

모션 데이터를 사용한 대화식 실시간 얼굴 애니메이션 (Interactive Realtime Facial Animation with Motion Data)

김 성 호(Sung-Ho Kim)¹⁾

요 약

본 논문은 광학식 얼굴 모션 캡처 데이터를 재사용하여 원하는 얼굴 표정 애니메이션을 실시간에 생성하는 방법을 기술한다. 이 방법의 핵심요소는 얼굴 표정들 간의 거리를 정의하고 이를 이용하여 표정들을 적당한 공간에 분포시키는 방법과 이 공간을 사용하여 실시간 표정 애니메이션을 생성하기 위한 사용자 인터페이스 기법이다. 우리는 약 2400여개의 얼굴 표정 프레임 데이터를 이용하여 공간을 생성하였다. 그리고 사용자가 이 공간을 자유롭게 항해할 때, 항해경로 상에 위치한 얼굴 표정 프레임 데이터들이 연속적으로 선택되어 하나의 애니메이션이 생성되도록 하였다. 약 2400여개의 얼굴 표정 프레임 데이터들을 직관적인 공간상에 분포하기 위해서는 데이터와 데이터 사이의 거리를 계산해야할 필요가 있고, 각 데이터와 데이터 사이의 최단거리를 구하는데 있어서는 Floyd 알고리즘을 사용하며, 이를 이용하여 Manifold distance를 구한다. 직관적인 공간에서의 데이터 배치는 얼굴 표정 프레임 데이터들의 Manifold distance를 이용하여 Multidimensional Scaling을 적용하고, 이로부터 2D 평면에 균일하게 분포하였다. 우리는 이와 같은 방법으로 기존의 얼굴 표정 프레임 데이터들 사이의 거리를 원형 그대로 살리면서 의미 있게 직관적인 공간에 분포한다. 그러므로 본 논문에서 제시한 기법은 사용자가 항해하고자 하는 얼굴 표정 프레임 데이터들이 항상 주변에 존재할 수 있기 때문에, 얼굴 표정 애니메이션을 생성하기 위해서 직관적인 공간을 제한 없고 자유로운 항해를 할 수 있다는 큰 장점을 가지고 있다. 또한 사용자가 원하는 얼굴 표정 애니메이션을 사용하기 쉬운 사용자 인터페이스를 이용하여 실시간으로 생성하여 확인하고 재생성할 수 있다는 것도 매우 효율적이다.

ABSTRACT

This paper presents a method in which the user produces a real-time facial animation by navigating in the space of facial expressions created from a great number of captured facial expressions. The core of the method is define the distance between each facial expressions and how to distribute into suitable intuitive space using it and user interface to generate realtime facial expression animation in this space. We created the search space from about 2,400 captured facial expression frames. And, when the user free travels through the space, facial expressions located on the path are displayed in sequence. To visually distribute about 2,400 captured facial expressions in the space, we need to calculate distance between each frames. And we use Floyd's algorithm to get all-pairs shortest path between each frames, then

1) 정회원 : 송의여대 정보통신계열 겸임교수

get the manifold distance using it. The distribution of frames in intuitive space apply a multi-dimensional scaling using manifold distance of facial expression frames, and distributed in 2D space. We distributed into intuitive space with keep distance between facial expression frames in the original form. So, The method presented at this paper has large advantage that free navigate and not limited into intuitive space to generate facial expression animation because of always existing the facial expression frames to navigate by user. Also, It is very efficient that confirm and regenerate with realtime generation using user interface easy to use for facial expression animation user want.

1. 서론

인간은 자신의 감정을 말(언어)보다는 다양한 얼굴 표정을 통해서 가장 잘 표현하며, 또한 우리는 상대방의 얼굴 표정을 보고 상대방의 현재 감정상태를 파악한다. 이와 같은 인간의 얼굴 표정은 수많은 감정의 복합적인 표현이므로 인간의 내면을 표출해주는 가장 일반적인 방법이다. 지금까지 인간은 3D 컴퓨터 그래픽스 기법을 통해서 인간의 얼굴 표정을 가장 잘 표현할 수 있는 방법을 수 없이 연구[1][2][3][4][5][6][7][8][9][10][11]해왔으며 지금도 수많은 연구를 진행하고 있다. 그러나 지금까지의 3D 얼굴 표정 애니메이션은 특정 인물의 얼굴 표정 애니메이션에 국한되어져 있었고 아직까지 많이 미흡한 실정이다. 또한 최근 모션 캡처를 사용한 캐릭터 애니메이션이 컴퓨터 애니메이션 분야에서 각광받으면서 얼굴 모션을 캡처하여 3D 캐릭터 애니메이션에 적용하기도 한다. 그러나 이들 또한 특정 캐릭터의 3D 얼굴 모델에 제한적인 얼굴 표정 애니메이션을 생성하는데 그치고 있고, 일부에서는 얼굴 표정 모션 데이터를 서로 다른 얼굴 모델에 적용하는 재사용(Retargeting) 사례도 최근에는 연구 개발되고 있으나 제한된 모션 데이터를 사용한 얼굴 표정 애니메이션에 대한 재사용일 뿐 사용자가 생성하고자 하는 얼굴 표정을 자유롭게 표현할 수는 없다. 그러므로 본 논문은 얼굴 표정을 전문적으로 연출하는 팬터마임 전문가 한 명(이하 '액터')을 섭외하여 심리학적으로 인간이 표현할 수 있는 수백

개의 다양한 얼굴 표정을 연출하게 하고, 이를 3D 데이터화하여 직관적인 공간에 특정한 법칙에 의해서 분포시킨다. 그리고 사용자가 직관적인 공간을 자유자재로 항해하도록 하여 사용자가 항해한 항해경로에 해당되는 얼굴 표정 프레임 데이터를 3D 얼굴 모델에 실시간 적용하여 다양한 얼굴 표정 애니메이션을 생성하도록 하는 것을 목표로 한다. 먼저 우리는 [그림 1]과 같이 한 액터의 얼굴에서 인간이 다양한 얼굴 표정을 연출할 때 주로 많이 사용되는 얼굴 주 근육 부분에 작은 반사 마커 100개를 부착하고 액터로 하여금 다양한 얼굴 표정을 연출하게 한다.



[그림 1] 액터의 얼굴에 부착한 마커 100개의 위치와 크기 및 얼굴 모션 캡처 장면

[Fig. 1] Position and size of 100 markers attached to face of actor and scene for facial motion capture

본 논문에서 우리는 얼굴 모션 캡처를 위하여 [그림 1]과 같이 Motion Analysis 사의 광학식

모션 캡처 시스템을 사용하였으며, 특수 카메라 8대를 액터의 얼굴 전면의 적당한 위치에 배치하고 초당 60프레임으로 498개의 다양한 얼굴 표정을 캡처하였다. 지금까지 우리는 얼굴 애니메이션과 관련된 연구 및 얼굴 표정 모션 캡처 데이터를 얻기 위한 준비 과정에 대하여 간략하게 살펴보았다. 제 2 장에서는 실험을 위한 얼굴 모션 캡처 데이터의 준비 작업에 대하여 언급할 것이며, 제 3 장에서는 직관적인 공간에 얼굴 표정 프레임 데이터들을 분포시키기 위한 사전 작업으로서 모든 얼굴 표정 프레임 데이터들의 Manifold distance를 구한다.

또한 우리는 구해진 Manifold distance를 이용하여 2D 평면상에 배치하기 위한 Multidimensional Scaling(이하 'MDS')에 대하여 기술한다. 그리고 제 4 장에서는 MDS에 의해서 2D 평면에 분포된 얼굴 표정 프레임 데이터들 위로 사용자의 항해와 항해에 의해서 생성된 실시간 얼굴 표정 애니메이션을 보여주기 위한 사용자 인터페이스 및 구현에 대하여 기술한다. 제 5 장에서는 본 논문의 결론 및 향후 연구에 대하여 기술한다.

2. 사전 작업

본 논문의 실험을 위해서 우리가 캡처한 얼굴 모션 데이터는 프레임 당 100개의 마커별 3D 위치 값 즉, X, Y, Z 위치 값을 가지고 있어서 한 프레임 당 300개의 실수 값을 가지고 있다. 본 연구를 위해서 우리가 캡처한 얼굴 표정 데이터들은 액터가 연출한 표정의 종류에 따라 캡처한 데이터의 프레임 수가 다르다. 즉, 우리가 모션 캡처한 얼굴 표정 데이터는 얼굴 표정의 유형에 따라서 작게는 46프레임에서 많게는 2,284프레임까지 다양하게 구성되어져 있다.

우리는 총 97000여개의 얼굴 표정 프레임으로 구성된 498개의 얼굴 모션 데이터를 이용하여 Manifold distance를 계산하고, 이를 논문에서

실험에 사용하고자 하였다. 그러나 우리는 각 얼굴 표정 프레임 데이터들 사이의 최단 거리를 구하는 알고리즘의 시간 복잡도가 $O(n^3)$ 만큼의 계산 시간을 요구하기 때문에, [그림 2]와 같이 10개의 얼굴 표정 이미지에 해당되는 2400여개의 얼굴 표정 프레임 데이터를 실험에 사용하였다.



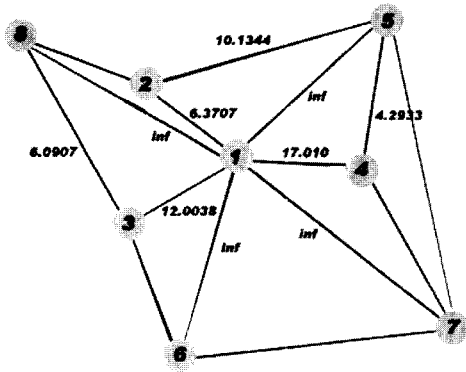
[그림 2] 얼굴 표정 모션 캡처를 위해 액터가 연출할 연극배우들의 다양한 얼굴 표정
[Fig. 2] Some facial expressions of player for facial expression motion capture

3. 얼굴 표정 프레임 데이터 분포

본 논문에서 우리는 실험에 사용할 약 2400여개의 얼굴 표정 프레임 데이터들을 직관적인 공간에 분포시키기 위해서 프레임과 프레임사이의 Manifold Distance를 계산할 필요가 있다. 왜냐하면, 우리가 사용할 얼굴 표정 프레임 데이터는 모든 프레임들이 시간을 축으로 연속적으로 생성된 프레임 데이터이기 때문에 프레임마다 인접한 프레임들이 존재하기 때문이고, 데이터 자체의 수치적인 의미뿐만 아니라 시각적으로 보여 지는 것도 중요하기 때문이다. 그러므로 우리는 직관적인 공간에 얼굴 표정 프레임 데이터들을 분포시키기 위한 준비과정으로서 약 2400여개의 얼굴 표정 프레임 데이터들 간의 Manifold Distance를 계산한다.

3.1 Manifold distance

일반적으로 그래프에서 노드와 노드 사이의 최단 거리를 구할 때에는 *Dynamic Programming* 많이 사용하며, 특히 Floyd 알고리즘은 모든 점에서 출발해서 출발 한 정점을 제외한 모든 정점을 도착점으로 하는 최단거리를 구하는 알고리즘이기 때문에 본 논문에서는 Floyd 알고리즘을 사용하여 얼굴 표정 프레임과 프레임간의 최단 거리를 구한다. Floyd 알고리즘의 입력 데이터는 2400여개의 얼굴 표정 프레임 데이터들을 이용하여 그래프 형태로 구성한 것으로서 얼굴 표정 프레임과 프레임 사이의 거리를 에지로 설정하고 이를 대칭 행렬로 구성한 것이다. 일반적으로 그래프에서 노드들 간의 최단거리라 함은 직선거리인 Euclidean distance를 의미한다. 그러나 본 논문에서 구하고자 하는 거리는 Manifold distance로서, 각 얼굴 표정 프레임으로부터 임의의 얼굴 표정 프레임까지의 거리를 구하기 위해서는 각 얼굴 표정 프레임마다 인접한 프레임이 있어야 하며 인접한 프레임을 거쳐서 임의의 얼굴 표정 프레임까지 도달하는 최단 거리를 구해야한다.



[그림 3] 각 얼굴 표정 프레임들 사이의 최단 거리를 구하기 위한 그래프 생성 방법

[Fig. 3] The graph generation method for all-pairs shortest path between each facial expression frames

그리고 각 얼굴 표정 프레임들의 인접 프레임을 구하기 위해서는 임의의 거리 값을 이용하여 인접거리 상한선으로 설정해야할 필요가 있다. 인접 거리 상한선 이하의 거리를 가지는 프레임은 인접한 얼굴 표정 프레임 데이터로 간주하여 프레임과 프레임간의 거리를 적용하고, 인접 거리 상한선보다 큰 거리 값을 가지는 프레임들까지의 거리는 거리가 존재하지 않기 때문에 무한대 값을 적용한다. 물론 본 논문에서 적용한 인접거리 상한선을 설정하는 것과 관련된 특별한 규정은 없으며, 임의로 인접거리 상한선의 범위를 수 개 설정하고 실험을 통하여 확인해야만 한다. 즉, [그림 3]과 같이 우리는 현재의 얼굴 표정 프레임을 1번이라고 가정할 경우 1번 프레임을 중심으로 인접거리 상한선(원형)의 영역 내에 있는 2번, 3번 및 4번 얼굴 표정 프레임을 인접 프레임으로 설정하여 직선거리인 Euclidean distance를 적용하였으며, 5, 6, 7, 8번 프레임까지는 인접 거리 상한선보다 멀리 떨어져 있기 때문에 거리가 존재하지 않아서 무한대 값을 적용한다. 우리는 이와 같은 과정을 2400여개의 모든 얼굴 표정 프레임 데이터에 적용하고, 정사각형 형태의 행렬로 표현하여 Floyd 알고리즘의 입력 데이터로 사용하고 <표 1>과 같은 정사각형 형태의 Manifold distance를 계산하였다.

<표 1> 각 얼굴 표정 프레임별 인접거리 상한선(30)에 의해서 계산된 정사각형 형태의 Manifold distance (Table 1) Square-form manifold distance calculated by upper limit(30) of adjacent distance between each facial expression frames

Frame	1	2	3	...	2406
1	0	6,3707	12,0038	...	87,3677
2	6,3707	0	6,0907	...	81,4968
3	12,0038	6,3907	0	...	75,7700
4	17,2010	11,1925	5,4657	...	70,3043
5	22,4206	16,1941	10,4168	...	65,4547
...
2406	87,3677	81,4968	75,7700	...	0

본 논문에서 우리가 계산한 Manifold distance는 각 얼굴 표정 프레임 데이터로부터 모든 얼굴 표정 프레임 데이터까지의 연속적인 얼굴 표정의 분포를 형성하기 위해서 필요하다. 또한 Manifold distance를 계산하여 형성된 얼굴 표정 프레임 데이터들의 구성은 직관적인 공간에 배치하기 위해서 필요하며, 얼굴 표정 프레임 데이터들을 직관적인 공간에 배치하기 위해서는 Multidimensional Scaling을 적용한다.

3.2 MDS (Multidimensional Scaling)

MDS는 고차원으로 구성된 데이터들 사이의 거리 또는 비유사성을 측정하고, 이를 이용하여 저차원인 2D 또는 3D 공간상의 점으로 표현함으로써 데이터들 간의 상대적인 위치를 표시하고, 데이터들 사이의 집단화를 시각적으로 표현하는 통계적 분석방법이다. 본 논문에서 우리는 얼굴 표정 프레임들의 수 k , n , 얼굴 표정 프레임들 사이의 Manifold distance로 구성된 비유사성인 대칭 행렬 d_{ij} , 얼굴 표정 프레임 데이터를 대표하며 2-차원 평면에 존재하는 좌표 값들의 집합 $y_i, i = 1, 2, \dots, k, y_i \in \mathbb{R}^2$ 등이 주어질 때, d_{ij} 와 가능한 가장 가까운 근사 값을 가지고 있는 Euclidean distance δ_{ij} 으로 이루어진 저차원 평면에 y_i 를 배치하기 위해서 MDS를 사용한다. 그리고 d_{ij} 와 저차원 평면에 투영된 거리 값 δ_{ij} 사이의 일치성을 측정하는 것은 식 (1)과 같이 Sum-of-squares 함수를 기반으로 한다.

$$\Phi_n = \sum_{i=1}^n \sum_{j=i}^n (d_{ij} - \delta_{ij})^2 \quad (1)$$

즉, 함수 Φ_n 는 2-차원 평면상의 특징 점으로 구성된 $y_i, i = 1, 2, \dots, k, y_i \in \mathbb{R}^2$ 이 입력으로 주어졌을 때 식 (1)과 같은 Sum-of-squares 함수로 계산되며, 얼굴 표정 프레임 공간상의 임의

의 점 i, j 간의 거리 d_{ij} 와 이 점들을 대표하는 2-차원 공간상의 점 i, j 간의 거리 δ_{ij} 와의 차의 제곱의 합을 구한다. 그리고 우리는 식 (1)의 값이 최소 값이 될 때까지 y_i 값을 재입력하고 반복 계산하며 더 이상 변경되지 않을 때 반복과정을 멈추게 된다.

우리는 식 (1)에서 입력으로 사용될 2-차원 평면상의 초기 좌표 값들의 집합 y_i 를 생성하기 위해서 스펙트럴 분해를 이용하며, 다음과 같이 3단계로 구성된 Torgerson의 Classical MDS를 위한 Scalar Product 알고리즘 [13]을 이용한다.

단계 1 : 각 얼굴 표정 프레임 데이터들의 오리지널 거리 값으로 구성된 비유사성 행렬 D 인 d_{ij} 로부터 Double-Centering 행렬 $B = b_{ij} = HAH$ 를 계산한다.

$$b_{ij} = - \left(\frac{\bar{d}_{ij}^2 - \bar{d}_i^2 - \bar{d}_j^2 + \bar{d}_{..}^2}{2} \right) \quad (2)$$

여기서 \bar{d}_{ij}^2 의 ²는 정사각의 비유사성 행렬을 Double Center한다는 의미로서 b_{ij} 는 'Double Centering'이라고 한다. Double Centering은 비유사성 행렬 D 의 각 원소로부터 각 행의 평균과 열의 평균을 빼고 모든 원소의 평균을 더하며 $-1/2$ 을 곱하는 것을 의미한다. 즉, 식 (2)에서 \bar{d}_i 는 행렬 D 의 i 번째 행의 평균, \bar{d}_j 는 j 번째 열의 평균 그리고 $\bar{d}_{..}$ 는 모든 원소의 평균을 의미한다. 만약 d_{ij} 가 거리 값을 가지고 있는 유사성 대칭 행렬이라면 b_{ij} 는 내적(Inner Product) 행렬로서

$$b_{ij} = (x_i - \bar{x})(x_j - \bar{x}) \quad (3)$$

식 (3)과 같이 계산되어진다. 그리고 우리는 식 (2)를 행렬 형식인 HAH 로 표현할 수 있는데,

여기서 $H = I - \frac{J}{N}$ 로서, I 는 항등 행렬, J 는 d_{ij} 의 크기가 N 일 경우 모두 1의 값을 가진 행렬 $N \times N$ 을 의미하며, $A = (a_{ij}) = -\frac{d_{ij}^2}{2}$ 을 의미한다. 본 논문에서는 HAH 를 사용하여 *Double Centering* 행렬 B 를 계산하였다.

단계 2 : 행렬 B 의 스펙트럴 분해는 다음 식 (4)와 같이 한다.

$$B = VD_{\lambda}V' \quad (4)$$

여기서 D_{λ} 는 *Eigenvalue*로서 $\lambda_1 \geq \dots \lambda_p > 0$ 를 대각원소로 하는 대각행렬이고, V 는 *Eigenvector*로서 v_1, \dots, v_p 를 열로 하는 직교행렬이다. 식 (4)는 단계 1에서 구한 행렬 B 를 입력으로 하여 계산한 *Eigenvector*, *Eigenvalue*와 같은 값을 가지며, 본 논문에서는 이를 사용한다.

단계 3 : 행렬 B 의 스펙트럴 분해로부터 $m (\leq p)$ 의 *Eigenvalue*와 이에 대응하는 *Eigenvector*를 가지고 크기가 $n \times m$ 인 행렬 $C = (v_1\sqrt{\lambda_1}, \dots, v_m\sqrt{\lambda_m})$ 를 계산한다. 만약 행렬 C 에서 $m = 1$ 이면 1-차원의 공간 좌표 값을 의미하고, $m = 2$ 인 경우는 2-차원의 공간 좌표 값을 의미한다.

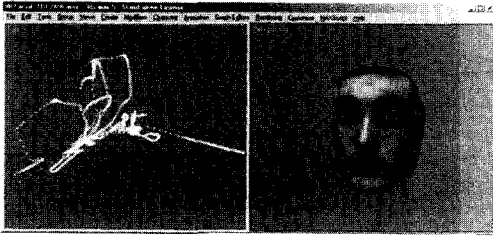
이와 같은 과정을 통하여 구해진 m -차원의 공간 좌표 값을 이용하여 우리는 MDS의 사용 목적에 따라 행렬 C 에서 적절한 차원의 수 m 을 찾아내는 것을 고려해야 한다. 그러나 본 논문에서는 얼굴 표정 프레임 데이터를 2D 평면에 배치하기 위한 것이 목적이기 때문에 이를 고려할 필요 없이 $m = 2$ 로 설정한다. 그리고 우리는 행렬 C 로부터 추출한 $m = 2$ 차원의 값을 식 (1)의 y_i 초기 좌표 값으로 사용한다. 또한 식 (1)의 값이 최소 값이 되어 더 이상 변경하지 않을 때가

지 새로운 y_i 를 재입력하여 반복하여야 한다. 그러나 본 논문에서 적용한 *Torgerson*의 *Classical MDS*를 위한 *Scalar Product* 알고리즘 [13]은 식 (1)에 사용될 y_i 의 최적의 값을 계산한 것이고, 우리가 구하고자 하는 MDS의 최종 결과이므로 이를 2D 평면에 사용할 좌표 값으로 설정하고 직관적인 공간에 분포시킨다.

4. 사용자 인터페이스 설계 및 구현

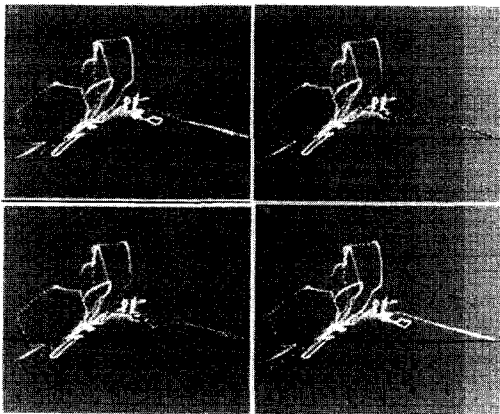
우리는 2400여개의 얼굴 표정 프레임 데이터를 사용하여 *Manifold distance*를 계산하고, 이를 직관적인 공간에 배치하기 위해서 MDS를 적용하였다. 그리고 MDS에 의해 나타난 최종 결과 y_i 는 [그림 4]와 같은 사용자 인터페이스를 기반으로 왼쪽과 같이 2D 평면에 2400여 개의 얼굴 표정 프레임 데이터를 연결하여 분포되어 있으며, 각 얼굴 표정 프레임 데이터들은 인접한 얼굴 표정 프레임 데이터들을 주변에 배치하고 있다. 이제 우리는 사용자로 하여금 MDS에 의해서 직관적인 2D 공간에 분포된 2400여개의 얼굴 표정 프레임 데이터 위로 마우스 등을 사용하여 자유자재로 항해하게 하고, 항해 경로에 해당되는 얼굴 표정 프레임 데이터들을 [그림 4]의 오른쪽 3D 얼굴 모델에 실시간으로 적용한다.

[그림 5]는 MDS에 의해서 직관적인 공간에 분포된 2400여개의 얼굴 표정 프레임 데이터들 위로 사용자가 항해하는 과정과 항해경로를 표현한 것이고, 오른쪽 아래 그림의 적십자는 항해 경로를 따라 얼굴 표정 프레임 애니메이션이 진행되는 현재 위치의 얼굴 표정 프레임 데이터를 가리켜주기 위한 프레임 지시자이다. 그리고 [그림 6]은 사용자가 다수의 항해 경로 즉, 한 프레임에서 출발하여 특정 프레임에서 멈추는 항해 경로를 여러 개 생성하여 서로 다른 색상으로 표현하고 있으며, 사용자가 항해한 항해 경로들 중 임의의 경로를 하나를 선택하여 재생할 수 있다.



[그림 4] (Left) MDS 결과를 직관적인 공간인 2D 평면에 배치한 모습, (Right) 사용자가 왼쪽의 직관적인 공간을 항해할 때 항해 궤도에 해당되는 얼굴 표정 프레임들 실시간으로 보여주기 위한 3D 얼굴 모델

[Fig. 4] (Left) Distributed shape over intuitive 2D space with MDS result, (Right) 3D facial model to display with realtime a facial expression frame come under navigation orbit, when user navigate over intuitive 2D space.



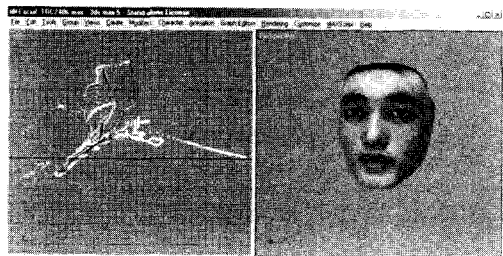
[그림 5] (Left Top) 직관적인 공간인 2D 평면에 배치된 얼굴 표정 프레임들, (Right Top) 직관적인 공간인 2D 평면상에서의 사용자 항해 과정, (Left Bottom) 사용자의 항해 종료 후에 표시된 항해 궤도, (Right Bottom) 항해 궤도를 재생하는 과정에서 보여 지는 현재 프레임 위치 (적색의 '+')

[Fig. 5] (Left Top) Facial expression frames distributed over intuitive 2D space, (Right Top) User navigation over intuitive 2D space, (Left Bottom) Navigation orbit displayed by user navigation, (Right Bottom) Position of current frame in process of replay of navigation orbit (red color mark '+')

우리는 얼굴 표정 프레임 데이터를 사용하여 Manifold distance를 계산할 때 인접거리의 상한선을 설정하여 각 얼굴 표정 프레임 데이터로부터 인접 프레임을 추출하였는데, 인접거리 상한선을 어떻게 설정하는가에 따라서 조금씩 다른 결과가 나온 것을 알 수 있었다. 즉, 우리는 [그림 7]과 같이 인접거리 상한선을 30, 40, 50으로 주었을 때의 MDS 결과가 서로 다른 모습으로 나타나고 있으며, 인접 거리 상한선을 60, 70, 80, 90, 100 등으로 설정하였을 경우에는 MDS의 결과가 유사하다는 것을 알 수 있었다.

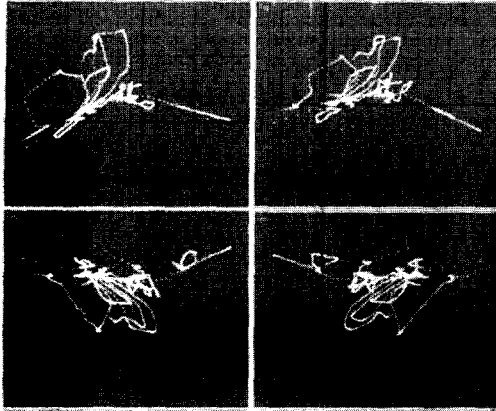
5. 결론 및 향후 연구

본 논문에서는 광학식 모션 캡처 시스템을 사용하여 한 액터로부터 수백 개의 다양한 얼굴 표정을 모션캡처 하였으며, 캡처한 데이터 중 실험을 위해 2400여개의 얼굴 표정 프레임 데이터를 사용하여 직관적인 공간에 의미 있게 분포시키기 위해서 Manifold distance를 구하였다.



[그림 6] (Left) 다수의 항해 궤도, (Right) 다수의 항해 궤도들 중 하나의 항해 궤도를 선택하고 재생하여 얼굴 표정을 실시간으로 보여주기 위한 3D 얼굴 모델

[Fig. 6] (Left) Several navigation orbits, (Right) 3D facial model to display realtime a facial expression animation with replay and select a orbit among several navigation orbits.



[그림 7] 인접거리 상한선에 따른 MDS 결과, (Left Top) 인접거리 상한선 : 30, (Right Top)인접거리 상한선 : 40, (Left Bottom) 인접거리 상한선:50, (Right Bottom)인접거리 상한선 : 60, & 70, 80, 90, 100과 유사

[Fig. 7] MDS result yield to upper limit of adjacent distance, (Left Top) upper limit of adjacent distance : 30, (Right Top) 40, (Left Bottom) 50, (Right Bottom) : 60, similar to 70, 80, 90, 100

즉, 우리는 얼굴 표정 프레임들 간의 최단 거리를 Floyd 알고리즘을 사용하여 계산하였고, 이를 위하여 각 얼굴 표정 프레임마다 인접거리 상한선을 설정하여 인접 프레임을 설정하였다. MDS는 고차원의 데이터를 저차원으로 줄여주는 기능을 하는 것으로서, 본 논문에서 제시한 Manifold distance와 MDS는 기존의 연속적인 성질을 가지고 있는 얼굴 표정 프레임 데이터들의 특성(거리)을 그대로 살리면서 2D 평면에 골고루 분포시키기 위한 최선의 방법으로 판단된다. 그러므로 사용하는 MDS 결과로 2400여개의 얼굴 표정 프레임 데이터를 2D 평면에 분포한 직관적인 공간을 자유자재로 항해할 경우 사용자가 원하는 매우 자연스러운 얼굴 표정 애니메이션을 실시간으로 생성할 수 있었으며, 또한 애니메이터로 하여금 확인 및 검증 절차를 거쳐 실시간으로 재생성도 가능하였다. 그러나 우리는 2400여개의 얼굴 표정 프레임 데이터가 분포된 직관적인 2D 평면 위로 사용자가 항해를 할 경우, 얼굴 표정의 분포가 전

체적으로 어떻게 구성되어 있는지 알 수 없기 때문에 수차례의 반복적인 경험에 의존할 수밖에 없다는 단점도 있었다.

우리는 본 논문에서 300-차원으로 구성된 2400여개의 얼굴 표정 프레임 데이터를 MDS를 사용하여 2-차원으로 줄이고 이를 이용하여 직관적인 2D 공간에 배치하여 실시간 얼굴 표정 애니메이션에 사용하는 실험을 하였다. 그러나 우리는 향후 본 논문에서 사용한 MDS 외에 SOM (Self-Organizing Map), CCA(Curvilinear Component Analysis), LLE(Locally Linear Embedding), PCAPROJ(Principal Component Analysis Projection), Sammon's Mapping 등과 같은 선형 및 비선형 투영 알고리즘을 사용하여 같은 실험을 해보아야 하고, 얼굴 표정 애니메이션을 위해서 사용한 모션 캡처 데이터인 얼굴 표정 프레임 데이터를 직관적인 2D 공간에 분포하기 위한 최적의 투영 알고리즘을 찾아야 한다. 또한 본 논문에서 제시한 실시간 얼굴 표정 애니메이션 생성용 사용자 인터페이스는 얼굴 표정 애니메이션을 전문적으로 제작하는 애니메이터들로 하여금 사용하게 하고, 실시간 얼굴 표정 애니메이션 제작상의 문제점이나 보완해야 할 사항들을 찾아 수정하여 보다 효율적인 사용자 인터페이스를 구축하고자 한다.

※ 참고문헌

- [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] 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, "Subdivision 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 3D 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] 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.
- [13] W. S. Torgerson, "Multidimensional scaling : I. theory and method", Psychometrika., 17:401-419, 1952.

김 성 호



1996.2 상지대학교 전산학 전공
(학사)

1998.8 승실대학교 대학원
3D 컴퓨터그래픽스 전공
(석사)

2001.11 승실대학교 대학원
3D 컴퓨터그래픽스 수료
(박사)

1997.6~1999.12 (주)진우 GDS

2000.1~2001.1 (주)조오그래프
선임연구원

2000.6~2001.9 (주)애니큐브
부설 연구소장

2001.10~2003. 현재
(주)포도시스템즈 자문

2002.3~2003. 현재
승의여대 정보통신계열
겸임교수

2003.6~2003. 현재
(주)엠아이엔디 연구실장
(예정)