
모션 데이터에 Isomap을 사용한 3차원 아바타의 실시간 표정 제어

Realtime Facial Expression Control of 3D Avatar by Isomap of Motion Data

김성호

상지대학교 컴퓨터정보공학부

Sung-Ho Kim(kimsh1204@sangji.ac.kr)

요약

본 논문은 Isomap 알고리즘을 사용하여 다량의 고차원 얼굴 모션 데이터를 2차원 평면에 분포시키는 방법론과, 사용자가 이 공간을 항해하면서 원하는 표정들을 선택함으로써 실시간으로 얼굴 표정 제어가 가능한 사용자 인터페이스 기법에 대하여 기술한다. Isomap 알고리즘은 세 단계의 과정으로 처리된다. 첫째, 각 표정 데이터의 인접표정을 정의하고, 둘째, 각 표정들 사이의 다양체 거리를 계산하여 표정공간을 구성한다. 표정공간의 생성은 임의의 두 표정간의 최단거리(다양체 거리)의 결정으로 귀결되고, 이를 위해 플로이드 알고리즘을 이용한다. 셋째, 다차원 표정공간을 가시화하기 위해서 다차원 스케일링을 사용하며, 2차원 평면에 투영시킨다. 인접표정을 정의하기 위한 최소 인접거리는 피어슨의 상관계수를 이용한다. 3차원 아바타의 얼굴 표정 제어는 사용자 인터페이스를 사용하여 2차원 공간을 항해하면서 실시간으로 제어한다.

■ 중심어 : | 얼굴 모션 캡처 | 얼굴 표정 상태 벡터 | 플로이드 알고리즘 | 실시간 표정 제어 | Isomap |

Abstract

This paper describe methodology that is distributed on 2-dimensional plane to much high-dimensional facial motion datas using Isomap algorithm, and user interface techniques to control facial expressions by selecting expressions while user navigates this space in real-time. Isomap algorithm is processed of three steps as follow; first define an adjacency expression of each expression data, and second, calculate manifold distance between each expressions and composing expression spaces. These facial spaces are created by calculating of the shortest distance(manifold distance) between two random expressions. We have taken a Floyd algorithm for it. Third, materialize multi-dimensional expression spaces using Multidimensional Scaling, and project two dimensions plane. The smallest adjacency distance to define adjacency expressions uses Pearson Correlation Coefficient. Users can control facial expressions of 3-dimensional avatar by using user interface while they navigates two dimension spaces by real-time.

■ keyword : | Facial Motion Capture | Expression State Vector | Floyd Algorithm | Realtime Expression Control | Isomap |

* 본 논문은 2006년도 상지대학교 교내 연구비 지원에 의해 수행되었습니다.

접수번호 : #061031-003

심사완료일 : 2007년 02월 21일

접수일자 : 2006년 10월 31일

교신저자 : 김성호, e-mail : kimsh1204@sangji.ac.kr

I. 서론

인간은 원래 자신의 내면 감정 상태를 언어로 표현하기도 하지만 얼굴 표정을 통하여 더욱더 잘 표현하며, 상대방의 얼굴 표정을 보고 감정 상태를 쉽게 파악할 수 있다. 그런 이유로 컴퓨터 그래픽스와 컴퓨터 애니메이션 분야뿐만 아니라 인간의 얼굴 표정을 필요로 하는 가상 현실 등의 많은 분야에서는 지금까지 인간의 다양한 얼굴 표정을 가상공간에서 잘 표현할 수 있는 많은 방법들이 연구[1-7]되어 왔었다. 특히 최근에는 모션 캡처 시스템을 사용한 캐릭터 애니메이션이 컴퓨터 애니메이션 분야에서 각광받으면서 배우의 얼굴 표정을 캡처하여 3차원 캐릭터 애니메이션에 적용하고 있다. 지금까지 모션 캡처한 데이터를 사용하는 방법으로 크게 두 가지가 제시되었다. 첫째는 배우의 모션 데이터를 새로운 모델에 재적용하는 모션 리타겱팅 기법[6][8]이다. 모션 리타겱팅 기법은 먼저 몸동작에 대한 것이 제시[8]되었고, 얼굴 표정에 대한 것도 제시[5][6]되었다. 특히 [5]에서는 얼굴 모션 캡처 데이터를 3차원 혼합형 얼굴에 매핑시킴으로써 대중적인 3차원 혼합형 얼굴 애니메이션을 직접적으로 만드는 반자동화 기법을 소개하고 있다. 그러나 [5]는 얼굴 모션 데이터를 혼합형 얼굴 모델에 매핑시킴으로써 실험에 사용한 모션 데이터만큼의 표정 애니메이션을 반자동적으로 만들 수 있다는 것에 한계를 두고 있다. 둘째는 배우의 몸동작이나 표정을 가능한 다양하게 대량으로 캡처하여 이 동작들이나 표정들 속에 들어있는 포즈들을 획득, 포즈 데이터베이스를 만든 후, 사용자가 특정 포즈들을 선택, 연결하여 새로운 동작을 생성하는 기법이다. 이 기법도 몸동작에 대한 것이 먼저 제시[8]되었고, 아직 얼굴 표정에 대한 것은 제시된 것이 없다. 최근 고차원 몸동작 데이터를 저차원 공간에 분포시킨 다음 애니메이션이 저차원 공간의 커브를 스케치하고 편집할 수 있고, 3차원 공간상의 캐릭터 포즈를 직접적으로 조작할 수 있거나 키 포즈를 설정해줌으로서 중간 모션들을 생성할 수 있다는 연구가 제시[13]되었으나 역시 얼굴 표정에 대한 것은 아니다. 그러므로 본 논문에서는 다량의 얼굴 모션 캡처 데이터를 2차원 평면에 분포시키고, 사용자가 적당한 공간을 향해하면서 실험에 사용한 모션

데이터보다 훨씬 많은 다양한 얼굴 표정들을 실시간으로 선택하여 원하는 표정을 제어하기 위한 방법을 기술한다. 먼저 얼굴 표정을 전문적으로 연출하는 배우의 도움을 받아 [그림 1]과 같이 광학식 모션 캡처 시스템을 사용하여 얼굴 표정을 캡처한다. 표정을 캡처할 때, 배우는 얼굴 주 근육 부분에 작은 반사 마커 100개를 부착한다. 100개 마커의 위치는 MPEG-4에서 정한 얼굴 특징점 위치 정보를 기반으로 25개를 같은 위치로 설정하고, 나머지 75개는 저자가 정밀한 표정을 캡처하기 위해 얼굴 표정을 결정짓는 대표적인 얼굴 근육 부위에 추가로 부착한 것이다. 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. 여기서 말하는 차원은 구면과 같은 다양체 공간에서의 차원으로서 그것이 몇 개의 수의 조(組)로 정의되는가에 따라 결정된다. 2차원 평면에서 원은 그 중심의 좌표(x, y)와 반지름 r 과의 조, 즉 좌표(x, y, r)를 가진 점으로도 나타낼 수 있고 이를 3차원 데이터라고 할 수 있다. 3차원 공간(x, y, r)과 시간(t)에 의하여 생각한 공간은 그 대상을 좌표(x, y, z, t)인 점으로 생각하면, 4차원이 된다. 그러므로 본 논문에서 사용하는 얼굴 모션 데이터는 각 프레임 당 100개 마커의 3차원 좌표 값으로 이루어진 좌표($x_1, y_1, z_1, \dots, x_{100}, y_{100}, z_{100}$)을 점으로 생각하면 300차원이 된다.



그림 1. 광학식 모션 캡처 시스템을 사용한 얼굴 모션 캡처 장면 및 배우의 얼굴에 부착한 마커 100개 부착위치

얼굴 표정 데이터는 3차원 공간상의 위치 값으로만 구성되어 있기 때문에, 얼굴 표정들 사이의 유사성을 수치적으로 표현하기가 어렵다. 얼굴 표정들 사이의 유사성은 300차원의 데이터를 2차원이나 3차원과 같은 저차원 평면에 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 Isometric feature mapping(이하 'Isomap')[11][12]로 2차원 평면에 투영시킨다. 실시간 표정 제어를 위해서는 사용자로부터 투영된 2차원 공간을 항해하게 하고, 항해경로의 각 점에 해당되는 얼굴 표정을 3차원 아바타의 얼굴 모델을 사용하여 디스플레이 한다.

II. 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 위치 값들로 이루어진 벡터를 이용하는 것이다. 본 논문에서 사용한 얼굴 모션 프레임 데이터는 100개의 마커를 사용하고, 한 마커는 3차원 좌표를 가지기 때문에, 벡터는 300차원이 된다. 이런 식으로 표현된 벡터를 '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. 원래 하나의 위치벡터로부터 계산된 하나의 거리행렬은 마커의 수 100개 사이의 임의의 거리를 계산하는 것으로서 100×100 형식의 행렬로 표현해야 한다. 그러나 거리행렬로 표현된 표정상태들 사이의 거리를 구하기 위한 사전 과정으로서 본 논문에서

는 거리행렬의 각 행을 죽 나열하여 그 결과를 하나의 벡터로 설정한다. 이때 임의의 두 마커간의 거리를 표현할 때, 중복되는 거리 표현은 생략한다. 그러므로 거리행렬의 각 행을 죽 나열한 하나의 벡터는 $100 \times 99 / 2 = 4950$ 차원으로 구성되며, 본 논문에서는 이를 '표정상태벡터'라고 한다.

III. Isomap (Isometric feature mapping)

앞에서 생성한 거리행렬 방식의 표정상태 공간은 4950차원 공간이다. 따라서 이 공간을 사용자가 항해하면서 원하는 표정을 선택하기란 불가능하다. 그러므로 원래 4950차원 표정공간의 구조를 근사적으로 표현하는 2차원 또는 3차원으로 공간을 구하여 이 공간을 항해하는 방법을 사용한다. 본 논문에서는 2차원 공간을 생성하고, 이를 위해 Isomap 알고리즘을 사용한다. Isomap 알고리즘은 고차원 데이터들 사이의 거리가 주어지면, 거리들의 분포를 보존하면서 2차원이나 3차원 등의 저차원 공간에 투영시킨 좌표들의 집합을 구하는 방법 중의 하나이다. Isomap 알고리즘은 다음과 같이 세 단계의 과정을 거쳐서 진행된다. 첫째, 앞에서 구한 각 얼굴 표정상태벡터로부터 표정상태벡터간의 거리가 일정한 거리 이내에 있는 인접표정을 정의하고, 둘째, 다양체 공간에서 모든 표정상태벡터들 사이의 다양체 거리(원래 Isomap 알고리즘에서는 직선거리를) 계산하여 표정공간을 생성하고, 셋째, 표정공간을 다차원 스케일링(Multidimensional Scaling, 이하 'MDS')[11][12]에 적용하여 2차원 평면에 투영한 2차원 좌표를 계산하는 것이다. Isomap의 셋째 단계는 고차원 공간을 저차원 공간으로 차원 축소하여 투영하는 방법 중의 하나로서, 여기에는 주성분분석(Principal Component Analysis, 이하 'PCA'), 다차원 스케일링, LLE(Locally Linear Embedding) 등이 있다. 주성분분석은 투영된 분산을 최대화하는 인접 구조 보존 방식으로서 역투영이 가능한 방법이고, LLE는 모든 데이터 점으로부터 가장 가까운 인접 데이터의 수를 임의로 정하고 이를 보존하는 인접 데이터 보존 방식이다. 그리고 MDS는 데이터 점들 사이의 상호 거리를 보존하기

위해 사용하는 거리 보존 방식으로서, PCA처럼 역 투영이 되지는 않는다. 그러나 본 논문의 목적상 역 투영은 불필요하기 때문에 언급하지 않는다. 그러므로 본 논문에서는 표정들 사이의 최단거리를 보존하면서 2차원 공간에 투영하기 위해 MDS 기법을 사용한다.

일반적인 Isomap 알고리즘의 첫째 및 둘째 단계는 다음과 같은 과정을 거친다. 표정상태를 표정상태벡터로 표현하면 표정 간의 직선거리는 두 표정상태벡터간의 직선거리로 표현된다. 표정상태벡터로 표현된 표정상태들의 공간은 임의의 두 표정상태벡터간의 거리를 계산함으로써 결정된다. 여기서 임의의 표정상태벡터간의 거리가 매우 가까운 두 표정은 서로 인접한 표정으로 간주한다. 그러나 본 논문에서 언급한 표정공간은 구면(球面)처럼 다양체 공간이다. 왜냐하면 한 표정에서 다른 표정으로 옮겨가는 과정은 얼굴의 여러 가지 제약조건으로 말미암아 복잡한 경로를 거치게 되기 때문이다. 그러므로 표정공간은 임의의 두 표정간의 거리를 두 표정상태벡터간의 직선거리로 정의할 수 있는 벡터공간이 아니다. 다양체 공간상에서의 두 점간의 거리는 직선거리가 아니라 한 점에서 이 공간을 벗어나지 않으면서 두 점 사이에 존재하는 점들을 거쳐서 다른 점까지 도달하는 최단경로의 길이로 정의한다.

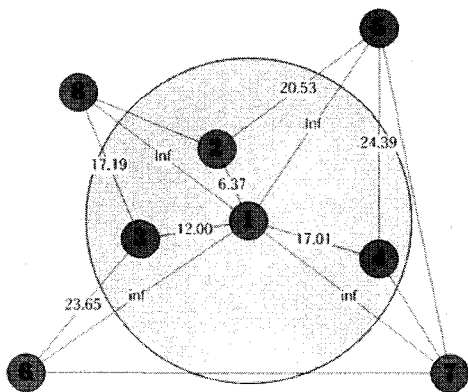


그림 2. 플로이드(Floyd) 알고리즘을 위한 그래프 생성 방법 : 인접표정간의 거리가 주어진 8개의 상태를 가정, 1번 상태를 기준으로 2, 3, 4번 상태는 1번 상태와 인접, 나머지는 그렇지 않다. 인접하지 않은 두 상태간의 거리는 무한대(inf, 한 상태에서 다른 상태까지 바로 이동할 수 없음을 의미)로 설정. 원의 반지름은 임의의 인접거리 한계 값을 나타낸다.

본 논문에서는 이 표정공간을 근사적으로 표현한다. 이를 위해 먼저, 두 표정상태벡터간의 직선거리가 일정 값 이하인 경우 이 직선거리가 두 표정간의 최단거리에 대한 근사치라고 간주한다. 이 조건을 만족하는 두 표정을 ‘인접표정’이라고 하는데, 임의의 표정에 대한 인접표정들은 [그림 2]와 같이 그래프를 형성하여 구한다. 즉, [그림 2]와 같이 표정상태벡터 1번을 기준으로 2, 3, 4 번은 인접표정으로 간주하고, 5, 6, 7, 8 번은 비 인접표정으로 간주하여 무한대(inf) 거리 값을 부여해 준다. 그리고 표정상태벡터 1번을 원점으로 설정한 원의 반지름은 인접거리 한계 값으로 설정한다. 인접표정이 주어지면 인접거리 영역 내에서는 한 표정에서 다른 표정까지 바로 이동할 수 있다고 본다. 두 표정이 인접해 있지 않은 경우에는 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 표정상태벡터간의 거리는 그 사이에 있는 인접 표정들 간의 거리들을 합하여 구한다. 이를 위해 본 논문에서는 최단거리를 구하는 다이내믹 프로그래밍 기법 중 시간 복잡도 $O(n^3)$ 을 가지는 플로이드 알고리즘[9]을 이용한다. 이렇게 임의의 두 표정간의 최단거리가 구해지면 해당 다양체 공간이 결정된다. 본 논문에서는 2400여개의 표정상태벡터를 사용하여 다양체 공간을 형성하였다.

4950차원의 표정상태벡터를 나타내는 n 개의 다차원 다양체 공간상의 점들을 $x_i, i = 1, \dots, n$, 임의의 표정상태벡터 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. Isomap의 마지막 세 번째 단계인 MDS 기법을 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. MDS는 고차원 데이터들 사이의 거리가 주어지면, 이 거리들의 분포를 대변하는 2차원이나 3차원과 같은 저차원 좌표들의 집합을 구하는 방법이다. 이때 이 좌표들의 차원은 필요에 따라 미리 정하는데, 본 연구에서는 구한 좌표들을 시각적으로 표현해야 되기 때문에, 2차원 좌표를 사용했다. Torgerson[11]에 의해서 개발된 MDS는 최적화 함수로 귀결되며, 다음과 같이 3 단계의 과정을 거쳐서 구한다. 첫째, Isomap 알고리즘의 첫째 및 둘째 단계에서 구해진 표정공간의 집합인 행렬 D 로부터 행렬 $A = (a_{ij}) = -d_{ij}^2/2$ 을 계산하고, 이를 이용하여 이중 중심화 행렬 $B = (b_{ij}) = HAH$ 를 구한다. 여기서 $H = I_n - \frac{J}{n}$ 로서, I_n 는 항등 행렬, J 는 d_{ij} 의 크기가 n 일 경우 모두 1의 값을 가진 행렬 $n \times n$ 을 의미한다. 둘째, 행렬 B 의 고유치와 고유벡터를 구하고, 고유치를 오름차순 정렬한다. 셋째, 본 논문에서 구하고자 하는 차원 수 2만큼 행렬 B 의 최대 고유치(λ) 2개를 추출하고, 이에 대응하는 고유벡터(v)를 사용하여 2차원 평면상의 $n \times 2$ 크기의 좌표들로 구성된 집합 $\{y_i\}$ 를 식 (2)와 같이 구한다.

$$\begin{aligned} \{y_i\} &= \{y_1, \dots, y_n\} \\ &= \{(v_{11} \sqrt{\lambda_{11}}, v_{12} \sqrt{\lambda_{12}}), \dots, (v_{n1} \sqrt{\lambda_{n1}}, v_{n2} \sqrt{\lambda_{n2}})\} \end{aligned} \quad (2)$$

본 논문에서는 2차원 평면상의 점들의 집합 $\{y_i\}$ 을 구하기 위해 Matlab V7.0을 사용하였다. Matlab 함수 Isomap에 비유사성 대칭행렬 D 와 차원의 수 2를 입력으로 사용하여 2차원 평면상의 점들의 집합 $\{y_i\}$ 를 구하였다.

IV. 사용자 인터페이스 및 실험

본 논문에서는 2400여개의 표정으로 구성된 표정공간을 생성하고, 이를 Isomap을 통해 2차원 공간으로 투영하였으며, 사용자로 하여금 이 공간을 향해하면서 실시간으로 얼굴 표정 제어를 가능하게 하였다. 이를 위한 사용자 인터페이스는 [그림 3]과 같으며, 3D Studio MAX R5.1의 MAXScript로 구현하였다.

[그림 3]의 오른쪽은 2400여개 표정상태벡터들 상호간의 거리정보를 유지한 상태로 2차원 평면에 분포시킨 항해공간이고, 좌상의 3차원 얼굴 모델은 사용자가 오른쪽의 항해공간을 향해할 때마다 실시간적으로 표정상태벡터에 해당되는 얼굴표정을 보여주게 된다. 또한 좌하는 오른쪽 항해공간에 분포된 작은 점들이 어떤 얼굴표정을 의미하는지를 보여주는 것으로서 대표적인 얼굴 표정들을 이미지로 표현한 것이다. 즉, 사용자가 좌하의 이미지를 보면서 오른쪽 2차원 항해공간에 분포된 각 얼굴 표정상태를 대표하는 작은 점을 마우스로 선택하면서 항해를 하면, 선택된 점에 해당되는 얼굴 표정이 좌상의 3차원 얼굴 모델에 실시간으로 나타나게 된다.

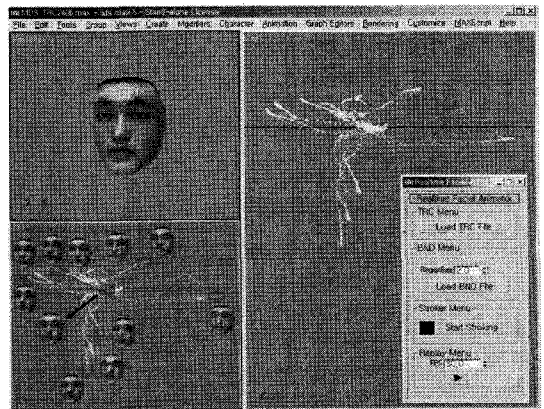


그림 3. 사용자 인터페이스(좌상: 3차원 아바타의 얼굴 모델, 좌하 :항해공간의 대표적인 얼굴 표정들을 애니메이션을 위해 이미지로 표현, 우: 항해공간과 메뉴), 항해공간에 분포된 각 얼굴 표정상태를 대표하는 작은 폴리곤을 마우스로 선택하면, 선택된 점에 해당되는 3차원 얼굴 모델이 실시간적으로 보여 짐으로써 실시간 표정제어가 가능해짐

사용자 인터페이스에서 애니메이터에 의해 선택된 점에 해당되는 얼굴 표정이 좌상의 3차원 얼굴 모델에 실시간으로 나타나게 하기 위한 원리는 다음과 같다. 즉, 3차원 얼굴 모델은 본 논문을 위해서 얼굴 모션 캡처를 할 때 배우의 얼굴에 부착된 마커 100개의 위치와 같은 지점에 해당되는 정점마다 100개 마커의 좌표 값을 각각 연결시켰다. 그리고 마커에 해당되는 정점이 모션 데이터에 의해서 움직일 때마다 주변 정점들을 포함하는 메시들은 정점으로부터 거리가 가까울수록 정점의 이동만큼 많이, 정점으로부터 거리가 멀수록 작게 움직이게 되며, 이를 위해서는 3D MAX의 플러그인 Bones Pro V3.0을 사용하였다. 그러므로 애니메이터가 항해공간에서 얼굴표정 데이터를 대신하는 작은 점들을 연속적으로 선택하게 되면 3차원 얼굴 모델은 실시간으로 모션 데이터를 표현해주게 된다.

[그림 4]는 Isomap에 의해서 생성된 2400여개의 2차원 표정공간을 사용자가 항해하고 있는 항해경로를 표시한 것이다. 사용자는 [그림 4]의 우상과 같이 마우스를 사용하여 항해 공간을 항해하고 동시에 3차원 얼굴 모델에 적용된 얼굴 표정을 보게 되는데, 이때에는 표정 하나하나를 항해 과정과 항해 속도에 따라 확인하면서 보게 된다.

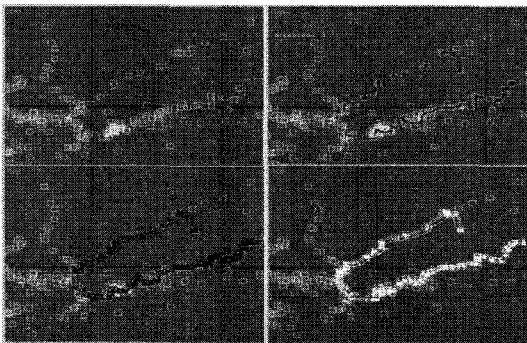


그림 4. 좌상 : 표정 공간, 우상 : 항해 과정, 좌하 : 항해 완료 및 항해 경로(궤도), 우하 : 재생 및 반복 재생

[그림 4]의 좌하와 같이 사용자의 항해 과정이 끝나면 우하와 같이 사용자의 항해 궤도에 해당되는 얼굴 표정들을 연속적으로 보고 얼굴 표정의 변화를 확인할 필요

가 있다. 이때에는 사용자의 항해 궤도를 처음부터 끝까지 자동으로 반복 항해하여 연속된 얼굴 표정의 변화를 3차원 얼굴 모델이 보여주게 된다. [그림 4]좌하는 사용자가 항해한 항해경로를 연결하여 보여주고 있으며, 우하의 적십자는 이때의 항해 경로를 따라가는 현재의 경로 위치를 가리켜주기 위한 지시자이다. 본 사용자 인터페이스는 사용자가 [그림 4]와 같은 항해경로를 다수 개 생성하여 서로 다른 색상으로 표현할 수 있으며, 사용자가 항해한 경로들 중 임의의 경로 하나를 선택하여 재생하고 확인하여 수정할 수 있다.

[그림 5]는 [그림 3]의 좌하에 있는 이미지로서 사용자가 모든 2차원 표정공간을 항해하면서 마우스로 선택한 경로들 상에 있는 대표적인 얼굴표정들을 표시한 것이다. 원래 얼굴모델은 [그림 3]과 같이 사용자 인터페이스의 좌상에 실시간으로 표시되지만, 애니메이터가 사용자 인터페이스를 사용하기 쉽게 하기 위해서 그리고 설명의 편의를 위해 항해공간에 표시하여 이미지로 제작하고 사용자 인터페이스에 삽입하였다.

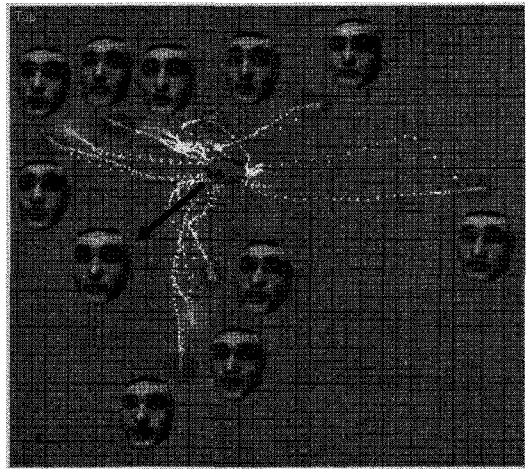


그림 5. 애니메이터가 항해공간에 분포된 각 얼굴 표정상태를 대표하는 작은 폴리곤들을 마우스로 선택한 경로들 상에 있는 대표적인 얼굴표정들. (원래 얼굴모델은 별도의 창에 표시되지만, 설명의 편의를 위해 항해공간에 표시함)

또한 [그림 5]는 거리행렬 방식에 의해 표현된 표정상태벡터를 사용한 결과로서 분포도의 중심부인 무표정 상

태를 중심으로 각 표정마다 방사형으로 분포가 되고 있음을 알 수 있다. 유사한 표정은 인접하게 분포가 되고 있고, 표정이 서로 많이 차이가 날수록 멀리 떨어져 분포가 되었음을 확인할 수 있다. 또한 각 표정마다 모션 데이터의 프레임단위로 연속적으로 적당한 거리를 두면서 분포가 되고 있음을 확인할 수 있다. 결국 2차원 항해공간에 분포시킨 분포도는 애니메이션이 실시간적으로 3차원 아바타의 표정을 제어할 수 있도록 하기 위해 매우 효율적으로 분포되어져 있다고 볼 수 있었다.

거리행렬 방식에 의해 표현된 표정상태벡터를 사용한 실험에서는 인접표정을 결정하는 인접거리 임계값을 실험에 의해 최적의 값을 결정해야한다. 본 논문에서는 이를 위하여 다음과 같은 기준을 적용하였다. 인접거리 임계값에 따라 4950차원 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 위해서 다차원 최단거리 분포와 2차원 최단거리 분포사이의 상관계수는 피어슨의 상관계수 r [10]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 식 (3)과 같이 계산되어진다.

$$r = \frac{\sum_{i=1}^n (v_{di} - \bar{V}_d)(v_{yi} - \bar{V}_y)}{\sqrt{\sum_{i=1}^n (v_{di} - \bar{V}_d)^2 \cdot \sum_{i=1}^n (v_{yi} - \bar{V}_y)^2}} \quad (3)$$

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 은 $r \geq 0.90$ 일 경우 매우 높은 상관관계를 가지며, $r=1.0$

일 때에는 일치한다. 본 논문에서는 가장 높은 상관관계를 가지는 인접거리 임계값을 찾기 위해 수많은 실험을 하였으며, 실험 결과 가장 높은 상관관계는 $r=0.9647$ 이고, 이때의 인접거리 임계값은 230mm이다. [그림 3]의 오른쪽 항해공간에 분포된 점들과 그림 5는 이 인접거리를 이용하여 구성한 2차원 최단 거리 분포이다.

본 논문에서는 [그림 5]와 같이 인접거리 임계값을 230mm로 주었을 때 생성되는 2차원 거리 분포를 사용하여 실시간 표정 제어를 하였다. 본 논문에서 자세히 언급하지는 않았지만 위치벡터방식으로 표정상태를 표현하는 경우, 위에서 언급한 결과들이 나오지 않았다. 즉, 두 공간간의 상관도를 가장 높게 만드는 인접거리 임계값을 사용했을 때, 임의의 두 표정 간의 경로가 존재하지 않는 경우가 있었다.

V. 결론

본 논문에서는 Isomap 알고리즘을 기반으로 다량의 얼굴 모션 데이터들을 2차원 공간에 분포시키고, 사용자가 이 공간을 항해하면서 원하는 얼굴 표정들을 실시간으로 선택하고 디스플레이 하는 방법을 기술하였다. 2400여개의 얼굴 표정 프레임을 사용하여 항해 공간을 구성하였으며, 이 공간을 구성하기 위해서는 표정상태벡터들의 집합을 Isomap의 3 단계에 적용하여 구하였다. 실시간 얼굴 표정 제어를 위해서 개발한 사용자 인터페이스는 표정상태 표현법을 위한 위치벡터 및 표정상태벡터 방식을 실험적으로 확인하는데 유용하게 사용되었다. 사용자 인터페이스는 얼굴 표정 제어를 위해 2차원 평면을 자유자재로 항해하면서 실시간으로 생성, 확인, 수정 및 재생성이 가능하기 때문에 유용하고 효율적이라는 것을 사용자로 하여금 확인할 수 있었다. 본 논문의 연구 결과는 기존의 관련 연구와 비교할 때, 모션 데이터를 사용한 3차원 아바타 얼굴 표정의 실시간 제어가 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 논문

의 연구 결과는 매우 적합하다. 또한 3차원 캐릭터 애니메이션을 필요로 하는 각종 방송, 광고, 애니메이션, 엔터테인먼트 등의 분야에서 3차원 아바타의 다양한 얼굴 표정 변화가 실시간이던 비실시간이던 요구되어질 때 본 연구의 결과물은 매우 중요한 역할을 할 수 있을 것으로 본다. 그리고 본 논문의 연구결과는 아바타의 표정제어와 관련된 분야뿐만 아니라 통계 및 마케팅 분야 등에서 자사 제품의 위치, 경쟁 제품의 위치 및 경쟁 강도를 파악하고, 이상점(Ideal point)을 발견하여 각종 문제점들을 해결하는데 유용하게 활용될 수도 있다.

참고문헌

[1] D. Terzopoulos, B. M. Hattal, B. Hofer, F. Parke, D. Sweetland, and K. Waters, Facial animation : Past, present and future, Panel, SIGGRAPH97, 1997.

[2] F. I. Parke and K. Waters, Computer facial animation, A K Peters, 1996.

[3] D. Fidaleo and U. Neumann, "Analysis of co-articulation regions for performance-driven facial animation," Journal of Visualization and Computer Animation, Vol.15, pp.15-26, 2004.

[4] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin, "Making Faces," ACM SIGGRAPH98 Conf., pp.55-66, 1998.

[5] Z. Deng, P. Y. Chiang, P. Fox, and U. Neumann, "Animating blendshape faces by cross-mapping motion capture data," Proceedings of the 2006 symposium on Interactive 3D graphics and games 2006, pp.43-48, 2006.

[6] C. Kouadio, P. Poulin, and P. Lachapelle, "Real-time facial animation based upon a bank of 3D facial expressions," Proc. Computer Animation 98, 1998.

[7] D. Vlasic, M. Brand, H. Pfister, and J. Popovic, "Face Transfer with Multilinear Models," ACM

Transactions on Graphics (TOG), Vol.24, pp.426-433, 2005.

[8] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive Control of Avatars Animated with Human Motion Data," ACM Transactions on Graphics (SIGGRAPH 2002), Vol.21, No.3, pp.491-500, 2002.

[9] R. W. Floyd, "Algorithm 97 : Shortest Path," CACM, Vol.5, p.345, 1962.

[10] Uprendra Shardanand, "Social information filtering for music recommendation," Master's thesis, MIT, 1994.

[11] W. S. Torgerson, "Multidimensional scaling: I. theory and method," Psychometrica., Vol.17, pp.401-419, 1952.

[12] T. Cox and M. Cox, Multidimensional Scaling. Chapman & Hall, London, 1994.

[13] H. J. Shin and J. H. Lee, "Motion Synthesis and Editing in Low-Dimensional Spaces," Computer Animation and Virtual Worlds (Special Issue: CASA 2006), John Wiley & Sons, Vol.17, pp.219-227, 2006.

저자 소개

김 성 호(Sung-Ho Kim)

정회원



- 1998년 8월 : 숭실대학교 컴퓨터학과 (공학석사)
- 2005년 2월 : 숭실대학교 컴퓨터학과 (공학박사)
- 2006년 3월 ~ 현재 : 상지대학교 컴퓨터정보공학부 교수

<관심분야> : 컴퓨터 그래픽스, 컴퓨터 애니메이션, 모션 캡처 애니메이션, 가상현실, Web3D, 멀티미디어 등