

반자동적인 대응점 찾기를 이용한 3차원 얼굴 모델 생성

(Building a 3D Morphable Face Model using Finding Semi-automatic Dense Correspondence)

최인호[†] 조선영^{**}
(Inho Choi) (Sun Young Cho)

김대진^{***}
(Daijin Kim)

요약 2D 기반의 얼굴 분석 및 처리 알고리즘은 포즈 및 조명에 강인하지 못한 문제점들이 존재한다. 이러한 이유로 과거 3D 기반의 얼굴 분석 및 처리 분야에 많은 연구를 진행하려 하였지만, 컴퓨팅 파워의 한계와 고속 스캐너의 부재 등으로 많은 연구가 진행되지 못하였다. 하지만 오늘날 하루가 다르게 빨라지고 있는 컴퓨터의 성능으로 인해 주춧던 연구들이 다시 진행되고 있다. 이에 본 논문에서는 널리 알려진 선형 모델 기반의 3D morphable face model을 제작하고 그 성능을 높이는 방법에 대한 구현 및 dense correspondence 문제를 해결하기 위한 방법을 제안한다.

키워드 : 얼굴 인식, 얼굴 모델, 3차원 얼굴

Abstract 2D face analysis has some limitations which are pose and illumination sensitive. For these reasons, even if many researchers try to study in the 3D face analysis and processing, because of the low computing performance and the absence of a high-speed 3D scanner then a lot of research is not being able to proceed. But, due to improving of the computing performance in these days, the advanced 3D face research was now underway. In this paper, we propose the method of building a 3D face model which deal successfully with dense correspondence problem.

Key words : Face analysis, Face model, 3D face, Morphable model

1. 서론

얼굴 검출, 인식 및 분석은 컴퓨터 비전과 패턴 인식 분야에서 매우 중요하며 흥미롭고 어려운 문제 중의 하나이다[1]. 실제로 얼굴 인식 및 분석을 통한 인증 시스템이나 출입통제 시스템은 많은 연구가 진행되고 있으며, 상용화를 위한 기술개발도 많이 이루어지고 있다. 하지만 이런 대부분의 연구는 2D 기반으로 진행되고 있으며 이 때문에 2D가 갖는 많은 문제점을 포함하고 있다[2].

먼저 포즈에 많은 영향을 받으며, 조명 모델링이 어렵다는 문제점이 있다. 물론 2D 기반의 포즈에 강인한 얼굴 분석 및 처리 방법들이 많이 연구되고 개발되고 있으며 조명에 강인한 알고리즘 개발에 대한 연구도 많이 진행되고 있다[3]. 하지만 입력영상이 2D라는 한계로 인하여 큰 발전이 없다는 것이다. 때문에 많은 연구자들은 3D기반의 얼굴 분석 및 처리 방법에 많은 관심을 가져왔으며, 많은 연구를 진행해왔다.

3D 기반의 얼굴 분석 및 처리 알고리즘은 시스템 성능의 한계, 3D 데이터 수집의 어려움 및 3D 스캐너 성능의 문제 등으로 인하여 많은 연구가 진행되지 못했다. 하지만 CPU의 처리속도가 매우 빨라짐으로 인해서 3D 기반의 얼굴 연구가 다시 진행되고 있으며, 그 성능 또한 좋아지고 있다[4,5].

본 논문에서는 3D 기반의 얼굴 처리 알고리즘을 위한 반자동적인 대응점을 찾아 3D 얼굴 모델을 제작하는 방법에 대해 제안했다. 3D 모델 중에서 선형 모델의 한 종류인 3D morphable face model을 제작하였으며, 이를 제작하기 위해 주성분분석을 이용하였다. 또한, 3D morphable face model의 성능을 향상시키기 위한 bootstrapping[6] 알고리즘을 적용하였다.

기존 논문에서는 3D morphable face model을 만들 때 얼굴 DB의 데이터를 정렬하기 위해서 optical flow와 피팅 알고리즘 및 bootstrapping을 사용하였다[5,6]. 그러나 이러한 방법을 사용하기 위해서는 피팅 알고리

· 이 논문은 산업자본부 지원으로 수행하는 21세기 프론티어 연구개발사업(인간기능 생활지원 지능로봇 기술개발사업)의 일환으로 수행되었습니다. 또한 본 논문은 POSTECH과 KT가 수행하는 KT VR Lab@University 공동연구 결과물의 일부입니다.

· 이 논문은 제34회 추계학술대회에서 '반자동적인 대응점 찾기를 이용한 3차원 얼굴 모델 생성'의 제목으로 발표된 논문을 확장한 것임

† 학생회원 : 포항공과대학교 컴퓨터공학과
ihchoi@postech.ac.kr

** 정 회원 : KT 미래기술연구소 책임연구원
sunny@kt.co.kr

*** 정 회원 : 포항공과대학교 컴퓨터공학과 교수
dkim@postech.ac.kr

논문접수 : 2007년 12월 14일

심사완료 : 2008년 7월 3일

Copyright©2008 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 컴퓨팅의 실제 및 레터 제14권 제7호(2008.10)

증과 얼굴 모델이 존재해야만 한다는 문제점이 있다. 이는 초기 얼굴 모델이 존재할 경우에만 사용 가능한 방법으로 본 논문에서는 반자동적인 대응점을 이용한 초기 얼굴 모델을 생성하는 방법을 제안하였다. 본 논문에서 제안한 방법으로 초기의 3D 얼굴 모델을 생성하고 생성된 얼굴 모델을 기존의 방법에 적용하면 더욱 섬세한 3D 얼굴 모델을 생성할 수 있다.

논문은 반자동적으로 대응점을 찾는 단계, 3D 얼굴 모델을 생성하는 단계, 3D 얼굴 모델의 성능을 향상하는 단계, 이러한 일련의 과정을 통한 3D 얼굴 모델 예제의 설명으로 구성되어 있다.

2. Finding Correspondence

2.1 Correspondence and Database

다양한 얼굴을 표현할 수 있는 변형 가능한 3D 얼굴 모델을 생성하기 위해서는 얼굴들을 충분히 표현할 수 있을 정도의 DB가 있어야 한다. 즉, DB를 가지고 여러 특징들을 표현할 수 있는 3D 얼굴 모델을 생성할 것이다. 현재 Beijing University of Technology의 BJUT-3D Face Database[7]를 3D 얼굴 DB로 사용 중에 있다. BJUT-3D Face Database는 중국인 남녀 각각 250명에 대한 3D 형태 정보, 색상 정보를 포함하고 있다.

다양한 얼굴에 대해 DB를 수집하고나서 각각의 얼굴 데이터에 대해서 어떠한 관련성이 있는지를 알아내야 한다. 각각의 얼굴 데이터에서 눈의 위치는 어떠한지, 코의 위치는 어디인지, 다른 부분들의 위치가 어디인지를 알아내야 3D 얼굴 모델을 생성하기 위한 데이터로써의 가치가 있다.

Correspondence는 수동으로 또는 자동으로 찾아낼 수 있다. 먼저, 수동으로 찾는다는 것은 얼굴 data의 각 특징점이 될만한 위치를 수동으로 지정하는 것이다. 그림 1은 AAM(Active Appearance Model)에서 수동으로 특징점을 지정하는 것을 보여준다.

이렇게 수동으로 특징점의 위치를 지정하는 것은 효율적이면서도 빠른 방법이기도 하지만 모든 점들간의 correspondence를 찾는 것이 아니고 몇몇 지정된 위치의 특징점에 대해서만 correspondence를 찾는다는 단점



그림 1 수동으로 correspondence를 찾는 예제

이 있다. 또한, DB의 양이 늘어날수록 수동으로 해야 할 일도 늘어난다. 물론 모든 특징점을 자동으로 찾는다는 것은 매우 어려운 일이다. 그리하여 본 논문에서는 반 자동적인 방법을 통하여 correspondence를 찾는 방법을 제안했다.

이는 기존의 얼굴 모델을 생성하는 알고리즘이 가지고 있는 초기 모델이 있어야 하는 점을 해결하기 위한 방법으로 사용되었으며, 이를 이용하면 빠른 시간에 초기 모델을 생성할 수 있다.

반 자동적인 방법으로 특징점들의 위치를 찾아내는 것을 위해서 1. 먼저 얼굴 데이터들의 몇몇 특징점 위치를 지정하고 2. 각 특징점들의 평균 위치로 얼굴 데이터를 정렬한다. 이때 평균 형태와 각 데이터들간의 차이를 이용하여 backward warping을 적용한다. 3. Warping된 데이터들을 초기값으로 지정하여 optical flow를 수행하면서 조밀한 correspondence를 찾아나간다. 이러한 방법은 수동으로 특징점들의 위치를 찾는 비효율적인 문제를 해결함과 동시에 완전 자동으로 특징점을 찾는 어려운 문제점을 해결할 수 있다.

2.2 Using optical flow from alignment data

Optical flow[8]는 어떠한 영상 시퀀스에서 각 프레임마다 영상이 변화해가는 정도를 각각의 픽셀에서 flow vector로 나타낸 것을 말한다. 이는 픽셀단위의 correspondence를 찾을 때 매우 유용하게 사용할 수 있다. 하지만 우리가 사용하려는 DB는 영상 시퀀스 형태로 존재하는 것이 아니므로 기존의 optical flow를 그대로 사용할 수가 없다. 따라서 각각 다른 영상에 대해서 수행할 수 있는 optical flow를 구현하였다.

먼저 optical flow를 적용할 영상을 생성해야 한다. 3D 얼굴 데이터들은 (x,y,z) 3차원으로 표현된 형태와 각 (x,y,z) 점에 대한 (r,g,b) 색상으로 구성되어 있다. (x,y,z) 3D 데이터에서는 일반적으로 optical flow를 적용하기가 쉽지 않으므로 좌표계 변경을 통하여 (u,v,rho)의 투영 실린더 좌표로 변형을 해야 한다.

먼저 Cartesian 좌표계를 극 좌표계로 변경을 해줘야 하는데, 이것은 그림 2에서 나타내는 좌표계 변환 공식

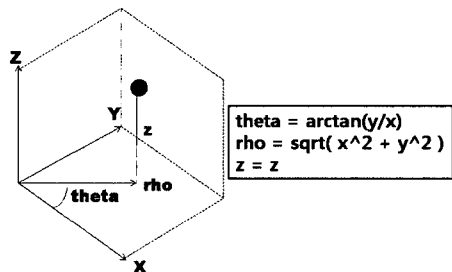


그림 2 3차원에서의 극 좌표계 변환

을 통하여 변경할 수 있다.

이렇게 변환한 극 좌표계를 식 (1)에 의하여 (u,v,rho)로 변환을 한다.

$$(u, v, \rho) = \left(\frac{W}{2\pi} \phi, \frac{H}{2} - ky, \rho \right) \quad (1)$$

여기서 W, H는 투영 영상의 폭과 높이를 나타내며 k는 높이에 대한 형태 파라미터이다. 본 논문에서는 400×400, 200×200 등 여러 크기의 투영 영상을 사용하였다. 각 투영 영상에 대해 형태에 대한 정보로 형태의 법선 벡터를 이용하였으며, 색상에 대한 정보는 색상의 미분 값을 이용하였다.

이렇게 변환한 영상들에서 optical flow를 통하여 flow vector를 구해야 하는데, 영상을 작게 만들수록 서로 다른 얼굴 영상들이 비슷해 보이기 때문에 영상들을 그대로 사용하지 않고 가우시안 피라미드를 통해서 여러 크기의 영상으로 변환 후 optical flow를 적용했다. 먼저 최상위 레벨의 가우시안 피라미드에서의 두 영상(가장 작은 영상)간의 optical flow를 구하고 그 flow vector를 하위의 가우시안 피라미드에서 optical flow를 구할 때 초기값으로 사용한다. 이러한 방법을 통하여 시퀀스 영상이 아닌 영상들간의 correspondence를 기존의 optical flow보다 더 잘 찾을 수 있다.

영상들의 정렬을 위한 optical flow 알고리즘은 그림 3과 같다. Optical flow를 통하여 각각의 얼굴 데이터들에 대한 correspondence를 찾을 수 있고, 이를 통하여 모든 얼굴 데이터를 같은 형태로 변형하여 3D 얼굴 모델을 생성할 수 있다.

그림 4는 기준 영상으로 설정한 데이터와 optical flow를 구하려는 목표 영상, 구한 optical flow를 통하여 변형한 영상을 보여준다. Optical flow를 적용하기 전의 영상(original)은 기준 영상과는 많이 다른 형태를

```

input: I1, I2
output: flow field s
begin
J1, J2 ← init feature of I1, I2 (normal, gradient)
J1', J2' with i = lmax ... 0 ← init pyramids of J1, J2
slmax ← 0
for l ← lmax to 0 do
    if l < lmax then
        ψ ← expand sl+1
    else
        ψ ← 0
    K ← warp J2' with ψ
    ρ ← flow from K to J1'
    sl ← smooth ρ + ψ
end for
s ← s0
end
    
```

그림 3 가우시안 피라미드를 이용한 optical flow

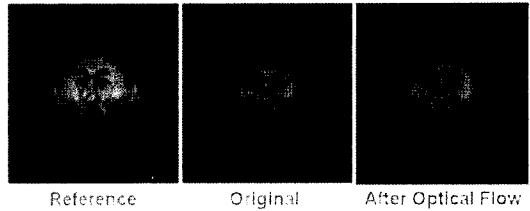


그림 4 Optical flow를 통한 정렬 결과



그림 5 형태 및 색상에 대한 평균 얼굴

가지고 있지만, optical flow를 통하여 구한 flow vector로 warping한 영상(After optical flow)은 기준 영상과 거의 비슷한 형태를 가지는 것을 볼 수 있다.

그리하여 그림 5와 같은 평균 얼굴을 만들 수 있었으며, 이 평균 얼굴은 3D 얼굴 모델 생성 단계, 성능 향상 단계에서 사용된다.

2.3 Semi-automatic correspondence

Optical flow를 이용하여 완전 자동으로 correspondence를 찾을 수 있다면 그 방법을 사용할 수 있겠지만, 초기 모델을 위해서는 몇몇의 특징점들을 수동으로 찾는 것이 매우 효율적이다. 그렇게 찾은 특징점을 기반으로 초기 3D 얼굴 모델을 생성한다.

특징점을 수동으로 지정하면 그림 1과 같은 형태로 나타난다. 이렇게 초기 데이터에 대해서 수동으로 특징점을 찾아내고 각 데이터들의 평균을 구한 후 평균 얼굴과 각 데이터들간의 차이를 이용하여 backward warping을 이용한 초기 모델 생성을 위한 정렬작업을 수행한다.

3. Building 3D Morphable Face Model

3.1 Make shape parameter and texture parameter

Optical flow로 기준 데이터와 각 데이터간의 flow vector를 구할 수 있으며, 구한 flow vector로 warping을 하여 얼굴의 전체적인 모양을 맞춰준 후 모든 데이터를 벡터로 변환하여 행렬로 표현한 후 주성분분석을 통하여 선형 모델을 생성한다. 여기서 주성분분석에 사용되는 행렬 분해 방법은 SVD(Singular value decomposition)을 사용했다. SVD는 행렬 분해 방법 중에서 가장 폭넓게 사용되는 방법이며 성능 또한 가능 좋은 것으로 알려져 있다.

주성분분석을 통하여 모델을 만드는 과정은 그림 6과

Subtract the mean from each data vector and put them into the matrix D as

$$D = ((X_1 - \bar{X}) | \dots | (X_s - \bar{X}))$$

The covariance matrix of matrix D can be written as

$$S = \frac{1}{s} DD^T$$

Let T be the s*s matrix as

$$T = \frac{1}{s} D^T D$$

Let e_i be the s eigenvectors of T with corresponding eigenvalues. $D e_i$ is the basis of data.

그림 6 주성분분석을 이용한 재생성

같다. 여기서 D는 평균을 뺀 데이터를 말하며, S는 D에 대한 공분산 행렬, T는 D의 전치 행렬에 대한 공분산 행렬이다.

그림 6의 알고리즘은 일반적인 주성분분석과 조금 다른 점이 있는데, 그것은 데이터의 공분산 행렬을 구할 때 전치에 대한 공분산 행렬을 구한다는데 있다. 이것은 데이터의 크기가 데이터의 개수보다 훨씬 많을 때 사용하는 방법이며 데이터의 공분산 행렬에 대한 고유벡터와 데이터와 전치 행렬에 대한 공분산 행렬의 고유벡터의 곱이 같기 때문에 이 방법을 사용할 수가 있다. 이렇게 주성분분석으로 선형 모델을 구하면 다음과 같은 모델이 완성된다.

$$S = S_0 + \sum \alpha_i S_i \quad (2)$$

$$T = T_0 + \sum \beta_i T_i \quad (3)$$

여기서 S는 형태, S_0 는 평균 형태, S_i 는 형태에 대한 기저들, α 는 형태의 기저에 대한 각각의 계수, T는 색상, T_0 는 평균 색상, T_i 는 색상에 대한 기저들, β 는 색상의 기저에 대한 각각의 계수이다. 즉, α , β 가 morphable model의 파라미터이다. α , β 를 조절하면 3D 얼굴 모델의 형태와 색상이 변화한다. 이를 통하여 입력 데이터에 피팅할 수가 있다.

3.2 Make triangle list and UV map

3D morphable face model은 최종적으로 3D 얼굴의 피팅 알고리즘에 사용된다. 추후에 사용할 피팅 알고리즘은 (x,y,z)의 Cartesian 좌표와 (u,v,rho)로 이루어진 투영 실린더 좌표 사이의 상호 변환 및 역 변환을 통하여 모델에 피팅하는 과정을 수행한다[9]. 이를 위해서는 triangle list와 평균 영상에 대한 정보가 있어야 한다. 이 정보들은 3D 얼굴 피팅 알고리즘에 사용된다[5,9].

Delaunay 방법은 triangle list를 만들 때 가장 많이 사용된다. 하지만 Delaunay 방법은 모든 외곽 점들을 연

```

input: uvMap, uvList
output: triangle list
for l ← 1 to size of uvList
  foreach 8 direction in pixel,
    search neighbor 2 pixels for pixel, in uvMap
    if neighbor 2 pixels of pixel, in uvMap is non-empty
      add 3 pixels to triangle list by
        clock-wise direction
    end foreach
  end for
end

```

그림 7 Triangle list 생성 알고리즘

결하기 때문에 만들어진 triangle 중에서 거의 직선의 형태를 가지는 것도 있다. 이는 triangle의 방향을 계산할 때 문제가 되므로 우리는 각 점들을 8방향으로 탐색하면서 인접한 점들에 대하여 시계 방향의 triangle을 만들었다. 이는 직선의 형태를 가지는 triangle을 생성하지 않기 때문에 피팅 알고리즘에서 방향성을 계산할 때 문제가 발생하지 않는다. 제안한 방법은 그림 7에 설명되어 있다. UV map은 색상들의 평균을 통하여 구한 평균 얼굴을 그림 5와 같은 형태의 rgb 영상으로 만들었다.

4. Improving 3D Morphable Model

앞에서 언급한 과정들을 통하여 3D morphable face model을 만들 수 있었다. 하지만 optical flow를 통하여 정확한 correspondence를 찾는 것은 매우 어려운 일이므로 dense correspondence의 성능을 높여야 한다.

Optical flow는 영상들의 시퀀스 중에서 각 프레임간의 점 사이의 flow를 찾는 것이다. 하지만 optical flow도 모든 영상들간의 correspondence를 잘 찾아주지 못하므로 bootstrapping[6]이라는 점진적인 성능 향상 알고리즘을 통하여 매 반복 별로 기준 영상을 업데이트한다. 업데이트된 기준 영상을 통하여 모델을 생성하고 모델을 각각의 얼굴 데이터에 피팅을 하여 피팅된 모델과 각 얼굴 데이터 간의 optical flow를 구한다. 이 방법을 이용하여 반자동적으로 만든 초기 얼굴 모델의 성능을 점진적으로 향상시킬 수 있다.

5. 실험 결과

언급한 방법들을 이용하여 3D 얼굴 모델을 그림 8과 같은 형태로 만들었다. 그림 8은 생성된 얼굴 모델과 그 얼굴 모델의 파라미터를 조절하여 새롭게 얻은 얼굴에 대한 예제를 보여준다. 첫 번째 그림은 얼굴 모델을 보여주며, 두 번째 그림은 얼굴 모델의 파라미터를 변형하여 얻은 새로운 얼굴을 보여준다.

생성된 모델은 약 20,000개의 버텍스로 구성되어 있으며, 3D 형태와 색상에 대한 각각 100개의 파라미터로



그림 8 생성된 얼굴 모델 및 변형된 얼굴의 예제

구성되어 있다.

그림 9는 하나의 얼굴이 어떠한 선형 조합으로 구성되어 있는지를 보여준다. 이때 얼굴 모델은 각 기저(오른쪽에서 4개)들의 계수를 조절하는 것을 통하여 가장 왼쪽에 있는 얼굴을 만들어낸다.

6. 결 론

본 논문은 3D 얼굴 처리 및 분석 알고리즘을 위한 반자동적인 대응점을 이용한 3D 얼굴 모델 제작 및 그 성능을 향상시키기 위한 알고리즘에 대해서 제안하였다. 3D 얼굴 연구는 과거에는 시스템의 제약 때문에 많은 연구가 진행 되지 못하였지만, 컴퓨팅 파워의 증가와 고속 스캐너의 등장으로 인해 다시 많은 연구가 진행되고 있다.

본 논문에서는 3D 모델로 많이 알려진 선형 모델 기반의 3D morphable face model을 주성분분석을 통한 재생성을 이용하여 만드는 방법과 correspondence를 optical flow를 통하여 반 자동적으로 찾는 방법, correspondence를 향상시켜 3D morphable face model의 성능을 향상시키는 방법에 대해 구현을 하였으며 반 자동적인 방법으로 optical flow의 초기위치를 정하는 방법에 대해서 제안하였다.

앞으로 3D 얼굴 분석 및 처리 알고리즘은 더욱 발전할 것이며 그 응용 또한 폭넓게 이용될 것이다. 이를 위해 3D 얼굴 모델의 생성은 매우 중요하며 본 논문에서 제안한 방법을 이용하면 기존의 방법이 가지고 있는 초기 얼굴 모델이 있어야 하며, 피팅 알고리즘도 존재해야 한다는 문제점을 해결할 수 있다. 또한 기존의 방법에 적용할 수 있는 초기 3D 얼굴 모델을 빠르고 효율적으로 생성할 수 있다.

참 고 문 헌

- [1] X. Tan, S. Chen, Z. Zhou and F. Zhang, "Face recognition from a single image per person: A survey," Pattern Recognition, Vol.39, No.9, pp. 1725-1745, 2006.
- [2] D. Beymer, "Face recognition under varying pose," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 756-761, 1994.
- [3] P. N. Belhumeur and D. J. Kriegman, "What is the set of Images of an Object Under All possible Lighting Conditions?," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 270-277, 1996.
- [4] M. J. Jones and T. Poggio, "Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes," International Journal of Computer Vision, Vol.2, No.29, pp. 107-131, 1998.
- [5] V. Blanz and T. Vetter, "A Morphable Model For The Synthesis Of 3D Faces," Proc. of the 26th annual Conf. Computer graphics and interactive techniques, pp. 187-194, 1999.
- [6] T. Vetter, M.J. Jones and T. Poggio, "A bootstrapping algorithm for learning linear models of object classes," Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, pp. 40-46, 1997.
- [7] BJUT, "BJUT- 3D Face Database," Beijing University of Technology.
- [8] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow," Artificial Intelligence, Vol.17, pp. 185-203, 1981.
- [9] S. Romdhani and T. Vetter, "Efficient, Robust and Accurate Fitting of a 3D Morphable Model," Proc. of the IEEE International Conf. on Computer Vision, pp. 59-66, 2003.

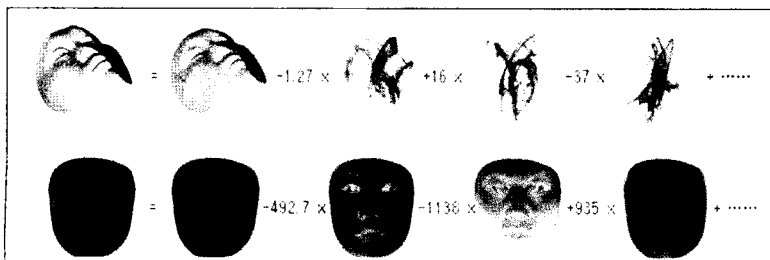


그림 9 얼굴 모델을 이용한 재생성 예제