to enforce inextensibility of the edge attached to it, the fast projection part computes a goal position and an relative vector ΔX_{FP} for the vertex to move toward to. Similarly, the shape matching part also computes a goal position and an relative vector ΔX_{SM} to help maintain the shape of the cluster the vertex belongs in. In [GHF⁺07], Goldental's fast projection doesn't conform to realistic dynamics. On the other hand, SM only consider the origin shape of the cluster. Thus at most of the time, the influence of FP or SM

part conflict with another one. Thus if we directly superimpose the result of one of the above two part on another part, there may occur unanticipated appearances of simulation. Therefore, more stronger conditions are required.

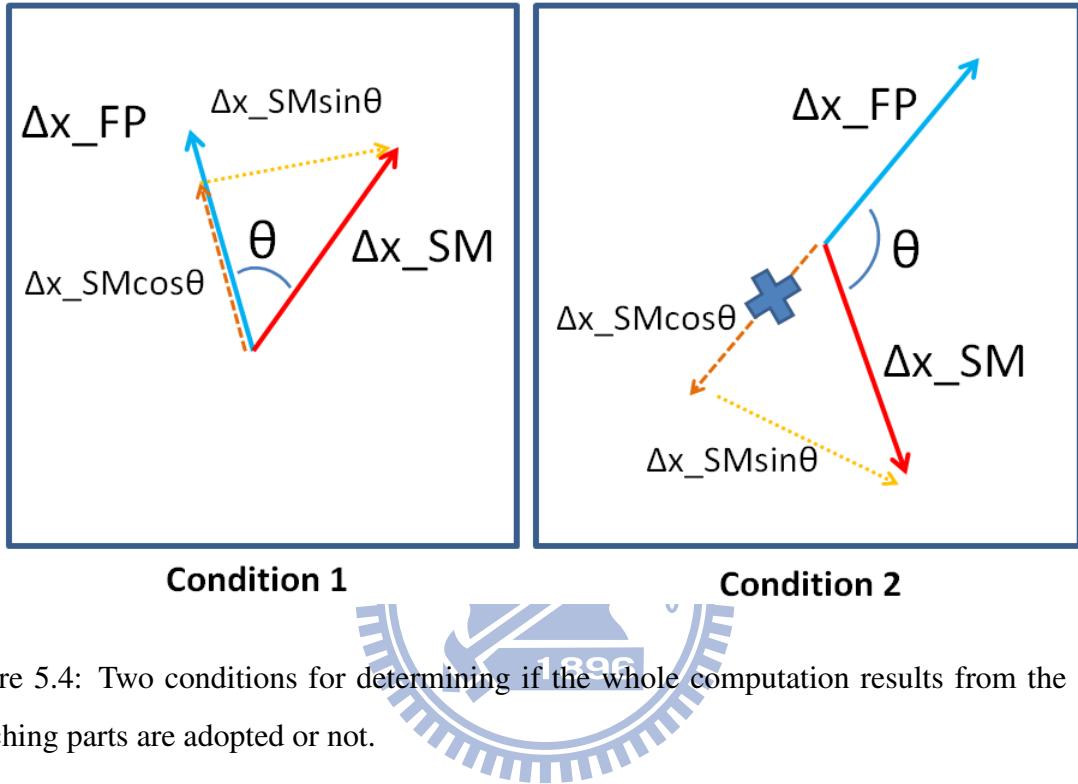


Figure 5.4: Two conditions for determining if the whole computation results from the shape matching parts are adopted or not.

In [GHF⁺07], FP has been proved as a stable method. So when we superimpose the outputs, we give first place to FP. In other words, we adopt the part of computation results from SM conditionally or even abandon the whole part from SM. In our thesis, for each cluster, we compute the direction where the cluster toward to by summing up ΔX_{FP} and ΔX_{SM} of each vertex. We compare the direction of the summation of ΔX_{FP} with the summation of ΔX_{SM} . As showing in figure 5.4, “ θ ” represents the angle between the summation of ΔX_{FP} and the summation of ΔX_{SM} . We define two conditions here. The first one of the conditions is that, if the summation of ΔX_{FP} has the opposite direction with the summation of ΔX_{SM} , then we directly abandon the computation results from SM. We use the notation “cond.1” to represent the first condition in the following part of our thesis. The condition is too strong so the probability of abandonment is more than half. Another weaker condition is defined as, if the summation of

ΔX_{FP} has the opposite direction with the summation of ΔX_{SM} , then we decompose the results from SM part into two directions of $\sin\theta$ and $\cos\theta$ and ignore the part in the $\cos\theta$ direction. The comparison of the influences from above two conditions is listed in next chapter. We use the notation “cond.2” to represent the second condition in the following part of our thesis.

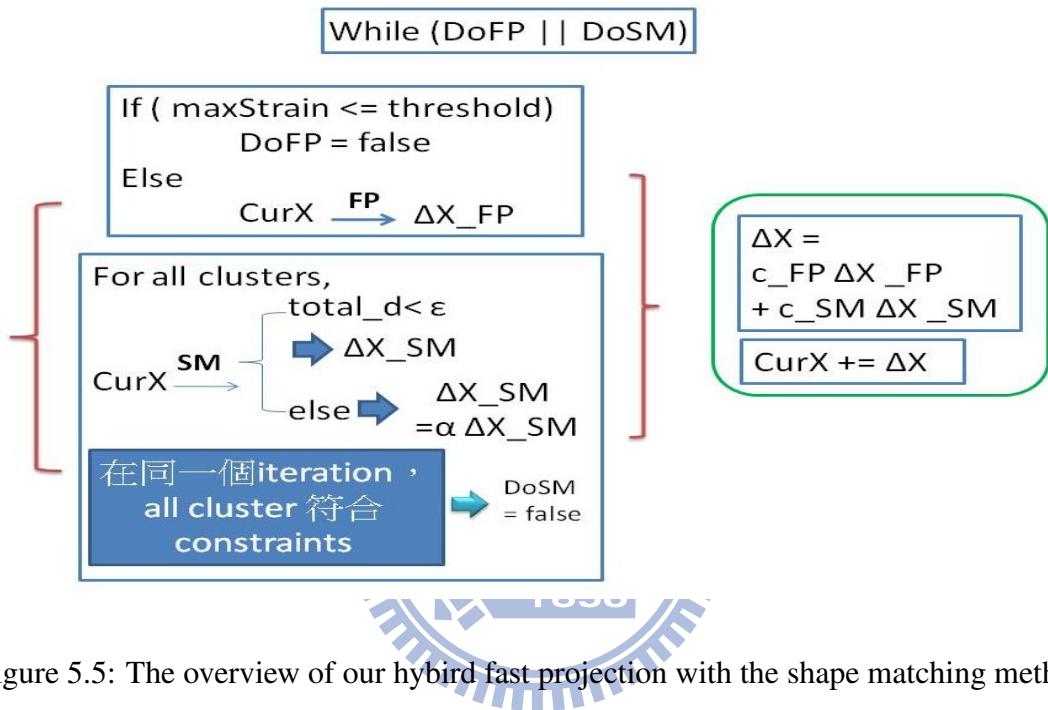
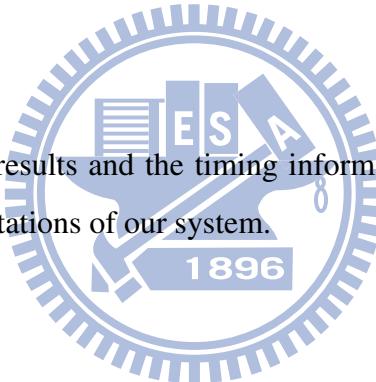


Figure 5.5: The overview of our hybrid fast projection with the shape matching method.

CHAPTER 6

Experiments and Results



In this chapter, we present the results and the timing information of different type of experiments, and then discuss the limitations of our system.

6.1 Experiments

We create the cloth model which consists of 1600 vertices and 3042 faces for the experiments. The dimension of the cloth is 200 timed by 200. Two experiments are designed. One is to allow the cloth which is pinned at two corners to relax under gravity. We call the experiment “EXP1”. Another is to catch the four corners of the cloth and let the cloth relax under gravity. We call the experiment “EXP2”.

In our experiments, we consider five different factors which are listed below:

1. Permission Strain:

We compare three simulations with progressively smaller permission strain. We choose 10%, 1%, and 0.1% for our candidates. EXP1 is adopted in this part of experiments. In this part, we embed 25 objects and each object corresponds to a patch that consists of 25

vertices.

2. Additional constraints:

Since it might be better if clusters are uniformly surrounded by additional constraints, we follow the idea to design the additional constraints. We design eight types of additional strain constraints into our system (see figure 6.1, 6.2 and 6.3). The yellow edges in figure 6.1, 6.2 and 6.3 represent the additional constraints and the red edges represent the edges of the cluster. By enforcing the constraints, for each cluster, its original shape could be approximated at the end of each frame during the simulation. We adopt EXP1 and EXP2 in this part of experiments and compare the results between each other.

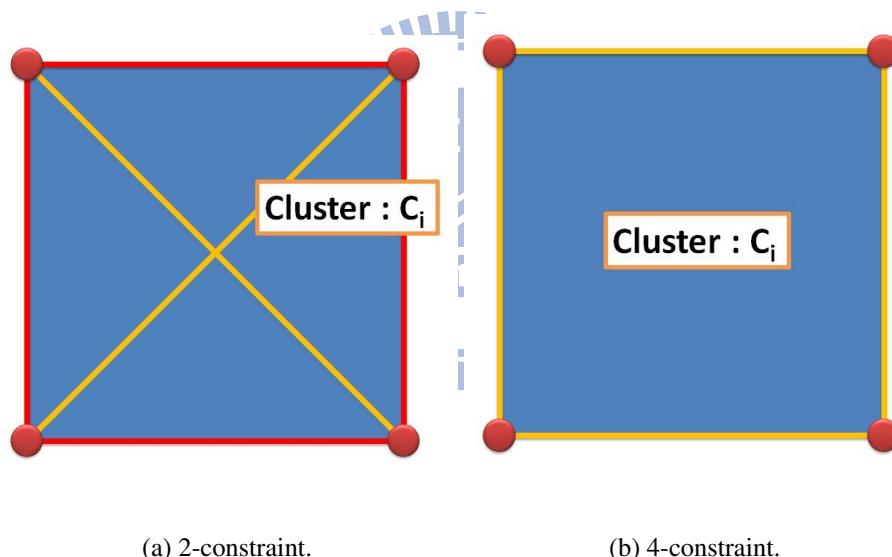


Figure 6.1: 2 and 4 additional constraints for strain limit. In (a), there are two constraints fitting the diagonals of the cluster. In (b), we add four constraints on every edge of the cluster.

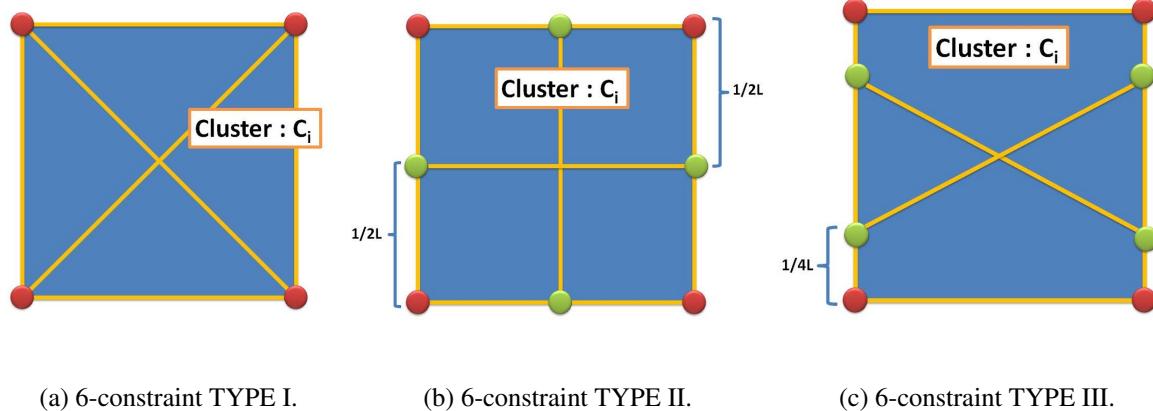


Figure 6.2: Three types of 6 additional constraints for strain limit. 6-constraint TYPE I is a combination of 2-constraint and 4-constraint. In 6-constraint TYPE II, we add two constraints connecting the each edge's midpoint to the midpoint of the edge's subtense. In 6-constraint TYPE III, we choose one pair of subtenses in the cluster. The constraints connect from the quarter of one edge to quarter of another.

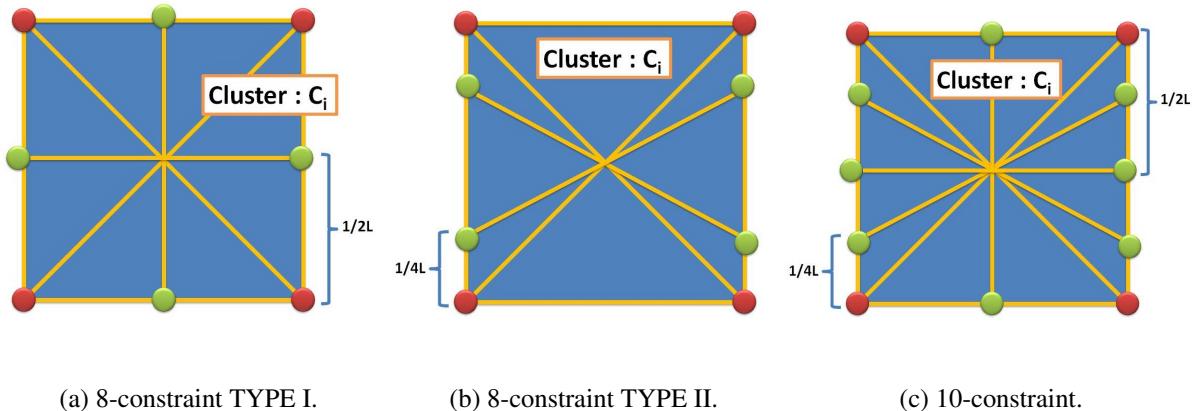


Figure 6.3: 8 and 10 additional constraints for strain limit. 8-constraint TYPE I is the combination of 2-constraint and 6-constraint TYPE II and 8-constraint TYPE II is combined by 2-constraint and 6-constraint TYPE III. 10-constraint contains all constraints in 8-constraint TYPE I and TYPE II.

3. Object Size:

The next experiment compares the influence caused by different sizes of embedded objects. We define three notations “single-radius”, “double-radius” and “triple radius” to represent three sizes of embedded objects. The object with single-radius corresponds to a cluster consisting of 4 vertices on the cloth. Similarly, the object with double-radius and triple radius correspond to a cluster consisting of 25 and 49 vertices, respectively (see figure 6.4). In figure 6.4, the red circles represent the boundary vertices and the green ones represent the interior vertices. In this part of the experiment, nine rigid objects were embedded onto the cloth. We adopt EXP1 and EXP2 in this part of experiments here.

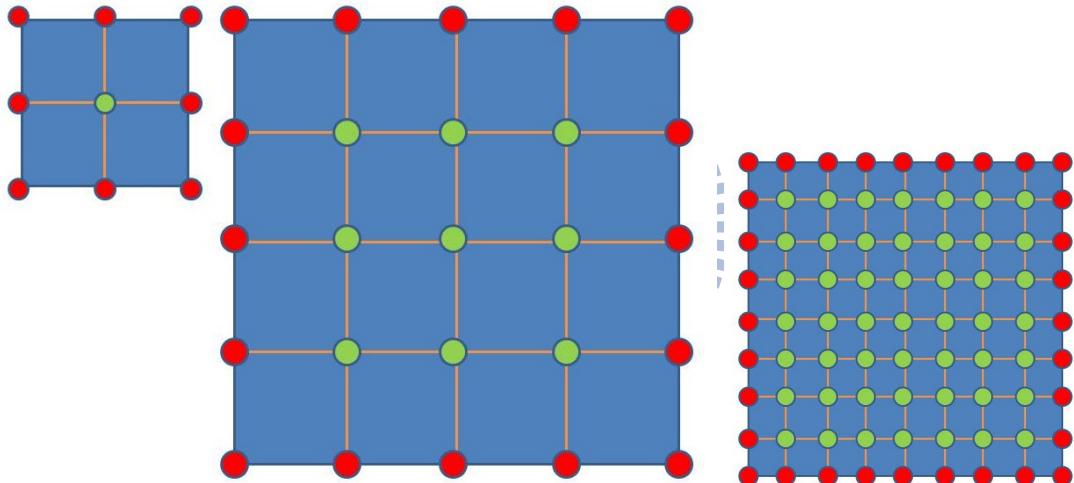


Figure 6.4: Three types of cluster corresponded by different sizes of objects.

4. Object Number:

We embed 2, 6, 9, 12, 16 and 25 objects onto the cloth. Each object corresponds to a region that consists of 49 vertices on the cloth. We adopt EXP1 and EXP2 in this part of experiments here.

6.2 Results

Our system spends probably 3-30 minutes running simulations (the cloth is parallel to xz-plane at the beginning, and approximately parallel to xy-plane in the end) in all designed experiments. There are 1471 frames for the entire simulation.

6.2.1 Permission Strain:

Figure 6.5 and 6.6 show the simulating results for a variety of permission strains. Figure 6.5 shows the state of the cloth in the middle of the simulation. Different between figure 6.5(a), (b) and(c) is not noticeable at the stage. The reason is that, for each constrained edge, we compute the distance from one end point to another at the first frame and set the distance as the initial length. Since we don't scale the initial length by a scalar bigger than 1.0, our cloth look stiffer. But there is still some difference to be observed when we zoom in the camera closer to the cloth. Figure 6.6 shows the results after we zoom at the last frame.

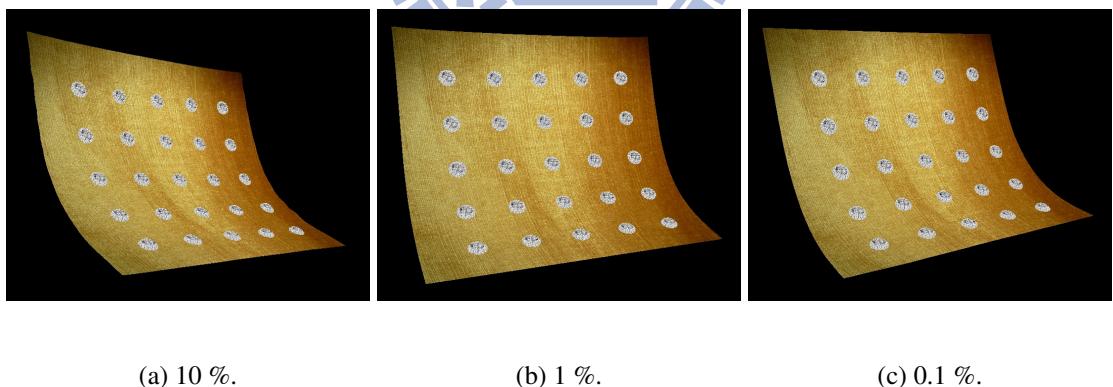


Figure 6.5: Different permission strain results in the middle of the simulation.

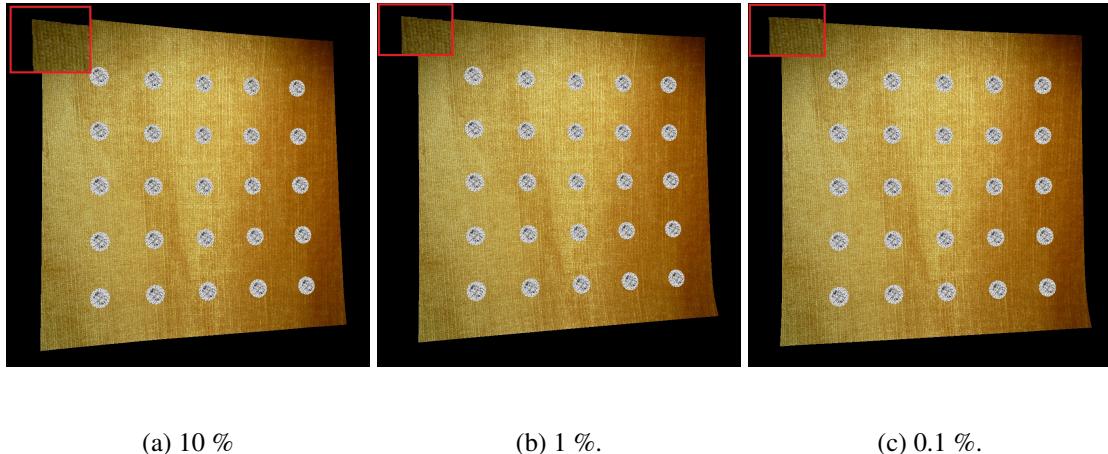


Figure 6.6: Different permission strain results at the last of the simulation.

6.2.2 Additional constraints:

We add eight types of strain constraints (see figure 6.1, 6.2 and 6.3) as mentioned above into every cluster.

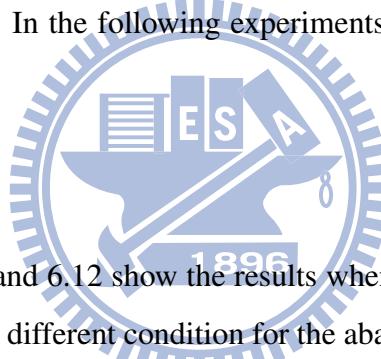
For each cluster, we measure how similar between our simulating result and its original shape. We use Müller et. al's shape matching sheme [MHTG05] as the measuring tools. Shape matching is performed to every cluster again and measure the distance for each vertex that still has to move from its current position x_i to achieve its goal position g_i . We take average of $\sum \|g_i - x_i\|^2$ as the average error for all vertices that belong in any cluster.

We give the experiment that relaxes the cloth that pinned two corners a notation “EXP1”, and another one that catches the four corners “EXP2”. Table 6.1 shows the average error for each vertex when we choose different types of additional constraints in EXP1 and EXP2. In each part of the experiment, we embed only one object corresponding to the cluster consists of 25 vertices onto the cloth.

Type	none	2	4	6-1	6-2	6-3	8-1	8-2	10
EXP1	5.03	5.06	5.00	4.96	4.95	5.07	4.99	5.02	4.97
EXP2	5.48	5.47	5.46	4.54	5.02	5.39	4.62	4.70	5.04

Table 6.1: The error for each vertex to maintain its original shape when we choose different type of additional constraints.

In table 6.1, we can see that more additional constraints don't bring us less error. In EXP1, there is not much difference in every experiment. In EXP2, more additional constraints also confirm the less error. But if the additional constraints are not chosen well, the result may be worse. 6-constraint TYPE I has the least error than others and 8-constraint TYPE I and TYPE II look like acceptable choices. In the following experiments, we add 6-constraint TYPE I in the following experiments.



6.2.3 Object size

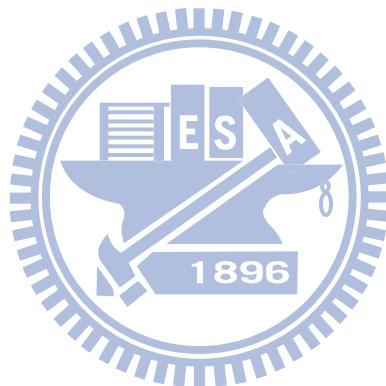
Figure 6.7 ,6.8, 6.9, 6.10 ,6.11 and 6.12 show the results when we embedded different sizes of objects onto the cloth and adopt different condition for the abandonment of the results from the shape matching part. In the experiment, we use 0.1% as the strain limit. In figure 6.7 and 6.10, the embedded object belongs to the single-radius type. We use double-radius in figure 6.8 and 6.11, and triple- in figure 6.9 and 6.12. We observe that, when the size is increased, the regions between one cluster and its neighboring ones look stiffer and stiffer. In figure 6.9 and 6.12, we can even see a obvious fold in the middle of two clusters.

Table 6.2 lists the average error for each vertex when objects with different size are embeded and different condition for the abandonment of the computation result from the shape matching part. We could see that if we increase the object size, then the average error decreases accordingly. Since we applied shape matching only on the boudary of every cluster and compute the positions of the interior vertices by interpolation, the error was scattered to every interior vertices, so the reduction was anticipated. When cond.2 is adopted, we derive the less error. But

the cloth is stabler if we adopt cond.1. Furthermore, in EXP2, the cloth even move toward the top right direction when we embed nine triple-radius objects.

	EXP1		EXP2	
	cond.1	cond.2	cond.1	cond.2
$r = 1$	5.19	2.88	5.98	5.44
$r = 2$	4.99	1.62	5.91	5.82
$r = 3$	2.71	1.27	6.23	5.60

Table 6.2: The average error for each vertex needed to maintain the original shape when we choose different sizes.



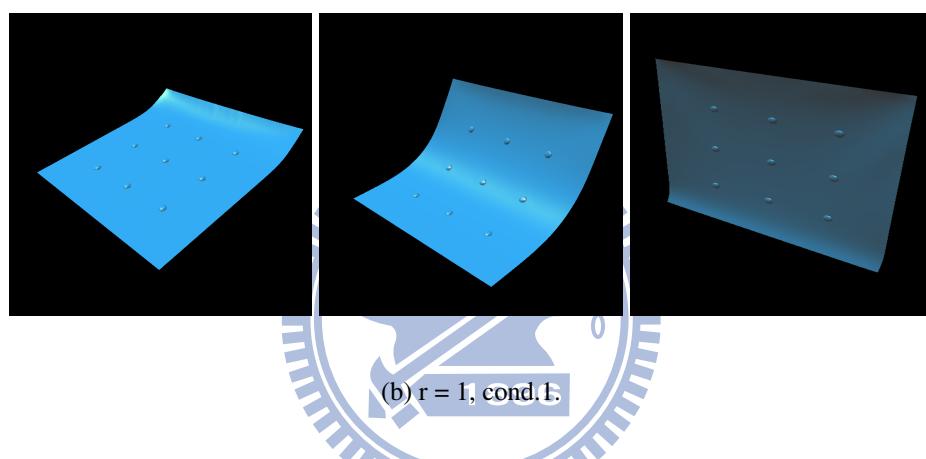
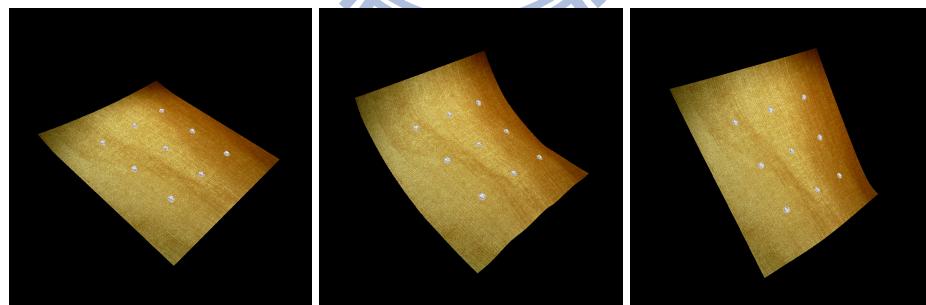
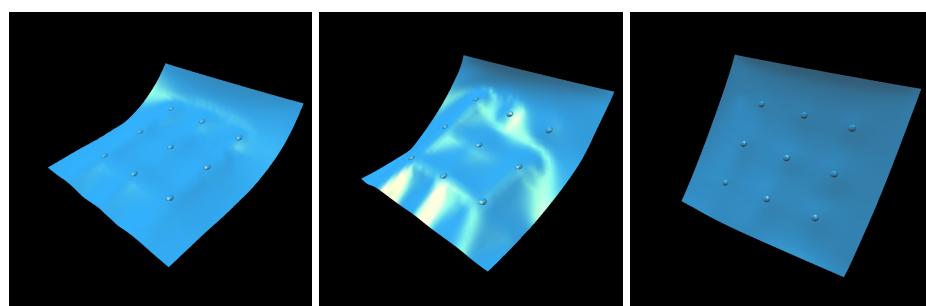
(a) $r = 1$, cond.1.(b) $r = 1$, cond.1.(c) $r = 1$, cond.2.(d) $r = 1$, cond.2.

Figure 6.7: The cloth is pinned at two corners and embedded by nine single-radius objects.

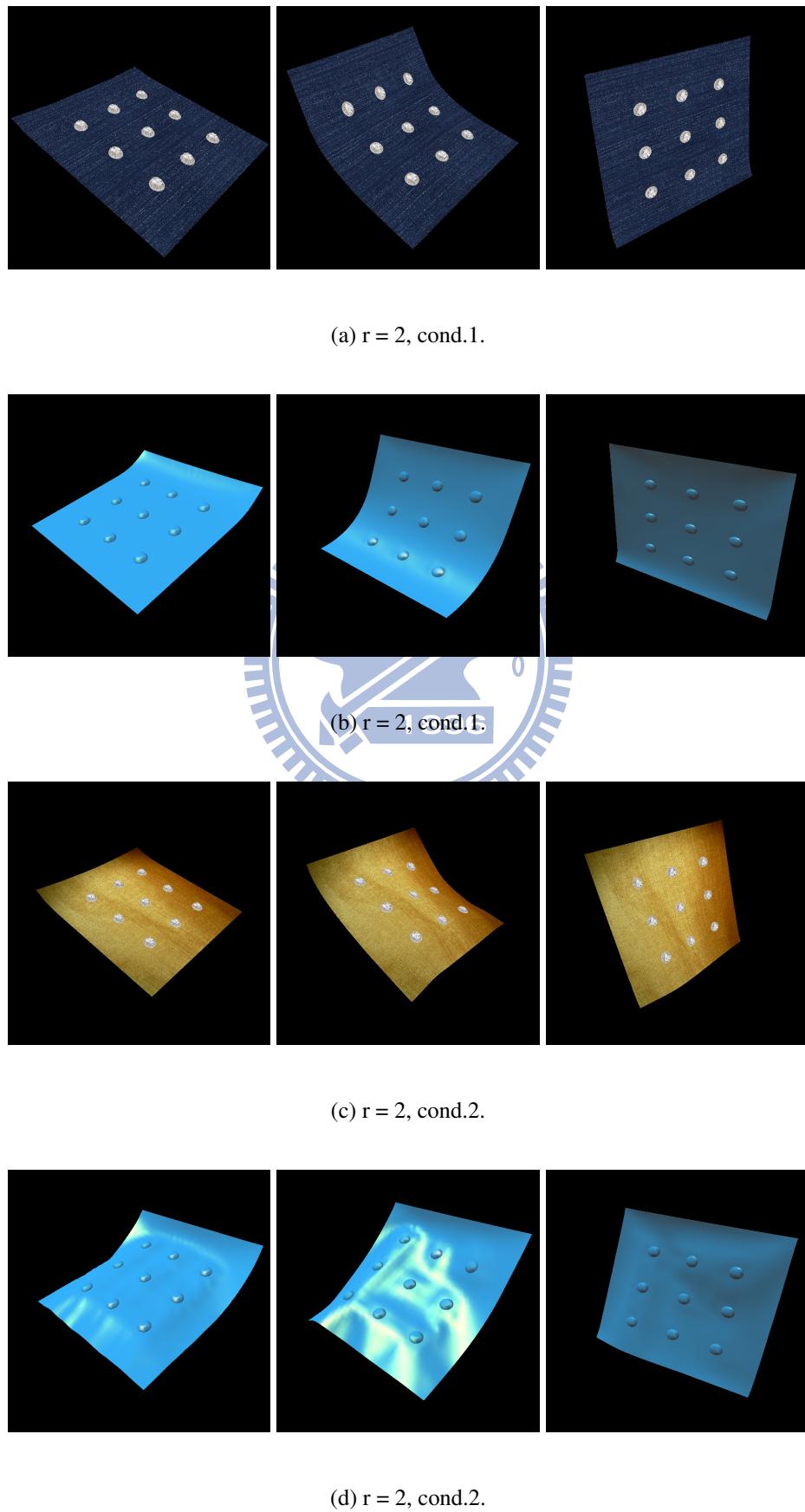


Figure 6.8: The cloth is pinned at two corners and embedded by nine double-radius objects.

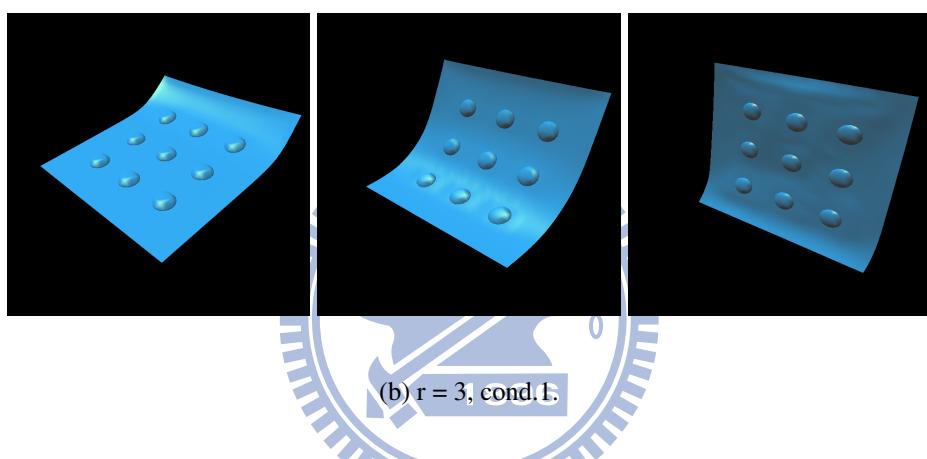
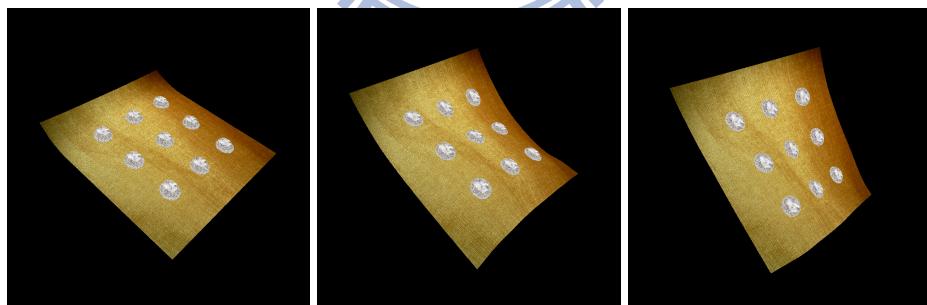
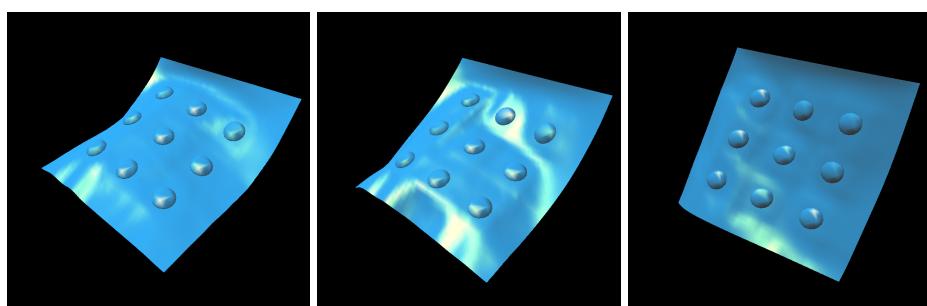
(a) $r = 3$, cond.1.(b) $r = 3$, cond.1.(c) $r = 3$, cond.2.(d) $r = 3$, cond.2.

Figure 6.9: The cloth is pinned at two corners and embedded by nine triple-radius objects.

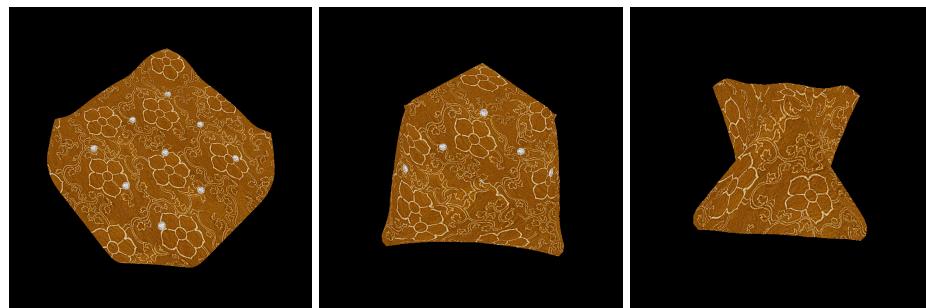
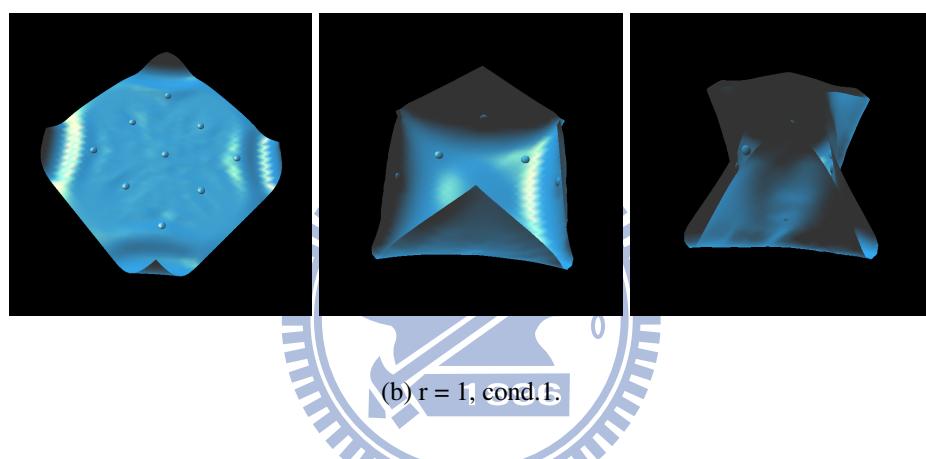
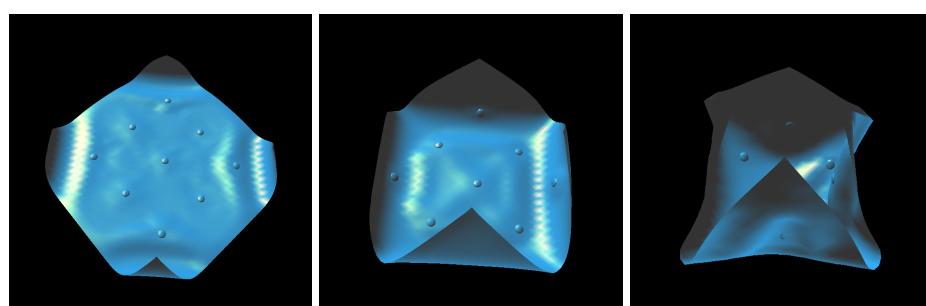
(a) $r = 1$, cond.1.(b) $r = 1$, cond.1.(c) $r = 1$, cond.2.(d) $r = 1$, cond.2.

Figure 6.10: The cloth allowed to relax under gravity is embedded by nine single-radius objects and its four corners are catched.

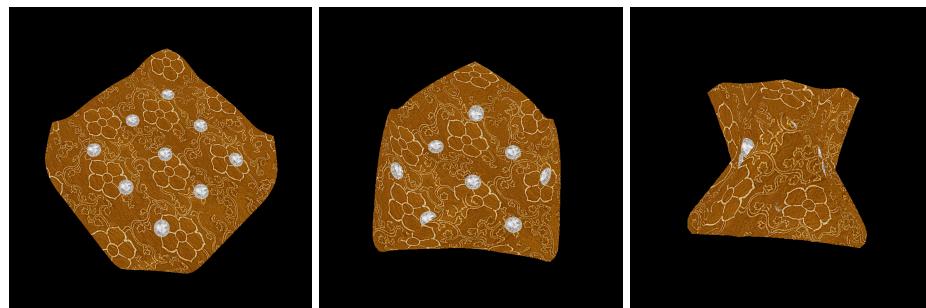
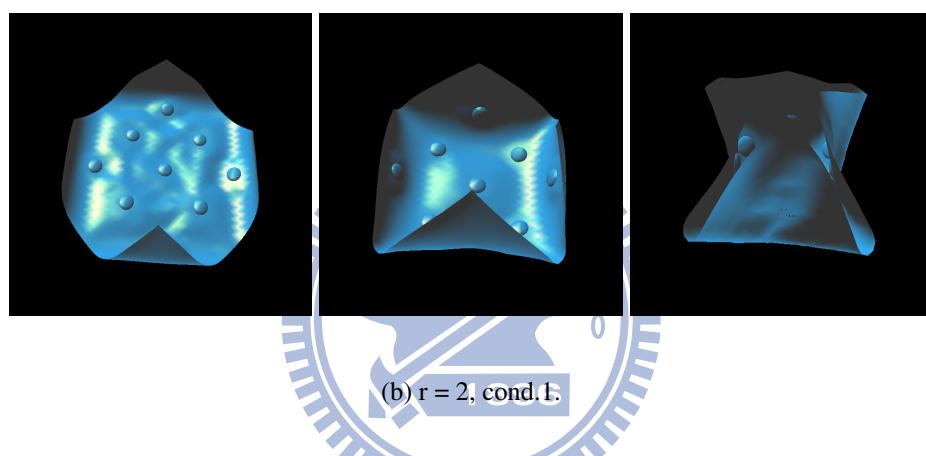
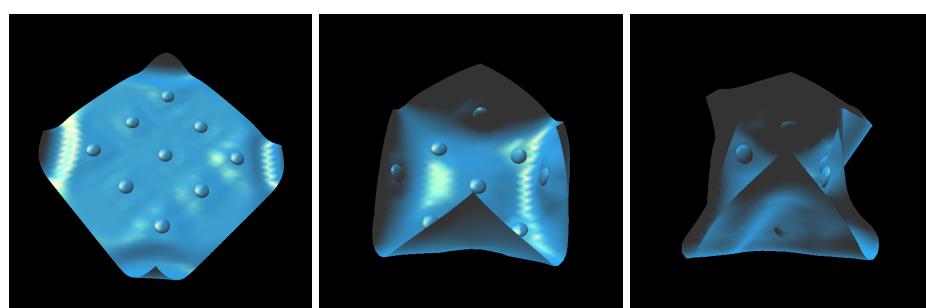
(a) $r = 2$, cond.1.(b) $r = 2$, cond.1.(c) $r = 2$, cond.2.(d) $r = 2$, cond.2.

Figure 6.11: The cloth allowed to relax under gravity is embedded by nine double-radius objects and its four corners are catched.

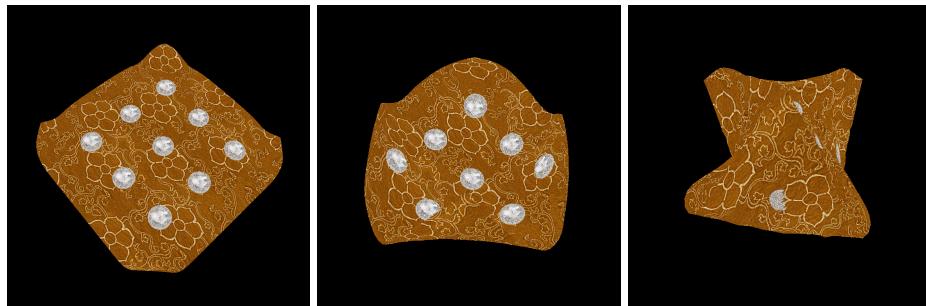
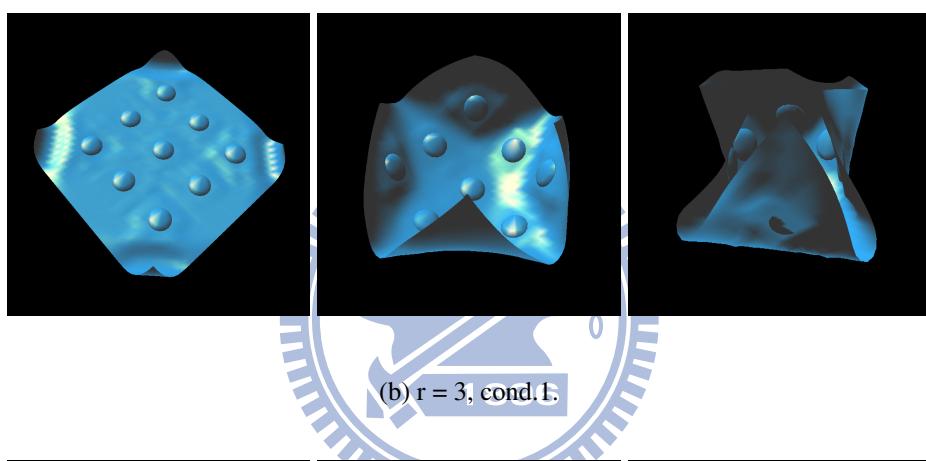
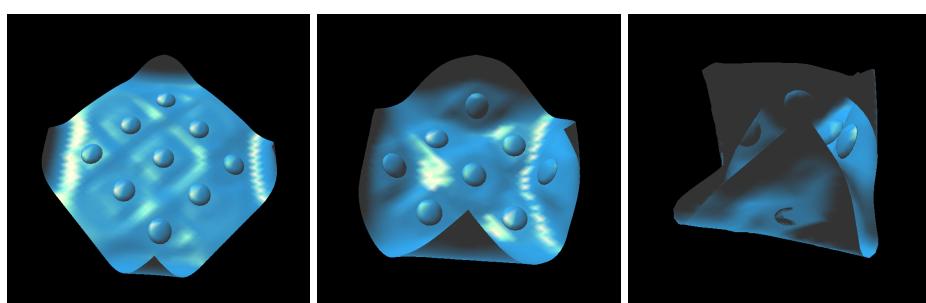
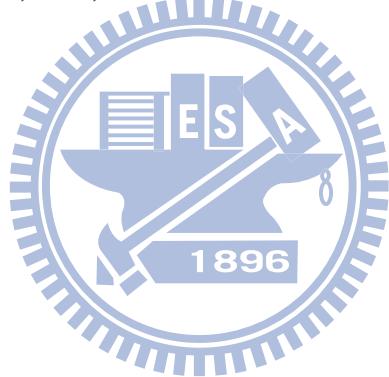
(a) $r = 3$, cond.1.(c) $r = 3$, cond.2.(d) $r = 3$, cond.2.

Figure 6.12: The cloth allowed to relax under gravity is embedded by nine triple-radius objects and its four corners are catched.

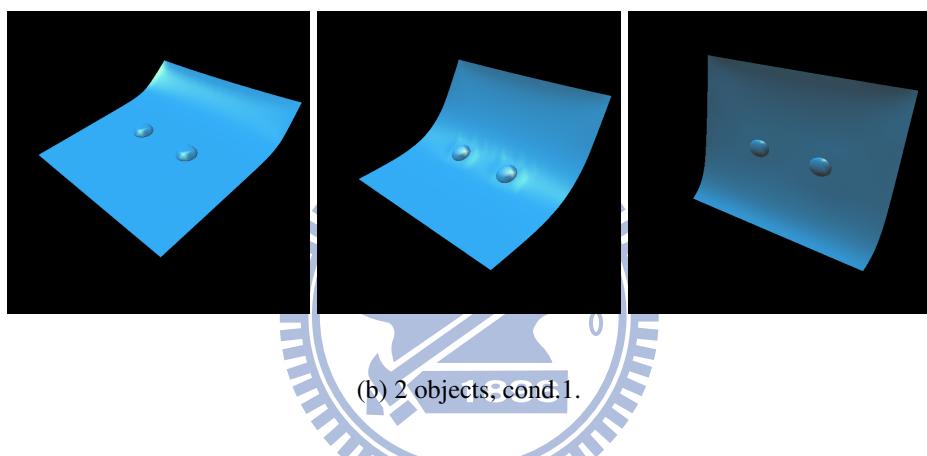
6.2.4 Number of objects

Figure 6.13, 6.14, 6.15, 6.16, 6.17, 6.18 and 6.19 show the results of simulation when we embed different numbers of objects onto the cloth and the cloth relaxes under gravity and is pinned its two corners. Figure 6.20, 6.21, 6.22, 6.23, 6.24, 6.25 and 6.26 show the results of simulation when we embed different numbers of objects onto the cloth and the cloth relaxes under gravity and its four corners are catched. We choose 0.1% as the strain limit and triple-radius as the sizes of the embdded objects. Similar to the observation of above part, since the soft part among two neighboring clusters become smaller when we increase the number of embedded objects, the intermediate look stiffer with the increase of object number. The appearance is noticeable in figure 6.16, 6.17, 6.18, 6.19, 6.23, 6.24, 6.25 and 6.26.

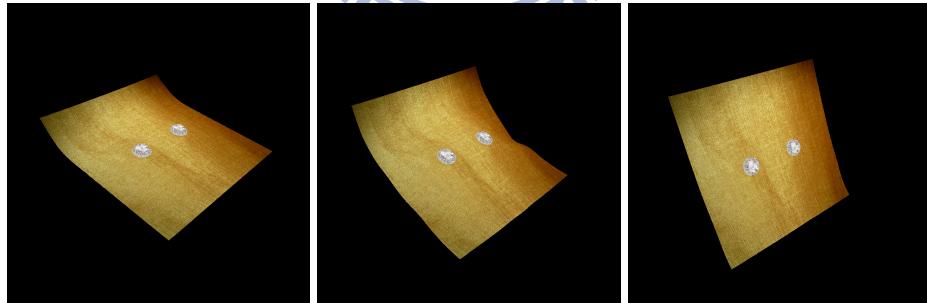




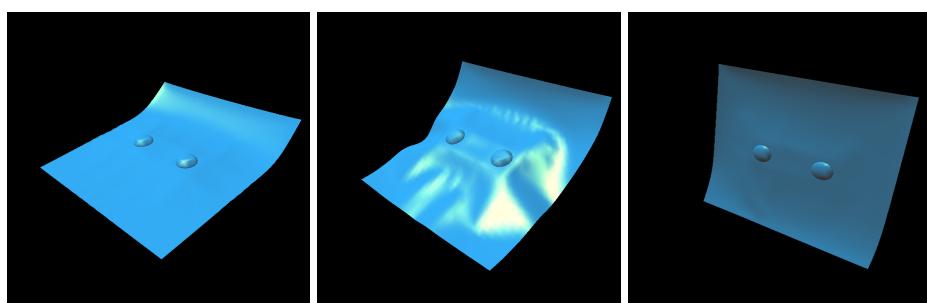
(a) 2 objects, cond.1.



(b) 2 objects, cond.1.

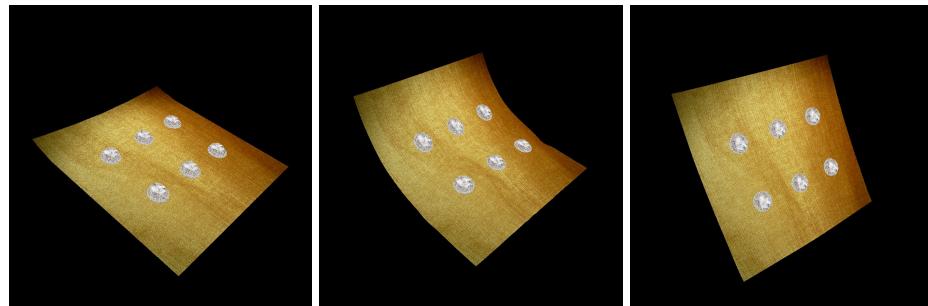


(c) 2 objects, cond.2.

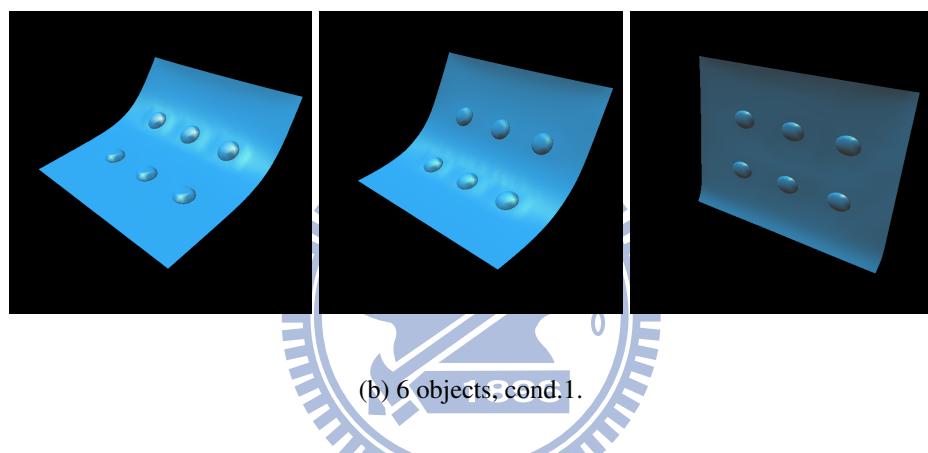


(d) 2 objects, cond.2.

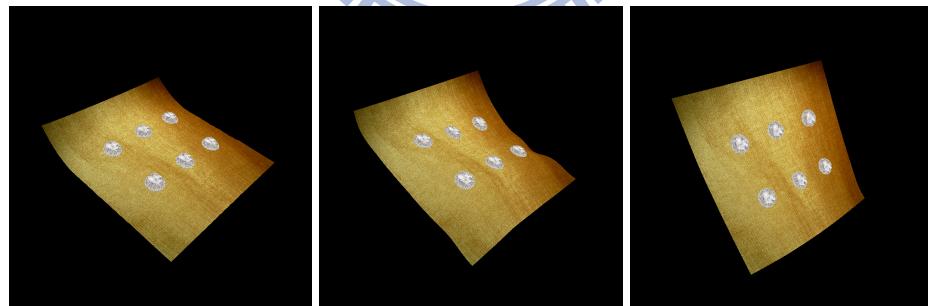
Figure 6.13: 2 rigid objects are embedded on the cloth which is pinned at two corners.



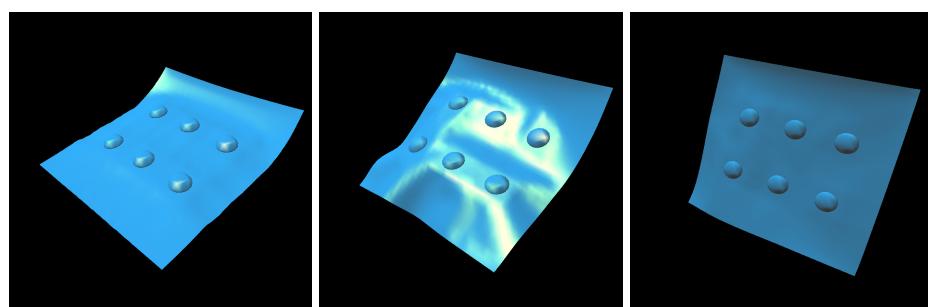
(a) 6 objects, cond.1.



(b) 6 objects, cond.1.



(c) 6 objects, cond.2.

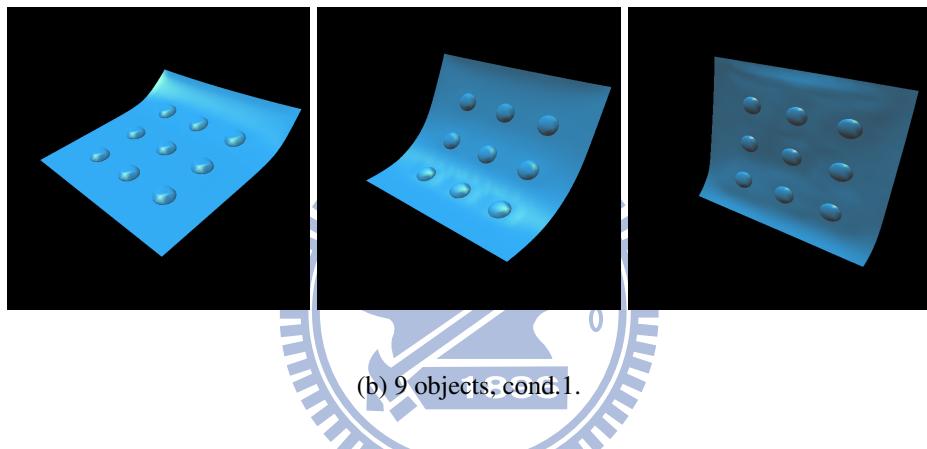


(d) 6 objects, cond.2.

Figure 6.14: 6 rigid objects are embedded on the cloth which is pinned at two corners.



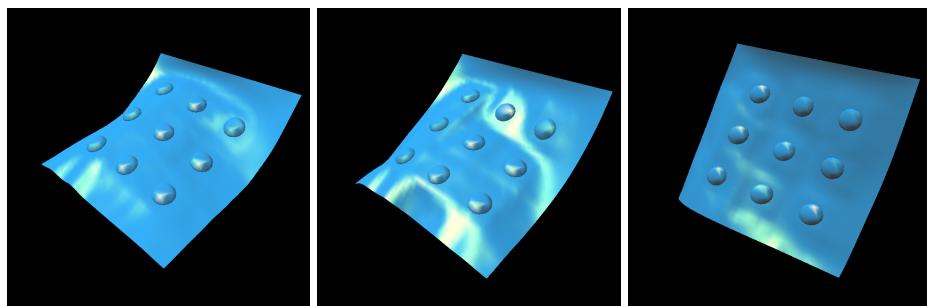
(a) 9 objects, cond.1.



(b) 9 objects, cond.1.

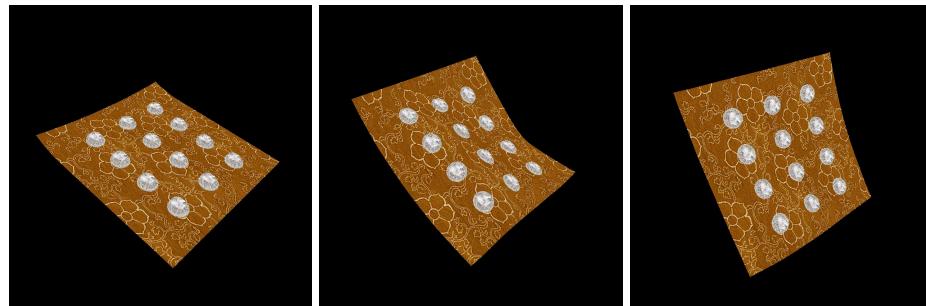


(c) 9 objects, cond.2.

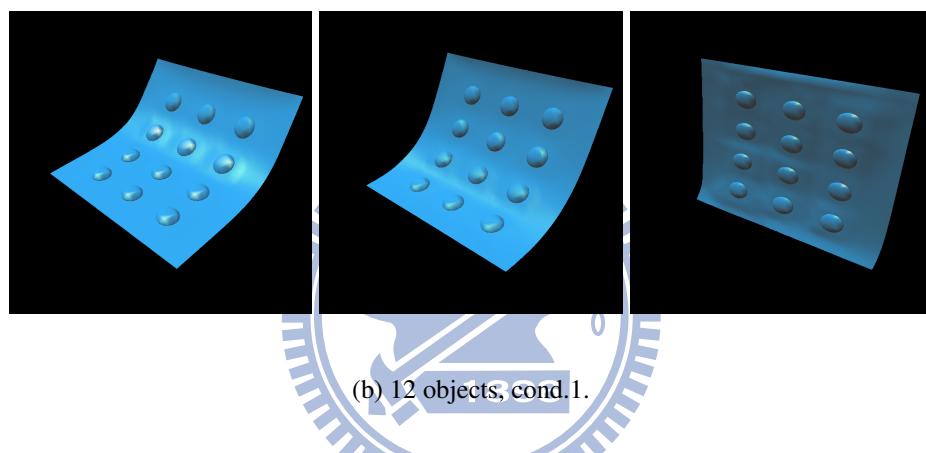


(d) 9 objects, cond.2.

Figure 6.15: 9 rigid objects are embedded on the cloth which is pinned at two corners.



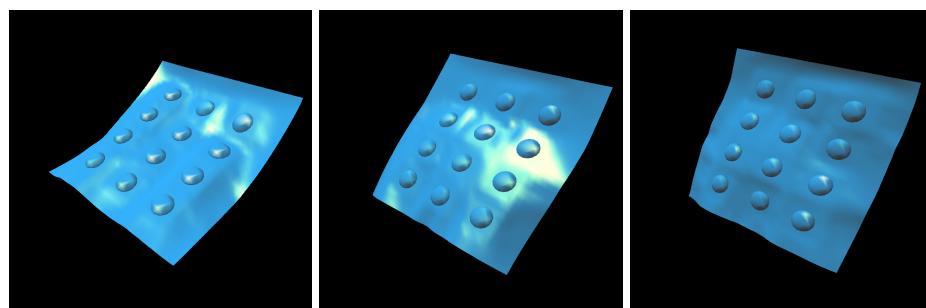
(a) 12 objects, cond.1.



(b) 12 objects, cond.1.

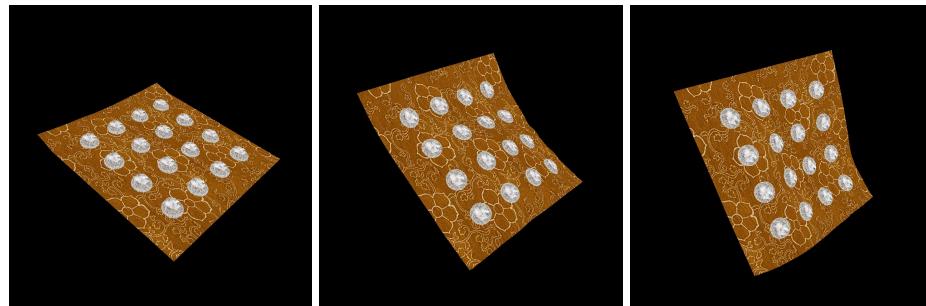


(c) 12 objects, cond.2.

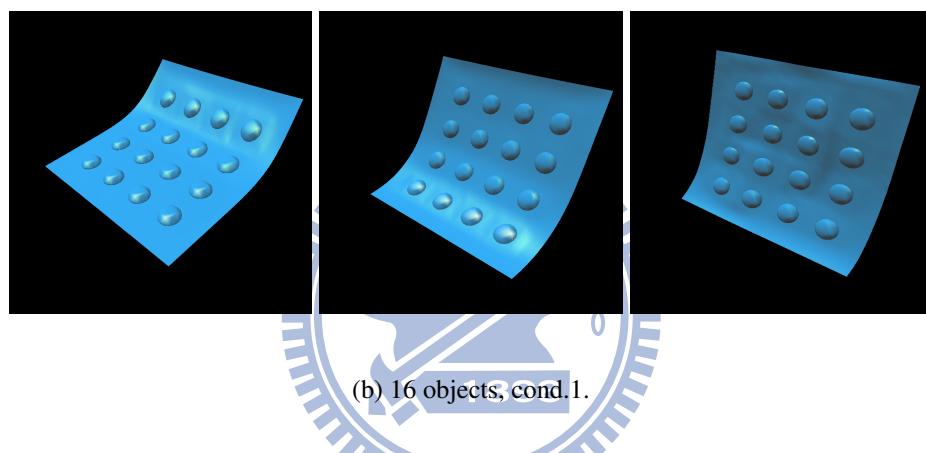


(d) 12 objects, cond.2.

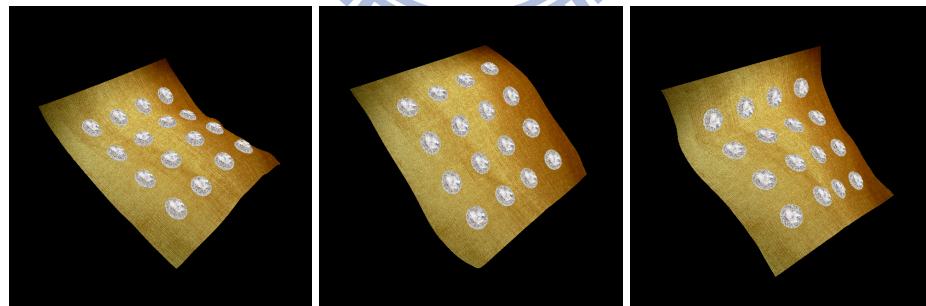
Figure 6.16: 12 rigid objects are embedded on the cloth which is pinned at two corners.



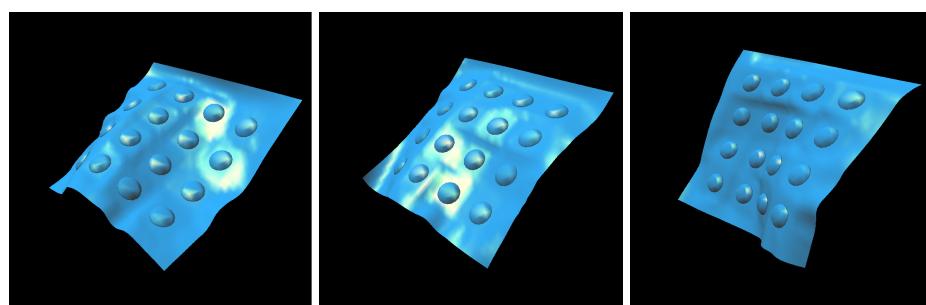
(a) 16 objects, cond.1.



(b) 16 objects, cond.1.

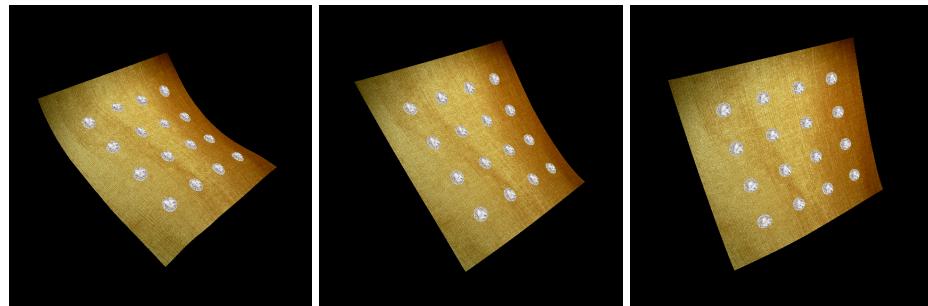


(c) 16 objects, cond.2.

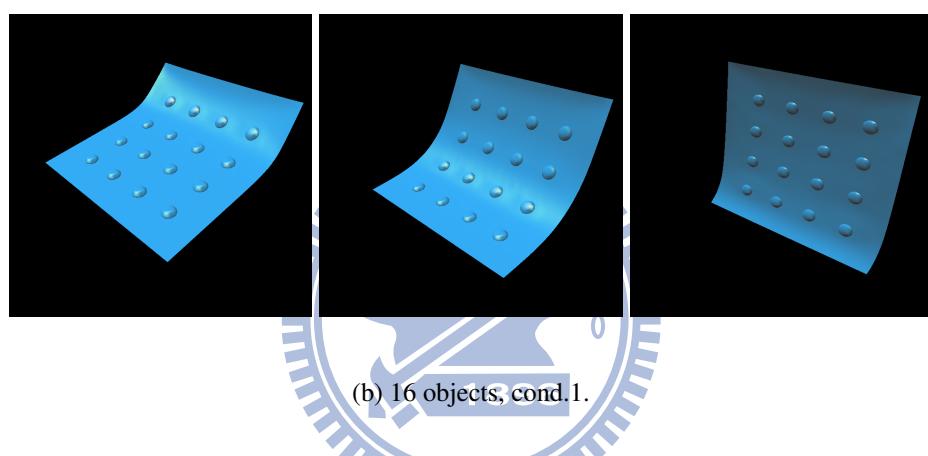


(d) 16 objects, cond.2.

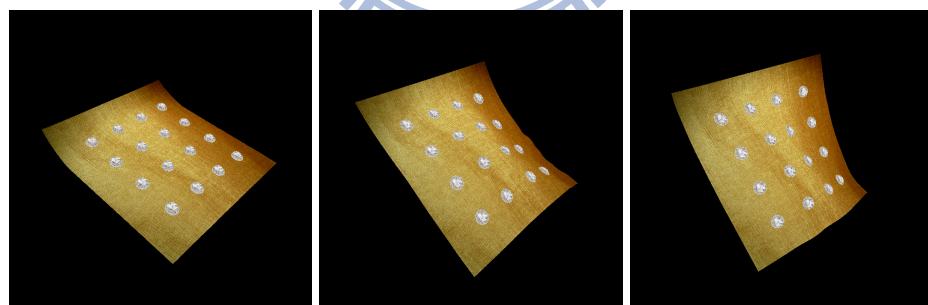
Figure 6.17: 16 triple-radius rigid objects are embedded on the cloth which is pinned at two corners.



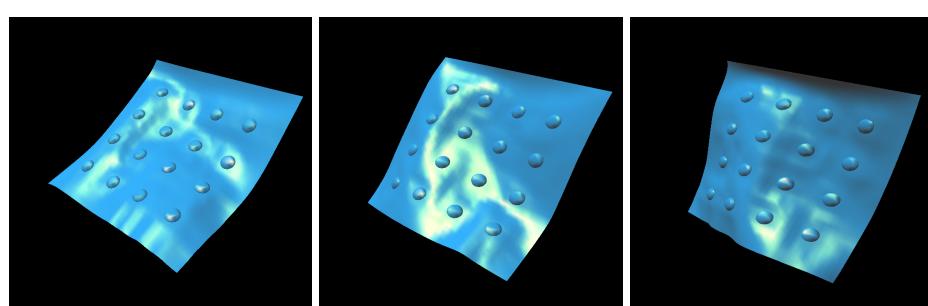
(a) 16 objects, cond.1.



(b) 16 objects, cond.1.

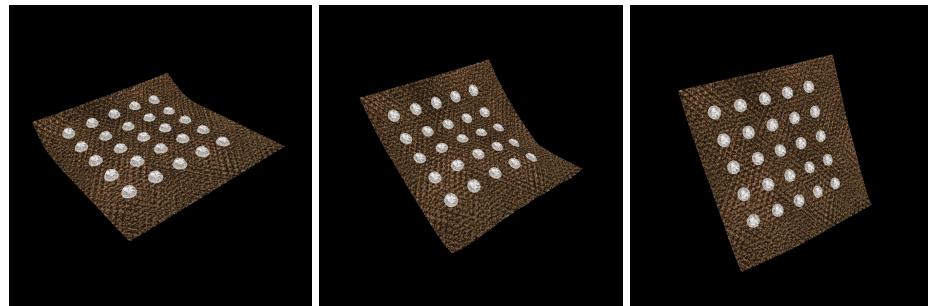


(c) 16 objects, cond.2.

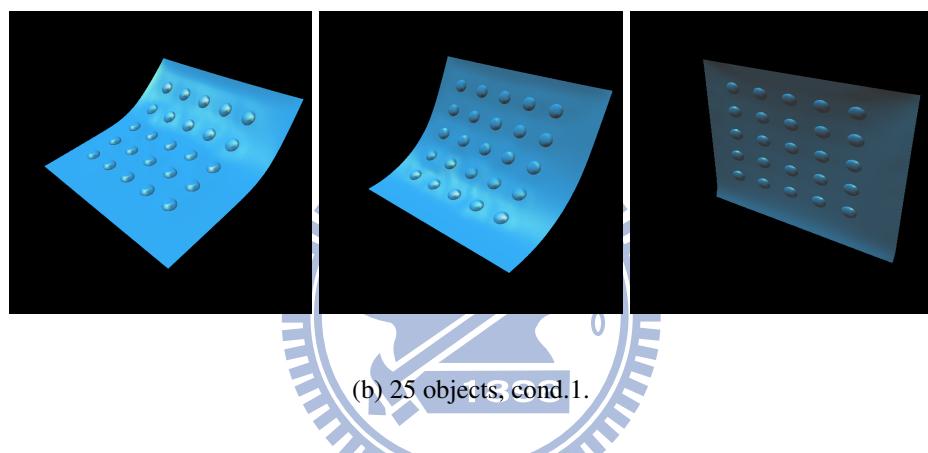


(d) 16 objects, cond.2.

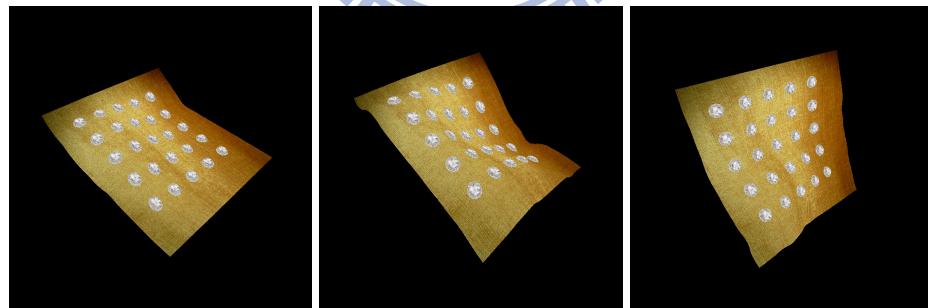
Figure 6.18: 16 double-radius rigid objects are embedded on the cloth which is pinned at two corners.



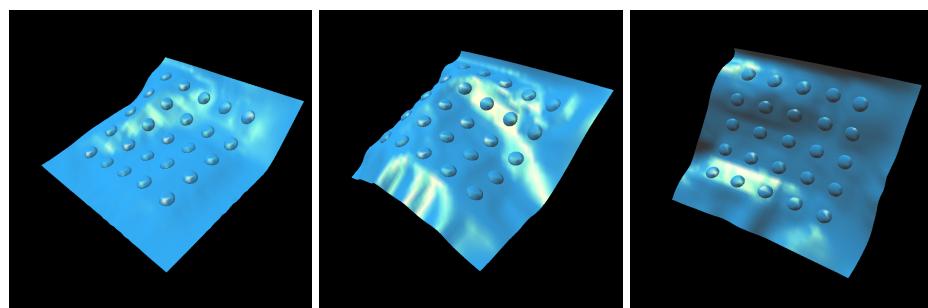
(a) 25 objects, cond.1.



(b) 25 objects, cond.1.

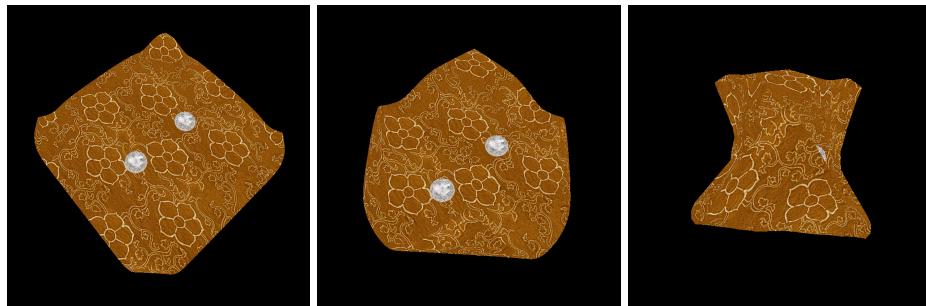


(c) 25 objects, cond.2.

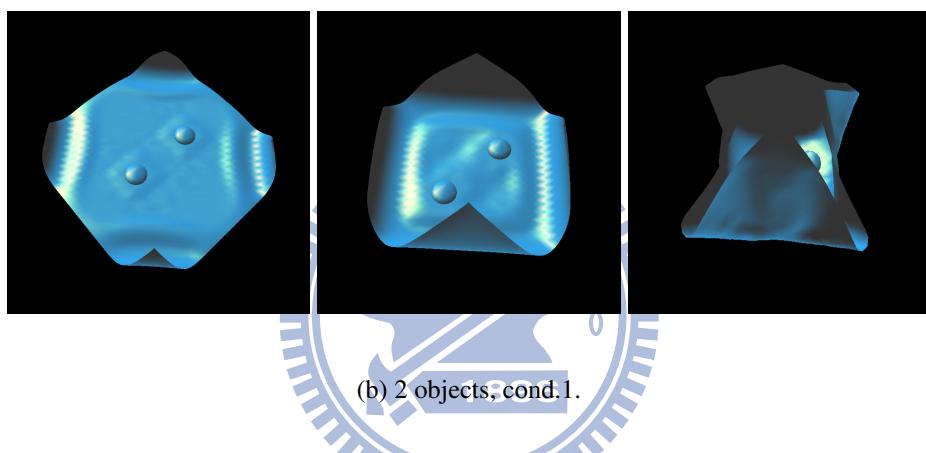


(d) 25 objects, cond.2.

Figure 6.19: 25 rigid objects are embedded on the cloth which is pinned at two corners.



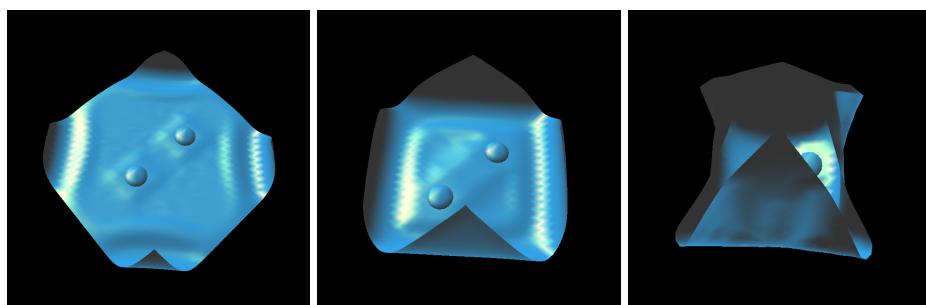
(a) 2 objects, cond.1.



(b) 2 objects, cond.1.

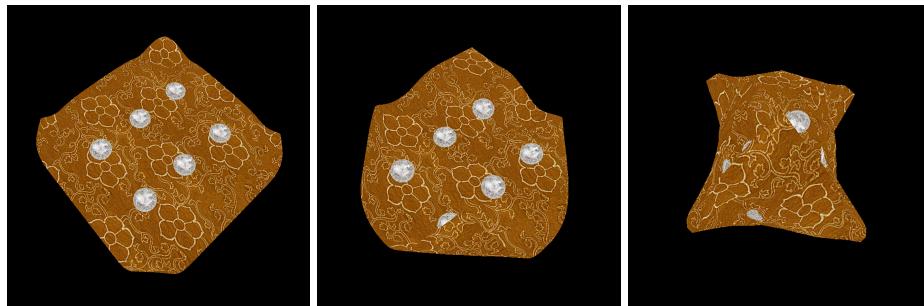


(c) 2 objects, cond.2.

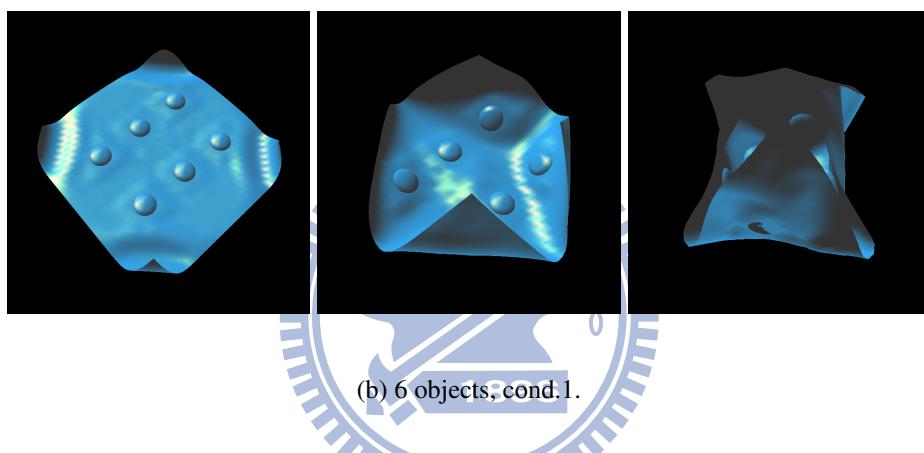


(d) 2 objects, cond.2.

Figure 6.20: The cloth is embedded with 2 objects and its four corners are catched.



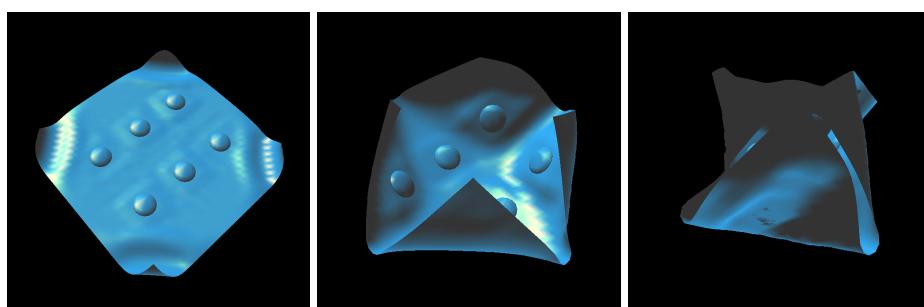
(a) 6 objects, cond.1.



(b) 6 objects, cond.1.

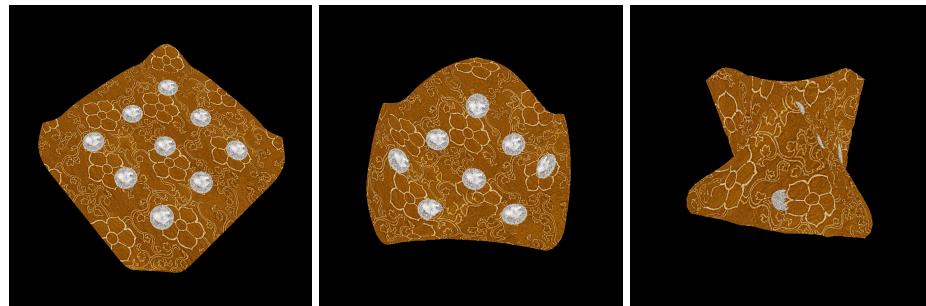


(c) 6 objects, cond.2.

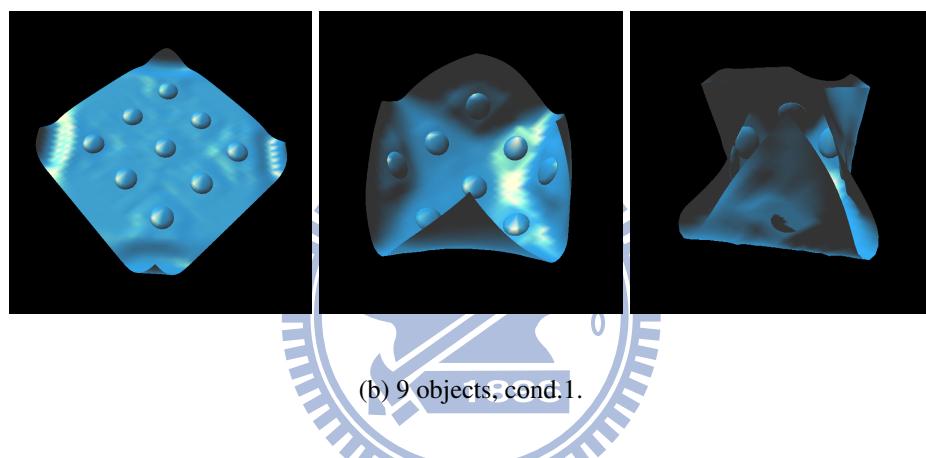


(d) 6 objects, cond.2.

Figure 6.21: The cloth is embedded with 6 objects and its four corners are catched.



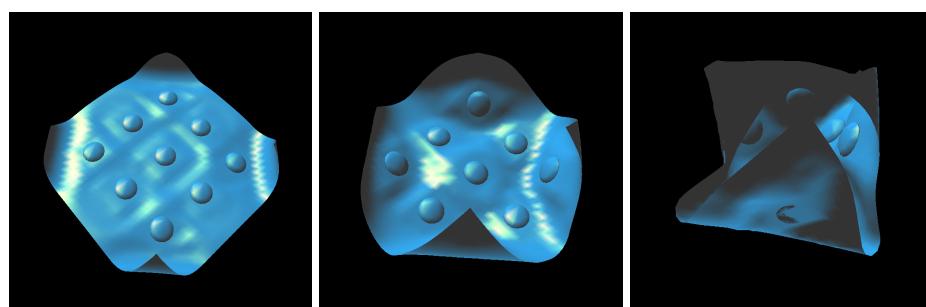
(a) 9 objects, cond.1.



(b) 9 objects, cond.1.

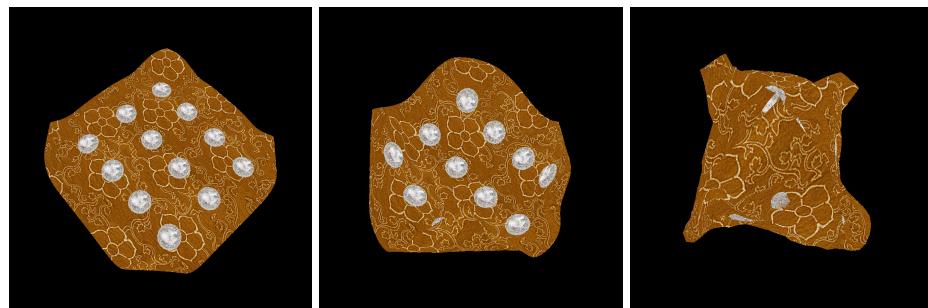


(c) 9 objects, cond.2.

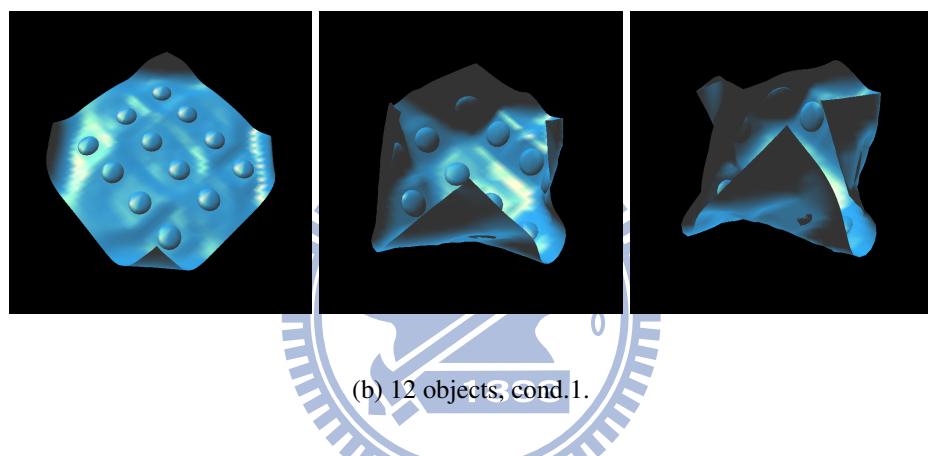


(d) 9 objects, cond.2.

Figure 6.22: The cloth is embedded with 9 objects and its four corners are catched.



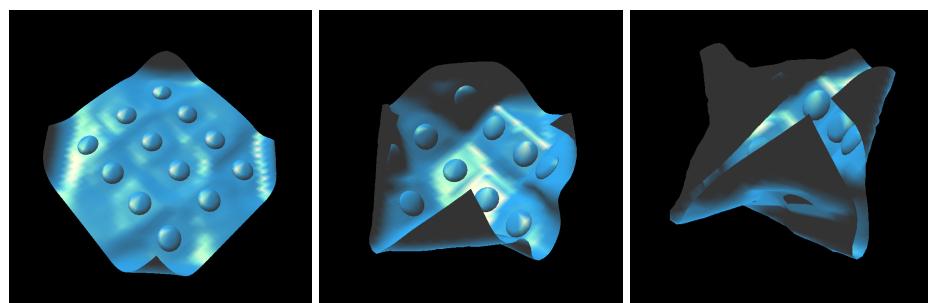
(a) 12 objects, cond.1.



(b) 12 objects, cond.1.

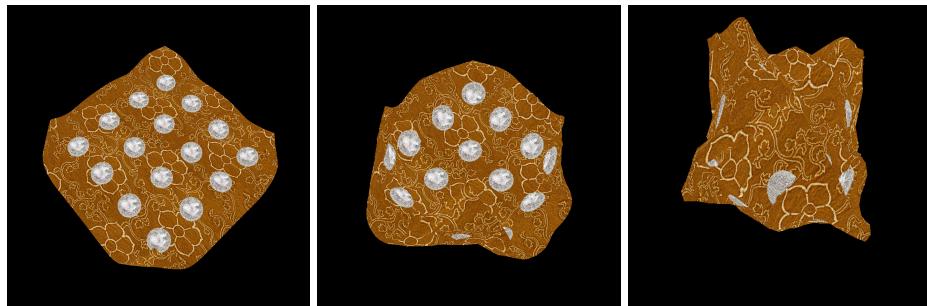


(c) 12 objects, cond.2.

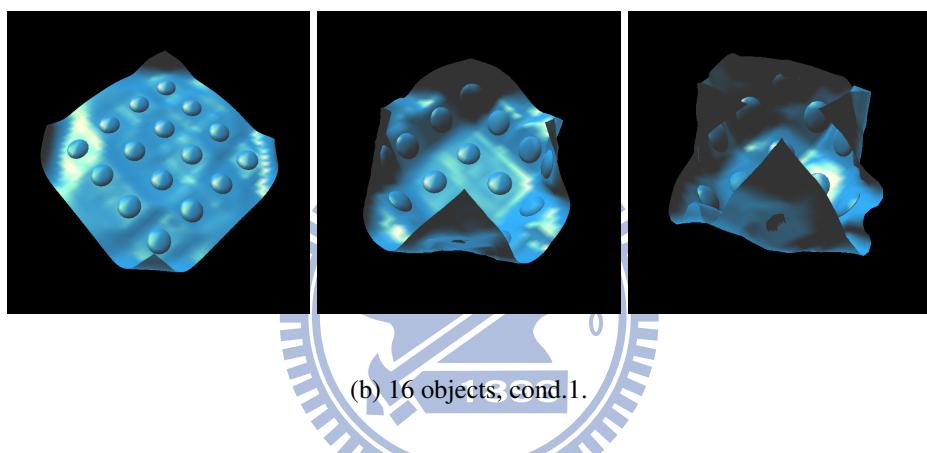


(d) 12 objects, cond.2.

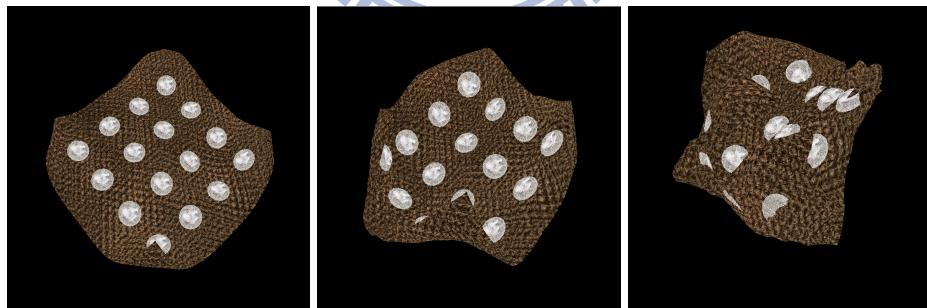
Figure 6.23: The cloth is embedded with 12 objects and its four corners are caught.



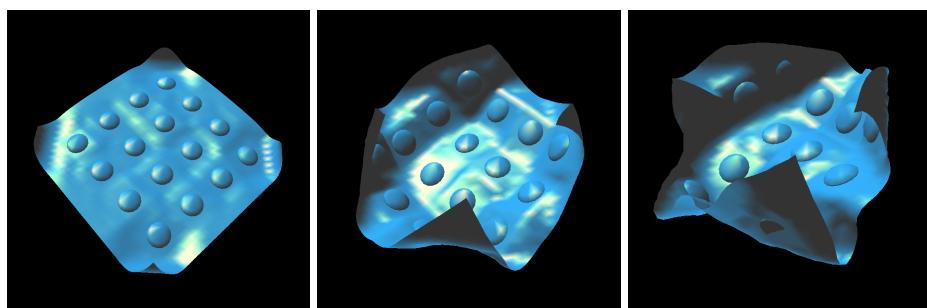
(a) 16 objects, cond.1.



(b) 16 objects, cond.1.



(c) 16 objects, cond.2.

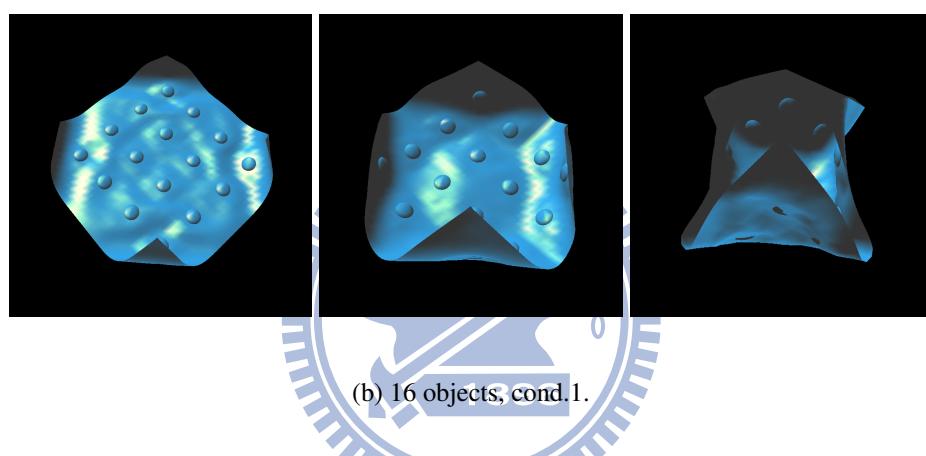


(d) 16 objects, cond.2.

Figure 6.24: The cloth is embedded with 16 triple-radius objects and its four corners are catched.



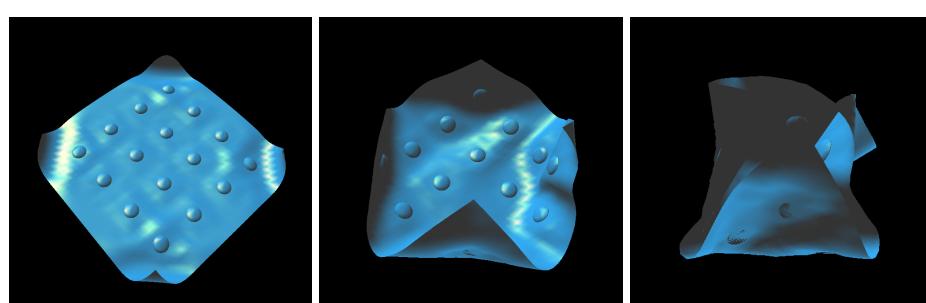
(a) 16 objects, cond.1.



(b) 16 objects, cond.1.

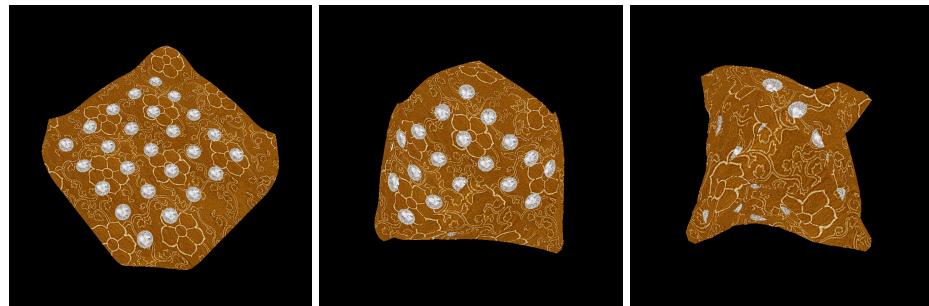


(c) 16 objects, cond.2.

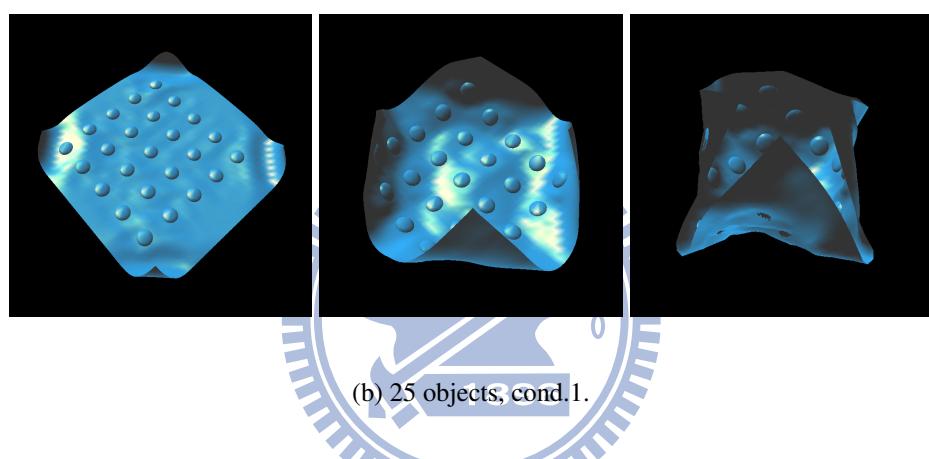


(d) 16 objects, cond.2.

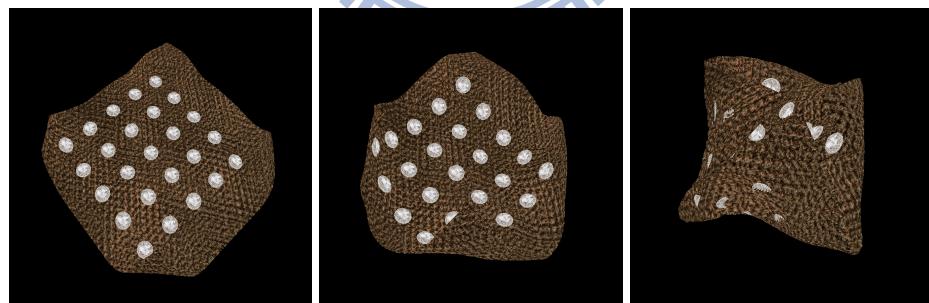
Figure 6.25: The cloth is embedded with 16 double-radius objects and its four corners are caught.



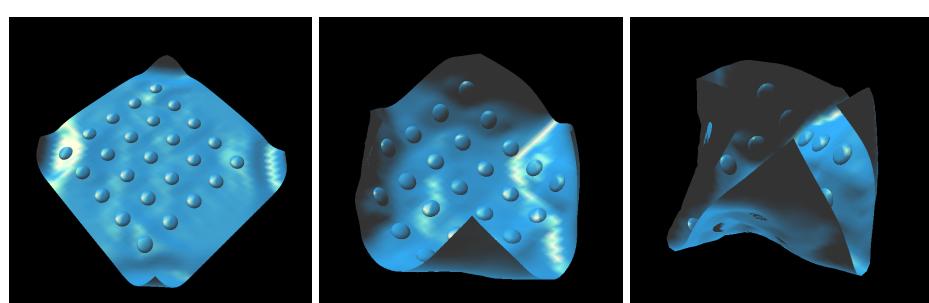
(a) 25 objects, cond.1.



(b) 25 objects, cond.1.



(c) 25 objects, cond.2.



(d) 25 objects, cond.2.

Figure 6.26: The cloth is embedded with 25 double-radius objects and its four corners are caught.

Table 6.3 lists the average error for each vertex to achieve the invariable shape in EXP1 and EXP2. We derive smaller errors if we embed more objects onto the cloth. We had the smallest error when 16 objects were embedded. If the number of objects is more than 25, our system will lead to a non-convergence. Similar to the above part, when we choose cond.2, we derive the less error. But the cloth is stabler if we adopt cond.1. Similarly, in EXP2, the cloth move toward the top right direction if we embed too many large objects onto the cloth.

Obj. Num.	Exp	2	6	9	12	16	25	
Cond.1	EXP1	5.12	3.79	2.71	1.91	2.05		-
	EXP2	2.32	5.47	6.23	6.57	$r = 3$	$r = 2$	$r = 3$
						7.08	7.15	-
	EXP1	3.07	1.59	1.27	1.31	1.94	1.60	-
Cond.2						$r = 3$	$r = 2$	1.78
EXP2	1.89	4.54	3.91	6.16	$r = 3$	$r = 2$	$r = 3$	
					5.92	6.31	-	
							4.94	

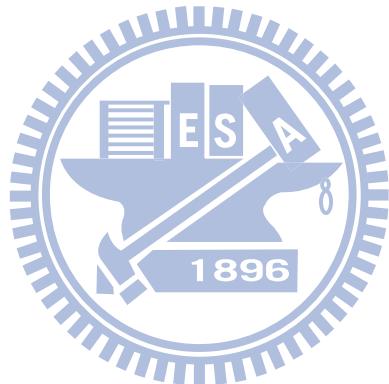
Table 6.3: Average error for each vertex sufficient to maintain the original shape when we choose different numbers of objects.

6.2.5 Timing information

Table 6.4 and 6.5 lists the computation times of the simulating results if we use different strains, object numbers and sizes as parameters in our approach. We use the notation “Total” to represent the average computation time for each frame. The notation “HFP” in table 6.4 and 6.5 represents the computation time spent in our hybrid fast projection stage. All the results were performed on Intel(R) 2.66 GHz quadcore CPUs with 4GB main memory, using a single thread implementation. The computation were all carried out in double precision floating point for solving the linear equations. We use PARDISO [SBR08][SWH07] as our solver. The time step is 1 *ms* in all experiments.

For different strain limits, we observe that the computation time significantly increases when the chosen maximum strain is down to 0.1%. The hybrid fast projection part of our approach

spends a lot of time to find a solution for the inextensibility enforcement and original shape maintenance. For different size of embedded objects, since we derive smaller average error to the original shape for each cluster when we choose bigger object size, the results of the shape matching part in the hybrid fast projection are seldom ignored. Therefore we can see that the reduction of the object size brings us better performance . For different number of embedded objects, the computation time is not proportional or inversely proportional to the number of embedded object. But if the objects are uniformly placed (9, 16 and 25 objects), more embedded objects results less computation time.



Obj. Num	2								
Radius	1			2			3		
Corr. Size	4			25			49		
Strain(%)	10	1	0.1	10	1	0.1	10	1	0.1
Total	151.65	386.95	1507.03	96.18	390.99	1526.89	125.76	349.30	1455.12
HFP	149.61	385.03	1505.13	94.23	388.93	1525.18	123.72	347.40	1453.39
Obj. Num	6								
Radius	1			2			3		
Corr. Size	4			25			49		
Strain(%)	10	1	0.1	10	1	0.1	10	1	0.1
Total	234.11	374.10	1634.92	113.37	340.84	1476.23	128.93	317.36	1524.59
HFP	232.10	372.07	1633.17	111.5	338.89	1477.47	127.18	315.57	1522.78
Obj. Num	9								
Radius	1			2			3		
Corr. Size	4			25			49		
Strain(%)	10	1	0.1	10	1	0.1	10	1	0.1
Total	124.60	450.90	1654.56	107.90	370.36	1477.16	58.92	352.09	1558.03
HFP	122.63	448.93	1652.57	105.99	368.42	1475.26	57.13	350.34	1556.27

Table 6.4: The computation time of the results using different experiment setting (1).

Obj. Num	12								
Radius	1			2			3		
Corr. Size	4			25			49		
Strain(%)	10	1	0.1	10	1	0.1	10	1	0.1
Total	139.71	411.15	1647.58	88.15	363.61	1708.22	90.82	259.04	1696.38
HFP	137.70	409.17	1645.56	86.12	361.74	1706.28	89.09	257.17	1694.58
Obj. Num	16								
Obj. Radius	1			2			3		
Corr. Cluster Size	4			25			49		
Strain Limit (%)	10	1	0.1	10	1	0.1	10	1	0.1
Total	149.05	355.75	2144.61	110.32	320.54	2080.86	51.69	225.29	1553.27
HFP	146.93	353.73	2142.65	108.39	318.57	2079.03	49.92	223.39	1551.48
Obj. Num	25								
Obj. Radius	1			2			3		
Corr. Cluster Size	4			25			49		
Strain Limit (%)	10	1	0.1	10	1	0.1	10	1	0.1
Total	133.85	403.73	2276.00	143.76	285.26	19587.48	99.03	-	-
HFP	131.69	401.42	2273.93	141.59	283.24	19585.54	97.04	-	-

Table 6.5: The computation time of the results using different experiment setting (2).

6.3 Limitations

For the permission strain limit is close to 10%, there exists unexpected bending waves on the upper half part of the cloth. 1% is a good choice for our system. Moreover, when the strain down to 0.1%, the performance becomes unacceptable. Our method does not have a good convergence if we embed more than 20 objects, and each object has radius more than triple.

CHAPTER 7

Conclusions



7.1 Summary

We propose a novel hybrid approach for simulating inextensible cloth embedded with rigid bodies. We compute the fitting regions of the cloth for every embedded object. Experimental results show that the simulation of the cloth embedded with the rigid objects is realistic. Except for the fast projection step, the progressive shape matching scheme also helps us to maintain the stability of the system. The progressive shape matching scheme has good convergence.

7.2 Future Work

In the future, we expect to implement a proper collision handler integrated with our system for handling collisions. The fast projection results in not realistic cloth simulation when we apply more than 10% as the strain limit (see figure 7.1). The problem lies in that the method is not in accordance with the dynamics. A better method to simulate the cloth using the fast projection and keep the stability is an important issue.

Since we do not adopt every computation results from the shape matching part, the shape of each cluster is just approximated but not be fixed. At any frame, if the results of the shape matching part is discarded, then the shape cannot be guaranteed to transform to its original look. Thus the efficient uses of the shape matching scheme is feasible for the future work. The convergence of our hybrid fast projection cost expensively if we embed too many large objects on the cloth. To speedup the performance in this part is also an important issue to be done. In our approach, we still adjust the positions of vertices directly but not take the conservation of momentum into account. This may cause instability. Thus the impulse is a better choice to adjust the information of the primitives.

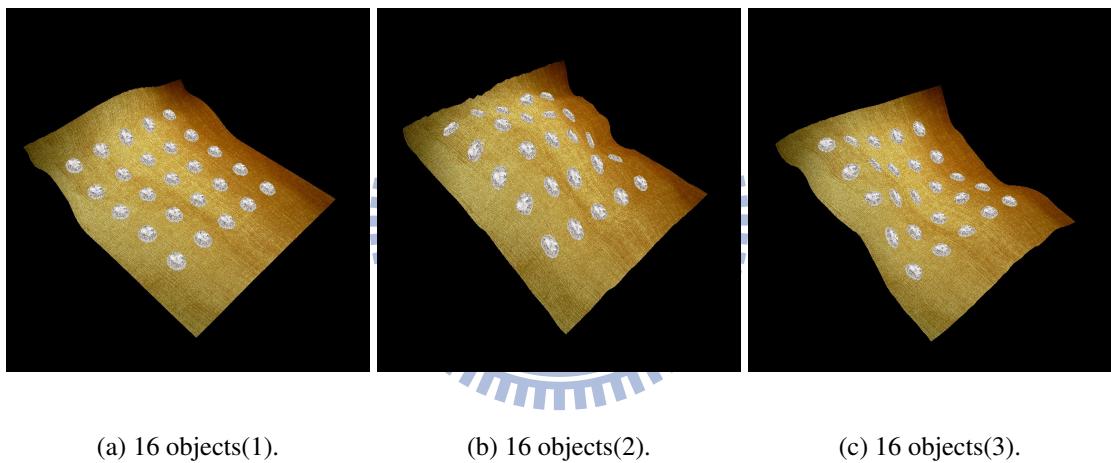


Figure 7.1: The unstable state of our simulation if the strain limit equals to 10%.

Bibliography

[Ber99] G. Van Den Bergen. Efficient collision detection of complex deformable models using aabb trees. *Graphics tools: The jgt editors' choice*, page 131, 1999.

[BFA02] R. Bridson, R. Fedkiw, and J. Anderson. Robust treatment of collisions, contact and friction for cloth animation. In *SIGGRAPH 2002 Proceedings*, pages 594–603. ACM, 2002.

[BHW94] D.E Breen, D.H. House, and M.J. Wozny. Predicting the drape of woven cloth using interacting particles. In *SIGGRAPH 1994 Proceedings*, pages 365–372. ACM, 1994.

[BW98] D. Baraff and A. Witkin. Large steps in cloth simulation. In *SIGGRAPH 1998 Proceedings*, pages 43–54, 1998.

[BW03] D. Baraff and A. Witkin. Untangling cloth. *ACM Transactions on Graphics*, 22(3):862–870, 2003.

[CK05] K.J. Choi and H.S. Ko. Stable but responsive cloth. In *ACM SIGGRAPH 2005 Courses*, page 1. ACM, 2005.

[CTM08] S. Curtis, R. Tamstorf, and D. Manocha. Fast collision detection for deformable models using representative-triangles. In *Proceedings of the 2008 symposium on Interactive 3D graphics and games*, pages 61–69. ACM, 2008.

[CYMTT92] M. Carignan, Y. Yang, N. Magnenat-Thalmann, and D. Thalmann. Dressing animated synthetic actors with complex deformable clothes. *ACM Siggraph Computer Graphics*, 26(2):99–104, 1992.

[EDC96] J.W. Eischen, S. Deng, and T.G Clapp. Finite-element modeling and control of flexible fabric parts. *IEEE Computer Graphics and Applications*, 16(5):71–80, 1996.

[FGLM01] M. Foskey, M. Garber, M.C. Lin, and D. Manocha. A voronoi-based hybrid motion planner for rigid bodies. In *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*. Citeseer, 2001.

[GHF⁺07] R. Goldenthal, D. Harmon, R. Fattal, M. Bercovier, and E. Grinspun. Efficient simulation of inextensible cloth. *ACM Transactions on Graphics*, 26(3):49, 2007.

[GLM96] S. Gottschalk, M.C. Lin, and D. Manocha. Obbtree: a hierarchical structure for rapid interference detection. In *SIGGRAPH 1996 Proceedings*, page 180. ACM, 1996.

[HCJ⁺05] M. Hong, M.H. Choi, S. Jung, S. Welch, and J. Trapp. Effective constrained dynamic simulation using implicit constraint enforcement. In *IEEE International Conference on Robotics and Automation*, volume 4, page 4520, 2005.

[HF07] M. Hutter and A. Fuhrmann. Optimized continuous collision detection for deformable triangle meshes. In *Proceedings of the 2007 WSCG*, pages 25–32, 2007.

[Hub93] P. Hubbard. Interactive collision detection. In *Virtual Reality*, pages 24–31. Proceedings of IEEE Symposium on Research Frontiers, 1993.

[HVTG08] D. Harmon, E. Vouga, R. Tamstorf, and E. Grinspun. Robust treatment of simultaneous collisions. In *ACM SIGGRAPH 2008 papers*, page 23. ACM, 2008.

[KHM⁺98] J.T. Klosowski, M. Held, J.S.B Mitchell, H. Sowizral, and K. Zikan. Efficient collision detection using bounding volume hierarchies of k-dops. *IEEE transactions on Visualization and Computer Graphics*, 4(1):21–36, 1998.

[KHY09] D. Kim, J. Heo, and S. Yoon. Hpccd: Hybrid parallel continuous collision detection. *Proceedings of the Korea Advanced Institute of Science and Technology. South Korea*, pages 235–240, 2009.

[LAM03] T. Larsson and T. Akenine-Möller. Efficient collision detection for models deformed by morphing. *The Visual Computer*, 19(2):164–174, 2003.

[LAM06] T. Larsson and T. Akenine-Möller. A dynamic bounding volume hierarchy for generalized collision detection. *Computers & Graphics*, 30(3):450–459, 2006.

[Lar] *Lagrange multiplier*. Available from: <http://engineering.purdue.edu/~mark/puthesis>.

[LKC96] J.D. Liu, M.T. Ko, and R.C. Chang. Collision avoidance in cloth animation. *The Visual Computer*, 12(5):234–243, 1996.

[MHTG05] M. Müller, B. Heidelberger, M. Teschner, and M. Gross. Meshless deformations based on shape matching. *ACM Trans. Graph.*, 24, 2005.

[MKE03] J. Mezger, S. Kimmerle, and O. Etzmuß. Hierarchical techniques in collision detection for cloth animation. *Journal of WSCG*, 11(2):322–329, 2003.

[MW88] M. Moore and J. Wilhelms. Collision detection and response for computer animation. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pages 289–298. ACM, 1988.

[OKN09] M. Ohta, Y. Kanamori, and T. Nishita. Deformation and fracturing using adaptive shape matching with stiffness adjustment. *Comput. Animat. Virtual Worlds*, 20:365–373, 2009.

[Pro95] X. Provot. Deformation constraints in a mass-spring model to describe rigid cloth behaviour. In *Graphics interface*, pages 147–147, 1995.

[Pro97] X. Provot. Collision and self-collision handling in cloth model dedicated to design garments. In *Graphics Interface*, volume 97, pages 177–189, 1997.

[RJ07] A.R. Rivers and D.L. James. Fastlsm: fast lattice shape matching for robust real-time deformation. *ACM Transactions on Graphics.*, 26, 2007.

[RKC02] S. Redon, A. Kheddar, and S. Coquillart. Fast continuous collision detection between rigid bodies. In *Computer graphics forum*, volume 21(3), pages 279–288. Citeseer, 2002.

[RLMK05] S. Redon, M.C. Lin, D. Manocha, and Y.J. Kim. Fast continuous collision detection for articulated models. *Journal of Computing and Information Science in Engineering*, 5:126, 2005.

[SBR08] O. Schenk, M. Bollhöfer, and R. A. Römer. On large scale diagonalization techniques for the Anderson model of localization. *SIAM Review*, 50(1):91–112, 2008. SIGEST Paper.

[SD92] K. Shoemake and T. Duff. Matrix animation and polar decomposition. In *Proceedings of the conference on Graphics interface*, volume 92, pages 258–264. Citeseer, 1992.

[SKTK95] A. Smith, Y. Kitamura, H. Takemura, and F. Kishino. A simple and efficient method for accurate collision detection among deformable polyhedral objects in arbitrary motion. In *vrais*, page 136. Published by the IEEE Computer Society, 1995.

[SOG08] D. Steinemann, M. A. Otaduy, and M. Gross. Fast adaptive shape matching deformations. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '08, pages 87–94, 2008.

[SSBT08] T. Stumpf, J. Spillmann, M. Becker, and M. Teschner. A geometric deformation model for stable cloth simulation. *Proc. VRIPHYS*, pages 13–14, 2008.

[SWH07] O. Schenk, A. Wächter, and M. Hagemann. Matching-based preprocessing algorithms to the solution of saddle-point problems in large-scale nonconvex interior-point optimization. *Comput. Optim. Appl.*, 36(2-3):321–341, April 2007.

[TCYM09] M. Tang, S. Curtis, S.E. Yoon, and D. Manocha. Iccd: Interactive continuous collision detection between deformable models using connectivity-based culling. *planning*, 3:4, 2009.

[TF88] D. Terzopoulos and K. Fleischer. Modeling inelastic deformation: viscoelasticity, plasticity, fracture. In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, page 278. ACM, 1988.

[TMT09] M. Tang, D. Manocha, and R. Tong. Multi-core collision detection between deformable models. In *2009 SIAM/ACM Joint Conference on Geometric and Physical Modeling*, pages 355–360. ACM, 2009.

[VMT94] P. Volino and N. Magnenat-Thalmann. Efficient self-collision detection on smoothly discretized surface animations using geometrical shape regularity. In *Computer Graphics Forum*, volume 13(3), pages 155–166. Amsterdam: North Holland, 1982-, 1994.

[VMT95] P. Volino and N. Magnenat-Thalmann. Versatile and efficient techniques for simulating cloth and other deformable objects. In *Proceedings of the 22nd annual*

conference on Computer graphics and interactive techniques, pages 137–144. ACM New York, NY, USA, 1995.

[VMT00] P. Volino and N. Magnenat-Thalmann. Implementing fast cloth simulation with collision response. In *Computer Graphics International*, volume 7(1), pages 9–4, 2000.

[VMTF09] P. Volino, N. Magnenat-Thalmann, and F. Faure. A simple approach to nonlinear tensile stiffness for accurate cloth simulation. *ACM Transactions on Graphics (TOG)*, 28(4):105, 2009.

[VMTJTm95] P. Volino, N. Magnenat-Thalmann, S. Jianhua, and D. Thal-mann. Collision and self-collision detection: Efficient and robust solutions for highly deformable surfaces. In *Computer Animation and Simulation*, volume 95, pages 55–65. Citeseer, 1995.

[WB05] W.S.K. Wong and G. Baciu. Dynamic interaction between deformable surfaces and nonsmooth objects. *IEEE Transactions on Visualization and Computer Graphics*, 11(3):329–340, 2005.

[WB06] W.S.K. Wong and G. Baciu. A randomized marking scheme for continuous collision detection in simulation of deformable surfaces. In *Proceedings of the 2006 ACM international conference on Virtual reality continuum and its applications*, page 188. ACM, 2006.

[WB07] W.S.K Wong and G. Baciu. Robust continuous collision detection for interactive deformable surfaces. *Computer Animation and Virtual Worlds*, 18(3):179–192, 2007.

[Ye08] J. Ye. Simulating inextensible cloth using impulses. In *Computer Graphics Forum*, volume 27(7), pages 1901–1907, 2008.

[YKC00] W.R Yu, T.J. Kang, and K. Chung. Drape simulation of woven fabric by using explicit dynamics analysis. *J Text Inst*, 91(1):285–301, 2000.

[ZM08] L. Zhang and D. Manocha. Constrained motion interpolation with distance constraints. In *International Workshop on the Algorithmic Foundations of Robotics (WAFR)*, 2008.

