

## Dynamic Remeshing for Real-Time Representation of Thin-Shell Tearing Simulations on the GPU

Jong-Hyun Kim\*

\*Associate Professor, College of Software and Convergence (Dept. of Design Technology), Inha University, Incheon, Korea

### [Abstract]

In this paper, we propose a GPU-based method for real-time processing of dynamic re-meshing required for tearing cloth. Thin shell materials are used in various fields such as physics-based simulation/animation, games, and virtual reality. Tearing the fabric requires dynamically updating the geometry and connectivity, making the process complex and computationally intensive. This process needs to be fast, especially when dealing with interactive content. Most methods perform re-meshing through low-resolution simulations to maintain real-time, or rely on an already segmented pattern, which is not considered dynamic re-meshing, and the quality of the torn pattern is low. In this paper, we propose a new GPU-optimized dynamic re-meshing algorithm that enables real-time processing of high-resolution fabric tears. The method proposed in this paper can be used for virtual surgical simulation and physics-based modeling in games and virtual environments that require real-time, as it allows dynamic re-meshing rather than pre-split meshes.

▶ **Key words:** Thin-shell, Cloth simulation, Dynamic remeshing, Tearing simulation, Real-time

### [요 약]

본 논문에서는 GPU기반으로 옷감을 찢는 데 필요한 동적 재메싱을 실시간으로 처리할 수 있는 방법을 제안한다. 얇은 셸 재질은 물리 기반 시뮬레이션/애니메이션, 게임, 가상현실 등 다양한 분야에서 활용되고 있다. 옷감을 찢는 것은 기하학과 연결 구조를 동적으로 갱신해야 되기 때문에 그 처리 과정이 복잡하고 계산량이 크다. 특히 인터랙티브 콘텐츠를 다루는 분야에서는 이 과정이 빠르게 수행되어야 한다. 대부분의 방법에서는 실시간을 유지하기 위해 저해상도 시뮬레이션을 통해 재메싱을 수행하거나 이미 분할된 패턴을 그대로 이용하기 때문에 동적 재메싱이라고 보기 어려우며, 찢어진 패턴의 품질이 낮다. 본 논문에서는 GPU에 최적화된 동적 재메싱 알고리즘을 새롭게 제안함으로써 고해상도 옷감 찢어짐을 실시간으로 처리할 수 있게 한다. 본 논문에서 제안하는 방법은 사전에 쪼개진 메쉬 형태가 아닌 동적 재메싱이 가능하기 때문에 가상 수술시뮬레이션이나, 실시간을 요구하는 게임 및 가상환경에서 물리 기반 모델링을 할 때 활용될 수 있다.

▶ **주제어:** 얇은 셸, 옷감 시뮬레이션, 동적 재메싱, 찢기 시뮬레이션, 실시간

- 
- First Author: Jong-Hyun Kim, Corresponding Author: Jong-Hyun Kim
  - \*Jong-Hyun Kim (jonghyunkim@inha.ac.kr), College of Software and Convergence (Dept. of Design Technology), Inha University
  - Received: 2023. 10. 16, Revised: 2023. 11. 21, Accepted: 2023. 11. 30.

## I. Introduction

가상환경에서는 물입을 향상시키기 위한 방법으로 캐릭터와 에이전트의 움직임에 사실적으로 표현하려고 노력하고 있으며, 이 이슈를 해결하는데 있어서 물리 기반 시뮬레이션 기법들이 활용되고 있다[1,2]. 그 중에서 옷감 시뮬레이션은 가상 패션 스튜디오, 옷감 제작, 애니메이션, 영상특수효과 등 다양한 산업에서 영향을 끼치고 있지만, 계산량이 크고 상태변화에 따라 옷감 형태가 달라지기 때문에 고해상도 시뮬레이션에서 찢어짐 효과를 표현하는 것은 어렵다[3-5]. 물리 기반 시뮬레이션 분야에서 옷감에 대한 연구는 주로 연속체역학을 기반으로 표현되는 옷감의 움직임에 초점을 맞추고 있다. 일반적인 모델은 옷감을 탄성 소재의 얇은 쉘로 표현하기 때문에 옷감의 거시적인 기하학 특징만을 모델링할 수 있다[6]. 이 모델은 압축, 신축, 구부러짐을 표현할 수 있기 때문에 다양한 재질을 옷감 시뮬레이션으로부터 표현할 수 있다. 그러나 옷감의 역학만으로 표현하기 어려운 것이 바로 옷감이 찢어지는 현상이다. 찢어지는 옷감은 늘어나는 변형이나 칼과 같은 명시적인 외력과의 상호작용으로 인해 메쉬 요소가 분할되는 것이다 (Fig. 1 참조). 이러한 동작은 재메싱 뿐만 아니라, 역학을 기반으로 움직이는 옷감 시뮬레이션에도 영향을 미친다.



Fig. 1. Real example of tearing cloth.

본 논문에서는 옷감이나 얇은 쉘 시뮬레이션에서 효율적으로 재메싱을 하기 위한 GPU기반 알고리즘을 제안한다. 찢어진 옷감에서는 옷감의 기하학적 구조 형태가 아닌 칼의 형태에 맞게 찢어질 수 있도록 교차된 부분에서 삼각형 기반의 재메싱을 모델링한다. 본 논문의 주요 기술적 기여는 고해상도 옷감의 찢김 시뮬레이션을 실시간에서 처리될 수 있도록 GPU에 친화적인 알고리즘을 소개한다.

## II. Related Work

옷감 시뮬레이션은 물리 기반 시뮬레이션과 애니메이션 분야에서 꾸준히 관심을 받아온 주제이다. Terzopoulos

*et al.*의 연구에서는 유한차분법을 이용하여 옷감과 같은 변형 가능한 표면을 시뮬레이션을 할 수 있는 방법을 제시했다[7-9]. 그 이후로 많은 연구자들이 계산의 효율성 [10,11], 수치 안정성[12,13], 충돌 처리[11,14,15], 옷감의 굽힘[13,16,17,18], 옷감의 비 확장성[14,19-21], 옷감 드레싱[22,23] 등 다양한 측면에서 옷감 시뮬레이션의 문제들을 해결해왔다. 연속체 접근 방식은 대부분의 옷감 특성을 표현하는데 효율적이지만, 옷감의 찢어짐 시뮬레이션은 일반적으로 동적 재메싱 과정이 포함되기 때문에 계산량이 크며, 실시간 애플리케이션에서는 사전에 분할된 메쉬를 이용하지 않고서는 실시간 처리가 어렵다 (Fig. 2 참조). 이러한 접근법은 찢어짐 패턴이 메쉬 해상도에만 의존하기 때문에 계산량은 적지만, 찢어지는 메쉬의 품질이 낮다.

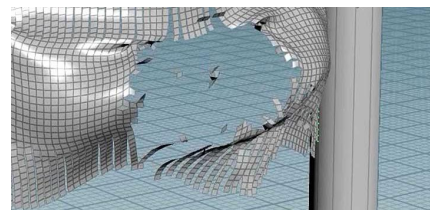


Fig. 2. Tearing simulation of cloth according to a pre-split mesh shape.

갑옷의 탄도 특성을 분석하기 위해 동적 옷감 찢어짐을 시뮬레이션 한 연구도 있다. Zeng *et al.*은 방탄재의 성능을 향상시키기 위해 직조된 원단의 내부 구조를 연구했다 [24]. 특히, 원단의 실 사이에서 발생하는 마찰에 초점을 맞추고 있다. Breen *et al.*은 실교차점을 나타내는 입자 기반 모델을 사용하여 드레이프 직물의 정적 모양을 시뮬레이션했다[25]. Kaldor *et al.*은 니트의 원사 수준을 표현할 수 있는 옷감 시뮬레이션 기법을 제안했다[26].

옷감 시뮬레이션의 역학을 다룬 기법들은 많지만 찢어지는 과정에 초점을 맞춘 연구는 많지 않다. Baraff와 Witkin은 옷감 시뮬레이션의 움직임을 안정적으로 표현하기 위한 수치 해법을 제시했다[12]. 이들은 큰 시간 간격 (Time-step)을 사용할 때 발생하는 수치 불안정성을 방지하기 위한 암시 적분 기법을 소개했다. 옷감은 삼각형 메쉬로 표현되며 각 삼각형의 변형을 기반으로 옷감의 내부 힘을 계산한다. Choi과 Ko는 안정적이고 강력한 상호작용 입자 시스템을 제안하여 효율성과 품질을 개선했다[13]. 본 논문은 옷감의 찢어짐 기법을 설명하는 데 중점을 두었기 때문에 옷감 내부의 동역학은 다루지 않는다.

Table 1. List of symbols used in the paper.

Name	Description
$C_l$ (CutEdge)	Edges created by truncated borders
$C_p$ (CutPoint)	Point where the CutEdges intersect
$C_{info}$ (CutInfo)	Shape information that is cropped from faces and edges. Compute on each Face and Edge in parallel.

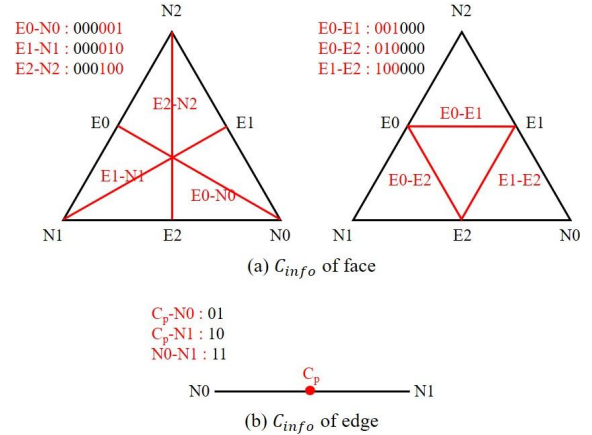
Terzopoulos와 Fleischer는 종이와 옷감 같은 소재를 찢일 수 있는 방법을 소개했다[7,27]. O'Brien와 Hodgins는 유한요소법을 기반으로 부서지기 쉬운 물체의 파괴를 시뮬레이션했으며[28], 이를 연성파괴로 확장했다[29]. Smith *et al.*은 외력에 의해 부서지기 쉬운 물체의 파괴를 효율적으로 시뮬레이션했다[30]. Norton *et al.*은 질량-스프링 시스템을 사용하여 큰 변형을 받아 부서지는 고체를 모델링했으며[31], Hirota *et al.*은 균열이 있는 진흙의 건조 과정을 모델링했다[32]. 대부분 사전에 분할된 메쉬 형태를 이용하며, 동적으로 재메싱을 활용하는 경우 계산량이 너무 커서 실시간 애플리케이션에서는 사용하기 어려운 구조이다.

### III. The Proposed Scheme

#### 3.1. Truncated Information of Elements

본 논문에서는 사용한 기호는 Table1과 같다. 제안하는 알고리즘에서는 효율적으로 재메싱 문제를 풀기 위해 페이스(Face)와 에지(Edge)에서 잘라지는 위치인 CutPoint(이하  $C_p$ )가 2개 이상은 될 수 없도록 제약함으로써 잘라지는 선인 CutEdge(이하  $C_e$ )의 개수가 최소가 되도록 했다. 여기서  $C_e$ 은 자르는 경계선으로 인해 생기는 선을 나타내며,  $C_p$ 는  $C_e$ 들이 교차하면서 생기는 점을 의미한다. 페이스와 에지에서  $C_p$ 가 두 개 이상 존재한다면 교차점들의 평균위치를  $C_p$ 로 설정한다. 페이스와 에지에서 생기는  $C_e$ 의 정보인 CutInfo를  $C_{info}$ 라고 부르겠다.  $C_{info}$ 는 페이스에서는 6비트의 데이터, 에지에서는 2비트의 데이터로 표현할 수 있다.

$E_i$ 는 페이스에서 에지를 나타내며,  $E_1$ 이면 첫 번째 에지를 나타낸다.  $N_i$ 는 페이스를 이루는 노드를 나타내며,  $N_1$ 이면 첫 번째 노드를 나타낸다.  $C_p$ 은 옷감 찢김에 의해서 새롭게 생성된 노드의 위치가  $C_p$ 이다. Fig. 3에서 보듯이 페이스는  $C_p$ 가 어떻게 생성되었느냐에 따라 몇 가지 패턴으로 분할되며, 이러한 정보를 가지고 있는 것이  $C_{info}$ 이다.


 Fig. 3.  $C_{info}$  of primitives ( $E$  : edge,  $N$  : node).

페이스는 3개의 노드와 에지로 구성되어 있으며,  $n$ 번째 노드와  $n$ 번째 에지는 서로 마주보고 있다. 페이스의  $C_{info}$ 에서 3비트는 에지와 노드 사이의  $C_e$ 를 나타내는데 사용되고, 나머지 3비트는 에지와 에지 사이의  $C_e$ 를 나타내는데 사용된다 (Fig. 3a 참조). 에지 자체가 CutEdge일 경우에만 CutInfo 값을 가지게 된다. 에지에  $C_p$ 가 있을 때 0번 노드부터  $C_p$ 까지의  $C_e$ 이면  $C_{info}$ 는  $01_{(2)}$ 이 되고, 1번 노드로부터  $C_p$ 까지면  $10_{(2)}$ 가 된다. 에지가 CutPoint를 가지고 없을 때는 에지 자체가  $C_e$ 이기 때문에 그 타입은  $11_{(2)}$ 가 된다 (Fig. 3b 참조). 단 에지의 이웃한 삼각형 개수가 한 개라면 0으로 통일한다.

#### 3.2. Generate New Nodes

이번 장에서는 새로운 노드 정보를 생성하는 과정에 대해서 설명한다. CutInfo를 가지고 새로운 노드를 생성해 주는데, 이때 노드의 개수는 다음과 같이 계산이 된다. 노드 같은 경우에는 해당 노드 주변의 CutEdge의 개수이고 (Fig. 4 참조), 에지나 페이스 같은 경우에는 CutPoint가 있을 때만 계산된다. 이때 새로 생성되는 노드의 개수는 CutPoint에 이웃한 CutEdge들의 개수가 된다 (Fig. 5와 Fig. 6 참조).

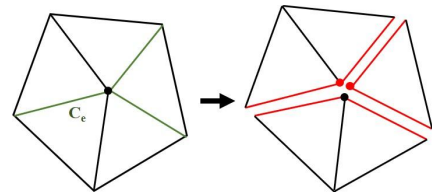


Fig. 4. Generate new nodes at node.

노드에서는 Fig. 4에서 보듯이 3개의 CutEdge가 있을 때 하나의 노드는 3개의 노드를 나눠지게 된다. 이때 주의해야 할 점은 기존의 노드는 보장을 해줘야하기 때문에 최소 1개의 노드는 생성해야 한다.

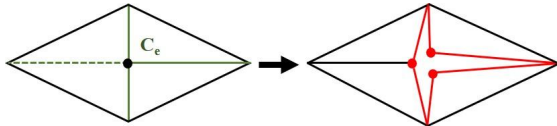


Fig. 5. Generate new nodes at edge.

Edge에서는 Fig. 5에서 보듯이 3개의 CutEdge가 있을 때 하나의 노드는 마찬가지로 3개로 나눠지게 된다. 페이스에서도 5개의 CutEdge가 있을 때 5개의 노드로 나눠지게 된다 (Fig. 6 참조). 결과적으로 CutEdge의 개수 만큼 새로운 노드를 생성한다.

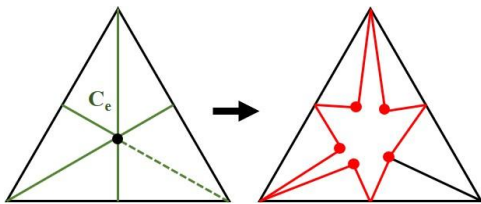


Fig. 6. Generate new nodes at face.

노드와 에지를 분할하는 과정에서 주의할 점이 있다. 삼각형에 온전히 둘러 쌓여있지 않을 때는 주변에 CutEdge가 하나 더 있다고 가정하여 처리한다 (Fig. 7 참조).

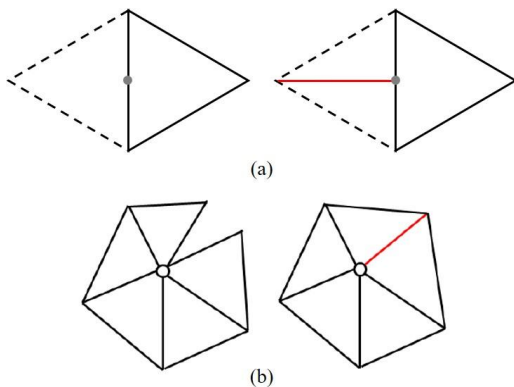


Fig. 7. Case of splitting edges and nodes : (a) when the edge has one neighbor triangle, (b) when the node is not completely surrounded by faces.

각 요소에서 계산한 새로운 노드 개수는 구간 합(Prefix Sum) 알고리즘을 통해 해당 요소에서 생성할 때 노드 인덱스를 저장한다. 생성되는 총 개수 크기의 새로운 노드

배열을 생성하고 각 요소에서 해당하는 노드 인덱스에 노드 정보를 저장한다 (Fig. 8).

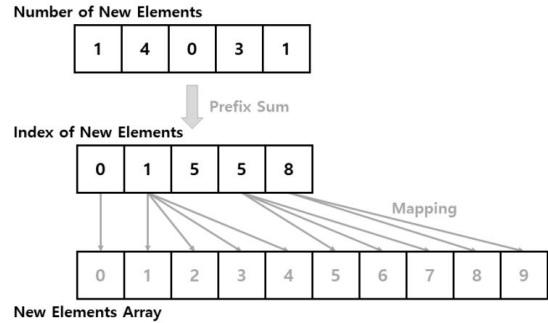


Fig. 8. Stores the number of new elements to be created from existing elements. Then, the index to be created from the element is obtained through the interval sum algorithm. Map new data to the corresponding index in each element.

### 3.3. Generate New Faces

이번 장에서는 새로운 페이스 정보를 생성하는 과정에 대해서 설명한다. 이때 크게 두 가지 경우로 나눠서 계산을 한다 : 1) 페이스에 CutPoint가 있을 경우 (Fig. 9 참조), 2) 페이스에 CutPoint가 없을 경우 (Fig. 10 참조).

페이스에 CutPoint가 있을 경우 크게 3개의 페이스로 분할이 된다 (Fig. 9b 참조). 그리고 에지에 CutPoint가 있을 경우에는 추가적으로 2개의 페이스를 더 분할해준다 (Fig. 9c 참조). 이 과정에서는 최소3개에서 6개까지 페이스가 분할된다.

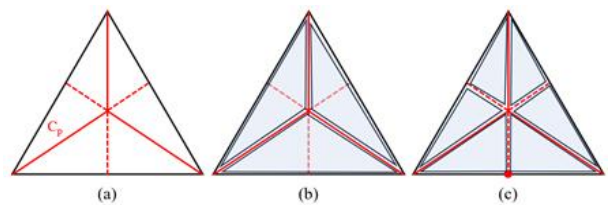


Fig. 9. When a face has a CutPoint.

페이스에 CutPoint가 없을 경우에는 추가적으로 3가지로 더 나눠지게 된다 : 2-1) 에지-노드에 CutEdge가 있을 경우를 먼저 계산해야 한다 (Fig. 10 참조). 크게 2개의 페이스로 분할할 하고 (Fig. 10b 참조), 기존 에지에 CutEdge가 있을 경우에는 추가적으로 2개의 페이스를 더 분할해준다 (Fig. 10c 참조). 이 과정에서 최소2개에서 4개까지 페이스가 분할된다.

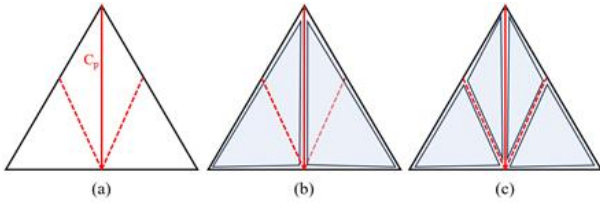


Fig. 10. The face does not have a CutPoint : When an edge-nod has a CutEdge.

2-2) 에지-에지 간의 CutEdge가 한 개이고, CutEdge의 반대 편 Edge에 CutPoint가 없을 경우는 Fig. 11과 같다. 이 같은 경우에는 총 3개의 페이스로 나뉘지게 된다 (Fig. 11b 참조).

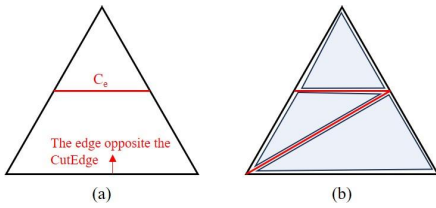


Fig. 11. Edge-edge : single CutPoint.

2-3) 에지-에지 간의 CutEdge가 두 개 있거나 (Fig. 12 참조), 혹은 에지-에지 간의 CutEdge가 한 개 인데, CutEdge 반대편의 에지에 CutPoint가 있을 때는 4개의 페이스로 분할된다 (Fig. 13 참조). 3.3장과 동일하게 각 페이스에 생성될 새로운 페이스의 인덱스와 총 크기의 새로운 페이스 배열을 생성한다 (Fig. 8 참조).

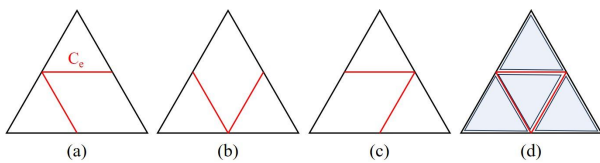


Fig. 12. Edge-edge : two CutEdges.

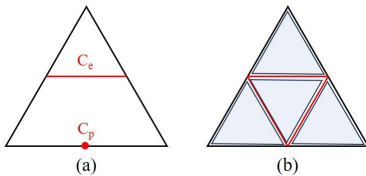


Fig. 13. Edge-edge : single CutEdge.

### 3.4. Face and Node Information

새로운 삼각형의 개수를 페이스마다 병렬적으로 계산하고, 계산된 새로운 페이스의 총 개수만큼 페이스 배열을 만들어 준다. 만들어진 페이스 배열에 삼각형의 3개 노드 인덱스를 매핑을 해준다. 매핑할 때 필요한 정보를

FaceInfo라고 부르겠다. 이 FaceInfo는 새로운 페이스의 3개 노드에 대한 위치 정보이며, 이것은 노드 정보 3개로 이루어져 있다. 노드 정보는 4개의 비트로 구성되어 있으며, FaceInfo는 총 12비트의 데이터로 구성되어 있다 (Fig. 14 참조).

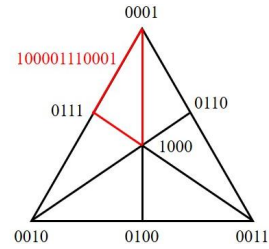


Fig. 14. FaceInfo (red : new face information, black : node information).

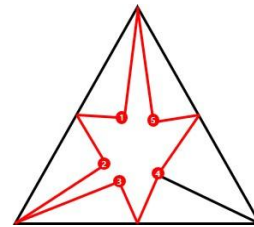


Fig. 15. Node mapping

만들어진 FaceInfo를 가지고 노드를 새로운 페이스 배열에 매핑을 해줘야 하며, 이 과정에서 노드와 에지, 그리고 페이스에서 각각 병렬적으로 매핑을 해준다. 페이스와 에지의 경우에는 CutPoint를 중심으로, 노드는 해당 노드를 중심으로 주변 FaceInfo에 해당하는 인덱스에 새로 만들어지는 노드의 인덱스를 매핑한다. 이때 다음 FaceInfo로 넘어가는 과정에서 공유하고 있는 해당 에지가 CutEdge일 경우 노드 인덱스를 변경해준다. 이 과정은 Fig. 15에서 노드1에서 노드2로 넘어갈 때의 상황이며, 이 과정은 노드4를 제외한 곳에서 모두 적용된다. 페이스를 공유하는 에지가 CutEdge가 아닐 경우에는 인덱스를 유지해주면서 매핑을 하며, 이 과정은 Fig. 15에서 노드4에 해당한다.

## IV. Experiment and Results

본 연구의 결과들을 만들기 위해 실험한 환경은 Intel Core i7-7700K CPU, 32GB RAM, Geforce GTX 1080Ti GPU가 탑재된 컴퓨터를 이용하였다. 본 논문에서 사용한 옷감 시뮬레이션은 제약(Constraint)이 변경되어도 계산

적으로 부담이 없고 병렬프로그래밍에 친화적인 알고리즘인 Jacobian Projective Dynamics를 통해 구현했다. 또한 Wang이 제안한 Chebyshev 반복법으로 가속화했다 [33]. 구간 합 알고리즘은 CUDA thrust 라이브러리의 exclusive\_scan을 이용하여 계산했다. 옷감을 찢는 과정에서는 본 논문에서 제안한 삼각형 기반 동적 재메쉬 패턴을 CUDA 스테드(Threads)로 전달하여 계산했다.

Fig. 16은 다른 해상도를 가진 3개의 옷감을 여섯 방향으로 자른 실시간 장면이다. 삼각형의 원본 형태에 맞게 잘라지기 보다는 컷 형태에 맞게 자연스럽게 잘라지는 결과를 보여주고 있다. 실시간 애플리케이션에서는 성능을 보장하기 위해 일반적으로 삼각형 메시에 맞게 요소를 자르지만, 본 논문에서 제안하는 방법은 옷감의 해상도와 관계없이 동적 재메싱을 실시간으로 처리했다. Table 2에서 Our method(GPU)에서는 제안한 방법을 통해 옷감을 찢었을 때 측정한 계산시간이며, 이를 통해 실시간 처리가 충분히 가능함을 보여준다.

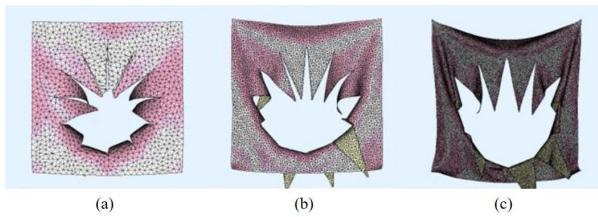


Fig. 16. A scene where the fabric is cut in six directions : (a) low resolution, (b) medium resolution, (c) : high resolution.

Fig. 17은 정해진 컷 형태가 아닌 사용자가 실시간으로 옷감을 자르고 있는 장면이다. 마우스 드래그&드롭을 이용하여 컷을 수평형태로 정의하고 있으며 그에 따라 동적 재메싱 및 옷감 시뮬레이션이 실시간으로 처리되고 있음을 잘 보여주고 있다.

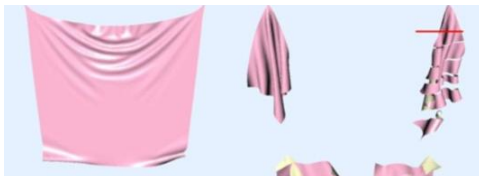


Fig. 17. Real-time cloth cutting scene.

Fig. 18은 사용자가 마우스로 선택한 정점을 뜯어내는 장면이며, 뜯어내는 외력이 임계 값 이상이면 찢어지도록 표현했다. 제안하는 방법은 GPU기반 동적 재메싱이기 때문에 옷감 해상도에 의존하지 않고 디테일하게 찢어지는 패턴을 실시간으로 잘 표현했다.

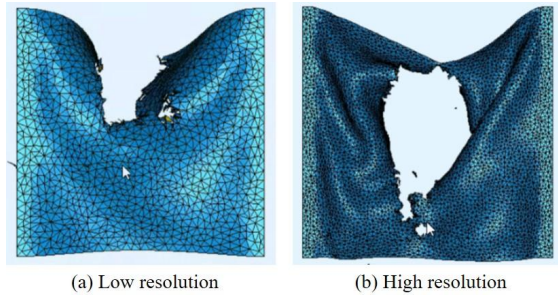


Fig. 18. Real-time cloth tearing.

Fig. 19는 열에 의해서 옷감이 수축되거나 녹은 장면을 표현한 결과이다. 앞에서 보여준 결과가 다르게 삼각형 기반에서 열 확산이 일어나며 임계 값 이상의 열이 발생하면 녹거나 수축되는 형태가 나타나도록 모델링했다. 이 과정에서도 앞에서 보여준 것 같이 모든 결과에서 실시간으로 옷감의 찢김 시뮬레이션이 잘 동작했다. 본 논문에서 실험한 옷감의 페이스 개수와 성능은 다음과 같다.

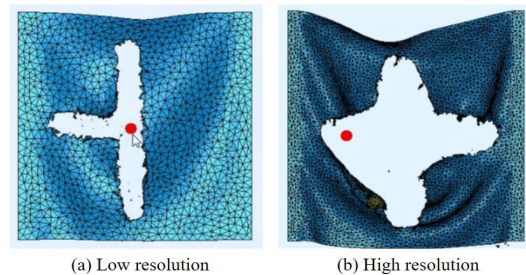


Fig. 19. Real-time cloth tearing by heat diffusion.

Table 2에서 보듯이 본 논문에서 제안한 GPU기반 방법은 같은 알고리즘을 CPU에서 비교했을 때, 그리고 이전 기법과 비교했을 때 월등히 빠른 결과를 보여주었다[5]. 이전 기법에서는 동적인 재메싱이 아닌, 입력 메시 해상도에 맞춰서 찢어지기 때문에 제안하는 방법보다 찢어지는 표면에서의 품질이 다소 떨어지지만, 속도가 빠르기 때문에 제안하는 방법과 성능을 비교했다. 그럼에도 불구하고 이전 기법은 CPU기반에서 동작하는 방법이기 때문에 GPU에 초점을 맞춘 제안하는 방법이 계산속도 측면에서 확연히 빠른 결과를 보여주었고, 모두 실시간 처리가 가능하기 때문에 가상현실이나, 메타버스와 같이 실시간을 요구하는 환경에서 활용성이 높을 거라 기대한다. 또한, 찢어지는 패턴이 많을수록 상대적으로 계산량이 많아지기 결과가 나타났지만, GPU기반 알고리즘에서 그 차이가 크지 않았기 때문에 고해상도에서도 실시간 옷감 찢기가 가능했다.

Table 2. Performance.

Fig.	Number of face	Computation time		
		Our method (GPU)	Our method (CPU)	Previous method[5]
16a	2,822	4.3 ms	3.1 s	3.0 s
16b	9,022	9.1 ms	9.4 s	9.2 s
16c	38,298	9.1 ms	17.5 s	16.4 s
17	38,298	10.1 ms	23.8 s	22.0 s
18a	2,822	4.3 ms	4.3 s	3.9 s
18b	9,022	9.1 ms	11.9 s	11.3 s
19a	2,822	4.3 ms	3.4 s	3.1 s
19b	9,022	9.1 ms	10.3 s	10.1 s

## V. Conclusion

본 논문에서는 옷감의 찢김 시뮬레이션을 효율적으로 처리하기 자료구조와, 이를 GPU로 개발할 수 있는 새로운 프레임워크를 제안했다. 기존에 방법들은 동적 재메싱을 처리하기 위해 지오메트리와 연결 정보를 업데이트 하는 방식에 초점을 두었지, 이를 GPU에 맞게 알고리즘을 확장하는 방법은 본 논문에서 제시한 방법이 처음이다. 특히, 매번 메쉬의 정점과 페이스 정보가 업데이트되기 때문에 달라지는 메쉬 구조는 단순히 GPU로 개발할 수 있는 구조가 아니다. 본 논문에서 제안하는 방법은 얇은 쉘 기반이지만 향후 다면체 내부로 확장하여 GPU기반 파괴 시뮬레이션 기법을 볼륨 기반의 메쉬로 확장할 계획이다. 제안하는 방법은 실시간을 요구하는 가상 수술 시뮬레이션, 가상현실, 게임, 메타버스 등에서 사용자와의 인터랙션을 요구하는 다양한 콘텐츠에 활용될 수 있을 거라 생각한다.

## REFERENCES

[1] Meseure, Philippe, Jérôme Davanne, Laurent Hilde, Julien Lenoir, Laure France, Frédéric Triquet, and Christophe Chaillou. "A physically-based virtual environment dedicated to surgical simulation." In *Surgery Simulation and Soft Tissue Modeling: International Symposium, IS4TM 2003 Juan-Les-Pins, France, 2003, Proceedings*, pp. 38-47. DOI: 10.1007/3-540-45015-7\_4

[2] Keckeisen, Michael, Stanislav L. Stoev, Matthias Feurer, and Wolfgang Straßer. "Interactive cloth simulation in virtual environments." In *IEEE Virtual Reality*, pp. 71-78. IEEE, 2003. DOI: 10.1109/VR.2003.1191123

[3] Tan, Qingyang, Zherong Pan, Lin Gao, and Dinesh Manocha. "Realtime simulation of thin-shell deformable materials using CNN-based mesh embedding." *IEEE Robotics and Automation*

*Letters* 5, No. 2, 2020, pp. 2325-2332. DOI: 10.48550/arXiv.1909.12354

[4] Metaaphanon, Napaporn, Yosuke Bando, Bing-Yu Chen, and Tomoyuki Nishita. "Simulation of tearing cloth with frayed edges." In *Computer Graphics Forum*, vol. 28, no. 7, pp. 1837-1844. Oxford, UK: Blackwell Publishing Ltd, 2009. DOI: 10.1111/j.1467-8659.2009.01561.x

[5] Souza, Marco Santos, Aldo Wangenheim, and Eros Comunello. "Fast simulation of cloth tearing." *SBC Journal on Interactive Systems* 5, No. 1, 2014, pp. 44-48. DOI: 10.5753/jis.2014.642

[6] Guo, Qi, Xuchen Han, Chuyuan Fu, Theodore Gast, Rasmus Tamstorf, and Joseph Teran. "A material point method for thin shells with frictional contact." *ACM Transactions on Graphics (TOG)* 37, No. 4, 2018, pp. 1-15. DOI: 10.1145/3197517.3201346

[7] Terzopoulos, Demetri, John Platt, Alan Barr, and Kurt Fleischer. "Elastically deformable models." In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pp. 205-214. 1987. DOI: 10.1145/37402.37427

[8] Terzopoulos, Demetri, and Kurt Fleischer. "Modeling inelastic deformation: viscoelasticity, plasticity, fracture." In *Proceedings of the 15th annual conference on Computer graphics and interactive techniques*, pp. 269-278. 1988. DOI: 10.1145/378456.378522

[9] Terzopoulos, Demetri, and Kurt Fleischer. "Deformable models." *The visual computer* 4, No. 6, 1988, pp. 306-331. DOI: 10.1007/BF01908877

[10] Volino, Pascal, Martin Courchesne, and Nadia 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*, pp. 137-144. 1995. DOI: 10.1145/218380.218432

[11] Volino, Pascal, and N. Magnenat Thalmann. "Implementing fast cloth simulation with collision response." In *Proceedings Computer Graphics International 2000*, pp. 257-266. IEEE, 2000. DOI: 10.1109/CGI.2000.852341

[12] Baraff, David, and Andrew Witkin. "Large steps in cloth simulation." In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pp. 43-54. 1998. DOI: 10.1145/280814.280821

[13] Choi, Kwang-Jin, and Hyeong-Seok Ko. "Stable but responsive cloth." In *ACM SIGGRAPH 2005 Courses*, pp. 1-es. 2005. DOI: 10.1145/566654.566624

[14] Bridson, Robert, Ronald Fedkiw, and John Anderson. "Robust treatment of collisions, contact and friction for cloth animation." In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 594-603. 2002. DOI: 10.1145/566654.566623

[15] Baraff, David, Andrew Witkin, and Michael Kass. "Untangling cloth." *ACM Transactions on Graphics (TOG)* 22, No. 3 2003,

- pp. 862-870. DOI: 10.1145/1201775.882357
- [16] Grinspun, Eitan, Anil N. Hirani, Mathieu Desbrun, and Peter Schröder. "Discrete shells." In Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 62-67. 2003.
- [17] Bridson, Robert, Sebastian Marino, and Ronald Fedkiw. "Simulation of clothing with folds and wrinkles." In ACM SIGGRAPH 2005 Courses, pp. 3-es. 2005.
- [18] Zhou, Chuan, Xiaogang Jin, and Charlie CL Wang. "Shear buckling and dynamic bending in cloth simulation." *Computer Animation and Virtual Worlds* 19, No. 3-4, 2008, pp. 493-503. DOI: 10.1002/cav.253
- [19] Provot, Xavier. "Deformation constraints in a mass-spring model to describe rigid cloth behaviour." In *Graphics interface*, pp. 147-147. Canadian Information Processing Society, 1995.
- [20] Müller, Matthias, Bruno Heidelberger, Marcus Hennix, and John Ratcliff. "Position based dynamics." *Journal of Visual Communication and Image Representation* 18, No. 2, 2007, pp. 109-118. DOI: 10.1016/j.jvcir.2007.01.005
- [21] Goldenthal, Rony, David Harmon, Raanan Fattal, Michel Bercovier, and Eitan Grinspun. "Efficient simulation of inextensible cloth." In *ACM SIGGRAPH 2007 papers*, pp. 49-es. 2007. DOI: 10.1145/1276377.1276438
- [22] Carignan, Michel, Ying Yang, Nadia Magnenat Thalmann, and Daniel Thalmann. "Dressing animated synthetic actors with complex deformable clothes." *ACM Siggraph Computer Graphics* 26, No. 2, 1992, pp. 99-104. DOI: 10.1145/142920.134017
- [23] Volino, Pascal, N. Magnenat Thalmann, Shen Jianhua, and Daniel Thalmann. "An evolving system for simulating clothes on virtual actors." *IEEE Computer Graphics and Applications* 16, No. 5, 1996, pp. 42-51. DOI: 10.1109/38.536274
- [24] Zeng, X. S., V. B. C. Tan, and V. P. W. Shim. "Modelling inter-yarn friction in woven fabric armour." *International Journal for Numerical Methods in Engineering* 66, No. 8, 2006, pp. 1309-1330. DOI: 10.1002/nme.1596
- [25] Breen, David E., Donald H. House, and Michael J. Wozny. "Predicting the drape of woven cloth using interacting particles." In Proceedings of the 21st annual conference on Computer graphics and interactive techniques, pp. 365-372. 1994. DOI: 10.1145/192161.192259
- [26] Kaldor, Jonathan M., Doug L. James, and Steve Marschner. "Simulating knitted cloth at the yarn level." In *ACM SIGGRAPH 2008 papers*, pp. 1-9. 2008. DOI: 10.1145/1399504.1360664
- [27] Terzopoulos, Demetri, and Kurt Fleischer. "Modeling inelastic deformation: viscoelasticity, plasticity, fracture." In Proceedings of the 15th annual conference on Computer graphics and interactive techniques, pp. 269-278. 1988. DOI: 10.1145/378456.378522
- [28] O'brien, James F., and Jessica K. Hodgins. "Graphical modeling and animation of brittle fracture." In Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp. 137-146. 1999. DOI: 10.1145/311535.311550
- [29] O'brien, James F., Adam W. Bargteil, and Jessica K. Hodgins. "Graphical modeling and animation of ductile fracture." In Proceedings of the 29th annual conference on Computer graphics and interactive techniques, pp. 291-294. 2002. DOI: 10.1145/566654.566579
- [30] Smith, Jeffrey, Andrew Witkin, and David Baraff. "Fast and controllable simulation of the shattering of brittle objects." In *Computer Graphics Forum*, Vol. 20, No. 2, pp. 81-91. Oxford, UK and Boston, USA: Blackwell Publishers Ltd, 2001. DOI: 10.1111/1467-8659.00480
- [31] Norton, Alan, Greg Turk, Bob Bacon, John Gerth, and Paula Sweeney. "Animation of fracture by physical modeling." *The visual computer* 7 (1991): 210-219. DOI: 10.1007/BF01900837
- [32] Hirota, Koichi, Yasuyuki Tanoue, and Toyohisa Kaneko. "Generation of crack patterns with a physical model." *The visual computer* 3, No. 14, 1998, pp. 126-137. DOI: 10.1007/s003710050128
- [33] Wang, Huamin. "A chebyshev semi-iterative approach for accelerating projective and position-based dynamics." *ACM Transactions on Graphics (TOG)* 34, No. 6, 2015, pp. 1-9. DOI: 10.1145/2816795.2818063

## Authors



Jong-Hyun Kim received the B.A. degree in the Department of Digital Contents at Sejong University in 2008. He received M.S. and Ph.D. degrees in the Department of Computer Science and Engineering at Korea University,

in 2010 and 2016. Prof. Kim is an Associate Professor in the College of Software and Convergence (Dept. of Design Technology) in Inha University. His current research interests include fluid animation and virtual reality.