

CCA 투영기법을 사용한 모션 데이터의 대화식 얼굴 표정 애니메이션

Interactive Facial Expression Animation of Motion Data using CCA

김 성 호*
Sung-Ho Kim

요 약

본 논문은 다량의 고차원 얼굴 표정 모션 데이터를 2차원 공간에 분포시키고, 애니메이터가 이 공간을 항해하면서 원하는 표정들을 실시간적으로 선택함으로써 얼굴 표정 애니메이션을 생성하는 방법을 기술한다. 본 논문에서는 약 2400여개의 얼굴 표정 프레임을 이용하여 표정공간을 구성하였다. 표정공간의 생성은 임의의 두 표정간의 최단거리의 결정으로 귀결된다. 표정공간은 다양체 공간으로서 이 공간내의 두 점간의 거리는 다음과 같이 근사적으로 표현한다. 임의의 마커 간의 거리를 표시하는 거리행렬을 사용하여 각 표정의 상태를 표현하는 표정상태벡터를 정의한 후, 두 표정이 인접해 있으면, 이를 두 표정 간 최단거리(다양체 거리)에 대한 근사치로 간주한다. 그리하여 인접 표정들 간의 인접거리가 결정되면, 이들 인접거리들을 연결하여 임의의 두 표정 상태간의 최단거리를 구하는데, 이를 위해 Floyd 알고리즘을 이용한다. 다차원 공간인 표정공간을 가시화하기 위해서는 CCA 투영기법을 이용하여 2차원 평면에 투영시켰다. 얼굴 애니메이션은 사용자 인터페이스를 사용하여 애니메이터들이 2차원 공간을 항해하면서 실시간으로 생성한다.

Abstract

This paper describes how to distribute high multi-dimensional facial expression data of vast quantity over a suitable space and produce facial expression animations by selecting expressions while animator navigates this space in real-time. We have constructed facial spaces by using about 2400 facial expression frames on this paper. These facial spaces are created by calculating of the shortest distance between two random expressions. The distance between two points in the space of expression, which is manifold space, is described approximately as following ; When the linear distance of them is shorter than a decided value, if the two expressions are adjacent after defining the expression state vector of facial status using distance matrix expressing distance between two markers, this will be considered as the shortest distance (manifold distance) of the two expressions. Once the distance of those adjacent expressions was decided, We have taken a Floyd algorithm connecting these adjacent distances to yield the shortest distance of the two expressions. We have used CCA(Curvilinear Component Analysis) technique to visualize multi-dimensional spaces, the form of expressing space, into two dimensions. While the animators navigate this two dimensional spaces, they produce a facial animation by using user interface in real-time.

Keyword : facial expression animations, expression state vector, distance matrix, manifold distance, Floyd algorithm, CCA.

1. 서론

인간은 자신의 감정을 말보다는 얼굴 표정을 통해서 더 잘 표현하며, 상대방의 얼굴 표정을 보고 상대방의 현재 감정 상태를 파악한다. 그런 연

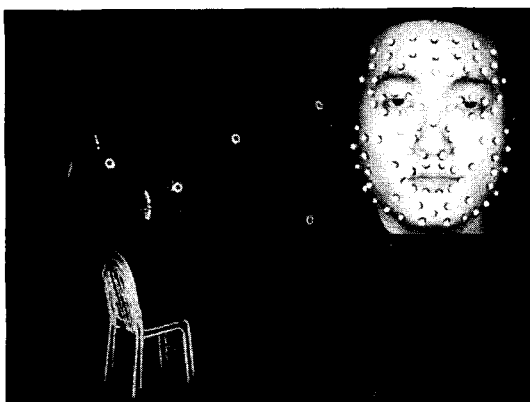
유로 지금까지 3차원 컴퓨터 그래픽스 기법을 통해서 인간의 얼굴 표정을 표현할 수 있는 많은 방법이 연구 [1-11]되었다. 최근에는 모션 캡처를 사용한 캐릭터 애니메이션이 컴퓨터 애니메이션 분야에서 각광받으면서 배우의 얼굴 모션을 캡처하여 3차원 캐릭터 애니메이션에 적용하고 있다. 캡처한 데이터를 사용하는 방법으로 크게 두 가지가 제시되었다. 첫째는 배우의 모션 데이터를

* 정 회 원 : 호서대학교 컴퓨터공학부 게임공학전공
시간 강사

kimsh1204@hotmail.com(제 1저자)

[2004/01/13 투고 - 2004/01/18 심사 - 2004/07/20 심사 완료]

새로운 모델에 재적용하는 모션 리타겟팅 기법 [10,12]이다. 모션 리타겟팅 기법은 먼저 몸동작에 대한 것이 제시[12]되었고, 얼굴 표정에 대한 것도 제시[10]되었다. 들쭉는 배우의 동작을 가능한 한 많이 캡처하여 이 동작 속에 들어있는 자세들을 획득, 자세 데이터베이스를 만든 후, 애니메이터가 특정 자세들을 선택, 연결하여 새로운 동작을 생성하는 기법이다. 이 기법도 몸동작에 대한 것이 먼저 제시[12]되었는데, 아직 얼굴 표정에 대한 것은 제시된 것이 없다.



(그림 1) 마커 100개 부착위치 및 광학식 모션 캡처 시스템을 사용한 얼굴 모션 캡처 장면

그러므로 본 논문에서는 다량의 얼굴 모션 캡처 데이터를 직관적인 공간에 분포시키고, 애니메이터가 적당한 공간을 향해하면서 원하는 얼굴 표정들을 실시간으로 선택하여, 표정 애니메이션을 생성하는 방법을 기술한다. 먼저 얼굴 표정을 전문적으로 연출하는 배우의 도움을 받아 그림 1과 같이 광학식 모션 캡처 시스템을 사용하여 얼굴 표정을 캡처한다. 표정을 캡처할 때, 배우는 얼굴 주 근육 부분에 작은 반사 마커 100개를 부착한다. 그런 다음 배우로 하여금 수개의 얼굴 모션을 연출하게 하고, 초당 60 프레임으로 캡처한다. 한 개의 마커는 3차원 좌표 값으로 표현되므로 100개의 마커위치로 표현되는 한, 표정은 300차원의 데이터이다. 얼굴 표정 데이터는 3차원 공

간상의 위치 값으로만 구성되어 있기 때문에, 얼굴 표정들 사이의 유사성을 수치적으로 표현하기가 어렵다. 얼굴 표정들 사이의 유사성은 300차원의 데이터를 2차원이나 3차원과 같은 저차원 평면상에 분포시킬 경우, 얼굴 표정들 사이의 거리 값으로 구분하기 때문에 필요하다. 그러므로 다양한 얼굴 표정들을 잘 구분할 수 있도록 하기 위해서 마커들 사이의 거리를 이용한 벡터행렬로 변경하고, 이를 Curvilinear Component Analysis (이하 'CCA') 투영 기법[13,14]으로 2차원에 투영시킨다. 실시간 표정 애니메이션을 생성하기 위해서는 애니메이터로 하여금 투영된 2차원 공간을 향해하게 하고, 향해경로의 각 점에 해당되는 얼굴표정을 3차원 얼굴 모델을 이용하여 디스플레이 한다.

2. 얼굴 표정상태 표현법

표정공간을 생성하기 위해서는 각각의 얼굴 표정상태를 수치적으로 표현해야 한다. 표정상태는 얼굴에 부착된 마커들의 위치에 의해서 결정된다. 표정상태의 표현은 표정들 간의 상대적인 거리 관계를 잘 표현하는 것이어야 하고, 이로써 얼굴 표정을 구분할 수 있어야 한다. 표정의 상태를 표현하는 가장 간단한 방법은 마커들의 위치들로 이루어진 상태벡터를 이용하는 것이다. 본 논문에서 사용한 얼굴 모션 프레임 데이터는 100개의 마커를 사용하고, 한 마커는 3차원 좌표를 가지기 때문에, 표정상태벡터는 300차원이 된다. 이런 식으로 표현된 표정상태벡터를 '위치벡터'라고 하자. 본 논문에서는 표정상태를 표현할 때, 위치벡터를 사용하지 않는다. 대신 위치벡터의 임의의 두 마커간의 거리를 표현하는 "거리행렬"을 이용하여 표정상태를 표현한다. 왜냐하면 거리행렬이 위치벡터보다 얼굴 마커들의 분포상태에 대한 정보를 더 많이 표시하고 있고, 따라서 두 표정간의 거리를 보다 더 정확하게 표현할 수 있기 때문이다. 두 위치벡터를 비교할 때, 각 마커의 위치들

간의 거리를 구함으로써 두 표정간의 거리를 표현한다. 이때 각 마커의 표정간의 거리는 다른 마커들과 관계없이 계산한다. 이에 비해, 거리행렬로 표정상태를 표현하면, 두 표정간의 거리를 구할 때 마커들 간의 상호관계가 자연스럽게 고려된다. 물론 표정의 거리행렬은 표정의 위치벡터로부터 구할 수 있지만, 위치벡터를 사용하는 경우, 이 위치벡터에 내포되어 있는 정보를 상태간의 거리를 구하거나 할 때 사용하지 않기 때문에, 유용한 정보를 내포하고 있다는 것이 도움이 되지 않는다. 따라서 이 정보를 명시적으로 표현하는 거리행렬 방식이 더 좋은 상태표현방식이다. 거리행렬은 위치벡터를 기반으로 다음과 같이 구한다. 즉, 하나의 위치벡터에서 마커와 마커들 사이의 거리를 식 (1)과 같이 직선거리 방법으로 계산한다.

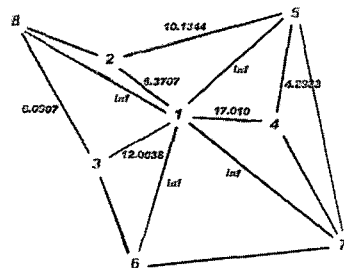
$$MD_{ii,c} = \sum_{i=1}^n \left(\sum_{j=1}^{99} \sum_{k=j+1}^{100} \sqrt{(M_{i,j_1:j_2} - M_{i,k_1:k_2})^2} \right) = \begin{pmatrix} md_{1,1} & \dots & md_{1,c} \\ \dots & \dots & \dots \\ md_{n,1} & \dots & md_{n,c} \end{pmatrix}, \quad ii = 1, \dots, n, \quad c = 1, \dots, 4950 \quad (1)$$

여기서 $MD_{ii,c}$ 는 위치벡터로부터 계산된 거리행렬 $md_{ii,1}, \dots, md_{ii,c}$ 들의 집합이다. ii 는 거리행렬집합의 행의 수이면서 얼굴 모션 프레임의 수 $n(=2400$ 여개)을 의미한다. c 는 마커 100개에 대한 마커들 사이의 거리 계산 회수 $100 \times (100-1)/2 = 4950$ 만큼의 거리 값의 수를 의미한다. M 은 위치벡터들의 집합, i 는 n 개의 위치벡터의 수, j_1, j_2 및 k_1, k_2 는 모션 프레임 데이터에서 프레임별 각 마커들의 3차원 좌표 값으로서 마커와 마커 사이의 거리를 구하기 위해서 사용된다. j 와 k 는 c 를 위한 변수로서, j 는 위치벡터별 100개 마커에서 마커와 마커 사이의 거리를 계산할 때 비교하기 위한 99개의 마커를 의미한다. 그리고 k 는 위치벡터별 마커 j 에 의해서 비교되어지는 100개의 마커를 의미한다.

3. 표정공간의 생성

표정상태를 거리행렬로 표현하면 표정간의 직선거리는 두 거리행렬간의 직선거리로 표현된다. 거리행렬로 표현된 표정상태들의 공간은 임의의 두 거리행렬간의 거리를 결정함으로써 결정된다. 본 논문에서는 거리행렬의 각 행을 죽 나열하여 그 결과를 하나의 벡터로 간주하고, 이 벡터간의 직선거리를 거리행렬간의 직선거리로 사용한다. 즉 임의의 두 마커간의 거리가 서로 비슷한 두 표정은 서로 인접한 표정으로 간주한다.

표정공간은 임의의 두 표정간의 거리를 두 표정거리 행렬간의 직선거리로 정의할 수 있는 벡터공간이 아니다. 한 표정에서 다른 표정으로 옮겨가는 과정은 얼굴의 여러 가지 제약조건으로 말미암아, 복잡한 경로를 거치게 되기 때문이다. 표정공간은 구면처럼 다양체 공간인 것이다. 다양체 공간상에서의 거리는 두 점간의 거리를 한 점에서 이 공간을 벗어나지 않으면서 다른 점까지 도달하는 최단경로의 길이로 정의[16]한다. 본 논문에서는 이 다양체 공간을 근사적으로 표현한다. 이를 위해 먼저, 두 거리행렬간의 직선거리가 일정 값 이하인 경우 이 직선거리가 두 표정간의 최단거리에 대한 근사치라고 간주한다. 이 조건을 만족하는 두 표정을 “인접 표정” 이라고 하는데, 임의의 표정에 대한 인접 표정들은 그림 2와같이 그래프를 형성하여 구한다.



〈그림 2〉 Floyd 알고리즘을 위한 그래프 생성방법 : 1 번을 기준으로 2, 3, 4 번은 인접표정, 5, 6 7, 8 번은 비 인접표정(무한대), 원의 반지름은 인접거리 한계 값

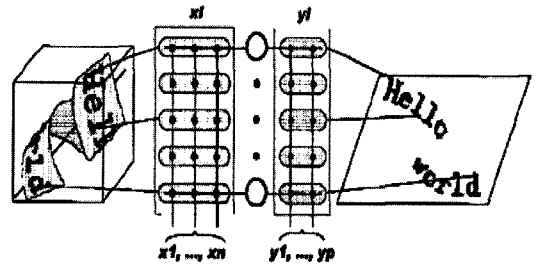
인접표정이 주어지면 한 표정에서 다른 표정까지 바로 이동할 수 있다고 본다. 두 표정이 인접해 있지 않은 경우에 한 표정에서 다른 표정으로 바로 이동할 수 없고, 그 사이에 있는 인접한 표정들을 통해서만 이동할 수 있다고 가정한다. 그러나 인접표정을 결정하는 한계거리를 미리 알기는 쉽지 않다. 따라서 이는 실험을 통해서 좋은 결과를 내는 한계 값을 정한다. 인접표정을 결정하는 인접거리 임계값은 실험에 의해 최적의 값을 결정해야 한다. 다만, 인접거리 임계값은 임의의 표정에서 다른 임의의 표정으로 이동하는 데 필요한 충분한 수의 인접표정들이 나오도록 설정되어야 한다. 인접표정들이 결정되면, 인접하지 않은 두 표정상태간의 거리는 그 사이에 있는 인접 표정들 간의 거리들을 합하여 구한다. 이를 위해, 최단거리를 구하는 알고리즘인 Floyd 알고리즘(다이내믹 프로그래밍 기법) [16]을 이용한다. 이렇게 임의의 두 표정간의 최단거리가 구해지면 해당 다양체 공간이 결정된다. 본 논문에서는 약 2400여개의 얼굴표정을 사용하여 다양체 공간을 형성하였다.

4. CCA 투영

앞에서 생성한 거리행렬 방식의 표정상태 공간은 300차원 공간이다. 따라서 이 공간을 애니메이션이 행해지면서 원하는 표정을 선택할 수는 없다. 그러므로 원래 표정공간의 구조를 근사적으로 표현하는 2차원 또는 3차원으로 공간을 구하여 이 공간을 향해하는 방법을 사용한다. 본 논문에서는 이를 위해 CCA 투영 기법[13,14]을 사용한다. CCA 투영은 고차원 데이터들 사이의 거리가 주어지면, 이 거리들의 분포를 보존하면서 2차원이나 3차원 등의 저차원 공간에 투영시킨 좌표들의 집합을 구하는 방법들 중의 하나이다. 이때 이 좌표들의 차원은 필요에 따라 미리 정하는데, 본 연구에서는 구한 좌표들을 시각적으로 표현해야 되기 때문에, 2차원 좌표를 사용했다.

표정상태를 나타내는 n 개의 다차원 다양체 공간상의 점들을 $x_i, i=1, \dots, n$, 임의의 표정상태 x_i 와 x_j 사이의 다양체 거리를 $d_{ij}, i, j=1, \dots, n$ 이라고 하고, 이를 행렬로 나타낸 것을 행렬 $D=d_{ij}$ 라고 하자. 행렬 D 는 대각 원소들이 0인 대칭행렬이고, 이와 같은 행렬을 비유사성 대칭행렬이라고 한다. 다차원 표정공간을 근사적으로 나타내는 2차원 평면상의 점들을 $y_i, i=1, \dots, n$, 2차원 점 y_i 와 y_j 간의 거리를 δ_{ij} 이라고 하자. CCA 투영을 적용하면, 2차원 거리들의 집합 $\{\delta_{ij}\}$ 가 다차원 다양체 거리들의 집합 D 와 가장 근사한 분포를 가지도록 하는 2차원 평면상의 점들의 집합 $\{y_i\}$ 을 구할 수 있다.

CCA 투영은 그림 3과 같이 벡터 양자화(Vector Quantization)와 비선형 투영(Nonlinear Mapping)의 두 단계의 층(Layer)으로 분리하여 수행한다.



(그림 3) CCA 투영의 네트워크 구조 벡터 양자화(x_i)와 비선형 투영(y_i)

첫째 단계인 벡터 양자화는 하부 다양체(Sub-manifold) 공간을 형성하기 위한 것이다. 일반적으로 양자화는 신호처리나 영상 압축에서 사용되는 용어로서, 실제의 입력신호는 연속적인 실수 값을 갖는데, 이러한 값을 그대로 표현하고자 한다면 데이터의 양이 무한하게 늘어나게 되어 실제 데이터를 처리하는 과정에서는 메모리나 속도 문제를 수반하게 될 것이다. 따라서 데이터의 양을 유한한 개수의 이산적인 값으로 표현해

야 할 필요가 생기는데, 이를 양자화라고 한다. 그러나 본 논문에서는 300차원의 데이터가 자체적으로 의미가 있기 때문에 양자화 과정을 거치지 않는다.

둘째 단계인 비선형 투영은 p 차원(< 300)의 출력 공간상의 양자화 된 벡터들의 비선형 투영으로부터 원하는 집합 $\{y_i\}$ 를 얻는 방법이다. 여기서 p 는 상태(Configuration) 벡터의 차원이며, 2차원이나 3차원 등의 차원을 결정하는 미리 정해진 숫자이다. 비선형 투영은 최적화 문제로 귀결되며, 다음 식 (2)와 같다.

$$E_{CCA} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (d_{ij} - \delta_{ij})^2 F(\delta_{ij}, \lambda_y) \quad (2)$$

여기서 $F(d_{ij}, \lambda_y)$ 는 공간 위상 보전을 강조하는 가중치 함수로서, 다음 식 (3)과 같이 구분하여 적용한다.

$$F(\delta_{ij}, \lambda_y) = \begin{cases} 1 & \text{if } \delta_{ij} \leq \lambda_y \\ 0 & \text{if } \delta_{ij} > \lambda_y \end{cases} \quad (3)$$

식 (2)의 최적화 문제는 변수 λ_y 의 초기값이 필요하며, 본 논문에서는 다음 식 (4)와 같이 다차원 다양체 공간상의 점 x_i 의 표준편차를 사용한다.

$$3 \times \max \left(\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i^2 - \bar{x}_i^2)} \right) \quad (4)$$

또한 최적화 문제는 변수 $\{y_i\}$ 의 초기 좌표 값 δ_{ij}^0 을 필요로 한다. 일반적으로 변수 $\{y_i\}$ 의 초기 좌표 값 δ_{ij}^0 은 임의로 설정하거나 입력 데이터인 행렬 D 의 Principal Component Analysis(PCA) [15]를 적용하는데, 본 논문에서는 임의로 설정한다. 변수 $\{y_i\}$ 의 초기 좌표 값 δ_{ij}^0 을 입력으로 사용한 최적화는 다음 식 (5)와 같이 최적화 함수가 최소가 될 때까지 집합 $\{y_i\}$ 를 반복적으로 갱신한다.

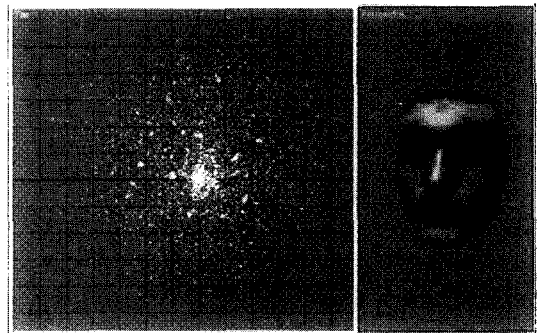
$$\Delta y_i = \alpha(t) F(\delta_{ij}, \lambda_y) \frac{(d_{ij} - \delta_{ij})}{\delta_{ij}} (y_j - y_i), \forall j \neq i \quad (5)$$

여기서 $\alpha(t) = \alpha_0 / (1+t)$ 는 시간과 동시에 줄어드는 학습 비율이며, 초기 값으로 0.5를 사용한다.

본 논문에서는 2차원 평면상의 점들의 집합 $\{y_i\}$ 을 구하기 위해 Matlab V6.5를 사용하였다. Matlab 함수 CCA에 표정상태벡터들로 구성된 집합 x_i , 차원의 수 2 및 비유사성 대칭행렬 D 를 입력으로 사용하여 2차원 평면상의 점들의 집합 $\{y_i\}$ 를 구하였다.

5. 사용자 인터페이스 및 실험

본 논문에서는 약 2400여개의 표정으로 구성된 표정공간을 생성한 후에 이를 CCA 투영기법을 통해 2차원 공간으로 투영하고, 애니메이터로 하여금 이 공간을 향해하면서 얼굴애니메이션을 생성하게 하였다. 이를 위한 사용자 인터페이스는 그림 4와 같다.

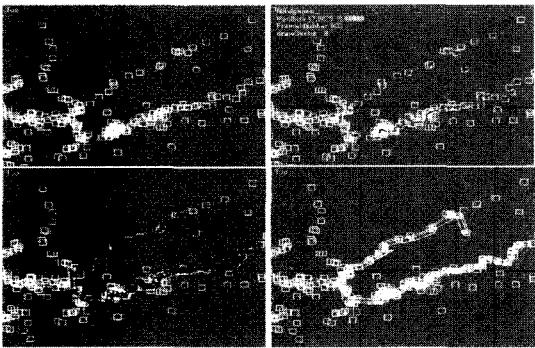


(그림 4) 사용자 인터페이스(좌: 향해 공간, 우: 3차원 얼굴 모델), 애니메이터가 향해공간에 분포된 각 얼굴 표정상태를 대표하는 작은 점을 마우스로 선택하면서 항해를 하면, 선택된 점에 해당되는 표정상태가 3차원 얼굴 모델에 나타난다.

그림 4의 왼쪽은 약 2400여개의 표정상태벡터를 2차원 평면에 분포시킨 향해 공간이고, 오른쪽

의 3차원 얼굴 모델은 애니메이터가 향해 공간을 향해할 때 실시간으로 얼굴표정을 보여주게 한다.

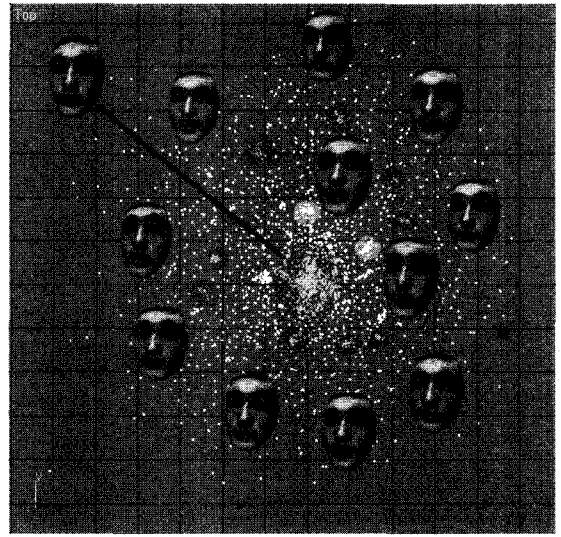
그림 5는 CCA 투영기법에 의해서 생성된 약 2400여개의 2차원 표정공간을 애니메이터가 향한 경로를 표시한 것이다. 애니메이터는 마우스를 사용하여 향해 공간을 향해하고 동시에 3차원 얼굴 모델에 적용된 얼굴 표정을 보게 되는데, 이때에는 표정 하나하나를 향해 과정과 향해 속도에 따라 확인하면서 보게 된다.



〈그림 5〉 좌상: 2차원 표정공간의 점들, 우상: 향해과정, 빨간색 점들이 향해 경로, 좌하: 라인으로 연결된 향해 경로, 우하: 적십자는 반복 재생 시 재생중인 현재 프레임

애니메이터의 향해 과정이 끝나면 애니메이터의 향해경로에 해당되는 얼굴 표정들을 연속적으로 보고 얼굴 표정의 변화를 확인할 필요가 있다. 이때에는 애니메이터의 향해경로를 처음부터 끝까지 자동으로 반복 향해하여 연속된 얼굴 표정의 변화를 3차원 얼굴 모델이 보여주게 된다. 그림 5 우하의 적십자는 이때의 향해경로를 따라가는 현재의 경로 위치를 가리켜주기 위한 지시자이다. 본 사용자 인터페이스는 애니메이터가 그림 5와 같은 향해경로를 다수 개 생성하여 서로 다른 색상으로 표현할 수 있으며, 애니메이터가 향한 경로들 중 임의의 경로 하나를 선택하여 재생하고 확인하여 수정할 수 있다. 그림 6은 애니메이터가 모든 2차원 표정공간을 향해하면서 마우스로 선택한 경로들 상에 있는 대표적인 얼굴

표정들을 표시한 것으로서, 원래 얼굴모델은 그림 4와 같이 사용자 인터페이스의 오른쪽 창에 실시간으로 표시되지만, 설명의 편리를 위해 향해공간에 표시하였다. 그림 6은 거리행렬 방식에 의해 표현된 표정상태를 사용한 결과로서 애니메이션을 생성하기에 매우 효율적으로 분포되어져 있었다.



〈그림 6〉 애니메이터가 모든 2차원 표정공간을 향해하면서 마우스로 선택한 경로들 상에 있는 대표적인 얼굴표정들을 설명의 편리를 위해 향해공간에 표시함

거리행렬 방식에 의해 표현된 표정상태를 사용한 실험에서는 인접표정을 결정하는 인접거리 임계값을 실험에 의해 최적의 값을 결정해야 한다. 본 논문에서는 임계값을 정할 때 다음과 같은 기준을 적용하였다. 인접거리 임계값에 따라 300차원 다양체 공간에서의 표정 최단거리 분포와 2차원 공간에서의 표정 최단거리 분포 사이의 상관도가 달라진다. 이때 가능하면 두 분포 사이의 상관도가 높게 나오는 인접거리 임계값을 사용하는 것이 좋다. 2차원 공간은 다차원 공간을 근사적으로 표현하는 것이므로 두 공간 사이의 상관도가 높을수록 좋다. 이 기준을 적용하기 위하여 다차원 최단거리 분포와 2차원 최단거리 분포사이의

상관도는 피어슨(Pearson)의 상관계수, r [17]을 이용하여 구하였다.

다양체 공간상의 임의의 표정상태 x_i 와 x_j 사이의 다양체 거리를 하나의 벡터로 표현하고, 이를 V_d 라고 하자. 2차원 평면에 투영된 임의의 점 y_i 와 y_j 사이의 최단거리를 하나의 벡터로 표현하고, 이를 V_y 라고 하자. 상관계수 r 은 $-1 \leq r \leq 1$ 의 범위 값을 가지며, 벡터 V_d 와 벡터 V_y 을 이용하여 다음 식 (6)과 같이 계산되어진다.

$$r = \frac{\sum_{i=1}^n (v_{di} - \bar{V}_d)(v_{yi} - \bar{V}_y)}{\sqrt{\sum_{i=1}^n (v_{di} - \bar{V}_d)^2 \cdot \sum_{i=1}^n (v_{yi} - \bar{V}_y)^2}} \quad (6)$$

여기서 $v_{di} \in V_d$, $v_{yi} \in V_y$, \bar{V}_d 는 V_d 의 평균, \bar{V}_y 는 V_y 의 평균을 의미한다. 일반적으로 상관계수 r 의 값이 $0.40 \leq r \leq 0.70$ 의 범위를 가지게 되면, 비교적 높은 상관관계를 가진다고 하고, 상관계수 r 은 $r \geq 0.90$ 일 경우는 매우 높은 상관관계를 가지며, $r=1.0$ 일 때에는 일치한다. 본 논문에서는 가장 높은 상관관계를 가지는 인접거리 임계값을 찾는다. 실험 결과 가장 높은 상관관계는 $r=0.5482$ 이며, 이때의 인접거리 임계값은 30mm이다. 매우 높은 상관관계를 가지는 상관계수 $r \geq 0.90$ 인 경우는 CCA의 변수 중 훈련 길이(Training length)의 값을 높게 변형함으로써 나타나기는 하였으나, 인접표정들이 제대로 표현되지 않고 도넛 모양의 공백이 많은 상태로 분포가 되어 본 논문의 목적에 맞지 않아 사용하지 않았다. 그림 6은 인접거리 임계값 30mm를 이용하여 구성한 2차원 최단 거리 분포이다. 두 공간간의 상관도를 비교적 높게 만드는 인접거리 임계값(15mm에서 40mm까지)을 사용했을 때, 임의의 두 표정 간의 경로가 존재하지 않는 경우는 없었다. 그러나 두 공간간의 상관도를 낮게 만드는 상관계수($r=0.0281$ 이하)일 때의 인접거리 임계값(10mm 이하)을 사용할 때는 경로가 존재하지 않

는 표정들이 많아진다. 이것은 인접거리 임계값을 작게 잡으면 임의의 두 표정사이에 인접표정들을 거쳐서 가는 최단 경로를 구할 때, 최단경로가 존재하지 않는 경우가 많이 생기기 때문이다.

본 논문에서는 인접거리 임계값을 30mm로 주었을 때 생성되는 2차원 거리 분포(그림 6)를 사용하여 애니메이션을 생성하였다. 본 논문에서는 자세히 언급하지 않았지만, 위치벡터로 표정상태를 표현하는 경우, 위에서 언급한 결과들이 나오지 않았다. 즉, 두 공간간의 상관도를 가장 높게 만드는 인접거리 임계값을 사용했을 때, 임의의 두 표정 사이의 경로가 존재하지 않는 경우가 있었다.

6. 결론

본 논문에서는 다량의 얼굴 모션 데이터들을 적당한 공간에 분포시키고, 애니메이션이 이 공간을 향해하면서 원하는 얼굴 표정들을 실시간으로 선택하고 디스플레이 하는 방법을 기술하였다. 약 2400여개의 얼굴 표정 프레임을 사용하여 향해 공간을 구성하였으며, 이 공간을 구성하기 위해서는 거리행렬들의 집합을 CCA 투영기법에 적용하여 계산하였다. 실시간 얼굴 표정 애니메이션을 생성하기 위해서 개발한 사용자 인터페이스는 표정상태 표현법을 위한 위치벡터 및 거리행렬 방식을 실험적으로 확인하는데 유용하게 사용되었다. 사용자 인터페이스는 애니메이션이 생성하고자 하는 얼굴 표정 애니메이션을 직관적인 공간을 자유자재로 향해하면서 실시간으로 생성, 확인, 수정 및 재생성이 가능하기 때문에 유용하고 효율적이라는 것을 애니메이션으로 하여금 확인할 수 있었다. 본 논문의 연구 결과는 얼굴 표정 애니메이션 생성과 관련된 기존의 관련 연구와 비교할 때, 모션 데이터를 사용한 3차원 아바타의 얼굴 표정 애니메이션을 실시간으로 생성하면서 디스플레이할 수 있고, 반복 재생이 가능하다는 것이 가장 큰 특징이다. 예를 들어 문자 위주의

채팅이나 메시지를 사용하여 상대방과 의사소통을 하는 도중에 자신의 감정을 3차원 아바타를 사용하여 실시간적으로 실감나게 표현해줄 필요가 있는데, 이런 경우 본 논문의 연구 결과는 매우 적합하다. 그러므로 본 논문의 연구결과는 3차원 아바타의 실시간 얼굴 표정 애니메이션 생성이 필요한 각종 응용 프로그램 분야에서 매우 유용하게 사용될 것이다.

참고 문헌

- [1] Demetri Terzopoulos, Barbara Mones-Hattal, Beth Hofer, Frederic Parke, Doug Sweetland, Keith Waters, "Facial animation : Past, present and future," Panel, SIGGRAPH97, 1997.
- [2] Frederic I. Parke, Keith Waters. "Computer facial animation," A K Peters, 1996.
- [3] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," Proc. ACM SIGGRAPH95 Conf., pp.55-62, 1995.
- [4] T. DeRose, M. Kass and T. Truong, "Sub-division surfaces in character animation," in Computer Graphics Proceedings, ACM SIGGRAPH, July 1998, Annual Conference Series, pp. 85-94.
- [5] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Frederic Pighin. "Making Faces," In SIGGRAPH 98 Conference Proceedings. ACM SIGGRAPH, July 1998.
- [6] Won-Sook Lee, Prem Kalra, Nadia Magnenat Thalmann, "Model based face reconstruction for animation," Proc. MMM'97 (World Scientific Press), Singapore, pp.323-338, 1997.
- [7] Won-Sook Lee, Nadia Magnenat-Thalmann, "From real faces to virtual faces: Problems and solutions," Proc. 3IA'98, Limoges (FRANCE), pp.5-19, 1998.
- [8] Won-Sook Lee, Nadia Magnenat-Thalmann, "Fast head modeling for animation," Journal Image and Vision Computing, Volume 18, Number 4, pp.355-364, Elsevier, 1 March, 2000.
- [9] Frederic Pighin, Jamie Hecker, Dani Lischinski, Richard Szeliski, David H. Salesin, "Synthesizing realistic facial expressions from photographs," ACM SIGGRAPH98, 1998.
- [10] Cyriaque Kouadio, Pierre Poulin, and Pierre Lachapelle, "Real-time facial animation based upon a bank of 3차원 facial expressions," Proc. Computer Animation 98, June 1998.
- [11] Sannier G, Magnenat-Thalmann N., "A flexible texture fitting model for virtual clones," Proceedings of Computer Graphics International, IEEE Computer Society, pp. 167-176, 1997.
- [12] Jeehee Lee, Jinxiang Chai, Paul Reitsma, Jessica Hodgins, and Nancy Pollard, "Interactive Control of Avatars Animated with Human Motion Data", ACM Transactions on Graphics (SIGGRAPH 2002), volume 21, number 3, 491-500, July 2002.
- [13] P. demartines, and J. Herault, "Curvilinear Component Analysis: A Self-Organizing Neural Network for Nonlinear Mapping of Data Sets", IEEE Trans. Neural Networks. vol. 8, NO 1, 1997.
- [14] J. A. Lee, A. Lendasse, N. Donckers, and M. Verleysen, "A Robust Nonlinear Projection Method", In M. Verleysen, editor, Proceedings of ESANN'2000.
- [15] Hotelling, H., "Analysis of a complex of

- statistical variables into principal components", Journal of Educational Psychology 24, pp 417-441, 1933.
- [16] R. W. Floyd, "Algorithm 97 : Shortest Path," CACM Vol.5, pp. 345, 1962.
- [17] Uprendra Shardanand, "Social information filtering for music recommendation", Master's thesis, MIT, 1994.

● 저 자 소개 ●



김 성 호(Sung-Ho Kim)

1996년 상지대학교 전산학과 졸업(학사)
1998년 숭실대학교 대학원 컴퓨터학과 졸업(석사)
2005년 숭실대학교 대학원 컴퓨터학과 졸업(박사)
1997년 1월~2000년 12월 숭실대학교, 숭의여대 시간강사
2000년 1월~2001년 1월 (주)조오그래프 선임연구원
2000년 6월~2001년 9월 (주)애니큐브 부설 연구소장
2002년 3월~2005년 2월 숭의여대 IT계열 멀티미디어콘텐츠전공 겸임교수
2003년 9월~현 재 (주)포도시스템즈 연구실장
2005년 3월~현 재 부천대학 정보통신과 겸임교수
2005년 3월~현 재 호서대학교 컴퓨터공학부 게임공학전공 시간강사
관심분야 : 컴퓨터그래픽스, 컴퓨터 애니메이션, 모션 캡처, 가상현실, 웹 3D, 멀티미디어 등.
E-mail : kimsh1204@hotmail.com