

얼굴의 움직임 추적에 따른 3차원 얼굴 합성 및 애니메이션

(3D Facial Synthesis and Animation for Facial Motion
Estimation)

박도영[†] 심연숙^{**} 변혜란^{***}

(Do Young Park) (Youn Sook Shim) (Hyeran Byun)

요약 본 논문에서는 2차원 얼굴 영상의 움직임을 추출하여 3차원 얼굴 모델에 합성하는 방법을 연구하였다. 본 논문은 동영상에서의 움직임을 추정하기 위하여 광류를 기반으로 한 추정방법을 이용하였다. 2차원 동영상에서 얼굴요소 및 얼굴의 움직임을 추정하기 위해 인접한 두 영상으로부터 계산된 광류를 가장 잘 고려하는 매개변수화된 움직임 벡터들을 추출한다. 그리고 나서, 이를 소수의 매개변수들의 조합으로 만들어 얼굴의 움직임에 대한 정보를 묘사할 수 있게 하였다. 매개변수화 된 움직임 벡터는 눈 영역, 입술과 눈썹 영역, 그리고 얼굴영역을 위한 서로 다른 세 종류의 움직임을 위하여 사용하였다. 이를 얼굴 모델의 움직임을 합성할 수 있는 단위행위(Action Unit)와 결합하여 2차원 동영상에서의 얼굴 움직임을 3차원으로 합성한 결과를 얻을 수 있다.

Abstract In this paper, we suggest the method of 3D facial synthesis using the motion of 2D facial images. We use the optical flow-based method for estimation of motion. We extract parameterized motion vectors using optical flow between two adjacent image sequences in order to estimate the facial features and the facial motion in 2D image sequences. Then, we combine parameters of the parameterized motion vectors and estimate facial motion information. We use the parameterized vector model according to the facial features. Our motion vector models are eye area, lip-eyebrow area, and face area. Combining 2D facial motion information with 3D facial model action unit, we synthesize the 3D facial model.

1. 서론

동영상에서의 움직임 추적에 대한 연구는 오랫동안 컴퓨터 시각분야에서 연구되어왔다. 특히 최근에는 컴퓨터 사용자의 얼굴 및 얼굴 요소의 움직임을 추적하는 연구들에 대하여 컴퓨터 시각분야 뿐 아니라, 인간과 컴퓨터간의 상호작용 (Human and Computer Interaction) 등 여러 분야에서 그 관심이 증가되고 있다. 얼

굴은 인간의 신체부위 중 가장 쉽게 개개인을 구분할 수 있고, 감정과 정서 등의 내적 상태를 잘 이해할 수 있도록 해주는 중요한 의사소통의 수단으로 여겨지고 있어, 인간과 컴퓨터간의 의사 전달에 있어서 효율적으로 인간의 감정을 인식하고 전달하여 대화할 수 있도록 하는 역할을 한다. 기존에는 사람과 컴퓨터간의 의사 전달 수단으로 특정 입출력 장치를 사용하였지만, 최근에는 인간의 음성이나 얼굴 등을 이용하여 자연스럽게 지능적인 인터페이스를 구현하고자 하는 연구들이 진행되고 있다. 이러한 인터페이스들을 제공하기 위한 얼굴 및 얼굴 요소의 움직임 추적에 대한 연구가 활발히 진행되고 있다. 동영상에서 얼굴 및 얼굴 요소의 움직임을 이해하기 위해서는 얼굴의 움직임 정보를 추출하여 해석해야 한다.

본 논문에서는 2차원 동영상에서 움직임을 추출하기

[†] 비회원 : 육군사관학교 전산학과 교수
dypark@kna.ac.kr

^{**} 비회원 : 연세대학교 인지과학 협동과정
heea@csai.yonsei.ac.kr

^{***} 종신회원 : 연세대학교 컴퓨터학과 교수
hrbyun@csai.yonsei.ac.kr

논문접수 : 1999년 3월 15일

심사완료 : 2000년 4월 10일

위하여 광류[9]를 이용하여 얼굴요소 및 얼굴의 움직임을 측정하였고, 이를 3차원 얼굴 모델에 적용하여 움직임을 표현하였다. 본 논문에서 제안하고 있는 시스템은 [그림 1]과 같이 2차원 동영상에서 추출된 얼굴 및 얼굴 요소의 움직임 측정값을 3차원 얼굴 모델에 적용하여 움직임을 합성시키는 과정으로 이루어진다.

본 논문에서는 2차원의 인접한 두 영상으로부터 계산된 광류를 가장 잘 고려하는 매개변수화 된 움직임 벡터들을 추출하고, 이를 소수의 매개변수들의 조합으로 만들어 얼굴의 움직임에 대한 정보를 묘사할 수 있게 하였다. 매개변수화 된 움직임 벡터는 얼굴 및 얼굴 요소의 특징에 따라 다른 벡터 모델을 사용한다. 본 연구를 위해서 사용된 움직임 벡터는 눈 영역, 입술과 눈썹 영역, 그리고 얼굴영역을 위한 세 종류의 모델을 사용하였다[25]. 2차원 동영상에서 매개변수화 된 움직임 벡터는 매 프레임마다 갱신되어 각 프레임에서 얼굴 및 얼굴 요소의 위치를 예측하고, 갱신된 벡터를 매개변수들의 조합으로 만들어 추출된 움직임에 대한 정보를 3차원 얼굴모델에 전달하여 합성할 수 있도록 하였다. 본 연구를 위해 사용된 3차원 얼굴 모델은 표준 얼굴형상에 대한 모델이 특정인의 얼굴에 정합되어 특정인의 텍스처가 입혀진 모델을 사용하였다. 그리고 이 모델에 P. Ekman과 W. V. Friesen이 제안한 FACS(Facial Action Coding System)이론에 근거하여 46개의 단위행위에 대한 움직임을 적용하였다. 2차원 영상에서 추출한 움직임에 대한 정보가 3차원 얼굴 모델의 단위행위(Action Unit)와 연결되어 동영상에서 얼굴의 움직임을 3차원 얼굴 모델이 합성할 수 있게 하였다.

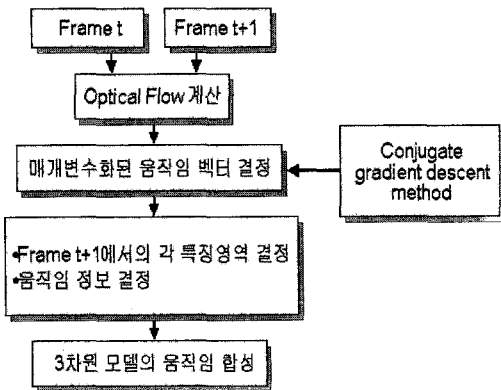


그림 1 시스템 구성도

본 논문의 구성은 다음과 같다. 2장에서는 동영상 추

적에 대한 기존연구들에 대하여 간단히 살펴보고, 3장에서는 본 논문에서 사용한 얼굴 모델 및 합성 방법에 대하여 소개하였다. 그리고 4장에서는 본 논문에서 제안하는 매개변수화 된 움직임 벡터 추출과정을 자세히 설명하였다. 5장에서는 실험결과 및 분석을, 마지막으로 6장에서는 본 연구에 대한 결론과 향후 연구방향을 서술한다.

2. 관련 연구

2.1 동영상 추적 관련 연구

동영상에서의 움직임 추정 방법은 크게 세 가지로 분류할 수 있는데, 블록기반(block-based) 추정 방법, 화소기반(pixel-based) 추정방법, 광류기반(optical flow-based) 추정방법 등이 있다. 블록기반 움직임 추정방법은 H.261, MPEG-1, MPEG-2 등과 같은 영상 부호화 표준안에 적용되어 널리 사용되고 있지만, 회전과 확대 및 축소를 보상하지 못하기 때문에 눈에 매우 거슬리는 블록화 잡음을 초래한다는 단점이 있다. 그리고 화소기반 움직임 추정 방법은 객체의 움직임을 정확하게 표현할 수 있다는 장점이 있지만, 모든 화소의 움직임 벡터를 전송해야 한다는 어려움과 각각의 많은 화소에 대하여 실제 움직임을 찾아야 한다는 단점이 있다. 한편, 광류 기반(optical flow-based) 추정방법은 시간에 따라 변화하는 영상의 강도패턴으로부터 2차원적 벡터분포를 계산하여 움직임을 추정하는 방법으로 지금까지 제안되어 온 많은 동영상 해석 방법들의 근간이 되고 있다. 이러한 광류 기반 방법은 계산 속도가 오래 걸린다는 단점이 있지만, 비교적 세밀한 움직임까지 추정할 수 있어 얼굴과 같이 미세한 부분의 움직임을 추정하는데 많이 사용되고 있다.

얼굴 추적은 회전변환, 평행 변환 등을 수행하는 강직성을 가진 물체의 움직임 추적과 관련된다. 반면, 얼굴 요소 추적은 얼굴의 해부학적 구조상 각 얼굴 요소가 움직일 수 있는 범위 내에서 다양한 비강직성의 변형을 추적하는 것과 관련된다. 얼굴 요소 추적에는 크게 두 가지의 방법이 제시되어 있다. 첫째는, 얼굴 요소의 형태를 정확하게 묘사하면서 추적을 시도하는 경계선 추적으로서 입의 경계선 추적을 시도한 [1][2]등의 연구가 대표적이다. 둘째는 얼굴 요소의 세부적인 형태에는 큰 관심을 두지 않으면서 얼굴 요소를 포함하는 영역을 추적하는 방법이다[3]. 본 논문에서는 두 번째의 방법을 이용하여 얼굴 요소 영역 추적을 시도한다.

얼굴 요소 영역의 움직임 없이 얼굴영역 추적을 시도한 접근 방법은 [4]에 제시되어 있다. 이 연구에서는 몇

개의 특징점 들의 추적으로 움직임을 나타내는 방정식을 해결함으로써 얼굴의 3차원적 움직임 파라미터가 계산된다. Essa[5]는 얼굴 요소 추적을 위해 3차원 모델에 기반한 접근 방식(3-D model based approach)을 제시하였다. 이는 3차원 얼굴모델의 메쉬(mesh)가 얼굴에 대응한다고 가정하여 얼굴 표정 연출동안 메쉬 점들의 위치를 재배치하기 위해 광류를 계산하는 방법이다. 이와 관련되어 템플릿 기반 방법을 사용한 연구에서는 각 얼굴 요소에 가능한 기저(basis) 템플릿을 데이터베이스로 구축한 후 현재 영상에서 얼굴 요소 형태를 기저 템플릿들을 인터플레이션함으로써 표현하게 하였다.

본 연구에서는 매개변수화 된 움직임 모델(벡터)을 이용하여 광류에 기반한 접근방식을 사용한다. 영상의 움직임 측정에 대한 매개변수화 된 모델은 강직성(rigid)을 가진 영상에서 움직임을 측정하는데 널리 쓰이고 있으며, 영상의 움직임 벡터가 저(底)차원의 다항식에 의해 표시할 수 있다고 가정하여 만들어진 것이다[6]. 그러나, 이러한 매개변수화 된 움직임 벡터는 비강직성(non-rigid)의 움직임을 측정하는 데는 유용하지 못하다. 하지만, 영상의 전체영역을 대상으로 하지 않고 일부영역만을 대상으로 모델링 한다면, 그 영역에 대한 비강직성의 움직임을 묘사할 수 있을 뿐 아니라, 매개변수들의 적절한 조합에 의해 움직임을 해석할 수 있는 정보까지 제공해 주게 된다. 따라서, 매개변수화 된 움직임 모델(벡터)은 매개변수들의 소수의 조합으로 얼굴 및 얼굴 요소의 강직한 움직임뿐만 아니라 비강직성을 가지는 움직임도 묘사해 낼 수 있게 된다.

본 연구에서는 한 영상에서 얼굴의 움직임을 묘사하기 위하여 얼굴의 대부분이 한 평면에 의해 모델링 될 수 있다고 가정하여 세 종류의 매개 변수화 된 움직임 벡터모델을 사용하였다. 눈 영역을 위한 간단한 어파인(affine) 모델, 입 영역과 눈썹 영역을 위해서는 웃는 표정을 연출할 경우와 같이 입의 곡률 이라든지 눈썹 영역의 아치(arch)형태를 고려하기 위한 곡률 매개변수가 추가된 어파인 모델, 마지막으로 얼굴 영역을 위한 평면(planar) 모델을 사용한다. 코 영역은 얼굴 표정 연출시 크게 움직이지 않는 점을 고려하여 추적 대상으로 삼지 않았다.

동영상의 매 프레임이 진행되면서 각 모델의 매개변수 값이 광류를 이용하여 측정이 되고 이 값에 따라 각 얼굴 요소 및 얼굴의 움직임 정보가 결정된다. 그러나, 본 연구에서는 첫 프레임에서 얼굴 요소 및 얼굴 영역을 찾는 것[7][8]은 고려하지 않았다. 다음 프레임에서 얼굴요소 및 얼굴의 위치를 예측하는 최적의 매개변수

화 된 움직임 벡터들을 찾기 위해 먼저 현재 프레임과 다음 프레임간의 광류가 계산된다. 본 연구에서 사용한 광류 계산 방법은 Ajit Singh[9]가 제안한 방법으로 두 단계의 과정을 이용한다. 이 방법은 기존의 광류 계산 방법과 달리 두 단계에 따라 광류가 계산되어지는데, 우선 첫 번째 단계에서는 시간의 변화에도 불구하고 두 영상간의 국부적인 이미지 속성이 유지된다는 성질을 이용하여 1차적인 광류가 계산된다. 그러나, 첫 번째 단계를 통해 계산된 광류는 다소 정확성이 떨어진다. 따라서 보다 정확한 광류를 계산하기 위한 두 번째 단계로 속도의 전이 과정을 거치게 된다. 즉, 주위의 속도 정보를 이용하여 속도벡터가 갱신되는 단계를 거치게 된다.

위의 두 과정을 거쳐 계산된 광류를 가장 잘 고려하는 매개변수화된 움직임 벡터들이 각 얼굴요소 및 얼굴의 움직임으로써 결정된 뒤, 얼굴요소 및 얼굴의 영역이 변형되고 다음프레임에서 최적의 매개변수화된 움직임 벡터를 찾는 과정의 시작 벡터로써 사용될 수 있다. 또한, 매 프레임에서 결정된 매개변수화된 움직임 벡터는 그 매개변수들을 적절하게 조합함으로써 각 영역에 대한 움직임 정보를 묘사할 수 있게 해 준다.

2.2 얼굴 모델링 및 애니메이션

얼굴 모델링 및 애니메이션에 관한 연구는 1970년대 F. I. Parke의 연구[19]가 최초의 것이었다. Parke는 그의 연구에서 머리를 매우 복잡한 폴리곤 형태로 나타내어 눈과 입을 열고 닫는 얼굴의 애니메이션을 구현하였다. 지속된 그의 연구는 그의 첫 매개변수화 된 얼굴 모델[20]을 완성하게 된다.

1974년과 1978년의 기간동안에는 3차원에 근거한 얼굴 애니메이션 개발이 이루어지지 않았다. 그러나 2차원적인 애니메이션 시스템의 개발은 코넬(Cornell)대학에서 지속되고 있었으며 이러한 시스템이 얼굴 애니메이션에 적용되었다.

1980년대에 들어서서, Platt[21]가 얼굴 근육에 의해 조절되는 얼굴 표정 모델을 제시하였다. 그후 Bergeron [22]가 제시한 연구는 얼굴 애니메이션에 대한 획기적인 사건이 되었는데 이것이 3차원적인 얼굴 표정과 립싱크(lip-synch)에 대한 첫 컴퓨터 애니메이션의 토대가 되었다. 1987년에는 Waters[23]가 얼굴 표정 애니메이션을 위한 새로운 근육 모델 접근방식을 발표하였고 이러한 접근으로 얼굴 근육을 조절함으로써 생성될 수 있는 다양한 얼굴 표정을 연출할 수 있게 되었다.

이후, 1990년대 사이버웨어 광학 레이저 스캐너의 발달은 얼굴 애니메이션을 위한 풍부한 데이터를 제공하

게 되었다. 이것으로 3차원 얼굴 애니메이션을 위한 수단으로 이미 등록되어 있는 얼굴 이미지 텍스처의 사용이 가능하게 되었다. 이와같은 진보된 이미지 프로세싱 기법과 스캐닝 기술로 새로운 스타일의 얼굴 애니메이션을 제공하게 되었으며, 이런 배경 안에서 1993년에는 개개의 얼굴 애니메이션이 물리적인 움직임 속성에 의해 결정되는 얼굴의 정규적인 표현으로 사상되는 기술[24]을 발표하게 되었다.

한편, 얼굴에 관하여 P. Ekman 등의 심리학자들은 얼굴 근육들의 긴장과 이완의 조합 결과로 나타나는 얼굴 표정이 어떤 근육들의 움직임이 합성되어 나타내어지는가에 대하여 연구를 행하였는데, P. Ekman과 W. V. Friesen이 1978년 제안한 FACS(Facial Action Coding System)[12]가 그 대표적인 연구이다.

본 논문에서는 KIST(Korea Institute of Science and Technology)에서 연구중이던 3차원 얼굴 모델을 사용하였으며 얼굴 움직임 애니메이션을 위해 위의 FACS를 coding하여 단위 움직임들의 합성으로 가능하게 하였다.

3. 얼굴 합성

3.1 얼굴모델 구성

본 논문에서 사용한 얼굴 모델은 KIST(Korea Institute of Science and Technology)에서 제작한 모델로, 총 469개의 꼭지점으로 이루어진 삼각형 패치(patch)로 구성된 3차원 형상모델이다. 이 모델은 얼굴의 움직임을 합성하는데 적합하도록 눈, 눈썹, 입 등의 부위는 비교적 세밀하게 구성하였으며, 적은 수의 메쉬로 움직임을 제어할 수 있도록 하였다. [그림2]는 3차원 표준형상모델을 보여주고 있다.



(a) wireframe모델 (b)폴리곤모델

그림 2 표준 형상 모델

이 표준형상모델을 이용하여 입력된 영상의 정면에

정합한 개인얼굴의 3차원 형상모델을 얻을 수 있다[10][11]. 이때, 표준형상모델은 3차원 어핀변환(affine transform)을 통하여 얼굴의 크기, 위치 등이 조절된다. 그리고 얼굴의 윤곽, 특징점 등과 같이 정밀하게 정합되어야 할 39개의 점을 지정하고, 그 이외의 정점들의 좌표는 특징점의 좌표를 이용하여 형상모델을 조정한다. 39개의 점들은 얼굴윤곽선을 나타내는 12개의 점과 얼굴의 특징점(눈썹, 눈, 코, 입 에 대하여 각각 6개, 8개, 5개, 8개)에 대한 27개의 점들로 구성된다. [그림3]은 표준형상모델을 이용하여 특정인의 얼굴을 3차원으로 모델링 한 결과를 보여주고 있다.



(a) wireframe모델 (b) 폴리곤 모델 (c) 텍스처 모델

그림 3 특정인의 얼굴에 정합된 모델

3.2 표정 단위 행위(Facial Action Unit) 구성

본 논문에서는 얼굴 및 얼굴의 움직임 추적에 따른 매개변수들을 이용하기 위하여 파라미터화된 모델 방법을 사용하였다. 얼굴의 움직임들을 파라미터화 하기 위하여 FACS(Facial Action Coding System)[12]를 coding하여 단위 움직임들의 합성으로 가능하게 하였다. P. Ekman 과 W. V. Friesen이 1978년 제안한 FACS(Facial Action Coding System)[12]는 얼굴 근육들의 긴장과 이완의 조합 결과로 나타나는 얼굴 표정에 대하여 어떤 근육들의 조합이 어떤 내적상태를 나타내는 표정과 연관이 있는가에 대한 연구 결과로, 인간의 내적상태를 나타내는 얼굴 표정을 위하여 모두 44개의 기본 단위 (AU, Action Unit)로 분류하여, 하나의 얼굴 표정은 각 AU의 조합으로 나타내도록 하였다.

본 논문에서는 2차원 동영상의 움직임에 대한 정보가 FACS를 이용한 단위행위(Action Unit)와 연결되어 동영상의 움직임이 3차원 얼굴모델에서 합성되도록 하였다. 표정 합성을 위한 표정 단위 행위(Facial Action Unit)가 적용되는 표정 합성 과정은 [그림4]와 같다. 몇 개의 단위행위(AU)들과 그 단위행위들의 강도(intensity)가 설정되면 해당되는 단위행위들이 주어진 강도만큼 3차원 얼굴 모델에 적용되어 기본 측정값이 결정된다. 그리고 나서, 이 값에 따라 중요 특징점의 위

치가 변화된다. 또한, 특징점을 제외한 나머지 점들이 특징점의 변화된 위치와 기본 측정값에 따라 보간(interpolation)이 이루어지면서 움직임에 대한 합성이 이루어진다. 이러한 방법은 비교적 간단하게 잘 정의된 단위행위들의 조합으로 다양한 표정들을 실시간으로 합성할 수 있다.

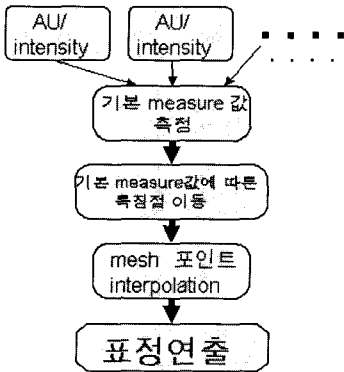


그림 4 얼굴 모델의 움직임 합성 과정

얼굴(눈, 입, 눈썹)의 상황을 정량화하기 위해 선정한 기본 측정값은 다음과 같다. 3차원 얼굴모델상에서 [그림5]의 점 A의 x좌표, y좌표, z좌표값이 각각 A.x, A.y, A.z로 표시된다고 하면,



그림 5 기본 측정값

$$\begin{aligned} \text{EYE_LID} &= (F.y - G.y + B.y - C.y) / 2 \\ \text{EYE_HEIGHT} &= (G.y - H.y + C.y - D.y) / 2 \\ \text{BROW_HEIGHT} &= (E.y - F.y + A.y - B.y) / 2 \\ \text{SX}[1] &= I.x - K.x, \quad \text{SY}[1] = I.y - L1.y, \quad \text{SZ}[1] = I.z - K.z \\ \text{SX}[2] &= M.x - I.x, \quad \text{SY}[2] = L2.y - O.y, \quad \text{SZ}[2] = O.z - K.z \\ \text{SY}[3] &= K.y - O.y, \quad \text{SZ}[3] = I.z - M.z \\ \text{SY}[4] &= M.y - O.y, \quad \text{SZ}[4] = O.z - M.z \end{aligned}$$

여기서 L1은 윗입술, L2는 아랫입술을 가리킨다. 다음의 [그림6]은 표정 단위 행위들의 조합으로 구성된 기본적인 6가지 표정을 합성한 결과이다.

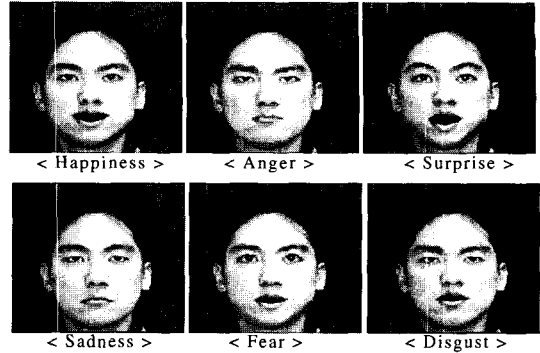


그림 6 6가지 기본 표정의 합성

4. 움직임 추적

본 연구에서는 얼굴 및 얼굴 요소의 움직임 추적을 위하여 동영상 프레임 t 와 그 다음 프레임 t+1을 비교하여 광류를 계산한다. 계산된 광류 정보는 얼굴 및 얼굴요소 영역의 움직임을 묘사하는 움직임 벡터를 결정하는 데 이용되며 이 과정에서 각 영역의 움직임을 최적으로 묘사하는 벡터값을 찾는 방법으로 'conjugate gradient descent'[13]의 방법을 적용하였다. 이것은 N 차원 선형시스템(A · x=b) 해결책을 위한 일반적인 방법이다. 이때, 매 프레임에서 각 영역을 나타내는 사각형 박스를 설정하는 데 처음 영상의 경우에는 인위적으로 각 박스를 설정하였다. 이후, 각 영역의 움직임을 묘사하는 최적의 매개변수화된 움직임 벡터를 이용해 매 프레임에서 각 영역에 대한 움직임 정보를 추출한다. 이러한 정보는 소수의 매개변수들의 조합으로서 그 정보값을 추출할 수 있다. 추출된 정보값은 3장에서 언급한 3차원 얼굴 모델의 단위 행위를 움직이는 힘으로 이용된다. 움직임 정보값을 이용함으로 3차원 얼굴 모델의 애니메이션을 구동시키는 46개의 단위행위 중 일부가 선택이 되고 선택된 단위행위는 추출된 정보량만큼 움직일 수 있다. 결론적으로, 2차원 동영상에서의 얼굴의 표정을 3차원 상에서 단위행위의 조합으로 움직임을 합성하는 시스템을 구성하고 있다.

4.1 매개변수화된 움직임 모델(벡터)

얼굴 요소 및 얼굴 영역에 대한 움직임 모델은 저(底)차원 다항식에 의해 표시될 수 있다[6]. [그림7]은 각

영역에 적용될 수 있는 움직임 벡터모델을 보여주고 있다.



그림 7 움직임 벡터 모델

앞에서 언급했듯이 영상의 움직임 측정을 위한 매개변수화된 모델은 저(底)차원의 다항식에 의해 표시할 수 있다[6]. 한편, 충분히 작은 영상 영역에서의 영역 움직임을 위한 모델은 6개의 파라미터로 구성된 어파인 벡터 모델로 충분히 가능하다[14]. 따라서, 특별한 속성이 없는 눈영역은 이러한 어파인 벡터 모델로 영역의 움직임을 충분히 묘사할 수 있다.

눈영역을 위한 어파인(affine) 움직임 모델은 다음과 같다.

$$u(x,y) = a_0 + a_1x + a_2y, \quad (1)$$

$$v(x,y) = a_3 + a_4x + a_5y. \quad (2)$$

여기서 매개변수 a_i 들은 상수들이며, $\mathbf{u}(x) = [u(x,y), v(x,y)]^T$ 는 2차원 영상의 한점 $\mathbf{x}=(x,y)$ 에서 수평방향 및 수직방향 움직임 벡터가 된다.

한편, 어파인 모델만으로는 사람의 얼굴 영역과 같은 부분의 움직임을 나타내기에는 적절하지 못하다. 왜냐하면, 얼굴 영역을 위해서는 보다 더 많은 정보를 표현할 수 있는 벡터 모델이 요구되는데, 이는 얼굴의 상하 좌우 회전이 고려되기 때문이다. 따라서, 본 연구를 위해서는 얼굴 영역이 하나의 평면에 의해 모델링 될 수 있다는 가정을 두고 두 개의 매개변수 p_0 와 p_1 이 덧붙여진 여덟 개의 매개변수를 지닌 평면(planar)벡터 모델을 사용하였다[15].

$$u(x,y) = a_0 + a_1x + a_2y + p_0x^2 + p_1xy, \quad (3)$$

$$v(x,y) = a_3 + a_4x + a_5y + p_0xy + p_1y^2. \quad (4)$$

얼굴영역이나 눈 영역의 움직임은 강직성을 가지는 움직임 속성을 보인다. 그러나, 입이나 눈썹과 같은 얼굴요소 영역은 강직성을 가지도록 모델링된 움직임 벡터 모델로는 그 움직임을 정확하게 묘사할 수가 없다. 이것은 입 영역이나, 눈썹 영역이 비강직성의 속성을 가지는 곡률에 대한 변형성을 가지고 있기 때문이다. 즉, 웃는 표정을 연출할 경우의 입의 곡률이라든지 눈썹 영

역의 아치(arch)형태를 고려하기 위한 곡률변수가 필요한 것이다. 따라서 (2)식에 곡률정보를 나타내는 파라미터 c 가 추가한 벡터 모델을 사용한다[15].

$$u(x,y) = a_0 + a_1x + a_2y, \quad (5)$$

$$v(x,y) = a_3 + a_4x + a_5y + cx^2. \quad (6)$$

이제, 제시된 매개변수화 된 움직임 벡터 모델에서 각각의 매수변수 혹은 그 조합은 다음과 같이 움직임에 대한 정량적인 정보를 가지고 있다[16].

$$\text{수평방향으로의 움직임} = a_0, \quad (7)$$

$$\text{수직방향으로의 움직임} = a_3, \quad (8)$$

$$\text{발산성(Divergence)} = a_1 + a_5 = (u_x + v_y), \quad (9)$$

$$\text{포임성(Curl)} = -a_2 + a_4 = -(u_y - v_x), \quad (10)$$

$$\text{변형성(Deformation)} = a_1 - a_5 = (u_x - v_y). \quad (11)$$

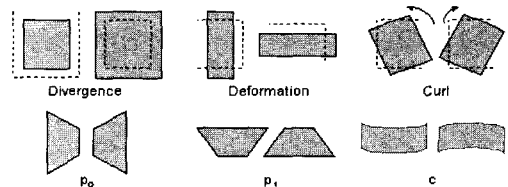


그림 8 움직임 정보에 따른 영역의 변형

여기서 밑첨자 x 와 y 는 영상에서 구해진 움직임 속도에 대한 편미분 연산자를 의미한다. 이를 도시한 것이 [그림8]이다. [그림8]에서 점선으로 둘러진 영역이 변형된 영역을 의미하며, 각 정보들의 왼쪽편은 그 값이 줄어든 경우이며, 오른쪽 편은 커질 경우를 보여주고 있다.

4.2 광류 (Optical flow) 계산

광류를 계산하는 방법은 크게 특징을 기반으로 한 방법과 밝기 경사를 기반으로 한 방법으로 나누어지는데, 이 중 밝기 경사를 기반으로 한 세 가지의 접근방식은 영상밝기의 미분값에 기반한(gradient-based)접근방식, 연속된 두 영상의 국부적인 영상밝기 분포가 유지된다는 가정을 이용한(correlation-based)접근방식, 시 공간 상에서 주파수성질을 이용한(spatio-temporal energy based)접근방식 등이 있다[9].

본 논문에서는 이 세 가지의 접근방식을 종합하였으며 두 단계의 과정을 거쳐 광류를 계산하고 있다. 첫 번째 단계에서는 시간 축에서 어떤 영상의 속성이 보존될 것이라는 가정을 두고 1차적인 속도 벡터필드를 찾는

과정이다. 그러나, 이러한 보존 정보는 실제 영상 어디에서나 그릇된 정보를 많이 포함하고 있다. 이미지의 어떤 영역에 있어서는 그것은 부분적인 정보일 수 있고 경우에 따라서는 전혀 정보가 아닐 수 있기 때문이다. 따라서 두 번째 단계에서는 정확한 광류를 계산하기 위해 속도의 전이 과정을 거치게 된다. 속도가 갱신될 필요가 있는 영상의 한 점을 두르고 있는 주변의 점들의 속도를 이용하여 보다 개선된 속도벡터필드를 구하는 과정인 것이다.

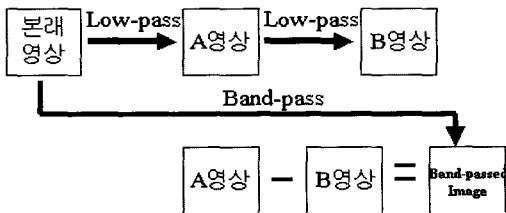


그림 9 영상의 band-pass 필터링

4.2.1 1단계 광류계산

본 논문에서는 광류 필드를 계산하기 위한 1단계 과정을 위해 Correlation-based 접근방식을 사용하였다. 이는 연속된 두 영상 혹은 그 이상의 영상에서 영상 세기의 국부적인 공간 분포가 유지된다는 가정을 하고 있는 것으로, 이러한 가정은 잡음 때문에 에러가 발생하기 쉽다. 따라서, 시간 축 상에 존재하는 두 영상 혹은 그 이상의 영상에서 잘 보존될 수 있는 영상 속성을 선택하는 것이 중요 문제가 된다. 이러한 영상 속성은 보다 더 정확한 광류 필드를 제공해 준다. 본 논문에서는 계산의 단순성을 위해 band-pass필터링이 수행된 영상의 국부적인 공간 분포를 사용하였다. 특히, 본 연구는 특별한 band-pass필터를 생성하기 보다는 한 영상을 연속해서 두번 low-pass한 후 low-pass된 두 영상의 차영상을 이용하였다.

시간축상에서 연속된 t 프레임과 t+1프레임 각각에 대해 band-pass필터링을 수행한 후, t프레임 영상(I₁)에서 각 점(x,y)을 중심으로 (2n+1)×(2n+1)의 윈도우W_p를 형성한다. 또한 t+1프레임 영상(I₂)에서 t프레임과 대응되는 점(x,y)을 중심으로 (2N+1)×(2N+1)의 탐색영역 W_s를 설정한다. 그러면, 식(12)를 사용해 탐색 영역 상에서 에러분포를 계산하고 이것을 식(13)을 사용해 반응분포로 변환하면 영상 t프레임에서의 한 점(x,y)가 탐색영역내의 각 점과 어느 정도 반응을 보여주는지 알게 된다.

$$E_c = \sum_{i=-n}^n \sum_{j=-n}^n (I_1(x+i, y+j) - I_2(x+u+i, y+v+j))^2, -N \leq u, v \leq N \quad (12)$$

$$R_c(u, v) = e^{-k \cdot E_c} \quad (13)$$

본 연구에서는 n=1, N=4, k=10⁻⁴의 값을 이용하여 반응분포를 계산하였다.

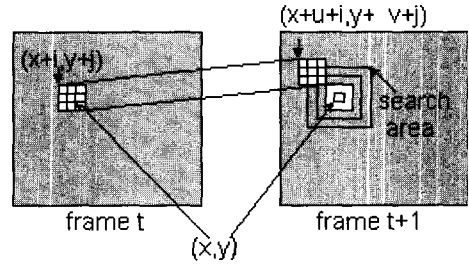


그림 10 반응분포계산

계산된 반응 분포 정보를 이용하여 1차적인 속도벡터를 계산할 수 있는데, 1차적인 속도벡터필드 즉, 광류 U_{cc}=(u_{cc}, v_{cc})는 (14)식을 이용하여 계산할 수 있으며 이 때 다음 단계에서 보다 더 정확한 광류를 계산하는데 필요한 분산행렬 S_{cc}를 계산하기 위해 (15)식을 이용하여 분산행렬을 구한다[17].

$$u_{cc} = \frac{\sum_u \sum_v R_c(u, v) u}{\sum_u \sum_v R_c(u, v)}, \quad v_{cc} = \frac{\sum_u \sum_v R_c(u, v) v}{\sum_u \sum_v R_c(u, v)} \quad (14)$$

$$S_{cc} = \begin{pmatrix} \frac{\sum_u \sum_v R_c(u, v) (u-u_{cc})^2}{\sum_u \sum_v R_c(u, v)} & \frac{\sum_u \sum_v R_c(u, v) (u-u_{cc})(v-v_{cc})}{\sum_u \sum_v R_c(u, v)} \\ \frac{\sum_u \sum_v R_c(u, v) (u-u_{cc})(v-v_{cc})}{\sum_u \sum_v R_c(u, v)} & \frac{\sum_u \sum_v R_c(u, v) (v-v_{cc})^2}{\sum_u \sum_v R_c(u, v)} \end{pmatrix} \quad (15)$$

4.2.2 2단계 광류 계산

광류를 계산하기 위한 2단계 과정의 핵심은 주변의 속도 정보를 고려함으로써 속도를 전파시키는 과정이다. 한 점을 중심으로 주변에 있는 각 점들의 속도는 중심점의 속도에 대한 측정치(혹은 가중치)라고 생각할 수 있다. 즉, 주변의 속도 정보를 이용하여 중심점의 속도를 계산할 수 있음을 의미한다. 따라서, 한 점을 중심으로 적당한 가중치 마스크(mask)를 사용함으로써 주변의 속도 정보를 유용하게 함으로 중심점의 속도값을 개선할 수 있다.

본 논문에서는 1단계를 통해 계산된 광류필드에 [표 1]의 가우시안(Gaussian) 마스크를 적용하여 영상의 각 점에서의 속도를 가중화하였다. 가중화 된 속도벡터 필

드를 R_n 로 표시하자. 이제 가중화 된 속도벡터필드 식(16)을 적용하여 새로운 속도 측정치 $U_n=(u_n, v_n)$ 을 구하며 더 정확한 광류를 계산하는 데 필요한 분산행렬을 S_n 을 (17)식으로 계산한다.

표 1 Gaussian Mask

0.0625	0.125	0.0625
0.125	0.25	0.125
0.0625	0.125	0.0625

$$u_n = \frac{\sum_u \sum_v R_n(u, v) u}{\sum_u \sum_v R_n(u, v)}, \quad v_n = \frac{\sum_u \sum_v R_n(u, v) v}{\sum_u \sum_v R_n(u, v)}, \quad -w \leq u, v \leq w \quad (16)$$

$$S_n = \begin{pmatrix} \frac{\sum_u \sum_v R_n(u, v)(u-u_n)^2}{\sum_u \sum_v R_n(u, v)} & \frac{\sum_u \sum_v R_n(u, v)(u-u_n)(v-v_n)}{\sum_u \sum_v R_n(u, v)} \\ \frac{\sum_u \sum_v R_n(u, v)(u-u_n)(v-v_n)}{\sum_u \sum_v R_n(u, v)} & \frac{\sum_u \sum_v R_n(u, v)(v-v_n)^2}{\sum_u \sum_v R_n(u, v)} \end{pmatrix} \quad (17)$$

그러면 현재, 속도에 대해 두 개의 측정치 U_{cc} 와 U_n 을 가지고 있다. 또한 분산행렬 S_{cc} 와 S_n 을 가지고 있다. 이 값들을 이용해 1단계에서 생성한 정보와 2단계에서 생성한 정보를 동시에 고려하는 속도에 대한 개선된 측정치를 계산할 수 있다. 그러나, 참(true) 속도 값에 비해 U_{cc} 는 식(18)정도의 에러 값이 존재하며, U_n 은 식(19)정도의 에러 값이 존재한다[17]. 따라서 이 두 에러 값을 가장 작게 하는 값 U 가 영상의 한 점에서의 참 속도 값이라 말할 수 있을 것이다. 이 참 값 U 는 식(20)을 만족해야 한다. 참 값 U 는 식(20)의 Gauss-Siedel relaxation 알고리즘[18]에 의해 식(21)의 값을 갖는다.

$$(U - U_n)^T S_n^{-1} (U - U_n) \quad (18)$$

$$(U - U_{cc})^T S_{cc}^{-1} (U - U_{cc}). \quad (19)$$

$$\text{Minimize} \int \int \{ (U - U_n)^T S_n^{-1} (U - U_n) + (U - U_{cc})^T S_{cc}^{-1} (U - U_{cc}) \} dx dy \quad (20)$$

$$U^{*+1} = [S_{cc}^{-1} + S_n^{-1}]^{-1} [S_{cc}^{-1} U_{cc} + S_n^{-1} U_n], \quad U^0 = U_{cc} \quad (21)$$

식(21)에서 U 가 식(20)문제의 답이 되며 2단계 과정을 통한 보다 더 정확한 광류 값을 제공한다. 보다 더 정확성이 있는 광류 필드를 계산하기 위해 매 프레임마다 2단계 과정을 반복 수행할 수 있다.

본 논문에서는 크기 440x330인 동영상을 9x9 윈도우의 탐색 영역을 사용하여 실험하였다. [그림11]는 두 프레임간의 광류 필드를 보여주고 있다. t프레임의 한 픽셀이 t+1프레임에서 좌우상하로 4픽셀이상 움직이지 않을 것이라는 가정을 두고 있는 것이다. 구해진 광류 필

드는 t프레임 위에서 그려져 있으며 두 프레임사이의 변화가 있는 곳에만 광류가 존재하고 있다.



(a) 계산된 광류

(b) Frame t

(c) Frame t+1

그림 11 광류 필드

4.3 최적의 움직임 벡터값 설정

매개변수화된 움직임 벡터에 대해 표기상 편리함을 위해 각각의 움직임 모델을 다음과 같이 정의한다.

$$X(x) = X \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & x & y & 0 & 0 & 0 & x^2 & xy & 0 \\ 0 & 0 & 0 & 1 & x & y & xy & y^2 & x^2 \end{bmatrix} \quad (22)$$

$$A = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ 0 \ 0 \ 0]^T \quad (23)$$

$$P = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ p_1 \ p_2 \ 0]^T \quad (24)$$

$$C = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ 0 \ 0 \ c]^T \quad (25)$$

$X(x)A$, $X(x)P$, $X(x)C$ 는 각각 어파인(affine), 평면(planar), 어파인+평면(affine+planar) 움직임 모델을 나타낸다. R을 얼굴 요소 혹은 얼굴 영역이라고 정의하면, C_{mR} 은 입 영역을 나타내고 마찬가지로 C_{lR} , C_{rR} , A_{lR} , A_{rR} , P_R 은 각각 눈썹영역, 눈 영역, 얼굴 영역을 나타낸다.

얼굴요소 영역 혹은 얼굴 영역의 움직임을 묘사하는 각각의 매개변수화 된 움직임 벡터가 광류를 통해 계산된 실질적인 움직임 필드를 가장 잘 고려하기 위해서는 어떤한 매개변수 값들로 구성되어야 하는 지를 발견하는 일이다. 다시 말해, 현재 프레임에서의 한 특정영역에 대한 초기의 움직임 벡터를 C_i 라고 할 때 실제의 광류를 가장 잘 고려하는 후보 움직임 벡터 C_j 를 찾는 것이다. 이를 위해 본 논문에서는 매개변수화 된 움직임 벡터에 의한 flow와 광류 사이의 최소자승기법(mean squared error)을 이용하였다(식26). 4.2.2 절에서 계산된 영상에서의 광류를 벡터 $[u', v']^T$ 로 움직임 벡터

에 의한 flow를 $[u, v]^T$ 로 정의하면,

$$e_k = (u - u')^2 + (v - v')^2, E(c_i, c_j) = \frac{1}{n} \sum_{e_k} e_k, \forall (x, y) \in R \quad (26)$$

n은 영역에 속하는 점들의 개수를 의미한다.

식(26)을 이용하여 실제의 광류를 가장 잘 고려하는 것은 식(27)을 만족하는 최적의 매개변수 벡터 c_j^* 이다.

$$c_j^* = \arg(\min_j E(c_i, c_j)) \quad (27)$$

이 때, E값을 최소로 하면서 수 개의 파라미터 값을 정확히 구해내는 것은 불가능한 일이다. 따라서, 국부적 최소치를 찾아야 할 것인데 이러한 최소치는 c_j (현재의 매개변수 벡터)를 시작 벡터, E를 에러함수로 하여 conjugate gradient descent[13] 방법을 이용하였다.

4.4 얼굴 요소 및 얼굴 영역의 예측, 변환

동영상의 두 프레임 사이에서 측정된 얼굴요소 및 얼굴 영역에 대한 매개변수화 된 움직임 벡터 값은 각 영역의 움직임에 대한 정보 뿐 아니라, 다음 프레임에서 특정 영역의 위치를 예측할 수 있게 한다. 눈 영역이나 얼굴 영역은 네 개의 꼭지점에 의해 표현될 수 있는 간단한 사변형이다. 이러한 영역상의 한 라인(line)은 어파인(affine) 움직임이나 평면(planar) 움직임에 대해서 여전히 하나의 라인으로 변형되기 때문에, 얼굴 영역이나 눈 영역의 네 꼭지점의 위치가 변하더라도 여전히 사변형으로 남아 있다. 따라서 각 영역의 네 꼭지점(x)에 $X(x)P + x, X(x)A1 + x, X(x)Ar + x$ 을 적용함으로써 꼭지점(x)의 위치를 변경하였다. 이로써 얼굴 영역이나 눈 영역이 설정된 최적의 매개변수화 된 움직임 모델로써 변형될 수 있다. 또한, 이렇게 변형된 영역은 현재 프레임에서 각 영역의 위치를 예측하고 있는 것이다. 그러나, 입 영역이나 눈썹 영역은 곡률변수(식6의 'c')로 인해 꼭지점의 단순한 변경으로 영역의 추적이 가능하지 않다. 따라서, 이러한 영역은 눈썹과 입에 대응되는 영상의 영역을 나타내기 위해 영상 마스크(image mask)를 이용하였다. 이러한 영상 마스크는 우선, 얼굴 영역을 위한 매개변수화 된 움직임 모델 P(식3.24)에 의해 변형이 되고 그리고 나서, 각각의 Cm, Clb, Crb등의 움직임 모델이 적용되도록 하였다.

4.5 움직임 정보에 의한 3차원 얼굴 모델의 움직임 변환

4.1절에서는 움직임 벡터를 구성하는 몇 개의 변수의 조합으로 나타낼 수 있는 움직임 정보를 제시하였다. 이제 이러한 움직임 정보가 3.1절에서 언급한 3차원 얼굴 모델에 전달되어 각 정보와 관련된 단위행위들이 작동

된다. 또한, 단위행위들의 움직임 량 역시 이 정보로부터 유추되어 진다.

본 논문에서는 이러한 움직임 정보의 의미를 단위행위의 정의와 연관지어 이 둘간의 상호관계를 이루었다. 각각의 움직임 정보에 담겨진 얼굴 요소 영역 행위의 의미는 다음과 같다.

각 움직임 정보는 실험을 통해 결정되는 임계치 값을 기준으로 얼굴 요소 영역의 행위를 정의하게 된다. 이를 테면, a0는 01을 기준으로 01보다 크면 오른쪽 이동을 의미하는 반면, -01보다 작으면 왼쪽 이동을 의미한다. 실험을 통해 초기의 몇 프레임시퀀스(frame sequence)로부터 경험적으로 결정되는 임계치(threshold)값을 0라고 하면, 표2와 같은 의미를 갖는다. 이러한 점에 착안하여 각각의 얼굴 요소 영역의 움직임을 나타내는 움직임정보와 같은 행위의 의미를 나타내는 단위행위와 연결짓는 것이다.

입 영역의 경우, 주된 움직임 정보는 곡률 정보와 변형성 정보 Def이며 이것과 그 움직임 양에 따라 단위행위(Action Unit) 14, 15, 18, 20, 24를 조절하도록 하였다. 눈썹 영역의 경우에서의 주된 움직임 정보는 a3와 발산성 정보 Div이며 이 값에 따라 눈썹과 관련된 단위행위 1, 2, 4등을 조절하도록 하였다. 눈 영역에 있어서는 매개변수 a3와 변형성 정보Def이며 따라서 눈과 관련된 단위행위5, 41등을 조절하였다[표3].

표 2 매개변수의 의미

움직임 정보	임계치값 $\theta (>0)$ 보다	행 위
a0	>01	오른쪽으로 이동
	<-01	왼쪽으로 이동
a3	<-02	위쪽으로 이동
	>02	아래쪽으로 이동
Div	>03	영역의 확장
	<-03	영역의 축소
Def	>04	수평방향으로 변형
	<-04	수직방향으로 변형
Curl	>05	시계방향으로 회전
	<05	반시계방향 회전
c	<-06	위쪽으로 굽어짐
	>06	아래쪽으로 굽어짐

표 3 움직임 정보와 단위행위

영역	중요 움직임 정보	관련 단위행위
입	Def, c	14,15,18,20,24
눈썹	a3, Div	1,2,4
눈	a3, Def	5, 41

5. 실험결과 및 분석

실험에 사용된 동영상은 성격이 다른 세 종류의 동영상이다. 이 동영상들은 흑백 CCD 카메라를 통해 입력된 AVI 화일을 프레임마다 스캔(scan)하여 각 프레임 영상을 256그레이 원시(raw)영상이 되도록 변형하여 시스템에 사용하였다. 실험에 쓰인 동영상은 각각 놀람(Surprise), 미소(smile), 눈깜박임(Blink)의 얼굴 표정을 연출하고 있다. 놀람 표정에서는 눈썹, 입, 눈 등의 영역이 확장되어 얼굴 요소의 전반적임 움직임이 고려되어 있고, 미소는 휘어지는 입술의 움직임에, 눈 깜박임 표정은 축소되는 눈 영역의 움직임에 주안점을 두고 있다. 결국 이 세 동영상에서 주어지는 얼굴 영역 및 얼굴 요소 영역의 움직임을 3차원 얼굴 형상 모델이 그 움직임을 매 프레임마다 그 움직임을 합성하는 것과 더불어 2차원 동영상에서 각 요소 영역의 추적한다. 시스템은 200MHz Pentium-pro PC에서 Visual C++5.0을 이용하여 구현하였다.

5.1 놀람 표정을 연출하는 동영상

놀람 표정을 연출하는 동영상은 총 60여 프레임의 영상으로 구성되어 있다. 계산된 광류 정보에 따라 얼굴 요소의 2차원적 추적과 3차원 얼굴 모델에 전달된 움직임 정보로써 움직임을 합성한 모습은 [그림12]과 같다. 맨 좌측 열은 동영상에서 나타나는 영상을 보여주고 있으며, 가운데 열은 계산된 광류에 따라 예측되어 추적된 얼굴 요소 영역을 보여주고 있다. 그리고 마지막 열은 3차원 얼굴 모델에 전달된 움직임 정보를 이용하여 변형되는 3차원 얼굴 모델을 보여주고 있다. 각 행은 3프레임간의 격차를 보여주고 있다. [그림12]의 첫행은 0프레임에서 영상, 초기의 얼굴 요소 영역, 초기의 3차원 얼굴 모델을 보여주며 마지막 행은 60번째 프레임에서의 영상 즉, 표정이 최고 극치일 때와 여러 과정을 거친 후의 추적되고 예측된 얼굴 요소 영역, 그리고 놀람 표정의 최고 극치일 때의 움직임을 합성한 3차원 얼굴 모델을 보여주고 있다. [표4]와 [표5]는 각각의 프레임에서의 입술과 눈썹에 대한 매개변수들의 변화값을 보여주고 있다. 실험 결과, 전반적으로 얼굴 요소 영역을 만족할 정도로 추적하고 있음을 알 수 있으며, 3차원 얼굴 모델 역시 동영상의 움직임에 따라 그 움직임을 대체로 잘 합성하고 있음을 알게 된다.

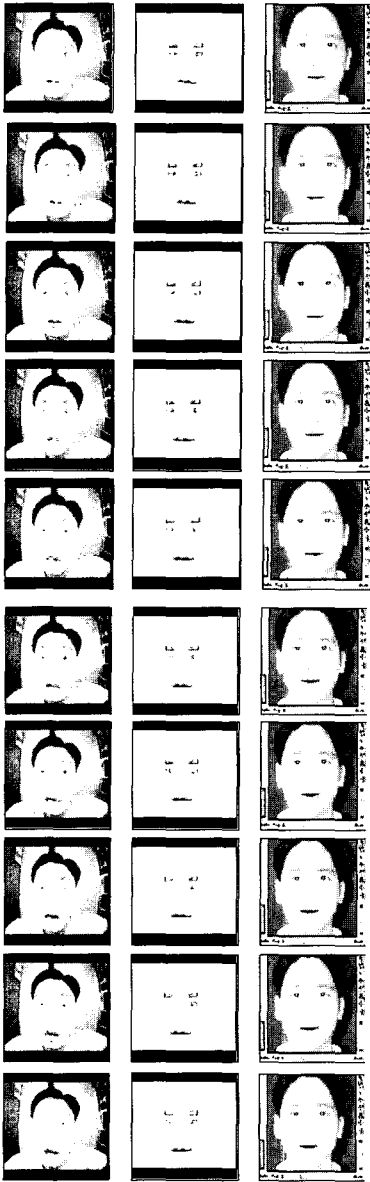
5.2 미소 표정을 연출하는 동영상

미소표정을 연출하는 동영상은 총 30개의 프레임으로 구성되어 있다. 놀람 표정과 같이 연속된 두 영상에서 광류의 계산이 이루어지며 계산된 광류 정보를 이용하

여 미소표정 연출과정에서 얼굴요소 영역의 추적과 움직임 정보를 이용하여 동영상의 움직임을 합성한 3차원 얼굴 모델의 움직임 합성 과정이 매 3프레임마다 나타난 결과를 [그림13]에서 보여주고 있다.



from frame 0 to frame 27

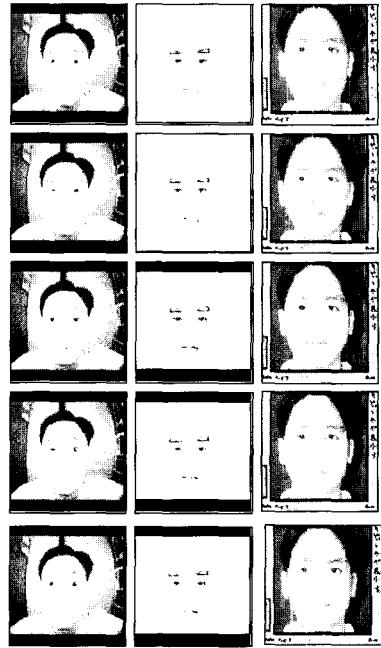


from frame 30 to 57

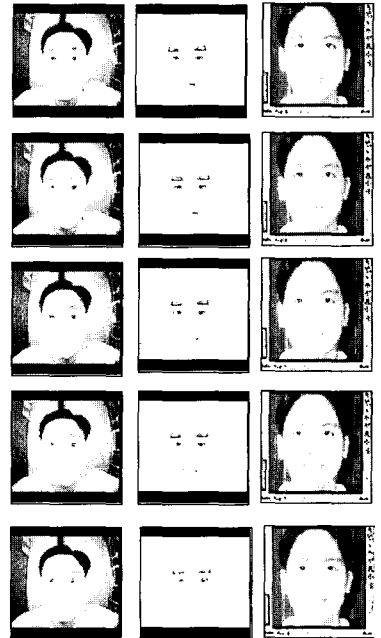
그림 12 놀람 표정의 추적과 합성

5.3 눈깜박임 표정을 연출하는 동영상

눈 깜박임 표정을 연출하는 동영상은 총 42개의 프레임으로 구성되어 있다. 위의 5장의 1절과 2절에서 제시된 것과 같이 눈 깜박임 표정에 대한 결과를 [그림14]에서 보여주고 있다.

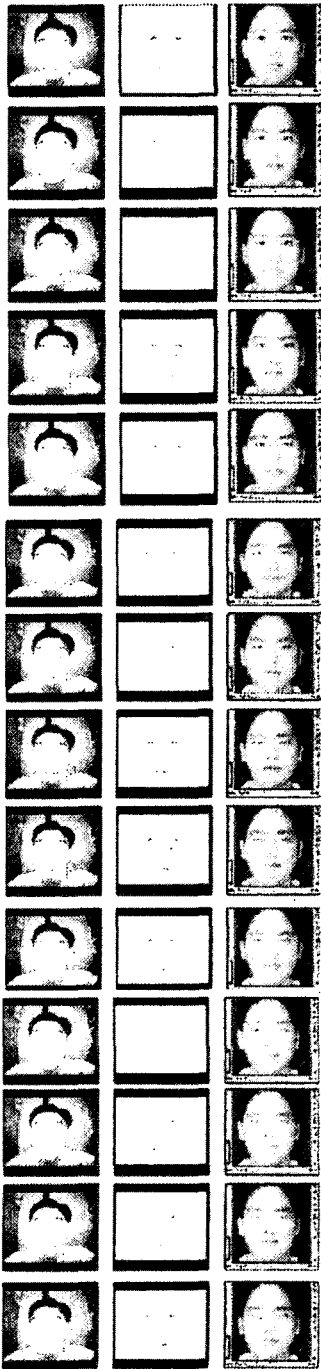


from frame 0 to frame 12



from frame 15 to 27

그림 13 미소 표정의 추적과 합성



0~39 프레임

그림 14 눈감박임 표정의 추적과 합성

표 4 놀람표정의 입술정보에 대한 변화

frame	m_a0	m_a3	m_Div	m_Curl	m_Def	m_c
t = 0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
t = 3	0.150711	0.147705	-0.008633	-0.002467	0.008229	-0.000043
t = 6	0.150711	0.147705	-0.008633	-0.002470	0.008229	-0.000032
t = 9	0.150711	0.147705	-0.008634	-0.002472	0.008228	-0.000011
t = 12	0.150711	0.147705	-0.008634	-0.002474	0.008228	-0.000002
t = 15	0.150711	0.147705	-0.008635	-0.002476	0.008229	-0.000018
t = 18	0.150711	0.147705	-0.008635	-0.002478	0.008229	-0.000043
t = 21	0.150711	0.147705	-0.008636	-0.002479	0.008230	-0.000064
t = 24	0.150711	0.147705	-0.008637	-0.002481	0.008231	-0.000077
t = 27	0.150711	0.147705	-0.008638	-0.002482	0.008232	-0.000091
t = 30	0.150711	0.147705	-0.008638	-0.002483	0.008234	-0.000096
t = 33	0.150711	0.147705	-0.008640	-0.002485	0.008238	-0.000099
t = 36	0.150711	0.147705	-0.008640	-0.002485	0.008240	-0.000108
t = 39	0.150711	0.147706	-0.008640	-0.002486	0.008242	-0.000111
t = 42	0.150711	0.147706	-0.008641	-0.002487	0.008243	-0.000115
t = 45	0.150711	0.147706	-0.008641	-0.002488	0.008243	-0.000109
t = 48	0.150711	0.147706	-0.008641	-0.002488	0.008245	-0.000103
t = 51	0.150711	0.147706	-0.008641	-0.002489	0.008245	-0.000099
t = 54	0.150711	0.147706	-0.008641	-0.002490	0.008245	-0.000094
t = 57	0.150711	0.147706	-0.008641	-0.002490	0.008245	-0.000083

표 5 놀람표정의 왼쪽 눈썹 정보에 대한 변화

frame	lb_a0	lb_a3	lb_Div	lb_Curl	lb_Def	lb_c
t = 0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
t = 3	0.085431	0.057576	0.034408	0.002207	-0.035694	-0.001156
t = 6	0.085436	0.057576	0.034558	0.002207	-0.035694	-0.001038
t = 9	0.085439	0.057575	0.034635	0.002207	-0.035943	-0.000918
t = 12	0.086508	0.057467	0.070416	0.003645	-0.066342	-0.001773
t = 15	0.086515	0.057466	0.070614	0.003651	-0.066526	-0.001594
t = 18	0.086521	0.057466	0.070795	0.003654	-0.066691	-0.001437
t = 21	0.086530	0.057464	0.071095	0.003662	-0.066953	-0.001285
t = 24	0.086533	0.057464	0.071179	0.003665	-0.067031	-0.001143
t = 27	0.086536	0.057464	0.071290	0.003667	-0.067132	-0.001026
t = 30	0.086538	0.057464	0.071346	0.003670	-0.067184	-0.000826
t = 33	0.086541	0.057464	0.071443	0.003667	-0.067279	-0.000732
t = 36	0.086558	0.057464	0.072071	0.003637	-0.067855	-0.000572
t = 39	0.086560	0.057464	0.072113	0.003633	-0.067903	-0.000493
t = 42	0.086561	0.057464	0.072146	0.003632	-0.067932	-0.000358
t = 45	0.086563	0.057464	0.072168	0.003631	-0.067956	-0.000191
t = 48	0.086564	0.057464	0.072193	0.003631	-0.067975	-0.000118
t = 51	0.086564	0.057464	0.072215	0.003634	-0.067989	-0.000092
t = 54	0.086565	0.057465	0.072223	0.003633	-0.067999	-0.000022
t = 57	0.086565	0.057465	0.072231	0.003627	-0.068001	-0.000056

6. 결론 및 향후 연구

본 논문에서는 산출된 광류 값에 따라 각 얼굴요소 및 얼굴 영역의 움직임을 가장 잘 묘사하는 매개변수화된 움직임 벡터 값을 찾아, 이 벡터 값에 따라 얼굴요소 및 얼굴영역의 위치를 예측하고, 움직임을 추적하였다. 그리고 이러한 정보들로 2차원 영상에서의 움직임을 3차원 얼굴모델로 합성하였다. 본 연구는 얼굴 움직임 추적 및 합성 뿐 아니라 얼굴표정인식 시스템에도 응용할 수 있을 것이다.

하지만, 본 연구에서는 프레임이 진행되면서 추출된 특징영역이 예상하는 영역보다 조금씩 넓어지게 되어 필요 없는 부분까지 포함함으로써 아주 세밀한 영역 추출과 움직임 정보를 찾는 데 어려움이 있었다. 따라서, 3차원 얼굴 모델 전체의 움직임은 합성할 수 있지만 세부적인 움직임을 합성하는 데는 정확성이 떨어졌다. 이는 좀 더 정확한 광류를 계산함으로써 개선될 수 있는데, 특히 입력되는 동영상의 조명상태가 가장 큰 영향을 미치는 요소로 보여진다. 따라서 앞으로는 조명과 같은 제약 조건에 영향을 받지 않고도 정확한 광류를 계산하는 방법과 속도를 향상하는 방법이 연구되어야 할 것이며, 이와 결합하여 보다 자연스러운 얼굴 합성 및 실시간 애니메이션이 이루어질 수 있으리라 본다.

참고 문헌

- [1] M. Kass., A. Witkin., and D. Terzopoulos, "Snakes: Active contour models." In Proc. of First International Conference on Computer Vision. pp259-268, 1987.
- [2] D. Terzopoulos., and K. Water., "Analysis and synthesis of facial image sequences using physical and anatomical models." IEEE Transactions on Pattern Analysis and Machine Intelligence. 15(6):569-579, 1993.
- [3] I. Essa., T. Darrel., and A. Pentland., "Tracking facial motion," In Proc. of the Workshop on Motion of Non-rigid and Articulated Objects, Austin, Texas, pp36-42, 1994.
- [4] A. Azarbayejani, B. Horowitz, and A. Pentland, "Recursive estimation of structure and motion using the relative orientation constraint," In Proc. of the Computer Vision and Pattern Recognition Conference, 1993.
- [5] I. A. Essa, and A. Pentland, "A vision system for observing and extracting facial action parameters," In Proc. of Computer Vision and Pattern Recognition, Seattle, WA, pp.76-83, 1994.
- [6] J. Bergen, P. Anadan, K. J. Hanna, and R. Hingorani, "Hierarchical model-based motion estimation," In Proc. of Second European Conference on Computer Vision, ECCV-92, G.Sandini(Ed.), Springer-Verlag, vol 588 of LNCS-Series, pp237-252, 1992.
- [7] Y. Yacoob, and L. S. Davis, "Labeling of human face components from range data," In Proc. of Computer Vision and Pattern Recognition, New York, pp. 592-593, 1993.
- [8] A. L. Yuille, and D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," In Proc. of Computer Vision and Pattern Recognition, pp.104-109, 1989.
- [9] Ajit Singh, Optic Flow Computation : A Unified Perspective, IEEE Computer Society Press, 1991.
- [10] 심연숙, 박도영, 변혜란. "휴먼 인터페이스를 위한 얼굴 합성 및 표정합성", 한국정보과학회 HCI '98학술대회 발표논문집, 1998. 2
- [11] 심연숙, 박도영, 전원종, 변혜란. "표준 한국인 얼굴 모델링 및 얼굴의 표정합성", 한국인지과학회 춘계학술발표논문집, 1998.
- [12] P. Ekman, and W. V. Friesen, "Facial Action Coding System," Palo Alto, CA, Consulting-Psychologist Press, 1978.
- [13] William H. Press, Brian P. Flannery, Saul A. Teukoisky, and William T. Vetterling, Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1988.
- [14] J. J. Koenderink, and A. J. van Doorn, "Invariant properties of the motion parallel field due to the movement of rigid bodies relative to an observer," Optica Acta, 22(9):773-791, 1975
- [15] M. J. Black, and Y. Yacob, "Tracking and recognizing rigid and non-rigid facial motions using local parameteric model of image motion," In Proc. of the International Conference on Computer Vision, pp. 374-381, IEEE Computer Society, 1995.
- [16] R. Cipolla, and A. Blake, "Surface orientation and time to contact from image divergence and deformation," In Proc. of ECCV-92, G.Sndini, Springer-Verlag, vol 588 of LNCS-Series, pp. 187-202, 1992.
- [17] J. V. Beck, and K. J. Arnold, "Parameter estimation in engineering and science". John Wiley, New York, 1977.
- [18] A. Ralston, and P. Rabionwitz, A First Course in Numerical Analysis, McGraw-Hill, New-York, 1978.
- [19] F. I. Parke. "Computer generated animation of faces," Master's thesis, University of Utah, Salt Lake City, UT, UTEC-CSc-72-120, 1972.
- [20] F. I. Parke. "A Parameteric Model for Human Faces," PhD thesis, Univirsity of Utah, Salt Lake City, UT, December UTEC-CSs-72-120, 1974.
- [21] S. M. Platt, "A system for computer simulation of

the human face" Master's thesis, The Moore School, University of Pennsylvania, Philadelphia, 1980.

- [22] P. Bergeron and P. Lachapelle, "Controlling facial expression and body movements," In Advanced Computer Animation, SIGGRAPH'85 Tutorials, Volum2, pages 61-79. ACM, New York, 1985.
- [23] K. Waters. "A muscle model for animating three-dimensional facial expressions," Computer Graphics (SIGGRAPH '87),21(4):17-24, July 1987.
- [24] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," Computer Graphics, 29(4):55-62, August 1995.
- [25] M. Black, Y. Yacoob, "Recognizing facial expressions in image sequences using local parameterized models of image motion," International Journal of Computer Vision,25(1),23-48, 1997



박도영

1997년 홍익대학교 수학과 학사. 컴퓨터 공학 공학사. 1999년 연세대학교 컴퓨터 과학과 공학석사. 1999년 10월 ~ 현재 육군사관학교 전산학과 강사. 관심분야는 인공지능, 영상처리, 영상인식, 컴퓨터그래픽



심연숙

1994년 서강대학교 전자계산학과 공학사. 1996년 서강대학교 전자계산학과 공학석사. 1997년 ~ 연세대학교 인지과학 협동과정 박사과정. 1997년 ~ 연세대학교 강사. 관심분야는 인공지능, HCI, 그래픽스.

변혜란

정보과학회논문지: 소프트웨어 및 응용
제 27 권 제 1 호 참조