

얼굴 표정공간의 계층적 가시화

(Hierarchical Visualization of the Space of Facial Expressions)

김 성 호 [†] 정 문 렬 ^{**}
(Sung-Ho Kim) (Moon-Ryul Jung)

요 약 본 논문은 애니메이터로 하여금 계층적 가시화 기법에 의한 표정들의 공간으로부터 일련의 표정을 실시간적으로 선택하게 함으로써 얼굴 표정 애니메이션을 생성하는 기법을 기술한다. 본 시스템에서는 약 2400여개의 표정 프레임에 이용하여 2차원 표정공간을 구성하였으며, 이를 위해서는 한 표정을 표시하는 상태표현으로 얼굴 특징 점들 간의 상호거리를 표시하는 *거리 행렬*을 사용한다. 또한 두 표정 간의 최단거리를 구하기 위해 다이나믹 프로그래밍 기법을 사용한다. 그리고 다차원 공간인 표정공간을 항해하기 위해서 다차원 스케일링 기법을 이용하여 2차원 공간으로 가시화한다. 그러나 표정의 수가 너무 많아 애니메이터가 항해를 하는데 어려움이 많기 때문에 계층적 가시화 기법이 필요하다. 표정공간을 계층적으로 분할하기 위해, 퍼지 클러스터링을 이용한다. 초기 단계에서는 약 10개의 클러스터 센터(후보 키 프레임)를, 단계가 증가될 때마다 후보 키 프레임의 수를 두 배로 하여 클러스터링하고 2차원 평면에 표시한다. 애니메이터는 초기 단계의 후보 키 프레임들 중에서 특정 키 프레임들을 선택하여 항해경로를 생성한다. 특정 단계에서는 그전 단계의 항해경로를 따라 새로운 키 프레임들을 선택하여 항해경로를 갱신하고, 마지막 단계에서 항해경로를 갱신하면 애니메이션 설정이 끝난다. 본 논문은 본 시스템이 어떤 효과가 있는지를 알기 위해 애니메이터들로 하여금 본 시스템을 사용하여 얼굴 애니메이션을 생성하게 하였으며, 그 결과를 평가한다.

키워드 : 얼굴 모션 캡처, 얼굴 애니메이션, 항해, 퍼지 클러스터, 계층적 가시화

Abstract This paper presents a facial animation method that enables the user to select a sequence of facial frames from the facial expression space, whose level of details the user can select hierarchically. Our system creates the facial expression space from about 2400 captured facial frames. To represent the state of each expression, we use the *distance matrix* that represents the distance between pairs of feature points on the face. The shortest trajectories are found by dynamic programming. The space of facial expressions is multidimensional. To navigate this space, we visualize the space of expressions in 2D space by using the multidimensional scaling(MDS). But because there are too many facial expressions to select from, the user faces difficulty in navigating the space. So, we visualize the space hierarchically. To partition the space into a hierarchy of subspaces, we use fuzzy clustering. In the beginning, the system creates about 10 clusters from the space of 2400 facial expressions. Every time the level increases, the system doubles the number of clusters. The cluster centers are displayed on 2D screen and are used as candidate key frames for key frame animation. The user selects new key frames along the navigation path of the previous level. At the maximum level, the user completes key frame specification. We let animators use the system to create example animations, and evaluate the system based on the results.

Key words : facial motion capture, facial animation, navigation, fuzzy clustering, hierarchical visualization

· 본 논문은 과학재단 특장기초 연구과제(R01-2002-000-00311-0)의 수행결과에 일부임

† 정 회 원 : 송의여대 정보통신계열 교수
kimsh1204@hotmail.com

** 종신회원 : 서강대학교 영상대학원 미디어공학과 교수
moon@mail.sogang.ac.kr

논문접수 : 2004년 2월 5일

심사완료 : 2004년 8월 9일

1. 서 론

인터넷과 멀티미디어의 기능이 고급화됨으로써 가상 공간 속에 사용자 자신을 대신하는 3차원 아바타의 필요성이 대두되었고, 지금은 많은 멀티미디어 분야에서 널리 사용되고 있다. 이런 가상공간에서 사용자들이 3차원 아바타를 표현할 때 가장 관심을 가지는 것은 역시

3차원 아바타의 얼굴 표정이다. 인간은 자신의 감정을 말보다는 얼굴 표정을 통해서 더 잘 표현하며, 상대방의 얼굴 표정을 보고 가장 빠르게 상대방의 현재 감정 상태를 파악한다. 그런 연유로 지금까지 3차원 컴퓨터 그래픽스 기법을 통해서 인간의 얼굴 표정을 표현할 수 있는 많은 방법이 연구[1-8]되었다. 가상공간에서 3차원 아바타의 얼굴 표정을 잘 표현하기 위한 방법으로는 모핑 기법, 골격구조를 이용한 물리적 기법, 근육기반 기법, 모션 캡처 기법 등이 연구되어져 왔거나 연구되어지고 있다. 3차원 모핑 기법을 사용한 표정 애니메이션 생성기법[5]은 일반 사용자가 서로 다른 두 표정을 보간시켜 수개 혹은 수십 개의 중간 표정들을 생성하고 이를 순서대로 연결하여 애니메이션을 생성하는 방식이다. 그러나 이 방법은 사용자가 모든 중간 표정들을 모핑 기법으로 일일이 생성시켜야 하는 복잡한 절차와 함께, 사용자가 표현하고자 하는 표정 애니메이션을 위해 중간 프레임들을 제대로 생성하지 못하는 단점이 있다. 다음으로 골격구조를 이용한 물리적 기법[8]은 3차원 스캔한 얼굴 모델을 기반으로 피부, 근육, 뼈 등의 계층 구조를 가진 물리적인 얼굴 모델을 개발하고, 이에 적응(Adaptive) 시뮬레이션 알고리즘(ASA)을 개발, 사용하여 해부학적인 표정 애니메이션을 생성한다. 그런데 이 기법은 너무 해부학적인 구조에 집중되어있어 애니메이터가 다양한 표정들을 연속적으로 생성하기가 어렵고 이들을 연결한 표정 애니메이션을 제대로 나타내지 못하는 단점을 가지고 있다. 근육기반 기법[7]은 근육 기반의 얼굴 모델을 사용하여 키 프레임 방식의 대화식 표정 애니메이션을 생성하기 위한 방식이다. 애니메이터는 근육 벡터들을 이동시킴으로서 특정 표정을 생성하고, 코사인 보간법을 사용한 표정들의 조합으로 다양한 표정을 생성한다. 그러나 이 기법 또한 애니메이터가 근육 벡터를 이동시켜서 생성하는 표정이 매우 부자연스러울 뿐만 아니라 복잡하기 때문에 이들의 조합으로 생성된 표정 애니메이션 또한 좋은 표정 애니메이션이라 할 수가 없다. 마지막으로 모션 캡처 기법이 있는데, 좀 더 실감나는 표정 애니메이션을 쉽게 생성하기 위해 최근에 많이 사용되는 방식이다. 모션 캡처한 데이터를 사용하는 방법으로 크게 두 가지가 제시되었는데, 첫째는 배우의 모션 데이터를 새로운 모델에 재적용하는 모션 리타겟팅 기법[6,9,10]이다. 모션 리타겟팅 기법은 먼저 몸동작에 대한 것이 제시[9,10]되었고, 얼굴 표정에 대한 것도 제시[6]되었다. [6]은 3차원 아바타를 사용하여 다수 개의 기본 표정들을 모델링하고 각 표정의 제어점들을 마커와 매핑시킨다. 또한 각 표정의 최대 모션을 모션 캡처하여 3차원 아바타에 적용한 표정 상태를 데이터베이스에 저장해둔다. 그리고 실시간 모션 캡처로부터

입력되어지는 표정 데이터와 가장 가까운 표정 상태를 데이터베이스에서 찾아 3차원 아바타를 재구성하여 보여주는 기법이다. 이는 복잡한 작업 과정과 고가의 실시간 모션 캡처 장비가 기본적으로 필요하기 때문에 일반적으로 사용되기에는 적합하지 않다. 둘째는 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[11]되었는데, 아직 얼굴 표정에 대한 것은 제시된 것이 없다. 또한 [11]에서 몸자세에 대해서는 자세공간을 계층적으로 표시하여 시스템 내에서 탐색을 쉽게 했으나, 탐색공간을 계층적으로 가시화하지는 않았다. 여기서 자세공간을 계층적으로 표시한다는 것은 현재 위치에서 인접 위치로 이동하기 위한 다양한 자세들을 미리 보여주고 사용자가 선택하게 하여 이동한 다음, 그 지점에서 다시 인접 위치로 이동하기 위한 다양한 자세들을 보여주고 선택하게 하여 이동하는 과정을 연속적으로 처리해준다는 것이다. 그리고 탐색공간을 계층적으로 가시화한다는 것은 현재 위치에서 인접 위치나 거리가 먼 위치로 이동할 때, 사용자가 낮은 계층에서 현재 지점과 목표 지점의 자세 및 중간 지점들의 대략적인 자세들을 선택하게 하고, 계층의 단계를 높이면서 대략적인 중간 지점들 사이의 세부적인 자세들을 선택하게 하기 위한 것으로서 매우 정밀하고 구체적인 자세 동작을 생성하기 위함이다. 그러므로 본 논문에서는 표정 애니메이션을 생성할 수 있도록 하기 위해 표정공간을 계층적으로 가시화하여 애니메이터가 쉽게 공간을 탐색하도록 하기 위한 방식을 제안한다.

표정 데이터를 사용하여 계층적 가시화 기법에 의한 표정 애니메이션을 생성하기 위해서는 다음과 같은 사전 작업이 필요하다. 먼저 얼굴 모션 캡처 작업을 수행할 배우의 정면 사진을 이용, 애니메이션 가능한 3차원 얼굴 모델링 작업으로 아바타를 생성한다.

그리고 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 서로 다른 10개의 얼굴 모션을 연출하게 하고 초당 60 프레임으로 캡처한다. 모든 얼굴 모션은 처음 무표정 상태에서 출발하여 특정 표정까지 진행한 다음, 다시 무표정 상태로 되돌아오는 방법으로 진행된다. 한 개의 마커는 3개의 좌표 값으로 표현되므로 100개의 마커 위치로 표현되는 한, 표정은 300차원의 데이터이다.

또한 100개 마커를 3차원 아바타의 얼굴 메쉬 각 부위(모션 캡처시 마커 부착 위치)에 연결하여 표정 재생

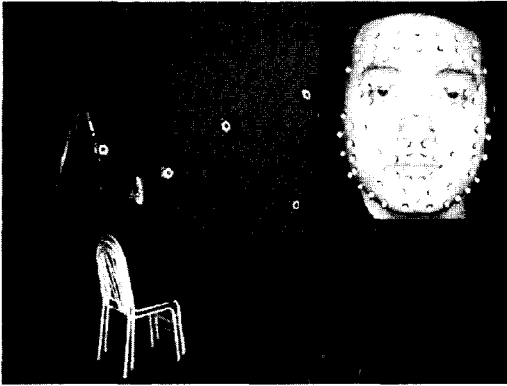


그림 1 마커 100개 부착위치 및 광학식 모션 캡처 시스템을 사용한 얼굴 모션 캡처 장면

이 가능하게 한다. 그리고 약 2400여개의 얼굴 표정 데이터에서 임의의 두 표정간의 최단경로를 결정하여 표정공간을 생성[12]하고, 이 공간을 다차원 스케일링(Multidimensional Scaling, 이하 'MDS')[13] 기법으로 2차원 평면에 투영시킨다. 본 논문에서는 이 과정들을 미리 구축하고, 2차원 평면에 투영된 약 2400여개의 MDS 결과 데이터를 계층적 가시화 실험에 그대로 사용한다.

계층적 가시화는 먼저 약 2400여개의 표정들을 사용하여 수개의 클러스터 센터들을 가지도록 클러스터링하여 대표적인 얼굴 표정만을 2차원 평면에 표시한다. 이는 애니메이션가 수개의 클러스터 센터들을 시각적으로 확인함으로써 표정 공간에 분포된 대표적인 표정과 위치를 쉽고 빠르게 파악할 수 있도록 하기 위해서이다. 그리고 매 단계마다 클러스터의 수를 두 배로 증가시키고 애니메이션으로 하여금 특정한 한 단계를 선택하도록 한다. 계층적 가시화의 최대 단계는 전체 표정의 수에 따라 자동적으로 결정된다. 본 논문에서는 계층적 가시화에 사용될 클러스터 센터들을 구하기 위해 퍼지 클러스터(Fuzzy C-Mean 클러스터링, 이하 'FCM') 기법[14-16]을 사용한다. 2차원 평면에 분포된 클러스터 센터들은 애니메이션을 위한 후보 키 프레임들로 설정하고, 애니메이션으로 하여금 특정 키 프레임들 선택하게 하여 선택한 순서대로 항해경로를 설정한다.

2. 표정공간의 생성 및 MDS

2.1 표정상태 표현법

표정공간을 생성하기 위해서는 각각의 얼굴 표정상태를 수치적으로 표현해야 한다. 표정상태는 얼굴에 부착된 마커 위치들에 의해 결정된다. 표정상태의 표현은 표정들 간의 상대적인 거리 관계를 잘 표현하는 것이어야

한다. 표정의 상태를 표현하는 가장 간단한 방법은 마커 위치들로 이루어진 상태벡터를 이용하는 것이다. 본 논문에서는 100개의 마커를 사용하고 각 마커는 3개의 좌표를 가지기 때문에, 표정상태벡터는 300차원이 된다. 이런 식으로 표정상태를 표현하는 방법을 '위치벡터방식'이라고 하자. 본 논문에서는 표정상태를 표현할 때, 위치벡터방식을 사용하지 않는다. 대신 임의의 두 마커간의 상호거리를 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.

2.2 표정공간의 생성

표정상태를 거리행렬로 표현하면 표정간의 직선거리는 두 거리행렬간의 직선거리로 표현된다. 거리행렬로 표현된 표정상태들의 공간은 임의의 두 거리행렬간의 거리를 결정함으로써 결정된다. 본 논문에서는 거리행렬을 하나의 벡터로 보고(행렬의 각 행을 줄 나열하여 그 결과를 하나의 벡터로 간주함), 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 구면과 같은 다양체(Manifold) 공간인 것이다. 다양체 공간상에서의 거리는 두 점간의 거리를 한 점에서 이 공간을 벗어나지 않으면서 다른 점까지 도달하는 최단경로의 길이로 정의[12]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 "인접표정"이라고 하는데, 임의의 표정에 대한 인접표정들은 그림 2에서 보는 것처럼 결정한다. 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 최적의 한계 값을 정해야한다. 다만, 인접거리 임계 값은 임의의 표정에서 다른 임의의 표정으로 이동하는 데 필요한 충분한 수의 인접표정들이 나오도록 설정되어야 한다.

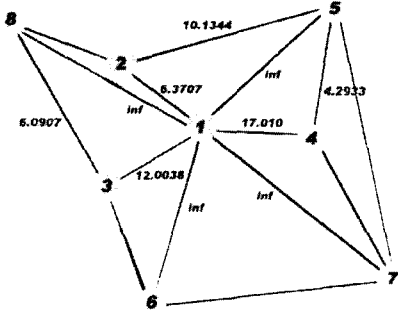


그림 2 플로이드(Floyd) 알고리즘을 위한 그래프 생성 방법 : 모두 8개의 상태가 주어졌다고 보고, 1번 상태를 기준으로 2, 3, 4 번 상태는 1번 상태와 인접해 있고, 5, 6 7, 8 번 상태는 인접해 있지 않다. 인접표정간의 거리는 특정 값으로 주어져 있다. 인접표정이 아닌 두 상태간의 거리는 무한대(inf)로 주어져 있는데, 이것은 한 상태에서 다른 상태까지 바로는 이동할 수 없다는 것을 의미한다. 원의 반지름은 인접거리 한계 값을 나타낸다.

본 논문에서는 최적의 인접거리 임계 값을 구하기 위하여 다음과 같은 실험 과정을 거쳤다. 먼저 실험에 사용한 얼굴표정의 다양체 공간은 300차원의 다차원 공간이다. 그리고 이 데이터에 MDS를 사용하여 2차원 평면에 투영시킨 결과는 2차원 공간이다. 이때 다차원 공간 상에서 표정들 사이의 최소거리와 최대거리를 고려한 인접거리 임계 값을 임의로 설정하여 구해진 표정들 사이의 최단거리 분포와 2차원 공간상에서 표정들 사이의 최단거리 분포가 얼마나 유사한지를 측정하여 가장 높은 유사도를 가진 인접거리 임계 값을 찾는다. 다차원 공간상에서 최단거리 분포와 2차원에서 최단거리 분포 사이의 유사도 측정을 위한 상관계수는 피어슨(Pearson)의 상관계수, r [17]을 사용하였다. 본 논문에서는 이와 같은 과정의 수많은 실험을 통하여 $r=0.9647$ ($r=1.0$ 이면 일치)일 때 최적의 인접거리 임계 값으로 230mm를 찾았다. 인접표정들이 결정되면, 인접하지 않은 두 표정상태간의 거리는 그 사이에 있는 인접 표정들 간의 거리들을 합하여 구한다. 이를 위해, 최단거리를 구하는 알고리즘인 플로이드(Floyd) 알고리즘(다이나믹 프로그래밍 기법)[12]을 이용한다. 이렇게 임의의 두 표정간의 최단거리가 구해지면 해당 다양체 공간이 결정된다. 본 논문에서는 모두 약 2400여개의 얼굴표정을 사용하여 다양체 공간을 형성하였다.

2.3 다차원 스케일링 (MDS)

앞에서 생성한 얼굴표정의 다양체 공간은 300차원 공

간이다. 따라서 이 공간을 에니메이터가 향해하면서 원하는 표정을 선택하는 것은 거의 불가능하다. 그러므로 원래 표정공간의 구조를 근사적으로 표현하는 2차원 또는 3차원으로 공간을 구하여 이 공간을 향해하는 방법을 사용한다. 이를 위해서는 고차원 공간을 저차원 공간으로 차원 축소하여 투영하는 방법이 필요하며, 여기에는 PCA(Principal Component Analysis), MDS, LLE(Locally Linear Embedding) 등이 있다. PCA는 투영된 분산을 최대화하는 인접 구조 보존 방식이고, LLE[18]는 모든 데이터 점으로부터 가장 가까운 인접 데이터의 수를 임의로 정하고 이를 보존하는 인접 데이터 보존 방식이다. 그리고 MDS는 데이터 점들 사이의 Pairwise distance 보존을 사용하는 거리 보존 방식이다. 그러므로 본 논문에서는 표정들 사이의 최단거리를 보존하면서 저차원 공간에 투영하기 위해 MDS 기법 [13]을 사용한다. MDS는 고차원 데이터들 사이의 거리가 주어지면, 이 거리들의 분포를 대변하는 좌표들의 집합을 구하는 방법이다. 이때 이 좌표들의 차원은 필요에 따라 미리 정하는데, 본 연구에서는 구한 좌표들을 시각적으로 표현해야 되기 때문에, 2차원 좌표를 사용했다. 그리고 거리 보존 방식인 MDS는 직선거리(Euclidean 거리)를 기본적으로 사용하는데, 본 논문에서는 표정들 사이의 거리가 인접 표정을 거쳐서 계산되기 때문에 다양체 거리를 사용한다.

표정상태를 나타내는 n 개의 다차원 다양체 공간상의 점들을 x_1, \dots, x_n , 임의의 표정상태 x_i 와 x_j 사이의 다양체 거리를 d_{ij} , $i, j=1, \dots, n$ 이라고 하자. d_{ij} 로 구성된 거리행렬을 $\{d_{ij}\}$ 로 표시한다. 행렬 $D=d_{ij}$ 는 대각 원소들이 0인 대칭행렬이다. 이를 MDS 이론에서 보통 비유사성 대칭행렬이라고 한다. 다차원 표정 공간을 근사적으로 나타내는 p 차원 공간상의 점들을 y_1, \dots, y_p 라 하자. 각 y_i 를 상태(Configuration) 벡터라고 한다. 상태 벡터들은 시각화하여 보여주는 것을 목적으로 하는 경우가 많으므로 p 는 보통 2 또는 3이다. p 차원 점 y_i 와 y_j 간의 거리를 δ_{ij} 이라고 하자. 일반적으로 MDS를 적용하면, p 차원 거리들의 집합 $\{\delta_{ij}\}$ 가 다차원 다양체 거리들의 집합 D 와 가장 근사한 분포를 가지도록 하는 p 차원 평면상의 점들의 집합 $Y=y_i$ 를 구할 수 있다.

본 논문에서는 2차원 평면상의 점들의 집합 $\{y_i\}$ 을 구하기 위해 Matlab V6.5를 사용하였다. Matlab의 cmdscale 함수에 비유사성 대칭행렬 D 와 차원의 수 2를 입력으로 사용하여 MDS를 구하고, 이를 집합 $\{y_i\}$ 로 사용하였다.

3. FCM(Fuzzy C-Mean) 클러스터링

MDS에 의해서 구해진 표정공간은 약 2400여개의 표정들이 2차원 공간에 분포되어져 있다. 애니메이션은 이 공간을 향해하여 다양한 표정 애니메이션을 생성한다. 그러나 2차원 표정 공간에 분포된 표정들의 수가 너무 많아 애니메이션이 향해하기에는 너무 복잡하고 난해하다. 그러므로 본 논문에서는 약 2400여개의 얼굴 표정들을 클러스터링 기법을 사용하여 수단계로 구분시킨 계층적 가시화 기법을 적용한다.

클러스터링은 n 개의 데이터가 주어졌을 때, 데이터들의 유사성을 기반으로 하여 그들을 그룹(Cluster)으로 지정하는 것이다. 본 논문에서 클러스터링을 하려는 이유는 주어진 n 개 데이터들의 특성을 그대로 보존하면서 유사한 표정들을 그룹으로 설정하고, 그룹(클러스터)의 센터를 그룹의 대표적인 표정으로 결정하기 위해서이다. 클러스터링 알고리즘에는 계층적(Hierarchical) 클러스터링[19,20]과 K-Means[21], FCM[14-16] 등이 있으며, 본 논문에서는 논문의 목적과 가장 잘 부합되는 FCM을 사용한다.

계층적 Clustering은 다차원 공간상의 유사한 데이터 점들의 클러스터를 결정하기 위해서 일반적으로 사용되는 방법 중의 하나이지만, 상부 트리로 계층이 상승될수록 연결(Link)에 의해 가지(Branch)의 수는 점차 줄어들게 되며, 최종 잎(leaves)으로 표현된 점들의 정보를 상실하게 되어 본 논문의 목적을 위해서는 적합하지 않다. 그리고 K-Means 알고리즘은 계산양이 너무 많아서 큰 데이터 집합을 위해서는 유용하지 않고, 전역 최소화를 찾기도 부적합하다. 그러므로 K-Means는 본 논문에서 사용하고자 하는 모션 데이터를 Clustering 하기에는 적합하지 않다.

FCM은 본 논문에서 사용한 것으로서 클러스터 지정에 있어서 불확실성을 가지고 있으며, 각 데이터에 주어진 각 클러스터에 속할 확률을 부여하여 데이터를 클러스터링하는 방법이다. 그리고 하나의 데이터가 두 개 이상의 클러스터에 속하게 하는 것을 허용하는 클러스터링 방법 중의 하나이다. FCM은 Dunn[16]에 의해서 개발되었으며, Bezdek[14]에 의해서 개선되어져 패턴 인식 분야에 자주 사용되어지고 있다.

클러스터하기 위한 얼굴 표정들의 집합을 $X = x_i, i=1, \dots, n$ 라고 하자. X 의 퍼지 부분집합인 클러스터는 'Membership 함수'라고 부르는 매핑 함수 $X \rightarrow [0,1]$ 로 표현된다. 여기서 $U_j(x_i)$ 는 얼굴 표정 x_i 의 클러스터 j 에 대한 Membership 계수를 나타낸다. 비록 하나의 얼굴 표정이라도 하나의 클러스터에 유일하게 속하지 않고, 다른 클러스터와의 임의의 위치 혹은

중양 부근에 놓여질 수도 있다. 즉, 모든 표정들은 지정된 수단계의 각 클러스터 센터로부터 측정된 서로 다른 Membership 계수를 가진다.

본 논문에서는 Membership 함수의 집합 와 클러스터 센터 c_j 의 집합으로 구성된 행렬 $C = \{c_j\}$ 를 구하기 위해 Matlab V6.5를 사용하였다. Matlab의 fcm 함수에 2차원 표정 공간과 각 계층별 임의로 설정한 클러스터 센터의 수 c 를 입력으로 사용하고, 이로부터 행렬 와 행렬 C 를 구하였다.

4. 사용자 인터페이스를 이용한 계층적 가시화와 실험

본 논문에서 사용하고자 하는 얼굴표정 공간은 2차원 공간이며, 애니메이션은 이 공간을 향해하여 다양한 표정 애니메이션을 생성한다. 그러나 2차원 표정 공간에 분포된 얼굴 표정들의 수가 너무 많아 애니메이션이 향해하기에는 너무 복잡하고 난해하다. 그러므로 본 논문에서는 약 2400여개의 얼굴 표정들을 계층적으로 가시화한다. 그리고 특정한 한 단계를 선택하여 향해하고 향해경로를 갱신하도록 한다.

4.1 사용자 인터페이스

계층적 가시화 초기단계에서는 수 개의 대표적인 얼굴 표정만을 사용자 인터페이스의 향해공간에 표시하고, 애니메이션은 이들 중 특정 표정들을 선택하여 초기 향해경로를 생성한다. 계층적 가시화를 위해서는 매 단계마다 클러스터의 수를 두 배로 한다. 그러면 계층적 가시화를 위한 최대 단계는 총 표정의 수에 따라 자동적으로 결정된다. 또한 매 단계마다 클러스터의 수단계 클러스터링한다. 그리고 애니메이션은 향해경로의 근처에 있는 키 프레임(클러스터 센터)들을 선택하여 향해경로를 갱신한다. 계층적 가시화를 위한 사용자 인터페이스는 그림 3과 같다.

4.2 표정들 사이의 연결 관계

계층적 가시화의 초기 단계는 약 2400여개의 표정을 FCM으로 클러스터 센터의 수를 약 10개 정도로 정하고 클러스터링한다. 그리고 클러스터 센터들을 후보 키 프레임들로 설정하여 그림 3과 같이 향해공간에 디스플레이 한다. 클러스터 센터의 표정은 향해공간에서 가장 가까이 있는 표정과 같다고 간주하고 그것을 애니메이션에게 보여준다.

애니메이션은 후보 키 프레임들 중에서 애니메이션에 사용할 키들을 선택하여 초기 향해경로를 생성한다. 이때 애니메이션은 표정의 계층적 분포도에서 표정들 사이의 이웃하는 연결 관계를 미리 알아야 한다. 그러므로 표정의 계층적 분포도는 단순히 표정만을 분포시키는

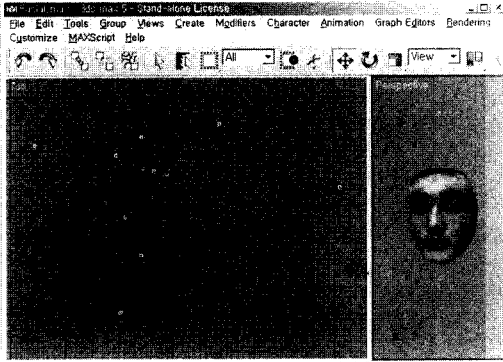


그림 3 사용자 인터페이스(좌: 항해공간과 우: 3차원 얼굴 모델), 애니메이터가 2차원 항해공간에 분포된 각 얼굴 표정상태를 대표하는 작은 점을 마우스로 선택하면서 항해를 하면, 선택된 점에 해당하는 3차원 얼굴 모델이 보여 진다.

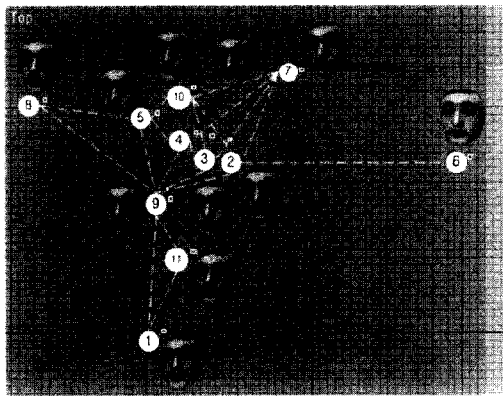


그림 4 계층적 가시화 초기 단계 : 약 10개 후보 키 프레임들의 분포와 표정들 간의 연결 관계. 표정은 설명의 편의를 위해 항해공간에 표시함

것이 아니라 그림 4와 같이 표정들 사이의 연결 관계까지 점선으로 표시를 해야 한다. 표정들 사이의 연결 관계는 약 2400여개의 모든 표정들을 분포시킨 상세 분포도에 포함되어있는 정보인데, 그림 4와 같은 초기 단계인 요약 분포도에도 표시를 해 주어야 한다. 표정의 상세 연결 관계로부터 클러스터 센터들의 연결 관계를 찾아내는 방법은 다음과 같다. 먼저 임의의 두 클러스터 센터간의 최단경로를 찾는다. 여기서 최단경로는 인접한 표정들을 거쳐서 가거나 거치지 않고 바로 가는 최단거리를 의미한다. 그러나 최단경로가 다른 클러스터 센터들이나 센터를 주변을 지나가면, 이것은 두 클러스터 센터간의 직접경로가 아니라고 본다. 즉, 표정들 사이의 연결 관계는 임의의 두 클러스터 센터 간에 직접 경로

만이 존재하여야 한다. 임의의 두 클러스터 센터간의 최단경로가 다른 클러스터 센터들이나 센터를 주변을 지나가는지는 다른 클러스터 센터를 중심으로 가상의 원을 그리고 그 원의 영역을 최단경로가 통과하는지를 확인하면 된다. 그리고 클러스터 센터를 중심으로 한 가상의 원의 반지름은 적당한 임계 값을 설정하여 실험으로 찾는다.

본 논문에서는 이와 같은 방법으로 표정들 간의 연결 관계를 구하고 그림 4와 같이 점선으로 표시하였다. 그림 4에서 연결 관계가 존재하지 않는 후보 키 프레임들 사이에서는 항해를 할 수가 없고, 항해를 하기 위해서는 점선으로 연결된 인접한 후보 키 프레임들을 거쳐서 항해해야만 한다. 예를 들어 키 프레임 1에서는 키 프레임 9, 11로 항해를 할 수 있지만, 키 프레임 1에서 키 프레임 8이나 5로 항해를 하고자 할 경우에는 인접한 키 프레임 11 혹은 키 프레임 9를 거쳐야 가능하다. 이는 표정이 급격하게 변하는 경우가 없고 인접한 표정을 거쳐서 항해를 해야 한다는 가정 하에서 구해진 표정들 간의 연결 관계이다. 또한 그림 4는 애니메이터가 2차원 표정공간에 있는 약 10개의 모든 후보 키 프레임들을 항해하면서 나타나는 얼굴 표정을 표시한 것으로서, 원래 얼굴모델은 그림 3에서와 같이 사용자 인터페이스의 오른쪽 창에 표시되지만, 설명의 편의를 위해 항해공간에 표시하였다.

4.3 계층적 가시화와 실험

그림 4를 이용하여 얼굴 표정 애니메이션을 생성하기에는 부적합한데, 이는 애니메이터가 선택한 키 프레임과 인접한 키 프레임들 보간 시켜주는 항해경로가 전혀 없기 때문이다. 그러므로 이들 사이에 적당한 개수의 새로운 후보 키 프레임들을 분포시켜서 세부적인 항해경로를 생성할 수 있도록 한다. 이를 위해서는 계층적 가시화가 필요하며, 매 단계마다 클러스터의 수를 두 배로 하면서 클러스터링한다. 그러면 모든 표정이 가시화되기까지의 최대 단계는 자동적으로 결정된다.

그리고 애니메이터로 하여금 적당한 단계를 선택하게 하고, 항해경로 근처에 있는 키 프레임들을 선택하여 항해경로를 갱신한다. 예를 들어, 애니메이터가 계층적 가시화 초기 단계인 약 10개의 후보 키 프레임들로 분포된 항해공간에서 키 프레임으로 7, 3, 9, 1을 그림 5와 같이 선택하여 초기 항해경로를 생성하였다고 하자. 계층적 가시화를 위해서 매 단계마다 클러스터의 수를 두 배로 하고 클러스터링하는데, 약 2400여개의 표정을 위한 최대 단계는 최소한 8단계($11 \times 2^8 = 2816$)가 필요하다. 본 논문에서는 이와 같이 수 개의 단계를 두어 애니메이터로 하여금 적당한 표정공간을 선택하여 항해하고 항해경로를 갱신할 수 있도록 하였다.

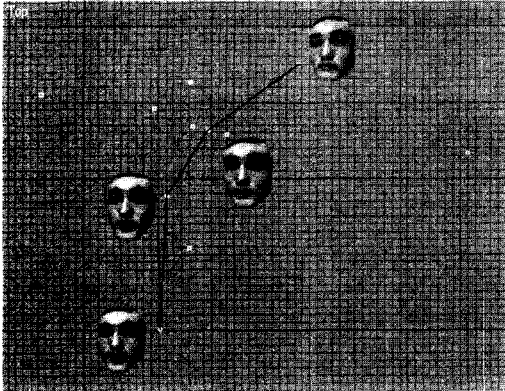


그림 5 계층적 가시화 초기 단계 - 약 10개 후보 키 프레임들 중에서 4개(7-3-9-1) 키 프레임을 선택한 초기 항해경로

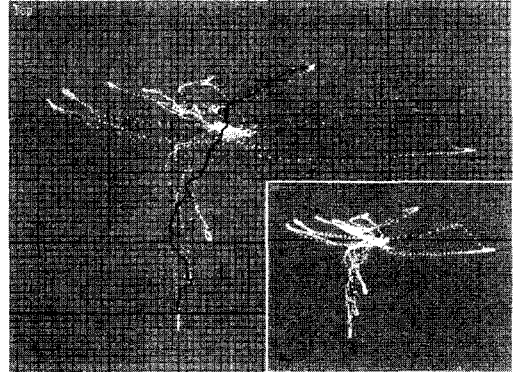


그림 7 계층적 가시화 마지막 단계 - 마지막 단계의 새로운 후보 키 프레임들을 표시(작은 창)하고, 기존의 항해경로와 가장 인접한 특정 키 프레임을 선택하여 갱신된 항해경로(큰 창)

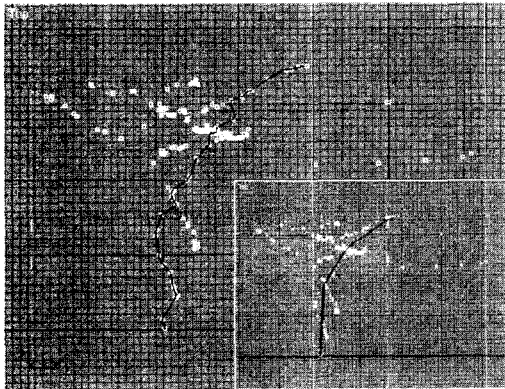


그림 6 계층적 가시화 3단계 - 초기 단계에서 선택한 키 프레임을 기반으로 새로 선택한 단계에서 새로운 후보 키 프레임들을 표시(작은 창)하고, 항해경로와 가장 인접한 특정 키 프레임을 선택하여 갱신한 항해경로(큰 창)

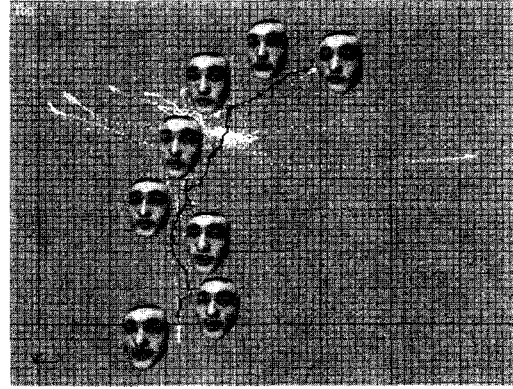


그림 8 계층적 가시화 마지막 단계에서 최종 갱신된 항해경로 상의 대표적인 얼굴 표정들을 설명의 편의를 위해 항해공간에 표시함

그림 6의 작은 창은 그림 5와 같이 초기 단계에서 생성한 항해경로를 그대로 보여주었고, 애니메이터가 새로 선택한 3단계의 후보 키 프레임들을 보여주고 있다. 또한 그림 6은 계층적 가시화 초기 단계에서 생성한 항해경로와 인접한 특정 키 프레임들을 선택하여 갱신한 항해경로를 보여준다. 애니메이터는 계층적 항해공간을 단계를 변형하면서 항해하는데, 각 단계마다 항해경로를 갱신하는 것이 아니라 각 단계별로 표정들의 분포만 나타날 뿐, 애니메이터가 마음에 드는 단계를 선택하여 항해경로를 갱신한다. 그리고 현재의 항해경로가 마음에 들지 않거나 그 근처에 있는 점들이 세부적인 항해경로의 점으로서 적합하지 않으면, 다시 이전 단계로 단계를

변형하여 새로운 경로를 정한 다음 다시 새로운 단계로 변형하여 갱신하도록 한다.

계층적 가시화의 마지막 단계는 애니메이터가 단계를 최대한으로 하였을 때이고, 약 2400여개의 표정들이 모두 분포되어지는 단계이다. 애니메이터는 마지막 단계에서 기존의 항해경로 근처에 표시된 후보 키 프레임들 중에서 기존의 항해경로와 가장 인접한 특정 키 프레임을 선택하고, 그림 7과 같이 수정 및 갱신한다.

애니메이터의 항해 과정이 끝나면 애니메이터의 항해 경로에 해당되는 얼굴 표정들을 연속적으로 보고 얼굴 표정의 변화를 확인할 필요가 있다. 이때에는 애니메이터의 항해 경로를 처음부터 끝까지 자동으로 반복 항해하여 연속된 얼굴 표정의 변화를 3차원 얼굴 모델이 보여주게 된다. 그림 7의 항해경로에 존재하는 적십자는

이때의 항해 경로를 따라가는 현재의 경로 위치를 가리켜주기 위한 지시자이다.

그림 8은 그림 7의 항해경로상의 적십자를 따라가면서 나타나는 대표적인 얼굴 표정이고, 그림 5로부터 시작된 계층적 가시화의 결과를 보여준다.

5. 결론

본 논문에서는 적당한 공간에 분포시킨 다량의 얼굴 모션 데이터를 계층적 가시화 기법에 의하여 애니메이터가 이 공간을 항해하면서 원하는 얼굴 표정들을 실시간으로 선택하고 디스플레이 하는 방법을 기술하였다. 계층적 가시화를 위해 초기 단계인 약 10개 후보 키 프레임들 시작으로 마지막 단계인 약 2400여개의 후보 키 프레임으로 구분된 항해 공간을 구성하였으며, 이를 위해서 FCM을 사용하였다. 각 단계별 FCM에 의해서 생성한 클러스터 센터들은 애니메이션을 위한 후보 키 프레임으로 설정하고, 이들 후보 키 프레임들 중 애니메이터로 하여금 선택하게 하여 키 프레임으로 설정하였다. 실시간 얼굴 표정 애니메이션을 생성하기 위해서 개발한 사용자 인터페이스는 단계별로 설정된 계층적 가시화 및 애니메이터에 의한 실시간 항해를 실험적으로 확인하는데 유용하게 사용되었다. 특히 계층적 가시화 초기 단계에서 약 10개 정도의 후보 키 프레임만을 표시하고 항해하게 함으로써, 애니메이터가 표정공간에 분포된 약 2400여개의 표정들 중에서 대표적인 표정의 상태와 위치를 보다 쉽고 빠르게 파악할 수 있었다. 그리고 애니메이터가 선택한 특정 단계에서는 초기 단계에서 생성한 항해경로를 가능한 유지하면서 기존의 항해 경로와 가장 인접한 점들을 지나도록 항해 및 갱신할 수 있었다. 이로 인하여 애니메이터가 생성하고자 하는 얼굴 표정 애니메이션의 질(Quality)도 매우 효율적인 것으로 판단되었다.

참 고 문 헌

[1] Demetri Terzopoulos, Barbara Mones-Hattal, Beth Hofer, Frederic Parke, Doug Sweetland, Keith Waters, "Facial animation : Past, present and future," Panel, SIGGRAPH97.

[2] Frederic I. Parke, Keith Waters. "Computer facial animation," A K Peters, 1996.

[3] Brian Guenter, Cindy Grimm, Daniel Wood, Henrique Malvar, and Frederic Pighin. "Making Faces," In SIGGRAPH 98 Conference Proceedings. ACM SIGGRAPH, July 1998.

[4] M. Escher, G. Sannier, N. Magnenat-Thalmann, "Real-Time Interactive Facial Animation," WSCG'99, Pilzen, 1999.

[5] F. Pighin, J. Hecker, D. Lichinski, R. Szeliski, and D. H. Salesin, "Synthesizing realistic facial expressions from photographs," in Proc. SIGGRAPH '98, pp.75-84, July 1998.

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

[7] Erol Fatih, Gudukbay Ugur, "An Interactive Facial Animation System," WSCG'2001 Conference Proceeding, 2001.

[8] Zhang Y., Prakash E., Sung E., "Real-time Facial Expression Animation on An Individualized Face Using Adaptive Simulation Algorithm," September 16 - September 21, GraphicCon'2002, 2002.

[9] Michael Gleicher, "Retargetting motion to new characters," Proceedings of SIGGRAPH 98. In Computer Graphics Annual Conference Series, 1998.

[10] Kwangjin Choi and Hyeongseok Ko, "On-line motion retargetting," In Proceedings of the International Pacific Graphics '99, 1999.

[11] Jehee 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.

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

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

[14] J.C Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.

[15] R.O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis," Wiley, New York, 1973.

[16] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters," Journal of Cybernetics 3: 32-57, 1973.

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

[18] Ya Chang, Changbo Hu, and Turk, M., "Manifold of Facial Expression," In IEEE International Workshop on Analysis and Modeling of Faces and Gesture, pp. 28-35, Oct. 17, 2003.

[19] Ying Zhao and George Karypis, "Evaluation of hierarchical clustering algorithms for document datasets," pages 515-524. ACM Press, 2002.

[20] S. C. Johnson, "Hierarchical Clustering Schemes," Psychometrika, 2:241-254, 1967.

[21] Seber, G.A.F., "Multivariate Observations," Wiley, New York, 1984.



김 성 호

1996년 상지대학교 이공과대학 전산학과 졸업(학사). 1998년 숭실대학교 일반대학원 컴퓨터학과(공학석사). 2001년 숭실대학교 일반대학원 컴퓨터학과(박사수료)
 1997년~1999년 숭실대학교 전자계산원 시간강사. 1999년~2000년 숭실대학교 컴퓨터학부 시간강사. 1999년~2002년 숭의여대 인터넷정보과 시간강사. 숭의여대 컴퓨터게임과 시간강사. 2002년~현재 숭의여대 정보통신계열 멀티미디어콘텐츠전공 겸임교수
 관심분야는 컴퓨터 그래픽스, 모션 캡처 애니메이션, 가상현실, Web3D, 멀티미디어 등



정 문 렬

1980년 서울대학교 계산통계학과 졸업(학사). 1982년 KAIST 전산학과(공학석사). 1992년 펜실베니아 대학교 전산학과(공학박사). 1992년~1994년 일본 구주공과대학교 정보공학부 조교수. 1994년~1999년 숭실대학교 컴퓨터학부 부교수
 2000년~현재 서강대학교 영상대학원 미디어공학과 부교수
 관심분야는 컴퓨터 그래픽스, 디지털 방송 등