

## 2.5D Mapping 모듈과 3D 의복 시뮬레이션 시스템

김 주 리<sup>\*</sup> · 김 영 운<sup>\*</sup> · 정 석 태<sup>\*\*</sup> · 정 성 태<sup>\*\*</sup>

### 요 약

본 논문은 패션 디자인 분야에서 완성된 의상의 모델 사진을 활용해 다양한 원단(직물)을 직접 Draping함으로써 새로운 디자인을 창출할 수 있고 직접 샘플이나 시제품을 제작하지 않고도 시뮬레이션만으로 의상 작품을 확인 할 수 있다. 또한 모델과 원단 이미지에 대한 데이터베이스를 구축하여 실시간으로 Mapping 결과를 확인할 수 있는 시스템을 구현하였다. 그리고 여기에서 추출되는 시제품을 3D 모델에 입혀 시뮬레이션 할 수 있도록 하기 위한 과정으로 우선 여러 옷감 조각들을 이용하여 가상의 3D 인체 모델에 옷을 입히기 위한 의복 시뮬레이션 시스템을 제안한다. 제안된 시스템은 3D 인체 모델 파일과 2D 재단 패턴 파일을 읽어 들인 다음에 mass-spring model에 기반한 물리적 시뮬레이션에 의해 의복을 착용한 3D 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 줄이는 방법을 이용하여 사실적인 영상을 생성할 수 있었고, 수초 이내에 가상 인체 모델에 의복을 입힐 수 있었다.

키워드 : 2.5D 매핑 모듈, 3D 의복 시뮬레이션, 충돌 처리, 질량-스프링 모델

## 2.5D Mapping Module and 3D Cloth Simulation System

Ju-Ri Kim<sup>\*</sup> · Young-Un Kim<sup>\*</sup> · Suck-Tae Joung<sup>\*\*</sup> · Sung-Tae Jung<sup>\*\*</sup>

### ABSTRACT

This paper utilizing model picture of finished clothes in fashion design field various material (textile fabrics) doing Draping directly can invent new design, and do not produce direction sample or poetic theme width and confirm clothes work to simulation. Also, construct database about model and material image and embodied system that can confirm Mapping result by real time. And propose clothes simulation system to dress to 3D human body model of imagination because using several cloth pieces first by process to do so that can do simulation dressing abstracted poetic theme width to 3D model here. Proposed system creates 3D model who put clothes by physical simulation that do fetters to mass spring model after read 3D human body model file and 2D foundation pattern file. System of this treatise examines collision between triangle that compose human body model for realistic simulation and triangle that compose clothes and achieved reaction processing. Because number of triangle to compose human body is very much, this collision examination and reaction processing need much times. To solve this problem, treatise that see could create realistic picture by method to diminish collision public prosecutor and reaction processing number, and could dress clothes to imagination human body model within water plant taking advantage of Octree space sharing techniques.

Key Words : 2.5D Mapping Module, 3D Cloth Simulation, Collision Detection, Mass-spring Model

### 1. 서 론

정보화 사회인 현실에서 패션 디자인은 제품의 트렌드 (Trend) 및 주요 마켓 대상을 결정하는 매우 중요한 과정으로써 유행의 변화와 소재의 다양성으로 인해 많은 어려움을 겪고 있다. 최적의 스타일 디자인 기획은 최종 생산제품의

판로를 결정하는 중요한 변수이기도 하다. 따라서 의상 디자인작업에 필요한 중요한 요소들을 체계적으로 관리하고 도식화 할 수 있는 디자인 시스템 기술을 필요로 하고 있다. 하지만, 현재의 의상 디자인 시스템 기술은 2D 영상에 의존하고 있으며 주어진 사진 객체를 카달로그 형식으로 보여주는 효과 밖에 가지고 있지 않다. 그렇기 때문에 패션디자인 분야에서는 직접 의상의 샘플을 제작하는데 많은 시간과 비용을 필요로 하고 있으며 많은 부분을 낭비하고 있는 현실이다[9, 10].

의복 디자인 시스템은 2D, 2.5D, 3D로 나누어 질수 있으

\* 본 연구는 산업자원부와 한국산업기술재단에서 시행한 지역전략산업 석·박사 연구인력 양성사업의 지원에 의해서 수행하였음.

<sup>†</sup> 준 회원 : 원광대학교 컴퓨터공학과 박사과정

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

논문접수 : 2006년 4월 17일, 심사완료 : 2006년 6월 7일

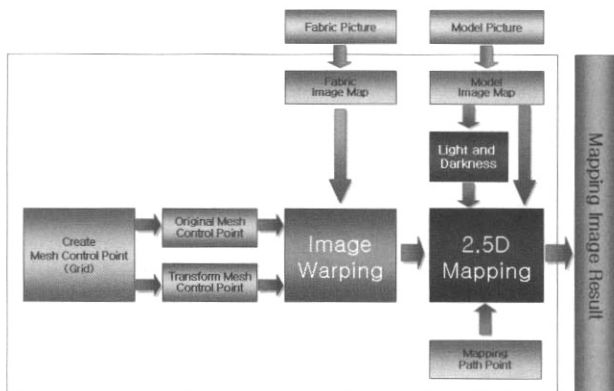
며, 2D 의복 디자인은 평면에 단순한 도형을 사용해 디자인 하며 의복을 쉽게 표현 할 수 있지만 사실감이 많이 떨어진다. 2.5D 의복 디자인은 3D 의복 디자인으로 가기 이전 단계로 원단을 워프(Warp) 기법을 사용해 맵핑함으로써 조금 더 사실적인 표현이 가능 해졌다. 본 시스템에서는 3D 의복 디자인 모듈을 설계하고 구현하였으며, 디자인된 3D 의복을 직접 착의 시물레이션 해 볼 수 있는 모듈까지 구현 하였다. 또한 착의 시물레이션을 통해 의복을 인체 모델에 입혀 봄으로써 눈으로 직접 확인해 볼 수 있으며, 다양한 각도에서 착의된 의복을 확인해 볼 수 있다. 본 논문의 시스템은 의상 샘플에 소요되는 시간과 비용을 줄이고자 의상 디자인에서 완성된 모델사진을 활용해 디자이너가 사진 모델에 외곽선 영역을 생성하고 원하는 원단(소재)을 Mapping 하도록 한다. 이는 원단이 가지고 있는 다양한 질감과 패턴에 따른 원본의 이미지를 그대로 살려 원단 디자인과 최종 제품의 상태를 정확하게 예측할 수 있는 기능을 제공함으로써, 새로운 디자인을 창출할 수 있고 직접 샘플이나 시제품을 제작하지 않고도 시물레이션만으로 의상 작품을 확인 할 수 있다. 또한 모델 사진을 활용한 3차원 의상 시물레이션을 하기 위한 전 과정으로, 3차원 인체 모델이 주어진 상태에서 2D 디자인 패턴을 이용하여 의복이 모델에 입혀지는 과정을 보여주는 시스템을 제안한다. 향후에는 의상 사진을 이용하여 Mapping된 디자인 패턴 결과물을 가지고 3차원 인체 모델에 직접 시물레이션 할 수 있도록 하여 사용자가 원하는 다양한 디자인과 소재를 선택하여 시물레이션 할 수 있도록 하고자 한다.

## 2. 2.5D Mapping 시스템

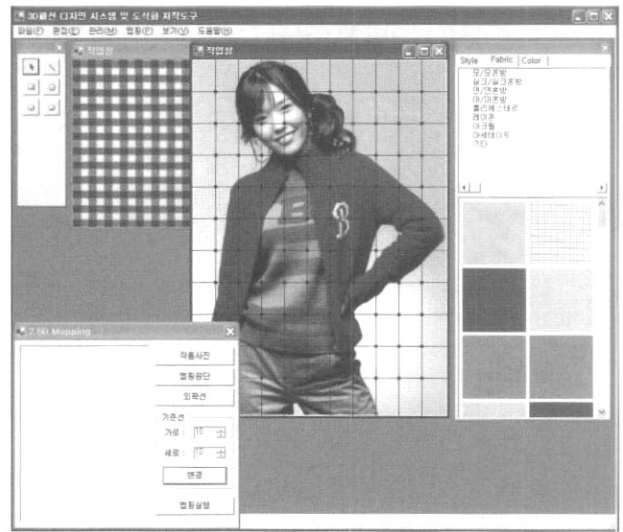
### 2.1 전체 시스템 구성

시스템 구성도는 (그림 1)과 같다. 크게 Image Warping, 2.5D Mapping 2개의 모듈로 구성되어져 있으며, 명암 추출과 적용 모듈, 영역(Path) 추출 모듈, Mesh 제어 점 추출과 변형 모듈 등으로 구성되어져 있다[13].

(그림 2)는 2.5D Mapping 시스템 구현 화면이다. 도식화에 필요한 Toolbar와 Style, Fabric, Color bar 등으로 구성



(그림 1) 2.5D Mapping 시스템 구성도



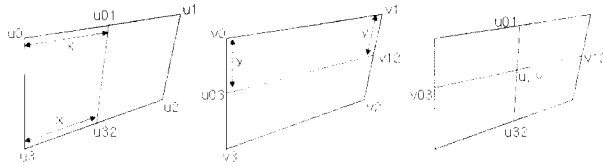
(그림 2) 2.5D Mapping 시스템 구현 화면

되어 있으며, 2.5D Mapping을 제어할 제어상자와 맵핑에 사용될 원단이미지 창으로 이루어져 있다. 또한 모델사진을 표현하고 메쉬의 생성과 변형, 맵핑 영역(Path) 생성, 메쉬 기반 워핑, 2.5D 맵핑 등의 기능을 가진 작업창이 제공된다. 원단이 가지고 있는 속성을 활용해 사실적인 Draping을 구현하기 위해 모델사진의 굴곡, 음영, 주름 등을 분석해 Mapping 시킬 원단에 적용하고 있다. 이를 구현하기 위해 모델 사진에서 음영 추출과 추출된 음영의 정보를 원단에 적용하는 기법을 사용하였으며, 이미지 워핑 기법을 사용해 원단의 주름과 굴곡을 표현하였다. 원단 데이터와 모델 데이터는 래스터(Bitmap) 형식이며, Mesh Point 생성, 맵핑 영역(Path) 생성 등의 데이터는 벡터(Vector)형식으로 이루어진다. 또한 벡터형식은 XML언어를 사용 하여 표현 하였으며, 문서파일의 저장 포맷으로 사용하여 문서 호환성과 확장성을 고려해 설계되었다[11, 12].

### 2.2 메쉬 워프 알고리즘

워핑(Warping)은 특정 규칙에 따라 입력된 이미지를 재추출하는 작업이다. 이 처리는 여러 다른 방향으로 이미지를 확장시킬 수 있으며, 고무줄이 위에 그려진 이미지를 처리하는 것과 비슷하다. 워핑 알고리즘은 제어 점, 제어 선, 그물망, 다각형 등 다양한 방법으로 구현이 가능하다. 본 논문에서는 메쉬(Mesh) 워프 알고리즘을 사용하였으며, 메쉬 워프 알고리즘은 두 단계 알고리즘으로 이루어져 있다. 첫 번째 단계에서 이미지의 매 행(Row)을 따라 처리하며, 두 번째 단계에서 이미지의 매 열(Col)을 따라 처리함으로써 결과를 얻을 수 있다. 입력 데이터로는 입력이미지 및 출력 이미지의 조각(Mesh)들과 제어 점들이 요구된다. 제어 점은 입력 제어 점과 출력 제어 점으로 나누어지며, 보간법에 의해 새로운 제어 점이 만들어진다.

보간법은 간단한 선형 보간법에서부터 복잡한 스플라인 함수까지 다양하다. 본 시스템에서는 양선형 보간법을 사용하였으며, (그림 3)은 양선형 보간법을 이용한 좌표결정을



(그림 3) 역 양선형 변형을 이용한 좌표결정

설명하고 있다.

양선형 보간법은 수평 또는 수직선을 따라 등간격의 점을 보존하지만 대각선을 보존 하지 못하며, 출력 영상 x, y를 통해 입력 영상의 u, v 좌표 계산을 단계적으로 계산을 한다. x는 행을 따라가며 나타나는 유리수 값들이며, 이 값은 새로운 점들을 산출하기 위해 출력 사변형의 윗부분과 바닥의 수평선을 따라가며  $u_{01}$ 과  $u_{32}$ 가 보간 된다. y는 u, v로부터 변형된 화소의 마지막 결과를 산출하기 위한 두 개의 점들 사이로 보간 되기 위해 사용된다. 기본적인 선형 보간법을 사용하기 위한 u와 v에 대해서 다음 식(1)과 같이 계산해야 한다.

$$\begin{aligned} u_{01} &= u_0 + x(u_1 - u_0) & v_{03} &= v_0 + y(v_3 - v_0) \\ u_{32} &= u_3 + x(u_2 - u_3) & v_{12} &= v_1 + y(v_2 - v_1) \\ u &= u_{01} + y(u_{32} - u_{01}) & v &= v_{03} + y(v_{12} - v_{03}) \end{aligned} \quad \text{식(1)}$$

### 2.3 명암 추출과 적용 모듈

입력받은 모델사진의 각 화소(Pixel)값의 명암 데이터를 추출한 후 Mapping 시킬 원단 이미지의 각 화소(Pixel)에 적용시키는 모듈이다. 모델 사진에서 추출한 화소 값을 RGB (Red/Green/Blue) 형태에서 HSI(Hue/Saturation/Intensity)형태로 변환해 명도(I)값을 추출한다. 추출된 명도(I)값은 원단 이미지의 같은 위치의 화소에 적용시킨다. 이를 처리하기 위해 RGB 모델을 HSI 모델로의 변환과 HSI모델에서 RGB모델의 변환 알고리즘이 필요하다[14, 15].

다음 식(2)를 이용하여 RGB 모델을 HSI 모델로 변환한다.

$$\begin{aligned} H &= \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right\} \\ S &= 1 - \frac{3}{(R+G+B)} \min(R, G, B) \\ I &= \frac{1}{3}(R+G+B) \end{aligned} \quad \text{식(2)}$$

HSI모델에서 RGB모델로의 변환은 컬러 영역이 어디에 존재 하는가에 따라 의존하며, 다음은 RG 영역( $0^\circ \leq H \leq 120^\circ$ ) 일 때의 식이다.

$$\begin{aligned} b &= \frac{1}{3}(1-S) \\ r &= \frac{1}{3} \left[ 1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right] \\ g &= 1 - (r+b) \end{aligned} \quad \text{식(3)}$$

다음 식(4)는 GB 영역( $120^\circ < H \leq 240^\circ$ )에 대해서다.

$$\begin{aligned} H &= H - 120^\circ \\ g &= \frac{1}{3} \left[ 1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right] \\ r &= \frac{1}{3}(1-S) \\ b &= 1 - (r+g) \end{aligned} \quad \text{식(4)}$$

BR 영역( $240^\circ < H \leq 360^\circ$ )에 대한 식은 다음과 같다.

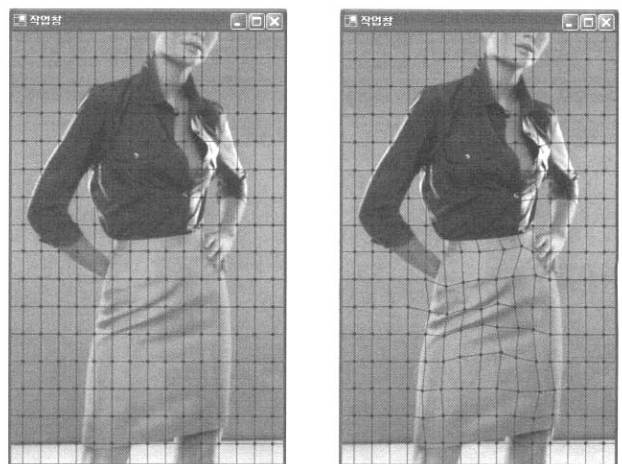
$$\begin{aligned} H &= H - 120^\circ \\ b &= \frac{1}{3} \left[ 1 + \frac{S \cos(H)}{\cos(60^\circ - H)} \right] \\ g &= \frac{1}{3}(1-S) \\ r &= 1 - (g+b) \end{aligned} \quad \text{식(5)}$$

r, g, b 값들은 R, G, B의 값으로 정규화 된다. 다음과 같이 R, G, B 값으로 변환한다.

$$R = 3Ir, \quad G = 3Ig, \quad 100B = 3Ib \quad \text{식(6)}$$

### 2.4 Mesh 생성과 변형 모듈

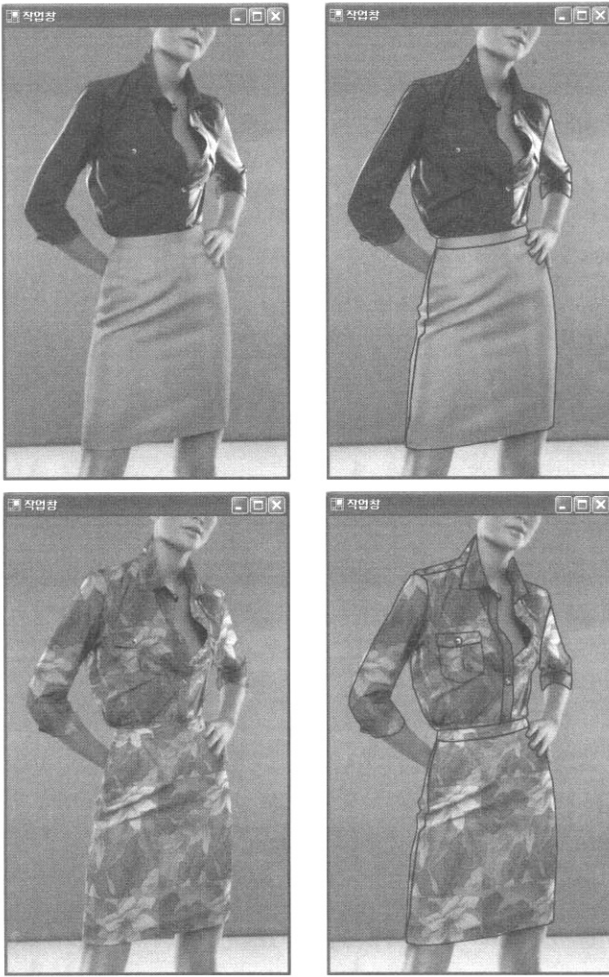
Mesh 생성과 변형 모듈은 워핑 처리에 필요한 제어 점 생성과 제어 점 변형을 위한 모듈로 구성되어 있으며, 가로/세로 기준 개수를 입력해 새로운 제어 점과 안내선을 표시한다. 초기 원 제어 점 데이터와 사용자로부터 변형된 제어 점을 자동 생성해 워핑 처리에 사용한다. (그림 4)는 생성(Source)된 제어 점과 변형(Dest)된 제어 점을 나타내고 있으며 제어 점 사이를 안내선으로 표시하고 있다[16-22].



(그림 4) Mesh 생성과 변형

### 2.5 영역(Path) 추출과 2.5D Mapping 모듈

영역 추출 모듈은 모델 사진으로부터 Mapping될 외곽선 영역을 사용자로부터 생성하는 모듈로 영역은 좌표 점들을 입력 받아 벡터(Vector) 구조로 저장된다. (그림 5)는 사용자로부터 직접 입력 받은 영역을 표시한 화면으로 여러 개



(그림 5) 맵핑 영역 추출

의 영역이 생성되었음을 볼 수 있다.

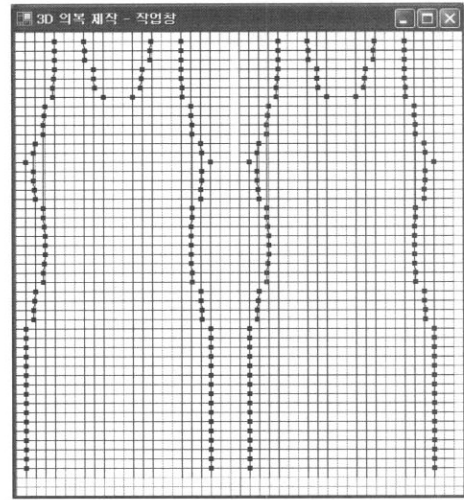
2.5D Mapping 모듈은 원단, 모델, 영역, 명암 데이터를 사용해 Mapping을 수행한다. Mapping 처리과정을 보면 원단 이미지와 Mesh 생성 및 변형 모듈에 의해 생성된 제어 포인트를 이용해 워핑 알고리즘을 수행한다. 모델 사진에서 추출된 명암데이터와 워핑 알고리즘이 적용된 원단을 영역(Path) 추출 모듈에 의해 생성된 외곽선 영역에 Mapping을 수행함으로써 그림 5와 같은 최종 결과물을 얻을 수 있다 [11, 12, 16, 17].

### 3. 3D 의복 디자인 시스템

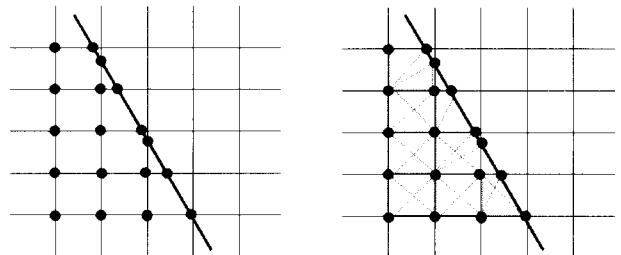
#### 3.1 2D 피스 디자인

2D 피스 조각을 제작하는 과정은 옷감 조각의 외곽선을 그리고 재봉 선을 긋는 과정으로 구성된다.

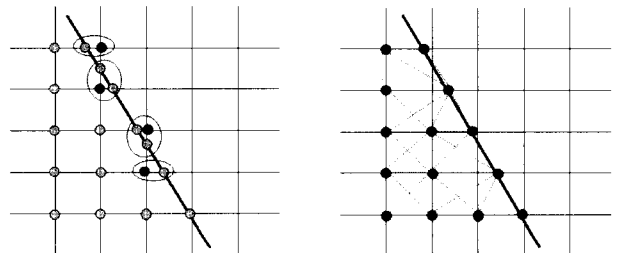
본 논문에서는 (그림 6)처럼 2차원 격자 위에서 연속적으로 격자점을 선택해 가면서 옷감 조각의 외곽선을 그릴 수 있는 2D 디자인 제작 시스템에서 직접 2D 피스를 제작하는 방법과 사진 상에서 외곽선을 그려 이를 2D 디자인 제작 시스템으로 가져오는 방법을 사용하고 있다. 외곽선이 모두



(그림 6) 2D 디자인 제작 시스템에서 제작한 피스 디자인



(그림 7) 교차점을 모두 질량 점으로 할 경우의 스프링 구조



(그림 8) 압축 방법을 사용한 경우의 스프링 구조

그러지 않으면 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 걱정해 주는 조단 곡선 이론(Jordan Curve Theorem)을 사용한다. 실제로 질량 점은 외곽선 내부뿐만 아니라 외곽선과 격자선의 교차점도 포함한다. 이를 위해 (그림 7)의 왼쪽처럼 외곽선과 격자선의 교차점을 모두 질량 점으로 할 경우 (그림 7)의 오른쪽처럼 복잡한 스프링 구조를 가지게 된다. 이런 구조는 전체 시스템의 성능을 저하시키게 되므로 좀 더 간단한 구조가 필요하다. 이를 위해 외곽선과의 교차로 생기는 질량 점들을 가장 가까이에 있는 격자점과 연관하여 압축하는 방법을 사용하고 있다. (그림 8)의 동그라미는 연관 관계를 나타내고, 진한 점이 격자점이고 흐린 점이 교차로 생기는 질량 점들이다. 각 격자점은 연관된 교차점 중 가장 가까이에 있는

점으로 대체되고, 이 점이 연관관계에 있는 점들을 대표하게 된다. 이렇게 해서 격자 점 하나가 연관된 교차점을 모두 대표하고 스프링 계산에 이용된다[1, 2, 5].

3.2 3D 의복 모델 데이터 읽기/저장

3D 의복 데이터를 사용하기 위해 본 시스템에서는 파일 (\*.cloth) 시스템을 사용하고 있다. (그림 9)는 3D 원피스 모델이 저장된 파일을 메모장에서 읽은 그림이다. 데이터 구조를 보면 크게 의복 정보와 바느질 정보로 나누어 질수 있다. 의복 정보에는 의복의 수, 작업 영역 크기, 면수(앞/뒤), 제어 점 데이터, 제어 점 보정 데이터 등으로 이루어져 있으며, 바느질 정보에는 바느질할 제어 점 과 제어 점들을 정의하고 있다.

3D 의복 데이터 구성은 <표 1>과 같다. 의복 특성을 고려해 데이터 저장 구조를 설계 하였으며, 하나의 데이터 파일에 여러 개의 의복 정보를 담을 수 있다. 또한 하나의 의복 정보에 여러 개의 피스정보를 함께 저장 할 수 있도록 설계하였다.



(그림 9) 3D 의복의 수치 데이터

<표 1> 3D 의복 데이터 구성

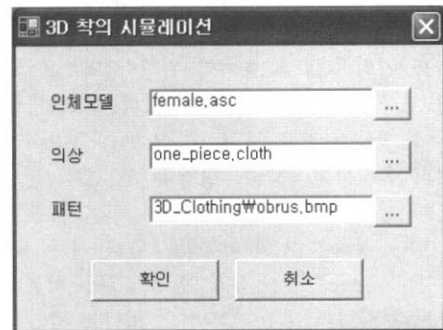
구 분	실 명
의복 수	3D 의복 개수
작업장 크기	전체 의복 작업장 크기
피스 개수	의복 피스 개수
첫 번째 피스크기	첫 번째 피스정보 : 피스명칭, 가로크기, 세로크기
첫 번째 좌표점 개수	좌표점 개수
첫 번째 좌표점 정보	y축, x축 첫 번째 값, x축 두 번째 값
첫 번째 좌표점 보정 개수	좌표점 보정 개수
첫 번째 좌표점 보정 정보	y축, x축, x축 보정 값, y축 보정 값, z축 보정 값
두 번째 피스크기	두 번째 피스정보 : 피스명칭, 가로크기, 세로크기
두 번째 좌표점 개수	좌표점 개수
두 번째 좌표점 정보	y축, x축 첫 번째 값, x축 두 번째 값
두 번째 좌표점 보정 개수	좌표점 보정 개수
두 번째 좌표점 보정 정보	y축, x축, x축 보정 값, y축 보정 값, z축 보정 값
...	...
바느질 개수	바느질 정보 개수, 첫 번째 피스, 두 번째 피스
바느질 정보	첫 번째 좌표점 y축, x축과 두 번째 피스의 좌표점 y축, x축

3.3 재봉과 제약 점 설정

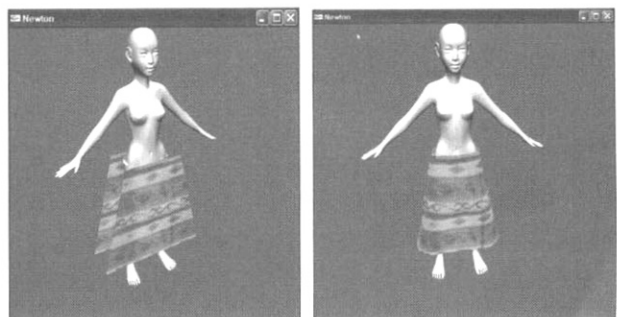
위에서 설명한 바와 같이 본 논문에서는 2.5D Mapping 시스템을 설계하여 영역 추출과 Mesh를 생성하는 방법을 적용시켜 기존의 사진에서 의상 패턴만 가져올 수 있는 방법을 적용시켜 누구나 손쉽게 Mapping된 외곽선 영역을 생성하고, 이를 2D 피스 디자인으로 바꿔준 후, 옷감의 움직임에 필요한 mass-spring 구조[6]를 생성을 하여 3D 모델에 적용하였다.

좀더 자세히 살펴보면, 먼저 3차원 시뮬레이션이 이루어지기 위해서는 사진에서 의상 패턴의 외곽선 영역을 생성하고 생성된 데이터는 다시 인체 모델에 Mapping 시킬 수 있도록 먼저 2차원 조각들로 생성되어야 한다. 이는 여러 개의 2D 피스들을 재봉하여 인체 모델에 입히기 위한 과정이다. 이렇게 디자인 된 2D 피스는 (그림 9)와 같이 수치로 바꾸어 cloth 파일로 저장해 주고 (그림 10)과 같이 인체 모델과 의상, 그리고 패턴 디자인을 조합하여 3D 시뮬레이션이 가능하도록 만들어준다. 3D 시뮬레이션이 가능하기 위해서는 2D 피스에 재봉 선과 제약 점이 필요하다. 제약 점은 인체 모델의 특정 부분에 조정된 의상의 입자 점을 뜻한다. 즉 제약 점은 인체의 특정 부분의 지역 좌표계에 고정된다. 따라서 2차원 조각들이 인체의 특정 부분에 고정되게 한다. 기존 연구[4]에서는 제약 점을 사용하면 의상이 인체를 뚫고 지나가는 현상을 막을 수 있지만 사실성을 떨어뜨리는 단점이 있었다.

본 연구에서는 인체를 뚫고 지나가는 현상을 막을 뿐만 아니라 사실성을 더해주고 있다. (그림 11)은 긴 치마를 시뮬레이션하기 위해 치마의 허리 부분에 설정된 제약 점들을



(그림 10) 3D 의복 착의 시뮬레이션



(그림 11) 긴 치마 시뮬레이션을 위한 제약 점 설정

보여준다.

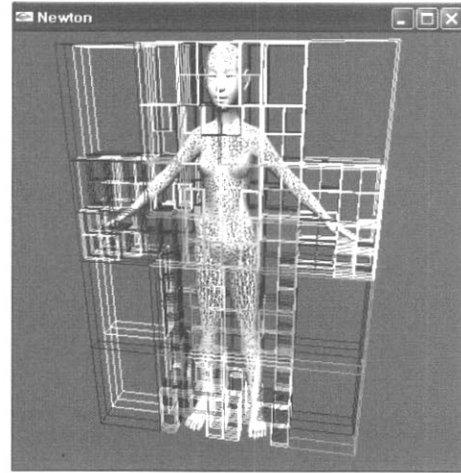
재봉은 (그림 11)처럼 옷감 조각들을 적절한 위치로 옮기고, 시뮬레이션 과정을 통해 재봉 선으로 연결된 옷감 조각들을 붙이는 과정이다. 재봉에서의 시뮬레이션 과정은 두 가지로 나뉜다. 첫째는 제약 점과 중력을 무시하고 길이 보정 없이 integration만을 사용하여 많은 수의 움직임 계산을 거치는 비부하(unload) 과정이고, 둘째는 제약 점과 중력을 고려하고 길이 보정 과정을 추가하여 적은 수의 움직임 계산 과정을 거치는 부하(load) 과정이다. 비부하 과정에서는 제약 점과 중력을 무시하고, 많은 움직임 계산 과정을 통해 옷감이 전체적인 균일 상태로 가도록 한다. 부하 과정에서는 인체 모델에 의상이 제대로 입혀지도록 제약 점과 중력을 무시하고, 많은 움직임 계산 과정을 통해 옷감이 전체적인 균일 상태로 가도록 한다. 부하 과정에서는 인체 모델에 의상이 제대로 입혀지도록 하기 위해서는 제약 점들의 위치를 적절히 조정해야 한다. 부하 과정은 애니메이션에서의 시뮬레이션 방법과 같다. 이 두 과정을 번갈아 적용하여 적절하게 옷감 조각들을 재봉해야 한다. 재봉이 끝나면 재봉선 사이의 스프링을 제거하고, 옷감 전체를 하나의 조각으로 합쳐 의상 제작을 완료할 수 있다[4].

#### 4. 충돌 처리

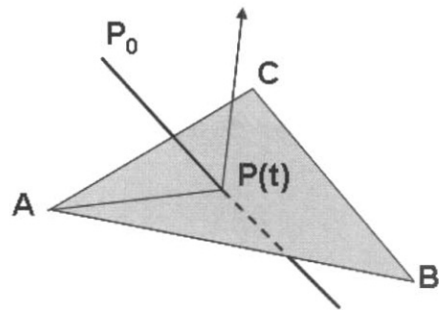
##### 4.1 충돌 검사

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[4]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 있다. 본 논문에서는 인체를 구성한 다각형과 옷의 입자 사이의 충돌 검사를 함으로써 보다 정확한 시뮬레이션을 위해 인체 모델을 (그림 12)와 같이 Octree 공간 분할 방법[6, 7]을 이용하여 나누어진 서브트리 각각의 공간 안에서 일어나는 충돌을 검사하고 처리한다. 이 방법을 사용하면 바운딩 볼륨 방법[23]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 있었다.

Provot[6]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 13)과 같이 시간  $t$ 에 따라 이동하는 점  $P(t)$ 와 삼각형(ABC)이 같은 평면이 있게 되는 시간을 계산함으로써 점과 삼각형의 충돌 검사를 수행할 수 있다.



(그림 12) 계층적 공간 분할 방법을 이용한 인체 모델의 분할



(그림 13) 충돌 검사

시간에 따른 점의 위치는 다음과 식(7)과 같다.

$$P(t) = P_0 + t \times V, \quad t \in [0, \Delta t] \quad \text{식(7)}$$

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

점과 삼각형이 같은 평면에 있게 되면 삼각형의 꼭지점 A와 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(8)과 같다.

$$AP(t) \cdot N = 0 \quad \text{식(8)}$$

$V = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix}$ ,  $N = \begin{bmatrix} n_x \\ n_y \\ n_z \end{bmatrix}$ ,  $P_0 = \begin{bmatrix} o_x \\ o_y \\ o_z \end{bmatrix}$ ,  $A = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$ 라 가정하고, 식(8)을 풀면  $t$ 는 다음 식(9)와 같이 계산할 수 있다.

$$t = \frac{(a_x n_x + a_y n_y + a_z n_z) - (o_x n_x + o_y n_y + o_z n_z)}{v_x n_x + v_y n_y + v_z n_z} \quad \text{식(9)}$$

이와 같이 구한 충돌 시간은 점과 평면과의 충돌을 계산한 것이므로, 충돌이 삼각형 내부에서 발생할 수도 있고, 삼