

## 2차원 패턴 디자인 모듈과 Octree 공간 분할 방법을 이용한 3차원 의복 시뮬레이션 시스템에 관한 연구

김주리<sup>\*</sup>, 정석태<sup>\*\*</sup>, 정성태<sup>\*\*\*</sup>

### 요 약

본 논문에서는 의복의 2차원 패턴을 이용하여 3차원 의복 모델을 제작하고 생성된 3차원 의복을 3차원 인체 모델에 자동으로 착의하는 3차원 패션 디자인 시스템을 제안한다. 이 시스템에서는 먼저 2차원 격자의 코너 점을 이용하여 2차원 옷감 조각을 디자인한 다음에 2차원 옷감 조각들 사이에 재봉되어야 할 제약점들을 기술함으로써 3차원 의복을 디자인한다. 그 다음에는 3차원 인체 모델 파일과 3차원 의복을 읽어 들인 다음 질량-스프링 모델에 기반한 물리적 시뮬레이션에 의해 의복을 착용한 3차원 모델을 생성한다. 본 논문의 시스템은 사실적인 시뮬레이션을 위하여 인체 모델을 구성하는 삼각형과 의복을 구성하는 삼각형 사이의 충돌을 검사하고 반응 처리를 수행하였다. 인체를 구성하는 삼각형의 수가 매우 많으므로, 이러한 충돌 검사 및 반응 처리는 많은 시간을 필요로 한다. 이 문제를 해결하기 위하여, 본 논문에서는 Octree 공간 분할 기법을 이용하여 충돌 검사 및 반응 처리 수를 줄였으며 수초 이내에 가상 인체 모델에 의복을 입힐 수 있었다.

## A Study on 2D Pattern Design Module and 3D Cloth Simulation System based on Octree Space Subdivision Method

Ju-Ri Kim<sup>\*</sup>, Suck-Tae Joung<sup>\*\*</sup>, Sung-Tae Jung<sup>\*\*\*</sup>

### ABSTRACT

This paper proposes a 3D fashion design system that generates a 3D clothes model by using 2D patterns of clothes and drapes the 3D clothes model on a 3D human model. In the proposed system, 2D patterns of clothes are designed by selecting corner points of 2D mesh. After designing 2D patterns, a 3D clothes model is designed by describing the control points to be connected between 2D patterns. The proposed system reads a 3D human body model file and the designed 3D clothes model and creates a 3D human model putting on the clothes by using the mass-spring model based physical simulation. It calculates collision and reaction between the triangles of human body model and those of clothes for realistic simulation. Because the number of triangles is very large, the collision and reaction processing need a lot of time. To solve this problem, the proposed system decreases the number of collision and reaction processing by using the Octree space subdivision technique. It took a few seconds for generating a 3D human model putting on the designed 3D clothes.

**Key words:** 2D Pattern design module(2차원 패턴 디자인 모듈), 3D cloth simulation(3차원 의복 시뮬레이션), mass-spring model(질량-스프링 모델)

※ 교신저자(Corresponding Author): 김주리, 주소: 전북 익산시 신용동 344-2번지(570-579), 전화: 063)850-6886, FAX: 063)850-8009, E-mail: cyanic@wonkwang.ac.kr  
접수일: 2006년 4월 11일, 완료일: 2007년 3월 7일

<sup>\*</sup>준회원, 원광대학교 컴퓨터공학과

<sup>\*\*</sup>중신회원, 원광대학교 전기전자 및 정보공학부 교수

(E-mail: stjoung@wonkwang.ac.kr)

<sup>\*\*\*</sup>중신회원, 원광대학교 전기전자 및 정보공학부 교수

(E-mail: stjung@wonkwang.ac.kr)

※ 연구지원사항: 이 논문은 2006년도 원광대학교의 교비 지원에 의해서 수행 됨.

## 1. 서론

패션 시장에서 과거에는 맞춤복 위주의 의상실이 의류시장의 중심을 이루고 있었으나, 의류산업의 발달과 더불어 점차 기성복 위주의 대량 생산체제가 이루어지게 되었다. 경제 발전과 더불어 문화적 수준도 향상되었고, 각 개인의 생활 스타일도 변화하여 개성이 중요시되는 사회가 되었다. 이에 따라 패션에 대한 소비자의 인식도 바뀌어, 오늘날 대부분의 패션 산업은 특정 기준에 따른 시장의 세분화가 이루어져 각기 다른 취향의 소비자를 만족시키는 다양한 상품이 갖추어지고 있다.

더욱이 오늘날 인터넷을 이용한 전자상거래, 가상쇼핑몰에서 의류의 구매 추세가 본격화되어, 가상패션코드의 필요성이 증대되고 있다. 현재의 가상패션코드는 대부분 2차원 영상에 의존하고 있다[1,2]. 그러나, 고객의 체형과 감성에 알맞은 패션코드를 통해 고객의 만족도를 극대화하기 위해서는, 의복의 3차원 모델의 생성과 3차원 드래핑을 통해서 가상공간에서의 3차원 패션코드 시스템의 개발이 필수적이다.

본 논문에서는 의복의 2차원 패턴 모델 제작 시스템을 이용하여 3차원 의복 제작에 필요한 2D 패턴을 도안하고 자동 생성하며 3D 시뮬레이션에 필요한 제약점과 제약점을 지정할 수 있도록 설계하였다. 디자인된 의복 부품별 2차원 모델은 봉합과정을 거쳐 3차원 인체 모델이 주어진 상태에서 충돌 처리 과정을 거쳐 의복이 모델에 입혀지는 3차원 시뮬레이션 과정을 보여줄 수 있다.

본 논문의 구성은 다음과 같다. 제2절은 전체적인 시스템 특징과 구현 방법에 대하여 설명하고, 제3절에서는 샘플 의상을 제작하기 위한 2차원 패턴을 설계하여 거기에 필요한 재봉과 제약점 설정에 대하여 설명한다. 제4절에서는 만들어진 옷감이 인체를 타고들지 않도록 옷과 인체 사이의 충돌 검사를 Octree 공간 분할 방법을 이용하여 처리해 주는 방법을 제안한다. 그리고 제5절에서는 2차원 패턴을 인체 모델에 3차원으로 입혀 시뮬레이션 하는 과정과 성능 평가의 결과를 제시한다. 마지막으로 제6절에서는 결론과 향후 필요한 연구과제에 대하여 기술한다.

## 2. 시스템 특징 및 구현

### 2.1 시스템 특징

실제 옷감 패턴을 디지털화하기 위해서는 그림 1과 같은 디지털화 도구를 사용하거나 큰 스캐너를 이용한다. 패턴 디자인 시스템이 없다면 디자이너는 매우 큰 종이 위에다 실제 크기의 패턴들을 직접 그려나가야 한다. 또 같은 재봉선에 해당하는 두 패턴의 외곽선 길이가 똑같이 디자인되어야 하므로 수작업으로는 많은 시행착오를 거쳐야 한다. 하지만 패턴 디자인 시스템은 이러한 검사를 자동으로 수행한다. 또한 여러 종류의 의상에 대한 많은 샘플 패턴들을 제공하므로 디자이너는 이를 변경하여 쉽게 패턴을 디자인할 수 있다.

본 논문의 시스템은 실제 사람이 의상을 입어보는 과정을 가상의 공간에서 직접 체험해 볼 수 있는 시스템으로 실제 옷을 입는 것과 비슷한 환경을 구성하기 위해 인체 모델과 의복 모델을 설계하였다.

그림 1은 2차원 패턴 모델 제작 시스템이다. 본 시스템은 도식화에 필요한 Toolbar와 Style, Fabric, Color bar 등으로 구성되어 있으며, 또한 3차원 의복 제작에 필요한 제어상자와 작업 창 그리고 3차원 의복을 착의 시뮬레이션 할 수 있는 기능들을 제공하고 있다. 의복 제어상자에는 제약점을 추가/삭제 할 수 있는 기능과 제약점 좌표 값 보정을 위한 기능, 바느질 제약점 추가/삭제 기능들도 제공하고 있다. 또한 작업 창에서는 디자이너가 직접 2차원 패턴을 디자인 할 수 있는 환경을 제공한다[3].

의복 시뮬레이션에서 가장 중요한 요소는 충돌처리이다. 의복 시뮬레이션에서 충돌처리 과정이 가장

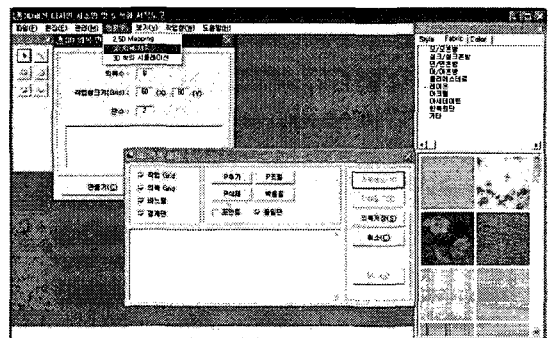


그림 1. 2차원 패턴 모델 제작 시스템

많은 처리 시간을 소모하며 이를 줄이기 위해서 Octree 공간 분할 방법을 사용하여 의복과 인체 모델의 충돌 처리를 해결하였다.

### 2.2 시스템 구현

3차원 의복 모델 최종 결과물을 얻기 위한 처리과정은 그림 2와 같다. 처리 과정을 보면 작업창 행렬을 생성하는데 필요한 가로, 세로 크기와 의복 수, 면수 등을 입력 받으며, 입력 받은 데이터를 가지고 행렬을 생성해 작업창을 구성한다. 이때 사용자로부터 2차원 패턴을 디자인하고, 디자인 된 패턴들 간의 박음질 선을 생성한다. 완성된 2차원 패턴을 바탕으로 수치 데이터를 생성해 3차원 의복 모델을 만든다. 이와 같은 과정을 거쳐 3차원 의복 모델 데이터가 생성되면 3차원 인체 모델을 이용하여 3차원 의복을 입히는 작업을 보여준다. 구현에 필요한 알고리즘과 모듈은 패턴 디자인 모듈, 3차원 의복 모델 데이터 읽기/저장 모듈, 의복 착의 시뮬레이션 모듈 등이 있다 [4,5].

## 3. 2차원 패턴 디자인 설계

### 3.1 2차원 옷감 조각 디자인 모듈

2차원 옷감 조각을 제작하는 과정에서는 옷감의 외형과 물리적인 특성을 설정하고, 재봉 선을 설정한다. 옷감의 외형을 디자인하기 위해서는 옷감의 외곽선에 대한 정보를 입력해야 한다. 입력된 외곽선으로부터 옷감의 기하학적/물리적 모델에 맞는 옷감 구조를 구하는데, 그림 3과 같이 임의의 삼각형들의 집합으로 옷감을 구성하거나, Provot의 옷감 모델에 맞게 격자 형태로 옷감을 구성한다[6,7]. 이렇게 기하학적 구조를 정한 뒤, 옷감에 맞는 물리적인 특성을 설

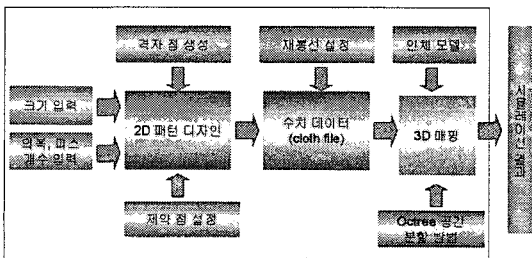


그림 2. 3차원 의복 제작 과정

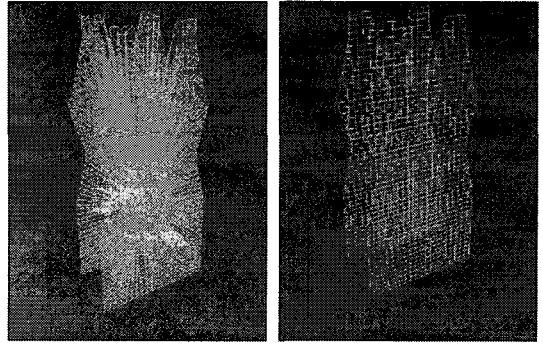


그림 3.  $m \times n$  질량 점의 직사각형 토폴로지 메쉬 형태

정한다. 입력되는 특성들은 시스템이 사용하는 옷감 모델에 따라 다르지만, 일반적으로 각 입자의 질량, 옷감이 늘어나거나 휘어지는 정도 등이 입력된다 [8-10].

본 논문은 그림 4처럼 2차원 격자 위에서 조각의 외곽선을 그리는 편집 프로그램을 개발하였다. 연속적으로 격자점을 선택해 가면서 옷감 조각의 외곽선을 그릴 수 있다. 외곽선이 모두 그려지면 옷감을 구성하는 스프링과 질량 점들을 계산하고, 외곽선 안의 격자점들이 질량 점들에 해당하며, 질량 점들 사이에 스프링이 연결된다. 외곽선 안의 질량 점들을 계산하기 위해서 다각형의 안과 밖을 결정해 주는 조단 곡선 이론(Jordan Curve Theorem)[11,12]을 사용한다. 실제로 질량 점은 외곽선 내부뿐만 아니라 외곽선과 격자선의 교차점도 포함한다. 외곽선과 격자선의 교차점으로 할 경우 복잡한 스프링 구조로 인하여 전체 시스템의 성능을 저하시킨다. 따라서 좀 더 간단한 구조를 필요로 하게 되는데 본 논문에서는 전체 시스템의 성능을 좀 더 향상시키기 위하여 그림 5와 같이 외곽선과의 교차로 생기는 질량 점들을 가장 가까이 있는 격자점과 연관하여 압축하는 방법을 사용하고 있다. 그림 5의 동그라미는 연관 관계를 나타낸다. 진한 점이 격자점이고 흐린 점이 교차로 생기는 질량 점들이다. 각 격자점은 연관된 교차점 중 가장 가까이 있는 점으로 대체되고, 이 점이 연관 관계에 있는 점들을 대표하게 된다. 이렇게 해서 격자점 하나가 연관된 교차점을 모두 대표하고 스프링 계산에 이용된다[7,13].

다음은 재봉 선을 그리는 과정이다. 의상은 여러 옷감 조각들이 재봉되어 제작하므로 서로 재봉 되어야 할 옷감 쌍의 외곽선 부분을 재봉 선으로 설정한다.

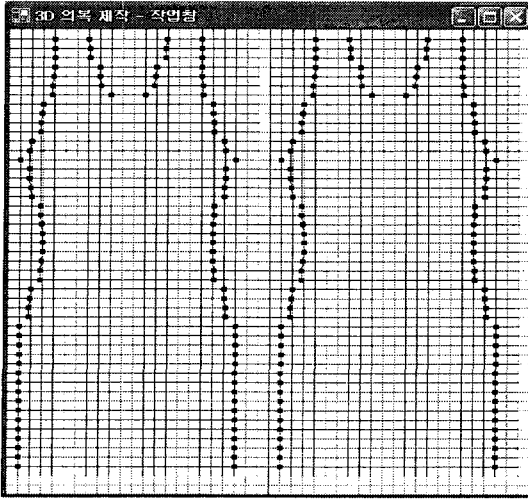


그림 4. 2차원 패턴 디자인 샘플

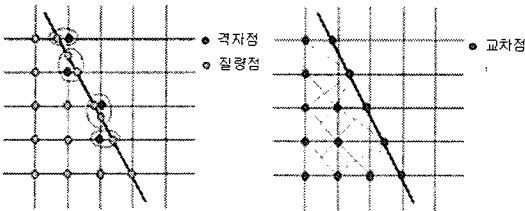


그림 5. 압축 방법을 사용한 경우의 스프링 구조

재봉 선을 설정하는 과정은 간단하다. 그림 4와 같이 재봉될 조각들의 외곽을 재봉 선으로 지정하고, 재봉 선끼리 쌍을 지어주면 된다. 이 때 각 재봉 선은 같은 수의 질량 점을 가지고 있어야 하며, 재봉 선 사이는 길이가 0인 스프링으로 연결된다.

### 3.2 재봉과 제약점 설정

본 절에서는 2차원 패턴 조각들을 인체 모델에 입히는 방법을 설명한다. 이 방법은 크게 두 가지로 나누어진다. 첫 번째는 Maya나 3D MAX 프로그램을 이용하여 의상의 한 부분을 인체 모델에 부착시켜 인체 동작과 똑같이 움직이도록 하는 방법이고, 두 번째는 여러 개의 2차원 조각들을 재봉하여 인체 모델에 입히는 방법이다. 본 논문에서는 2차원 조각 패턴을 이용하여 구현하였다. 그림 4에서 보는 것과 같이 2차원 패턴은 여러 개의 제약점, 제약점과 제약점을 이은 선, 제약점과 제약점을 이은 바느질 선으로 이루어져 있으며, 만들어진 2차원 패턴 디자인은 옷감의 움직

임에 필요한 질량-스프링 구조를 생성을 하여 파일(\*.cloth) 시스템을 사용하여 저장하고 이 파일은 3차원 모델에 적용하여 시물레이션 할 수 있다.

질량-스프링 구조를 적용하는 이유는 힘의 계산이 쉽고 간단한 미분 방정식을 풀어 입자의 위치를 계산할 수 있으므로 구현이 쉽고 옷감의 움직임을 빠르게 계산할 수 있기 때문이다.

그림 6은 3차원 원피스 모델이 저장된 파일을 메모장에서 읽어온 그림이다. 데이터 구조를 보면 크게 의복 정보와 바느질 정보로 나누어진다. 의복 정보에는 의복의 수, 작업 영역 크기, 면수(앞/뒤), 제약점 데이터, 제약점 보정 데이터 등으로 이루어져 있으며, 바느질 정보에는 바느질 할 제약점 들을 정의하고 있다.

3차원 인체 모델에 의상을 입히기 위해서는 여러 조각들을 인체 모델 주위에 적절하게 위치시키고, 재봉 과정을 거쳐 옷감 조각들을 붙여준다. 여기서 제약점은 인체 모델의 특정 부분의 고정된 의상의 입자 점을 뜻하는데, 즉 제약점은 인체의 특정 부분의 지역 좌표계에 고정된다. 따라서 2차원 조각들이 인체의 특정 부분에 고정되게 한다.

기존 연구[14]에서는 제약점을 사용하면 의상이 인체를 뚫고 지나가는 현상을 막을 수 있지만 사실성을 떨어뜨리는 단점이 있었다. 본 논문에서는 인체를 뚫고 지나가는 현상을 막을 뿐만 아니라 인체와 의복 간의 충돌 계산량을 줄여줌으로써 좀 더 자연스러운 사실성을 더해주고 있다. 그림 7은 4가지 디자인의 의상을 시물레이션하기 위해 각각에 설정된 제약점 들을 보여준다.



그림 6. 3차원 의복의 수치 데이터

재봉은 그림 7처럼 옷감 조각들을 적절한 위치로 옮기고, 시뮬레이션 과정을 통해 재봉 선으로 연결된 옷감 조각들을 붙이는 과정이다. 재봉에서의 시뮬레이션 과정은 두 가지로 나뉜다. 첫째는 제약점과 중력을 무시하고 길이 보정 없이 많은 수의 움직임 계산을 거치는 비부하(unload) 과정이고, 둘째는 제약점과 중력을 고려하고 길이 보정 과정을 추가하여 적은 수의 움직임 계산 과정을 거치는 부하(load) 과정이다. 비부하 과정에서는 제약점과 중력을 무시하고, 많은 움직임 계산 과정을 통해 옷감이 전체적인 균일 상태로 가도록 한다. 비부하 과정에서는 인체 모델에 의상이 제대로 입혀지도록 제약점과 중력을 고려하여 옷감이 전체적인 균일 상태로 가도록 한다. 부하 과정에서는 인체 모델에 의상이 제대로 입혀지도록 하기 위해서는 제약점들의 위치를 적절히 조정해야 한다. 부하 과정은 애니메이션에서의 시뮬레이션 방법과 같다. 이 두 과정을 번갈아 적용하여 적절하게 옷감 조각들을 재봉해야 한다. 재봉이 끝나면 재봉 선 사이의 스프링을 제거하고, 옷감 전체를 하나의 조각으로 합쳐 의상 제작을 완료 할 수 있다 [14].

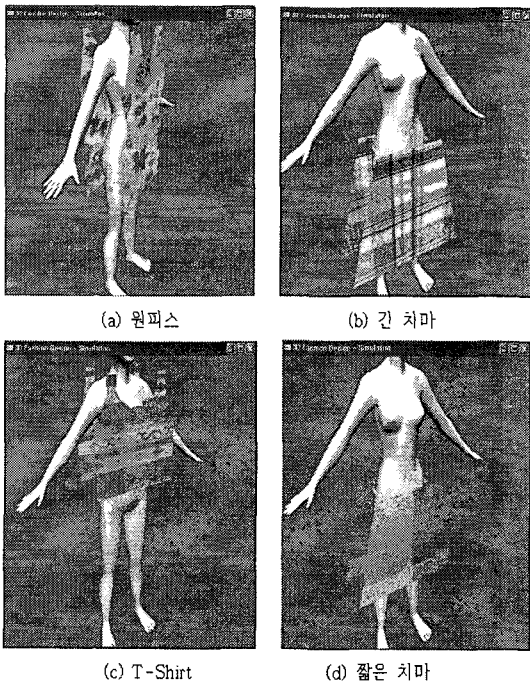


그림 7. 각 의상의 시뮬레이션을 위한 제어점 설정

## 4. 충돌 처리

### 4.1 충돌 검사

의복 착용에서 옷감이 인체를 파고들면 옷감 사이로 인체가 보이게 되고 이것은 현실성을 떨어뜨리기 때문에 정확한 충돌 검사와 반응 처리가 필요하다. 옷과 인체 사이의 충돌 여부는 매 시간마다 옷의 입자가 인체 표면의 안쪽에 위치하는지의 여부로 판단할 수 있다. 기존 연구[14]에서는 충돌 검사와 반응 처리 속도를 높이기 위한 방법으로 타원과 구를 사용한 바운딩 볼륨을 이용하고 있다. 그러나 이 방법은 충돌 검사에 대한 시간은 줄일 수 있으나, 옷감이 인체로부터 떨어져서 움직이는 문제가 발생할 수 있기 때문에, 본 논문에서는 인체를 구성한 다각형과 옷의 입자 사이의 충돌 검사를 함으로써 보다 정확한 시뮬레이션을 수행하였다. 인체 모델을 그림 8과 같이 Octree 공간 분할 방법을 이용하여 공간 분할하고 계층적으로 옷감을 이루는 입자 점들 사이의 충돌을 검사하였다.

Octree 공간 분할 방법이란 입체의 형태의 입체를 공간적으로 분할하는 일종의 화면 분할법 중의 하나로 공간을 8조각으로 계속 나누어 관리하는 방법을 말한다. 각 노드당 최대 8개의 자식들을 가질 수 있는 트리구조로 노드의 자식들은 부모입방체를 8등분하는 동일한 크기의 입방체들이 된다. 공간 분할은 자식노드를 통한 공간 분할로 입방체가 어떤 특정한 크기가 될 때까지, 또는 각 노드에 담긴 다각형들이 일정한 개수가 될 때까지 반복된다.

그림 9에서 보여주는 것처럼 Octree 공간 분할 기법을 인체 모델에 적용 시켰을 때 물체가 바로 주변의 물체들과 충돌하게 되므로 물체가 차지하는 육면체들 안의 물체들과의 비교만으로 나누어진 서브트리 각각의 공간 안에서 일어나는 충돌을 검사하고 충돌 부분이 없으면 그 입방체는 제외시키고 충돌 부분이 있는 입방체만을 처리한다. 이 방법은 공간상에서 여러 물체들이 운동하는 경우에 효율적이며, 공간이 아닌 물체 자체를 분할하여 복잡한 물체들 사이의 충돌 검사 방법으로 사용되기도 한다. 이 방법을 사용하면 바운딩 볼륨 방법[15,16]을 사용하는 것에 비해 시뮬레이션 속도는 느리지만, 각 공간 안에 삼각형의 점과 면이 충돌하는 경우만 체크하여 저장하기 때문에 계산량을 줄일 수 있고 보다 자연스러운 시뮬레이션 결과를 볼 수 있다.

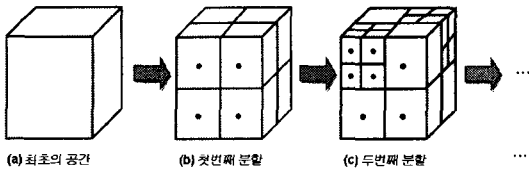


그림 8. Octree 공간 분할 방법

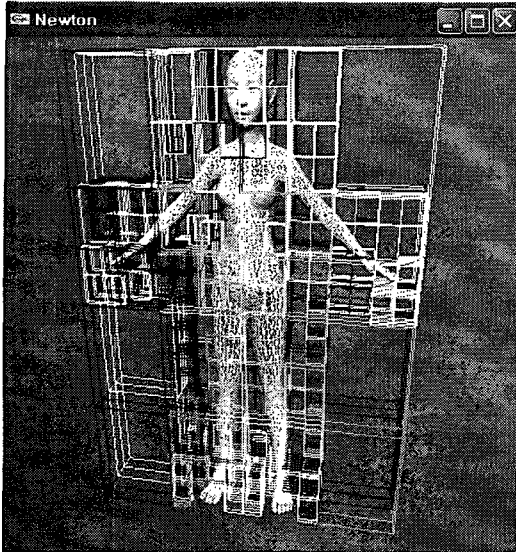


그림 9. Octree 공간 분할 방법을 이용한 인체 모델의 분할

Provot[17]은 충돌하기 위한 필요조건을 설정함으로써 문제를 간단하게 풀 수 있는 방법을 개발하였다. 필요조건은 점과 삼각형이 충돌할 때 삼각형을 포함하는 평면과 점이 같은 평면상에 위치한다는 것이다. 따라서 그림 10과 같이 시간  $t$ 에 따라 이동하는 점  $P(t)$ 와 삼각형(ABC)이 같은 평면에 있게 되는 시간을 계산함으로써 점과 삼각형의 충돌 검사를 수행할 수 있다.

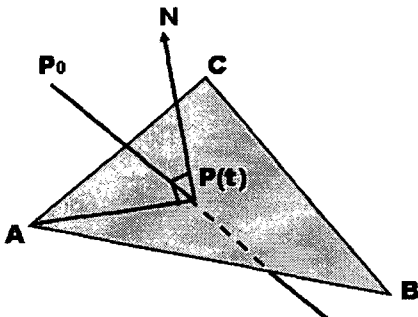


그림 10. 충돌 검사

시간에 따른 정점의 위치는 다음과 식 (1)과 같다.

$$P(t) = P_0 + t \times V, \quad t \in [0, \Delta t] \quad (1)$$

여기서  $P_0$ 은 시작 위치이고,  $V$ 는 속도벡터,  $\Delta t$ 는 시뮬레이션 시간 간격을 나타낸다.

점과 삼각형이 같은 평면에 있게 되면 삼각형의 꼭지점  $A$ 와 점  $P(t)$ 가 이루는 벡터  $AP(t)$ 와 평면의 법선 벡터  $N$ 의 사이 각도는 90도이고 두 벡터의 내적 값은 법선 벡터가 수직이므로 0이다. 이를 식으로 나타내면 식 (2)와 같다.

$$AP(t) \cdot N = 0 \quad (2)$$

$$V = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}, N = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}, P_0 = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix}, A = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} \text{라 가정하고,}$$

식 (2)을 풀면  $t$ 는 다음 식 (3)과 같이 계산할 수 있다.

$$t = \frac{(a_x n_x + a_y n_y + a_z n_z) - (o_x n_x + o_y n_y + o_z n_z)}{v_x n_x + v_y n_y + v_z n_z} \quad (3)$$

이와 같이 구한 충돌 시간은 점과 평면과의 충돌을 계산한 것이므로, 충돌이 삼각형 내부에서 발생할 수도 있고, 삼각형 외부에서 발생할 수도 있다. 따라서 충돌 시간을 계산한 다음에는 충돌 위치를 검사한다. 충돌이 삼각형 내부에서 발생하면 식 (4)의 조건을 만족하는  $u, v$  값이 존재한다.

$$t \in [0, \Delta t], \exists u, v \in [0, 1], u + v \leq 1, \quad (4)$$

$$AP(t) = uAB + vAC$$

식 (9)와 같이 구한 충돌 시간을 식 (4)에 대입하고 연립방정식을 풀면  $u, v$  값에 대한 계산식은 식 (5)과 같다.

$$u = \frac{(p_x - a_x)(c_y - a_y) - (p_y - a_y)(c_x - a_x)}{(b_x - a_x)(c_y - a_y) - (b_y - a_y)(c_x - a_x)} \quad (5)$$

$$v = \frac{(p_x - a_x)(b_y - a_y) - (p_y - a_y)(b_x - a_x)}{(c_x - a_x)(b_y - a_y) - (c_y - a_y)(b_x - a_x)}$$

여기에서  $P(t), A, B, C$ 의 위치는 다음과 같이 가정하였다.

$$P(t) = \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix}, \quad A = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}, \quad B = \begin{bmatrix} b_x \\ b_y \\ b_z \end{bmatrix}, \quad C = \begin{bmatrix} c_x \\ c_y \\ c_z \end{bmatrix}$$

$u, v$  값을 계산한 다음에는 식 (4)의 조건이 만족되는지 검사하여 충돌 위치가 삼각형 내부인지 외부인지를 판별한다.

4.2 충돌 반응 처리

반응 처리는 충돌 검사 후에 충돌된 물체들을 적절히 위치시키는 과정이다. 이 과정을 통해 옷감이 인체 안으로 들어가거나 인체를 뚫고 지나가는 침투 현상을 막아야 한다. 또한 반응 처리 후에 옷감의 전체적인 균일을 유지해야 한다. 이 두 가지가 충돌 반응 처리의 목표가 된다. 정확한 반응 처리를 위해서는 정확한 충돌 시간과 위치를 알아내고 충격량과 마찰력을 생성하여 다음 위치를 계산해 주어야 한다. 인체와 옷감 사이의 충돌은 완전 비탄성 충돌로 생각하고 충돌 반응 처리를 계산한다[17,18].

그림 11은 입자가 이동하여 인체를 뚫고 지나가는 상황을 나타낸다. 여기서 입자가 이동하다가 인체의 평면과 충돌하는 순간에는 충격량이 발생한다. 이를 통해 충돌한 평면의 수직 방향에 해당하는 속도가 결정되는데, 이는 운동량 보존의 법칙에 의해 좌우되며 완전 비탄성 충돌인 경우 수직 방향 속도는 없어진다. 본 논문에서는 충돌 평면에 수직 방향으로 더 이상 진행이 불가능하므로 수직 방향의 속도 성분을 제거하고 충돌 평면에 수평 방향 성분의 속도만을 유지하여 다음 위치를 계산한다.

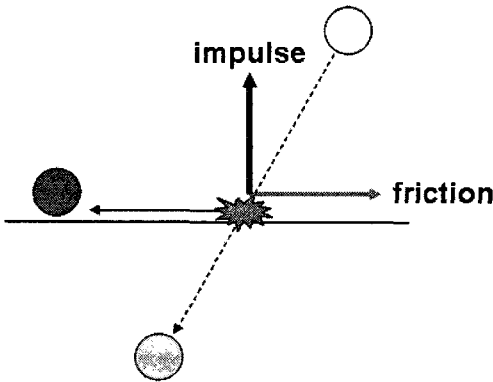


그림 11. 충돌 반응 처리 모델

충돌이 발생한 경우에는 먼저 입자의 위치를 수정한다. 입자의 새로운 위치는  $P'$ 는

$$P' = P + t \times V \tag{6}$$

로 계산하며 여기서  $t$ 는 충돌 시간을 나타내고  $P$ 는 이전의 위치,  $V$ 는 속도 벡터를 나타낸다.

충돌이 발생하지 않은 경우, 입자의 위치는

$$P' = P + \Delta t \times V \tag{7}$$

로 계산하며,  $\Delta t$ 는 시뮬레이션 시간 간격을 나타낸다.

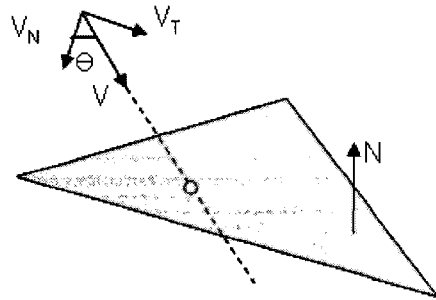


그림 12. 충돌 상황

그림 12와 같이 충돌이 발생한 경우 입자의 속도는 수직 방향 성분은 없어지고 수평 방향 성분만 남게 된다. 수직 방향과 수평 방향의 속도는 식 (8)로 계산한다.

$$\begin{aligned} V_N &= -N \times |V| \times \cos\theta \\ &= -N \times |V| \times \left( \frac{V}{|V|} \cdot (-N) \right) \\ V' &= V_T = V - V_N \end{aligned} \tag{8}$$

$N$ 은 평면의 법선벡터로 정규화 된 값을 나타내고  $V$ 는 속도벡터,  $V'$ 는 새로운 속도벡터를 나타낸다.

정리하면 옷감과 인체사이의 충돌이 검출되면, 옷의 입자의 다음 시간 간격에서의 속도는 충돌한 인체 표면의 법선 방향의 속도와 부합되도록 충돌 면에 수직 방향으로는 더 이상 진행할 수 없기 때문에 수직 방향의 속도 성분을 제거하고 수평 방향 성분의 속도만을 유지한다.

### 5. 의복 시뮬레이션의 3차원 렌더링

본 절에서는 2차원 패턴을 인체 모델에 입히는 과정에 대해 설명한다. 본 논문의 시스템에서는 먼저 그림 13과 같은 3차원 인체 모델을 읽어 들인다. 이 인체 모델은 3차원 MAX에서 제작하여 ASC 형태로 저장된 것이다. 인체 모델을 읽어 들인 후에는 옷감 패턴을 읽어 들인다.

그림 14와 같이 인체 모델 주위에 옷감 패턴을 적절한 곳에 위치하도록 배치시킨 다음, 미리 설정된 재봉 선을 따라 힘을 부여하면서 시뮬레이션을 수행하여 옷감 패턴을 바느질 하는 과정을 보여준다. 그림 15는 그림 14에서 보여주는 두 장의 2차원 패턴을 주어진 패턴 라인에 따라 옷감 조각이 함께 꿰매어지는 시뮬레이션 결과이다.

본 논문에서는 2차원 패턴 디자인 모듈은 Intel(R) Core(TM)2 CPU 6400 2.13GHz, 2GB RAM을 가진 PC에서 C#과 SVG(Scalable Vector Graphics)[18] 프로그램을 이용하여 구현하였으며, 3차원 시뮬레이션은 C++와 OpenGL을 이용하여 구현하였다. 3차원 시뮬레이션 시스템에서 이용한 의상은 짧은 치마, 긴 치마, T-Shirt, 원피스 등을 제작하여 사용하였고, 인체 모델을 각 공간 분할 방법을 사용했을 경우와 사용하지 않을 경우로 나누어 실험하였다.

표 1과 같이 같은 조건상에서 시뮬레이션 했을 때, 본 논문에서 제시한 Octree 공간 분할 방법을 이용한 3차원 의복 시뮬레이션 시스템의 결과가 공간 분할 방법을 사용하지 않은 경우에 비해 시뮬레이션 시간이 빨라지는 것을 볼 수 있다. 또한 기존의 바운딩

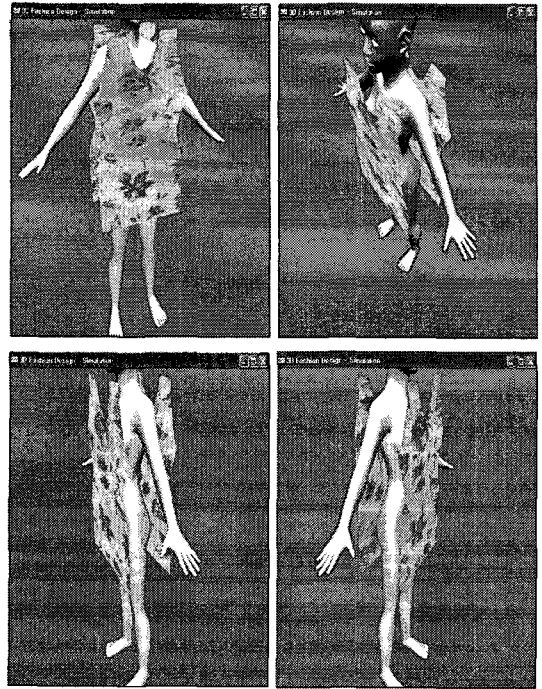
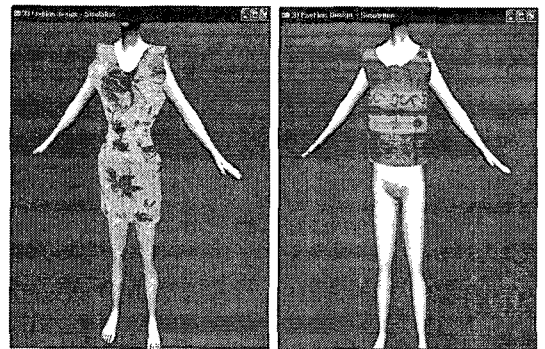


그림 14. 2차원 패턴의 위치 조정



(a) 원피스

(b) T-Shirt

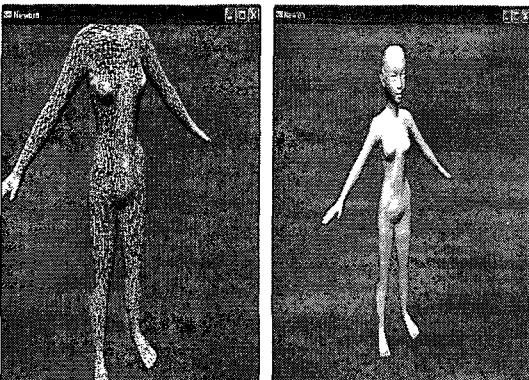
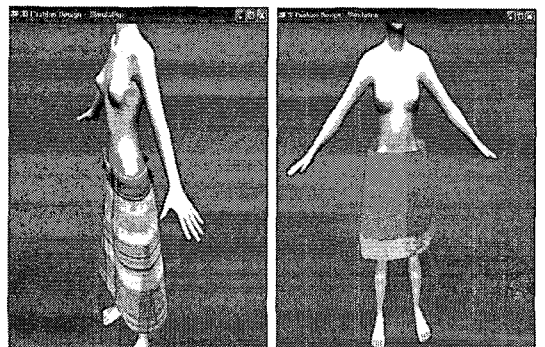


그림 13. 인체 3차원 Tri-mesh 모델 & 3차원 인체 모델



(c) 긴 치마

(d) 짧은 치마

그림 15. 3차원 의복 시뮬레이션 결과



표 1. 시뮬레이션 시간 결과 (단위 : 초)

종 류	짧은 치마	긴 치마	T-shirt	원피스
바운딩 볼륨 방법을 사용한 경우	4.2	6.0	6.3	7.2
Octree 공간 분할 방법을 사용한 경우 (3차원 의복 시뮬레이션 시스템)	7.8	9.6	10.2	11.8
Octree 공간 분할 방법을 사용하지 않은 경우	18.5	22.7	25.2	35.0

볼륨 방법[14]에서는 움직이지 않는 인체 모델에 짧은 치마를 입히는 경우 본 연구보다 빠른 시뮬레이션 결과를 보여주었지만, 이는 선분과 선분 사이의 충돌과 다각형 모델의 점과 옷감의 삼각형 사이의 충돌을 고려하지 않았기 때문이다. 반면, 본 논문에서 이용한 Octree 공간 분할 방법의 경우 기존의 바운딩 볼륨 방법[14]이 가지고 있는 옷감이 인체 모델에 침투하거나, 부분적으로 늘어나는 현상 그리고 인체 모델에 다소 떨어져 시뮬레이션 되는 단점을 보완하였다. 또한 본 논문에서 제시한 Octree 공간 분할 방법을 사용했을 경우와 사용하지 않았을 경우를 비교했을 때 Octree 공간 분할 방법을 사용했을 경우 계산량을 현저히 줄여줌으로 인해 속도의 향상과 자연스러운 시뮬레이션 결과를 볼 수 있다. 하지만 본 논문은 바운딩 볼륨 방법[15,16]을 사용했을 때보다는 속도가 다소 떨어지는 단점을 가지고 있다.

## 6. 결 론

본 논문에서는 의복의 2차원 패턴을 직접 디자인 하여 사용할 수 있는 디자인 모듈을 이용하여 제작하고, 이를 이용하여 실시간에 상호작용이 가능한 3차원 의복 시뮬레이션 방법을 제안하였다. 지금까지의 관련 연구와 실험 내용을 바탕으로 실시간성에 가장 적합한 Provot의 스프링-질량 모델을 옷감 모델로 선택하였고, 자연적인 힘에 의해 변화되는 옷감의 움직임을 계산하는 방법으로 오일러 적분법과 길이 보정 방법을 선택하였다. 이 과정에서 인체와 옷감의 충돌 처리는 공간을 8조각으로 계속 나누어 각 노드에 담긴 다각형들이 일정한 개수가 될 때까지 반복되는 Octree 공간 분할 방법을 이용하여 인체 모델을 근사하고 이를 충돌 계산에 이용하였다.

본 논문의 3차원 의복 시뮬레이션 시스템을 이용하면, 실제 의복을 제작하지 않고서도 체계적이고, 효율적인 의복 DB를 구축할 수 있으며, 의복의 디자인, 제작과정에서도 생산성을 향상시킬 수 있을 것이며, 의상 디자이너가 의복을 실제로 제작해 보지 않고서도 가상으로 디자인을 제작하여 그 결과를 시각적으로 확인할 수 있게 함으로써 디자인 시간과 비용을 줄이는데 활용 될 수 있을 것이다.

향후 연구 과제로는 3차원 의복 시뮬레이션 과정에서 사용하기 쉽고 조금 더 자연스러운 매핑처리를 위한 알고리즘 연구와 시뮬레이션 속도를 향상 시켜, 보다 빠르고 자연스러운 시뮬레이션 결과를 얻고자 한다.

## 참 고 문 헌

- [1] N. Ahuja and C. Nash, "Octree representation of moving objects," *Computer Vision Graphics and Image Processing*, Vol. 26, pp. 27-216, 1984.
- [2] H.H. Chen and T.S. Huang, "A survey of construction of Octree," *Computer Vision Graphics and Image Processing*, Vol. 43, pp. 409-431, 1988.
- [3] "Tex-Design System," Koppermann, 2003.
- [4] 강성휘, *다양한 의복의 2D 패턴구성을 통한 3D 모델의 자동생성*, 명지대학교, 일반대학원, 2001.
- [5] 최우혁 외, "스커트 3D 모델의 기하학적 생성방법," *한국의류학회*, pp. 770-777, 2003.
- [6] X. Provot, "Deformation Constraints in a Mass-Spring Model to Describe Rigid Cloth Behavior," *Proc. Graphics Interface*, pp. 147-154, 1995.
- [7] Fashion Studio, <http://www.dynagraphicsinc.com>
- [8] T.I. Vassilev, "Dressing Virtual People," *SCI*, pp. 23-26, 2000.
- [9] L. Chittaro and D. Corvaglia, "3D Virtual Clothing: from Garment Design to Web3D Visualization and Simulation," *SIGGRAPH*, pp. 73-84, 2003.

[10] Y.M Kang and H.G Cho, "Real-time Animation of Complex Virtual Cloth with Physical Plausibility and Numerical Stability," *Journal of PRESENCE - Teleoperators & Virtual Environments* 13(6), pp. 668-680, 2004.

[11] O. Cismasu, "The Jordan Curve Theorem for Polygons," *Computational Geometry*, pp. 308-507, 1997.

[12] J. McCleary, *Geometry from a differentiable viewpoint*, Cambridge Univ. press. Cambridge, 1994.

[13] D. Moore and J. Warren, "Compact Isocontours from Sampled Data," *Graphics GemIII, Academic Press*, pp. 23-28, 1992.

[14] 오승우, *Interactive Garment Animation*, 한국과학기술원, 일반대학원, 2001.

[15] J.T. Klosowski and M. Held & J.S.B. Mitchell, "Efficient Collision Detection Using Bounding Volume Hierarchies of K-dops," *IEEE Transactions on Visualization and Computer Graphics*, Volume 4, pp. 1-27, 1998.

[16] P. Volino and N. Magnenat-Thalmannhen, "Efficient self-collision detection on smoothly discretized surface animation using geometrical shape regularity," *EuroGraphics*, pp 155-166, 1994.

[17] X. Provot, "Collision and self-collision handling in cloth model dedicated to design garments," *Graph Interf'97*, pp. 147-154, 1997.

[18] D. Baraff and A. Witkin, "Large Steps in Cloth Simulation," *Computer Graphics(Proc. SIGGRAPH)*, pp. 43-54, 1998.

[19] J.D. Eisenberg, *SVG Essentials*, O'Reily, Gravenstein Highway North Sebastopol, 2002.



김 주 리

1998년 2월 원광대학교 컴퓨터 공학과 학사  
 2002년 2월 원광대학교 교육대학원 전자계산교육전공 석사  
 2004년~현재 원광대학교 컴퓨터 공학과 박사과정

관심분야 : 컴퓨터 그래픽스, XML, 컴퓨터 애니메이션



정 석 태

1989년 2월 전남대학교 전산학과 학사  
 1996년 3월 쓰꾸바대학 멀티미디어전공 석사  
 2000년 7월 쓰꾸바대학 멀티미디어전공 박사  
 2001년~현재 원광대학교 전기전

자 및 정보공학부 교수

관심분야 : 공간 파서 생성기, 비주얼 시스템, 오감 정보 통신



정 성 태

1987년 2월 서울대학교 전자계산기공학과 학사  
 1989년 2월 서울대학교 전자계산기공학과 석사  
 1994년 8월 서울대학교 컴퓨터 공학과 박사  
 1999년~1999년 미국 Univ. of

Utah 과학재단지원 해외 Post-Doc.

1995년~현재 원광대학교 전기전자 및 정보공학부 교수  
 관심분야 : 영상 인식, 영상 기반 렌더링, 컴퓨터 그래픽스