

스케치 인터페이스를 이용한 데이터 기반 얼굴 애니메이션

주은정[○] 안소민 이재희
서울대학교 컴퓨터공학부
{ejjoo, itscluster, jehee}@mrl.snu.ac.kr

Data-driven Facial Animation Using Sketch Interface

Eunjung Ju[○] Sohmin Ahn Jehee Lee
School of Computer Science and Engineering, Seoul National University

요 약

자연스러운 얼굴 애니메이션 생성은 캐릭터 애니메이션 분야에서 중요한 문제이다. 지금까지 얼굴 애니메이션은 3차원 모델링 프로그램을 이용한 전문 애니메이터들의 수작업을 통해 생성되거나, 필요한 움직임 데이터를 직접 동작 캡처함으로써 만들어 왔다. 그러나 이러한 방식은 일반 사용자가 쉽게 접근할 수 없으며 많은 시간과 비용을 요구한다는 단점이 있다. 본 연구에서는 실제에 가깝고 자연스러운 얼굴 애니메이션을 만들기 위해, 누구나 쉽게 사용할 수 있는 직관적인 방식의 스케치 인터페이스를 이용하고자 한다. 이를 통해 키-프레임을 생성하는 시스템을 구축하고, 얼굴 캡처를 통하여 얻은 데이터로부터 추출한 얼굴 표정간의 전이 정보를 이용하여 키-프레임을 보간하는 방식을 제안한다. 본 시스템은 전문 애니메이터가 아닌 일반 사용자도 쉽고 빠르게 다양한 감정을 표출하며, 동시에 말하는 얼굴 애니메이션을 만들 수 있도록 한다.

Abstract

Creating stylistic facial animation is one of the most important problems in character animation. Traditionally facial animation is created manually by animators or captured using motion capture systems. But this process is very difficult and labor-intensive. In this work, we present an intuitive, easy-to-use, sketch-based user interface system that facilitates the process of creating facial animation and key-frame interpolation method using facial capture data. The user of our system is allowed to create expressive speech facial animation easily and rapidly.

키워드 : 얼굴 애니메이션, 스케치 인터페이스, 데이터 기반 애니메이션, 애니메이션

Keyword : facial animation, sketch interface, data-driven animation, animation

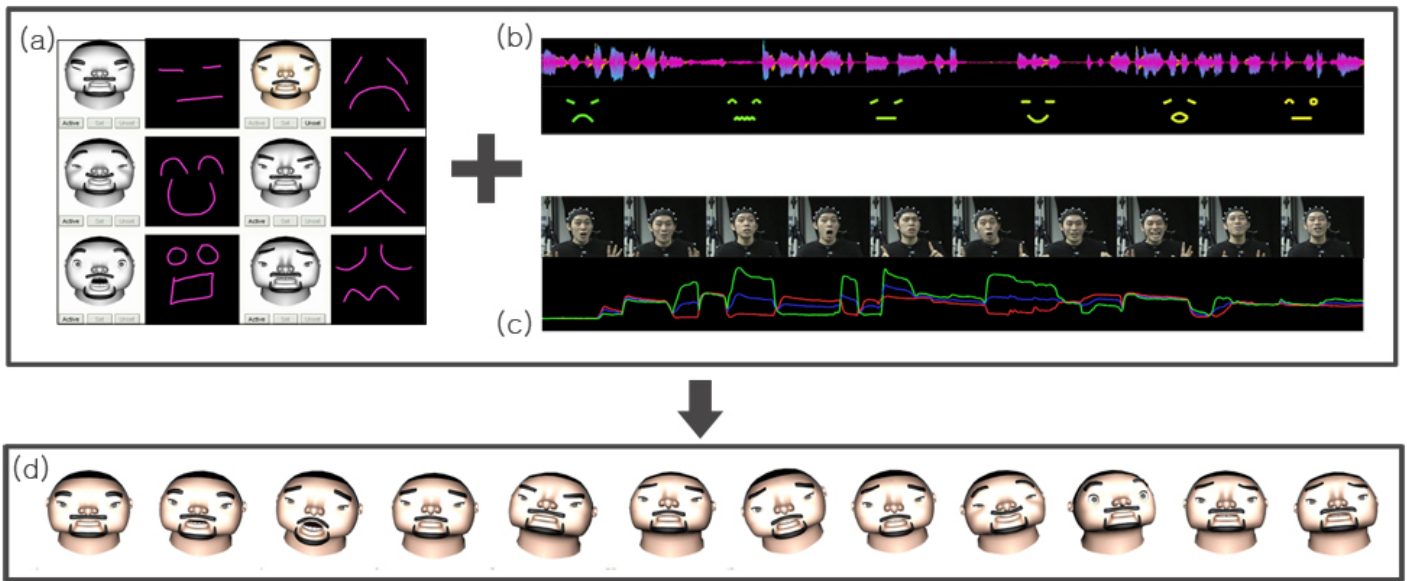


1. 서론

자연스러운 얼굴 애니메이션 생성은 캐릭터 애니메이션 분야의 중요한 문제이다. 애니메이션 영화나 컴퓨터 게임 같은 다양한 애니메이션 응용 분야에서는 섬세하고 스타일리시한 애니메이션을 필요로 하며, 이를 위해 다양한 동작 데이터를 직접 캡처하거나 애니메이터의 수작업을 통해 원하는 동작을 만들어 내는 방식을 이용해 왔다. 그러나 캐릭터의 움직임을 동작 데이터 캡처를 통해서 얻는 것은 많은 시간과 비용이 들고, 필요로 하는 동작이 실제 사람으로부터 캡처할 수 없는 경우

도 있다. 또한 애니메이션 속의 캐릭터는 실제 사람의 동작보다 과장된 움직임으로 표현해야 하기 때문에 동작 캡처 데이터를 이용하는 것과 함께 애니메이터들의 수작업 후처리 과정도 필요하다.

또 다른 방식인 애니메이터들이 처음부터 수작업으로 애니메이션을 만드는 경우는 키-프레임 보간법을 이용한다. 키-프레임 보간법은 캐릭터 합성의 여러 가지 기법 중에서 가장 전통적인 방식이면서도 여전히 현업에서 많이 사용되고 있는 기법으로서, 애니메이션 생성에 있어 효과적인 방식으로 알려져 있



[그림 1] 전체적인 시스템 구조 (a) 사용자가 서로 다른 표정의 얼굴 모델을 자신만의 스트로크로 정의한다. (b) 음성 데이터를 기반으로 하는 타임라인의 원하는 위치에 키-프레임을 스케치한다. (c) 얼굴 캡처 데이터로부터 추출한 전이 정보를 이용하여 키-프레임 사이를 보간한다. (d) 이러한 과정들을 통해 감정을 표출하며 말하는 얼굴 애니메이션이 생성 된다.

다. 이것은 키-프레임이 되는 캐릭터를 직접 모델링하고 그 프레임들 사이를 보간해서 전체 프레임을 완성하는 방식이다.

키-프레임 보간법으로 만드는 애니메이션은 자연스럽게 사실적인 애니메이션 생성을 위해 두 가지의 중요한 작업이 필요하다. 첫째는 현재 상태의 특성을 잘 표현하는 키-프레임 캐릭터를 만드는 것이고 둘째는 그러한 키-프레임을 많은 개수로 잡아서 보간 되는 구간을 적게 하는 것이다. 그러나 3차원 캐릭터의 몸동작이나 얼굴 표정 등의 모델을 생성하고 제어하기 위해 캐릭터의 3차원 좌표를 2차원 입력장치인 마우스나 타블렛 등으로 조작하는 것은 직관적이지 못하고 많은 노동력을 요하며 전문적인 훈련을 통해서만이 가능한 어려운 작업이다.

본 논문에서는 3차원 캐릭터의 키-프레임링 작업을 용이하게 하는 직관적인 방식의 스케치 인터페이스와 적은 수의 키-프레임으로도 자연스러운 애니메이션 생성을 가능하게 하는 데이터 기반 보간법을 제안한다. 스케치 인터페이스는 몇 개의 간단한 스트로크를 입력으로 하는 시스템으로써, 사용자가 다양한 얼굴 표정의 상징적인 특징을 몇 개의 선으로 표현하면 키-모델을 생성해준다. 또한 이 키-프레임들의 보간에 있어서, 실제 사람의 얼굴 움직임 정보를 가지고 있는 얼굴 캡처 데이터로부터 추출한 표정간의 전이 정보를 이용함으로써 적은 개수의 키-프레임으로도 실제에 가깝고 자연스러운 애니메이션이 생성되도록 한다.

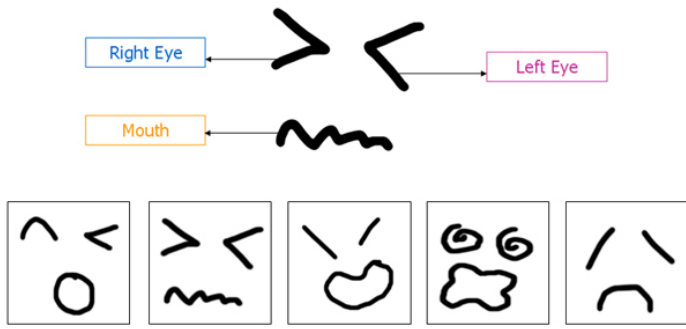
2. 관련 연구

얼굴 애니메이션은 다양한 형태를 기반으로 애니메이션을 생성하는 기법들이 개발되어 왔다. 이미지나 비디오 같은 2차원

데이터를 기반으로[1,2,3] 애니메이션을 생성하는 것이 가장 많이 이용되어 오던 방식이며 그 외에 실제 사람 얼굴의 근육 모델을 모방[4] 하거나 동작 캡처 혹은 얼굴의 스캔 데이터를 이용하여 3차원 캐릭터 얼굴 애니메이션을 제어[5]하는 방식 등이 있다. 또한 애니메이션 생성의 목적에 있어서는 좀 더 자연스럽게 실제에 가까운 얼굴 표정을 표현[6,7,8,9]하고자 하는 것과 감정의 표현보다는 말하는 입모양의 애니메이션에 주목하여 말하는 얼굴 애니메이션[10]을 만들고자 하는 것 등이 있었다.

최근에 와서는 실제 사람처럼 감정 표현과 말하는 것 두 가지를 동시에 표현하는, 감정을 표현하면서 말하는 얼굴 애니메이션에 관한 연구[11,12,13]가 활발히 진행되고 있다. 이를 통해 지금까지 분리된 형태로 발전해왔던 두 가지가 합쳐져서 더 사실적인 얼굴 애니메이션 생성이 가능해지도록 하고 있다. 이에 본 연구에서도 감정을 표현하면서 말하는 얼굴 애니메이션을 생성한다.

우리는 얼굴 애니메이션을 생성하는데 있어서 스케치 인터페이스 방식을 사용하여 일반 사용자도 쉽게 사용할 수 있는 시스템을 개발하고자 한다. 스케치는 모션 합성과 상호작용하는 캐릭터 애니메이션 생성 등에 있어서 널리 사용되어 왔다. 캐릭터의 걸거나 뛰는 경로들을 스케치하거나[14,15], 모션의 2차원 자세를 스케치하여 3차원 자세를 만들어 냄으로써 애니메이션을 생성[16] 하였다. 또한 Li[17]은 3차원의 모형이나 애니메이션 생성을 위해 2차원 스케치 입력을 이용하였고, Thorne[18]은 캐릭터 애니메이션의 움직임 생성을 위해 사용자에 의해 정의된 스케치를 분석하는 인터페이스를 개발하였다. 이러한 연구들에서 보여지듯이 스케치 인터페이스는 사용자에게 있어 직관적인 입력 시스템으로써 비전문가 사용자들이 쉽게 애니메이션을 생성하는 것을 가능하게 해준다.



[그림 2] (위) 얼굴의 상징적인 특징을 왼쪽, 오른쪽 눈과 입, 3개의 스트로크로 표현하기로 정의한다. (아래) 세 개의 스트로크로 다양한 감정의 얼굴 표정을 스케치 할 수 있다.

3. 전체적인 시스템 구조

본 시스템은 전문 애니메이터에 의해 모델링된 다양한 표정 및 입모양을 가진 3차원 캐릭터 얼굴 모델과 실제 사람의 얼굴 캡처 데이터를 기본으로 한다. 우리는 먼저 각 사용자에게 데이터베이스에 있는 여러 가지 표정의 얼굴 모델에 대해 자신만의 스트로크로 표정을 정의하도록 한다. 그 다음 목표하는 애니메이션의 바탕이 될 음성데이터를 선택한다. 이 음성 데이터의 소리로부터 각 발음의 가중치를 추출하고, 서로 다른 입모양의 얼굴 모델을 이 가중치에 따라 혼합함으로써 말하는 얼굴 애니메이션이 만들어진다. 이렇게 만들어진 애니메이션에 표정을 가진 키-프레임을 설정함으로써 감정을 표현하면서, 동시에 말하는 얼굴 애니메이션이 생성된다. 키-프레임 설정을 위해서는 음성 데이터와 같은 길이의 타임라인이 주어지는데, 그 타임라인 위의 표정이 바뀌길 원하는 시간에 사용자가 스케치를 함으로써 키-프레임을 지정한다. 마지막으로 이 키-프레임 사이의 프레임들이 캡처 데이터로부터 추출한 전이 정보를 기반으로 보간됨으로써 애니메이션이 완성된다. 위의 과정을 거쳐서 하나의 애니메이션 클립이 만들어지는 것은 각 과정이 실시간으로 진행된다. ([그림 1] 참조)

이후로 4장에서는 스케치 인터페이스를 이용하여 키-프레임을 만드는 것을, 5장에서는 얼굴 캡처 데이터를 이용해 키-프레임을 보간하는 법을 설명할 것이다.

4. 스케치 인터페이스를 이용한 키-프레이밍

우리는 키-프레임 보간법으로 캐릭터 합성을 할 때의 키-프레임 얼굴 모델을 스케치 인터페이스를 이용하여 생성하고자 한다. 그러나 얼굴 모델을 만드는 데에 있어서 개략적인 2차원 선의 입력만으로 3차원의 모델을 구성해 내는 것은 과소결정 문제이므로 이 문제를 해결하기 위해서는 목표 얼굴 모델에 대한 사전에 가정된 기본 지식이 필요하다.

본 연구에서는 사전 지식으로 전문 애니메이터들이 만든 다양한 표정의 얼굴 모델을 이용한다. 사용자가 입력한 2차원의 그

림으로부터 직접적으로 3차원의 캐릭터 모델을 재구성해내는 것이 아니라, 애니메이터에 의해 만들어진 얼굴 모델들로부터 사용자의 입력에 가장 잘 맞는 모델을 찾는 것이다. 우리는 이러한 얼굴 모델을 몇 개의 영역으로 나누어서 각각의 데이터베이스에 저장하고, 사용자의 스케치에 대해 각 영역을 독립적으로 검색하여 이용한다. 이것은 사용자가 키-프레임을 스케치했을 때 단지 이미 만들어져있는 모델들 중에 하나를 찾아내는 것이 아니라, 사용자의 각 스트로크에 맞는 모델들을 찾아 합성함으로써 새로운 얼굴 모델을 생성하게 됨을 의미한다.

이후에는 얼굴 표정을 스케치하는 방법의 정의, 사용자에게 의 해 스케치 된 선의 분석, 그리고 마지막으로 입력된 스케치에 매치되는 모델을 찾아서 키-모델을 구성하는 방법을 차례로 설명할 것이다.

4.1 얼굴 스케치 방식의 정의

정의된 형식 없이 자유롭게 그려진 그림은 그 해석의 범위가 매우 광범위하여 분석이 어렵기 때문에, 스케치 인터페이스를 이용하기 위해서는 먼저 입력 스트로크의 형식이 정의되어야 한다. 사람의 몸의 경우 한 가지 특정한 자세에 대해서 사람마다 몸의 마디 각 부분의 길이와 관절의 각도가 약간씩 다를 수는 있지만 전체적인 골격의 윤곽선은 동일한 형태를 보여준다. 반면 얼굴 표정의 경우는 한 가지 특정한 감정에 대해서도 사람마다 그것을 표현하는 표정이 다르기 때문에 실제 얼굴 각 부분(눈썹, 눈과 입의 윤곽, 코의 위치 등)의 윤곽선을 그리는 것만으로는 일반적인 감정 상태를 표현하기 어렵다. 이를 위해 본 연구에서는 표정을 몇 개의 스트로크를 이용하여 얼굴의 주요한 부분을 상징적으로 표현하는 것으로 정의하고 오른쪽 눈, 왼쪽 눈, 입, 3개의 선으로 표현하기로 하였다.

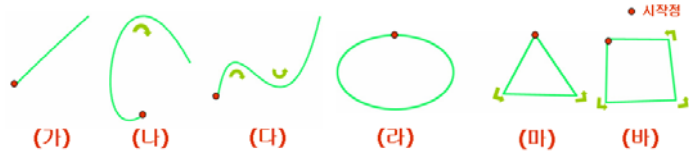
이렇게 3개의 선으로 표현되는 얼굴의 표정에 대한 인식 역시 사람마다 다르다. 예를 들어 화가 난 표정에 대해서 양쪽 눈과 입을 그려서 표현하라고 했을 때, 그 결과는 사람에 따라 다르게 나타날 수 있다. 그래서 실제로 사용자가 이러한 스케치 인터페이스를 사용하도록 하려면 사용 전에 미리 각자 자기만의 스트로크로 각 표정을 정의하는 과정이 필요하다. 사용자는 이 시스템의 데이터베이스에 저장되어 있는 몇 가지 다른 표정을 가진 얼굴 모델에 대해서 그 모델을 3개의 선으로 표현해서 정의내리는 과정을 거치게 된다. 이 과정을 통해 이후에 이 사용자가 키-프레임을 생성하고자 할 때 원하는 얼굴 표정을 미리 정의내린 스트로크로 표현하면, 시스템은 데이터베이스로부터 가장 잘 매치되는 얼굴 모델을 찾아주게 된다.

4.2 스케치 분석

사용자의 스케치를 분석하기 위해, 우리는 하나의 스케치가 두 가지 특징을 가진 것으로 상정하였다. 첫 번째 특징은 사용자가 그린 스케치의 근사곡선으로 스케치의 대략적인 형태를 표현한다. 두 번째 특징은 이 근사곡선을 따라 나타나는 스케치의 섬세한 부분으로 스케치의 세부적인 형태를 나타낸다.



[그림 3] 스케치는 대략적인 근사 곡선과 그 근사곡선을 따라 나타나는 세부적인 모양을 특징으로 갖는다.



[그림 4] 대략적인 형태의 종류, (가)직선 (나)한번꺾임 (다)두번꺾임 (라)급변점 없음 (마)두개의 급변점 (바)세개의 급변점

[그림 3]에서 보이는, 스케치의 두 가지 특징은 사람이 매우 쉽고도 직관적으로 구별해 낼 수 있다. 이렇게 하나의 스케치로부터 서로 다른 두 가지 형태를 추출하고, 각 특징을 매개화하는 값을 찾아내어 그 특징을 규정함으로써 우리는 한 스케치에서 두 가지 의미 단위를 분석, 비교할 수 있는 가능성을 제공한다.

4.2.1 대략적 형태 인식

우리는 사용자가 그린 스케치의 대략적인 형태를 얻기 위해 최소제곱 곡선근사를 이용하였다. 이 곡선근사를 위하여 3차 비-스플라인(B-Spline)을 사용하였기 때문에, 스케치는 [그림 4]의 (가)직선형태, (나)한번 꺾이는 곡선형태, (다)두 번 꺾이는 곡선 형태 중 하나의 형태로 근사된다.

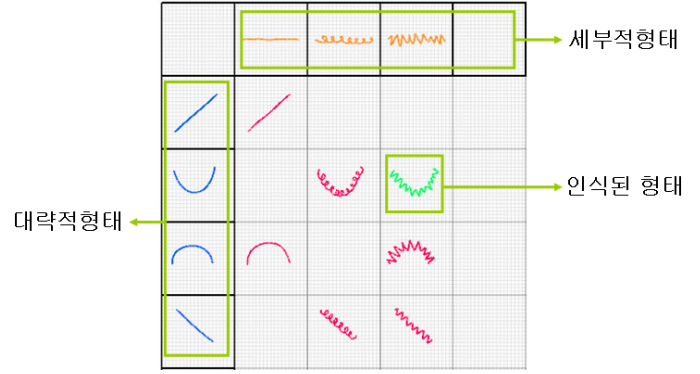
스케치의 근사곡선을 위의 세 가지로 분류하기 위해 곡선을 일정 간격으로 분절하고, 각 분절된 구간의 양끝을 시작점과 끝점으로 갖는 연속된 벡터의 집합을 구한다. 그리고 이 벡터들이 곡선을 따라 변화하는 추이로부터 곡선이 현재 꺾여지고 있는 방향을 알아낸다. 꺾여지는 방향은 연속된 두 벡터의 외적을 구한 뒤, 외적 벡터의 방향을 이용하여 판별하였다.

이렇게 구해진 근사곡선은 그 특징을 규정하는 값으로 4개의 비-스플라인(B-Spline) 제어점을 사용한다. 단, 예외적으로 직선의 경우는 제어점이 아닌 기울기로 규정한다. 이 제어점은 근사곡선의 크기에 의존하지 않고 그 형태에만 의존하기 위해 정규화 된다. 여기서 사용된 정규화는, 제어점들의 중점을 찾고 그 중점에서 제어점들까지의 거리의 합이 1이 되도록 하는 방법을 사용하였다.

곡선근사에 의해 대략적인 형태를 얻기 힘든 원, 삼각형, 사각형과 같은 스케치는 미리 정의된 규칙에 의해 판별된다.

4.2.2 세부적 형태 인식

사용자가 그린 스케치의 세부형태곡선은 스케치된 선과 그 근



[그림 5] 가로축은 세부적 형태, 세로축은 대략적 형태인 테이블에 사용자가 입력한 스케치가 분류된 결과

사곡선의 변위차로 정의된다. 즉, 근사곡선의 관점에서 바라본 스케치의 형태가 스케치의 세부형태이다. 세부형태곡선을 구하기 위해 우리는 아래와 같은 벡터를 스케치로부터 얻는다. 스케치가 사용자로부터 총 n 개의 점을 입력받아 만들어진 것이라 가정하면

$$\overrightarrow{V_i} = \overrightarrow{P_i P_{i+1}} \quad (P_i \text{는 스케치의 } i\text{번째 점}, 1 \leq i \leq n-1)$$

인 벡터 $n-1$ 개를 얻을 수 있다. 그리고 P_i 에서 근사곡선에 수선을 내려 만난점의 단위접선벡터를 $\overrightarrow{T_i}$, 그리고 접선벡터로부터 반시계방향으로 $\frac{\pi}{2}$ 만큼 회전한 단위법선벡터를 $\overrightarrow{N_i}$ 라 하면, 세부형태의 좌표 (X_i, Y_i) 는 다음과 같다. 여기서, $\|\overrightarrow{T_i}\| = \|\overrightarrow{N_i}\| = 1$ 이다.

$$X_i = \sum_{k=1}^i (\overrightarrow{V_k} \cdot \overrightarrow{T_k}), \quad Y_i = \sum_{k=1}^i (\overrightarrow{V_k} \cdot \overrightarrow{N_k})$$

모든 세부형태는 다른 세부형태와의 차이점을 비교하기 위해 그 특징을 규정하는 곡률의 도수분포로 표현되는데,

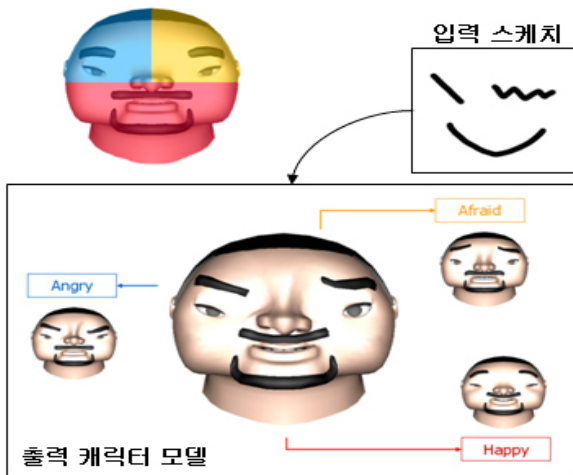
$$\overrightarrow{U_i} = (X_{i+1} - X_i, Y_{i+1} - Y_i)$$

라고 정의하면, 이 도수분포표의 곡률은 세부형태곡선의 연속된 벡터 $\overrightarrow{U_i}$ 에서 $\overrightarrow{U_{i+1}}$ 로의 회전각으로 표현된다. 회전각은 다음과 같이 정의된다.

$$C_i = \tan^{-1} \left(\frac{(\overrightarrow{U_i} * \overrightarrow{U_{i+1}}) \cdot (0,0,1)}{\overrightarrow{U_i} \cdot \overrightarrow{U_{i+1}}} \right) \quad (\text{단, } \|\overrightarrow{U_i}\| = \|\overrightarrow{U_{i+1}}\| = 1)$$

(회전각 C_i 는 $-\pi \leq C_i \leq \pi$ 인 범위를 갖고, 반시계 방향일 때 양수)

이렇게 구해진 전체 회전각들의 빈도는 30개의 등분된 구간을 갖는 도수분포표를 구성하는데, 여기서 i 번째 구간의 도수는 다음과 같이 정규화된다.



[그림 6] (좌측 위) 하나의 얼굴 모델을 세가지 색으로 구분되는 세 개의 영역으로 나눈다. (아래) 서로 다른 표정으로 정의된 세 개의 스트로크 입력에 대해 세 가지 표정이 혼합된 새로운 캐릭터 모델이 생성된다.

$$N_i = \frac{F_i}{\sum_{k=1}^{30} F_k} \quad (\text{단, } 1 \leq i \leq 30)$$

N_i : i 번째 구간의 도수

F_i : i 번째 구간에 해당하는 회전의 빈도

이렇게 만들어진 두 도수분포표 H_A 와 H_B 의 거리차이를 구하는 식은 다음과 같다.

$$Dist(H_A, H_B) = \sum_{i=2}^{29} \{0.2(|N_i^A - N_{i-1}^B|) + 0.6(|N_i^A - N_i^B|) + 0.2(|N_i^A - N_{i+1}^B|)\}^2$$

N_i^A : H_A 의 i 번째 구간의 도수

N_i^B : H_B 의 i 번째 구간의 도수

위의 거리함수는 i 번째 구간의 도수를 비교할 때, $i-1, i+1$ 번째 구간에도 가중치를 주어 계산하므로 오차에 강인해진다.

하나의 스케치는 [그림 5]에서와 같이, 대략적인 형태를 행으로 하고 세부적인 형태를 열로 하는 표에 분류된다. 이 표는 사용자가 등록한 스케치로부터 대략적인 형태와 세부적인 형태를 얻어내어 분류기준을 만든다. 이 표를 이용하여 우리는 같은 대략적인 형태를 가지지만 다른 세부 형태의 스케치, 또는 같은 세부 형태지만 다른 대략적인 형태를 갖는 스케치들의 관계를 체계적으로 분류할 수 있다.

4.3 키-프레임 캐릭터 모델 생성

우리의 시스템은 하나의 키-프레임 캐릭터를 생성하기 위하여 얼굴 모델을 세 부분으로 나누어서 사용한다. 사용자의 스케치 입력이 양쪽 눈과 입, 3개의 스트로크로 구성되었듯이 얼굴 모



[그림 7] (좌측) 바이콘 광학 시스템을 이용하여 얼굴 캡처를 하는 장면이다. (우측) 얼굴 전면에 68개의 마커를 머리위에 7개의 마커를 붙여서 얼굴의 움직임을 캡처한다.

델 또한 [그림 6]에서처럼 왼쪽 눈, 오른쪽 눈, 입 세 개의 영역으로 분리된 형태로 각각의 데이터베이스를 형성한다. 이렇게 각 영역과 표정에 따른 분리된 모델들을 가지고 있음으로써, 사용자 입력인 세 개의 스트로크가 미리 정의된 얼굴 표정 중에서 서로 다른 세 개의 표정을 표현하더라도 그에 맞는 얼굴 모델을 생성하는 것이 가능하게 된다. 세 개의 얼굴 영역에서 서로 다른 표정을 표현 했을 경우, 각 얼굴 영역의 데이터베이스로부터 각각의 스트로크에 가장 잘 매치되는 세 개의 얼굴 모델을 찾는다. 그 다음 세 영역의 서로 다른 모델들을 다시 하나의 얼굴 모델로 결합시킴으로써 여러 가지 표정들이 섞여 있는 새로운 얼굴 모델의 표현이 가능하게 된다.

이러한 방식을 통해 실제 가지고 있는 각 한 가지 표정만을 표현하는 얼굴 모델들을 이용하여 그 이상의 다양한 표정을 나타내는 새로운 모델들을 만들어 내는 효과를 얻을 수 있다.

5. 얼굴 캡처 데이터 기반 키-프레임 보간법

우리는 기본적인 애니메이션 생성에 있어서 혼합 모형 모델을 이용한다. 혼합 모형 모델은 기본이 되는 몇 개의 혼합 모형 모델에 가중치를 적용시켜 더함으로써 새로운 얼굴 모델을 만들어 낸다.

$$B_{\text{new}} = N + \sum_{i=1}^n \alpha_i (N - B_i)$$

N : 무표정 얼굴 모델

B_i : 감정을 가진 얼굴 모델

n : 감정을 표현하는 얼굴 모델 개수

α_i : 서로 다른 얼굴 모델을 혼합할 때의 가중치

위와 같이 새로운 얼굴 모델을 만드는 것은 얼굴의 세 영역에 독립적으로 적용되며 각 영역별로 만들어진 세 개의 얼굴 모델들이 영역별로 합쳐져서 하나의 완성된 모델이 만들어진다.

스케치 인터페이스를 이용하여 만들어진 키-프레임의 얼굴 모델들은 키-프레임 사이를 보간함으로써 하나의 완성된 애니메이션 클립이 된다. 보간에 있어서 가장 일반적인 보간법인 선형 보간법은 두 개의 키-프레임 양쪽의 키-모델에 대해 각 모델을 구성하는 수치들의 차를 보간해야 할 프레임의 길이로 나

뉘서 그 값만큼을 매 프레임마다 증가시키는 방식이다. 각 모델을 구성하는 수치는 대표적으로 캐릭터 모델을 구성하는 각 점의 3차원 좌표로 볼 수 있으며 사람 몸 캐릭터의 경우는 각 관절의 위치와 각도, 혼합 모형 모델의 경우는 모델의 혼합 가중치 등이 될 수 있다. 이런 보간법을 통해, 키-프레임 사이의 프레임들이 시작 지점부터 끝 지점까지 일정한 속도로 모델의 모양이 바뀌는 형태로 채워진다. 그러나 이것은 키-프레임간의 간격이 넓을 경우 단조롭고 부자연스러운 애니메이션을 생성하는 문제가 있어서, 자연스러운 애니메이션을 위해서는 조밀한 키-프레이밍이 필요하다.

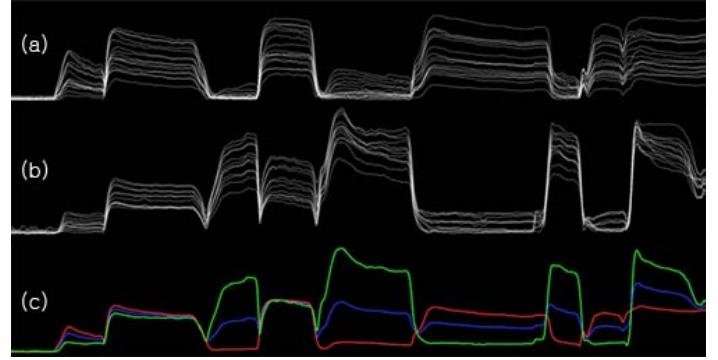
이러한 문제점을 보완하기 위해 우리는 데이터 기반 보간법을 제안한다. 이 방법은 사람의 얼굴 움직임을 캡처한 데이터로부터 실제 사람이 얼굴 표정을 바꿀 때의 전이 정보를 추출하여 이것을 키-모델간의 보간시에 가중치로 이용하는 것이다. 이로써 적은 개수의 키-프레임으로도 사실적이고도 자연스러운 애니메이션이 생성될 수 있다. 이후에는 실제 사람의 얼굴 캡처 데이터를 얻고 그 데이터로부터 얼굴 표정간의 전이 정보를 추출하는 방법과 추출된 전이 정보를 이용하여 키-프레임을 보간하는 알고리즘을 설명할 것이다.

5.1 얼굴 캡처 및 표정간의 전이 정보 추출

다양한 표정을 가진 얼굴 모델의 키-프레임 보간을 위해서는 실제 사람이 표정을 바꿀 때 얼굴 전면의 움직임에 대한 정보가 필요하다. 본 연구에서는 이러한 실제 사람 얼굴의 데이터를 얻기 위해 마커 기반의 모션 캡처를 이용하였다. [그림 7]에서 보이는 바와 같이 사람의 얼굴 전반에 마커를 붙이되, 표정 변화에 영향을 많이 미칠 것이라고 생각되어 지는 눈썹과 입 주변에 더 많은 마커를 붙여서 68개의 마커를 얼굴 표면에 붙였다. 또 머리의 전역적인 움직임을 얻기 위해 머리 윗부분에 7개의 마커를 추가하여 붙임으로써 총 75개의 마커로 얼굴의 움직임을 캡처하였다. 이렇게 캡처된 데이터는 얼굴에 붙어있었던 마커들의 3차원 좌표 형태로 저장되어서 매 프레임마다의 얼굴 전면의 절대적 위치를 표현해 준다.

이 데이터를 키-프레임의 보간에 이용하려면 서로 다른 표정으로 전이 할 때의 시간에 따른 변화 정보를 추출해야 한다. 우리는 애니메이션 생성에 혼합 모형 모델을 이용하기 때문에, 비어있는 프레임들을 채우기 위해서 각 프레임에서 혼합 모형의 혼합 가중치를 설정해 주어야 한다. 또한 이 혼합 가중치는 각 프레임마다 하나의 값이므로, 캡처 데이터로부터 추출하는 전이 정보 역시 각 프레임마다 1차원 데이터 형태이어야 한다.

현재 필요한 데이터는 시간에 따른 상대적 변화 정도이므로 각 데이터의 절대적인 수치가 아닌 기준 모델로부터의 상대적 값을 알아야 한다. 이것을 위해 기준 모델을 무표정한 사람의 얼굴 상태로 정하고, 모든 프레임에 대해서 각 마커의 변화량을 다음과 같이 정의하였다.



[그림 8] 가로축 : 시간(프레임), 세로축 : 현재 시간의 표정에서의 마커와 무표정 상태의 마커와의 기하학적 거리 값
(a) 눈 위쪽 영역의 마커 16개에 대한 그래프
(b) 눈 아래쪽 영역의 마커 12개에 대한 그래프
(c) 빨간색 - 눈 위쪽 영역 마커들의 평균값
초록색 - 눈 아래쪽 영역 마커들의 평균값
파란색 - 전체 마커의 평균값
빨간색과 초록색의 그래프는 구간별로 같은 형태를 나타내거나, 상하 대칭의 형태를 나타낸다.

$$D_j = \sqrt{(x_j - x_{Nj})^2 + (y_j - y_{Nj})^2 + (z_j - z_{Nj})^2}$$

D_j : j 번째 마커의 변화량

x_j, y_j, z_j : j 번째 마커의 x, y, z 좌표

x_{Nj}, y_{Nj}, z_{Nj} : 무표정 상태의 j 번째 마커의 x, y, z 좌표

이러한 마커의 변화량은, 얼굴을 눈 위쪽과 아래쪽 두 영역으로 나누었을 때 같은 영역 내의 마커들끼리는 비슷한 움직임을 보였다. 또한 한 영역 내에 속한 전체 마커들의 변화량의 평균값을 그 영역을 대표하는 변화량으로 보았을 때, 눈 위쪽과 아래쪽 영역의 시간에 따른 변화량은 구간별로 같은 형태를 가지거나 상하 대칭을 이루었다. 같은 형태인 것과 상하 대칭인 것 모두 시간에 대해서는 변화의 정도가 일정한 것이므로 이 영역별 변화량의 평균값을 얼굴 표정의 변화를 표현하는 전이 정보로 정의한다. ([그림 8] 참조)

$$w = \frac{1}{k} \sum_{i=1}^k \left(\frac{1}{n_i} \sum_{j=1}^{n_i} D_{ij} \right)$$

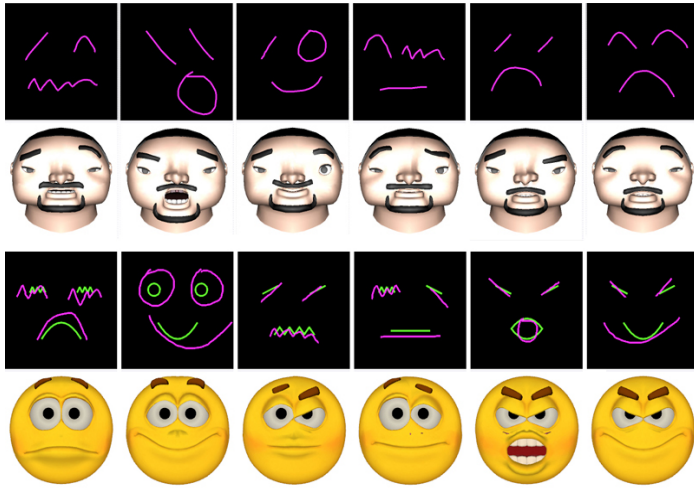
w : 전이 정보 값

D_{ij} : i 번째 영역의 j 번째 마커의 변화량

n_i : i 번째 영역내의 마커 개수

k : 얼굴 영역의 개수

본 시스템에서는 얼굴을 눈 위쪽과 아래쪽 2개의 영역으로 나누고, 각 영역내의 마커는 특별히 얼굴 표정에 영향을 미친다고 생각되는 16개와 12개의 마커를 선택하여 이용하였다. 이 과정을 통해 총 28개 마커의 3차원 좌표값인 84차원 데이터를 목표하였던 매 프레임에 대한 1차원 데이터 형태로 차수를 감소시킬 수 있게 된다.



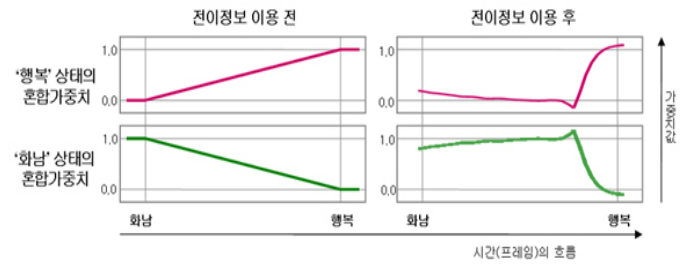
[그림 9] 사용자의 스케치 입력에 의해 생성되는 얼굴 모델들은 양쪽 눈과 입이 서로 다른 감정을 표현하는 얼굴도 표현해 준다.

5.2 데이터 기반 키-프레임 보간법

본 장에서는 얼굴 캡처 데이터로부터 추출한 표정간의 전이 정보를 실제로 키-프레임 보간에 적용시키는 방법에 대해 논의한다. 앞서 설명했듯이 이 시스템의 애니메이션 생성 방식은 혼합 모형 모델이므로, 각 키-프레임 사이의 혼합 가중치를 결정해 주면 그 가중치에 따라 키-모델들을 혼합하여 프레임들이 채워지며 완성된 애니메이션이 생성된다.

캡처 데이터로부터 추출한 전이 정보는 프레임에 따라 얼굴의 변화 정도를 표현하는 1차원 벡터 형태로 저장되어 있다. 이 데이터의 특징은 그 값이 시간의 변화에 따라 긴 구간 일정한 정도를 유지하다가 중간 중간 급격한 변화를 보이는 구간이 있다는 것이다. 바로 이런 변화의 정도가 급격한 구간이 실제로 얼굴 표정의 전이가 일어난 시간이다. 데이터를 분석한 결과 얼굴 표정의 전이는 오랜 시간에 걸쳐서 일정한 비율로 일어나는 것이 아니라 아주 짧은 간격의 시간 동안에(1초 이내) 급격한 변화로 일어나는 것을 발견할 수 있었다. 이러한 특성 때문에 서로 다른 표정의 키-프레임을 선형 보간법으로 보간했을 경우 부자연스러운 애니메이션이 생성되는 것이다.

전이 정보를 가져와서 적용시킬 목표 데이터가 키-프레임 형태이므로 전이 정보 역시 키-프레임에 기반하여 가공해야 한다. 먼저 변화량의 급격한 증감이 일어나는 시점, 즉 표정간의 전이가 일어나는 시점을 기준으로 데이터를 자른다. 이렇게 잘라 놓은 데이터의 각 조각은 한번 표정의 전이가 일어난 후 제 자리를 찾은 시점부터 다음의 표정 전이가 일어나는 시점까지의 정보를 가지고 있다. 각 조각의 시작지점의 변화량을 0으로 끝지점의 변화량을 1로하고, 이것을 조각의 길이로 나누어 비례 크기 조절을 한 수치를 혼합 가중치로 이용한다. 이 혼합 가중치들을 키-모델들을 혼합 할 때의 가중치로 이용하여 혼합 모형모델을 만듦으로써 캡처 데이터로부터 얻은 전이 정보가



[그림 10] 혼합 모형 모델에 쓰일 혼합 가중치 값이다. 가로축 : 시간(프레임), 세로축 : 가중치값, (좌) 전이정보 이용 전 - 가중치가 단조로운 형태로 변한다. (우) 전이정보 이용 후 - 실제 사람의 움직임 데이터에서 추출한 전이 정보형태대로 가중치가 변화한다.

미리 만들어져있는 3차원 얼굴 모델에 매핑되게 된다.

이 조각을 실제 사용할 때에 이러한 각 조각의 프레임 길이와 만들고자 하는 애니메이션의 키-프레임 사이의 길이가 맞지 않는 문제점이 있다. 이것은 전이 정보를 가지고 있는 데이터 조각의 앞부분은 특별한 변화 없이 비슷한 정도의 값이므로 이 부분을 늘리거나 줄여서 목표 애니메이션 키-프레임의 길이에 맞추는 방식으로 해결하였다.

6. 실험결과

실험에 사용된 3차원 얼굴 모델은 전문 애니메이터에 의해 제작된 것으로 감정을 표현하기 위해 6가지 표정에 대한 얼굴 모델과, 말하는 애니메이션을 만들기 위해 8개의 입모양에 대한 얼굴 모델을 이용하였다. 또한 기본적으로 전체 애니메이션은 이 14개의 얼굴 모델의 혼합 모형 방식에 의해 초당 30프레임으로 생성되었다.

얼굴 캡처 데이터는 바이콘 광학 시스템에서 초당 120프레임으로 캡처하였고 캐릭터 애니메이션의 초당 프레임수에 맞추기 위해 30프레임으로 다운 샘플링하여 사용하였다. 총 75개의 마커를 사용하였으며 출력되는 얼굴 동작 데이터는 각 마커의 절대적 3차원 위치를 가지고 있다.

[그림 9]은 사용자의 여러 가지 스케치에 대해 만들어진 얼굴 모델이다. 사용자의 세 개의 스트로크가 서로 다른 세 개의 표정을 각각 표현할 때에 본 시스템은 그에 맞는 새로운 얼굴 모델을 생성해주었다. 또한 키-프레임의 보간에 있어서는 캡처 데이터로부터 추출한 전이 정보를 이용함으로써 [그림 10]에서 보이는 것처럼 시간에 따른 혼합 가중치가 선형의 단조로운 형태에서 전이 정보에 따른 비선형의 곡선을 이루게 됨으로써, 더 자연스럽고 사실적인 얼굴 애니메이션을 생성하였다. 본 시스템을 이용하여 한편의 애니메이션을 만들기 위해 실행되는 모든 과정은 실시간으로 진행된다.

7. 결론 및 향후연구

스케치 인터페이스는 누구나 쉽게 직관적으로 사용 가능하다. 이러한 장점에 따라 전문 애니메이터가 아닌 일반 사용자도 본 시스템을 이용해서 쉽게 원하는 얼굴 애니메이션을 만들 수 있다. 뿐만 아니라 스케치의 결과를 검색하거나 보관하는 과정 또한 자동적으로 실시간에 실행됨으로써 짧은 시간에 용이하게 자연스러운 얼굴 애니메이션 생성이 가능하다.

본 연구의 한계점으로는 스케치로부터 3차원의 모델을 만들어 내는 과소결정 문제를 캐릭터 모델로 구성된 데이터베이스로부터 스케치에 가장 잘 맞는 모델을 검색해서 찾아오는 방식으로 해결하기 때문에, 생성되는 애니메이션의 질이 데이터베이스의 양에 제한을 받는다는 것이 있다.

이러한 한계점을 극복하기 위한 향후 연구로는 본 시스템에서처럼 개략적인 스케치로 찾아낸 감정을 표현한 얼굴 모델을 기본으로 하여, 이것에 덧붙여서 눈썹의 위치를 바꾼다거나 입의 벌리는 정도를 조절하는 것 등을 스케치 입력으로 표현하는 시스템을 개발하는 것이 있다. 이를 통해 한정된 데이터베이스에서 나온 제한된 개수의 얼굴 모델을 사용자별로 섬세한 부분을 조절할 수 있게 함으로써 키-모델 생성의 자유도를 증가시킬 수 있다.

또한 하나의 스케치는 대략적인 형태와 세부적인 형태로 나뉘어져 각각의 특징을 표현한다는 것을 이용하여 우리는 대략적인 형태가 같고 세부 형태가 다른 스케치, 또는 세부 형태가 같고 대략적인 형태가 다른 스케치로 분류할 수 있다. 이를 이용하여 사용자가 같은 웃는 입모양을 그리더라도 세부 형태의 변화를 주는 것으로 캐릭터의 성별이나 나이를 규정하도록 하는 인터페이스 개발이 가능하다.

참고문헌

[1] Blanz, V., Vetter, T., A morphable model for the synthesis of 3-D Faces, ACM Transactions on Graphics (SIGGRAPH 1999), 187-194
[2] Pyun, H., Kim, Y., Chae, W., Kang, H. W., Shin, S. Y., An example-based approach for facial expression cloning, ACM Transactions on Graphics (SIGGRAPH 2006), 23
[3] Chai, J., Xiao, J., Hodgins, J., Breen, D. Vision-based Control of 3D Facial Animation, Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2003, 193-206
[4] Sifakis, E., Selle, A., Robinson-Mosher, A., Simulating Speech with a Physics-Based Facial Muscle Model, Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006, 261-270
[5] Lorenzo, M. A. S., Edge, J. D., King, S. A., Maddock,

S. C., Use and Re-use of Facial Motion Capture Data, Vision, Video and Graphics 2003, 135-142
[6] Noh, J., Neumann, U., Expression cloning, SIGGRAPH 2001, 277-288
[7] Liu, Z., Shan, Y., Zhang, Z., Expressive expression mapping with ratio images, ACM Transactions on Graphics (SIGGRAPH 2001), 271-276
[8] Byun, M., Badler, N., FACEMOTE: Qualitative Parametric Modifiers for Facial Animations, ACM Transactions on Graphics (SIGGRAPH 2002), 65-72
[9] Pyun, H., Kim, Y., Chae, W., Kang, H. W. & Shin, S. Y., An example-based approach for facial expression cloning, ACM Transactions on Graphics (SIGGRAPH 2006), 23
[10] Ezzat, T., Geiger, G. & Poggio, T., Trainable videorealistic speech animation, ACM Transactions on Graphics (SIGGRAPH 2002), 21, 388-398
[11] Cao, Y., Tien, W. C., Faloutsos, P. & Pighin, F., Expressive speech-driven facial animation, ACM Transactions on Graphics (SIGGRAPH 2005), 24, 1283-1302
[12] Vlasic, D., Brand, M., Pfister, H. & Popovic, J., Face transfer with multilinear models, ACM Transactions on Graphics (SIGGRAPH 2005), 24, 426-433
[13] Deng, Z. & Neumann, U., eFASE: Expressive Facial Animation Synthesis and Editing with Phoneme-Isomap Controls, Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2006, 251-259
[14] Kovar, L., Gleicher, M. & Pighin, Motion Graphs, ACM Transactions on Graphics (SIGGRAPH 2002), 473-482
[15] Lee, J., Chai, J., Reitsma, P. S. A., Hodgins, J. K. & Pollard, N. S., Interactive control of avatars animated with human motion data, ACM Transactions on Graphics (SIGGRAPH 2002), 21, 491-500
[16] Davis, J., Agrawala, M., Chuang, E., Popović, Z. & Salesin, D., A Sketching Interface for Articulated Figure Animation, Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2003, 320-328
[17] Li, Y., Gleicher, M., Xu, Y. & Shum, H., Stylizing Motion with Drawings, Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation, 2003, 309-319
[18] Thorne, M., Burke, D. & van de Panne, M., Motion doodles: an interface for sketching character motion, ACM Transactions on Graphics (SIGGRAPH 2004), 23, 424-431