

## 3차원 스캔 얼굴 모델의 텍스처 매핑

정철희<sup>o</sup> 조선영\* 이명원

수원대학교 인터넷정보공학과

\*KT 미래기술연구소

[whiskerfe@hanmail.net](mailto:whiskerfe@hanmail.net), [suny@kt.co.kr](mailto:suny@kt.co.kr), [mwlee@suwon.ac.kr](mailto:mwlee@suwon.ac.kr)

### Texture Mapping of 3D Scan Face Models

Chul Hee Jung<sup>o</sup>, Sun Young Cho\*, Myeong Won Lee

Dept. of Internet Information Engineering, The U. of Suwon

\*KT Future Technology Laboratory

#### 요 약

3D 스캐너의 보급으로 3차원 모델 생성이 가능하게 되었으나 결과로 얻은 형상이 아직 사진에서와 같은 사실적 묘사에 미치지 못하고 있다. 본 논문에서는 얼굴의 3D 스캔 데이터에 효율적인 텍스처 매핑을 통해 사실적 렌더링 결과를 얻을 수 있는 방법을 기술한다. 3D 얼굴 스캔 데이터와 얼굴의 사진 이미지의 좌표를 정확하게 맞추어 3D 스캔 데이터와 얼굴 이미지의 버텍스를 매치시켜주고, 얼굴 이미지의 해당 버텍스에 들어 있는 칼라 값을 3D 스캔 데이터의 버텍스에 넘겨주는 텍스처 매핑을 구현한다. 본 논문에서는 정면, 좌측, 우측 3장의 이미지를 이용하여 간단히 멀티텍스처 매핑을 수행하는 방법과 이 때 발생하는 사진 간의 경계선에서 발생하는 문제 해결에 대해 기술한다.

#### 1. 서론

최근 3D 스캐너의 보급으로 실제 인물이나 물건들의 사실적인 묘사가 가능해져서 3차원 기하 데이터를 정확하게 얻게 되었다. 그러나, 아직까지는 사람이나 물체 표면의 속성을 나타내는 재질에 있어서는 사진과 같이 사실적 표현을 제공하지 못하고 있는 것이 문제이다.

3D 모델 데이터에 2D 사진 이미지로 텍스처 매핑을 하기 위해서는 2D 사진 이미지에 3차원 정보가 이미 배제되어 있기 때문에 정확한 3차원 속성 정보를 얻어내기 어려운 점이 있다. 그러나, 2D 사진의 재질의 성능과 사용의 편리성을 고려할 때 효율적인 텍스처 매핑이 제공될 수 있다면 스캔 데이터에 사실적 텍스처 표현을 쉽게 할 수 있어서 인물의 사실적 렌더링이 보다 쉽게 이루어질 수 있을 것이다.

본 연구와 관련한 연구 현황으로는 3장의 사진 이미지로부터 얼굴 텍스처 매핑을 자동 생성하기 위한 그래픽 사용자 인터페이스 개발[1] 및 레이저 스캐너 데이터를 이용하여 물리 기반의 근육과 두개골의 변형으로 3차원 얼굴 모델을 생성한 연구 결과가 있다[2]. 또한 사용자 정의의 텍스처 특징과 메쉬의 기하 데이터를 대응시키는 프로그램[3] 및 삼각형이나

사각형 외의 비정규 면을 위한 텍스처 매핑이 소개되었다[4]. 그리고, 텍스처의 지역적 그리고 전역적 구조를 유지하면서 최소한의 왜곡으로 임의 곡면에 텍스처 매핑하는 기술[5], 뷰에 의존하는 실시간 텍스처매핑[6], 대규모 장면을 위한 텍스처 매핑[7][8], 그래픽스 하드웨어를 이용한 텍스처 매핑[9], 다면체의 특징 기반의 텍스처 매핑 처리 기술[10] 등이 있다.

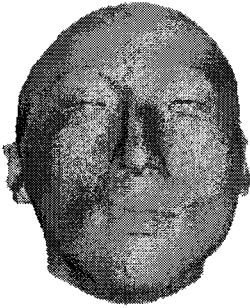
본 논문에서는 사진의 재질 데이터를 이용하여 사람 얼굴의 3D 스캔 데이터에 효과적으로 텍스처 매핑을 할 수 있는 기법을 기술한다.

#### 2. 3D 얼굴 스캔 데이터와 사진 이미지의 좌표 매치

3D 스캔 데이터에 2D 사진 이미지를 이용하여 텍스처 매핑을 하려면 스캔 데이터의 좌표와 사진 이미지의 좌표를 일치시키는 일이 필요하다(그림 1).

본 연구에서 사용된 3D 스캔 데이터의 버텍스 수는 약 50,000 개이며 각 버텍스의 컬러 값을 사진 이미지로부터 가져오기 위해서는 먼저 스캔된 얼굴 모델의 각 좌표에 대응하는 사진 이미지의 좌표를 찾아야 한다.

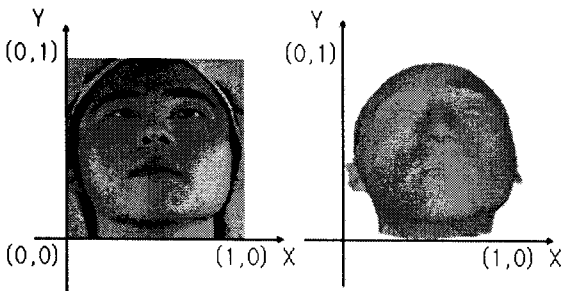
\* 본 연구는 기술표준원의 국가표준, 제품안전관리, 인증제도 선진화 기반구축을 위한 학술연구용역사업의 지원으로 수행되었음.



(그림 1) 3D 얼굴 스캔 데이터

본 연구에서의 3D 스캔 데이터의 버텍스 수는 약 50,000 개이며 각 버텍스의 컬러 값을 사진 이미지로부터 가져오기 위해서는 먼저 스캔된 얼굴 모델의 각 좌표에 대응하는 사진 이미지의 좌표를 찾아야 한다.

3D 얼굴 스캔 데이터와 얼굴 사진 이미지의 좌표를 정확하게 맞추기 위해서 먼저 3D 얼굴 스캔 데이터와 얼굴 사진 이미지의 좌표를 각각 0~1 좌표값의 범위로 조정하였다(그림 2).



(그림 2) 3D 스캔 데이터와 얼굴 이미지의 좌표 조정

3D 스캔 데이터의 좌표를 0~1 사이값으로 맞추기 위해서는 얼굴 스캔 모델의 모든 X, Y, Z 좌표 중에서 최소값과 최대값을 구한다. 3D 스캔 데이터 각각의 버텍스 마다 X, Y, Z 좌표의 최소값을 빼주어 (0, 0, 0) 지점을 3D 스캔 데이터 모델을 그리는 시작점으로 만들고, 3D 스캔 데이터 각각의 버텍스들을 X, Y, Z 좌표의 최대값으로 나누어서 3D 스캔 데이터의 좌표가 0 ~ 1 사이의 좌표를 가지도록 하였다.

```
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; }
```

얼굴 이미지는 (0, 0) 지점부터 그리기 때문에 좌표를 0 ~ 1 사이값으로 만들기 위해서는 이미지의 X, Y 좌표를 너비값과 높이값으로 나누어 주면 된다.

```
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; }
```

3D 스캔 데이터와 얼굴 이미지의 좌표 범위를 맞추어 준 뒤에는 3D 스캔 데이터의 각 버텍스를 얼굴 이미지의 픽셀과 매치시켜주는 일이 필요하다. 본 연구에서 3D 스캔 데이터의 버텍스 수 약 50,000 개에 대해, 얼굴 사진 이미지는 300 \* 300 크기의 이미지를 사용해서 90,000 개의 픽셀을 가진다.

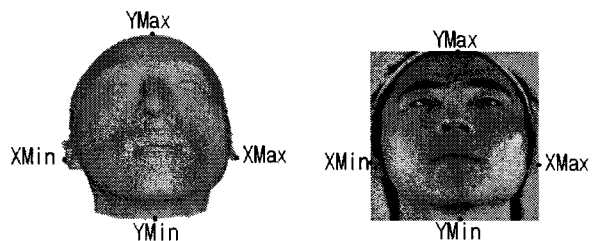
3D 얼굴 스캔 데이터의 각 버텍스 좌표 위치와 얼굴 사진 이미지의 픽셀 좌표 위치가 대부분 동일하지 않으므로 3D 얼굴 스캔 데이터의 각 버텍스 좌표와 가장 가까운 거리에 있는 얼굴 이미지의 픽셀 좌표를 구해야 한다. 이 때 얼굴 스캔 모델의 한 점과 사진 이미지의 한 점 간의 거리를 구하는 다음 공식이 사용된다.

$$\text{Distance} = \text{fabs}(\text{sqrt}(\text{pow}(\text{fabs}(x_2 - x_1), 2) + \text{pow}(\text{fabs}(y_2 - y_1), 2))))$$

3D 스캔 데이터의 각 버텍스에 대해 Distance 값이 가장 작은 이미지의 픽셀을 구해 매치시킨다.

### 3. 얼굴 모델의 컬러 값

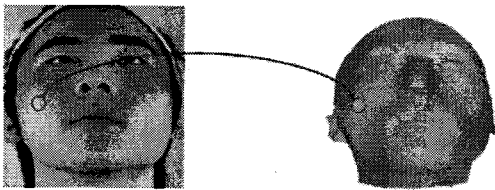
3D 스캔 모델의 좌표와 사진 얼굴 이미지의 좌표를 일치시킨 후에는 3D 얼굴 스캔 모델의 각 좌표에 대해서 얼굴 사진 이미지로부터 각 버텍스를 위한 컬러 값을 취하는 일이 필요하다. 이를 위해 매치시킨 얼굴 이미지 픽셀에 들어있는 컬러 값을 3D 스캔 데이터의 버텍스에 남겨준다. 이때 3D 스캔 데이터 좌표와 얼굴 이미지 좌표의 최소값과 최대값을 맞추어 주었기 때문에 (그림 3)과 같이 3D 스캔 데이터와 얼굴 이미지의 양쪽 귀 끝과 이마 끝, 목선을 맞추어서 두 데이터가 일치하게 만든다.



(그림 3) 얼굴 스캔 좌표와 사진 이미지의 좌표

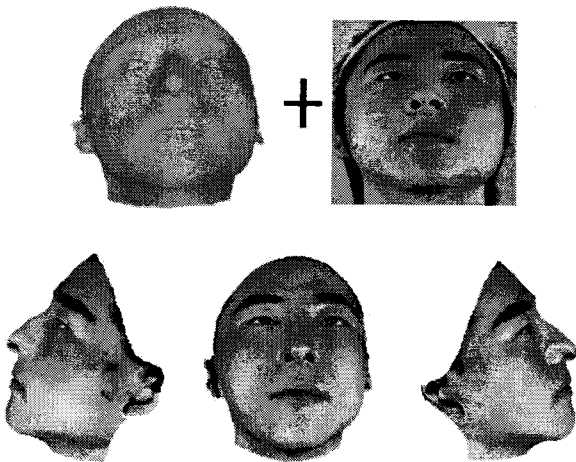
3D 스캔 데이터가 가지고 있는 버텍스의 좌표와 일치하는 얼굴 이미지의 픽셀 좌표가 없는 경우에는 가장 가까이 있는 픽셀 좌표를 찾아 얼굴 이미지의 칼라값을 다음과 같이 3D 스캔 데이터의 각 버텍스에 넘겨주게 된다(그림 4).

```
//Model_Vertex : 3D 스캔 데이터의 버텍스
//BMP_Vertex : 얼굴 이미지의 버텍스
Model_Vertex->Red = BMP_Vertex->Red;
Model_Vertex->Green = BMP_Vertex->Green;
Model_Vertex->Blue = BMP_Vertex->Blue;
```



(그림 4) 칼라 값 전달

그 결과 (그림 5)와 같이 3D 얼굴 스캔 좌표에 사진 이미지의 컬러 값을 적용시켜 사실적인 얼굴 표현이 가능하게 된다. 그런데, 이 때 정면 이미지만을 사용하여 컬러 값을 취한 경우에는 3D 얼굴 스캔 데이터의 정면은 잘 나오나 좌우측면이 늘어지는 것을 볼 수 있다. 이 부분을 해결하기 위해서 본 연구에서는 정면 이미지만이 아닌 좌우측 이미지도 같이 사용하는 멀티텍스처 매핑을 사용하였다.



(그림 5) 텍스처 매핑 결과

#### 4. 멀티텍스처 매핑

본 절에서는 3D 스캔 데이터의 얼굴 좌우측면의 늘어짐을 없애기 위해 정면의 이미지만이 아닌 정면, 좌측,

우측 3장의 이미지를 함께 사용하는 멀티텍스처 매핑에 대해 기술한다. 멀티 텍스처 매핑이 가장 잘 구현되기 위해서는 정면, 좌측, 우측 3 장 이미지의 각 얼굴이 일치하는 부분에서 밝기나 색조, 채도 등이 최대한 동일하게 나와야 한다. 본 연구에서는 3 장의 이미지가 최대한 변화 없이 동일하게 나오게 위해 얼굴을 찍는 위치와 환경, 조명 값 등을 동일하게 해 주었다(그림 6).

본 연구에서의 멀티텍스처 매핑은 3D 스캔 모델 버텍스의 컬러 값을 취하는데 있어서 스캔 모델의 정면에 해당하는 버텍스에는 정면의 사진 이미지를 사용하고, 스캔 모델의 좌우측 버텍스에는 각각 사진의 좌측 이미지와 우측 이미지를 사용하여 매핑시켜준다.

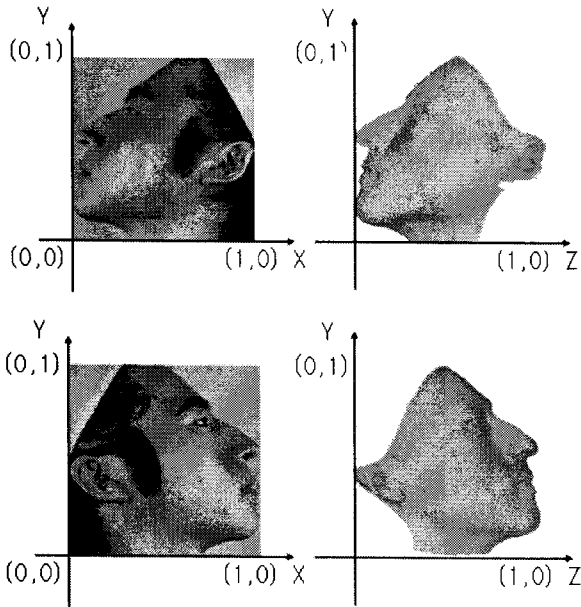
좌측	정면	우측

(그림 6) 좌측, 정면, 우측 이미지

3D 스캔 데이터에 좌측과 우측 이미지를 매핑하기 위해서는 X,Y 평면이 아닌 Z,Y 평면을 이용하여 매핑을 해야 한다. 좌우측 이미지도 정면을 매핑할 때와 같이 좌표의 범위를 0 ~ 1 사이값으로 맞추어서 3D 스캔 데이터와 얼굴 이미지가 일치하도록 조정해준다.

얼굴 스캔 데이터의 좌표와 사진 이미지의 좌표를 매치 시켜주기 위해서는 얼굴 이미지의 X, Y 좌표와 3D 스캔 데이터의 Z, Y좌표를 이용하여 매치시켜준다 (그림 7). 3D 스캔 데이터에 정면, 좌측 및 우측 이미지가 매핑될 영역을 구하기 위해서는 MPEG-4 에서 정의하고 있는 FDP(Facial Definition Parameter) 특징점을 이용하였다(그림 8). FDP란 실시간으로 아바타를 표현하고 여러 가지 다양한 움직임을 표현하기 위해 얼굴 객체에 한하여 중요한 점들을 특징점으로 정의한 것이다[11].

본 연구에서는 FDP의 점 4.5 와 점 4.6 의 X좌표를 사용하여 점 4.5 의 우측 부분은 얼굴 사진의 좌측 이미지, 점 4.5 와 점 4.6 사이는 정면 이미지, 점 4.6 의 좌측 부분은 얼굴 사진의 우측 이미지를 사용하여 앞의 방법으로 텍스처 매핑을 실행하였다(그림 9).



(그림 7) 3D 스캔 데이터와 좌우측 이미지의 매치

### 5. 멀티 텍스처 매핑의 후 처리

본 절에서는 앞에서 기술한 3D 얼굴 스캔 데이터에 정면, 좌측, 우측 이미지를 멀티 텍스처 매핑한 후에 정면과 좌측 얼굴, 그리고 정면과 우측 얼굴의 경계선에 생길 수 있는 컬러 값의 불연속성에 대한 후 처리에 대해 설명한다.

사진 이미지 3 장을 사용하여 텍스처 매핑을 하게 되면, 아무리 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 멀티 텍스처 매핑된 3D 스캔 데이터는 이미지 앞, 좌, 우 경계부분에서 칼라값이 차이가 나게 된다(그림 10).



(그림 10) 멀티텍스처 매핑

이 부분을 해결하기 위해 본 연구에서는 (그림 11)과 같은 3D 스캔 데이터 경계 부분에서 양끝 칼라값의 차이를 구해 그 차이만큼 경계 부분이 조금씩 보간되도록 하였다.

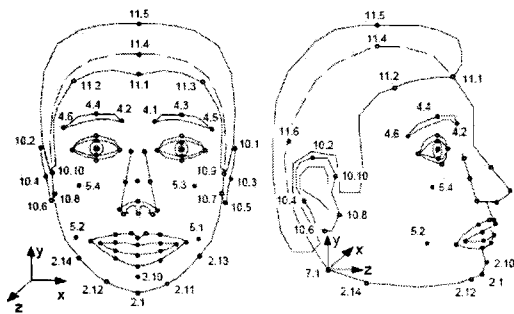


(그림 11) 3D 스캔 데이터의 경계 부분

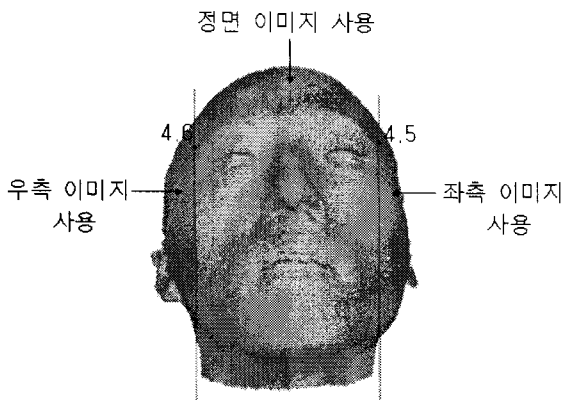
(그림 12)는 3 장의 사진 이미지로 멀티텍스처 매핑한 후 3D 스캔 데이터에서의 앞, 좌, 우 이미지 경계부분을 보간해준 결과를 보여준다.

### 6. 결론

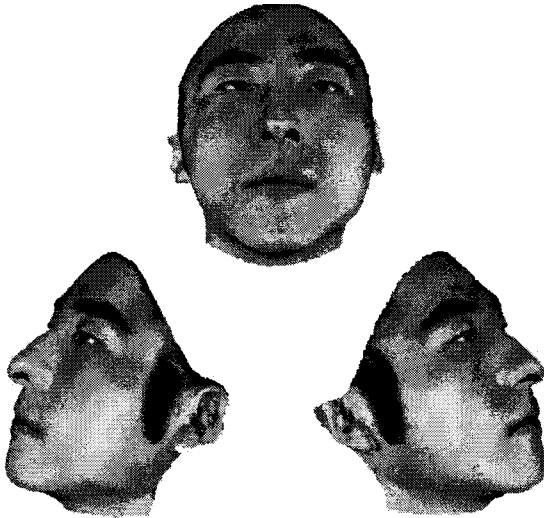
본 연구에서는 3D 얼굴 스캔 데이터에 얼굴의 사진 이미지 3 장을 이용하여 효과적으로 텍스처 매핑 할 수 있는 시스템을 개발하였다. 본 연구의 결과는 3D 스캔 모델에 사진 이미지의 텍스처를 그대로 표현하므로 사진 이미지와 동일한 해상도의 재질로 3차원 얼굴 모델을 완성한다는 점에서 다른 도구를 사용하여 사실과는 다른 임의의 조작을 할 필요 없이 유용하게 활용될 수 있다.



(그림 8) MPEG-4 SNHC에서 제안한 FDP 특징점



(그림 9) 3D 스캔 데이터의 매핑 영역 분할



(그림 12) 보간된 멀티텍처 매핑 결과

본 연구의 결과는 실제 사람 얼굴을 3D 스캐너로 정확히 측정하고 실제 모습과 동일한 사진 이미지를 사용하여 얼굴의 표면을 완성하고 있어서 실제 얼굴과 동일하다고 볼 수 있다. 따라서 실제 사람 얼굴 표현이 필요한 의로나 성형 분야에서 활용될 수 있다.

향후 연구 과제로는 코, 입, 귀 등 굴곡이 있는 섬세한 모델에서 사진 이미지를 이용하여 실제 모델에 근사한 텍스처 매핑 기술을 개발하는 일이다. 그리고, 얼굴의 앞면과 옆면뿐 아니라 사람의 얼굴과 머리를 포함하는 전체 두상의 사실적 표현도 해결해야 할 문제이다.

#### 참고문헌

- [1] 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
- [2] Yuencheng Lee, Demetry Terzopoulos and Keith Waters, "Realistic Modeling for Facial Animation", ACM SIGGRAPH, pp. 55-62, 2005.
- [3] Vladislav Kraevoy, Alla Sheffer and Craig Gotsman, "Matchmaker: constructing constrained texture maps", ACM Transactions on Graphics (TOG), pp. 29-36, July 2003.
- [4] 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.
- [5] 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.

[6] 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 (GRAPHITE '05), pp. 560-564, November 2005.

[7] 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.

[8] Lingyun Liu, Ioannis 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 - Volume 2 (CVPR'06), pp. 2293-2300, June 2006.

[9] Martin Kraus, "Adaptive texture maps", Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware, pp. 7-15, 2002.

[10] Eugene Zhang, Konstantin Mischaikow and Greg Turk, "Feature-based surface parameterization and texture mapping", ACM Transactions on Graphics (TOG), pp. 396-403, January 2005.

[11] ISO/IEC IS 14496-2 Visual, 1999.