

# 3D 스캔 데이터를 이용한 얼굴 애니메이션 시스템

구 본 관<sup>†</sup> · 정 철 희<sup>\*\*</sup> · 이 재 윤<sup>\*\*\*</sup> · 조 선 영<sup>\*\*\*\*</sup> · 이 명 원<sup>\*\*\*\*\*</sup>

## 요 약

본 논문에서는 3차원 얼굴 스캔 데이터와 사진 이미지를 이용하여 고품질의 3차원 얼굴 모델과 모핑 애니메이션을 생성하는 시스템 개발에 대해 기술한다. 본 시스템은 얼굴 특징점 입력 도구, 얼굴 텍스처매핑 인터페이스, 3차원 얼굴 모핑 인터페이스로 구성되어 있다. 얼굴 특징점 입력 도구는 3차원 텍스처매핑과 모핑 애니메이션을 위한 보조 도구로서 얼굴의 특징점을 입력하여 텍스처매핑과 임의의 두 얼굴간의 모핑 영역을 정할 때 사용된다. 텍스처매핑은 3D 스캐너로부터 획득한 얼굴의 기하 데이터에 세 방향의 사진 이미지를 이용하여 매핑한다. 3D 얼굴 모핑은 얼굴 특징점 입력 도구로부터 얻은 특징점을 중심으로 얼굴 영역을 분류하여 임의의 두 얼굴 간의 영역간 매핑을 실현한다. 본 시스템은 사용자가 별도의 프로그래밍 작업 없이 대화형 인터페이스에서 3D 스캐너에서 획득한 얼굴 메쉬 데이터를 이용하여 사진 이미지로 텍스처 매핑을 실행하여 사실적인 3D 얼굴 모델을 얻을 수 있고, 임의의 서로 다른 얼굴 모델들간의 모핑 애니메이션을 쉽게 실현할 수가 있다.

키워드 : 3D 얼굴 모델링, 3D 얼굴 애니메이션, 3D 얼굴 텍스처매핑, 3D 얼굴 모핑, 얼굴 정의 인자

## A Facial Animation System Using 3D Scanned Data

Bon Gwan Gu<sup>†</sup> · Chul-Hee Jung<sup>\*\*</sup> · Jaeyun Lee<sup>\*\*\*</sup> · Sun Young Cho<sup>\*\*\*\*</sup> · Myeong Won Lee<sup>\*\*\*\*\*</sup>

## ABSTRACT

In this paper, we describe the development of a system for generating a 3-dimensional human face using 3D scanned facial data and photo images, and morphing animation. The system comprises a facial feature input tool, a 3-dimensional texture mapping interface, and a 3-dimensional facial morphing interface. The facial feature input tool supports texture mapping and morphing animation - facial morphing areas between two facial models are defined by inputting facial feature points interactively. The texture mapping is done first by means of three photo images - a front and two side images - of a face model. The morphing interface allows for the generation of a morphing animation between corresponding areas of two facial models after texture mapping. This system allows users to interactively generate morphing animations between two facial models, without programming, using 3D scanned facial data and photo images.

Keywords : 3D facial modeling, 3D facial animation, 3D facial texture mapping, 3D facial morphing, Facial definition parameter

## 1. 서 론

최근 3차원 얼굴 표현을 위한 모델링과 렌더링에 관한 많은 연구가 많이 이루어졌다[1]. 이러한 연구 중에는 여러 표정의 얼굴을 자동 생성하거나 3D 모델에 얼굴의 여러 움직임 생성에 초점을 둔 연구가 있고 실시간 응용을 위한 얼굴 애니메이션 엔진 개발과 MPEG-4 기반의 얼굴 애니메이션

플레이어를 포함하는 프레임워크에 관한 연구가 진행되었다 [2-5]. Gaspard 등에 의한 FaceEngine은 스캐너에 의한 실사 모델이 아니라 그래픽스 도구에 의한 합성 얼굴 모델에 근육의 움직임을 생성하는 파라미터 정의에 의해 얼굴 모델을 생성하였다[6]. Igor 등에 의한 얼굴 모델도 간단한 합성 얼굴 모델로서 모바일 플랫폼에서의 MPEG-4 기반의 얼굴 애니메이션 플레이어를 구현하였다[7].

본 연구는 3D 스캐너에 의한 3D 원시 데이터와 고해상도 사진 이미지를 이용하여 실사에 가까운 3D 얼굴 모델을 구현하고, 얼굴 모델들간 자연스러운 모핑 애니메이션을 쉽게 생성해주는 대화형 얼굴 모핑 애니메이션 시스템 개발을 목적으로 한다. 본 논문은 이러한 시스템 개발에 필요한 3D 텍스처매핑, 3D 얼굴의 대응 영역 설정, 3D 얼굴 모핑 알고

† 정 회 원 : (주)슈프리마 기업부설 연구소 연구원  
\*\* 정 회 원 : Dodaam Systems Simulation Division Engineer  
\*\*\* 정 회 원 : (주)아크로디아코리아 기술기획팀  
\*\*\*\* 정 회 원 : KT 중앙연구소 부장  
\*\*\*\*\* 정 회 원 : 수원대학교 인터넷정보 공학과 부교수  
논문접수 : 2010년 8월 16일  
수정일 : 1차 2010년 10월 26일  
심사완료 : 2010년 11월 15일

리즘 구현과 시스템 기능에 대해서 기술한다.

## 2. 3D 얼굴 텍스처매핑

3D 스캐너는 실제 인물이나 물건들의 사실적인 묘사가 가능하여 3차원 기하 데이터를 정확하게 얻을 수 있다. 그러나, 아직까지는 사람이나 물체 표면의 속성을 나타내는 재질에 있어서는 사진과 같이 사실적 표현을 제공하지 못하고 있다. 본 연구에서는 3D 스캔 데이터와 얼굴 이미지의 수정 없이 3D 스캔 데이터에 얼굴 사진 이미지를 효과적으로 텍스처 매핑하여 사실적 렌더링 결과를 얻을 수 있는 방법을 개발하였다.

본 연구에서의 텍스처매핑 방법과 유사한 연구로서 3장의 사진 이미지로부터 얼굴 텍스처 매핑을 자동 생성하기 위한 그래픽 사용자 인터페이스 개발되었으나 정면과 좌우 측면 사이의 이미지 경계선에서 자연스럽게 않은 이미지 왜곡의 단점을 가진다[2]. 텍스처매핑의 다른 연구로 사용자 정의의 텍스처 특징과 메쉬의 기하 데이터를 대응시키는 프로그램 [4] 및 삼각형이나 사각형 외의 비정규 면을 위한 텍스처매핑이 소개되었다[5]. 그리고, 텍스처의 지역적 그리고 전역적 구조를 유지하면서 최소한의 왜곡으로 임의 곡면에 텍스처 매핑하는 기술[6], 뷰에 의존하는 실시간 텍스처매핑[7], 대규모 장면을 위한 텍스처 매핑[8,9], 그래픽스 하드웨어를 이용한 텍스처 매핑[10], 다면체의 특징 기반의 텍스처 매핑 처리 기술[11] 등이 있다. 본 연구에서의 텍스처매핑 기술은 3D 스캐너로 취득한 원시 얼굴 데이터의 저해상도 재질 문제를 해결하기 위해 사진을 이용하여 실사에 가까운 고해상도 재질의 3D 얼굴 모델 데이터를 생성할 수 있도록 한다.

### 2.1 3D 스캔 데이터와 얼굴 이미지의 일치

텍스처 매핑을 수행하기 위해서는 먼저 3D 스캔 데이터와 얼굴 이미지를 일치시켜야 한다. 정면 이미지에서는 양쪽 귀 끝과 이마 끝, 목선을 맞추어주고 좌우측 이미지는 코 끝과 귀 끝, 이마 끝, 목선을 맞추어 준다(그림 1). 그리고, 3D 스캔 데이터와 얼굴 이미지를 일치시켜도 3D 스캔 데이터와 얼굴 이미지의 좌표치가 다르기 때문에 좌표를 맞추는 작업이 필요하다. 이를 위해 3D 스캔 데이터와 얼굴 이미지의 좌표를 각각 0~1 좌표값의 범위로 조정하였다. 이때 3D 스캔 데이터는 3차원이고 얼굴 사진 이미지는 2차원이므로 얼굴의 정면 이미지와 3D 스캔 데이터의 좌표를 맞추기 위해 3D 스캔 데이터의 X, Y 좌표와 맞추고, 좌우측 이미지는 3D 스캔 데이터의 Z, Y 좌표와 맞추어 준다. 다음은 3D 스캔 데이터의 좌표를 0~1 사이값으로 맞추기 위한 코드이다:

```
for (i = 0; i < Vertex_Number; i++) {
    Vertex[i]->X = (Vertex[i]->X - Xmin) / Xmax;
    Vertex[i]->Y = (Vertex[i]->Y - Ymin) / Ymax;
    Vertex[i]->Z = (Vertex[i]->Z - Zmin) / Zmax; }
```

Vertex[i] 는 얼굴 모델의 임의의 정점을 말하며, Xmin, Ymin, Zmin과 XMax, YMax, ZMax는 각각 이미지의 최소점과 최대점의 좌표를 나타낸다.

얼굴 사진 이미지는 (0, 0) 지점부터 그리기 때문에 좌표를 0 ~ 1 사이값으로 만들기 위해서는 이미지의 X, Y 좌표를 너비값과 높이값으로 나누면 된다. 다음은 얼굴 사진 이미지 데이터의 좌표를 0~1에 맞추기 위한 코드이다:

```
for (i = 0; i < width * height; i++) {
    BMP_Vertex[i]->X = BMP_Pixel[i]->X / width;
    BMP_Vertex[i]->Y = BMP_Pixel[i]->Y / height; }
```

BMP\_Vertex는 정규화시킨 이미지 좌표이고 BMP\_Pixel은 원본 이미지 좌표를 나타낸다.

### 2.2 3D 스캔 데이터 정점과 얼굴 이미지의 픽셀 매치

3D 스캔 데이터와 얼굴 이미지의 좌표를 0~1 사이값으로 맞추어도 3D 스캔 데이터의 정점(버텍스) 좌표와 얼굴 이미지의 픽셀 좌표가 정확하게 일치하지는 않는다. 따라서 3D 스캔 데이터의 각 정점을 사진 이미지의 픽셀과 매치시켜주는 일이 필요하다. 3D 스캔 데이터의 각 정점 좌표 위치와 얼굴 사진 이미지의 픽셀 좌표 위치가 대부분 동일하지 않으므로 3D 얼굴 스캔 데이터의 각 정점 좌표와 가장 가까운 거리에 있는 얼굴 이미지의 픽셀 좌표를 구해야 한다. 이 때 얼굴 스캔 모델의 한 점과 사진 이미지의 각 픽셀 간의 거리를 구하여 가장 가까운 픽셀을 선택하여 매치시킨다. 얼굴 모델의 약 50,000개의 3D 스캔 데이터의 각각의 정점에 대해 가장 거리가 가까운 얼굴 이미지의 픽셀 좌표를 매치시켜야 하므로 3D 스캔 데이터의 정점 수보다 얼굴 이미지의 픽셀 수가 더 많아야 한다. 그러나 얼굴 이미지의 픽셀 수가 너무 많아질 경우 계산량의 증가로 프로그램 성능이 떨어질 수 있으므로 본 연구에서는 300 \* 300 크기의 얼굴 이미지를 사용해서 정면, 좌측, 우측 이미지가 각각 3D 스캔 데이터의 2배 정도인 90,000개의 픽셀을 갖도록 하였다.

3D 스캔 데이터의 정점과 얼굴 이미지의 픽셀을 다 매치시켜준 뒤에는 매치된 3D 스캔 데이터의 정점의 컬러 정보에 얼굴 이미지 픽셀의 컬러 값을 다음과 같이 적용시킨다:

```
Model_Vertex->Red = BMP_Pixel ->Red;
Model_Vertex->Green = BMP_Pixel ->Green;
Model_Vertex->Blue = BMP_Pixel ->Blue;
```

Model\_Vertex는 3D 얼굴 모델의 각 정점을 말하며 BMP\_Pixel은 얼굴 사진 이미지 중 얼굴 모델의 각 정점에 가장 가까운 픽셀을 나타낸다.

### 2.3 3D 얼굴 스캔 데이터의 멀티 텍스처매핑

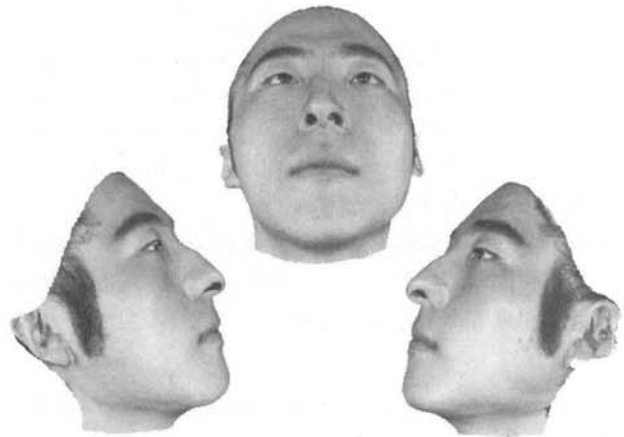
3D 스캔 데이터에 정면, 좌측 및 우측 이미지가 매핑될

영역을 구하기 위해서 MPEG-4에서 정의하는 FDP(Facial Definition Parameter) 특징점을 이용하였다(그림 1). FDP란 실시간으로 아바타를 표현하고 여러 가지 다양한 움직임을 표현하기 위해 얼굴 객체에 한하여 중요한 점들을 특징점으로 정의한 것이다[12].

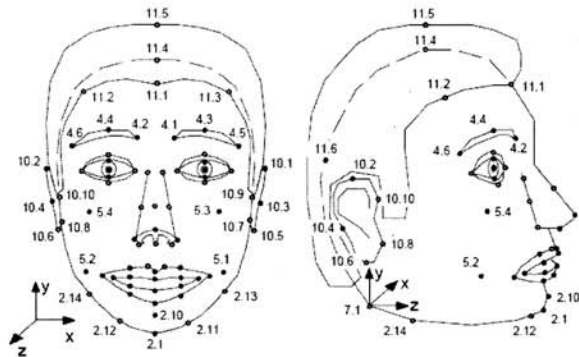
본 연구에서는 FDP의 점 4.5와 점 4.6의 X 좌표를 사용하여 점 4.5의 우측 부분은 얼굴 사진의 좌측 이미지, 점 4.5와 점 4.6 사이는 정면 이미지, 점 4.6의 좌측 부분은 얼굴 사진의 우측 이미지를 사용하여 멀티 텍스처매핑을 실행하였다(그림 2). 얼굴 텍스처매핑이 가장 잘 구현되기 위해서는 정면, 좌측, 우측 3장 이미지의 각 얼굴이 일치하는 부분에서 밝기나 색조, 채도 등이 최대한 동일하게 나와야 한다. 본 연구에서는 세 장의 이미지가 최대한 변화 없이 동일하게 나오게 하기 위해서 얼굴을 찍는 위치와 환경, 조명 값 등을 동일하게 하였다. 그러나 동일한 환경에서 사진을 찍었다 하더라도 서로 다른 세 방향에서 사진을 찍기 때문에 조명의 위치나 방향으로 인해 사진 이미지 컬러에서 밝기나 채도, 색조 등에서 약간씩의 차이가 나게 된다. 따라서 텍스처 매핑된 3D 스캔 데이터는 이미지 앞, 좌, 우 경계 부분에서 컬러 값이 차이가 나게 된다. 본 연구에서는 멀티 텍스처 매핑으로 완성된 3차원 얼굴 모델의 좌, 우 경계 부분에서의 컬러 값 차이를 없애기 위해 얼굴 모델의 경계 부분의 구간 1에서 구간 2 사이의 영역을 좌우로 100등분하여 컬러값을 보간하였다(그림 3). (그림 4)는 경계선에서의 컬러 보간을 통하여 얻은 멀티 텍스처매핑 결과이다.



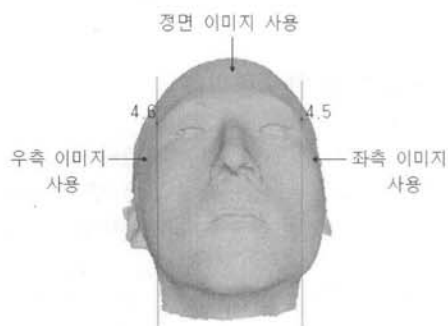
(그림 3) 3D 스캔 데이터의 경계 부분



(그림 4) 경계선 보간을 이용한 멀티 텍스처매핑 결과



(그림 1) MPEG-4 SNHC에서 제한한 FDP 특징점



(그림 2) 3D 스캔 데이터의 매핑 영역 분할

### 3. 얼굴 특징점 입력 도구

앞 절에서 기술한 멀티 텍스처매핑을 위해서 3D 스캔 데이터와 얼굴 사진의 매핑에는 MPEG-4에서 정의하는 FDP(Facial Definition Parameter) 특징점을 이용한다. 얼굴 특징점은 모델에 따라 그 위치가 달라질 수 있으므로 사용자 입력에 의한 특징점 정의가 필요하다. 이를 위해 본 연구에서는 3D 얼굴 모델 위에 정확하게 FDP 설정을 할 수 있도록 FDP 입력 도구를 개발하였다. FDP 입력 도구의 초기 실행 화면은 대부분의 3D 데이터 편집 툴에서 사용되는 화면과 비슷하게 구성되었다. 사용자는 Top, Front, Side, Free 네 개의 화면을 통해 모델을 이동 및 회전시킬 수 있으며 하단부에 있는 상태 표시줄에는 현재 마우스 포인터가 위치하는 곳의 3차원 좌표가 출력되므로 현재의 위치를 쉽게 파악할 수 있다(그림 5). Side View의 경우에는 반대편을 볼 수 있는 메뉴를 추가하여 3차원 데이터를 2차원 평면 상에서 완벽하게 표현할 수 있도록 구성하였다. <표 1>은 FDP 입력 도구 메뉴에서 제공하는 기능을 보여준다.

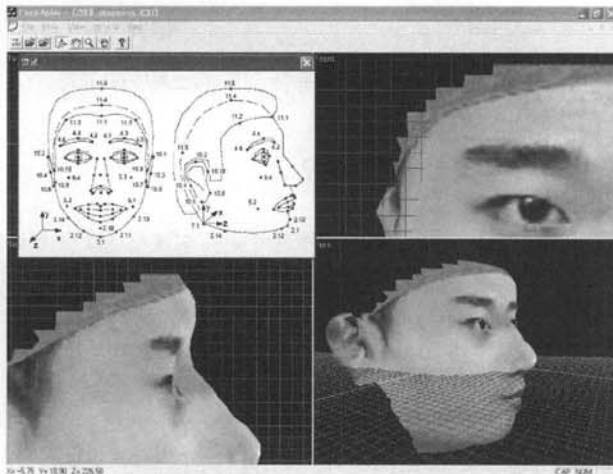
FDP 입력 도구를 이용하여 텍스처 매핑 처리된 얼굴 모델에 대해 FDP를 입력하는 순서는 다음과 같다.

가. 텍스처 매핑 처리가 된 3차원 얼굴 모델을 읽어들인다.  
 나. 프로그램 내에 저장되어 있는 MPEG-4 FDP 그림을 띄우고 Zoom, Translate 등을 통해 모델을 참조하기 좋게 위치시킨다.

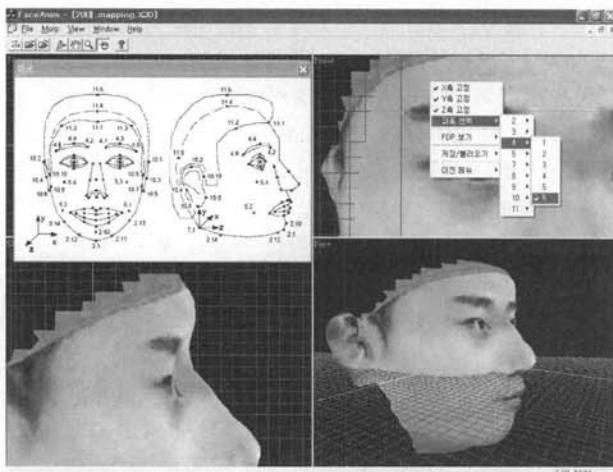
〈표 1〉 FDP 입력 도구 메뉴

종류	기능
Rotate	얼굴 모델을 회전시킨다. Free View 에서 가능
Zoom	얼굴 모델을 확대/축소시킨다.
Translate	얼굴 모델을 이동시킨다.
FDP	얼굴 모델에 FDP 입력을 시작한다.
View	Top/Front/Side/Free View 변경
Side View Change	현재 보고 있는 모델의 반대편 화면으로 전환한다.

- 다. FDP를 입력할 위치를 정하고 모델을 맞춘다. (그림 6)은 왼쪽 눈썹 끝에 있는 FDP 4.6을 입력하는 예를 보여준다.
- 라. Top, Front, Side 화면을 보며 사용자가 편한 위치에서 각 특징점을 위한 X, Y, Z 좌표를 선택한다. 이 때 먼저 Side View 상에서 Z축을 고정시킨 후, Front View 상에서 X축과 Y축을 선택한다.
- 마. 앞 단계에서 선택된 좌표의 해당 FDP 번호를 선택한다 (그림 1). X, Y, Z축 모두 선택하였으므로 해당 FDP 번호(4.6)에 입력시킨다(그림 6).



(그림 5) FDP 입력 도구

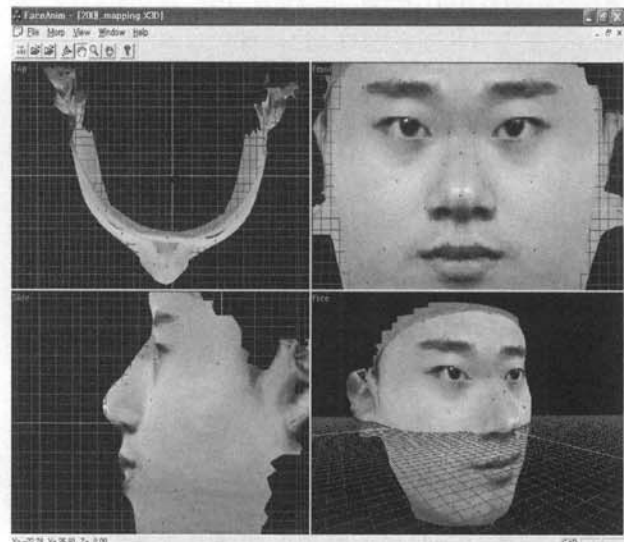


(그림 6) FDP 번호 입력

바. FDP를 모두 입력한 후 파일로 저장한다. 본 연구의 FDP 입력 도구는 FDP를 모두 입력한 경우와 입력을 완료하지 않은 모든 경우에 대해서 어느 때나 현재까지의 작업 내용을 파일로 저장할 수 있도록 하였다. FDP 파일에 저장되는 내용은 FDP의 Major, Minor 번호를 나타내며 그 뒤에 있는 실수 데이터 세 개는 각각 해당 위치의 X, Y, Z 좌표 값이고 마지막 정수 값은 정점의 인덱스를 나타낸다(그림 7). (그림 8)은 FDP 입력 도구를 이용하여 MPEG-4 얼굴 특징점을 입력한 최종 결과 화면의 예를 보여준다.

```
[ 2][ 1] -3.538849 -68.000000 45.031586 13360
[ 2][ 2] -5.238182 -22.000000 60.723831 46483
[ 2][ 3] -5.259933 -24.000000 59.209579 944
[ 2][ 4] 10.653214 -22.000000 55.927505 46494
[ 2][ 5] -19.587791 -22.000000 57.454788 46474
[ 2][ 6] 3.302895 -22.000000 58.546860 46489
[ 2][ 7] -12.330750 -22.000000 60.239182 46479
...
```

(그림 7) FDP 저장 파일 내용



(그림 8) FDP 입력 최종 결과 화면

#### 4. 3D 얼굴 모핑 애니메이션

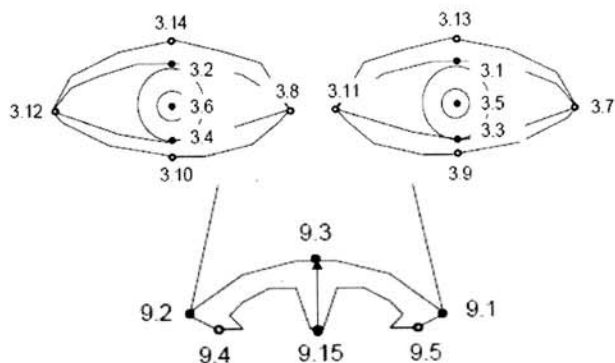
3D 스캐너로 취득한 3차원 얼굴 모델링 데이터를 이용하여 서로 다른 임의의 두 얼굴 모델 간의 애니메이션 생성할 때 어려운 점은 일반적으로 두 모델 간의 정점의 수가 다르고 대응하는 정점을 모른다는 것이다. Volker 등에 의한 Morphable Model에서도 이러한 서로 다른 폴리곤 수를 가지는 임의의 두 얼굴간의 모핑 문제는 다루지 못하고 있다

[13]. 이 문제를 해결하기 위해서 본 연구에서는 FDP를 기준으로 하여 두 얼굴 모델의 대응하는 영역을 찾고 이들 영역간의 폴리곤 매핑을 실현하는 방법을 고안하였다. 본 절의 내용은 이러한 임의의 두 모델간 모핑 애니메이션 알고리즘 개발에 대해 기술한다.

#### 4.1 얼굴의 정면 조정

임의의 두 얼굴 모델간의 모핑 애니메이션을 실현하는데 있어서 얼굴 모핑을 처리하기 전에 두 얼굴 모델을 정면으로 조정하는 전처리 과정이 필요하다. 이것은 각 얼굴 형상들의 바라보는 방향과 얼굴 영역을 일치시켜서 서로 매핑되는 위치에서 변환이 이루어지도록 하기 위한 것이다. 얼굴을 정면으로 두는 조정을 위해서는 앞에서 기술한 FDP를 사용하여 다음과 같이 위치시켰다(그림 9).

- 가. 먼저 FDP 점 중 [3.5]와 [3.6]을 사용하여 Y축을 중심으로 정면을 바라볼 수 있도록 회전시킨다.
- 나. Y축 중심으로 정면 조정후에 FDP 점 중 [9.3]과 [9.15]를 이용하여 X축을 중심으로 정면을 바라볼 수 있도록 회전시킨다.



(그림 9) 얼굴 정면 조정

#### 4.2 얼굴 펼침 처리

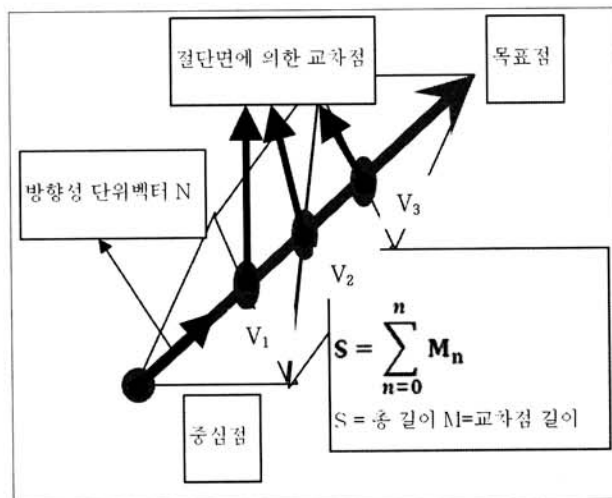
임의의 두 얼굴 모델간 애니메이션을 위해서는 두 얼굴의 대응하는 영역을 매치시키고 이들 영역간의 모핑 변환을 수행해야 한다. 그런데, 3D 얼굴 모델에서 눈, 귀, 코 등의 부분과 같이 굴곡이 심한 형태를 가지는 경우에는 대응하는 점들을 찾기가 어렵기 때문에 정확한 보간 데이터 추출이 어렵다. 이러한 문제를 해결하기 위하여 본 연구에서는 각 얼굴 모델의 정점과 폴리곤들을 서로 겹치지 않도록 펼친 후에, 앞에서 기술한 FDP를 기준으로 하여 영역 간 대응하는 정점과 폴리곤들이 변형되도록 하는 방법을 구현하였다. 얼굴을 하나의 평면상에 펼침 처리를 할 때 굴곡이 심한 부분의 데이터 손실을 막기 위한 방법으로 얼굴 중심점에서 모든 정점에 이르는 직선형 경로를 이용하였다(그림 10)[14]. 이 직선형 경로 방법은 얼굴 중심점에서 얼굴의 각 점들로 연결되는 모든 2차원 직선으로 모델을 절단하게 된다. 이와 같이 절단한 후에 생성된 절단면과 다각형과의 교차점을 구



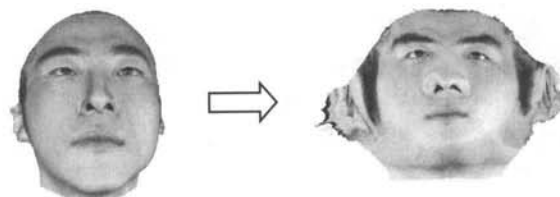
(그림 10) 직선형 경로

하여 그 모든 길이를 합한다(그림 11).

(그림 11)에서 V1, V2, V3 이 절단면과 다각형과의 교차점을 의미한다. 얼굴 중심점으로부터 모든 정점에 대한 경로 상의 교차점들의 거리의 합을 S라고 할 때 각 정점의 3D 모델상의 방향성 단위 벡터(N)와 S를 곱한 값으로 각 정점의 새로운 펼쳐진 좌표를 구한다. (그림 12)는 이 방법으로 얻어진 펼쳐진 얼굴 모델의 결과를 보여준다.



(그림 11) 직선에 의한 절단면의 교차점

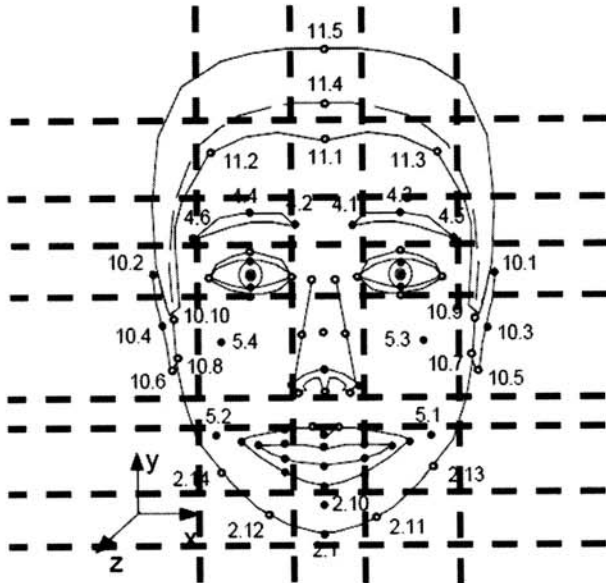


(그림 12) 펼쳐진 얼굴 모델

#### 4.3 얼굴의 대응 영역 설정

두 얼굴 간 모핑 애니메이션을 위해서는 얼굴의 같은 영역끼리의 애니메이션이 생성되도록 해야 한다. 이를 위해서 두 얼굴의 펼쳐진 모델을 통해 두 얼굴의 유사 영역끼

리 보간되도록 얼굴 특징점을 기준으로 얼굴 영역을 설정하여 대응하는 같은 영역끼리 보간될 수 있도록 준비한다. 각 영역은 (그림 13)과 같이 FDP를 경계로 하는 영역으로 설정한다.



(그림 13) FDP를 이용한 대응 영역 설정

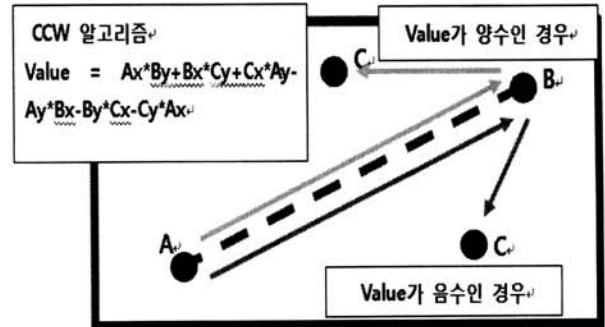
#### 4.4 얼굴 모핑 알고리즘

본 연구에서는 서로 다른 두 얼굴 모델 사이에서 모핑을 위해 두 얼굴의 대응하는 영역간의 보간 방법을 정의하였다. 펼쳐진 얼굴 모델에 얼굴 영역을 설정한 뒤에 두 얼굴 모델간의 보간 데이터를 얻는 알고리즘을 구성하였다. 이 알고리즘은 대응하는 영역 내의 대응하는 폴리곤을 구하고, 영역별 각 정점에 대한 두 모델간 대응 위치를 찾아서 모핑을 위한 중간값을 구한다.

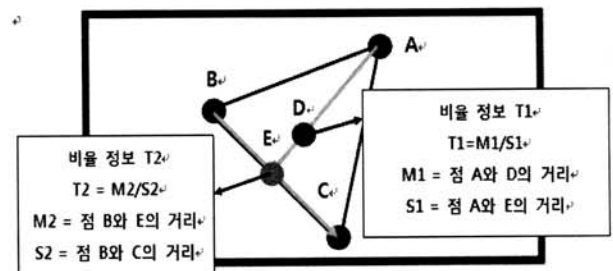
두 얼굴 사이의 대응하는 폴리곤의 모핑을 구하기 위한 방법으로 한 얼굴 모델 A의 한 정점과 다른 얼굴 모델 B의 폴리곤과의 교차 처리를 통해서 모핑을 위한 보간 값을 계산하는 알고리즘을 구성하였다. 본 논문에서의 얼굴 모핑 알고리즘은 다음의 세 단계를 거친다.

첫 번째 단계에서는 각 대응 영역에서 얼굴 모델 A의 한 점이 얼굴 모델 B의 어느 다각형에 포함되어 있는가를 CCW (Counter Clock Wise) 기하 알고리즘을 이용하여 찾는다(그림 14). CCW 알고리즘은 세 점의 방향이 반시계 방향인지 시계 방향인지를 판단하는데 사용하는 알고리즘을 말한다[15].

두 번째 단계에서는 모든 정점에 대해 대응하는 폴리곤 탐색을 완료한 후에 (그림 15)에서와 같이 각 정점과 해당 폴리곤의 비율 데이터를 계산한다. 이 비율 데이터를 이용하여 3D 메쉬 데이터의 모핑에 필요한 모든 중간 정점을 위한 보간 데이터를 추출한다. 다시 말해서, 3D 얼굴의 펼쳐진 모델의 각 정점에 대해 시작점으로부터 대응하는 폴리곤에 이르기까지의 모핑을 위한 모든 중간 정점 좌표값을 비



(그림 14) CCW 알고리즘



(그림 15) 정점의 비율 정보

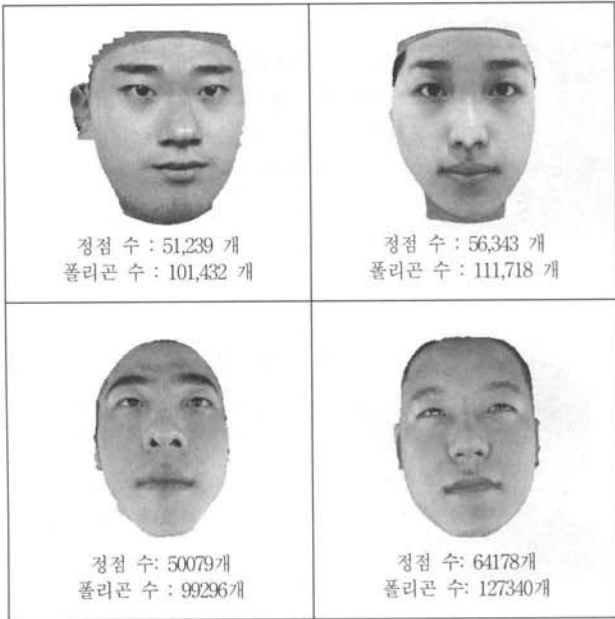
율 정보를 이용하여 보간값을 구하여 추출해낸다.

마지막 단계에서는, 모든 정점에 대해 모핑에 필요한 좌표치와 컬러값을 위의 비율정보를 이용하여 각 정점 당 모든 중간 값들을 계산하여 렌더링을 실행한다.

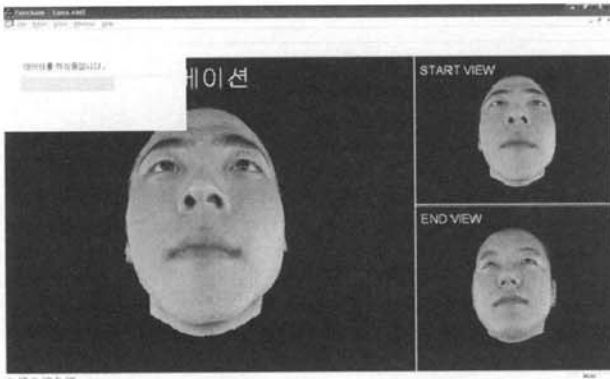
## 5. 실험 및 분석

본 연구의 실험 데이터를 위해 기술표준원의 Size Korea 사업에서 생성된 원시 얼굴 DB와 함께 3D 스캐너로 직접 취득한 얼굴 데이터를 사용하였다. 프로그래밍 환경은 비주얼 C++와 OpenGL 그래픽 라이브러리를 사용하였다. 얼굴의 원시 스캔 데이터는 사람의 얼굴만이 아닌 인체 전체를 스캐닝한 데이터이기 때문에 얼굴 데이터만을 추출하였을 경우 버텍스 수가 5,000개 정도로 해상도가 많이 떨어진다. 이러한 원본 3D 스캔 데이터를 고해상도로 만들기 위해 3D 스캔 데이터의 버텍스 수를 증가시키고, 인덱스를 새롭게 정렬시켜 주는 작업이 필요하다. 이를 위해 5,000개의 버텍스의 중간마다 새로운 버텍스를 추가하였다. 이러한 전처리 과정을 거친 후에 본 연구에서 모핑 애니메이션을 위해 사용된 얼굴 모델들의 정점 개수는 20,000~60,000개이고 폴리곤 수는 50,000~120,000개 범위를 가지고 있다(그림 16).

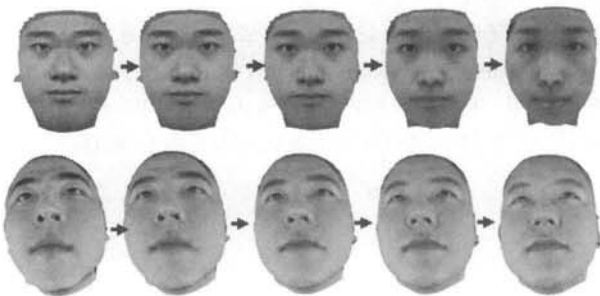
(그림 17)은 본 연구에서 구현한 모핑 애니메이션 생성 도구이며 (그림 18)는 모핑 애니메이션을 수행한 결과를 보여준다. 이 결과 그림에서 볼 때 3D 스캐너가 취득한 얼굴 모델에 비해서 실제 얼굴 모습을 한 해상도가 높은 3차원 얼굴 모델과 자연스러운 모핑 애니메이션 결과를 확인할 수 있다.



(그림 16) 모핑에 사용된 얼굴 모델의 예



(그림 17) 얼굴 모핑 애니메이션 생성 도구



(그림 18) 3D 얼굴 모핑 애니메이션 생성 결과

## 6. 결 론

본 논문에서는 3D 스캐너에 의해 획득한 얼굴 원시 데이터와 2D 얼굴 사진을 이용하여 실사적인 3D 얼굴 모델을 생성한 후에, 이 얼굴 모델들 간의 변형을 시뮬레이션하는 모핑 알고리즘을 적용하여 3D 얼굴 애니메이션을 자동 생

성해주는 시스템 개발에 대해 기술하였다. 본 시스템의 특징은 얼굴 사진 이미지를 이용하여 실사에 가까운 사실적 3D 얼굴 모델과 임의의 두 얼굴간의 모핑 애니메이션을 쉽게 생성시키는 기능을 갖는다. 본 연구에서 모핑 애니메이션 구현에 있어서 중요한 기능으로는 얼굴 모델의 펼침 처리 알고리즘을 개발하고, 특징점 입력 인터페이스에 의한 얼굴간의 영역 기반 대응 방법을 통하여 코, 눈, 귀 등과 같이 굴곡이 심한 부분에서의 보간 데이터가 제대로 추출되지 않았던 기존 연구의 문제점을 해결한 점과, 서로 다른 폴리곤 수를 가지는 임의의 두 얼굴의 모핑 애니메이션을 자동으로 생성할 수 있다는 점이다.

이 시스템은 3D 얼굴 애니메이션 생성 및 얼굴 관련 제품 제작에 활용될 수 있고 모핑 알고리즘에서 사용된 직선형 경로의 기본적인 알고리즘에 의해 3D 모델의 치수나 면적 데이터 추출 등에 활용될 수 있다. 또한, 3D 얼굴 형상 모델을 이용하여 얼굴 전체뿐 아니라 눈, 코, 귀 등의 부분적 가상 성형의 시뮬레이션으로도 활용이 가능하다. 이외에도 향후 얼굴과 관련한 많은 산업 즉, 의료, 성형, 미용, 3차원 캐릭터, 의공학, 인간공학, e-헬스 등의 분야에서 과학적인 3차원 데이터 작성과 시뮬레이션에 사용할 수 있을 것이다.

## 참 고 문 헌

- [1] Jorg Haber, Demetri Terzopoulos, "Facial modeling and animation", ACM SIGGRAPH 2004 Course Notes, No.6, 2004.
- [2] Silvina L. Ferradal and Juan C. Gomez, "A Graphical User Interface for Automatic Facial Texture Mapping Based on Orthogonal Photos", 11th International Multimedia Modelling Conference (MMM'05), pp.271-276, January 2005.
- [3] Yuencheng Lee, Demetry Terzopoulos and Keith Waters, "Realistic Modeling for Facial Animation", ACM SIGGRAPH, pp.55-62, 2005.
- [4] Vladislav Kraevoy , Alla Sheffer and Craig Gotsman, "Matchmaker: constructing constrained texture maps", ACM Transactions on Graphics (TOG), pp.29-36, July 2003.
- [5] Jin Jin Zheng and Jian J Zhang, "Texture Mapping on Irregular Topology Surface", Sixth International Conference on Information Visualisation (IV'02), pp.323, July 2002.
- [6] Gaspard Breton, Christian Bouville, and Danielle Pele, "FaceEngine A 3D Facial Animation Engine for Real Time Applications", Proceedings of WEB3D'2001, pp.15-22, 2001.
- [7] Igor S. Pandzic, "Facial Animation Framework for the Web and Mobile Platforms", Proceedings of Web3D, pp.27-34, 2002.
- [8] G. Zigelman , R. Kimmel and N. Kiryati, "Texture Mapping Using Surface Flattening via Multidimensional Scaling", IEEE Transactions on Visualization and Computer Graphics, pp.198-207, April 2002.
- [9] Damien Porquet , Jean-Michel Dischler and Djamchid Ghazanfarpour, "Real-time high-quality View-Dependent

Texture Mapping using per-pixel visibility”, Proceedings of the 3rd international conference on Computer graphics and interactive techniques in Australasia and South East Asia.

- [10] Hiroyuki Inatsuka , Makoto Uchino and Masahiro Okuda, “Level of Detail Control for Texture on 3D Maps”, 11th International Conference on Parallel and Distributed Systems - Workshops (ICPADS'05), pp.206-209, July 2005.
- [11] Lingyun Liu , Loannis Stamos , Gene Yu , George Wolberg and Siavash Zokai, “Multiview Geometry for Texture Mapping 2D Images Onto 3D Range Data”, 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Vol.2 (CVPR'06), pp.2293-2300, June 2006.
- [12] ISO/IEC IS 14496-2 Visual, 1999.
- [13] Volker Blanz, Thomas Vetter, “A morphable model for the synthesis of 3D faces”, Proceedings of the 26th annual conference on Computer graphics and interactive techniques, pp.187-194, 1999.
- [14] 이성열, 이해영 “절단면을 이용한 메쉬에서의 직선형 경로 찾기”, 2006 컴퓨터 그래픽스 학회 하계 논문집, pp.77-78, 2006.
- [15] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, Computational Geometry: Algorithms and Applications, Springer, 2010.



**구 본 관**

e-mail : bon1305@gmail.com  
 2009년 수원대학교 인터넷정보공학과(학사)  
 2009년~현재 (주)슈프리마 기업부설 연구소 연구원  
 관심분야: 3D, 임베디드시스템



**정 철 희**

e-mail : whiskerfe@hanmail.net  
 2009년 수원대학교 인터넷정보공학과(학사)  
 2009년~현재 (주)Dodaam Systems Simulation Division Engineer  
 관심분야: 3D 입체 영상



**이 재 윤**

e-mail : leejae44@naver.com  
 2010년 수원대학교 인터넷정보공학과(학사)  
 2009년~현재 (주)아크로디아코리아 기술기획팀  
 관심분야: 모바일3D, 3D UI, 3D 게임



**조 선 영**

e-mail : suny@kt.com  
 1991년 서강대학교 전자계산학과(학사)  
 1993년 서강대학교 전자계산학과(석사)  
 2004년 서울대학교 전기전자 공학부 (박사 수료)  
 1993년~현재 KT 중앙연구소 부장  
 관심분야: 컴퓨터그래픽스, HCI, 지능형데이터처리 등



**이 명 원**

e-mail : mwlee@suwon.ac.kr  
 1981년 서울대학교(학사)  
 1984년 서울대학교 계산통계학과 전산전공(석사)  
 1990년 일본 동경대학(The U. of Tokyo) 정보과학과(박사)  
 1990년~1993년 일본 Kubota Corp. 및 동경대학(The U. of Tokyo) 연구원  
 1984년~1986년 데이콤연구소 연구원  
 1993년~1996년 KT 멀티미디어 연구소 선임연구원  
 1996년~2001년 수원대학교 컴퓨터학과 전임강사, 조교수  
 2002년~현재 수원대학교 인터넷정보공학과 부교수  
 관심분야: 컴퓨터그래픽스, 가상현실, 애니메이션, 웹3D