

고밀도 얼굴 모션 캡처 데이터를 새로운 얼굴로 리타겟팅하는 기법

나경건, 정문렬

서강대학교 영상대학원
미디어 공학과 미디어 랩

요약

본 논문은 사람의 실제 얼굴에서 캡처된 얼굴 모션 데이터를 새로운 얼굴에 재적용하는 리타겟팅 기법을 제안한다. 본 기법은 형태가 매우 상이한 얼굴 모델에도 적용이 가능하며 특히 주름같은 세밀한 모션의 리타겟팅에 적합하다. 본 기법은 다중 해상도 메쉬 즉 노말메쉬(normal mesh)를 사용함으로써 소스와 타겟의 계층적으로 대응관계를 결정하고 계층적으로 리타겟팅한다. 노말 메쉬는 주어진 메쉬를 베이스 메쉬(base mesh)와 일련의 노말 오프셋(normal offsets)을 이용하여 근사시킨 계층적 메쉬이다. 본 리타겟팅 기법은 우선 베이스 모션을 소스 모델에서 타겟 모델로 리타겟팅한 후 그 모션 위에 노말 오프셋의 모션을 계층적으로 더해준다. 이 기법은 형태가 매우 세밀한 모션에 대하여 안정적이면서도 정교한 리타겟팅 결과를 생성한다.

제 1 절 서론

얼굴 애니메이션은 애니메이터들에 의해서 생성이 되고 있으며 실제적으로 얼굴 표정의 애니메이션을 제작할 때 애니메이터들은 매 3-5 프레임마다 얼굴을 조작해 주어야 하는 수고를 아끼지 않는다. 그리고 얼굴 모델이 바뀌면 그 작업을 되풀이한다. 이에 대한 대안으로 보다 쉬운 방법으로 얼굴 애니메이션을 생성하고자 하는 기법들이 많이 제안되었다. Lee, Platt, Waters 그리고 Terzopolous [13, 19, 10, 21] 등은 얼굴 표면 아래에 뼈와 근육 모델을 설정하여 이 모델의 파라미터들을 조절하여 얼굴 애니메이션을 생성했다. 하지만 아직까지 사실적인 근육 모델이 발견되지는 못했다. Parke와 Karla [17, 9]는 얼굴 표면을 의미있는 부위들의 집합으로 나타내고 부위의 움직임을 통해 얼굴 애니메이션을 표현하려 하였다. Pighin과 Guenter [18, 4]는 배우로부터 얼굴의 모양과 모션을 캡처하여 이를 캐릭터의 표정 애니메이션 생성에 사용하였다.

그러나 이 방법들은 한번 생성된 표정 모션을 새로운 얼굴 모델에 적용하는 문제는 생각하지 않고 있다. 최근에는 배우로부터 캡처한 얼굴동작을 새로운 얼굴 모델에 적용 (retargeting) 하는 방법들이 제시되었다. Noh [16]의 연구는 최초의 얼굴 리타겟팅에 관한 연구라고 볼 수 있다. Pyun [20]은 사례기반기법 (example based method)을 이용, 보다 안정적이고 정확한 리타겟팅 결과를 생성하였다. 모션 리타겟팅 기술은 원래 Gleicher [3]에 의해 몸동작 애니메이션에 대하여 최초로 개발되었고 Lee와 Choi [12, 1]에 의해 확장되었다. 그러나 이러한 기술들은 얼굴에 쉽게 적용되지 않는다. 왜냐하면 몸동작 애니메이션은 관절각으로 표현되는 반면에 얼굴 애니메이션은 얼굴표면의 수많은 점들에 의해 표현되기 때문이다.

본 논문의 목적은 실제 배우로부터 고밀도 얼굴 모션 데이터를 캡처해서 새로운 얼굴에 리타겟팅하는 것이다. 본 기법은 Noh [16]의 기법이 가지고 있는 문제점에서 출발하므로 이를 간단히 요약한다. 이 방법은 기본적으로 두 단계로 이루어진다. 첫 단계에서는 소스 메쉬를 타겟의 형태에 맞게 물평한

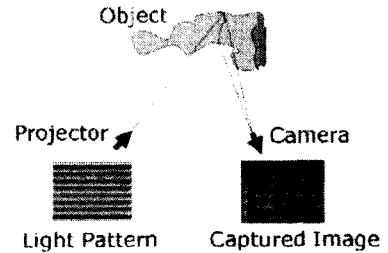


그림 1: 실시간 stripe image 들을 얻기 위한 캡처 시스템

으로써 꼭지점 (vertex) 간 대응관계를 설정한다. 두번째 단계에서는 초기 프레임에서 소스와 타겟의 대응하는 꼭지점의 접평면위에 지역좌표계를 만들고 두 꼭지점이 해당 국지좌표계를 기준으로 동일한 국지변위 (모션벡터)를 가지도록 타겟 꼭지점의 변위를 생성한다. 꼭지점의 모션벡터는 초기프레임에서의 꼭지점 위치와 현재 프레임에서의 꼭지점 위치간의 차이로 표시한다. 이 방법을 표정이 큰 경우나 형태의 차이가 많은 모델에 적용할 경우 이웃한 꼭지점들의 모션벡터가 서로 교차하는 현상 (self-intersection) 이 적지 않게 발생한다.

본 논문에서는 꼭지점 모션벡터들간의 교차를 줄이기 위해 얼굴 모델을 계층적으로 표현하고 꼭지점의 모션벡터도 계층적으로 표현하였다. 계층별로 꼭지점 모션벡터를 표현하면 이들이 서로 교차할 가능성이 줄어든다. 리타겟팅은 모션을 베이스 모션(특징 모션)과 계층적 세부 모션으로 나누어 한다. 본 논문에서는 얼굴 모델을 계층적으로 표현하기 위해 재메쉬화 기법 (Remeshing) [5, 6, 11, 7]을 사용 하였고 다중해상도 메쉬로는 노말메쉬 [5]를 사용하였다.

제 2 절 고밀도 모션 캡처 데이터

본 논문에서 사용한 고밀도 모션 캡처 데이터는 컴퓨터 비전 기술인 structured light method [8, 10, 15]나 3D laser scan 을 이용하여 얻어진 것으로 기존의 적외선 마커를 얼굴에 부착하여 트랙킹한 데이터보다 훨씬 정확하고 상세한 얼굴 표정을 얻을 수 있다. structured light method 는 고속 카메라를 이용, 실시간으로 stripe image sequence 를 얻어낸 후 그것들로부터 3차원 정보를 추출함으로써 매 프레임의 얼굴 표정 데이터를 얻을 수 있는 장점을 가진다. 그림 1 은 비전 기술을 이용한 모션 캡처 시스템이고 그림 2 는 그 결과 얻어낸 임의의 프레임에서의 얼굴 표정 데이터이다.

제 3 절 대응관계의 결정

고밀도 모션 캡처 데이터는 기존의 마커를 얼굴에 부착하여 트

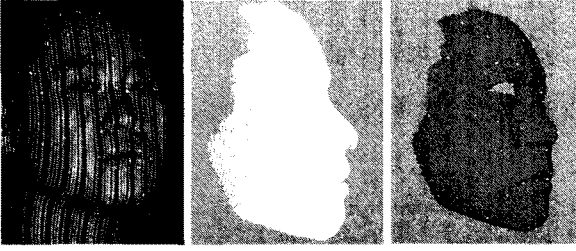


그림 2: 컴퓨터 비전 기술을 이용하여 얻어낸 stripe image(좌) point-cloud data(중) 3D 메쉬(우)

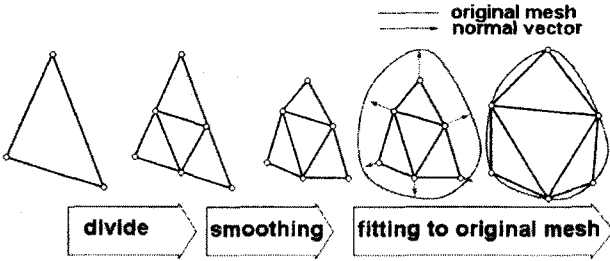


그림 3: 재메쉬화과정

랙킹한 데이터보다 훨씬 정확하고 상세한 얼굴 표정을 얻어 낼 수 있다. 그러나 이 데이터는 어느 한 점이 다음 프레임에서 어느 점에 대응하는지는 알 수 없다. 따라서 그림 2에서와 같이 캡처할 때 배우의 얼굴에 특징점을 찍어 그 특징점들에 대해서는 트래킹함으로써 프레임 간 대응 관계를 알 수 있게 한다. 그리고 세밀한 부분들에 대해서는 매 프레임마다 특징점으로 만든 특징메쉬로부터 재메쉬화(Remeshing)함으로써 세부 대응 관계를 결정한다. 서로 대응되는 특징메쉬를 같은 방식으로 재메쉬화하면 새로 생긴 꼭지점들도 서로 대응된다고 볼 수 있다. 따라서 특징메쉬가 대응되는 두 메쉬의 세부 대응관계는 두 메쉬를 재메쉬화함으로써 결정된다. 소스메쉬와 타겟메쉬의 대응관계도 재메쉬화 과정을 거침으로써 대응관계가 결정된다. 재메쉬화 기법 [5, 6, 11, 7]으로써 노말메쉬 (normal mesh) [5] 기법을 사용하였다. 노말메쉬 (Normal mesh)은 다음과 같이 생성된다. 첫째, 사용자가 지정한 베이스 메쉬에 한번의 서브디비전을 적용하여 서브디비전 메쉬를 만들고 이로부터 원 메쉬 (input mesh)까지의 높이, 즉 법선 방향 오프셋(normal offsets)을 구하여 다음 레벨 메쉬를 만든다. 둘째, 다음 단계 메쉬를 "베이스 메쉬"로 보고 첫째 단계 과정을 반복적으로 적용되 생성된 메쉬가 원 메쉬에 충분히 가까울 때까지 적용한다. 본 논문에서는 타겟 메쉬와 각 프레임에서의 소스 메쉬를 노말 메쉬로 표현한다.

본 논문에서는 서브디비전 과정에서 butterfly scheme [22] 대신 "divide and smoothing" 기법을 사용하였다. 이 기법을 이용하는 모달메쉬 생성과정은 그림 3과 같이 분할 (divide), 스무딩 (smoothing), 원 메쉬까지의 노말 오프셋 구하기 (fitting)를 반복적으로 수행한다. 노말 메쉬 형성 과정에서 스무딩 기법을 사용한 이유는 스무딩이 꼭지점 노말벡터들간의 교차를 감소시키기 때문이다. 그리고 butterfly scheme은 꼭지점 노말벡터들의 방향이 그 이웃들과 차이가 많이 나는 경향이 있다. Loop subdivision scheme [14]을 이용할 경우는 노말벡터 교차 오차는 상당히 적게 나오지만 서브디비전 과정에서 기존의 특징점

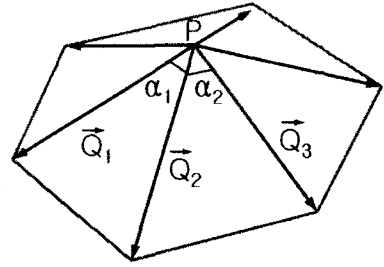


그림 4: Smoothing : Q_i 는 edge vector, P 는 중심 vertex, α_i 는 edge 사이각이다

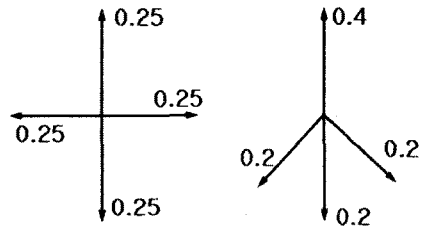


그림 5: edge 방향성에 따른 Floater's weight

이 원래 위치로부터 옮겨진다는 문제가 있다. 본 기법에서는 소스 베이스 메쉬와 타겟 베이스 메쉬상에 있는 특징 꼭지점을 이용하여 두 메쉬간의 대응관계를 설정한다. 따라서 서브디비전 과정에서 이 특징 꼭지점들의 위치가 이동하면 두 메쉬간의 대응관계가 원래 지정하였던 것과 달라지는 문제가 있다. 본 논문에서 사용한 스무딩 알고리즘은 다음 식 1과 같다. 이 알고리즘은 노말벡터들의 방향을 평균화하고 기존의 특징점이 원래 위치로부터 거의 이동하지 않게 해준다.

$$P_{new} = P_{old} + \lambda \sum w_i \vec{Q}_i : \quad (1)$$

$$\lambda = 1 / \sqrt{\sin \sum \alpha_i / 4}$$

여기서 Q_i 는 edge vector, P 는 중심 vertex, α_i 는 edge 사이각이다. w_i 는 Floater's weight [2]으로써 그림 5와 같이 edge들의 방향성에 따라 다른 값을 가진다. 이것은 입의 점에 대하여 본래 위치에서 법선 방향으로의 이동을 유도하면서 옆으로 치우치는 현상을 줄여준다. 즉, 노말메쉬 형성과정에서 기존 특징점의 이동을 무시할 수 있을 만큼 매우 작게 해준다.

그림 6과 그림 7은 노말메쉬를 이용하여 소스 모델의 프레임 간 대응관계와 소스와 타겟 모델간의 대응관계를 결정하는 모습을 보여주고 있다.

그림 7에서 보듯이 캡처할 때 정한 배우얼굴의 특징점들과 똑같은 수만큼 대응하는 타겟 메쉬상의 특징점들을 사용자가 지정하게 된다. 소스 특징점을 연결하여 소스 특징 메쉬를 만들고, 타겟 특징점들을 연결하여 타겟 특징 메쉬를 만든다. 자동으로 결정할 수 있는 특징점들은 이마 꼭대기나 눈, 입, 코의 끝, 턱 밑, 좌우 볼의 끝 등이 있겠으나 형태가 많이 다른 경우에는 자동적인 방법으로 대응관계를 찾기가 어렵다. 따라서 본 기법에서는 특징점간의 대응관계를 사용자가 직관적으로 결정하도록 한다.

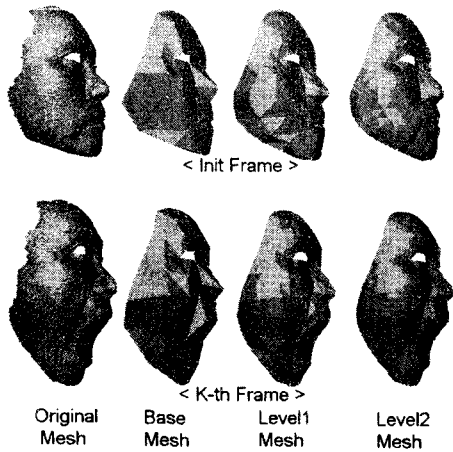


그림 6: 노말메쉬에 의한 소스 모델의 프레임간 대응관계 결정

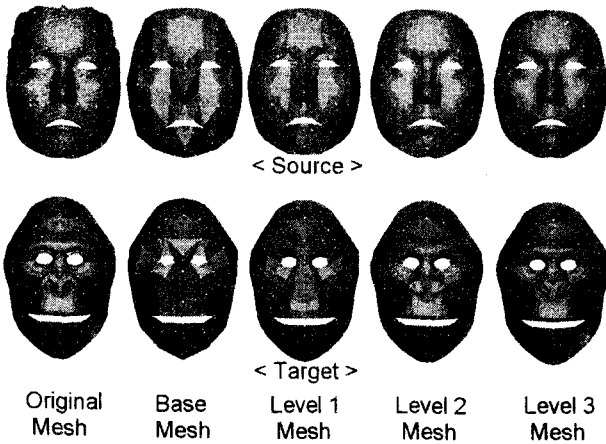


그림 7: 노말메쉬에 의한 소스 모델과 타겟 모델간의 대응관계 결정

제 4 절 계층적 리타겟팅

본 기법은 특징 모션위에 상세한 모션을 계층적으로 더한다. 이에 먼저 특징 메쉬의 리타겟팅을 어떻게 하는지를 기술한다.

4.1 특징 메쉬의 리타겟팅

특징메쉬 꼭지점들의 운동은 임의의 프레임에서의 꼭지점들과 초기프레임에서의 꼭지점들간의 변위(모션벡터)로 나타낸다. 특징메쉬의 리타겟팅을 위해서는 소스 특징 메쉬의 꼭지점 모션벡터를 타겟 특징메쉬상의 대응되는 꼭지점 모션벡터로 변환해야 한다. 이를 위해 본 논문에서는 Noh [16]가 제안한 모션 벡터 기법을 사용하였다. 그러나 베이스 메쉬(특징 메쉬) 특징점에 대해서만 이 기법을 적용하고 상세 메쉬는 재메쉬화 과정에서 구한 계층적 노말 오프셋을 이용하여 리타겟팅한다. 특징메쉬와 상세메쉬로 나누어서 각기 다른 리타겟팅 방식을 적용하는 이유는 원메쉬에 모션 벡터 기법을 적용하면 꼭지점 모션 벡터들이 서로 교차할 가능성이 높아지기 때문이다.

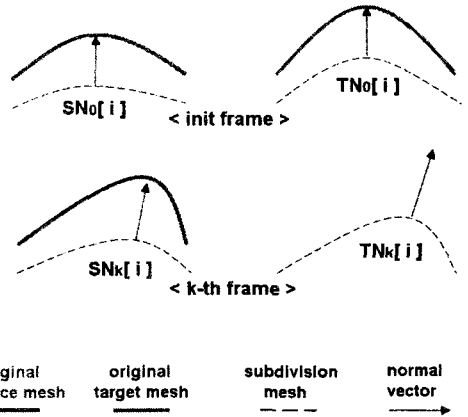


그림 8: k번째 프레임 i단계에서의 상세 메쉬 리타겟팅: 현재 프레임 k의 i단계에서의 타겟 상세 메쉬는 소스의 노말 오프셋의 프레임간(초기 프레임과 현재 프레임) 편차를 초기 프레임에서 구한 타겟의 노말 오프셋에 더함으로써 구해진다.

4.2 상세 메쉬의 리타겟팅

지금까지의 과정에서 얻은 정보는 다음과 같다. 첫째 각 프레임마다 재메쉬화 과정에서 얻은 각 단계의 소스 노말 메쉬의 노말 오프셋, 둘째 초기 프레임에서의 재메쉬화 과정에서 얻은 각 단계의 타겟 메쉬의 노말 오프셋, 셋째 각 프레임마다의 리타겟팅된 타겟 특징 메쉬이다. 이제 임의의 프레임에서의 타겟 특징 메쉬에 뼈대에 살을 입히듯 세부적인 리타겟팅이 적용할 것이다. 이를 위해 우선 초기 프레임과 현재 프레임간의 소스의 노말 오프셋 차이를 구하고 이를 초기 프레임에서 구한 타겟의 노말 오프셋에 더한 값을 현재 프레임의 타겟의 노말 방향으로 적용하게 된다. 그림 8은 노말 메쉬(상세 메쉬) 리타겟팅을 설명한다.

노말 오프셋은 다중해상도적으로 1 단계에서 i 단계까지 타겟 특징 메쉬에 더하여지며 결과적으로 세부적인 표현이 드러나게 된다. 노말 오프셋에 의한 세부 모션의 리타겟팅은 초기 프레임과 현재 프레임간의 노말 오프셋의 편차를 구하는 데서 시작한다. 1단계 초기 프레임에서의 소스의 노말 오프셋 $SN_0[1]$ 과 타겟의 노말 오프셋 $TN_0[1]$ 은 다음과 같이 정의된다.

$$\begin{aligned} SN_0[1] &= S_0[1] - sub(S_0[0]) \\ TN_0[1] &= T_0[1] - sub(T_0[0]) \end{aligned} \quad (2)$$

여기서 SN, TN 는 소스와 타겟의 노말 오프셋이고 그 값은 스칼라이다. $S_0[1], S_0[0], T_0[1], T_0[0]$ 는 각각 소스와 타겟의 1단계, 0단계 메쉬들이다. $sub()$ 는 "divide and smoothing" 과정을 거친 메쉬이다. 특별히 0단계 메쉬는 특징 메쉬를 나타낸다. 따라서 k 번째 프레임, i 단계에서의 소스의 노말 오프셋은 다음과 같이 정의된다.

$$SN_k[i] = S_k[i] - sub(S_k[i-1]) \quad (3)$$

본 논문에서는 노말 방향으로의 노말 오프셋의 변화가 세부 모션이라 가정한다. k 번째 프레임, i 단계에서의 소스의 노말 오프셋의 편차는 다음과 같이 정의된다.

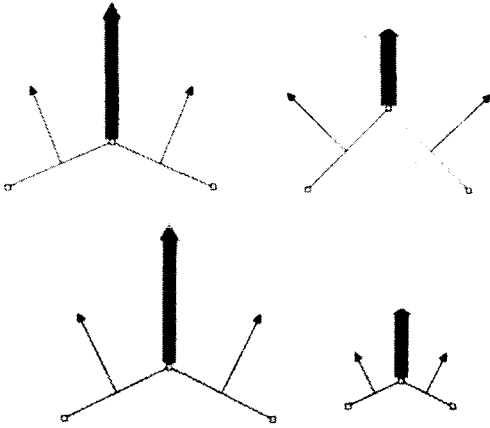


그림 9: 꼭지점의 주변 면적과 꼭지점의 곡률에 따른 노말벡터 계산. 위 그림은 꼭지점의 주변 면적은 같으나 곡률이 다른 경우에 노말벡터를 비교한 것이고 아래 그림은 꼭지점의 곡률은 같으나 주변 면적이 다른 경우에 노말벡터를 비교한 것이다

$$\Delta D_k[i] = SN_k[i] - SN_0[i] \quad (4)$$

이 편차는 매 단계마다 구해지며 타겟에 적용된다. 현재 프레임 k에서의 타겟 상세 메쉬들은 이 편차들을 타겟에 더함으로써 다음과 같이 구해진다.

$$\begin{aligned} T_k[1] &= sub(T_k[0]) + (TN_0[1] + \delta \Delta D_k[1]) \\ T_k[2] &= sub(T_k[1]) + (TN_0[2] + \delta \Delta D_k[2]) \\ T_k[3] &= sub(T_k[2]) + (TN_0[3] + \delta \Delta D_k[3]) \\ &\dots \end{aligned} \quad (5)$$

δ 는 스케일 상수로써 소스와 그와 대응하는 타겟의 임의의 j번째 점(vertex)에서 노말벡터를 각각 \vec{sn}_j, \vec{tn}_j 라 할때 다음과 같이 구해진다.

$$\delta = \sqrt{|\vec{tn}_j| / |\vec{sn}_j|} \quad (6)$$

\vec{sn}_j, \vec{tn}_j 은 꼭지점(vertex)을 둘러싸는 면(face)들의 외적(cross product) 즉 면의 노말벡터들을 평균하여 구한다. 이때 면의 노말벡터들을 정규화(normalize)하지 않고 평균한다. 이렇게 함으로써 꼭지점의 노말벡터의 크기 $|\vec{sn}_j|, |\vec{tn}_j|$ 는 꼭지점 주변의 면적과 꼭지점의 곡률을 동시에 나타낼 수 있다. 본 논문에서는 노말 방향으로의 모션의 크기를 비교하기 위해서 두 가지 직관적 가정을 하였는데 첫째로 주변 면적이 넓을수록 꼭지점은 더 멀리 이동하고 둘째로 꼭지점을 포함한 부위가 평평할수록 노말 방향으로 이동할 수 있는 꼭지점의 잠재력이 더욱 크다는 것이다. 따라서 노말벡터의 크기는 그림 9에서와 같이 꼭지점에서의 곡률에 반비례(상단 그림)하고 주변 면적에 비례(하단 그림)한다. δ 는 소스와 타겟의 꼭지점의 노말벡터 크기의 비로써 결정되고 결과적으로 타겟의 꼭지점의 이동 거리를 결정한다.

그림 10 은 점진적으로 리타겟팅된 타겟의 상세 메쉬를 보여주고 있다.

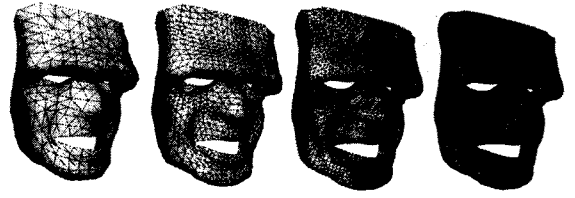


그림 10: 다중해상도적 리타겟팅 결과. 오른쪽으로 갈수록 더 세부적인 표현을 가지는 타겟 상세메쉬들을 나타낸다.

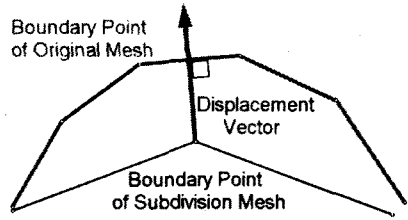


그림 11: 경계점에 대한 처리. 노말 방향으로의 교점이 아닌 오리지널 메쉬의 경계선과 수직으로 만나는 교점으로의 변위벡터를 적용한다.

4.3 경계점의 처리

본 논문에서는 재메쉬화 과정에서 눈이나 입의 경계 를 이루는 점(boundary point)들에 대해서는 노말 방향으로의 교점이 아닌 오리지널 메쉬의 경계선(boundary edge)에 위치시킨다. 따라서 이 경계점에 대해서는 노말 오프셋이 아니라 변위벡터(displacement vector)를 저장한다. 그림 11 에서와 같이 노말 메쉬의 경계점을 지나는 직선과 오리지널 메쉬의 경계선이 직각으로 만나는 교점으로 노말 메쉬의 경계점을 위치시킨다. 이때의 차이를 벡터로 저장한다. 경계점은 리타겟팅하지 않고 초기 프레임에서 구한 타겟의 변위벡터를 그대로 매 프레임의 경계점에 적용을 한다.

제 5 절 오리지널 타겟 메쉬의 변형

지금까지의 소스 노말 메쉬의 모션이 타겟의 노말 메쉬로 어떻게 변환되는가에 대해 서술했다. 즉 모션 리타겟팅은 오리지널 소스 메쉬와 오리지널 타겟메쉬간에 일어난 것이 아니라 이들을 재메쉬화한 소스노말메쉬와 타겟노말메쉬간에 일어난 것이다. 여기서 타겟 노말메쉬가 오리지널 메쉬의 특징을 제대로 표현하려면 높은 단계의 노말 메쉬가 요구되어지고 그것의 용량은 상당히 커지게 된다. 따라서 세부적인 리타겟팅이 어느 정도 되었다고 판단되는 단계에서 타겟 노말 메쉬의 모션을 오리지널 타겟메쉬에 적용할 필요가 있다. 이를 위해, 우선 초기 프레임의 오리지널 타겟메쉬상의 모든 꼭지점을 초기 프레임의 노말메쉬상의 면들위에 투영시킨다. 여기서 노말메쉬와 오리지널 메쉬와의 차이를 메쉬 텍스처라 정의하자. 재메쉬화 과정에서 손실된 오리지널 메쉬의 정보를 메쉬텍스처의 형태로 보관하는 것은 의미가 있다.

그림 12는 초기 프레임에서의 메쉬텍스처를 구하는 방법을 설명한다. $P1_0, P2_0, P3_0$ 는 노말메쉬 상의 면을 이루는

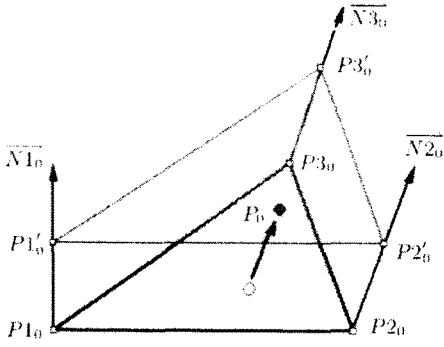


그림 12: 초기 프레임에서의 메쉬텍처 : P_0 는 오리지널 메쉬 위의 꼭지점이고 $P_{1_0}, P_{2_0}, P_{3_0}$ 는 노말메쉬상의 꼭지점이다. $N_{1_0}, N_{2_0}, N_{3_0}$ 는 $P_{1_0}, P_{2_0}, P_{3_0}$ 에서의 노말벡터이다. 꼭지점 P_0 는 $P_{1_0}, P_{2_0}, P_{3_0}$ 와 $N_{1_0}, N_{2_0}, N_{3_0}$ 에 의해 나타내어진다

세 꼭지점들이고 P_0 는 오리지널 메쉬의 꼭지점이다. 그리고 $N_{1_0}, N_{2_0}, N_{3_0}$ 은 각각 $P_{1_0}, P_{2_0}, P_{3_0}$ 의 노말벡터이다. 우선 $P_{1_0}, P_{2_0}, P_{3_0}$ 이 이루는 면과 평행하면서 꼭지점 P_0 를 지나는 평면을 구하고 이 평면과 $N_{1_0}, N_{2_0}, N_{3_0}$ 의 교점을 $P'_{1_0}, P'_{2_0}, P'_{3_0}$ 라 하면 꼭지점 P_0 는 다음과 같이 barycentric coordinates로 나타내어 질 수 있다.

$$P_0 = \alpha P'_{1_0} + \beta P'_{2_0} + \gamma P'_{3_0} \quad (7)$$

α, β, γ : barycentric coordinates

그리고 식 7은 $\vec{N}_{1_0}, \vec{N}_{2_0}, \vec{N}_{3_0}$ 를 이용하여 다음과 같이 표현되어 질 수 있다.

$$P_0 = \alpha(P_{1_0} + i\vec{N}_{1_0}) + \beta(P_{2_0} + j\vec{N}_{2_0}) + \gamma(P_{3_0} + k\vec{N}_{3_0}) \quad (8)$$

$(i = |\vec{P}_{1_0}P'_{1_0}|/|\vec{N}_{1_0}|, j = |\vec{P}_{2_0}P'_{2_0}|/|\vec{N}_{2_0}|, k = |\vec{P}_{3_0}P'_{3_0}|/|\vec{N}_{3_0}|)$

초기 프레임에서 구한 (α, β, γ) 와 (i, j, k) 는 메쉬텍처를 구성한다. 이것을 현재 프레임 k 에서의 $P_{1_k}, P_{2_k}, P_{3_k}$ 와 $\vec{N}_{1_k}, \vec{N}_{2_k}, \vec{N}_{3_k}$ 에 적용하면 현재 프레임에서의 P_k 를 얻을 수 있다.

$$P_k = \alpha(P_{1_k} + i\vec{N}_{1_k}) + \beta(P_{2_k} + j\vec{N}_{2_k}) + \gamma(P_{3_k} + k\vec{N}_{3_k}) \quad (9)$$

메쉬텍처를 적용하여 구한 타겟 오리지널 메쉬의 모션을 그림 13에서 보여주고 있다.

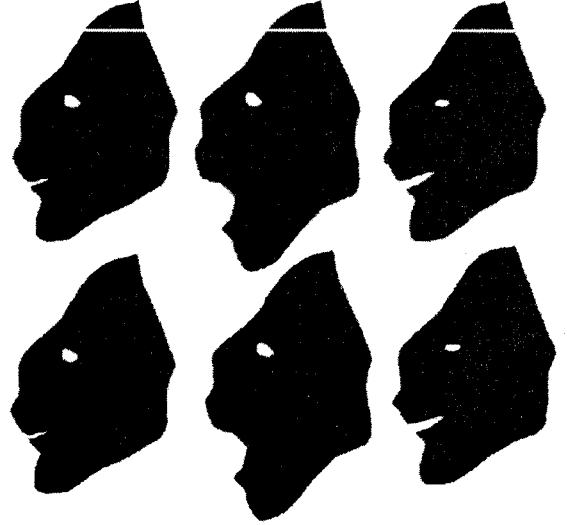


그림 13: 상세메쉬 리타겟팅에 의해 변형된 각 프레임의 타겟 노말메쉬(위 그림)에 초기 프레임에서의 메쉬텍스처를 더한 메쉬(아랫 그림)

제 6 절 실험 및 결과

6.1 Piercing error

재메쉬화 과정에서 노말 메쉬를 생성할 때 self-intersection 같은 부적절한 경우가 있다. 이는 이웃하는 점들의 노말벡터가 서로 엇갈리는 경우에 자주 발생한다. 이러한 경우를 piercing error라 하고 서브디비전 단계에서 butterfly scheme과 loop scheme 그리고 "divide and smoothing"을 적용하였을 때 각각의 예러율은 표 1과 같다.

butterfly scheme			
name	vertex	piercing error vertex	%
gorilla	17,470	53	0.3033 %
ox	17,470	27	0.1545 %
frankenstein	17,470	15	0.0858 %
loop scheme			
name	vertex	piercing error vertex	%
gorilla	17,470	7	0.0401 %
ox	17,470	6	0.0343 %
frankenstein	17,470	4	0.0228 %
smoothing			
name	vertex	piercing error vertex	%
gorilla	17,470	14	0.0801 %
ox	17,470	8	0.0458 %
frankenstein	17,470	8	0.0458 %

표 1: The piercing error

Piercing error의 경우에는 노말 오프셋을 이용하지 않고 이웃하는 점들의 평균 좌표값으로 대신하게 된다.

6.2 결과

본 논문에서는 캡처된 사람의 얼굴 모션을 사람의 얼굴 구조와 차이가 많은 고릴라, 소, 프랑케스타인에 리타겟팅하여 보았다. 그림 14 는 그 결과를 보여준다. 상세 메쉬 리타겟팅은 제 4단계 노말 메쉬까지 적용하였다. 특징 메쉬 리타겟팅 과정에서 모션 벡터 방식을 적용하였으나 모션의 변위가 큰 경우 무자연스런 결과를 내므로 수작업으로 이를 수정해 수었다.

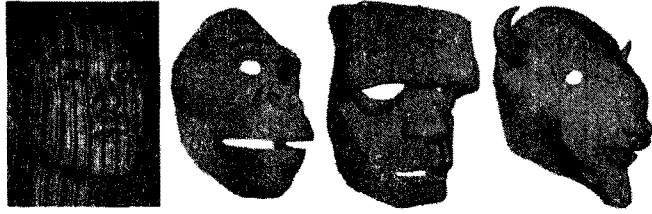
제 7 절 결론

본 논문에서는 고밀도 얼굴 모션 데이터를 형태가 사람과 다른 얼굴에 리타겟팅하는 효과적인 방법을 제시하였다. 기본적인 아이디어는 모션을 기본적인 모션과 세부적 모션들의 합으로 나타내는 것으로부터 출발하였다. 이는 오리지널 메쉬를 특징 메쉬와 각 단계의 노말 메쉬로 재구성함으로써 가능하게 되었다. 특징 메쉬 리타겟팅에 대해서는 기존의 모션 벡터 방식을 사용하였고 노말 메쉬 리타겟팅에 대해서는 노말 오프셋을 이용한 방식을 사용하였다. 특징 메쉬의 기본적인 모션 위에 세부적인 모션이 단계적으로 리타겟팅됨으로써 보다 안정적인 고 정교한 결과를 내었다. 실험적으로 기본적인 모션의 리타겟팅 결과가 좋으면 세부적인 모션의 리타겟팅도 그에 따라 좋은 결과를 내게 된다는 것을 알게 되었다. 또한 기본 모션을 사용자가 제어함으로써 얼굴 애니메이션의 효율적인 수정 또는 편집을 가능하게 하였다. 따라서 모션을 기본 모션과 세부 모션으로 분해하여 리타겟팅한 시도는 그 의미가 있다고 할 수 있겠다. 본 연구에 보완되어야 할 부분은 다음과 같다. 특징 메쉬 리타겟팅 과정에서 적용한 모션 벡터 방식은 형태가 매우 다른 타겟 모델의 경우 매우 불안정하다. 때문에 사용자의 직관을 이용하는 사례 기반 기법(example based method)을 적용하여 보다 안정적이고 직관적인 결과를 낼 필요가 있다. 또한 주름 같은 미세한 특징이 소스에서 타겟에 전달될 때 보다 저용량으로 보다 명확히 전달되려면 특징선을 고려한 재메쉬화 기법이 필요하다.

참고 문헌

- [1] Kwangin Choi and Hyeongeok Ko. On-line motion retargeting. In *Journal of Visualization and Computer Animation*, pages 223-243, 2000.
- [2] M.S. Floater. Parameterization and smooth approximation of surface triangulations. In *Computer Aided Geometric Design* 14, pages 231-250, 1997.
- [3] Michael Gleicher. Retargeting motion to new characters. In *SIGGRAPH 98*, pages 33-42, 1998.
- [4] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *SIGGRAPH 98*, 1998.
- [5] I. Guskov, K. Vidimce, W. Sweldens, and P. Schroder. Normal meshes. In *SIGGRAPH 00*, 2000.
- [6] I. Guskov, W. Sweldens, and P. Schroder. Multi-resolution signal processing for meshes. In *SIGGRAPH 99*, 1999.
- [7] H. Hoppe. Progressive meshes. In *SIGGRAPH 96*, pages 99-108, 1996.
- [8] Chang-Soo Je. Color-encoded fringe projection and processing for acquisition of real time 3d range data. Master's thesis, Graduate School of Media Communications, Sogang University, 2002.
- [9] P. Karla, A. Mangili, N.M. Thalmann, and D. Thalmann. Simulation of facial muscle actions based on rational free form deformations. In *Eurographics 92*, pages 69-69, 1992.
- [10] R. Klette, K. Schluns, and A. Koschan. *Computer Vision: Three dimensional Data from Images*. Springer, 1998.
- [11] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH 98*, pages 106-114, 1998.
- [12] Jeehe Lee and SungYong Shin. A hierarchical approach to interactive motion editing for human-like figures. In *SIGGRAPH 99*, pages 39-48, 1999.
- [13] Y. Lee and D. Terzopoulos. Realistic face modeling for animation. In *SIGGRAPH 96*, pages 62-66, 1996.
- [14] C. Loop. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah, Department of Mathematics, 1987.
- [15] A. M. McIvor and R. J. Valkenburg. Calibrating a structured light system. In *Image and Vision Computing New Zealand, (Lincoln, Canterbury)*, pages 167-172. Industrial Research Limited, Aug 1996.
- [16] J.Y. Noh and Ulrich Neumann. Expression cloning. In *SIGGRAPH01*, pages 271-288, 2001.
- [17] F. I. Parke. Parameterized models for facial animation. In *IEEE Computer Graphics and Application*, volume 2(9), pages 61-68, 1982.
- [18] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D.H.Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH 98*, pages 76-84, 1998.
- [19] S. Platt and N. Badler. Animating facial expression. In *Computer Graphics*, volume 16(3), pages 246-262, 1981.
- [20] Hyewon Pyun and Sung Young Shin. An example-based approach for facial expression cloning. In *Eurographics 03*, pages 167-176, 2003.
- [21] D. Terzopoulos and K. Waters. Physically-based facial modeling, anaalysis, and animation. In *SIGGRAPH 87*, pages 17-24, 1987.
- [22] D. Zorin. Interpolating subdivision for meshes with arbitrary topology. In *SIGGRAPH 96*, pages 189-192, 1996.

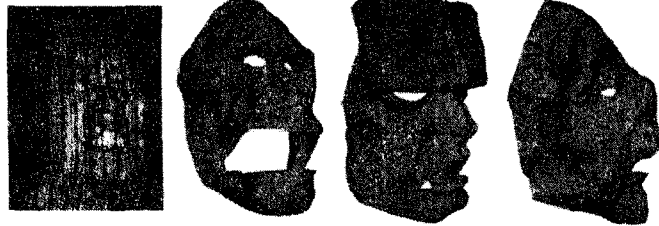
init frame



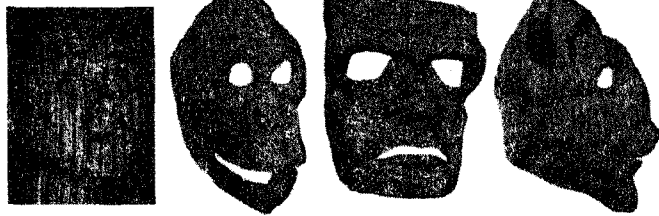
frame 1



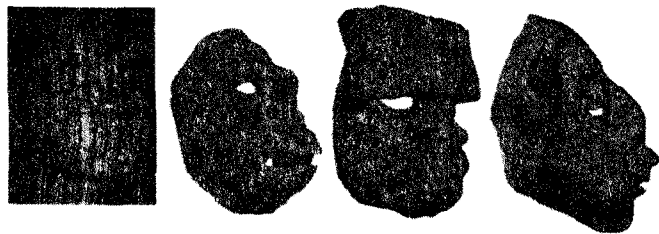
frame 2



frame 3



frame 4



frame 5

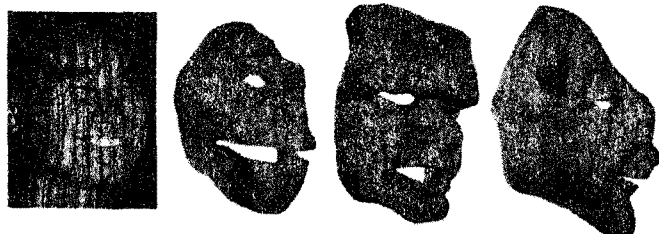


그림 14: 비전 기법을 이용하여 얻은 고밀도 모션 캡처 데이터를 gorila .frankenstein .ox 에 각각 리타겟팅한 결과