

## 효율적인 애니메이션을 위한 모션 에디팅 방법의 계산량분석에 관한 연구

### Analysis on the Computational Complexities of Motion Editing for Graphic Animation

이 지 흥, 김 인 식, 김 성 수  
(Jihong Lee, Insik Kim, and Sungsu Kim)

**Abstract :** Regarding efficient development of computer graphic animations, lots of techniques for editing or transforming existing motion data have been developed. Basically, the motion transformation techniques follow optimization process. To make the animation be natural, almost all the techniques utilize kinematics and dynamics in constructing constraints for the optimization. Since the kinematic and dynamic structures of virtual characters to be animated are very complex, the most time-consuming part is known to the optimization process. In order to suggest some guide lines to engineers involved in the motion transformation, in this paper, we analyze the computational complexities for typical motion transformation in quantitative manner as well as the possibility for parallel computation.

**Keywords :** motion transformation, motion capture, space time editing

#### I. 서론

최근의 영상매체에서 다루는 영상들은 컴퓨터 그래픽스 기술 없이는 제작이 불가능하다고 해도 과언이 아닐 정도로 컴퓨터 그래픽스 기술에 의존하고 있다. 이러한 컴퓨터 그래픽스 기술의 핵심은 얼마나 현실감 있느냐하는 것이라는 것에는 이견이 없다. 이렇게 현실감 있는 영상제작을 위해서는 가상적인 캐릭터의 기구학적, 동력학적 특성과 환경적 특성이 완벽히 고려되어야만 그 현실감이 보장된다고 할 수 있다. 그런데 이러한 가상 캐릭터의 여러 특성을 가장 실질적으로 반영하는 영상 생성 방법은 모션 캡처(Motion Capture) 방법일 것이다. 모션캡처[1]은 센서를 사용하여 움직이는 대상의 동작 데이터를 얻는 기술로써, 현실 세계에 존재하는 물체의 동작을 직접 받아들이기 때문에 현실감 있는 데이터를 얻을 수 있다는 장점을 갖고 있다. 이러한 장점 때문에 현재 애니메이션 분야에 널리 이용되고 있다. 하지만, 모션 캡처 시스템의 가격이 고가이고, 대부분의 장비들이 제한된 환경 하에서만 작동하도록 설계되어 있기 때문에 어느 정도의 가동(setup) 시간을 필요로 하는 단점을 지니고 있으며, 캡처된 데이터의 재사용률이 낮다. 예를 들면, 사람의 걷는 동작을 갖고 있는 사용자가 걷는 동작중 장애물이 발생했을 경우의 데이터가 필요하다고 가정하자. 이 경우 사용자는 다시 장애물을 모션캡처 환경에 첨가하여 다시 동작 데이터를 캡처해야 한다. 이처럼 어느 정도 비슷한 동작들을 필요할 때마다 다시 캡처해야 하는 번거로움을 갖고 있다. 그리고 모션 캡처가 불가능한 대상체, 예를 들면 공룡이

나, 우주생물들의 동작은 현실적으로 얻어낼 수 없다는 근본적인 문제점을 가지고 있다.

한편, 모션 트랜스포메이션(Motion transformation)은 쉽게 얘기하면 모션 캡처 기술을 사용하여 얻어진 데이터들을 사용자가 원하는 데이터로 변환하는 기술이다. 이 기술은 기존 데이터들을 사용하여 새로운 데이터를 생성하는 것이기 때문에 모션 캡처 기술에 비하여 최소한의 시간과 비용으로 새로운 데이터를 만들 수 있다는 장점을 가지고 있다. 따라서, 기존 데이터를 갖고 있다는 가정 하에서는 모션 트랜스포메이션 기술이 유용하게 쓰일 수 있다. 위에서 예들 든 걷는 동작의 경우 모션 트랜스포메이션 기법은 사용하여 장애물을 구속조건으로 설정하여 기존의 모션 데이터를 장애물에 맞도록 바꿀 수 있다. 물론 모션캡처가 현실 세계의 데이터를 받아들이기 때문에 어느 경우든 정확한 데이터를 얻을 수 있지만, 비슷한 동작의 변형의 경우 모션 트랜스포메이션이 모션캡처 보다 시간을 절약하며 사용자가 원하는 결과를 얻을 수 있다. 여기서 동작 데이터의 변환에 사용되는 여러 가지 제약 조건이란 사실상 가상 캐릭터의 기구학적, 동력학적 특성을 말한다. 다시 말해서 이 제약 조건에 충실하면 할수록 그 동작이 현실감 있게 보이게 되므로 기존 동작의 변환이라는 목적과 최종 결과 동작이 일련의 제약 조건을 만족시켜야 한다는 전형적인 최적화 문제로 정리된다. 이러한 최적화 문제로 정리된 모션 트랜스포메이션 기법은 우선 프레임 수에 비례하여 변수가 늘어나며, 기구학적 제약조건은 행렬 연산으로, 동력학적 제약 조건은 행렬 연산 및 미분 방정식 형태로 주어지게 되며, 최적 화하려는 목적함수나 제약조건을 나타내는 조건 식이 대부분의 경우 비선형적인 형태를 가지게 되어 한번의 계산에 의해 한번에 (Closed form)로 해를 구하는 것이 불가능하다. 그러므로 반복적인 계산에 의해 점차적으로 최적해로 접근해야 하는

접수일자 : 2001. 6. 1., 수정완료 : 2001. 10. 26.  
이지흥, 김인식, 김성수 : 충남대학교 메카트로닉스공학과(jihong@cnu.ac.kr/mulderf@hitel.net/kimss@pony.cnu.ac.kr)  
※ 본 논문은 정보통신연구진흥원의 지원하에 이루어졌으며, 과제 번호는 2001-152-2입니다.

데 이 과정이 현실감 있게 동력학식을 자세히 적용할수록 많은 시간을 요구하게 된다. 본 연구에서는 이러한 모션 트랜스포메이션의 여러 기술 중에서 가장 현실을 많이 고려한 스페이스 타임(Space-time edition) 방법[2]의 내용을 정리하고 가장 많은 계산량을 요구하는 동력학 식을 구성하는 여러 요소 중에서 어떤 항들이 어떤 상황에서 어느 정도의 계산량을 차지하는가 하는 문제를 분석하여 근사화를 하는 가이드라인을 제시하고자한다.

## II. 기존의 연구들

모션 트랜스포메이션에 의한 컴퓨터 애니메이션에 관한 연구는 다양한 방법에 의해 발전되어 왔다. 그 중에서도 특히 제약조건(constraint)을 기반으로 한 방법은 애니메이션 연구의 큰 줄기를 이루고 있다. 제약조건을 해결하는 방법은 여러 사람에 의해 연구되어 졌으며, 그 중에서도 Badler [3]은 애니메이션에서 관절을 가진 캐릭터의 위치 제어를 위해 여러 가지 제약조건을 해결하는 방법을 제시했으며 Witkin[4]는 제약조건을 에너지의 함수로 정의하고 해결하는 방법을 제시했다.

제약조건을 기반으로 한 방법 중 하나가 역기구학(Inverse kinematics)을 이용하는 것이다. 역기구학은 주로 로봇틱스와 같은 기계적인 분야에서 로봇의 동작을 제어하기 위해 연구되어졌으나, Welman[5], Zhao과 Badler[6]은 컴퓨터 애니메이션에서 관절을 가진 캐릭터의 동작을 만들기 위해 이 방법을 이용했으며 그 유용성을 증명했다.

물리적 법칙을 이용한 시물레이션 기법 또한 제약조건을 기반으로 한 컴퓨터 애니메이션의 한 가지 방법이다. 이것은 시물레이션 할 대상을 먼저 수학적으로 모델링 한 다음 그 대상을 원하는 동작을 취하게 하기 위해서 가해져야 할 힘과 토크를 구해야 하는 과제를 안고 있다. Reeves[7]이 개체 시스템(particle system)에서 물체의 운동을 수학적으로 모델링하는 기법을 소개하였으며, 연구가 진행됨에 따라 더 복잡한 시물레이션 대상과 주위 상황을 얼마나 더 정확히 모델링 하는가 하는 것이 또한 연구 과제로 등장하게 되었다. 이러한 의미에서 Baraff[8]의 연구는 충돌이나 마찰과 같은 복잡한 영향에 대해서도 더 정확하게 모델링 할 수 있도록 한 계기가 되었다. 시물레이션 기법에서는 제약조건을 만족하는 동작을 만들기 위해 역동력학 및 동력학을 그 도구로 사용하며, 역동력학은 Armstrong과 Green[9], Wilhelms [10], Issacs와 Cohen[11], 그리고 고형석과 Badler[12]에 의해 소개되었다. 그리고 Barzel과 Barr[13]와 Platt[14]는 동력학적 제약조건을 소개하였다.

지금까지 설명한 제약조건을 기반으로 한 애니메이션 방법은 각각의 프레임만을 고려한 방법이다. 그러므로 이전 프레임 및 다음 프레임과의 연결에 있어서 약간의 부자연스러움을 느낄 수 있다. 이러한 점을 보완하기 위해 공간-시간(spacetime) 제약조건 개념이 도입되었다. 이 방법은 Witkin 과 Kass[15]와 Brotman과 Netravali[16]에 의해 처음으로 소개되었으며 전체 동작의 제약조건을 동시에 고려하면서 역기구학나 역동력학을 이용하여 최적의 동작을 찾는다. 이 방법은 캐릭터의 동작에 있어서 처음과 끝을 동시에 고려하

기 때문에 좀더 자연스러운 동작을 연출할 수 있는 장점이 있지만 많은 양의 데이터를 한꺼번에 계산한다는 점에서 계산량이 너무 많고 복잡해지는 단점이 있다. 그래서 얼마나 효율적으로 공간-시간 제약조건 문제를 해결하는가 하는 것이 과제가 되었으며 Ngo와 Marks[17], 그리고 Panne과 Fiume [18]이 그에 대한 연구를 하였다.

지금까지 다룬 기법들은 주로 모션 생성에 관련된 방법들이다. 이러한 방법들은 모두 특정 목적에 의해 만들어진 동작은 그 외의 다른 목적으로는 다시 사용할 수 없는 단점을 갖고 있다. 그래서 많은 사람들은 그러한 단점을 보완하기 위해 미리 만들어진 동작을 새로운 동작으로 변형하는 분야로 최근의 애니메이션 연구들은 많이 바뀌어 가고 있다.

특히 Bruderlin과 Williams[19]는 모션 캡처 시스템이나 키 프레임 시스템에서 만들어진 파라미터들의 값들을 동작 신호로 취급하며 그 신호를 변형하는 방법을 제시하고 있으며, Witkin과 Popovic[20]은 키 프레임 방법을 이용하여 걸어가는 사람의 형상이나 테니스에서 스윙을 하는 것과 같은 복잡한 모션을 에디팅하는 방법을 제시해 주고 있다. 이 외에도 미리 만들어진 캐릭터의 동작을 다른 동작을 하게끔 변형시키는 모션 에디팅에 관한 연구는 Gleicher와 Litwinowicz [21], Gleicher[22], 그리고 Boulic과 Thalmann[23]에 의해 진행되어졌다. 이와는 조금 다른 개념으로 미리 만들어진 캐릭터의 동작을 다른 캐릭터의 동작으로 변형하는 모션 리타게팅(Retargeting)에 관한 연구도 활발히 진행되고 있다. Hodgins와 Pollard[24]는 시물레이션 된 캐릭터의 동작을 새로운 캐릭터의 동작으로 바꾸기 위한 연구를 했으며, Gleicher [25] 또한 동작 리타게팅에 관한 연구를 하였다. 최근 들어 모션 캡처 기술은 캐릭터 애니메이션에 있어서 가장 각광받는 기술 중 하나가 되었다. 특히 광학식 혹은 마그네틱 모션 캡처 시스템을 이용해서 실제 배우나 사물들의 움직임을 동작 신호 혹은 동작 데이터로 저장함으로써 실제와 같은 동작 애니메이션을 만들 수 있다. 그러나 이러한 모션 캡처 시스템은 몇 가지의 단점을 가지고 있다. 그 중 하나는 움직이는 배우나 물체에 부착되어 있는 센서의 움직임을 측정함에 있어서 약간의 측정오차이다. 이 오차에 의해서 만들어진 애니메이션은 떨림과 불안정함을 갖게 된다. 또 다른 단점은 모션 캡처 시스템에서 얻어진 동작 신호를 애니메이션으로 만들 경우 실제 동작과는 약간의 차이가 발생한다는 것이다. 이러한 점을 보완하기 위해서 Bruderlin과 Williams[19], Witkin과 Popovic[20], Unuma, Anjyo와 Takeuchi [26], 그리고 Perlin[27]은 모션 캡처 시스템에서 만들어진 불안정한 신호를 조작함으로써 실제적이고 안정된 동작이 만들어지도록 하는 방법을 제시하고 있다.

또한 국내에서도 탁세윤과 고형석[28], 최광진, 박상현 그리고 고형석[29] 등에 의해서 이와 같은 연구가 진행되었다. 탁세윤과 고형석은 역기구학과 모션 캡처를 조합한 EGIK (example guided 역기구학)를 소개함으로써 어느 정도의 한계가 있는 모션 캡처 시스템의 데이터를 이용해서 많은 DOF(Degree Of Freedom)를 가진 모델을 효율적으로 애니메이션 할 수 있는 방법을 제시하였으며, 최광진, 박상현 그리고 고형석은 모션 캡처 시스템에서 얻어진 조인트 각도와

링크 길이의 부정확함 때문에 이들 데이터를 바탕으로 애니메이션을 제작할 경우 엔드-이펙터의 경로가 실제 동작과는 어느 정도의 차이가 존재한다는 생각으로부터 엔드-이펙터의 경로를 실제 동작과 같게 만들기 위한 동작 데이터의 조작 알고리즘을 소개하였다.

모션 캡처 시스템으로부터 얻어지는 동작 신호의 조작과 더불어 그 신호를 조작함으로써 다른 캐릭터 혹은 다른 환경에서의 동작으로 변형시키는 모션 에디팅과 모션 리타게팅 또한 현재 국내에서 진행되고 있는 컴퓨터 애니메이션에 관한 연구의 큰 축을 이루고 있다. 특히 이제희와 신성용[30]과 최광진과 고희석[31]이 그에 관한 연구를 하였다. 최광진과 고희석은 Whitney[32]에 의해 소개된 역속도제어(Inverse rate control)를 기반으로 하는 온-라인 모션 리타게팅의 예를 보여주고 있으며 그들의 OMR(On\_line Motion Retargeting) 필터에 의해 캡처된 신체 끝단 부위의 위치 및 관절 각도의 정확성을 높였다. 이 외에도 새로운 역기구학 해결법을 소개하고, 그로 인해 37개의 DOF를 가진 캐릭터의 동작을 에디팅 할 수 있게 한 이제희와 신성용[33]의 모션 에디팅에 관한 연구가 있었으며, 탁세윤, 송오영 그리고 고희석[34]은 균형 잡히지 않은 동작을 균형 잡히도록 변형하는 방법을 소개하였으며, 이들은 움직이는 동작의 균형문제를 다룸으로써 기존의 정지 자세에서의 균형을 다루던 연구와 차별을 두었다. 또한 최근 들어 실시간 가상 인물 애니메이션 시스템이 활발하게 보급되면서 피부 모양을 자연스럽게 유지하면서도 고속으로 동작할 수 있는 인체 변형 방법이 요구되고 있다. 이에 대해 신승협과 신성용[35]는 가상 인물을 구성하는 매쉬의 정점들을 회전 연산만으로 이동시켜 실시간에 인체를 변형하는 모델을 제안하였고, 가상 캐릭터의 사실적인 애니메이션에 있어서 얼굴 표정 생성의 중요성이 부각되면서 컴퓨터 애니메이션의 새로운 연구 분야로 자리 잡고 있다.

### III. 이론

본 장에서는 모션 트랜스포메이션 수행 과정에 관하여 설명한다. 모션 트랜스포메이션은 최적화 기법을 사용하여 구속조건을 만족하는 해를 찾는 과정이다. 따라서, 최적화 과정에 대한 설명과 최적화 과정에 사용되는 구속조건 중 가장 중요한 조건인 가상 캐릭터 동력학식에 관하여 먼저 살펴보기로 하자.

#### 1. 동작 데이터

보통 동물들은 여러 개의 관절을 사용하여 자신의 몸을 움직이거나 물체를 운반하는 일을 수행한다. 결국, 모든 관절 값들을 안다면 우리는 그 물체의 현재 상태를 표시할 수 있다.  $n$ 개의 관절을 갖는 물체의  $m$ 번째 동작 프레임의 상태를  $Q_m$ 이라 표시하고, 각 관절들을  $q_i$ 라고 표시하면  $Q_m$ 은  $Q_m = [q_1^m, q_2^m, q_3^m, \dots, q_i^m, \dots, q_n^m]^T$ 로 표시된다. 앞으로 어떤 고정된 프레임을 명시하는 것이 분명한 경우에  $Q_m = [q_1, q_2, q_3, \dots, q_i, \dots, q_n]^T$ 로 간단히 표시하기로 함을 밝혀둔다. 따라서, 모션 캡처 기법에 의해서 얻은 데이터들은 이러한  $Q_m$ 의 집합이라고 할 수 있다. 전체 캡처 데이터를  $Q$ 라 하면  $Q$ 는 (1)과 같이 표시된다.

$$Q = \bigcup_m Q_m \quad (1)$$

여러 가지 센서를 사용하여  $Q_m$ 들을 얻는 방법이 모션 캡처이며, 여러 가지 방법에 의하여 얻어진  $Q$ 를 사용자의 요구에 맞게 변형하여  $Q'$ 을 만드는 방법이 모션 트랜스포메이션이다. 이제  $Q \rightarrow Q'$ 로 바꾸는 과정에 대하여 살펴볼 것이다.

#### 2. 최적화 기법

$Q \rightarrow Q'$ 으로 변환하는 과정은 구속 조건과 목적 함수로 구성된 최적화 과정을 필요로 하게 된다. 예를 들면, 모션 캡처를 통하여 얻어진 걷는 데이터를 한쪽 다리를 부상당한 사람의 걸음으로 바꾸고 싶다고 가정하자. 이 경우, 구속 조건은 부상당한 다리의 관절을 사용하지 않는 것이며, 목적 함수는 사용자의 목적에 따라서 달라진다. 물체의 움직임을 최소화하고 싶다면, 에너지 함수가 목적 함수가 된다. 자연스러운 움직임을 강조하고 싶다면 근육변수(나중에 추가 설명을 하기로 한다)의 부드러움(smoothness) 함수가 목적함수로 된다. 이를 수식적으로 표현하면 다음과 같다.

$$\min E(q(t), t) \text{ subject to } C(q(t), t) = 0 \quad (2)$$

여기서,  $E(q(t), t)$ 는 목적함수,  $C(q(t), t)$ 는 구속 조건 식을 표현한다. (2)를 만족하는  $q$ 들을 구하는 것이 최적화 과정이다. 본 논문에서는 최적화 기법으로 SQP(Sequential Quadratic Programming) 기법을 사용하기로 한다. 모션 트랜스포메이션은 여러 번의 최적화 과정이 필요하며,  $q_i$ 를 구하는 과정마다 최적화를 위한 목적 함수와 구속조건이 달라진다.

#### 3. 구속조건

구속조건은 위치(pose), 기계적(mechanical), 동력학(dynamic) 구속조건 세 가지가 있다. 위치 구속조건은 물체의 위치를 표현하는 것으로 대상물이 사람일 경우 신체의 특정 부위가 일정 시간에 어느 장소를 지나야 한다는 의미를 내포하고 있다. 따라서, 우리는 걷는 동작의 발의 위치를 바꾸어 사람의 걸음걸이를 바꾸거나, 머리를 숙이고 걸게 하고 싶을 경우 위치 구속조건을 사용해야 함을 알 수 있다. (3)은 위치 구속 조건 중 발 위치 구속 조건을 수식으로 표현한 것이다.

$$C_p = P_{foot}(Q_m) - P_{floor} = 0 \quad (3)$$

$P_{foot}$ 은 물체의 특정 부위의 위치를 표시하며,  $P_{floor}$ 는 물체가 지나가야 할 위치를 표현한다. 두 번째 기계적인 구속 조건은 위치 구속 조건과 비슷하지만 힘을 포함하고 있다는 점이 다르다. 이 구속조건은 움직이는 대상과 주위 환경과의 관계를 변화시킬 경우 사용된다. 사람의 걷는 동작은 일반적인 인도를 걸을 때와 미끄러운 눈길을 걸을 때, 지면의 반발력이 심한 장소를 걸을 때 각각 다르다. 이러한 현상의 원인은 일반적인 지면과 발 사이의 힘의 합이 0이 되어야 하기 때문이다. 미끄러운 눈길을 걸을 경우는 일반적인 인도와 마찰력이 다르기 때문에 평범한 걸음걸이를 하게 되면 균형을 잃거나 넘어지게 된다. 또한, 지면의 반발력이 심한 경우에는 지면의 반발력을 제거하기 위한 힘이 필요하게 된

다. 이러한 이유로 기계적 구속 조건이 필요하다. 기계적 구속 조건을 유도하기 위해서는 Lagrange multiplier theorem을 소개한다.

**Theorem 1. Lagrange Multiplier Theorem**  
 다음 식을 만족하는 벡터  $s$ 와  $b$ 와 행렬  $A$ 가 있다고 하자.  
 $s^T b = 0$   
 $A s = 0$   
 이때, 다음 식을 만족시키는 Lagrange multiplier가 존재한다.  
 $s^T b + s^T A^T \lambda = 0$

Lagrange multiplier는 동역학 방적 식의 구속 조건을 구할 때 사용된다. 마지막으로 동역학 구속조건은 모션 트랜스포메이션 과정중 발생할 수 있는 비현실성을 제거하기 위하여 사용된다. 즉, 동역학 구속조건을 만족하지 않은 상태에서 얻어진 결과는 현실과 동떨어진 결과라고 볼 수 있다. 따라서, 동역학 구속 조건을 포함하지 않는 모션 트랜스포메이션은 현실 세계와 동떨어진 결과를 얻을 수 있다는 측면에서 중요한 구속조건이다. 또한, 중력을 포함하고 있기 때문에 중력이 다른 달이나 화성에서 사람의 걸음걸이가 어떻게 달라질지 짐작 할 수 있게 해준다. 기계, 동역학 구속조건과 관련된 수식은 다음절에서 다룬다.

4. 기계적인 동역학 구속조건

$$C_n = tr(\frac{\partial W_n}{\partial q_j} \widehat{M}) + g \frac{\partial W_n}{\partial q_j} \widehat{c}_n + \sum_k \frac{\partial F_k \phi_{Fk}}{\partial q_j} + \sum_l q_l^i \frac{\partial C_m}{\partial q_j} \quad (4)$$

(4)는 인체 동역학 식을 표시한 것으로 물체의 kinetic energy와 관련된  $tr(\frac{\partial W_n}{\partial q_j} \widehat{M})$ , 물체의 potential energy 항인  $g \frac{\partial W_n}{\partial q_j} \widehat{c}_n$ , 물체에서 발생하거나 외부에서 작용하는 힘을 표시하는  $\sum_k \frac{\partial F_k \phi_{Fk}}{\partial q_j}$  항, 마지막으로 mechanical 구속 조건 항인  $\sum_l q_l^i \frac{\partial C_m}{\partial q_j}$  으로 구성된다. 이제 (4)에 대하여 살펴보고, 각 항들의 유도 과정을 알아본다.

4.1 노드

먼저 모션 트랜스포메이션에 사용되는 물체들은 그림1과 같은 기구학적 구조로 되어있다.  $R_0$ 는 중심이 되는 좌표계로써 대상체가 사람일 경우 골반에 해당한다. 중심 좌표계를 중심으로 신체가 구성되어 있기 때문에  $R_0$ 가 변하게 되면  $R_0$ 를 기반으로 하는 모든 좌표계가 변화하는 구조로 되어 있다.

즉,  $R_i$ 를  $i$ 번째 노드(관절)이라 하고  $i-1$ 번째 노드를  $R_{i-1}$ 라 하면  $R_i$ 는  $R_{i-1}$ 의 자식 노드이며,  $R_{i-1}$ 은  $R_i$ 의 부모 노드가 된다. 부모 노드가 변하게 되면 자식 노드는 같이 변하

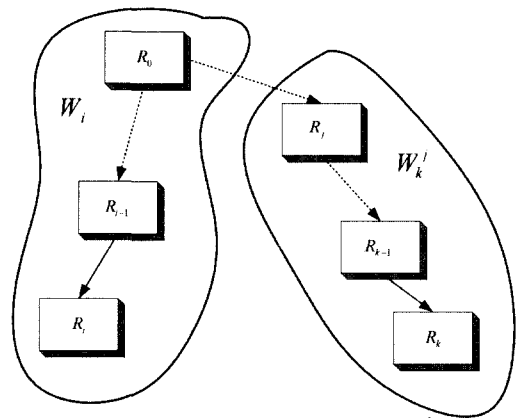


그림 1. 캐릭터의 계층 표시

Fig. 1. Notation for character hierarchy.

며, 자식노드는 부모 노드에 아무런 영향을 주지 못한다. 이러한 자식, 부모 노드는 상대적인 개념으로 다른 모든 노드에도 적용된다.

4.2 위치, 속도, 가속도 행렬과  $\widehat{M}_i, \widehat{c}_i$

$W_i$ 는 기본 좌표계로 표시된  $i$ 번째 노드의  $4 \times 4$  좌표변환(Homogeneous transformation) 행렬이다. 이 행렬은  $i-1$ 번째 노드의 좌표변환 행렬인  $W_{i-1}$ 에서 파생되는 행렬로

$$W_i = W_{i-1} R_i \quad (5)$$

로 표현된다.  $R_i$ 는  $i-1$  좌표계로 표시된  $i$ 번째  $4 \times 4$  좌표변환 행렬을 나타낸다. 또한,  $W_i$ 을 미분한 결과인  $\dot{W}_i$ 와  $\ddot{W}_i$ 는

$$\dot{W}_i = \dot{W}_{i-1} R_i + W_{i-1} Q R_i \quad (6)$$

$$\ddot{W}_i = \ddot{W}_{i-1} R_i + 2 \dot{W}_{i-1} \dot{R}_i + W_{i-1} Q \dot{R}_i \quad (7)$$

(6),(7)과 같은 순차적인(recursive) 형태로 구할 수 있다.  $Q$ 는 행렬  $R_i$ 을 미분하는 행렬로 회전 행렬에 따라 달라진다.

$\widehat{M}_i, \widehat{c}_i$ 는 순차적인 형태로 표현된 관성, 무게 중심과 관련된 새로운 변수를 표시한다. 새로운 변수를 사용하여 동역학 식을 보다 간결하게 만들 수 있다.

$$\widehat{M}_i = M_i \ddot{W}_i + \sum_j R_j \widehat{M}_j \quad (8)$$

$$\widehat{c}_i = m_i c_i + \sum_j R_j \widehat{c}_j \quad (9)$$

$M_i$ 는  $i$ 번째 노드의  $4 \times 4$  관성 행렬,  $m_i$ 는  $i$ 번째 노드의 질량,  $c_i$ 는  $i$ 번째 노드의  $4 \times 1$  무게중심 행렬이다.

4.3 가해지는 힘,  $\sum_k \frac{\partial F_k \phi_{Fk}}{\partial q_j}$

물체에 가해지는 외부 힘이나 물체의 내부 힘을 표현하는 식이다. 살아있는 생물의 경우 자신의 근육을 사용하여 움직인다. 따라서, 물체의 내부 힘은 근육에 의해 발생하는

힘을 말하는 것이다. 여기서 우리는 새로운 변수인  $q_{mi}$ 를 소개한다. 근육은 수축만을 통해서 운동을 유발하지만 대부분의 경우 쌍으로 존재하여 서로의 수축운동이 반대편의 팽창을 유발하므로 이를 기계적으로 모델링하기 위해 댐프드 서보 모델(damped servo model)을 사용하였다. 따라서, 근육에 의해 발생하는 힘( $F_m$ )은 (10)으로 표시된다.

$$F_m = k_s(q_k - q_{mk}) + k_d(\dot{q}_k - \dot{q}_{mk}) \quad (10)$$

$k_s$ 는 스프링 상수이며  $k_d$ 는 댐핑 상수이다. 이외의 외부에서 작용하는 힘( $F_A$ )를 더하면 물체에 가해지는 모든 힘을 표현하게 된다.

4.4 기계적 구속조건,  $\sum_j a_j^i \frac{\partial C_m}{\partial q_j}$

$C_m$ 은 물체가 주위 환경과 접촉하는 부분의 자세(Pose) 구속 조건을 표시한다. 모션 트랜스포메이션을 적용할 물체가 사람인 경우 사람의 발과 지면이 접촉하고 있기 때문에 발의 위치를 표시하는 식이 되며,  $\frac{\partial C_m}{\partial q_j}$ 는 로보틱스 분야에서 자주 사용되는 자코비안(Jacobian)이 된다. 동력학 방정식에서 자코비안에 Lagrangian multiplier  $q_j^i$ 를 곱하게 되면 제약조건 힘을 표현하게 된다.

IV. 인체 동력학식의 계산량

지금까지 우리는 인체 동력학식을 구성하는 각 항들에 대하여 살펴보았다. 일반적으로 동력학식은 많은 계산량과 비선형성을 갖는 것으로 알려져 있다. 본 논문에서 사용한 동력학식도 많은 계산량과 비선형성을 갖고 있기 때문에, 최적해를 찾기 위한 최적화 과정에서 대부분의 시간을 차지한다. 이제 (4)의 계산량을 수식적으로 구해 보겠다.

(4)는  $4 \times 4$  행렬과  $4 \times 1$  행렬의 곱셈과 덧셈으로 구성되어 있다. 각 식의 계산량을 알아보면 표 1 과 같다. 표 1에서 구한 계산식을 사용하여 모션 트랜스포메이션의 대상이 사람인 경우 계산되는 계산량을 알아보겠다. 계산량을 알아보기 전에 먼저 Popovic이 주장한 원래 모델과 간단화된 모델에 대하여 설명하고, 두 모델의 계산량의 차이가 얼마나 되는지 알아보도록 하겠다. 본 논문의 서두에서도 언급했듯이 사람의 신체는 많은 관절로 구성되어 있다. 이렇게 많은 관절을 모두 모델링 하려면 이에 대응하는 많은 변수들을 필요로 하게 된다. 따라서, 원래 모델은 사람의 모든 관절을 모두 변수로 지정하여 모델링한 모델을 일컬으며, 간단화된 모델은 사람의 관절 중 해당 동작에 크게 관여하는 관절들만을 모델링한 모델이다. 예를 들어 제자리멀리뛰기 동작을 생각해 보면, 두 팔과 두 다리는 모두 동일한 동작을 하며 뛰게 된다. 따라서, 사용자는 두 팔 중에 한 팔을, 두 다리 중에 한 다리를 생략하여 모델링 하게 된다. 이러한 모델을 간단화된 모델이라 한다. 원래 모델은 현실 세계의 물체를 그대로 모델링했기 때문에 최적화 과정을 거쳐 얻어진 결과는 그대로 원래 모델에 적용할 수 있는 반면, 변수가 많기 때문에 최적화 과정에서 시간의 소모가 많아진다. 한편, 간단

표 1. 인체 동력학 계산식.

Table 1. Computational complexity for calculating human dynamics.

식	곱셈	덧셈
$W_i = W_{i-1} R_i$	$64(n-1)$	$48(n-1)$
$\dot{W}_i = \dot{W}_{i-1} R_i + W_{i-1} \dot{R}_i \dot{q}_i$	$208(n-1)$	$160(n-1)$
$\ddot{W}_i = \ddot{W}_{i-1} R_i + 2 \dot{W}_{i-1} \dot{R}_i \dot{q}_i + W_{i-1} ( \ddot{Q} \dot{R}_i \dot{q}_i^2 + \dot{R}_i \ddot{q}_i )$	$385(n-1)$	$288(n-1)$
$\frac{\partial W_i}{\partial q_j} = W_{k-1} \frac{\partial R_k}{\partial q_j} W_i^{k-1}$	$128(n-1)$	$96(n-1)$
$\hat{c}_i = m_i c_i + \sum_j R_j \hat{c}_j$	$20n-16$	$16n-28$
$\hat{M}_i = M_i \hat{W}_i^T + \sum_j R_j \hat{M}_j$	$128n-64$	$112n+64$
$g \frac{\partial W_n}{\partial q_j} \hat{c}_n$	$20n$	$15n$
$\sum_k \partial F_k \frac{\partial F_k}{\partial q_j}$	$4n$	$3n$
$\sum_j q_j^i \frac{\partial C_m}{\partial q_j}$	$36n$	$30n$
$tr( \frac{\partial W_n}{\partial q_j} \hat{M} )$	$64n$	$4n$
$C_n = tr( \frac{\partial W_n}{\partial q_j} \hat{M} ) + g \frac{\partial W_n}{\partial q_j} \hat{c}_n + \sum_k \partial F_k \frac{\partial F_k}{\partial q_j} + \sum_j q_j^i \frac{\partial C_m}{\partial q_j}$	$1057n-865$	$772n-556$

가정 - Jacobian을 먼저 구한 상태이며, 외부에서 작용하는 힘이 중력뿐이고, 오직 발만이 외부와 접촉하고 있다. n=물체의 자유도

화된 모델은 변수가 원래 모델에 비하여 적기 때문에 최적화 과정에서 소요되는 시간을 줄일 수 있다. 하지만, 간단화된 모델을 다시 현실 세계의 원래 모델로 적용해야 하기 때문에 추가적인 최적화 과정이 필요하다.

지금까지 설명한 두 모델 각각의 동력학식의 계산량을 계산해 보도록 하겠다. 먼저 대상체를 사람으로 설정하고, 원래 모델과 간단화된 모델의 변수를 설정하였다. 그림 2는 각 모델의 차이점을 표시하고 있다. (●)는 3 자유도 joint, (⊗)는 2 자유도 joint

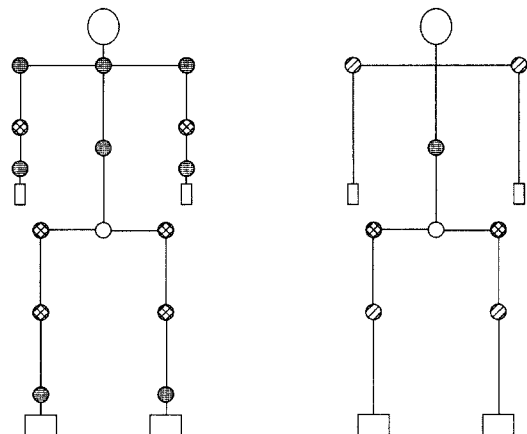


그림 2. 원래 모델 과 간단화된 모델.  
Fig. 2. Structure of complex model and simplified model.

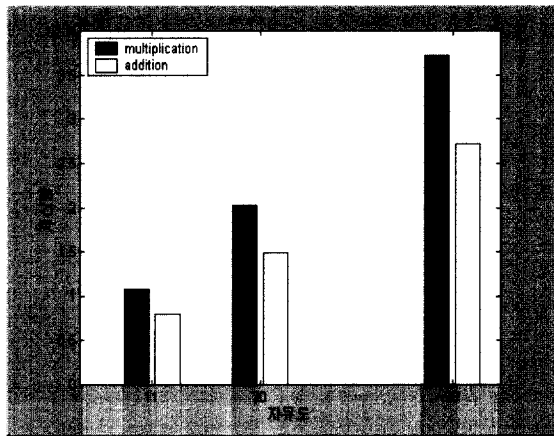


그림 3. 계산량 결과.  
Fig. 3. Computational complexity(actual computation).

는 2자유도 조인트, ⊗는 1 자유도 조인트를 표시한다. 각 모델들의 자유도를 계산하면, 원래 모델은  $3 \times 8 + 2 \times 6 = 36$ 이며, 간단화된 모델의 경우는  $3 \times 1 + 2 \times 2 + 1 \times 4 = 11$ 이다. 계산량을 구해보면 그림3의 결과를 얻었다. 자유도가 20인 모델을 추가하였다.

표 1에서 구한 계산식과 모델별 자유도를 사용하여 우리는 계산시간을 계산 할 수 있다. Intel사의 Pentium processor (100MHz)를 사용하고, 필요한 변수들은 모두 메모리 상에 존재한다고 가정하였을 경우 계산 시간을 다음과 같다. 참고로, 곱셈에 필요한 클럭(Clock)은 11클럭이며, 덧셈의 경우에는 3클럭이다.

1회의 동력학식을 계산하기 위한 소요 시간은 표 2와 같다. 1회 계산 소요 시간이 매우 짧은 것을 알 수 있다. 하지만, 비선형적인 동력학식을 만족하는 변수들의 값을 찾기 위해서 최적화 도구인 SQP는 반복적으로 동력학식을 계산한다. 따라서, 초기 변수 입력 후 계산된 동력학식을 줄이기 위해서는 소요되는 계산시간은 얼마나 동력학식을 반복적으로 계산하였는가에 달려있다. 또한, 입력 변수의 차원에 따라서 계산 시간은 기하 급수 적으로 증가하게 된다. 실험적으로 프레임의 수가 12이고 자유도가 11인 경우, SQP는 동력학식을 1회 감소하기 위하여 약 550번의 연산을 수행한다. 따라서, 초기 입력으로 사용되는 변수들의 값이 계산 시간과 동력학식의 수렴을 좌우한다. 3장에서 소개한 근육변수의 경우 정확한 초기값을 사용자가 알 수가 없기 때문에 주어진 모션 데이터를 기반으로 근육변수들의 초기값을 찾아야 한다.

표 2. 계산 소요 클럭수(Pentium 100MHz 사용시).  
Table 2. Number of clocks of calculation(Using Pentium 100MHz)

자유도	곱셈	덧셈	합계	소요 시간
11	118382클럭	23808클럭	142190클럭	0.0014s
20	223025클럭	44652클럭	267677클럭	0.0027s
36	409057클럭	81708클럭	490765클럭	0.0049s

V. 계산 시간 단축을 위한 제안

본 장에서는 계산 시간 단축을 위한 방법을 제안한다. 본 연구에서 사용한 기법을 소개하고, 추후로 시도할 방법에 대해서도 논한다.

1. 병렬화 기법

4장에서 언급한 계산 시간은 1개의 프로세서(processor)를 사용했을 경우를 고려한 것이다. 만약  $p$ 개의 프로세서를 사용한다면, 계산 시간은 이론적으로  $\frac{1}{p}$ 만큼 감소하게 된다. 본 연구에서는 병렬화 기법을 사용하기 위하여 슈퍼컴퓨터를 사용하였으며, 슈퍼컴퓨터에서 제공하는 병렬화 도구 중 OpenMP를 사용하였다. OpenMP는 사용이 간편하기 때문에 초보자도 쉽게 사용할 수 있는 장점을 갖고 있다. 우리는 일반 PC에서 1개의 프로세서가 수행하던 일을 슈퍼컴퓨터에서 4개의 프로세서가 수행하도록 하였다. 하지만, 본 연구에서 다루는 계산식의 대부분이  $4 \times 4$  행렬을 사용하고, 같은 식을 반복하여 계산하는 구조이기 때문에 반복적으로 되풀이되는 계산 과정에서 계산 시간보다 병렬화 시키는 과정이 오히려 overhead로 작용했다.

2. 프레임 나누기

총 프레임을  $f$ 개로 나누어 계산하는 방식이다. 프레임의 수가 적어지면 최적화 해야할 변수의 차원도 줄어들기 때문에 계산 시간을 기하 급수적으로 감소하는 장점을 갖고 있다. 하지만, 프레임을 어떻게 나눠야 하며, 나뉜 프레임 간의 불연속이 발생할 수 있기 때문에 이점에 대한 연구가 필요하다.

3. 표(Table) 방식

최적화 과정에서 가장 많은 시간을 차지하는 동력학식을 계산하지 말고 최적화 과정 전에 미리 계산하여 표로 만든 후, 최적화 과정에서 필요시 표를 참조하는 방식이다. 물론 표는 동작의 종류에 따라서 달라져야 하며 보간 방법을 사용하여 표에 생략된 값들을 계산해야 한다. 이 방법은 메모리상의 변수를 읽어 들이는 방법이기 때문에 계산 시간을 현저히 줄일 수 있다. 하지만, 동작의 종류마다 표가 달라져야 하며, 너무 방대한 표를 만들어야 하기 때문에 오히려 최적화 준비과정에서 많은 시간을 소비하게 된다. 또한, 보간 방법에 따라서 결과가 달라질 수 있다.

VI. 시스템 구성

본 연구는 모션 트랜스포메이션 시스템 시험 구현을 위하여 상용 애니메이션 프로그램인 3D MAX STUDIO(이하 3D MAX)를 사용하여 모션 트랜스포메이션 시스템을 구성하였다. 3D MAX에서는 애니메이션을 만들기 위하여 다양한 파일 포맷을 지원하는데, 이 중 BVH(Biovision Hierarchy) 파일을 이용하였다. 그림 4는 전체 시스템 흐름을 표시하고 있다. 본 장에서는 현재 구성된 시스템에 대하여 설명한다.

1. BVH 파일 및 파일 변환 프로그램

3D MAX에서 지원하는 애니메이션을 위한 포맷 중 하나이다. BVH파일은 크게 두 부분으로 구성되는데, 인체 변수들을 정의하는 부분과 총 프레임 개수와 프레임별 관절 값을 정의하는 부분이다. 우리는 BVH 파일의 프레임별 관절

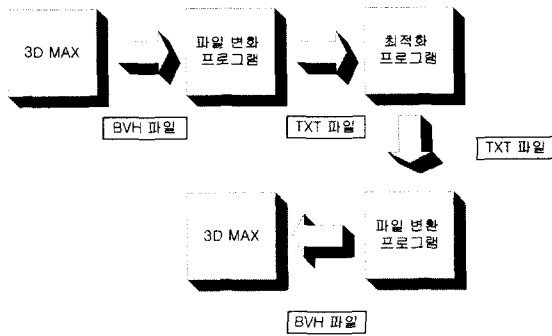


그림 4. 시스템 흐름도.  
Fig. 4. Program flow of developed system.

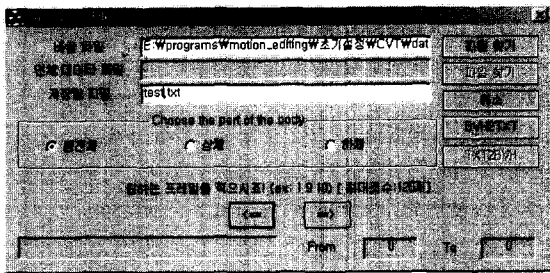


그림 5. 파일 변환 프로그램.  
Fig. 5. File conversion program.

값만을 추출하여 모션 트랜스포메이션을 위한 최적화 프로그램에 사용하였다. 그림 5는 파일 변환 프로그램의 모습이다. BVH 파일 포맷을 텍스트 파일 포맷으로 변환하고, 역으로 텍스트 파일 포맷을 BVH 파일 포맷으로 변환한다.

2. 최적화 프로그램

본 연구에서 사용한 프로그램은 AEM design[36] 사의 CFSQP(Feasible Sequential Quadratic Programming)이다. 이 프로그램을 사용하는 과정을 그림 6에 표시하였다. 사용자는 구할 변수의 초기 값과 변수들의 범위를 설정한다. 그 후, 최적화시킬 목적 함수와 구속 조건을 정하면 CFSQP는 구속 조건을 만족하는 변수들을 찾는다.



그림 6. 최적화 과정.  
Fig. 6. Process of optimization.



그림 7. 3D MAX상의 Biped.  
Fig. 7. A biped on the 3D MAX.

3. 3D-MAX

애니메이션을 만들거나 편집하기 위한 상용 프로그램으로 다양한 파일 포맷을 지원한다. 이 중 Biped는 인간의 애니메이션을 제작하는데 유용하게 쓰인다. 그림 7은 biped의 모습으로 인체의 골격만을 표시한 것을 알 수 있다. 사용자는 인체의 골격을 구성하고, BVH 파일을 이용하여 애니메이션을 만든다. 애니메이션이 완성되면, 사용자는 자신이 디자인한 캐릭터를 biped에 적용하여 애니메이션을 완성할 수 있다.

VII. 결론 및 향후 연구방향

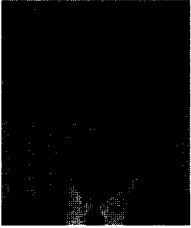
지금까지 우리는 모션 트랜스포메이션의 정의와 수행과정에 대하여 알아보았다. 또한, 일반적으로 모션 트랜스포메이션에 사용되는 여러 기법 중 최적화 도구를 사용한 기법에 대하여 설명했다. 최적화 기법을 사용할 경우 구속조건을 만족하는 여러 해 중에 목적 함수를 만족하는 해를 찾기 때문에 최적화 과정을 거쳐 얻어진 결과물은 사용자가 원하는 결과를 그대로 반영한다. 하지만, 최적화 과정에 사용되는 구속 조건 중 동력학식은 매우 비 선형적이고, 계산량이 많기 때문에 수렴시간이 길어지는 문제점이 있다. 본 연구는 문제가 되는 동력학식의 계산량을 실수연산 개수라는 측면에서 또 계산 시간이라는 측면에서 알아보았고, 이 문제점을 해결하기 위한 몇가지 방법을 제안하였다. 향후 연구과제로는 본 논문에서 제안한 방법을 사용하여 계산시간을 줄이는 여러 가지 방법을 고안해내는 일과 상용 가능한 모션 에디팅 시스템의 개발 등을 들 수 있다. 그리고, 향후 연구방향은 본 논문에서 제안한 프레임 나누기 기법을 사용한 모션 트랜스포메이션과정을 연구할 것이다.

참고문헌

[1] 강호석, "동작 캡처 기술," 전자공학회지. vol. 25, no. 2. 1998.  
 [2] Zoran Popovic and Andy Witkin, "Physically based motion transformation," *Computer Graphics (SIGGRAPH)* 1999.  
 [3] Norman Badler, Kamran Manoocherhri, and Graham Walters, "Articulated figure positioning by multiple constraints," *IEEE Computer Graphics and Applications*, pp. 28-38, June,

- 1987.
- [4] Andrew Witkin, Kurt Fleischer, and Alan Barr, "Energy constraints on parameterized models," *Computer Graphics*, vol. 21, no. 4, pp. 225-232, July, 1987.
  - [5] Chris Welman, "Inverse kinematics and geometric constraints for articulated figure manipulation," Master's thesis, Simon Fraser University, September, 1993.
  - [6] Jianmin Zhao and Norman Badler, "Inverse kinematics positioning using nonlinear programming for highly articulated figures," *ACM Transactions on Graphics*, vol. 13, no. 4, pp. 313-336, October, 1994.
  - [7] W. T. Reeves, "Particle systems-a technique for modeling a class of fuzzy objects," *ACM Transaction on Graphics*, vol. 2, pp. 91-108, April, 1983.
  - [8] David Baraff, "Coping with friction for non-penetrating rigid body simulation," *Computer Graphics(proc.SIGGRAPH)*, vol. 25, pp. 31-40, July, 1991.
  - [9] William Armstrong and Mark Green, "The dynamics of articulated rigid bodies for purpose of animation," *The Visual Computer*, vol. 1, pp. 231-240, 1985.
  - [10] Jane Wilhelms, "Using dynamic analysis for realistic animation of articulated bodies," *IEEE Computer Graphics and Applications*, pp. 12-27, June, 1987.
  - [11] Paul Issacs and Michael Cohen, "Controlling dynamics simulation with kinematics constraints, behavior function and inverse dynamics," *Computer Graphics(Proc. SIGGRAPH)*, vol. 21, no. 4, pp. 215-224, 1987.
  - [12] Hyeongseok Ko and Norman I. Badler, "Animating human locomotion with inverse dynamics," *IEEE Computer Graphics and Applications*, vol. 16, no. 2, pp. 50-59, March, 1996.
  - [13] Ronen Barzel and Alan H. Barr, "A modeling system based on dynamic constraints," *Computer Graphics(Proc. SIGGRAPH)*, vol. 22, pp. 179-188, 1988.
  - [14] John Platt, "A generalization of dynamic constraints," *CGVIP: Graphical models and Image Proceeding*, vol. 54, no. 6, pp. 516-525, November, 1992.
  - [15] Andrew Witkin and Michael Kass, "Spacetime constraints," In John Dill, editor, *Computer Graphics(Proc. SIGGRAPH)*, vol. 22, pp. 159-169, August, 1988.
  - [16] Lynne Shapiro Brotman and Arun N. Netravali, "Motion interpolation by optimal control," In John Dill, editor, *Computer Graphics (Proc. SIGGRAPH)*, vol. 22, pp. 309-315, August, 1988.
  - [17] J. Thomas Ngo and Joe Marks, "Spacetime constraints revisited," In James Kajiya, editor, *Computer Graphics(Proc. SIGGRAPH)*, vol. 27, pp. 343-350, August, 1993.
  - [18] Michiel van de Panne and Eugene Fiume, "Sensor-actuator networks," In James Kajiya, editor, *Computer Graphics (Proc. SIGGRAPH)*, vol. 27, pp. 335-342, August, 1993.
  - [19] Armin Bruderlin and Lance Williams, "Motion signal processing," In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings, Annual Conference Series*, pp. 97-104, August, 1995.
  - [20] Andrew Witkin and Zoran Popovic, "Motion warping," In Robert Cook, editor, *SIGGRAPH95 Conference Proceedings, Annual Conference series*, pp. 105-108, August, 1995.
  - [21] Michael Gleicher and Peter Litwinowicz, "Constraint-based motion adaptation," *Journal of Visuaization and Computer Animation*, to peter.
  - [22] Michael Gleicher, "Motion editing with spacetime constraints," In Michael Cohen and David Zelzer, editor, *Proceedings 1997. Symposium on interactive 3D Graphics*, pp. 139-148, April, 1997.
  - [23] Bolic R, Thalmann D., "Combined direct and inverse kinematic control for articulated figure motion editing," *Computer Graphics Forum*, vol. 11, no. 4, pp. 189-202, 1992.
  - [24] Jessica Hodgins and Nancy Pollard, "Adapting simulated behaviors for new characters," In Turner Whitted, editor, *Computer Graphics (Proc. SIGGRAPH)*, pp. 153-162, August, 1997.
  - [25] Michael Gleicher, "Retargetting motion to new character," *SIGGRAPH 98 Conference Proceedings, Annual Conference Series*, pp. 33-42. Addison-Wesley: Reading, MA, July, 1998.
  - [26] Munetoshi Unuma, Ken Anjyo, and Ryoza Takeuchi, "Fourier principles for emotion-based human figure animation," In Robert Cook, editor, *Computer Graphics (Proc. SIGGRAPH)*, pp. 91-96, August, 1995. ACM-0-89791-701-4.
  - [27] Ken Perlin, "Real time responsive animation with personality," *IEEE Transactions on Visualization and Computer Graphic*, pp. 5-15, March, 1995.
  - [28] Seyoon Tak and Hyeongseok Ko, "Example Guided Inverse Kinematics," *In the Proceedings of the International Conference on Computer Graphics and Imaging (CGIM) 2000*, Las Vegas, Nevada, USA, pp. 19-23, November, 2000.
  - [29] Kwangjin Choi, Sanghyun Park, and Hyeongseok Ko, "Processing motion capture data to achieve positional accuracy," *The Journal of Graphical Models and Image Processing*, pp. 260-273, 1999.
  - [30] J. Lee and S. Y. Shin, "Multiresolution motion analysis with application," *International Workshop on Human modelling and Animation*, pp. 131-143, 2000.
  - [31] Kwangjin Choi, Hyeongseok Ko, "Online Motion Retargetting," *The Journal of Visualization and Computer Animation*, vol. 11, Issue 5, 2000
  - [32] D. E. Whitney, "Resolved motion rate control of manipulators and human prostheses," *IEEE Trans. Man-Machine Sys.*, MMS-10, pp. 47-53, 1969
  - [33] Jehhee Lee and Sungyong Shin, "A hierarchical approach to interactive motion editing for human-like figures," *ACM SIGGRAPH '99*, 1999.
  - [34] Seyoon Tak, Oh-young Song, and Hyeongseok Ko, "Motion Balance Filtering," *EUROGRAPHICS 2000*, Interlaken, Switzerland, August 21-25, 2000. *Computer Graphics Forum*, vol. 19, no. 3, 2000.
  - [35] S. H. Shin and S. Y. Shin, "Real-time human body deformation based on rotation angle linterpolation," *Proc. KISS Annual Conference*. Korea, 2000
  - [36] AEM design homepage, "<http://gachinese.com/aemdesign/>".





### 이 지 흥

1983년 서울대 전자공학과 졸업. 1985년 한국과학기술원 진지 및 전자공학과 석사. 1991년 한국과학기술원 전기 및 전자공학과 박사. 2001년~현재 충남대학교 메카트로닉스공학과 교수. 관심분야는 로봇틱스, 인공지능.



### 김 인 식

2000년 충남대학교 메카트로닉스공학과 졸업. 2000~현재 충남대 메카트로닉스공학과 석사 과정. 관심분야는 로봇틱스, 인공지능.

### 김 성 수

2001년 경일대학교 제어계측공학과 졸업. 2001~현재 충남대학교 메카트로닉스공학과 석사 과정. 관심분야는 로봇틱스, 인공지능.