

관절 자세 군집화(JPC)를 활용한 모션 단순화 기법

안 정현, 원 광연
한국과학기술원 전자전산학과 가상현실 연구실
(chocchoggi, wohn)@vr.kaist.ac.kr

Motion Simplification using Joint Posture Clustering (JPC)

Junghyun Ahn, Kwangyun Wohn
KAIST EECS VR lab.

요 약

캐릭터 애니메이션 기술의 발달로 가상공간에 애니메이트되는 캐릭터의 수가 점점 증가되고 있으며, 캐릭터 자체 골격구조의 관절 개수와 캐릭터를 덮고 있는 메쉬의 폴리곤 개수도 점점 증가하는 추세이다. 따라서, 실시간 가상환경에서 다수의 캐릭터를 전처리 과정 없이 시뮬레이션할 경우 전체 군중시스템 성능의 저하가 예상된다. 본 논문에서는, 이러한 문제점을 해결하기 위해 모션 다단계(motion level-of-detail) 기법을 제시한다. 모션 단순화 기법은 캐릭터의 움직임에 제어하는 골격(관절)구조와 캐릭터의 형태를 시각적으로 표현하는 기하(메쉬)구조를 단순화 하는 방법으로 기존 동작과 단순화된 동작의 차이를 최소화 한다. 골격구조 단순화를 위한 JPC(joint posture clustering)방법은 특정 관절의 연속된 모션 시퀀스에서의 유사 자세 집단을 추출하여 하나의 자세로 표현하는 방법으로, 모션의 특성에 따라 동적으로 관절을 단순화하여 관절 시뮬레이션 시간을 줄이는 방법이다. JPC방법은 골격구조가 시간에 따라 동적으로 변형되기 때문에 골격구조의 계층구조를 재 구축할 시간이 필요하지만, 기존 동작과 유사성을 잃지 않는 단순화된 동작 생성이 가능하다. 유사 자세 집단을 추출하기 위해 전체 모션 시퀀스에서 관절의 프레임간 자세 차이를 수식화하여 테이블 형태로 구성하고 이를 통해 기존 동작의 유사성을 잃지 않으며 관절의 단순화율을 최대화 할 수 있는 알고리즘을 제시한다. 또한, 실시간 군중 환경의 성능을 더욱 향상시키기 위해 시간에 따라 변형되는 캐릭터 메쉬의 단순화 기법을 적용한다. 실험결과 모션 다단계 기법은 실시간 군중환경에서 캐릭터의 수가 많고 복잡한 골격구조와 기하구조로 구성된 관절 제척의 변화가 심하지 않은 동작에 대해 특히 효율적이다.

1. 서론

영화, 게임, 군중 시뮬레이션 기술발전으로 가상환경 내부에 움직이는 캐릭터 수, 캐릭터 자체의 관절 수 그리고 캐릭터 표면을 나타내는 메쉬의 복잡도가 증가하고 있다. 이러한 추세에 따라, 다수 캐릭터를 움직이기 위해 필요한 계산량을 단순화 할 수 있는 방법에 대한 연구의 필요성이 대두되고 있다. 영화, 애니메이션과 같은 군중환경에서는 캐릭터를 움직이는 실시간 계산이 필요하지 않기 때문에 계산량을 줄이는 것이 필수적이지 않지만, 이미지 공간을 적게 차지하는 캐릭터의 움직임을 계산하는 자원을 낭비할 필요는 없다. 또한, 게임, 네트워크 가상환경과 같은 실시간 군중환경의 경우 영상의 질과 계산량을 조절하는 메쉬 다단계 기법(geometric level-of-detail)과 같은 맥락으로 애니메이션의 질과 계산량을 조절하는 모션 다단계 기법(motion level-of-detail)에 대한 연구가 필요하다.

따라서, 본 논문에서는, 실시간 군중환경의 성능향상을 위한 전처리 방법인 모션 다단계 기법에 대해 기술한다. 캐릭터 애니메이션이 가장 많이 활용되는 영화나 게임분야에서도 물리적인 시뮬레이션 보다는 모션캡처를 선호하기 때문에 실시간 환경의 모션 데이터베이스는 모션캡처 데이터들로 구성된다.

일반적인 실시간 군중환경에서의 캐릭터 움직임을 제공하는 방법은 모션 데이터베이스와 캐릭터 데이터베이스에서 선택된 동작을 모델에 매핑하여 가상환경으로 변환시킨다. 하지만, 군중환경 내부에 캐릭터가 다수 존재할 경우 관절 시뮬레이션과 캐릭터 렌더링에 소요되는 비용이 크게 작용한다. 이에 대한 비용을 줄이기 위해 그림 1과 같이 시뮬레이션 성능향상을 위한 골격구조 단순화와 렌더링 성능향상을 위한 기하구조 단순화를 전처리 과정에 포함하여 군중환경의 성능향상을 꾀할 수 있다.

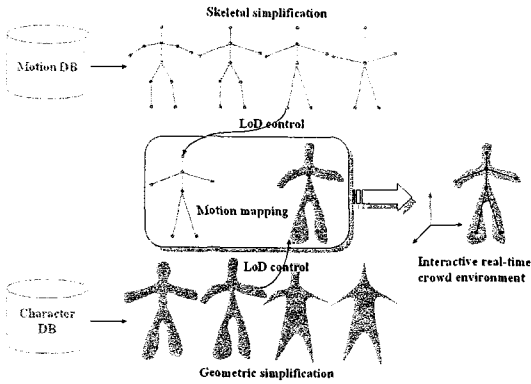


그림 1. 모션 다단계 기법을 위한 실시간 균중환경

골격구조 단순화: 모든 모션데이터는 관절체의 계층구조를 포함한다. 이러한 골격구조는 시간에 따라 변하는 트리 형태로 정의되고 단순화를 통해 관절 시뮬레이션 비용을 줄일 수 있다. 또한, 캐릭터 스키닝 과정에서 발생하는 점 변환 비용도 줄일 수 있다. 골격구조 단순화의 기존사례는 물리적인 시뮬레이션의 성능향상을 위한 분야에서 사용되었다[1][2]. 그러나, 기존 연구들은 단순화를 위한 수작업이 필요하고 균중환경에 적합하지 않다. 골격구조 단순화의 기존사례는 다음 장에 세부적으로 기술한다.

기하구조 단순화: 렌더링 성능 향상을 위한 기하구조 단순화에 대한 기존연구들은 주로 변형되지 않는 메쉬구조에 대해 적용된다[3][4][5]. 그러나, 캐릭터의 기하구조는 시간에 따라 변형되기 때문에 이를 그대로 적용할 수 없다. 따라서, 캐릭터의 메쉬를 각 관절 세그먼트 별로 분리한다. 균중 애니메이션 분야에서 이러한 기하구조를 단순화하기 위해 영상기반 방법을 통해 단순화한 사례가 있다[6][7]. 그러나, 영상기반 방법은 캐릭터 움직임의 사실성이 떨어지고 가상공간 내부의 카메라 이동에 제약이 있다.

본 논문에서는 위에 기술한 골격구조 단순화와 기하구조 단순화 기존연구의 문제점을 고려한 전처리 기법을 제안한다. 골격구조 단순화에 수작업을 요하지 않고 임의의 관절 구조와 모션에 적용할 수 있는 단순화 기법을 고려하기 위해, 골격구조 자체를 단순화 하기 보다 모션에 종속적인 단순화 방법이 필요하다. 모션 다단계 기법의 골격구조 단순화를 위한 JPC(joint posture clustering)방법은 관절 궤적을 분석하여 모션 시퀀스의 연속적인 유사 자세들을 하나의 자세로 대체되도록 그룹화 한다. 관절의 단순화율을 최대화 하면서 기존동작과 차이를 최소화하는 최적의 자세 집단을 추출하기 위해 각 관절의 자세들 간의

차이를 수식화 하였다. 각 관절의 시간에 따른 자세 간 차이를 테이블 형태로 구성하여 최적의 유사 자세 집단을 임계치에 따라 추출하였다. 유사 자세 집단 내부에는 이를 대표하는 대표자세를 추출하고 애니메이션 과정에서 대표 자세들을 순차적으로 변환한다. JPC방법은 모션 시퀀스에 따라 골격구조가 동적으로 변하기 때문에 골격구조 자체를 단순화하는 방법에 비해 성능이 떨어지는 경우가 발생하지만, 임의의 관절과 모션에 자동으로 적용할 수 있고 기존동작과의 유사성을 잃지 않는 단순화된 동작을 생성할 수 있다.

기하구조 단순화 측면에서는 기하구조를 보다 사실적으로 단순화 하기 위해 영상기반 방법보다는 progressive mesh방법[3]을 캐릭터의 부위별로 적용하여 스키닝 과정에서 발생하는 기하구조의 변형 오차를 최소화 하였다.

2. 관련 연구

‘균중 애니메이션’ 분야는 가상환경에 존재하는 다수의 캐릭터 움직임을 보다 사실적으로, 보다 효율적으로 그리고 보다 쉽게 제공하기 위해 다음 3가지 관점을 대상으로 연구되는 캐릭터 애니메이션 기술로 정의할 수 있다.

2.1 장면의 사실성 (Scene reality)

가상공간의 다수 캐릭터를 실제 균중이 움직이는 것처럼 표현하기 위해서는 균중을 제어할 수 있는 균중 시스템 모델링[8][9]이 필요하다. 또한 사실적인 균중의 유동을 표현하기 위해 균중의 전체적인 유동을 분석해야 한다. 이러한 균중의 유동을 분석하기 위한 기존 연구로 긴급대피 상황에서 균중의 유동을 분석하였다[10][11]. 또한, 균중의 전체 행위를 사실적으로 표현하기 위해 캐릭터 개개의 행위규칙을 정의해야 한다. 이러한 조건을 만족하기 위해 어항 속의 물고기들의 행위규칙을 모델링하여 가상환경의 변화에 따라 행위양식을 상황에 맞게 제공하였다[12]. 동물이 아닌 인체의 행위규칙에 관한 연구도 그와 비슷하게 모델링 되었다[13][14]. 그 외에 장면의 사실성을 위해 균중환경 내부에 존재하는 캐릭터 간 충돌회피 문제나 움직임의 변환[15][16]에 대한 연구도 필수적이다.

2.2 상호작용성 (Interactivity)

균중 개개의 행위규칙만으로는 애니메이터가 원하는 결과물을 생성하기 힘들다. 따라서, 앞서 기술한 균중

시스템 모델에 개개의 혹은 그룹의 움직임을 사용자가 직접 개입할 수 있는 방법이 필요하다[17][18].

2.3 성능향상 (Performance)

균중 애니메이션의 가장 중요한 요소 중 하나는 다수의 캐릭터를 시물레이션 하는 과정에서 발생하는 시스템 성능을 향상시키는 과제이다. 다수 캐릭터를 움직이는 균중 애니메이션에서 성능향상을 위한 연구는 크게 관절 시물레이션의 성능향상을 위한 골격구조 단순화와 렌더링 성능향상을 위한 기하구조 단순화 방법이다.

모션 블렌딩을 물리적으로 시물레이션 하는 과정의 성능을 향상시키고 최적화 모델을 보다 쉽게 수립 시키기 위해 캐릭터의 골격구조를 단순화한 사례가 있다[1]. 다수의 관절을 무게중심점 하나로 대체하거나, 좌우 대칭적인 자세를 반으로 나누고 또는 팔다리의 꺾임이 적은 곳에서 하나의 막대형태로 대체하여 단순화시켰다. 그러나, 이와 같은 방법으로 단순화 할 경우 모션에 따라 수작업이 많이 필요하고 적용되지 않는 모션들이 많이 존재한다.

또 다른 연구에서는 물리적 시물레이션의 계산량을 조절하기 위해 관절체의 시물레이션 레벨을 물리, 운동학, 무게중심의 3레벨 모델로 나누어 계산하였다[2]. 이는 시물레이션의 질과 계산량을 조절하는 가능성만을 제시하였을 뿐, 임의의 관절체를 대상으로 일반화하기 힘들다.

기하구조 단순화를 활용한 관련연구는 영상 기반 방법을 적용하여 렌더링 성능을 향상시켰다[6][7]. 인체의 움직임 시퀀스를 캐릭터 중심으로 카메라 위치에 따라 이미지 시퀀스로 대체하여 시스템의 성능향상을 꾀하였다. 이는 렌더링 성능향상을 기대할 수 있으나 움직임이 사실적이지 않고 카메라 움직임에 제한이 있다.

캐릭터 모션자체의 다단계 동작을 생성하는 연구는 [20]에서 찾아볼 수 있다. Coordinate-invariance를 보장하는 방법으로 다단계 동작을 생성할 수 있으며, 키 프레임 애니메이션이나 모션 편집 기법에 유용하게 활용할 수 있는 방법이다.

본 논문은 실시간 균중환경의 성능향상을 위해 가상 캐릭터의 골격구조와 기하구조를 위에 제시된 문제점을 토대로 해결한다.

3. 균중환경에서의 관절 시물레이션

가상환경에서 기하구조의 복잡도에 의한 시스템 성능 저하는 잘 알려져 있고 많은 연구 성과가 이루어졌다.

하지만, 캐릭터 애니메이션에서 관절 움직임의 영향으로 발생하는 성능에 대한 연구는 찾아보기 힘들다. 본 장에서는 다수 캐릭터에 모션을 입히는 과정에서 발생하는 성능저하 요인을 분석한다.

그림 2는 앞서 기술한 실시간 균중환경의 캐릭터 애니메이션 과정이다. 전처리(Preprocessing) 과정에는 모션과 캐릭터 내부의 골격구조를 공간 상에 일치시키고 두 구조 사이의 매핑되는 관절 간의 변위를 계산한다. 장면생성과정(Scene generation)에는 모션 데이터베이스로부터 선택된 동작의 시간에 일치하는 자세로 변환(posture transform)하고, 자세변환 결과와 앞서 전처리 과정에서 매핑된 모션, 캐릭터 두 구조 간의 변위를 이용하여 스킨닝(skinning) 과정을 거친다. 스킨닝을 통해 변형된 캐릭터 메쉬는 가상공간으로 변환(world transform)된다. 캐릭터 메쉬는 최종적으로 렌더링(rendering)과정을 거쳐 애니메이션 프레임을 완성한다.

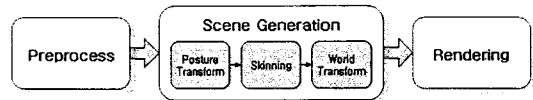


그림 2. 관절 시물레이션 과정

이러한 애니메이션 과정에서 일반적으로 알려진 렌더링 과정에서의 성능저하 이외에 장면생성 과정에서도 관절 시물레이션에 의한 성능저하가 발생하는데, 943개의 다각형과 50개의 관절을 가진 1000개의 캐릭터를 실험하여 성능을 분석하였다. 실험결과는 그림 3과 같이 하나의 애니메이션 프레임을 완성하기 위해 장면을 생성하는 시간이 렌더링 과정의 성능저하만큼 큰 비중을 차지하였다. 따라서, 전처리 과정에 모션 골격구조와 캐릭터 기하구조를 단순화 할 경우 전체 애니메이션 시스템의 성능향상이 예상된다.

렌더링 (58%)	자세변환 (23%)	스킨닝 (11%)	기타 (8%)
장면생성 (42%)			

그림 3. 캐릭터 애니메이션 장면생성과 렌더링 시간 비교

4. 골격구조 단순화 기법 (JPC)

모션의 관절에 작용하는 회전 값 변화는 움직임이 미미한 경우에도 매시간 새로운 자세를 계산하여 캐릭터 관절에 입히는 작업을 반복한다. 실제로 모션의 각 관절

변화를 살펴볼 때 특정 시간 범위 내에서 관절 변환이 필요 없는 경우가 발생한다.

골격구조 단순화 기법인 JPC방법은 각 관절 전체 모션의 회전 값 변화를 분석하여 기존 동작과 오차가 적은 집단을 추출한다. 결과적으로 모든 관절이 n 개의 프레임에서 $m < n$ 인 m 개의 집단으로 축소되어 불필요한 변환은 제거한다. 단순화된 모션의 오차를 최소화 하기 위해 관절의 노드들과 관절 사이의 링크들로 구성된 기존 모션 $M(t)$ 와 단순화된 모션 $M'(t)$ 는 서로 $M(t) \supset M'(t)$ 의 관계를 갖는다. 따라서, 관절이 단순화 된 경우에도 단순화된 골격구조는 기존 모션의 골격구조의 부분 집합이 된다. JPC에 의해 단순화된 골격구조는 단순화된 집단의 키 자세를 유지하면서 자신의 정보를 부모관절에 전달한다.

4.1 자세 간 오차 정의

본 논문에서는 최적의 자세 집단을 추출하기에 앞서 각 관절의 시간에 따른 자세 차이를 수식화 한다. 관절 단순화의 의미는 특정 프레임 자세가 그 프레임을 이루는 집단의 대표 자세로 대체되기 때문에 대표 자세와 기존 자세 간의 오차 정의가 필요하다. 관절 j 의 자세 간 오차는 거리오차(E_{pos})와 회전오차(E_{ori})로 분류되며, 수식 (1)과 같이 정의된다.

$$E_j(t, t_{ref}) = E_{pos,j}(t, t_{ref}) + \alpha E_{ori,j}(t, t_{ref}) \quad (1)$$

시간 t 는 모션시퀀스의 현재시간을 나타내고 t_{ref} 는 t 와 비교되는 같은 모션시퀀스내의 다른 시간을 나타낸다. 상수 α 는 거리오차와 회전오차 사이의 관계를 나타내는 가중치로 관절의 자유도 중에 자식관절 방향 축의 회전존재 여부에 따라 변한다. 자세 간 오차는 그림 4에 도시된 값들을 토대로 정의된다.

관절의 궤적: 관절 j 에서 자식관절로 향하는 벡터를 v_j 라 할 때 관절 j 가 모션의 회전정보인 $q_j(t)$ 로 인해 발생하는 관절 궤적은 구상에 맞힌다. 이러한 궤적 좌표들은 식 (2)와 같이 표현된다. 궤적의 좌표를 단위 구 상에 나타내기 위해 관절 j 와 자식관절 c 간의 거리 $l_{j,c}(t)$ 를 활용한다.

$$v_j'(t) = \frac{q_j(t) \cdot v_j \cdot q_j(t)^{-1}}{\|l_{j,c}(t)\|} \quad (2)$$

두 자세 간 사이 각: 관절 궤적의 좌표 $v_j'(t)$ 를 이용하여 두 자세 간 사이 각은 식 (3)과 같이 간단하게 유도된다.

$$\theta_j(t, t_{ref}) = \arccos\left(\frac{v_j'(t) \cdot v_j'(t_{ref})}{\|l_{j,c}(t)\|^2}\right) \quad (3)$$

하위관절로 인한 가중치: 자식관절을 많이 포함하고 있는 관절의 경우 관절 움직임에 따라 자식관절 움직임도 영향을 받으므로 약간의 움직임에도 오차가 크게 작용한다. 따라서, 각 관절 계층구조 상의 위치에 따라 식 (4)와 같은 가중치 $r_j(t)$ 를 정의한다. $l_{j,c}(t)$ 는 관절 j 와 자식관절들 중의 하나인 c 관절 사이의 거리를 나타낸다.

$$r_j(t) = \sum_c^{leaf} l_{j,c}(t) \quad (4)$$

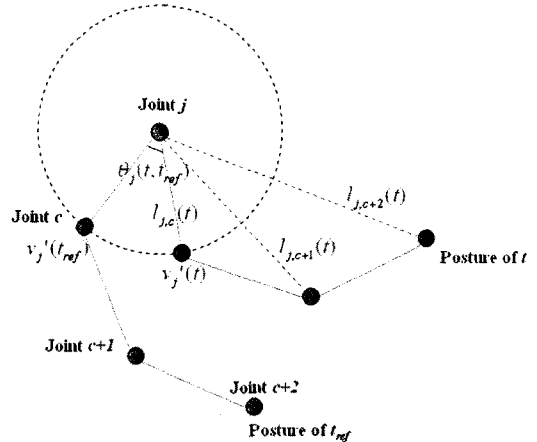


그림 4. 관절 자세 간의 오차 계산을 위한 정의

거리 오차: 위에 정의된 식 (3)과 (4)에 의해 두 자세 간 거리 오차는 식 (5)와 같이 정의된다. 식 (5)의 의미는 관절 j 의 움직임으로 인한 관절 j 와 하위관절들의 이동을 표현한다.

$$E_{pos,j}(t, t_{ref}) = \|r_j(t) \cdot \theta_j(t, t_{ref})\| \quad (5)$$

회전 오차: 관절의 궤적만을 고려하여 정확한 오차를 계산하기 힘들다. 자식관절 방향의 축을 중심으로 회전한 경우 궤적 상에 맞히는 위치는 같게 나타난다. 이러한 문제를 해결하기 위해 관절 j 의 시간 t, t_{ref} 의 회전 값인 $q_j(t)$ 와 $q_j(t_{ref})$ 의 logarithm[16][19]을 통해 회전에 대한 오차를 고려한다.

$$E_{ori,j}(t, t_{ref}) = 2 \|\log(q_j(t)^{-1} \cdot q_j(t_{ref}))\| \quad (6)$$

4.2 자세 집단 영역 추출

기존 동작과 단순화된 동작 간의 오차범위 내에 허용되는 최적 자세 집단을 추출하기 위해, 관절의 프레임(총 프레임: n 개) 간 자세의 차이를 비교할 수 있는 $n \times n$ 테이블로 구성하였다. 테이블의 각 원소는 $E_j(t, t_{ref})$ 이고 $t = t_{ref}$ 인 경우 서로 같은 자세 간 차이 비교이므로 원소 값은 0이다.

기본적으로 좌측상단 좌표 t, t_{ref} 가 같고 크기가 $m \times m$ ($m \leq n$) 인 마스크 내부의 모든 원소 값이 오차허용범위 이하일 경우 그 마스크는 자세 집단의 후보로 등록된다. 그림 5는 관절 자세 간 차이가 15.0 이하인 자세 집단을 특정 모션의 특정영역인 158~166프레임에서 추출하는 과정을 도시한다. 테이블 대각선 원소는 0 이고 그로부터 거의 대칭인 결과를 나타낸다. 대칭되는 원소들 사이에 발생하는 차이는 식 (4)에서 하위 관절에 의한 가중치가 t 에 따라 미미한 차이를 보이기 때문이다. 테이블 원소가 거의 대칭이므로 최적의 집단을 추출하는 알고리즘은 테이블 우측상단의 원소들만 활용하고 좌측하단의 대칭 원소들은 최종적으로 대표 자세를 생성할 때만 활용한다.

Rejected cost: 15.00

frame	158	159	160	161	162	163	164	165	166
158	0.00	16.36	29.36	35.58	32.67	30.24	30.09	30.74	30.98
159	16.34	0.00	13.00	19.28	16.35	14.00	13.82	14.41	14.66
160	29.30	13.00	0.00	6.34	3.38	1.30	1.11	1.47	1.88
161	35.49	19.25	6.34	0.00	2.96	5.18	5.38	4.88	4.63
162	32.62	16.34	3.38	2.96	0.00	2.32	2.50	1.95	1.80
163	30.31	14.05	1.30	5.21	2.33	0.00	0.20	0.51	0.60
164	30.12	13.86	1.11	5.40	2.51	0.20	0.00	0.63	0.80
165	30.68	14.40	1.47	4.89	1.94	0.51	0.62	0.00	0.57
166	30.88	14.63	1.88	4.62	1.79	0.60	0.80	0.57	0.00

Rejected Area
 Clustered Area

그림 5. 자세 집단 추출을 위한 테이블

도시된 테이블의 결과는 9개 프레임의 9개 자세가 {158}, {159, 160}, {160 ~ 166}의 3개의 자세로 단순화된다. 158 프레임에 해당하는 자세 집단 후보는 단일 프레임을 나타내므로 최적 자세 집단 추출 과정에는

배제된다. 자세 집단 후보로 등록된 테이블의 모든 집단 영역은 크기가 $m \times m$ 인 정사각형을 나타낸다. 그러나, 영역 간에는 그림 5의 점선과 같이 교차영역이 생성된다. 최종 집단 영역을 구하기 위해 다음 알고리즘을 제안한다.

- STEP 1: 테이블 각 행의 자세 집단 영역 추출
- LOOP START
- STEP 2: 최대 자세 집단 영역 검색
- STEP 3: 최종 리스트에 최대 자세 집단 영역 추가
- STEP 4: 최대 자세 집단 영역 삭제
- STEP 5: 남은 자세 집단 영역 갱신
- LOOP END
- STEP 6: 각 최종 자세 집합의 대표 자세 생성

STEP 1은 테이블 우측상단 원소들만 활용한다. 각 행의 자세 집단 영역은 대각선 원소로부터 오차범위가 넘어가는 원소가 검색되기 전까지의 영역으로 정의된다. STEP 5는 남은 자세 집단들의 영역을 최종 리스트에 포함된 영역과의 교차 부분을 제거함으로써 갱신된다.

4.3 자세 집단의 대표 자세

최종 자세 집단 리스트가 생성되면 각 자세 집단은 애니메이션 과정에 하나의 자세로 유지되므로 이에 해당하는 대표 자세를 계산한다. 대표 자세는 자세 집단 영역에 포함된 어떤 프레임으로도 대체가 가능하지만, 그림 6에서 보듯이 행의 합이 최소가 되는 프레임이 기존 동작과의 차이를 최소화 할 수 있다. 따라서 자세 집단의 대표 자세는 식 (7)과 같이 표현된다.

$$\min_{t_{ref}} \left(\sum_t^m E_j(t, t_{ref}) \right) \quad (7)$$

frame	160	161	162	163	164	165	166	Error
160	0.00	6.34	3.38	1.30	1.11	1.47	1.88	15.48
161	6.34	0.00	2.96	5.18	5.38	4.88	4.63	29.37
162	3.38	2.96	0.00	2.32	2.50	1.95	1.80	14.90
163	1.30	5.21	2.33	0.00	0.20	0.51	0.60	10.15
164	1.11	5.40	2.51	0.20	0.00	0.63	0.80	10.65
165	1.47	4.89	1.94	0.51	0.62	0.00	0.57	9.99
166	1.88	4.62	1.79	0.60	0.80	0.57	0.00	10.26

Key Posture
 Clustered Area

그림 6. 자세 집단의 대표 자세 추출

그림 6에서 165프레임 행 내부 원소들은 {160 ~ 166} 각각의 프레임이 165프레임 자세를 유지했을 때 발생하는 기존 동작과 자세 간 오차이고, 애니메이션이 수행되는 동안 최소 오차 비용이 적용되므로 자세 집단 최적의 대표 자세가 된다.

그림 7은 JPC방법을 통해 자세 집단이 추출된 이후 궤적 변화를 도시한다. 오차 허용 범위가 증가할수록 자세 집단 영역은 커지며 성능이 향상 되지만, 기존 동작과의 차이는 증가한다.

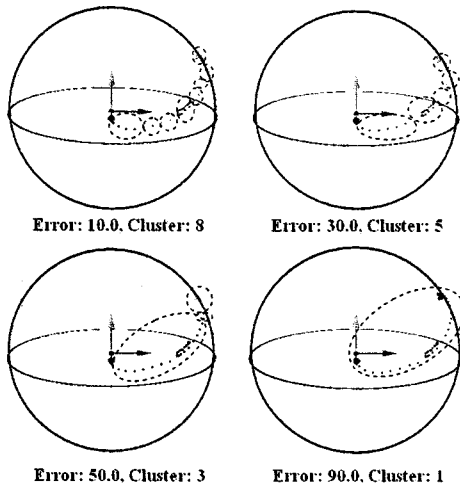


그림 7. JPC방법을 이용한 궤적의 변화

모션 데이터베이스 내부에 존재하는 모션이 긴 경우 테이블 크기로 인해 문제의 복잡도($O(n^2)$)가 증가한다. 이러한 문제를 해결하기 위해 테이블 구성 전, 관절 궤적 속도를 계산하여 오차범위 초과구간을 분리하고 문제의 복잡도를 줄인다. 기본적으로 오차의 정의가 자세 간 각속도를 표현하는 $r\theta$ 형태이기 때문에 $E_{pos, j}(t, t-1)$ 를 이용하여 문제의 크기를 줄일 수 있다.

5. 기하구조 단순화

JPC방법으로 단순화된 움직임은 캐릭터 메쉬와 함께 단순화 될 경우 보다 성능을 향상 시킬 수 있다. 일반적인 정적 메쉬 모델에 대한 단순화 기법을 캐릭터 메쉬에 적용하여 기하구조를 단순화한다. 그러나, 캐릭터의 경우 시간에 따라 변형되는 메쉬구조 이므로 기존 방법을 그대로 활용할 수 없다. 일반적으로 캐릭터 메쉬를

형성하는 점들은 다수의 관절 세그먼트를 공유하는 점($0 < w < 1$)들과 하나의 세그먼트와 연결된 점($w=1$)들로 분류된다. 캐릭터 스키닝 과정에서 이를 점들의 가중치인 w 값으로 표현한다.

본 논문은 기하구조 단순화 과정에서 변형되는 기존 캐릭터 메쉬와 단순화된 메쉬의 오차를 최소화 하기 위해 $w=1$ 인 점들에 대해서만 progressive mesh 방법[3]을 적용하였다. 이와 같이 기하구조 단순화를 활용할 경우 모션에 의해 관절이 움직여도 기존 모델과의 차이를 유지할 수 있다. 그림 8은 기하구조 단순화 결과를 나타낸다. 그림에서 보듯이 60%가 넘게 단순화된 메쉬도 기존 메쉬와 유사한 모습을 갖는다.

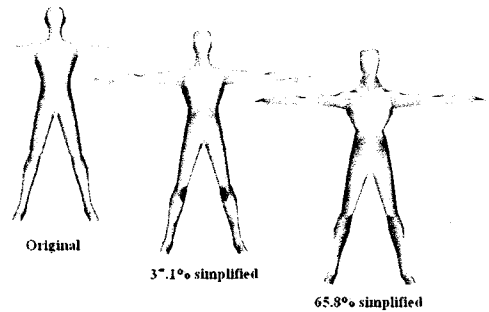


그림 8. 기하구조 단순화 결과

6. 실험 및 결론

6.1 실험결과

본 논문에서는 실시간 균중환경의 성능향상을 위해 캐릭터의 골격구조 단순화 기법을 제시하였고 성능을 보다 향상 시키기 위해 기존 메쉬 단순화 방법을 캐릭터 메쉬 구조에 적합하게 변형하여 적용하였다. 따라서, 본 절에서는 캐릭터 골격구조 단순화 방법인 JPC방법을 이용한 실험결과와 골격구조 단순화와 기하구조 단순화 기법을 포함한 모션 다단계 기법이 적용된 실시간 균중환경에서의 시스템 성능 실험에 대해 기술한다.

골격구조 단순화 실험: JPC방법은 오차 허용 범위 내에서 가장 큰 자세 집단을 추출한다. 따라서, 기존 동작과 단순화된 동작의 시간에 따른 자세가 다르게 표현된다. 그림 9 (a), (b), (c)에서는 관절의 개수가 다르고 움직임의 특성이 다른 각각의 모션들에 대해 단순화 레벨에 따른 자세의 차이를 비교하였다. 그림에서 보듯이 단순화

레벨을 결정하는 오차 허용 범위가 낮을수록 기존 동작과 비슷한 형태로 단순화 되고 오차 허용 범위가 높을수록 기존 동작과는 차이가 있지만 단순화율이 증가한다.

JPC방법은 시간에 따라 동적으로 변하는 골격구조를 생성하기 때문에 단순화율은 전체 모션의 단순화된 관절 비율로 계산된다. 표 1은 위에 기술한 골격구조 단순화 기법에 대한 실험을 나타낸다. 표에서 보듯이 골격구조에 관절이 많을수록 캐릭터 내부에 존재하는 관절 밀도가 증가하여 관절 개수가 적은 구조에 비해 단순화율이 높게 나타난다. 또한, 관절 움직임이 적은, 변화도 낮은 모션이 변화도 높은 모션에 비해 높은 단순화율을 보인다. 변화도 낮은 동작의 경우 연속적으로 유사한 관절 자세가 빈번히 발생하면서 자세 집단 영역이 크게 형성되고 단순화율을 높인다.

Motion	Original	E=5.0	E=50.0	E=100.0	E=300.0
춤	0.0	32.1	70.8	94.1	95.8
밀기	0.0	54.9	84.3	94.3	96.4
보기	0.0	73.4	92.6	97.8	98.2

표 1. 모션 별 골격구조 단순화율 (%)

모션 다단계 기법을 이용한 성능 실험: 모션 다단계 기법의 성능을 실험하기 위해 모션 다단계 기법을 포함한 실시간 군중환경을 구현하였다. 그림 10에서 보듯이 28개 관절들과 943개 다각형들로 구성된 256개의 가상 캐릭터와 80개의 관절과 7000여 개의 다각형들로 이루어진 100개의 가상캐릭터를 OpenGL 환경에 로딩하였다. 각각의 캐릭터에는 JPC방법과 기하구조 단순화 기법을 적용하였다. 실험데모는 아래에서 확인할 수 있다. (http://vr.kaist.ac.kr/~zhaoyue/research/movie/molod_full.wmv)



(a) 춤 동작 (관절: 28개, 모션 변화도 높음)



(b) 밀기 동작 (관절 80개, 모션 변화도 높음)



(c) 보기 동작 (관절 80개, 모션 변화도 낮음)

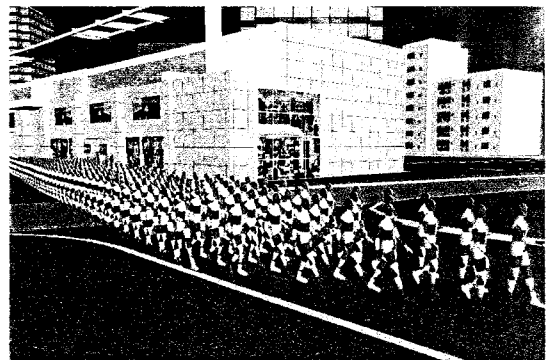


그림 9. 단순화 동작의 예

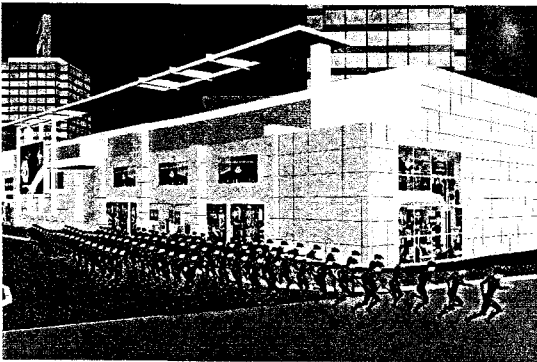
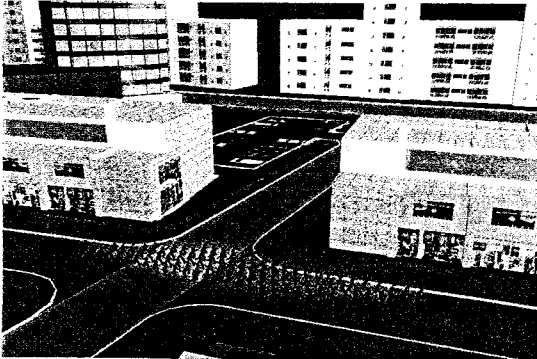


그림 10. 모션 다단계 기법을 적용한 실시간 균중환경

실험은 그림 11에서 보듯이 도시된 각각의 case에 대해 실험을 수행하였다. 그 결과, 캐릭터 애니메이션에서 렌더링 비중이 캐릭터를 움직이는 시뮬레이션에 비해 크게 작용하는 만큼 기하 구조만을 단순화한 경우가 골격구조만을 단순화한 경우보다 성능의 향상은 좋지만, 골격구조 단순화도 전체 시스템 성능 향상을 위해 필수적이다. 또한, 골격구조와 기하구조를 동시에 단순화할 경우 시스템 성능을 보다 향상시킬 수 있다.

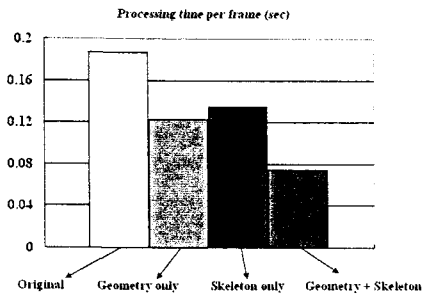


그림 11. 실시간 균중환경 성능 실험

6.2 결론 및 향후 과제

본 논문에서는 실시간 균중환경 전처리 과정인 모션 다단계 기법에 대해 기술하였다. 기존 동작과 유사한 골격구조 단순화를 위해 모션에 종속적인 단순화 기법을 제시하였다. 또한, 기하구조 단순화 기법을 적용하여 시스템 성능을 더욱 향상 시켰다. 골격구조와 기하구조의 상세도 레벨은 캐릭터와 카메라 간 거리 측정을 통해 제어되었다. 실험결과 단순화된 캐릭터의 영상과 움직임이 기존동작과 유사함을 보였다. 제시된 전처리 방법은 모션의 변화도가 낮고 관절 수가 많은 캐릭터에 대해 특히 효율적인 결과를 보였다.

모션 다단계 기법을 도시모델이나 게임과 같은 복잡한 가상환경에 적용할 경우 가시성 문제로 인해 보다 향상된 결과가 예측된다. 또한, 미래의 실시간 균중환경에 캐릭터 수가 더욱 증가하고 캐릭터 자체의 관절 수와 캐릭터 메쉬의 복잡도가 증가될 것으로 추정되어, 모션 다단계 기법에 대한 연구의 필요성은 가속화 될 것으로 전망된다.

향후 연구로 모션 다단계 기법의 완성성을 높이기 위해 아래와 같은 세 가지 측면을 고려해 볼 수 있다.

골격구조 단순화: 본 논문에서 제시된 골격구조 단순화 기법은 모션에 종속적인 단순화 기법으로 임의의 모션에 단순화를 자동으로 적용할 수 있는 대신 시간에 따라 골격의 구조가 동적으로 변화되어 계층구조를 재구성하는 비용 부담이 있다. 골격구조를 단순화하는 다른 방법으로 모션에 독립적으로 골격구조를 사전에 단순화 하고 단순화된 골격구조에 모션을 입히는 방법을 고려할 수 있다. 이와 같이 접근하는 경우 계층구조를 재구성하는 비용 부담을 줄일 수 있다.

기하구조 단순화: 제시된 기하구조 단순화는 골격구조와 독립적으로 단순화를 거친다. 캐릭터 관절 세그먼트의 관절 정보와 메쉬 정보를 하나로 모델링하여 단순화 할 경우 보다 만족스러운 결과를 기대할 수 있다.

상세도 레벨 제어: 본 논문의 상세도 레벨은 캐릭터의 위치와 카메라의 위치 간 거리를 토대로 제어된다. 레벨 제어를 보다 효율적으로 하기 위해 거리 이외에 캐릭터의 움직임 속도, 영상에 투영된 캐릭터 크기, 영상에서 캐릭터의 위치 등을 고려해야 한다.

감사의 글

본 연구는 가상현실 연구센터의 지원으로 이루어졌음을 밝힙니다.

참고문헌

- [1] Z. Popovic and A. Witkin, Physically based motion transformation, *ACM SIGGRAPH 99*, pages 11-20, 1999.
- [2] D. Carlson and J. Hodgins, Simulation levels of detail for real-time animation, *Graphics Interface 97*, 1997.
- [3] H. Hoppe, Progressive mesh, *ACM SIGGRAPH 96*, pages 99-108, 1996.
- [4] M. Garland and P. Heckbert, Mesh simplification with quadric error metrics, *ACM SIGGRAPH 97*, pages 209-216, 1997.
- [5] H. Kim and K. Wohn, Multiresolution model generation with geometry and texture, *VSMM 2001*, 2001.
- [6] F. Tecchia and Y. Chrysanthou, Real-time rendering of densely populated urban environments, *10th Eurographics Workshop on Rendering*, pages 45-56, 2000.
- [7] A. Aubel, R. Boulic and D. Thalmann, Real-time display of virtual humans: Level of details and impostors, *IEEE Transactions on Circuits and Systems for Video Technology, Special Issue on 3D Video Technology*, 10(2): 207-217, 2000.
- [8] E. Bouvier, E. Cohen and L. Najman, From crowd simulation to airbag deployment: particle systems, a new paradigm of simulation, *Journal of Electronic Imaging*, 6(1): 94-107, 1997.
- [9] S. Musse and D. Thalmann, Hierarchical model for real time simulation of virtual human crowds, *IEEE Transactions on Visualization and Computer Graphics*, 7(2): 152-164, 2001.
- [10] D. Helbing, A fluid-dynamic model for the movement of pedestrians, *Complex Systems*, pages 391-415, 1992.
- [11] S. Shekhar and Q. Lu, Evacuation planning algorithms: A capacity constrained routing approach, 2002.
- [12] X. Tu and D. Terzopoulos, Artificial fishes: physics, locomotion, perception, behavior, *ACM SIGGRAPH 94*, 1994.
- [13] D. Brogan and J. Hodgins, Group behaviors for systems with significant dynamics, *Autonomous Robots*, 4: 137-153, 1997.
- [14] S. Musse and D. Thalmann, A model of human crowd behavior, *Workshop of Computer Animation and Simulation of Eurographics 97*, 1997.
- [15] C. Reynolds. Flocks, herds and schools: A distributed behavioral model, *ACM SIGGRAPH 87*, 1987.
- [16] T. Kim, S. Park and S. Shin, Rhythmic motion synthesis based on motion beat analysis, *ACM SIGGRAPH 03*, 2003.
- [17] B. Blumberg and T. Galyean, Multi-level direction of autonomous creatures for real-time virtual environments, *ACM SIGGRAPH 95*, pages 47-54, 1995.
- [18] K. Perlin and A. Goldberg, Improv: A system for scripting interactive actors in virtual worlds, *ACM SIGGRAPH 96*, pages 205-216, 1996.
- [19] J. Lee, J. Chai, P. Reitsma, J. Hodgins and N. Pollard, Interactive control of avatars animated with human motion data, *ACM SIGGRAPH 02*, 2002.
- [20] J. Lee and S. Shin, A Coordinate-Invariant Approach to Multiresolution Motion Analysis, *Graphical Models (Formerly GMIP)*, 63(2): 87-105, 2001.