

삼각형 메쉬로 이루어진 3D 모델의 변형을 위한 IK 계산 가속화

박현아^o 강다은 권태수

한양대학교 일반대학원 컴퓨터소프트웨어학과

{qkrusdk0303, teddysiah, taesoo}@hanyang.ac.kr

An Accelerated IK Solver for Deformation of 3D Models with Triangular Meshes

Hyunah Park^o Daeun Kang Taesoo Kwon

Department of Computer Science, Hanyang University Graduate School, South Korea

요약

본 연구는 골격이 있고 삼각형 메쉬로 이루어진 3D 모델의 변형을 빠른 연산 속도로 구현하는 것을 목표로 한다. 이를 위해 삼각형 메쉬 정점의 위치를 빠른 속도로 계산할 수 있는 IK 풀이 방법을 연구하고 해당 인터페이스를 개발하였다. 모델 표면 상에 한 개 이상의 마커를 지정하고 마커의 목표 위치를 설정하면, 이 시스템은 마커의 목표 위치를 기준으로 가속화된 IK 풀이를 통해 모델 표면을 구성하는 삼각형 메쉬 정점의 위치를 계산한다. 메쉬의 위치를 결정하는 데에는 각 마커와 해당 마커에 영향을 미치는 관절, 그리고 해당 관절의 상위(부모) 관절에 대하여 계산을 수행하는데, 이 과정에서 빈번하게 사용되는 중복된 항(terms)이 발생한다. 이러한 중복항을 사전에 계산해 둬으로써 기존의 삼중 중첩 반복 구조의 계산 절차를 이중 중첩 반복 구조로 개선하여 모델 변형 결과를 신속하게 구현할 수 있다. 제안된 가속화된 IK 풀이 방법은 LBS 기법으로 구현된 3D 모델을 다루거나 마커 없이 단순 촬영만으로 대상 물체를 추적하는 무마커 추적 관련 연구 등 다양한 분야에서 유용하게 활용할 수 있다.

Abstract

The purpose of our research is to efficiently deform a 3D models which is composed of a triangular mesh and a skeleton. We designed a novel inverse kinematics (IK) solver that calculates the updated positions of mesh vertices with fewer computing operations. Through our user interface, one or more markers are selected on the surface of the model and their target positions are set, then the system updates the positions of surface vertices to construct a deformed model. The IK solving process for updating vertex positions includes many computations for obtaining transformations of the markers, their affecting joints, and their parent joints. Many of these computations are often redundant. We precompute those redundant terms in advance so that the 3-nested loop computation structure was improved to a 2-nested loop structure, and thus the computation time for a deformation is greatly reduced. This novel IK solver can be adopted for efficient performance in various research fields, such as handling 3D models implemented by LBS method, or object tracking without any markers.

키워드: 삼각메쉬모델, 가속화된 IK 풀이, 모델 변형, 선형 블렌드 스킨닝, 무마커 트래킹

Keywords: triangular meshed model, accelerated IK solver, model deformation, linear blend skinning, markerless tracking

1. 서론

컴퓨터 그래픽스, 특히 컴퓨터 비전 및 애니메이션 분야에서 사용되는 3D 모델은 메쉬(mesh)로 이루어져 있다. 모델의 동작(motion)이나 외부 영향에 의해 모델의 모양이 변형(deformation)되

면 영향을 받는 모든 메쉬에 변화가 일어나는데, 이는 상당한 계산량을 요구한다.

본 연구는 골격(skeletons)이 있고 삼각형 메쉬(triangular meshes)로 이루어진 3D 모델의 변형을 빠른 연산 속도로 구현

*corresponding author: Taesoo Kwon/Hanyang University(taesoo@hanyang.ac.kr)

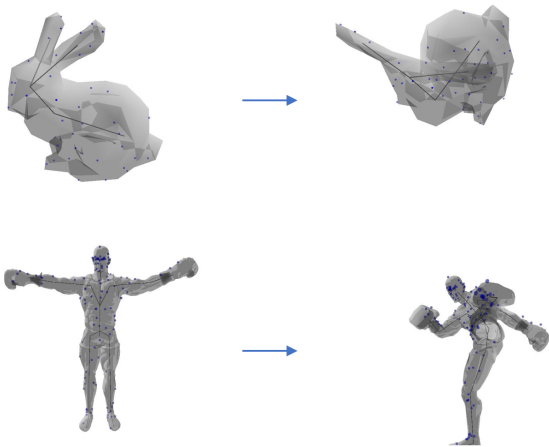


Figure 1: Sample results from our work. Our novel accelerated IK solver deforms a 3D model in high speed based on the target positions of markers on the model surface.

하는 것을 목표로 한다. 이를 위해 삼각형 메쉬 정점의 위치를 빠른 속도로 계산할 수 있는 역기구학(inverse kinematics, IK) 풀이 방법을 연구하고 해당 인터페이스를 개발하였다. 모델 표면에 한 개 이상의 마커(markers)를 지정하고 마커의 목표 위치를 설정하면, 이 시스템은 마커의 목표 위치를 기준으로 가속화된 IK 풀이를 통해 모델 표면을 구성하는 삼각형 메쉬 정점의 위치를 계산한다(Figure 1 참조). 메쉬의 위치를 결정하는 데에는 각 마커와 해당 마커에 영향을 미치는 관절(joints), 그리고 해당 관절의 상위(부모) 관절에 대하여 연산을 수행하는데, 이 과정에서 빈번하게 사용되는 중복된 항(terms)이 발생한다. 이러한 중복항을 사전에 계산해 둬으로써 비교적 빠른 속도로 메쉬 위치 계산을 진행하며, 모델의 변형된 모습을 신속하게 결과 화면으로 확인할 수 있다.

본 연구는 3D 모델 변형에 필요한 IK 풀이를 가속화된 방법으로 연산 속도를 개선했다는 데에 의의가 있다. 기존의 삼중 중첩 반복(3-nested loop) 구조의 계산 절차와 달리, 중복항을 사전에 계산함으로써 이중 중첩 반복(2-nested loop) 구조의 계산 절차를 수행한다. 이러한 개선된 계산 구조를 통해 5장에서 후술할 내용과 같이 모델 변형 결과를 빠르게 구현할 수 있다.

본 연구에서 제안하는 가속화된 IK 풀이 방법은 3D 모델을 다루는 다양한 분야에서 활용할 수 있다. 예를 들어, 선형 블렌드 스킨닝(linear blend skinning, LBS) 기법으로 구현된 3D 모델을 다루는 데 자연스럽게 빠른 속도의 모델 변형 또는 제어(control)가 가능하다. 또한 사용자와의 상호작용 인터페이스를 접목하여 3D 모델의 편집 도구(editing tool)로도 응용할 수 있다. 가속화된 IK 풀이 방법은 컴퓨터 비전 분야에도 적용할 수 있는데, 특히 마커 없이 단순 촬영만으로 대상 물체를 추적하는 무마커 추적(markerless tracking) 관련 연구에서 유용하게 활용될 수 있다.

본 논문은 총 6장으로 구성되어 있으며, 2장에서는 마커 기반 3D 모델 변형에 관련된 기존 연구를 소개한다. 3장에서는 모델 표면에 마커를 선택하고 관련 메쉬 정보를 저장하는 과정을

서술하고, 4장에서는 기존의 마커 기반 캐릭터 변형에 사용되는 기존의 일반적인 IK 풀이 방법을 소개한다. 5장에서는 해당 가속화된 IK 풀이 방법을 적용한 실험 결과를 그림과 함께 제시하며, 6장에서는 본 연구의 의의와 장단점, 기대효과 등을 서술한다.

2. 관련 연구

이 장에서는 본 연구의 주제와 구현 방식이 밀접하게 관련된 기존 연구를 소개한다. 다양한 관련 연구 중에서도, 특히 본 연구에서 3D 모델 변형을 위한 연산 방식으로써 다루었던 IK 풀이 방법 및 이를 이용한 모델 변형 관련 연구를 소개한다. 또한 마커 기반 모델 변형 기술이 유용하게 활용되는 무마커 추적 관련 연구를 소개하며, 해당 연구의 필요성과 활용방안을 제시하고자 한다.

2.1 IK 풀이를 이용한 모델 변형 관련 연구

IK는 모델의 자세(모양)를 특징적으로 결정짓는 부분인 엔드 이펙터(end-effectors)를 목표 위치로 위치시키기 위해 모델의 관절 각도(joint angle)를 계산하는 운동 방정식(kinematic equations)이다[1]. IK 및 그 풀이 방법을 다루는 연구는 다양한 분야에 널리 수행되어 왔는데, 다양한 야코비 행렬을 이용하여 광범위한 기술 분야에서 적용되기도 한다[2]. 컴퓨터 그래픽스 분야에서 IK는 특히 관절 캐릭터(articulated characters)의 동작(자세)을 제어하는 수단으로 빈번히 채택된다. 또한 관절 캐릭터의 동작 생성 소요시간 단축 및 결과 품질의 향상을 위해, 다양한 IK 최적화 풀이 방법에 관한 연구가 수행되었다[3]. 최근에는 인공지능 기술의 발전에 발맞추어 데이터 기반(data-driven) 동작 생성 방식과 기계학습(machine learning)을 융합함으로써 보다 우수한 품질의 캐릭터 애니메이션 연구가 빈번히 수행되고 있다[4, 5]. IK 풀이를 효율적으로 수행하기 위한 다양한 접근에 관한 연구는 오늘날 까지도 꾸준히 지속되고 있으며[6], 본 논문에서는 야코비 행렬(Jacobian matrix)의 계산을 간소화하여 연산 소요시간을 단축하고자 하였다.

캐릭터 모델, 특히 동작에 따른 모양 변화가 크게 수반되는 연체 캐릭터(soft-bodied characters)와 같은 모델을 제어하는 데에는 많은 연산량을 수반하는 메쉬 모양 변화를 빠르고 자연스럽게 적용(update)하는 것 또한 중요하다. LBS 기법은 골격을 가진 모델의 메쉬를 변형하는 가장 대표적인 알고리즘으로, 캐릭터 관절의 움직임(transformation)에 따라 주변의 다각형 메쉬(polygon mesh)의 정점 위치를 계산하여 메쉬의 모양을 변형시킨다[7]. LBS 기법과 IK 풀이를 적용한 캐릭터 모델 변형 관련 연구는 꾸준히 이루어져 왔다. 메쉬 IK는 예제 메쉬(example meshes)로부터 유의미한 모양 변화 공간(shape space)을 학습하여 모델 변형 연산을 수행하는 방법이다[8], 앞선 메쉬 IK를 응용하여 일부 자유도(degree of freedom, DOF)에 대해서만 연산을 진행하고, 나머지에 대해서는 자동 예측을 통해 모델 변형 연산 속도를 가속화한 시도가 있었다[9], 또한 파충류, 포유류, 곤충, 어류 등

각기 다른 몸체 변형 스타일을 가진 다양한 캐릭터에 두루 적용할 수 있는 빠르고 자연스러운 메쉬 IK 풀이 방법이 연구하기도 하였다[10]. 이외에도 IK 풀이를 통해 모델 변형의 속도 및 변형 결과 품질 성능을 높이기 위한 연구는 다수 진행되어 왔으며, 본 연구 또한 기존 삼중 중첩 반복 계산 구조의 IK 풀이 방법의 연산 속도를 가속화하기 위해 계산식 정리를 시도하고 실험을 반복하였다. 본 연구에서 소개하는 가속화된 IK 풀이 방법은 기존 풀이 연산에서 빈번하게 사용되는 중복항을 사전에 계산함으로써 삼중 중첩 반복 구조를 이중 중첩 반복 구조로 축약하여 연산 속도를 높였다.

2.2 무마커 추적 관련 연구

모션 캡처(motion capture, mocap)는 실제 물체나 생물체를 대상으로 그 움직임이나 변화를 추적하는 것으로, 현실에서의 대상 물체의 움직임을 가상 모델의 데이터 변화 형태로 기록하는 것이다. 가장 일반적인 모션 캡처는 마커 기반 추적(marker-based tracking)으로, 관측 대상에 추적 가능한 소형 기기(마커)를 부착하여 마커의 위치 변화를 감지 및 기록한다. 이는 마커를 직접 부착하므로 정확도가 일정 수준 보장되지만, 마커를 부착함으로써 물체의 변화 가능성에 영향을 주어 자연스러운 관측이 어려운 경우도 있으며, 물체의 변화에 따라 마커가 가려지거나 관측 카메라 기준으로 마커의 위치가 중첩될 경우에는 그 정확성이 떨어질 수도 있다. 또한 마커 부착이 실질적으로 불가능한 물체는 실험 대상으로 취급하기 어렵다는 단점이 있다. 무마커 추적은 카메라에 관측되는 단편적인 이미지(images)만으로 그 모양을 추적하는 것으로, 마커 기반 추적 기술의 단점을 보완할 수 있는 가능성이 있다[11]. Microsoft사의 Kinect와 Intel사의 RealSense는 대표적인 무마커 모션 캡처 카메라로, 저가형으로 제작되어 사용자의 자세 및 동작을 추적하여 가벼운 오락 및 체형 콘텐츠를 즐길 수 있도록 널리 보급되어 있다[12, 13]. Simi Shape 와 같은 고차원 무마커 모션 캡처 장비 또한 다양한 연구 분야에서 활용되고 있으며[14], 기계학습을 접목하여 여러 사람들의 신체부위를 실시간으로 추적하는 OpenPose, DensePose 기술 또한 널리 응용되고 있다[15, 13]. 무마커 추적 기술은 다양한 분야에서 널리 사용되고 있으며, 그 필요성이 대두됨에 따라 무마커 추적의 정확도 및 속도 성능을 높이기 위한 연구 또한 다양하게 진행되고 있다[16]. 본 연구는 삼각형 폴리곤(polygon)으로 이루어진 3D 메쉬 모델의 표면에 마커를 생성하고, 마커의 목표 위치를 기반으로 모델 변형을 신속하게 연산하고자 한다. 해당 기술을 응용하여 3D 모델의 편집 도구로 활용할 수 있고, 마커 기반 또는 무마커 추적 결과를 빠르게 보정하는 데에 적용할 수 있다.

3. 마커 선택 및 메쉬 정보 저장

이 장에서는 삼각형 메쉬로 이루어진 3D 모델의 표면에 임의의 점을 마커로 선택하고, 마커의 목표 위치를 기반으로 IK 풀이를

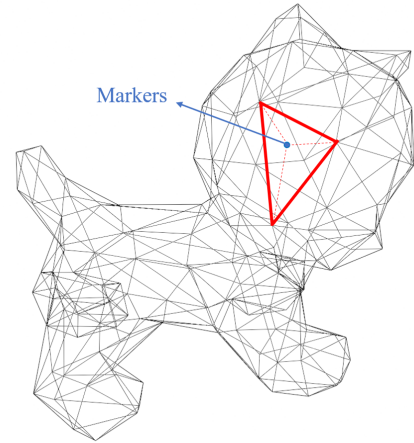


Figure 2: To approximate marker-less mocap applications, we selecting markers only on visible surfaces. For a visible triangle, markers can be generated at random locations or manually by the user.

수행하기 위해 마커를 포함하는 삼각형 메쉬에 관한 정보를 저장하는 과정을 간략히 소개한다. 본 연구에서는 표면이 삼각형 폴리곤으로 이루어진 3D 모델의 변형을 다루는데, 삼각형은 3D 모델을 구성할 수 있는 가장 단순한 폴리곤으로 모델의 모양 변화가 비교적 자유롭고 계산 또한 용이하다.

마커는 사용자의 모델 관측 시점(viewport) 기준으로 가시적인 모델 표면에 생성하는 것으로, 표면을 구성하는 삼각형 메쉬 내부의 임의의 한 점을 마커로서 선택할 수 있다(Figure 2 참조). 사용자가 한 점을 마커로 선택하기 위해, 관측 시점으로 부터 모델을 향하는 카메라의 초점에 대한 벡터와 폴리곤의 법선 벡터(normal vector)의 내적(dot product)을 통해 부호를 비교해서 가시적인 마커를 포함할 수 있는 폴리곤으로 인식한다. 이 때 마커를 포함한 삼각형 폴리곤의 세 정점(vertices)의 지역 좌표계(local coordinate) 및 해당 좌표계 기준의 지역 위치(local position), 세 정점의 위치가 마커의 위치에 미치는 영향력을 반영하는 가중치(weights)를 저장한다. 저장된 폴리곤 메쉬의 정보는 마커의 목표 위치를 기반으로 IK 풀이 연산을 수행할 때 사용된다.

Figure 2는 삼각형 메쉬로 이루어진 모델 표면에 임의의 점을 마커로 선택하고, 해당 마커를 포함하는 삼각형 폴리곤이 결정된 예를 보여준다. 마커는 모델 관측 시점 기준으로 가시적인 표면 메쉬 부분에 생성하며, 각 삼각형 폴리곤마다 한 개의 마커를 생성할 수 있다. 따라서 사용자의 모델 관측 시점에 따라 가시적인 메쉬가 달라지면서 마커의 생성 가능한 범위 및 최대 개수 또한 달라질 수 있다.

Figure 3은 삼각형 메쉬로 이루어진 고양이 모양의 3D 모델을 사용하여 모델 표면 상에 생성 가능한 모든 마커를 생성한 결과를 보여준다. 관측 시점으로부터 관측되는 모든 표면 폴리곤마다 각각 한 개씩의 마커를 생성한 것으로, 모델 변형 기준으로써 그림에 표시된 것과 같이 모든 마커를 사용할 수도 있고 일부 마커만을

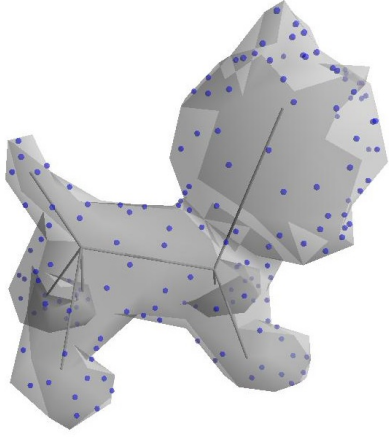


Figure 3: The result of the marker generation process. The sample area and the number of markers can be different depending on the viewport. For model deformation, one or more markers can be used as positional constraints.

선택하여 모델 변형을 구현할 수도 있다. 또한 그림에 제시된 시점과 달리 사용자의 의도에 맞게 관측 시점을 바꾸어 모델의 다른 신체부위에 마커를 생성할 수도 있다. 마커는 위치 제약조건(position constraints)의 역할을 하므로 마커의 개수에 따라 모델 변형의 결과가 달라지는데, 마커의 개수가 많아질수록 비교적 절제된 변형이 구현된다(5.1절 참조).

4. 가속화된 IK 풀이

이 장에서는 마커의 목표 위치를 기준으로 모델 표면상의 삼각형 메쉬를 변형하기 위해 그 위치를 계산하는 IK 풀이 방법을 소개한다. 기존의 풀이 방법을 소개하고, 연산 속도의 성능 향상을 위해 기존 방법으로부터 가속화된 IK 풀이 방법을 유도하는 과정을 설명한다.

4.1 IK 풀이의 목적 함수

마커의 목표 위치는 모델을 변형하는 데 작용하는 위치 제약조건의 역할을 한다. 일반적으로 해당 제약조건에 따라 모델을 변형하려면 목적 함수(objective function)와 그 편미분식(gradient)을 다음과 같이 정의한다(수식 1, 2 참조).

$$f_k(\mathbf{q}) = \|\mathbf{x}_k(\mathbf{q}) - \mathbf{x}'_k\|_2^2 \quad (1)$$

$$\mathbf{g}_k(\mathbf{q}) = \frac{\partial f_k(\mathbf{q})}{\partial \mathbf{q}} = 2(\mathbf{x}_k(\mathbf{q}) - \mathbf{x}'_k)^\top \frac{\partial \mathbf{x}_k(\mathbf{q})}{\partial \mathbf{q}} \quad (2)$$

수식 1 및 2에서 \mathbf{q} 는 3D 모델의 특정 자세(모양새)를 가리키며, $f_k(\mathbf{q})$ 는 현재 자세 \mathbf{q} 의 표면에 존재하는 k 번째 마커의 목표 위치에 대한 목적 함수이다. 이 때 \mathbf{x}_k 또는 $\mathbf{x}_k(\mathbf{q})$ 는 k 번째 마커의 현재 위치, \mathbf{x}'_k 는 목표 위치를 가리킨다. 수식 1의 목적 함수는

마커의 현재 위치와 목표 위치 간의 오차(difference)를 계산하여 그 값을 최소화하는 방향으로 모델을 변형시키며, 수식 2는 목적 함수를 편미분한 것이다.

k 번째 마커의 위치값 $\mathbf{x}_k(\mathbf{q})$ 은 모델 내부에 존재하는 관절(joints)의 값을 이용하여 다음과 같이 표현할 수 있다(수식 3, 4 참조).

$$\mathbf{x}_k(\mathbf{q}) = \sum_i w_k^i \mathbf{T}_i(\mathbf{q}) \mathbf{v}_k^i \quad (3)$$

$$\mathbf{x}_k(\mathbf{q}) = \sum_i w_k^i (\mathbf{R}_i \mathbf{v}_k^i + \mathbf{p}_i) \quad (4)$$

수식 3에서 \mathbf{v}_k^i 는 k 번째 마커의 위치에 영향을 주는 i 번째 관절 기준의 지역 좌표계 상으로 표현된 해당 마커의 위치를 가리키고, \mathbf{T}_i 는 관절의 전역 좌표계를 표현하는 행렬, w_k^i 는 i 번째 관절값이 k 번째 마커 위치에 영향을 주는 정도를 나타내는 가중치를 나타낸다. 이 때 좌표계 \mathbf{T}_i 는 회전(rotation) 행렬 \mathbf{R}_i 와 이동(translation) 요소 \mathbf{p}_i 로 구성되어 있으며($\mathbf{T}_i = (\mathbf{R}_i, \mathbf{p}_i)$), 따라서 수식 3는 수식 4로 바꾸어 쓸 수 있다.

또한 목적함수의 편미분인 수식 2는 다음과 같이 다시 정리할 수 있다(수식 5 참조).

$$\mathbf{g}_k(\mathbf{q}) = \frac{\partial f_k(\mathbf{q})}{\partial \mathbf{q}} = \sum_i 2w_k^i (\mathbf{x}_k(\mathbf{q}) - \mathbf{x}'_k)^\top \mathbf{J}_i \quad (5)$$

수식 5에서 \mathbf{J}_i 는 편의상 기존 수식 2의 $\frac{\partial \mathbf{x}_k(\mathbf{q})}{\partial \mathbf{q}}$ 을 치환한 것으로, \mathbf{J}_i 은 3개 행 및 (관절의 수 \times 각 관절의 자유도(degree of freedom, DOF))개의 열로 이루어진 야코비 행렬의 형태를 지닌다. 행렬의 각 열(columns)은 관절의 회전축 벡터 \mathbf{a} 및 마커의 현재 위치 \mathbf{x} 와 관절의 현재 위치 \mathbf{s} 간의 오차의 외적, 즉 $\mathbf{a} \times (\mathbf{x} - \mathbf{s})$ 형태로 이루어져 있다. 예를 들어 i 번째 관절에 대해서 \mathbf{J}_i 의 l 번째 열은 $\mathbf{J}_i[l] = \mathbf{a}_i \times (\mathbf{x}_k^i - \mathbf{s}_i)$ 이다. 여기서 마커 위치 \mathbf{x}_k^i 는 다음과 같이 표현할 수 있다(수식 6 참조).

$$\mathbf{x}_k^i = \mathbf{R}_i \mathbf{v}_k^i + \mathbf{p}_i \quad (6)$$

이 때 \mathbf{a} , \mathbf{x} , \mathbf{s} 는 모두 전역 좌표계 기준으로 표현된 값으로, 사전에 계산된 값이다.

앞서 제시한 목적 함수 및 편미분을 그대로 연산을 수행하는 것은, 다음과 같은 세 가지 요소에 대하여 동시에 계산하는 것이다.

- 각 마커 k 에 대하여
- 마커의 위치를 정의하는 각 관절 i 에 대하여
- 해당 관절의 그 부모 관절에 대하여 (\mathbf{J}_i 의 0이 아닌 컬럼(non-zero columns)에 대한 연산)

계산식은 삼중 중첩 반복 계산 구조로 구현되는데, 이러한 중첩 반복은 계산량이 방대하여 실제 계산을 수행하는 데 소요시간의 부담이 크다. 다음 절에서는 이러한 목적 함수와 편미분의 연산을

기존의 IK 풀이 방법으로 수행하는 방법을 서술하고, 해당 연산을 가속화시키는 아이디어를 소개한다.

4.2 기존 IK 풀이 방법

4.1절에서 정의한 수식을 풀이하는 데 앞서 기존의 목적 함수(수식 1)와 편미분식(수식 5)을 편의상 다음과 같이 요약하였다(수식 7, 8 참조).

$$f_k(\mathbf{q}) = \|\mathbf{x}_k(\mathbf{q}) - \mathbf{x}'_k\|_2^2 = \Delta S_k \cdot \Delta S_k \quad (7)$$

$$\mathbf{g}_k(\mathbf{q}) = \sum_i 2w_k^i (\mathbf{x}_k(\mathbf{q}) - \mathbf{x}'_k)^T \mathbf{J}_i = \sum_i \delta_k \mathbf{J}_i \quad (8)$$

수식 7에서 ΔS_k 는 $\Delta S_k = \mathbf{x}_k(\mathbf{q}) - \mathbf{x}'_k$ 를 축약한 것이고, 수식 8에서 δ_k 는 $\delta_k = 2w_k^i (\mathbf{x}_k(\mathbf{q}) - \mathbf{x}'_k) = 2w_k^i \Delta S$ 를 축약한 것이다.

이러한 목적함수와 편미분식의 IK 풀이를 위해서는 다양한 방법이 있는데, 가장 단순한 방법은 수식을 있는 그대로 계산하는 것이다. 편미분식의 경우, 모든 마커에 대하여 계산을 수행할 때 각 마커에 대한 계산값을 순차적으로 더한다(수식 9 참조).

$$\mathbf{g}_k(\mathbf{q})[l] += \delta_k \cdot (\mathbf{a}_l \times (\mathbf{x}_k^i - \mathbf{s}_l)) \quad (9)$$

수식 9는 삼중 중첩 반복 구조 내에서 계산되고, 각 마커에 대한 연산을 반복할 때마다 $\delta \cdot (\mathbf{a} \times (\mathbf{x} - \mathbf{s}))$ 값을 계산하므로 많은 연산량(소요시간)을 요구한다.

4.3 가속화된 IK 풀이 방법

이 절에서는 앞서 서술한 기존의 IK 풀이 연산식을 보다 간략하게 정리하여 삼중 중첩 반복 구조의 연산 과정을 이중 중첩 반복 구조로 재구성하는 아이디어를 순차적으로 서술한다.

앞선 수식 6에서 마커의 위치를 $\mathbf{x}_k^i = \mathbf{R}_i \mathbf{v}_k^i + \mathbf{p}_i$ 와 같이 정의하였고, 야코비 행렬 \mathbf{J}_i 의 각 열은 $\mathbf{c} = \mathbf{a} \times (\mathbf{x} - \mathbf{s})$ 의 형태를 지닌다(수식 9 참조).

$$\mathbf{c} = \mathbf{a} \times (\mathbf{x} - \mathbf{s}) = \widehat{\mathbf{a}}(\mathbf{x} - \mathbf{s}) \quad (10)$$

수식 10에서 햇(hat, $\widehat{\cdot}$) 연산자는 두 벡터에 관한 외적식을 행렬 곱으로 표현한다. 예를 들어 임의의 벡터 \vec{a} 와 \vec{b} 에 대하여 $\vec{a} \times \vec{b} = \widehat{a}\vec{b} = -\vec{b} \times \vec{a} = -\widehat{b}\vec{a}$ 가 성립한다. 여기서 햇 변수는 3×3 크기의 반대칭 행렬(skew-symmetric matrix)의 형식을 갖추고 있다.

수식 10에서 현재 위치 \mathbf{x} 에 수식 6을 대입하면, 수식 8의 $\delta_k \mathbf{J}_i$ 은 \mathbf{J}_i 의 각 열에 대하여 다음과 같은 형태의 연산을 수행한다(수식 11 참조).

$$\begin{aligned} \delta_k \mathbf{c}_i &= (\delta_k \widehat{\mathbf{a}}(\mathbf{p}_i - \mathbf{s}_k^i)) + (\delta_k \widehat{\mathbf{a}}(\mathbf{R}_i \mathbf{v}_k^i)) \\ &= (\delta_k \widehat{\mathbf{a}}(\mathbf{p}_i - \mathbf{s}_k^i)) + (\delta_k \widehat{(\mathbf{R}_i \mathbf{v}_k^i)})(-\mathbf{a}) \end{aligned} \quad (11)$$

즉, $\mathbf{g}_k(\mathbf{q})$ 을 계산하기 위해서는 각 마커 k 와 연관된 i 번째 관절, 그리고 야코비 행렬 \mathbf{J} 의 각 열 l 에 대하여 정리된 수식 11을 반복적으로 계산하여 더한다(\sum). 이러한 삼중 중첩 반복 구조로 이루어진 수식을 쉽게 설명하기 위해 인덱스(index) i 와 l 은 고정하고, 마커 인덱스 k 에 대한 반복 구조에 대해서만 살펴해보도록 한다. 가장 안쪽 반복문이 두 번 실행되는 간단한 경우를 가정하면, 편미분식 기여분의 합은 다음과 같이 표현할 수 있다(수식 12 참조).

$$\begin{aligned} &\delta^1 \mathbf{c}^1 + \delta^2 \mathbf{c}^2 \\ &= (\delta^1 \widehat{\mathbf{a}}(\mathbf{p} + \mathbf{R}\mathbf{v}^1 - \mathbf{s})) + (\delta^2 \widehat{\mathbf{a}}(\mathbf{p} + \mathbf{R}\mathbf{v}^2 - \mathbf{s})) \\ &= (\delta^1 + \delta^2) \widehat{\mathbf{a}}(\mathbf{p} - \mathbf{s}) \\ &\quad + ((\delta^1 \widehat{(\mathbf{R}\mathbf{v}^1)}) + (\delta^2 \widehat{(\mathbf{R}\mathbf{v}^2)}))(-\mathbf{a}) \end{aligned} \quad (12)$$

수식 12를 일반화하면, 편미분식의 계산 과정은 다음과 같이 같이 요약할 수 있다(수식 13 참조).

$$\begin{aligned} \mathbf{g}(\mathbf{q})[l] &= \sum_{i,k} \delta_k \mathbf{c}_i^l \\ &= \sum_i \left(\sum_{k \in N_i} \delta_k \right) \widehat{\mathbf{a}}_i(\mathbf{p}_i - \mathbf{s}_l) \\ &\quad + \sum_i \left(\sum_{k \in N_i} \delta_k \widehat{(\mathbf{R}_i \mathbf{v}_k^i)} \right) (-\mathbf{a}_l) \end{aligned} \quad (13)$$

수식 13에서 N_i 는 i 번째 관절과 연관된 마커들의 인덱스의 집합이고, 여기서 초록색 밑줄로 표시된 각 계수 $\sum \delta_k$ 및 $\sum \delta_k \widehat{(\mathbf{R}_i \mathbf{v}_k^i)}$ 항은 사전에 계산이 가능한 연산 내용이다. 전처리(pre-process)된 계산 내용을 이용하여 IK 목적 함수의 편미분식을 계산하면, 각 마커에 대한 반복 연산이 불필요하므로 기존의 삼중 중첩 루프를 두개의 이중 중첩 반복 계산 구조로 단축하는 것이 가능하다. 첫번째 이중 중첩 반복 구조는 k 와 i 에 대한 이중 반복문으로, 전처리 항들을 i 를 인덱스로 하는 배열에 저장한다. 또 두번째 이중 중첩 반복 구조는 i 와 l 에 대한 이중 반복문으로, 수식 13에 대한 연산을 수행하는 것에 해당한다.

앞서 4.2절에서 소개한 기존의 IK 풀이 방법은 삼중 중첩 반복 구조로 연산을 수행하므로 연산량이 방대하고 모델 변형 소요시간의 부담도 크다. 본 연구는 기존의 IK 풀이 수식들을 정리하여 빈번히 중복되는 항을 발견하고 사전에 계산함으로써, 목적 함수의 편미분식을 연산하는 과정에서 각 마커에 대한 반복 계산을 줄였다. 결과적으로 전반적인 IK 연산 과정을 이중 중첩 반복 구조로 단축함으로써 모델 변형을 신속하게 구현할 수 있었다. 가속화된 IK 풀이 방법을 통한 모델 변형 결과와 연산 소요시간의 측정 결과는 5장에서 확인할 수 있다.

Table 1: Statistics for the five models used for our experiments: a cuboid sponge, an iguana, a cat, a bunny, and a human shaped models. All of these models are composed of triangular meshes and skeletons.

Model Type	Skeletons	Vertices	Meshes	Markers
Cuboid sponge	6	654	1040	520
Iguana	18	124	244	136
Cat	16	159	314	158
Bunny	8	184	364	163
Human	26	13806	12764	7526

5. 실험 결과

이 장에서는 본 연구의 3D 모델의 신속한 변형을 위한 가속화된 IK 풀이 방법의 작동을 확인하기 위한 실험내용 및 그 결과를 소개한다. 또한 기존의 삼중 중첩 반복 구조의 계산 방법과 모델 변형 수행 속도를 비교하여 성능을 평가하고자 한다.

실험에는 i7-6700k CPU, 16G RAM, GeForce GTX TITAN X GPU를 장착한 컴퓨터를 사용하였다. 실험에 사용한 삼각형 메시로 이루어진 3D 모델로는 직육면체 스펀지, 이구아나, 고양이, 토끼, 인체 캐릭터 등 다섯 가지를 준비하였다. Table 1은 실험에 사용한 다섯 가지 모델의 구현 정보를 표시한 것으로, 해당 표에서 Skeletons, Vertices, Meshes 열은 각각 모델이 가진 뼈대의 수, 모델 표면을 구성하는 정점 및 삼각형 폴리곤의 수를 나타낸 것이다. 또한 Markers 열은 모델의 표면에 적용할 수 있는 마커의 최대 개수로, 실험 당시의 임의의 관측 시점으로부터 생성할 수 있는 마커의 개수를 센 것이다. 삼각형 폴리곤 한 개당 한 개의 마커를 생성할 수 있으므로, 이는 관측 시점 기준으로 가시적인 표면 폴리곤들의 개수라고도 할 수 있다. 이 때 가속화된 IK 풀이 연산 과정에서 발생하는 중복항의 계산은 전처리 과정으로 사전에 수행되었다.

5.1 모델 변형 실험 결과 화면

실험에 사용된 각각의 다섯 가지 모델은 300 프레임(frames)에 걸쳐 특정 동작을 취하는데, 각 프레임별로 동작에 알맞게 변형된 모델의 표면 위치를 마커의 목표 위치로 대응시킨다. 실험 당시 임의의 관측 시점으로부터 생성 가능한 마커들 중에서 임의로 일부의 마커만을 선택하여(1개, 5개, 10개, 50개, 120개) 모델 변형을 시도하였는데, 이 때 마커의 목표 위치는 모델 변형 과정에서 위치 제약조건의 역할을 수행하는 셈이다. 실험에서는 모델의 동작과정에서 대응된 마커의 목표 위치를 기준으로, 기본 자세(rest pose)의 모델을 가속화된 IK 풀이 방법을 통해 변형시켰다.

Figure 4-8은 앞서 언급한 다섯 가지 모델의 기본 자세와 특정 동작을 수행한 후 변형된 자세, 모델의 기본 자세에서 임의로 선택한 마커와 마커의 목표 위치를 기준으로 가속화된 IK 풀이 방법으로 변형된 결과를 보여준다. 마커의 개수는 위치 제약조건과 비례하므로, 마커의 수가 적을수록 모델의 자유로운 자세 변형

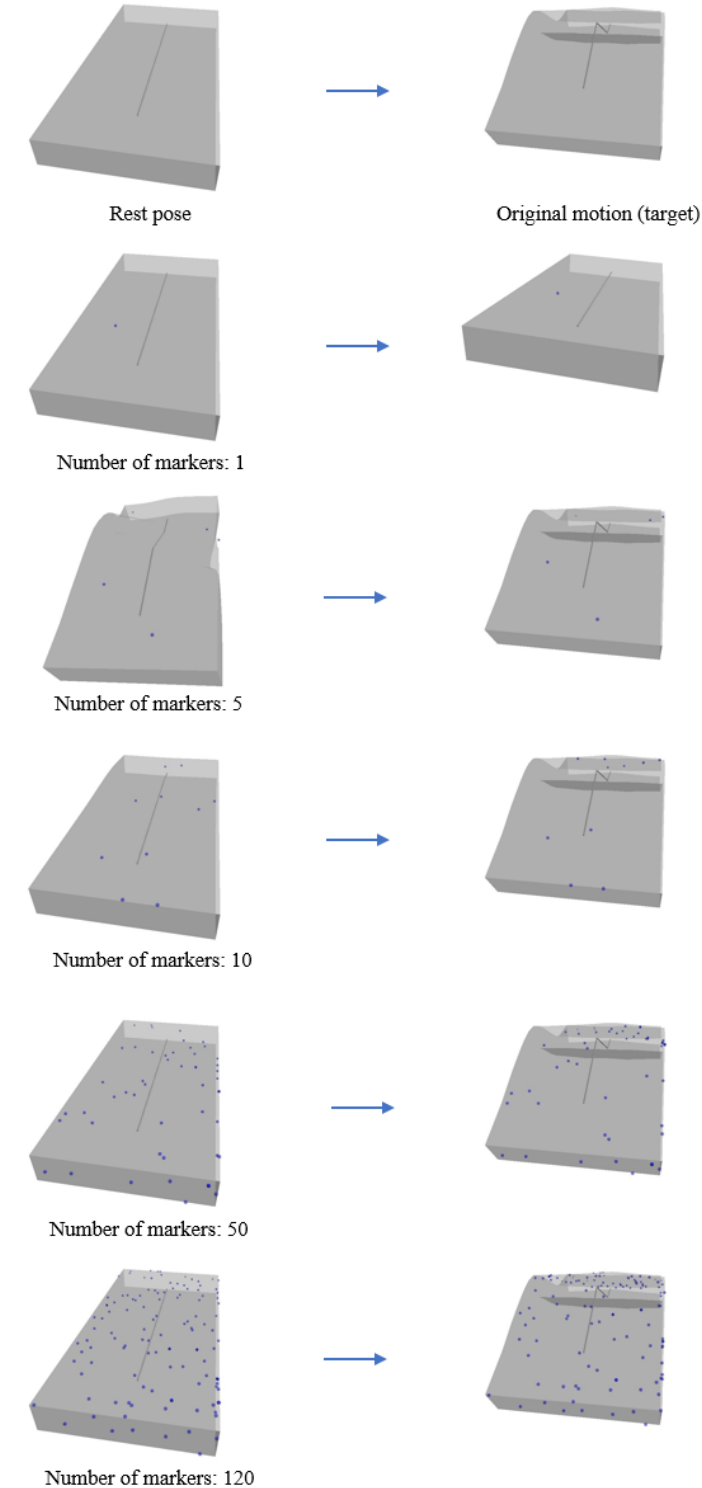


Figure 4: The results of testing with a cuboid sponge model: rest pose, target pose, and deformed pose based on markers. Comparing to the target pose, the smaller number of markers causes more uncontrolled deformation, and larger number of markers causes more controllable deformation.

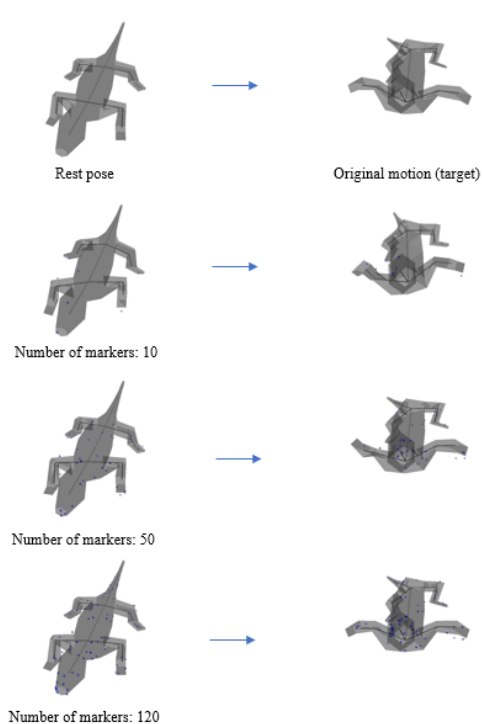


Figure 5: The test result using the iguana model: rest pose, target pose, and deformed pose based on markers. Compared to the target pose, the smaller number of markers causes more uncontrolled deformation, and using a larger number of markers produces satisfactory deformation similar to the ground truth.

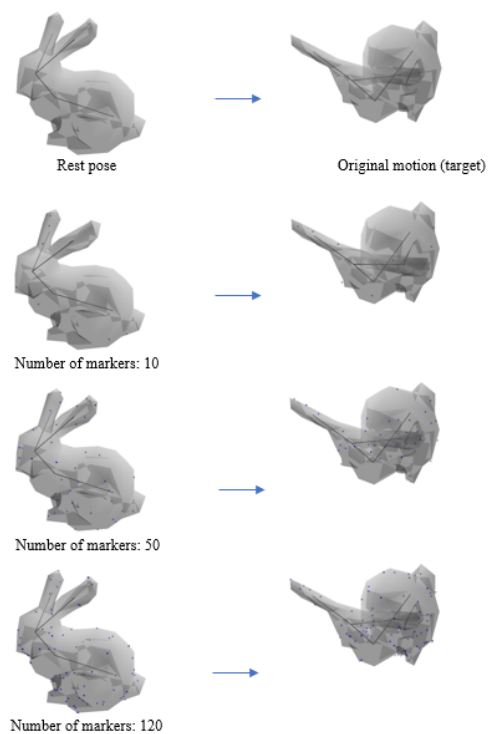


Figure 7: The test results with the bunny model: rest pose, target pose, and deformed pose based on markers.

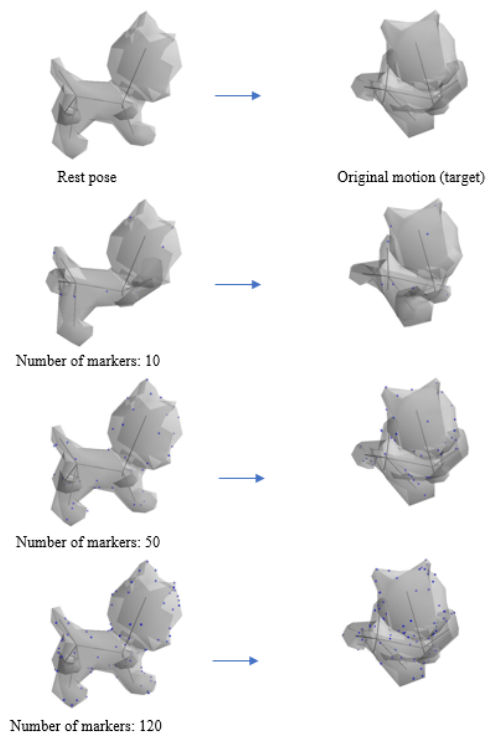


Figure 6: The test results using the cat model: rest pose, target pose, and deformed pose based on markers.

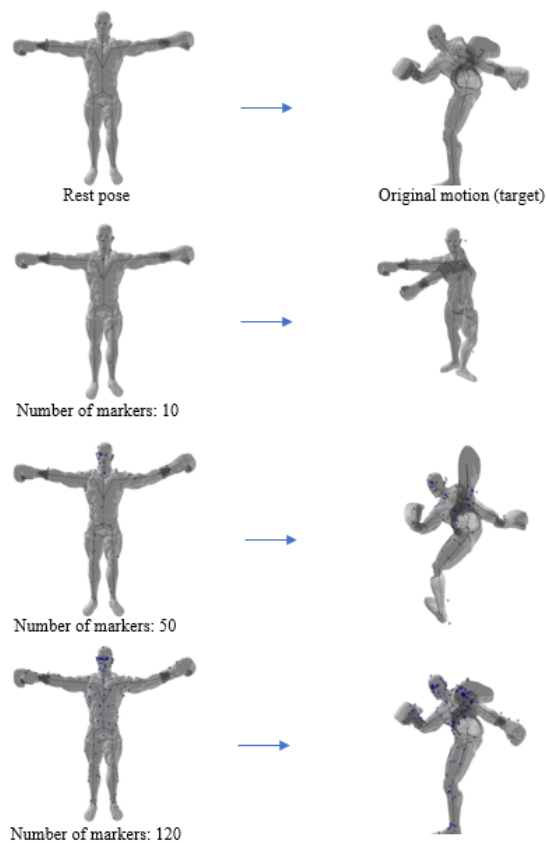


Figure 8: The test results with the human model: rest pose, target pose, and deformed pose based on markers.

이 이루어지고 반대로 마커의 수가 많을수록 절제된 자세 변형이 이루어지는 경향을 확인할 수 있다. 즉, 마커가 적을수록 제약조건이 적어 모델을 기본 자세로부터 목표 자세로 제어하는 것이 비교적 어렵고, 마커가 많을수록 목표 자세와 가깝게 제어하는 것이 가능하다. 그러나 마커가 1개인 경우에는 해당 마커 이외의 다른 제약조건이 없으므로, 변형 시 모델의 형태 변화보다는 단순 이동 및 회전이 이루어진다. 또한 마커의 수가 적을수록 메쉬 정점의 위치가 목표 자세와 비교하여 크게 변화하는 대신 삼각형 메쉬 모양의 변화는 상대적으로 적고, 마커의 수가 많을수록 메쉬 모양이 크게 변화하는 대신 목표 자세와 비교하여 위치가 적게 변화한다. 결론적으로 마커의 수가 많을수록 그 변형 결과 자세가 목표 자세와 비슷해지는 것을 확인할 수 있다.

Figure 4는 직육면체 스펀지 모델의 마커의 개수를 변화시키며 모델 변형을 시도한 결과이다. 마커가 1개일 때 변형 결과는 모델 전체가 단순 이동한 것이나, 마커의 갯수가 증가할수록 변형 결과가 목표 자세와 가까워지는 것을 확인할 수 있다. 직육면체 스펀지 모델의 경우 그 모양새가 매우 단순하여, 마커의 수가 적은 경우에도 충분히 목표 자세와 유사한 변형 결과를 도출할 수 있다. Figure 5는 이구아나 모델을 실험한 것이다. 마커 개수가 적을 때(5개, 10개), 변형 결과는 목표 자세와 크게 다른 양상을 보이는데, 이는 앞다리 부분에 마커가 설정되지 않아서 위치 제약조건이 부자연스럽게 적용되었기 때문이라 할 수 있다. 그러나 마커의 개수가 많아질수록, 모델 표면 메쉬에 마커들이 고르게 분포하게 되어 변형 결과가 목표 자세와 유사하게 자연스럽게 연출되는 것을 확인할 수 있다. Figure 6는 고양이 모델을 실험한 것으로, 이구아나와 마찬가지로 마커의 개수가 적을 때에는 다리 부분에 마커가 설정되지 않아서 변형이 부자연스러우나 마커의 개수가 증가할수록 고르게 분포되어 변형 결과가 목표 자세와 유사해지는 것을 확인할 수 있다. Figure 7는 토끼 모델을 실험한 것으로, 마찬가지로 마커의 개수가 적을 때에는 귀 부분에 마커가 설정되지 않아서 변형 결과가 목표 자세와 다른 양상을 보인다. Figure 8는 인체 모델을 실험한 것으로, 프레임 증가에 따라 발차기 동작을 수행한다. 마커가 1개일 때 변형 결과는 다른 모델과 마찬가지로 모델 전체가 단순 이동한다. 마커의 개수가 적을 때(5개, 10개)에는 모델의 복잡한 모양새 대비 마커가 충분히 고르게 분포되지 않아, 팔과 다리 부분의 변형 결과가 목표 자세와 전혀 다른 양상을 보인다. 그러나 충분히 많은 숫자의 마커가 설정되었을 때에는 변형 결과가 점차적으로 목표 자세와 가까워지는 것을 확인할 수 있다.

5.2 기존 IK 풀이 방법과의 비교

본 연구는 3D 모델의 변형을 빠른 연산 속도로 구현하는 것을 목표로 하며, 기존의 삼중 중첩 반복 구조를 개선하여 이중 중첩 반복 계산을 수행함으로써 그 성능을 향상시켰다. 성능 향상의 정확한 확인을 위하여 모델 변형 실험을 기존의 IK 풀이 방법(삼중 중첩 반복)과 새롭게 제안하는 가속화된 IK 풀이 방법(이중 중첩

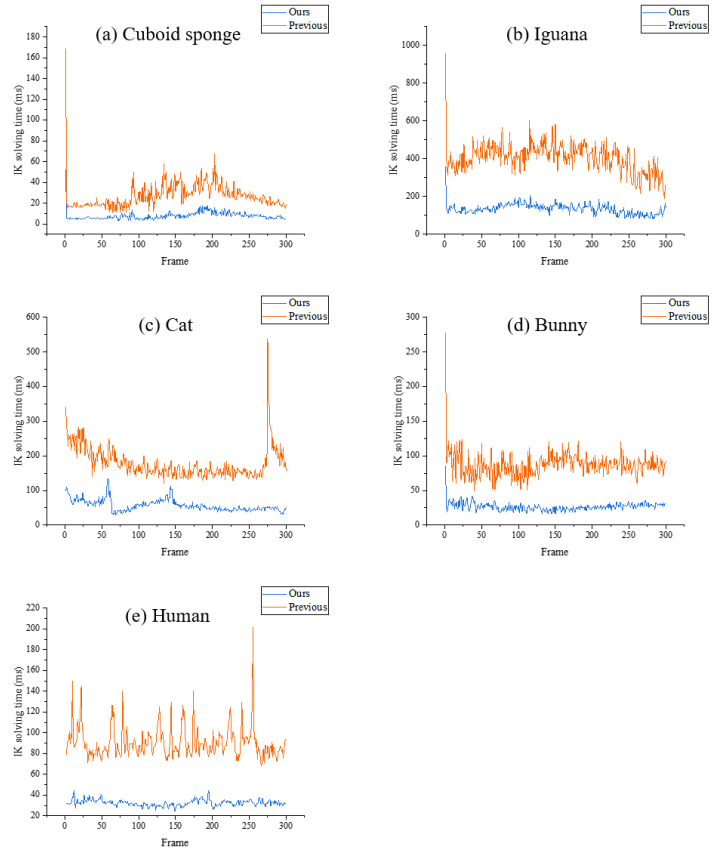


Figure 9: The comparison of IK solving time between a previous method and our method using 120 markers. The orange line indicates the previous method, and the blue line indicates our method. For most frames, our method shows a higher performance than the previous method.

반복) 두 가지 방식으로 수행하여 그 속도를 비교하였다.

Figure 9은 매 프레임마다 IK 풀이에 소요되는 연산 시간을 그래프로 작성한 것이다. 해당 실험은 다섯 가지 모델에 120개의 마커를 생성시킨 후에 관측한 것으로, 그래프에서 주황색 선은 기존의 삼중 중첩 반복 구조의 계산 방식에 따른 소요시간을, 파란색 선은 새롭게 제안하는 이중 중첩 반복 구조의 계산 방식에 따른 소요시간을 표시한 것이다. 그림을 통해 실험에 사용한 모델과 상관없이 기존 IK 풀이 방법보다 새로운 가속화된 IK 풀이 방법이 빠른 속도로 연산을 수행하는 것을 확인할 수 있다. Figure 9(a)는 직육면체 스펀지 모델을 사용하여 실험을 수행 및 비교한 결과이다. 해당 실험에서 기존 IK 풀이 방법의 경우 전체 프레임 평균 27.0183ms의 소요시간이 걸리고, 새로운 가속화된 IK 풀이 방법의 경우 평균 7.6984ms의 소요시간이 걸리는데 이는 기존보다 약 3.50배 빠른 속도이다. Figure 9(b)는 이구아나 모델을 사용하여 실험한 것으로, 기존 IK 풀이 방법의 전체 프레임 평균 연산 소요시간은 405.4133ms이고, 새로운 가속화된 IK 풀이 방법의 평균 연산 소요시간은 134.1667ms로 기존보다 약 3.02배 빠른 속도이다. Figure 9(c)는 고양이 모델을 사용하여 실험한 것으로, 기존 IK 풀이 방법의 전체 프레임 평균 연산 소요시간

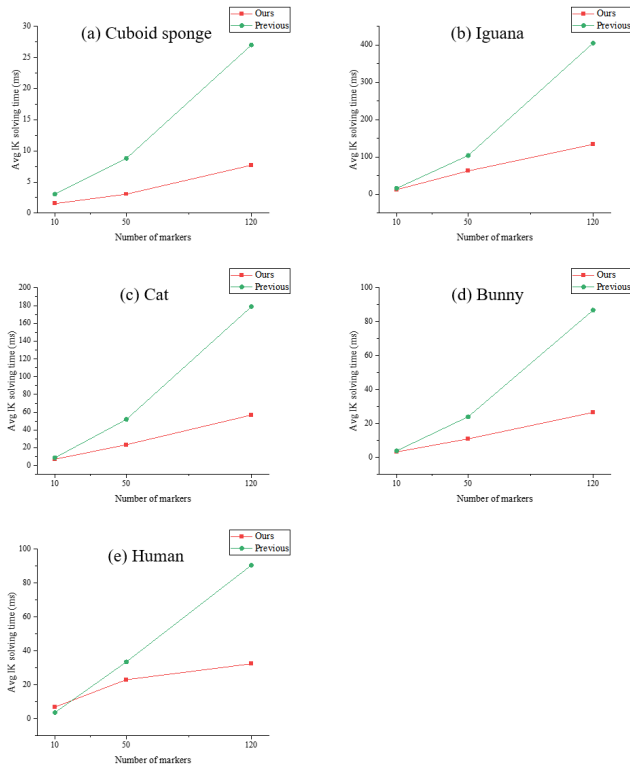


Figure 10: The comparison of the average solving time with respect to the number of markers between the previous method and our method. The green line indicates the average IK solving time performed by the previous method, and the red line indicates that for our method. In most cases, our method shows higher performance than previous method.

은 178.8598ms이고, 새로운 가속화된 IK 풀이 방법의 평균 연산 소요시간은 56.8361ms로 기존보다 약 3.14배 빠른 속도이다. Figure 9(d)는 토끼 모델을 사용하여 실험한 것으로, 기존 IK 풀이 방법의 전체 프레임 평균 연산 소요시간은 86.8782ms이고, 새로운 가속화된 IK 풀이 방법의 평균 연산 소요시간은 26.6069ms로 기존보다 약 3.26배 빠른 속도이다. Figure 9(e)는 인체 모델을 사용하여 실험한 것으로, 기존 IK 풀이 방법의 전체 프레임 평균 연산 소요시간은 90.5867ms이고, 새로운 가속화된 IK 풀이 방법의 평균 연산 소요시간은 32.4783ms로 기존보다 약 2.78배 빠른 속도이다. 이러한 실험을 통해 새롭게 제안하는 이중 중첩 반복 구조의 IK 풀이 방법이 기존의 방법보다 모든 모델 평균 약 3.14배 빠른 연산 속도로 모델 변형을 구현하는 것을 확인할 수 있다.

Figure 10은 마커의 개수가 각각 10개, 50개, 120개로 증가함에 따른 전체 프레임의 평균 연산 시간을 그래프로 작성한 것이다. 그래프에서 초록색 선은 기존의 이중 중첩 반복 구조의 계산 방식에 따른 평균 소요시간을, 빨간색 선은 새롭게 제안하는 삼중 중첩 반복 구조의 계산 방식에 따른 평균 소요시간을 표시한 것이다. 마커의 개수가 증가할수록 제약조건이 증가하여 연산 시간 또한 증가하는데, 해당 실험의 결과 마커 개수가 증가함에 따라 두 IK 풀이 방법의 연산 시간이 증가하는 기울기가 크게 벌어지는 것을 확인할 수 있다. 인체 모델을 사용한 실험에서 마커가 10

개 생성된 경우, 기존의 IK 풀이 방법(3.7711ms)이 새로운 가속화된 IK 풀이 방법(6.9772ms)보다 평균 연산 소요시간이 빠른데 이는 3.2061ms 정도의 근소한 차이라고 할 수 있다. 이러한 예외적인 상황을 제외하면, 전반적으로 새로운 가속화된 IK 풀이 방법이 기존의 IK 풀이 방법보다 연산 속도가 빠른 양상을 보이며, 마커(제약조건)의 수가 증가할수록 그 성능 향상의 폭이 크게 드러나는 것을 확인할 수 있다.

6. 결론

본 연구는 골격이 있고 삼각형 메시로 이루어진 3D 모델의 변형을 빠른 연산 속도로 구현하는 것을 목표로 한다. 이를 위해 삼각형 메시의 위치를 빠른 속도로 계산할 수 있는 IK 풀이 방법을 연구하였다. 모델 표면에 한 개 이상의 마커를 지정하고 마커의 목표 위치를 설정하면, 이 시스템은 마커의 목표 위치를 기준으로 가속화된 IK 풀이를 통해 모델 표면을 구성하는 삼각형 메시 정점의 위치를 계산한다. 이 계산 과정에서 빈번하게 사용되는 중복항을 사전에 계산해 둬으로써 메시 위치 계산에 소요되는 시간을 단축하였으며, 모델의 변형된 모습을 신속하게 결과 화면으로 확인할 수 있다.

마커는 위치 제약조건으로 작용하므로, 가속화된 IK 풀이 방법의 실험 결과는 마커의 개수에 따라 달라지는 양상을 보인다. 마커의 개수가 적을수록 모델을 기본 자세로부터 목표 자세로 제어하는 것이 비교적 어렵고, 마커가 많을수록 목표 자세와 가깝게 제어하는 것이 가능하다. 즉 마커의 수가 많을수록 그 변형 결과 자세가 목표 자세와 비슷해지는 것을 확인할 수 있다. 또한 기존의 일반적인 IK 풀이 방법과 비교했을 때 그 연산 속도가 평균 3.14배 빨라졌으며, 마커의 수가 많을수록 성능 향상이 크게 두드러지는 것을 확인할 수 있었다.

본 연구는 3D 모델 변형에 필요한 IK 풀이를 가속화된 방법으로 연산 속도를 개선했다는 데에 의의가 있다. 기존의 삼중 중첩 반복 구조의 계산 절차와 달리, 중복항을 사전에 계산함으로써 이중 중첩 반복 구조의 계산 절차를 수행한다. 이러한 개선된 계산 구조를 통해 모델 변형 결과를 빠르게 구현할 수 있고, 이러한 가속화된 IK 풀이 방법은 3D 모델을 다루는 다양한 분야에서 활용할 수 있다. 예를 들어, LBS 기법으로 구현된 3D 모델을 다루는 데 자연스럽고 빠른 속도의 모델 변형 또는 제어가 가능하다. 또한 사용자와의 상호작용 인터페이스를 접목하여 3D 모델의 편집 도구로도 응용할 수 있다. 가속화된 IK 풀이 방법은 컴퓨터 비전 분야, 특히 무마커 추적 관련 연구에서 유용하게 활용될 수 있다.

본 연구에서는 모델 표면에 존재하는 마커들의 목표 위치를 위치 제약조건으로 삼아 IK 풀이를 수행한다. 이는 등식 제약조건(equality constraints)으로서, 제약조건의 만족을 위해 마커로 선택된 정점의 위치 계산에만 변형이 집중되어 부자연스러운 변형 결과를 도출하기도 한다. 따라서 향후 연구로는 위치 제약조건 이외에도 다양한 제약조건을 적용하여 보다 자연스럽고 안정적인 변형을 연출하고자 한다. 예를 들어 법선 벡터 제약조건(nor-

mal constraints)을 적용할 경우, 법선 벡터를 이용하여 평면의 유무를 확인함으로써 메쉬 관통(penetration) 현상을 개선할 수 있다. 또한 실루엣 제약조건(silhouette constraints)을 적용할 경우, 목표 자세와 변형 결과 자세 간의 오차를 보완할 수 있고, 이외에도 광선 제약조건(ray constraints), 평면 제약조건(plane constraints), 여러가지 비동식 제약조건(inequality constraints) 등을 적용하여 결과 품질을 개선할 수 있다. 이렇게 개선된 IK 풀이 방법은 실제 관측 데이터를 활용하는 컴퓨터 비전 기반 어플리케이션에서 유용하게 도입될 수 있다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2019R1A4A1029800, NRF-2020R1A2C1012847).

References

- [1] A. Aristidou, J. Lasenby, Y. Chrysanthou, and A. Shamir, "Inverse kinematics techniques in computer graphics: A survey," Computer Graphics Forum, pp. 35–58, 2018.
- [2] S. R. Buss, "Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods," IEEE Journal of Robotics and Automation, Tech. Rep., 2004.
- [3] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, ser. SIGGRAPH '99, 1999, p. 39–48.
- [4] X. Wu, M. Tournier, and L. Reveret, "Natural Character Posing from a Large Motion Database," IEEE Computer Graphics and Applications, pp. 69–77, 2011.
- [5] D. Holden, J. Saito, and T. Komura, "A deep learning framework for character motion synthesis and editing," ACM Trans. Graph., 2016.
- [6] J. Kim, Y. Seol, H. Kim, and T. Kwon, "Interactive character posing with efficient collision handling," Computer Animation and Virtual Worlds, vol. 31, no. 3, p. e1923, 2020.
- [7] A. Jacobson, Z. Deng, L. Kavan, and J. Lewis, "Skinning: Real-time shape deformation," in ACM SIGGRAPH 2014 Courses, 2014.
- [8] R. W. Sumner, M. Zwicker, C. Gotsman, and J. Popović, "Mesh-based inverse kinematics," in ACM SIGGRAPH 2005 Papers, ser. SIGGRAPH '05. Association for Computing Machinery, 2005, p. 488–495.
- [9] A. Jacobson, I. Baran, L. Kavan, J. Popović, and O. Sorkine, "Fast automatic skinning transformations," ACM Trans. Graph., vol. 31, no. 4, 2012.
- [10] K. Wampler, "Fast and reliable example-based mesh ik for stylized deformations," ACM Trans. Graph., vol. 35, no. 6, 2016.
- [11] L. Mündermann, S. Corazza, and T. Andriacchi, "The evolution of methods for the capture of human movement leading to markerless motion capture for biomechanical applications," Journal of NeuroEngineering and Rehabilitation, 2006.
- [12] Z. Zhang, "Microsoft kinect sensor and its effect," IEEE multimedia, vol. 19, no. 2, pp. 4–10, 2012.
- [13] L. Keselman, J. Iselin Woodfill, A. Grunnet-Jepsen, and A. Bhowmik, "Intel realsense stereoscopic depth cameras," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 1–10.
- [14] L. Becker and P. Russ, "Evaluation of joint angle accuracy using markerless silhouette based tracking and hybrid tracking against traditional marker tracking," Poster für Masterarbeit bei Simi Reality Motion Systems GmbH und der Otto-von-Guericke-Universität Magdeburg, 2015.
- [15] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "Openpose: Realtime multi-person 2d pose estimation using part affinity fields," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 1, pp. 172–186, 2021.
- [16] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," ACM Trans. Graph., vol. 35, no. 4, 2016.
- [17] O. C. Zienkiewicz, R. L. Taylor, and J. Z. Zhu, The finite element method: its basis and fundamentals. Elsevier, 2005.
- [18] J. Kim and N. S. Pollard, "Fast simulation of skeleton-driven deformable body characters," ACM Trans. Graph., vol. 30, no. 5, 2011.

〈 저 자 소 개 〉



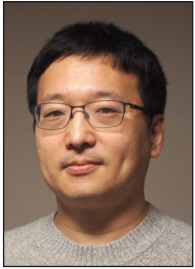
박 현 아

- 2015-2019 세종대학교 디지털콘텐츠학과 학사
- 2020-현재 한양대학교 컴퓨터소프트웨어학과 석사
- 관심분야: Character Animation, Physically-based Animation, Deep Reinforcement Learning
- <https://orcid.org/0000-0001-7642-8247>



강 다 은

- 2010-2014 숙명여자대학교 컴퓨터과학 학사
- 2014-현재 한양대학교 컴퓨터소프트웨어학과 석박사 통합과정
- 관심분야: Character Animation, Physically-based Animation
- <https://orcid.org/0000-0003-2362-7669>



권 태 수

- 1996-2000 서울대학교 전기컴퓨터공학부 학사
- 2000-2002 서울대학교 전기컴퓨터공학부 석사
- 2002-2007 한국과학기술원 전산학전공 박사
- 관심분야: Character Animation, Physically-based Animation
- <https://orcid.org/0000-0002-9253-2156>