한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 28, No. 2, P. 29~41

저가형 모션 캡처 장비를 이용한 실시간 상호작용 애니메이션 시스템

김정호° 강다은 이윤상 권태수*

한양대학교 일반대학원 컴퓨터소프트웨어학과

{cpls93, teddysiah, yoonsanglee, taesoo}@hanyang.ac.kr

Real-time Interactive Animation System for Low-Priced Motion Capture Sensors

Jeongho Kim° Daeun Kang Yoonsang Lee Taesoo Kwon*

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

요 약

본 논문에서는 대표적인 보급형 장비인 키넥트를 활용하여 실시간으로 사용자 캐릭터의 자세를 제어하고, 상대 캐릭터와 함께 자연스러운 상호작용 동작을 수행하는 실시간 상호작용 애니메이션 시스템을 소개한다. 해당 상호작용 애니메이션 시스템은 실시간으로 두 캐릭터의 상호작용 동작을 연출하는 시스템으로, 사용자는 키넥트를 이용한 자세 입력을 통해 사용자 캐릭터의 동작을 제어하고 상대 캐릭터는 사용자 캐릭터의 동작에 따라 반응하는데 이 반응 동작은 시스템에 의해 자동으로 결정된다. 전처리 과정은 예제 동작 데이터 정보를 사전에 관측 및 분석하여 맵핑 모델을 생성하고, 실시간 처리 과정에서는 사용자의 실시간 입력에 맞는 두 캐릭터의 자세(동작)을 실시간으로 생성 및 보정 후 최종 결과 애니메이션을 화면에 출력한다. 실험 결과를 통해 해당 시스템은 사용자의 입력 동작에 맞추어 상대 캐릭터는 적절한 대응 동작을 수행하고, 화면상의 두 캐릭터가 서로 상호작용 동작을 연출하는 것을 확인할 수 있다. 본 논문에서 제안하는 기술 및 아이디어는 응용하여 실제 사용자 상호작용 소프트웨어 개발에 적용할 수 있고, 이를 통해 사용자에게 더 나은 몰입감을 제공할 수 있을 것이다.

Abstract

In this paper, we introduce a novel real-time, interactive animation system which uses real-time motion inputs from a low-cost motion-sensing device Kinect. Our system generates interaction motions between the user character and the counterpart character in real-time. While the motion of the user character is generated mimicking the user's input motion, the other character's motion is decided to react to the user avatar's motion. During a pre-processing step, our system analyzes the reference motion data and generates mapping model in advance. At run-time, our system first generates initial poses of two characters and then modifies them so that it could provide plausible interacting behavior. Our experimental results show plausible interacting animations in that the user character performs a modified motion of user input and the counterpart character properly reacts against the user character. The proposed method will be useful for developing real-time interactive animation systems which provide a better immersive experience for users.

키워드: 보급형 모션 캡처 장비, 키넥트, 실시간 상호작용 애니메이션, 예제 기반 애니메이션
Keywords: popularly distributed motion capture devices, Kinect, real-time interactive animation, example-based animation

1. 서론

최근 저가형 동작 인식 장비(motion capture devices)가 널리 보급되고 있으며, 대표적인 보급형 동작 인식 장비로는 마이크로소프트사(Microsoft社)의 키넥트(Kinect), 인텔사(Intel社)의 리얼센스(Realsense) 등이 있다[1, 2]. 보급형 동작 인식 장비는 사용자의 특정 자세 또는 연속 동작을 보이는 그대로 인식하여 대응 소프트웨어에 입력값으로 전달한다. 이러한 장비를 이용한 소프트웨어는 일반적으로 사용자를 가상현실 내에 존재하는 캐릭터

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

Received : 2022.03.04./ Review completed : 1st 2022.04.25. / Accepted : 2022.05.26.
DOI : 10.15701/kcgs.2022.28.2.29
ISSN : 1975-7883(Print)/2383-529X(Online)

에 대입하여, 사용자가 직접 동작에 따라 사용자 캐릭터를 제어하는 방식으로 실행된다. 이는 마우스, 키보드, 조이스틱 등의 기존의 입력 장치에 비해 훨씬 높은 자유도의 입력을 가능케 하며, 이에 따라 대응 소프트웨어의 컨텐츠의 폭이 넓어지게 되었다. 이를 통해 사용자는 상대적으로 저렴한 가격의 장비를 통해 직접 가상현실의 주인공으로 참여하고 있는 듯한 분위기를 체험할 수 있으며, 보다 큰 재미를 느낄 수 있다. 꾸준한 수요로 인해 보급형 동작 인식 장비와 관련 소프트웨어의 시장은 빠른 속도로 성장하고 있으며, 이에 발맞추어 소비자, 즉 사용자의 요구사항 수준 또한 높아지고 있다. 관련 컨텐츠의 다양성은 물론이거니와 더 높은 몰입감을 위한 입출력 성능의 향상이 요구되고 있는 상황이다.

보급형 동작 인식 장비는 최소한의 사양의 하드웨어로 구성되어 있어 그 성능이 낮은 편인데, 특히 사용자의 자세 입력의 정확도가 낮은 점을 지적할 수 있다. 사용자의 신체부위가 서로 겹치는 듯한 자세는 인식하기 어려우며, 간단한 자세라도 인식 과정에서 자체적인 노이즈(noise)가 발생할 수 있다. 입력 정확도가 낮으면 연관된 소프트웨어를 작동하는 데에도 영향을 끼치며, 결과적으로 사용자의 만족감을 저하시키므로 성능 보완의 필요성이 대두된다. 이러한 성능 한계를 극복하기 위한 방법은 크게 두 가지가 있다. 첫 번째는 하드웨어 자체의 사양을 높이는 것인데 이는 가장 직관적이고 확실한 방법이나, 보급형 시장의 특성상 비용적 측면에서 실현이 어려운 편이다. 두 번째는 소프트웨어 차원에서 입력 정확도를 보완하여 사용자의 몰입감을 향상시키는 방법이다. 입력된 자세로부터 사용자가 실제로 의도하고자 했던 자세가 무엇이었는지 유추하고 입력 자세를 목적에 맞게 보정하면, 사용자는 장비의 성능적 한계를 체감하지 않고 소프트웨어와의 상호작용에 몰입할 수 있게 된다.

본 연구는 앞서 언급한 보급형 동작 인식 장비의 성능 한계 극복 방법 중 후자의 아이디어에 착안하여, 입력 정확도를 보완한 실시간 상호작용 애니메이션 기술 구현을 목표로 하였다. 사용자가 제어하는 '사용자 캐릭터(user character)'와 컴퓨터가 자동으로 제어하는 '상대 캐릭터(counterpart character)'가 상호작용하는 애니메이션 기술을 구현하기 위해서는 장비의 입력 정확도를 고려하는 것과 더불어 사용자가 소프트웨어의 실행 목적에 맞게 적합한 자세를 입력하도록 유도하는 것 또한 중요하다. 이를 위해서는 사용자가 소프트웨어가 지원하는 여러 가지 종류의 상호작용 동작 중 하나를 소프트웨어 상에 존재하는 가상 환경에 맞게 수행할 수 있어야 한다. 하지만 사용자가 존재하는 현실 세계와 사용자 캐릭터가 존재하는 가상 환경은 일반적으로 일치하지 않으므로 실제로 사용자가 가상 환경의 상대 캐릭터와 상호작용하기 위한 자세를 정확하게 입력하는 것은 불가능하다. 사용자와 소프트웨어의 원활한 상호작용을 위해서는 사용자의 입력 자세를 그대로 사용자 캐릭터에 적용하지 않고, 대략적인 자세 분석을 통해 목적(사용자 의도)에 맞는 동작을 파악하고 가상 환경에 맞게 보정하여 사용자 캐릭터의 최종 동작으로서 출력할 필요가 있다.

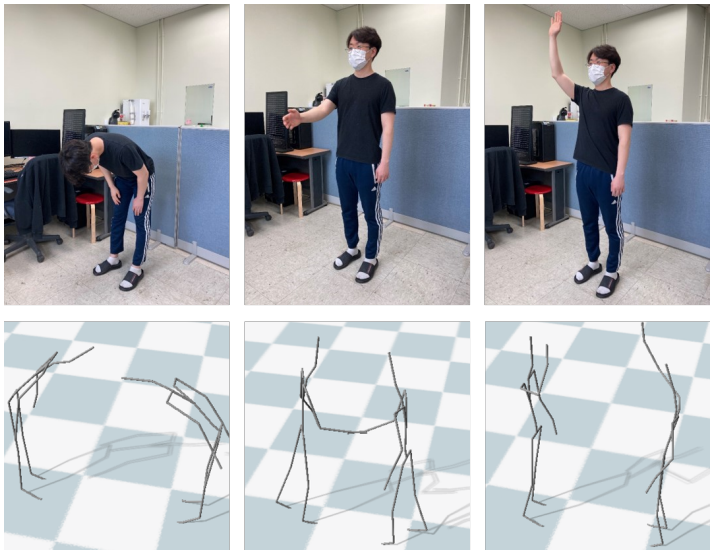


Figure 1: The first row shows a scene where the user inputs poses in real-time with Kinect, and the second row shows the resulting motions of both the user character and the counterpart character.

본 논문에서는 대표적인 보급형 장비인 키넥트를 활용하여 실시간으로 사용자 캐릭터의 자세를 제어하고, 상대 캐릭터와 함께 자연스러운 상호작용 동작을 수행하는 실시간 애니메이션 시스템을 소개한다(Figure 1 참조). 해당 상호작용 애니메이션 시스템은 실시간으로 두 캐릭터의 상호작용 동작을 연출하는 시스템으로, 사용자는 키넥트를 이용한 자세 입력을 통해 사용자 캐릭터의 동작을 제어하고 상대 캐릭터는 사용자 캐릭터의 동작에 따라 반응하는데 이 반응 동작은 시스템에 의해 자동으로 결정된다. 이때 사용자 캐릭터의 동작은 실제 입력 동작을 그대로 적용하는 대신, K-최근접 이웃(K-nearest neighbor, KNN) 알고리즘을 통해 기존의 2인 동작으로 구성된 학습용 모션 데이터로부터 가장 유사한 동작을 탐색하여 시스템이 지원하는 상호작용 동작에 적합하게 보정된다. 또한 상대 캐릭터의 동작은 사용자 캐릭터의 동작에 대응되는 예제 데이터 동작을 차용한다. 두 캐릭터의 초기 자세는 역기구학(inverse kinematics, IK)을 이용하여 캐릭터에 합성 및 보정되며, 부가적으로 충돌 회피 기능을 적용하여 보다 한층 더 자연스러운 자세로 교정함으로써 사용자의 소프트웨어 몰입감을 유도할 수 있다.

본 연구에서는 보급형 동작 인식 장비 및 관련 소프트웨어 시장의 성장에 발맞추어 보급형 장비의 성능적 한계를 보완하고 실시간으로 상호작용 애니메이션을 연출하는 시스템의 구현을 시도하였다. KNN 알고리즘을 이용한 동작 탐색 및 동작 합성 기술을 통해 애니메이션 동작 품질을 높이되, 실시간 작동을 위하여 최적화된 시스템 구성을 설계하였다. 이러한 시스템을 통해 사용자는 직접 동작을 입력하고 사용자 캐릭터의 동작에 따라 상대 캐릭터가 실시간으로 반응 동작을 수행하는 것을 확인할 수 있다. 사용자와의 실시간 상호작용을 통해 두 캐릭터가 인사, 악수, 하이파이브 등의 상호작용 동작을 수행할 수 있으며, 자세의 높

낮이나 각도 변화에도 자연스럽게 대응할 수 있다. 본 논문에서 제안하는 기술 및 아이디어는 실제 사용자 상호작용 소프트웨어 개발에 적용될 수 있고, 이를 통해 사용자에게 더 나은 몰입감을 제공할 수 있을 것이다.

본 논문은 총 8장으로 구성되어 있으며, 2장에서는 캐릭터 상호작용 동작 애니메이션 기술 및 사용자와 시스템 간의 실시간 상호작용 애니메이션 시스템에 관련된 기존 연구를 소개한다. 또 3장에서는 본 논문에서 연구개발한 실시간 상호작용 애니메이션 시스템의 전반적인 설계 구성과 작동 흐름을 소개한다. 이어 4장에서는 키넥트 입력 자세 정보 및 입력 처리에 대해 구체적으로 서술하고, 5장에서는 예제 동작 데이터 정보를 사전에 관측 및 분석하는 전처리 과정을, 6장에서는 사용자의 입력에 맞는 동작을 생성하는 실시간 처리 과정에 대하여 자세히 설명한다. 마지막으로 7장에서는 실제 실험자가 키넥트를 이용하여 본 시스템의 실시간 작동을 실험한 내용과 그 결과를 제시하고, 8장에서는 본 연구의 의의와 한계, 향후 연구계획 및 응용 방향에 대하여 서술한다.

2. 관련 연구

2.1 예제 기반 캐릭터 상호작용 동작 생성 관련 연구

영화, 애니메이션, 게임 등 대다수의 시각 컨텐츠를 제작하는 데 다수의 캐릭터 또는 캐릭터 및 오브젝트 간의 상호작용 동작은 빈번히 사용된다. 이러한 수요와 함께 해당 동작을 묘사하기 위한 애니메이션 및 시뮬레이션 기술에 관한 연구 또한 꾸준히 이루어져왔다. 단일 캐릭터의 동작을 묘사하는 대다수의 연구는 동작의 자연스러움과 동작에 따른 몸체 변화에 초점을 맞추어 진행한다. 상호작용 동작의 경우, 단순 동작 구현에 그치지 않고 두 캐릭터 또는 오브젝트 간의 접촉과 그로 인한 반응 동작 또는 변화를 구현해야 하므로 그 난이도가 높다. 예를 들어 캐릭터 또는 오브젝트가 상호작용 할 때는 특정 점이 접촉되거나 일정 거리를 유지해야 하고, 두 몸체가 서로 관통(penetration)해서는 안 된다. 또한 상호작용의 시점 이후로 반응 동작을 취하는 경우도 있고, 캐릭터 및 오브젝트의 특성에 따라 상호작용 시점 전후로 몸체의 변화가 일어나기도 한다. 이처럼 자연스러운 상호작용 동작의 구현에는 각 캐릭터의 자세 이외에도 고려해야 할 요소가 다수 존재하며, 더 나은 동작 묘사를 위해 다양한 연구들이 진행되어왔다.

상호작용 동작에 관한 연구는 상호작용의 주체에 따라 구분할 수 있다. 단일 캐릭터와 오브젝트 간의 상호작용(물건 집기 등), 두 캐릭터 간의 상호작용(커플 댄스 등), 그리고 나아가 캐릭터와 오브젝트 수를 늘려 더 복잡한 상호작용을 구현한 응용연구들도 있다. 또한 캐릭터 동작 생성을 하는데 가장 직관적인 방법은 모션 캡처 데이터베이스로부터 필요한 동작에 관한 정보를 참조하는 것이다. 동작 생성의 참고가 되는 모션 데이터를 예제 동작(reference motion) 또는 예제 데이터(reference data)라고도 하는데, 상호작용 동작 생성 기술에도 예제 데이터 기반으로 작동하는 기술 연구가 다양하다. 본 논문은 두 캐릭터의 상호작용 동작을 예제 기반(example-based)으로 생성하는 기술을 연구개발한 것으로, 이 절에서는 본 연구와 마찬가지로 둘 이상의 캐릭터의 상호작용 동작을 예제 기반으로 생성하는 기존 연구들을 소개하고자 한다.

Kim et al.은 예제 춤동작으로부터 동작의 리듬 패턴인 '모션 비트(motion beats)'를 분석하여, 소리 입력의 리듬 신호에 맞추어 2인 캐릭터의 볼룸 댄스 동작이나 군중 캐릭터의 행렬 행진 동작을 생성하는 애니메이션 기술을 연구하였다[3]. 모션 비트의 추출은 기존 Jones와 Boltz의 연구에서 차용하여 동작의 방향 변화를 기준으로 이루어지며, 모션 비트에 따라 모션 그래프를 작성하고 실시간 탐색을 통해 각 동작을 연속으로 블렌딩(blending)하였다[4]. Hsu와 Gentry, Popovic 또한 볼룸 댄스 동작을 생성하는 애니메이션 기술을 연구하였는데, 동작별로 의미를 부여한 맵핑 예제(mapping instance)로 이루어진 데이터베이스를 작성함으로써 사용자가 입력한 이동 경로를 따라 볼룸 댄스를 추는 2인 캐릭터 애니메이션을 자동적으로 생성할 수 있었다[5].

Kwon et al.은 동작 모델링, 상호작용 모델링, 동작 합성 과정을 모두 포함한 예제 기반 2인 캐릭터 상호작용 동작 애니메이션 시스템을 연구개발하였다[6]. 해당 시스템은 강제 기반 방식(force-based method)으로 예제 동작 데이터를 분할(segmentation)하고, 각 분할 동작을 학습 기반으로 분류하며, 베이즈 네트워크(Bayesian network)를 통해 동작 간의 전이(motion transition)를 결정한다. 활용 예로 킥복싱과 같은 복잡한 발 접촉과 두 캐릭터 간의 상호접촉이 동반된 높은 난이도의 상호 작용 동작도 자연스럽게 구현함으로써 해당 시스템의 성능을 입증하였다. Shum et al.은 킥복싱과 레슬링 같은 두 캐릭터의 복잡한 상호작용이 포함된 2인 격투 스포츠 동작을 생성하는 새로운 기술을 소개하였다[7]. 해당 연구에서는 예제 데이터로 상대적으로 정확도가 떨어지는 2인 캡처 동작 데이터를 사용하는 대신, 1인 개별 동작 데이터로부터 상호작용 동작으로서 대응할 수 있는 두 동작을 선택하여 각각의 캐릭터에 입힘으로써 2인 애니메이션을 생성하는 기술을 연구하였다. 상호작용에 적합한 동작을 선별하는 방법으로는 이산 공간 시스템(discrete space system) 내에서 연속 동작 계획을 배치하는 데 유용한 '시간적 확장 접근법(temporal expansion approach)'이라는 독자적인 방법을 개발하였다. 사용자 제어 캐릭터의 동작에 따라 가상의 상대 캐릭터의 반응 동작이 가장 상호작용에 적합한 동작으로 자동 결정되므로, 직접 2인 예제 데이터를 사용하는 것보다 폭넓은 상호작용 동작을 생성할 수 있다. Shum et al. 은 또한 이러한 2인 상호작용 동작 생성 기술 연구를 응용하여 2인 이상의 다수의 캐릭터에 적용할 수 있는 기술을 연구개발하였다[8]. 서로 상호작용 중인 캐릭터들의 다음 동작을 결정하기 위해 강화 학습(reinforcement learning) 기법을 적용하고자 했으나, 다수의 캐릭터에 대응하는 넓은 상태 공간(state space)에서는 강화 학습 기법을 수행하는 것이 어렵다. 따라서 전체 상태 공간 내에 의미 있는 상태(동작)만 모아놓은 부

분 공간(sub-spaces)을 만들고, 두 캐릭터의 상호작용 동작을 그래프 형식으로 정리한 '상호작용 그래프(interaction graph)'라는 유한 상태 기계(finite state machine)을 설계하였다. 해당 기술은 여러 명의 캐릭터가 권투를 하거나 떼 지어 이동하거나 함께 짐을 나르는 등의 상호작용 동작을 생성하는 데 우수한 성능을 보여주었다. 또한 Shum et al.은 몇 명의 캐릭터 상호작용 동작 생성에 그치지 않고 단체 스포츠, 격투, 물건 옮기기 등 수십, 수백명에 달하는 캐릭터들이 얽힌 상호작용 동작 생성 기술을 제안하였다[9]. 해당 연구에서는 게임 트리를 확장하여 상호작용 동작을 사전에 계산하고, 시공간적으로 연쇄된 '상호작용 패치(interaction patches)' 자료 구조 형태로 저장함으로써 대량의 동작 생성 계산을 수행할 수 있었다. 이후 Shum et al.은 2인 이상의 캐릭터가 함께 일하거나 스포츠 게임을 겨루는 동작을 구현하는 동작 합성 기술을 연구개발하였다[10]. 기존의 1인 예제 동작 데이터를 사용하여 2인 캐릭터의 상호작용 동작을 생성하는 연구를 응용한 것으로[7], 1인 동작을 탐색하는 모션 그래프를 작성하고 최소-최대 탐색(min-max search)을 통해 게임 트리를 평가함으로써 캐릭터들의 다음 동작을 결정하는 방식으로 구현하였다. 해당 연구와 유사한 연구 주제로 Wampler et al.은 두 캐릭터가 스포츠 게임을 겨루는 동작 계획 기술을 연구개발하였다[11]. 캐릭터의 다음 동작은 제로섬 마르코프 게임 모델(zero-sum Markov game model)을 차용하여 상대 캐릭터의 현재 동작과 앞으로 예상되는 반응 동작을 고려하여 결정하도록 구현하였다. 이러한 동작 생성 기술은 상대 캐릭터의 다음 동작을 예상하고 상대 캐릭터의 공격의 효과를 최대한 줄이기 위한 대응 동작을 결정함으로써 두 캐릭터가 겨루는 장면을 흥미롭게 묘사할 수 있다.

본 연구에서는 두 명의 연기자가 상호작용하는 것을 동시에 포착한(capture) 2인 모션 캡처 데이터를 예제 데이터로 사용하되, 사용자 캐릭터는 보급형 동작 인식 장비를 통해 입력된 사용자의 동작을 가상 환경에 맞게 최대한 모방하고, 가상의 상대 캐릭터의 동작은 사용자 캐릭터의 동작에 따라 결정되는 두 캐릭터의 상호작용 동작 생성 기술을 다룬다. 캐릭터 동작을 입히는 데에는 사용자 입력 동작이나 예제 데이터의 동작을 그대로 모방하지 않고, 가상 환경 및 상호작용 상황에 맞게 자세의 높낮이나 각도와 같은 세부적인 요소를 조정한다. 예제 데이터를 참조하는 데에는 K-최근접 이웃 알고리즘을 이용하여 사용자 입력과 유사한 동작을 선별하고, 두 캐릭터에 상호작용 동작을 입히고 편집하는 데에는 역기구학을 사용하였다. 결과적으로 두 캐릭터가 실시간으로 인사, 악수, 하이파이브 등의 상호작용 동작을 자연스럽게 수행할 수 있다.

2.2 실시간 사용자 입력에 따른 상호작용 동작 생성 관련 연구

두 캐릭터의 상호작용 동작 생성 및 편집에 관한 기술 연구는 더 높은 품질과 작동 효율을 위해 끊임없이 다양한 방식으로 이루어져 왔다. 그에 비해 사용자와 컴퓨터가 모션 인식 장비를 통해 실시간으로 상호작용하며 두 캐릭터의 상호작용 동작을 결정하는 상호작용 시스템에 관한 기술 연구는 상대적으로 적은 편이다. 실시간 상호작용 시스템의 구현에는 동작의 품질은 물론이고 사용자와 즉흥적인 상호작용을 수행할 수 있도록 빠른 반응 속도도 중요하므로, 일정 수준의 결과 동작 품질을 유지하면서 계산량 또한 최적하는 데에 초점을 맞추어 연구가 진행되어 왔다.

Lee와 Lee는 두 캐릭터의 상호작용 동작을 생성하는 데 앞서 분류되지 않은 방대한 양의 예제 모션 데이터를 사전에 계산해 둠으로써 실시간 계산량을 줄이고자 하였다[12]. 기존 연구들과 유사하게 각 동작을 탐색하여 최적의 경로(다음 동작)를 찾는 방법을 택하는 대신, 상대 캐릭터의 동작에 대해 반응하는 동작을 결정하는 제어 정책(control policy)을 사전에 미리 계산함으로써 실시간 작동에 필요한 시간 비용을 감소시킬 수 있었다. Ho와 Komura는 사용자 캐릭터와 상대 캐릭터가 실시간으로 레슬링을 할 수 있는 상호작용 시스템을 연구개발하였다[13]. Ho와 Komura가 이전에 연구개발한 위상 좌표(topology coordinates)를 사용하여 유한 상태 기계를 사전에 계산함으로써 상대 캐릭터가 실시간으로 공격 또는 방어 동작을 선택하여 취할 수 있도록 설계하였다[14]. 해당 시스템을 통해 사용자는 상대 캐릭터와 폭넓은 자유도의 동작과 빠른 반응속도로 레슬링 경기를 즐길 수 있다. 이후 Ho et al.은 사용자가 광학 모션 캡처 장비를 이용한 실시간 동작 입력으로 제어할 수 있는 두 캐릭터의 댄스 또는 격투 등의 상호작용 동작을 생성하는 실시간 상호작용 애니메이션 시스템을 연구개발하였다[15]. 해당 연구에서는 높은 난이도의 제약조건이 부가되는 상호작용 동작을 자연스럽게 생성하기 위해 Ho et al.의 상호작용 메쉬(interaction mesh) 관련 이전 연구를 응용하여 다양한 동작 변화를 꾀하도록 하였다[16]. 시스템 작동 실험 결과, 사용자의 입력에 따라 댄스 또는 격투와 같은 상호작용 동작을 빠르게 생성할 수 있었으며, 생성된 동작은 예제 데이터와 근소한 차이를 보여 결과 동작의 자연스러운 정도를 입증하였다. 또한 Kulpa et al.은 상호작용 동작의 원활한 실시간 생성을 위해 캐릭터 동작 변화에 필요한 계산량을 줄이고자 새로운 아이디어를 제시하였다[17]. 일반적으로 캐릭터 동작을 변화시키는 데에는 각 관절들(joints)을 특정값으로 회전시키는 과정이 포함되는데 이는 상당한 계산량을 요구한다. 관절 회전을 계산하는 대신 예제 데이터 상의 루트 관절(root joint)의 이동을 정규화(normalize)함으로써 시스템 캐릭터와 예제 데이터 캐릭터의 구조를 대응시키고 동작을 입히는 데 필요한 계산량을 줄이는 데 성공하였다.

본 연구에서는 사용자와 시스템이 실시간으로 입력 및 출력을 교환하는 상호작용 시스템을 연구개발하였다. 실시간 입력을 반영하여 결과 화면을 출력하기 위해서는, 실시간으로 결과 동작을 생성하는 과정에 필요한 계산량을 최소화해야 한다. 따라서 전처리 과정을 따로 두어 예제 동작 데이터의 정보를 사전에 학습하고, 예제 데이터를 특징 모델을 사용하여 동작 탐색 및 생성을 신속하게 수행할 수 있도록 하였다.

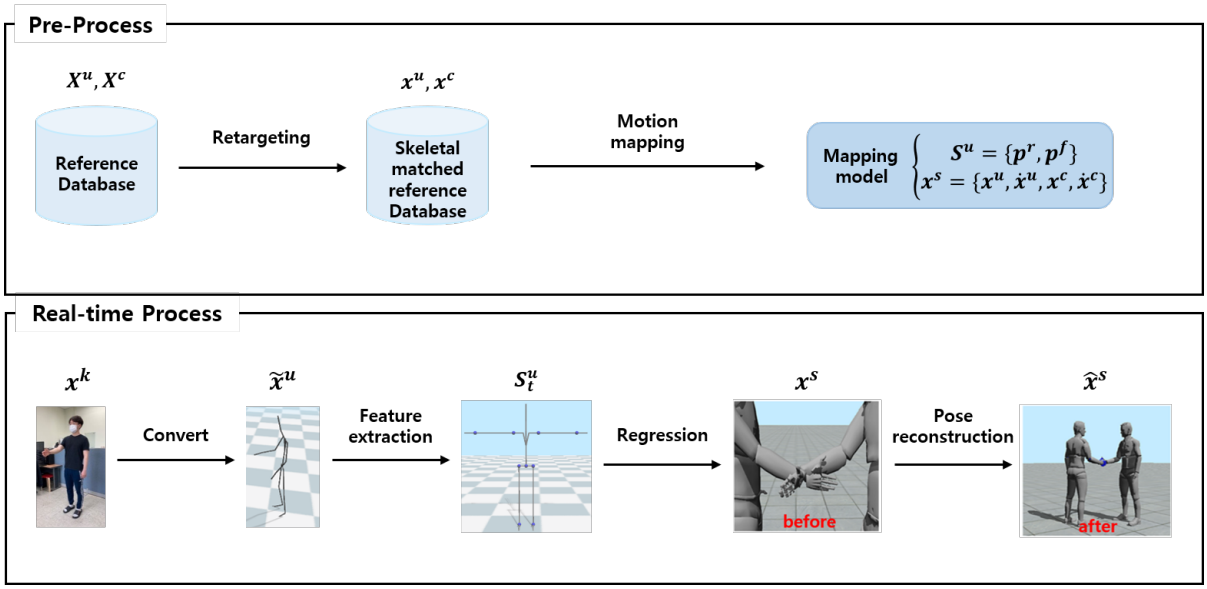


Figure 2: Overview of the real-time interactive animation system.

3. 시스템 흐름도

이 장에서는 본 논문에서 연구개발한 보급형 동작 인식 장비를 이용한 실시간 상호작용 애니메이션 시스템의 전반적인 구동 과정을 설명한다(Figure 2 참조). 해당 상호작용 애니메이션 시스템은 사용자가 키넥트를 이용해 실시간으로 입력한 자세 정보를 입력으로 받으며, 시스템의 작동을 거쳐 사용자 캐릭터와 상대 캐릭터, 즉 두 캐릭터가 상호작용 동작을 수행하는 애니메이션을 결과화면으로 출력한다. 또한 시스템의 작동은 크게 예제 데이터의 전처리, 캐릭터 동작의 실시간 생성 과정을 거쳐 이루어진다.

전처리 과정은 예제 동작 데이터 정보를 사전에 관측 및 분석하는 과정으로, 이는 실시간 작동 과정에서 예제 동작 데이터를 참조하고 본 애니메이션 시스템에 적용시키는 과정을 한층 더 신속하게 처리하기 위해 필수적이다. 골격 매칭(skeleton matching) 단계에서는 예제 데이터 내의 캐릭터 모델과 본 애니메이션 시스템에서 사용하는 캐릭터 모델의 골격 구조를 대응시킨다. 또 동작 맵핑(motion mapping) 단계에서는 예제 동작 정보를 가상 환경에 빠르게 적용할 수 있도록 단순화된 정보로 변환시켜 저장하는데, 이러한 동작 정보 및 맵핑 관계가 담긴 맵핑 모델(mapping model)을 생성한다.

실시간 처리 과정에서는 사용자의 실시간 입력에 맞는 두 캐릭터의 자세(동작)을 실시간으로 생성하고, 자연스러운 동작 연출을 위한 자세 보정 후 최종 결과 애니메이션을 화면에 출력한다. 리그레션(regression) 단계에서는 사용자의 실시간 입력을 전처리 과정에서 생성된 맵핑 모델로 전달하여 사용자 입력에 맞는 캐릭터 자세에 관한 특징 변수를 정의하고 초기 자세를 생성한다. 다음으로 자세 재건(posture reconstruction) 단계에서는 리그레션 단계에서 얻은 초기 자세를 토대로 충돌 감지 및 역기구학 등 자연스러운 자세 보정을 거쳐 최종 애니메이션을 생성한다.

본 연구의 애니메이션 시스템의 작동 과정을 구체적으로 설명하기에 앞서, 본 논문의 4장에서는 키넥트로부터 입력된 자세 정보를 본 연구의 애니메이션 시스템에 적용하기 위한 입력 데이터 가공(변환) 과정을 설명한다. 이어 5장과 6장에서는 상호작용 동작 애니메이션 생성에 필요한 전처리 과정 및 실시간 과정을 각각 자세하게 설명한다.

4. 키넥트 입력 정보 처리

이 장에서는 시스템 작동에 앞서 시스템의 입력값인 키넥트 자세 정보를 가상 환경에 작동할 수 있도록 가공한 내용을 설명한다. 본 연구에서는 사용자의 실시간 자세(동작)를 입력받기 위한 입력 장치로서 키넥트를 사용하였다. 키넥트는 대표적인 보급형 동작 인식 장비로서, RGB 카메라와 적외선 카메라를 이용하여 사용자 자세에 대한 색상 및 깊이 정보를 인식한다. 키넥트로 추적한 사용자의 자세 및 동작 정보는 Figure 3과 같은 인체 캐릭터 모델의 형태로 전달된다. 해당 캐릭터 모델은 총 19개의 관절로 이루어져 있으며, 각 관절에 대한 위치 및 회전 각도 정보가 저장되어 있다. 키넥트 입력 정보인 19개의 관절 중 골반의 중심부에 위치한 'WAIST' 관절을 캐릭터 모델의 중심점, 즉 루트 관절로 간주하였다.

본 시스템은 사용자 자세 입력 과정에서 키넥트로부터 전달받은 정보 x^k 를 시스템 작동 환경에 맞게 가공한다. 키넥트의 자세 인식 정보는 왼손 좌표계(left-handed coordinate system)를 기준으로 수치화되어 있는데, 본 연구의 애니메이션 시스템은 오른손 좌표계(right-handed coordinate system) 환경에서 개발되었으므로 좌표계 변환이 필요하다. 따라서 키넥트로부터 전달된 자세 정보는 왼손 좌표계에서 오른손 좌표계로의 변환을 수행한다. 또한 키넥트 전달 정보는 전역 좌표계를 기준으로 관측된 값으로,

Figure 3: The joint names and skeleton structure of the human character tracked with Kinect.

이를 가상 환경에 그대로 사용할 시 사용자의 관측 위치(사용자가 키넥트 앞에 서있는 위치)가 특정 위치로 온전히 일치하지 않으면 사용자 캐릭터가 화면에 제대로 표시되지 않는다. 사용자가 키넥트의 위치에 구애받지 않고 자유롭게 시스템 작동에 참여할 수 있도록, 키넥트로 입력된 캐릭터의 좌표와 가상 환경의 좌표 간의 오차인 오프셋(offseet)을 보정하여 사용자 캐릭터가 화면의 가운데 및 가상 환경의 지표면 위에 올바르게 표시되도록 하였다. 앞서 언급한 Figure 3의 캐릭터와 해당 시스템에서 사용되는 캐릭터인 Figure 4의 관절 구조는 비슷하지만 정확히 일치하지 않는다. 따라서 시스템 캐릭터의 각 관절 정보는 키넥트로 추적된 사용자의 각 관절 중 가장 가까운 위치에 있는 관절 정보를 사용한다. 마지막으로 키넥트 입력 정보는 모든 관절의 위치 정보와 회전 정보를 포함하는데, 가상환경의 캐릭터 동작을 제어하는 데에는 루트 관절의 위치 및 회전 정보와 나머지 관절의 회전 정보만을 필요로 한다. 따라서 키넥트 입력 정보 중 필요한 해당 정보만을 사용하되, 루트 관절 이외의 관절에 대한 정보는 루트 관절에 대하여 상대적인 회전값으로 변환하여 캐릭터 모델 제어에 사용한다. 위 과정을 거쳐 사용자의 동작이 시스템 내 사용자 캐릭터의 자세 $\tilde{x}^u$ 를 생성한다.

5. 예제 동작 데이터 학습

이 장에서는 상호작용 애니메이션 시스템의 작동 과정 중 전처리 과정 전반에 걸쳐 자세히 설명한다. 전처리 과정에서는 예제 동작 데이터 정보를 사전에 관측 및 분석하며, 이는 실시간 작동 과정에서 예제 동작 데이터를 참조하고 본 애니메이션 시스템에 적용시키는 과정을 한층 더 신속하게 처리하기 위해 필수적이다. 이 장의 5.1절에서는 예제 데이터 내의 캐릭터 모델과 본 애니메이션 시스템에서 사용하는 캐릭터 모델의 골격 구조를 대응시키는 골격 매칭 과정을 설명하고, 5.2절에서는 예제 동작 정보를 가상 환경에 빠르게 적용할 수 있도록 단순화된 정보로 변환시켜 저장하는 맵핑 모델 생성 과정을 설명한다.

5.1 캐릭터 모델간의 골격 매칭

본 논문에서 연구개발한 애니메이션 시스템은 2인 상호작용 동작 데이터를 예제 데이터로서 참조하고 이를 본 시스템의 캐릭터 모델에 적용함으로써 결과 애니메이션을 생성한다. 예제 데이터에서 사용된 캐릭터 모델은 일반적으로 시스템 내 키넥트의 입력으로 동작하는 사용자 캐릭터 모델과 그 골격 구조 및 크기가 다르다. 따라서 예제 데이터를 본 시스템의 캐릭터 모델에 적용하기에 앞서 예제 데이터의 캐릭터 모델과 결과 애니메이션에 사용될 시스템 내 캐릭터 모델의 골격 구조를 매칭 시켜야 한다. 이렇듯 골격 구조가 다른 두 모델이 서로 정보를 호환하여 사용할 수 있도록 서로 상응하는 관절끼리 매칭 시키는 것을 본 논문에서는 '골격 매칭'이라 한다.

전처리 과정의 첫 번째 작업으로서 예제 데이터의 캐릭터 모델과 Figure 4의 시스템 내 캐릭터 모델 간의 골격 구조를 매칭시키는 단계를 진행하였다. 본 연구의 애니메이션 시스템에서는 동작을 수행하는 주체로서 Figure 4와 같은 골격 구조를 가진 캐릭터 모델을 사용하였는데, 이는 총 21개의 관절로 이루어져 있다. Figure 4에서 파란색으로 표시된 9개의 관절은 특징 관절(feature joints)을 표시한 것으로, 동작 생성 시 자세를 결정하는 주요 관절로 취급한다.

두 캐릭터 모델 간의 관절 매칭은 수작업으로 서로 대응하는 관절들을 1대1로 매칭시켜 진행하였다. 관절 매칭 수작업의 편의를 개선하기 위해, Figure 5와 같은 인터페이스를 개발하였다. 해당 인터페이스는 화면에 두 캐릭터 모델을 동시에 띄워두고 서로 대응되는 관절을 차례로 선택하고 매칭 정보를 저장할 수 있다. 기존에 매칭된 관절쌍에 대하여 수정 및 삭제도 가능하며, 몇 번의 단순 클릭 반복으로 손쉽게 관절 매칭을 진행할 수 있다. 해당 기능 및 인터페이스는 시스템으로부터 독립하여 사용할 수도 있으며, 추후 여러 가지 캐릭터 모델을 다루는 유사 연구에도 폭넓게 사용할 수 있다.

5.2 맵핑 모델 생성

골격 매칭을 통해 예제 데이터 내의 캐릭터 모델과 본 시스템의 캐릭터 모델 간의 관절 구조 매칭이 이루어지면, 다음으로는 예제 데이터에 포함된 각종 동작들을 정보화하여 저장한다. 본 연구의 상호작용 애니메이션 시스템에서는 2인 상호작용 동작으로 구성된 모션 캡처 데이터베이스를 예제 동작 데이터베이스로 사용하는데, 맵핑 모델 생성 과정에서는 이 예제 데이터 내에 있는 두 캐릭터의 관절 정보(위치, 회전 등)가 프레임(frame)별로 변화하는 양상을 몇 가지 변수를 사용하여 체계화한다. 즉 예제 데이터

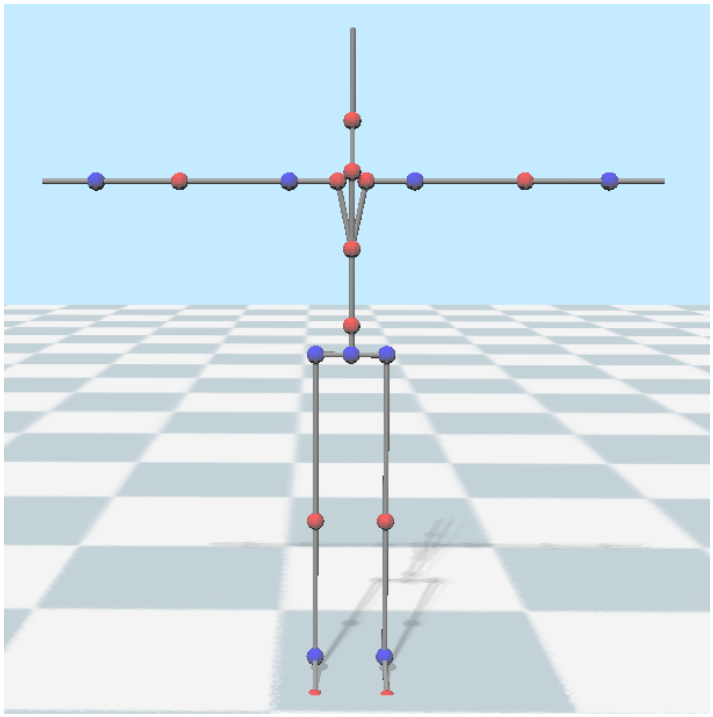


Figure 4: Character model and joint structure used in the system.

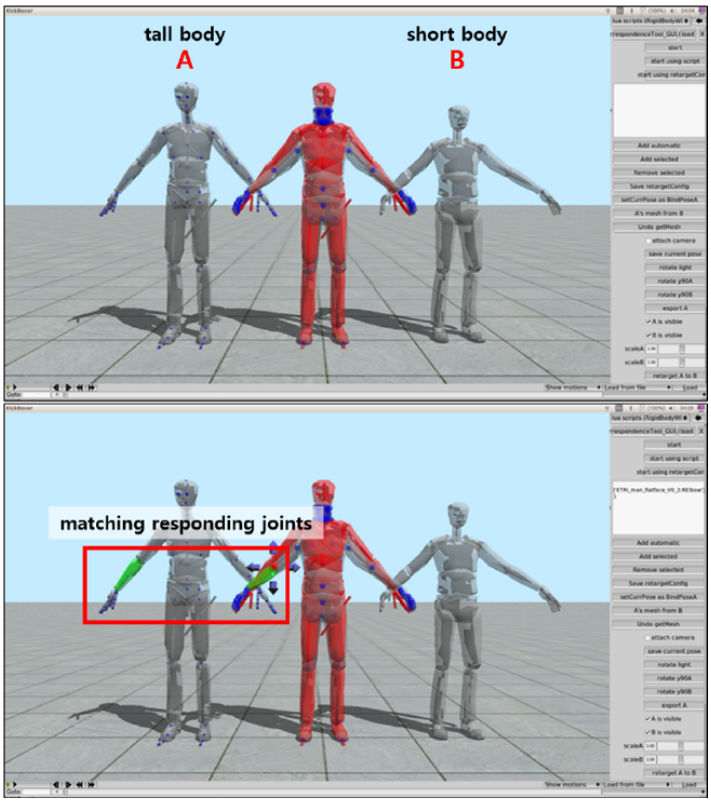


Figure 5: An example of the use of skeleton matching interface.

내에 있는 여러 가지 동작별로 각 관절의 상태정보 변화를 파악하여 그 특징을 데이터 형태로 저장해놓는 것인데, 쉽게 말하자면 동작 별로 관절 변화의 규칙을 세우는 과정이라 할 수 있다. 이 때 예제 데이터에 존재하는 모든 관절에 대한 정보를 저장하는 것이 아니라, 동작의 종류를 결정짓는 특징적 요소가 되는 관절에 한정하여 단순화된 동작 정보를 저장한다.

맵핑 모델 생성 과정은 단순히 예제 데이터에 포함된 관절 정보를 복사하는 것이 아니라, 특징 모델 $S^u = \{p^r, p^f\}$ 에 동작의 종류를 결정짓는 최소한의 정보만을 추출하여 가공 및 저장하는 것이다. 여기서 특징 모델 S^u는 결과 애니메이션에서 사용될 캐릭터 모델의 전체 관절 구조 그대로가 아닌 9개의 특징 관절(Figure 4 참조)의 위치들을 지칭하는 것이다. 그리고 특징 모델은 사용자에 해당하는 캐릭터에서 추출된 특징 관절로 이루어져 있다. 예제 데이터의 캐릭터 관절 정보를 그대로 참조하여 결과 애니메이션을 생성하는 것은 그 차원이 높아 실시간 과정에서 자세를 계산하는 데 많은 계산량을 요구한다. 따라서 예제 데이터의 동작 정보에서 특징 모델을 추출하여 실시간 과정의 작동 속도를 높이고자 한다. 예제 데이터의 캐릭터 모델 기준에서 특징 모델 기준으로 예제 동작 정보를 재계산하므로, 본 논문에서는 이러한 과정을 편의상 동작 맵핑이라고 부르기로 한다. 이 과정에서 도출한 예제 데이터의 동작과 특징 모델의 변환 관계들을 맵핑 모델 M이라고 하는데, 맵핑 모델은 전처리 과정의 출력값이며 실시간 처리에서 사용된다.

$$x^s = M(p) \tag{1}$$

본 연구에서는 예제 데이터 캐릭터의 관절 위치 변화를 체계화하기 위해 동작의 종류를 결정짓는 특징 속성에 대한 매개변수를 몇 가지 설정하였다. Table 1은 사용자 캐릭터와 상대 캐릭터의 관절의 상태를 표시하는 상태 매개변수 $x^s = \{x^u, \dot{x^u}, x^c, \dot{x^c}\}$를 나열한 것이고, Table 2는 사용자 캐릭터와 상대 캐릭터의 상태를 제어하는 제어 매개변수 p^r, p^f를 나열한 것이다.

Table 1: The state parameters of the user character and counterpart character

parameters	explanation
x^u	user character pose
$\dot{x^u}$	velocity of the user character joints
x^c	counterpart character pose
$\dot{x^c}$	velocity of the counterpart character joints

Table 2: The control parameters that change the state of the user character and counterpart character

parameters	explanation
p^r	root position of the x^u
p^f	feature joint positions of the x^u relative to p^r

Table 1 에서 사용자 캐릭터 자세를 나타내는 $x^u = \{r^u, q^u\}$는 루트의 위치와 회전 정보를 표현하는 4×4 행렬 r^u와 나머지 관절의

각도 q^u를 포함한다. 상대방 캐릭터의 자세 $x^c = \{r^c, q^c\}$도 역시 루트의 위치 r^c와 나머지 관절의 각도 q^c를 가진다. 이때 상대 캐릭터의 루트 정보 r^c는 사용자 캐릭터의 루트 정보에 상대적으로 저장한다:

$$r^c \leftarrow (r^u)^{-1} r^c. \tag{2}$$

본 상호작용 애니메이션 시스템은 위와 같은 전처리 과정을 거쳐 결과적으로 맵핑 모델을 생성한다. 이는 예제 동작 데이터를 프레임 단위로 사용자 캐릭터와 상대 캐릭터의 상태 매개변수 형태로 저장하고 예제 동작과 특징 모델 사이의 변환 규칙을 정의하는 제어 매개변수들로 이루어진 일종의 동작 정보 패키지라고 볼 수 있다. 생성된 맵핑 모델은 다음 장에서 설명하는 두 캐릭터의 상호작용 동작을 생성하는 실시간 과정에서 초기 자세를 설정하는 데 사용된다.

6. 실시간 상호작용 애니메이션 생성

이 장에서는 상호작용 애니메이션 시스템의 작동 과정 중 실시간 처리 과정 전반에 걸쳐 자세히 설명한다. 실시간 처리 과정에서는 사용자의 실시간 입력에 맞는 두 캐릭터의 자세(동작)을 실시간으로 생성하고, 자연스러운 동작 연출을 위한 자세 보정 후 최종 결과 애니메이션을 화면에 출력한다. 이 장의 6.1절에서는 사용자의 실시간 입력에 맞게 캐릭터 자세에 관한 매개변수를 정의하는 리그레션 과정을 설명하고, 6.2절에서는 특징 모델로부터 사용자 캐릭터와 상대 캐릭터 자세를 계산 및 보정하여 최종 애니메이션을 생성하는 자세 재건 과정을 설명한다.

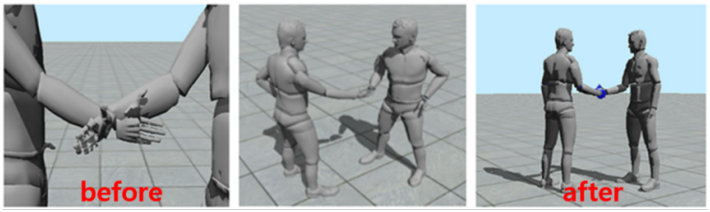
6.1 리그레션

리그레션은 사용자의 실시간 자세 입력에 따라 사용자 캐릭터와 상대 캐릭터의 상호작용 동작을 생성하는 과정의 첫 번째 단계이다. 이 단계에서는 전처리 과정에서 생성된 맵핑 모델에 따라 단순화된 특징 모델 S^u를 토대로 제어 매개변수를 측정하고 KNN-IDW를 이용하여 상태 매개변수를 측정한다[18]. 실시간 사용자 입력 자세로부터 특징 모델을 추출하고 이 특징 모델이 가지고 있는 정보인 제어 매개변수 p^r, p^f를 맵핑 모델의 입력으로 전달하면 출력으로 상태 매개변수인 $x^s = \{x^u, \dot{x^u}, x^c, \dot{x^c}\}$을 얻는다.

다음 프레임에 취해야 할 자세는 목표 시간 내에 취해야 할 목표 자세(동작)에 따라 결정되는데, 실시간 동작에서는 사용자의 입력 동작에 따라 목표 자세 및 다음 프레임에 취해야 할 자세가 결정될 것이다. 상태 매개변수를 계산하는 데에는 K 근접 역거리 가중치 기법(K-nearest neighbor inverse distance weighting, KNN-IDW)을 사용하였는데, 해당 방식은 KD 트리 형태의 자료구조를 빠르게 탐색하는 데 적합하기 때문이다. 이는 비선형 계산이 이루어지는 리그레션 과정에 유리하다.

다음은 사용자 캐릭터와 상대 캐릭터가 현재 자세로부터 다음 프레임의 자세를 취하기 위한 제어 매개변수와 상태 매개변수의 관계를 보여준다. 여기서 사용되는 모든 입력 및 출력 매개변수는 예제 데이터로부터 참조된 것으로, 사전에 벡터 형식으로 변환된 값이다. 제어 매개변수를 얻기 위하여 프레임 t의 사용자 입력 자세 $\tilde{x}_t^u$에서 특징 모델을 벡터 형태인 S_t^u로 변환한다. 이때 t의 범위를 현재 t, 과거~현재 $t-w:t$, 과거~미래 $t-w/2:t+w/2$와 같이 3가지 경우로 나눌 수 있다. 예를 들어 3번째 경우 특징 벡터는 $S^u_{t-w/2:t+w/2} \in \left\{S^u_{t-w/2}, S^u_{t-w/2+1}, \cdots, S^u_{t+w/2}\right\}$로 나타낼 수 있다. 첫 번째 경우를 제외한 나머지 경우들은 특징 벡터의 크기 w만큼의 딜레이(delay)가 있다. 그리고 세 번째 경우는 미래 프레임을 보기 때문에 완전히 실시간으로 진행할 수 없고 사전에 녹화된 동작 정보를 사용한다. 특징 벡터는 제어 매개변수인 $p^r_{t-w/2:t+w/2}$와 $p^v_{t-w/2:t+w/2}$를 포함하고 맵핑 모델의 입력으로 전달된다. 상태 매개변수 x^s는 맵핑 모델의 출력으로 얻을 수 있다. x^s에는 사용자 캐릭터 자세 x^u와 상대 캐릭터 자세 x^c의 정보가 포함되어 있는데, 이때 x^c의 루트는 사용자 캐릭터 x^u의 루트에 상대적으로 저장되어 있기 때문에 수식 3를 이용하여 전역 좌표계 값으로 돌려주어 리그레션 단계를 마친다:

$$r^c \leftarrow r^u r^c. \tag{3}$$



(a) (b) (c)

Figure 6: (a) shows the hand skin of the two characters penetrate without contact in the process of generating a handshake motion, and (b), (c) show the final pose corrected through collision detection and inverse kinematics.

6.2 자세 재건

자세 재건 단계에서는 이전 리그레션 단계에서 얻은 결과 x^s에 포함된 자세 정보 x^u, x^c를 결과 화면에 출력될 사용자 캐릭터와 상대 캐릭터에 적용하여 자세를 생성한다. 이 때 계산된 캐릭터 전체 관절의 자세는 초기 자세로, 사용자의 입력 의도에 맞는 동작 및 상호작용 패턴을 묘사한다. 초기 자세만 적용한 경우 결과 모션의 품질이 불안정하였다. 이것을 개선하기 위해 x^s에 포함된 사용자와 상대 캐릭터 관절의 속도 정보 $\dot{x^u}$, $\dot{x^c}$를 이전 자세에 적분하여 얻은 자세 정보와 x^u, x^c를 0.95 : 0.05 비율로 혼합하여 적용하였다. 이 결과에 가우시안 필터를 추가로 적용하여 결과 모션의 품질을 상당부분 개선하였다.

초기 자세는 캐릭터 모델의 관절 구조, 즉 뼈대를 기준으로 설계된 위치이다. 따라서 두 캐릭터의 상호접촉이 있는 동작의 경우 접촉면의 위치가 관절 기준으로 계산된다. 이 때 모델링 과정에서 부피가 있는 사람 형태의 스킨을 씌우면, 관절(뼈대) 간의 접촉을

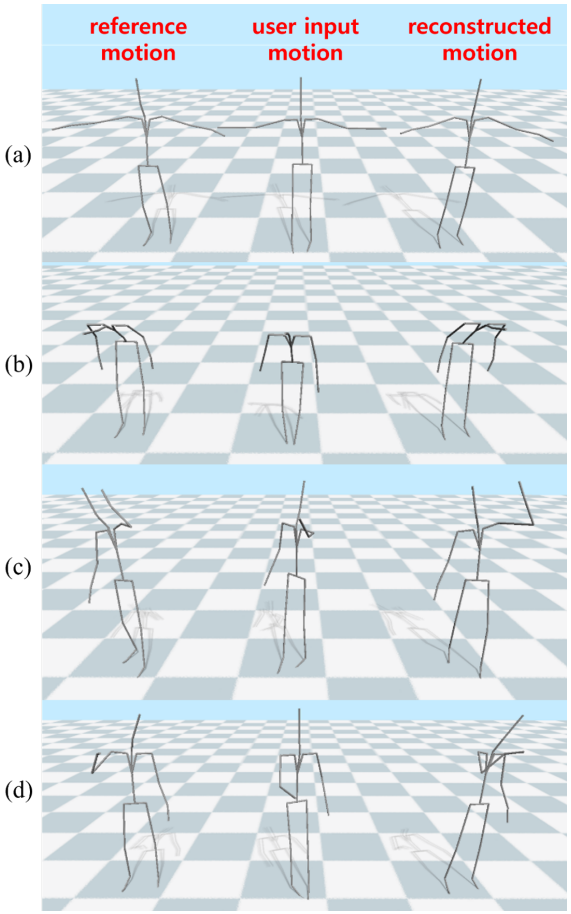


Figure 7: Comparison between reference motion, user input motion, and reconstructed motion.

달성하기 위해 두 캐릭터의 스킨(피부 표면)이 정확하게 맞닿지 않거나 관통하는 현상이 발생하기도 한다(Figure 6 참조). 비록 초기 자세가 정확하게 계산되었다고 하더라도 경우에 따라 최종 결과 애니메이션에서는 부자연스럽게 연출되는 상황이 발생할 수 있다. 이러한 상황을 방지하기 위해 상호접촉이 발생하는 관절을 엔드 이펙터(end-effectors)로 설정하고 충돌 감지 기술을 더하여 자세 오류를 발견한다.[19] 해당 오류에 대하여 [19]에서 제안한 충돌 회피 기술을 통해 자연스러운 접촉이 가능한 자세로 관절 위치를 재계산 한 후에 역기구학 풀이를 통해 캐릭터의 자세를 수정하고, 그 결과 결과화면에 출력될 최종 자세 $\hat{x^s}$가 생성된다. Figure 6은 이러한 자세 보정 과정의 예를 보여준다. 좌측 화면에서는 악수 동작 생성 과정에서 초기 자세 모델링 결과 두 캐릭터의 손 표면이 접촉하지 않고 관통하는 현상이 발생하였다. 중앙 및 우측 화면에서는 충돌 회피를 통해 수정된 자세로 자연스러운 악수 동작을 연출한다.

Figure 7은 여러 가지 동작에 대해서 예제 동작 데이터의 자세, 사용자의 키넥트 입력 자세, 그리고 본 애니메이션 시스템을 통해 재건된 최종 자세를 비교한 것이다. (a)는 기본자세인 T 자세를 취한 것으로, 사용자는 팔이 다소 구부러진 자세를 취하고 있으나 자세 재건 결과 예제 자세처럼 팔을 올곧게 뻗은 자세로 보정되었다. (b)는 허리 숙여 인사하는 자세, (c)는 손을 높이 들어 내미는 하이파이브 자세, (d)는 손을 적당한 높이로 들어 내미는 주먹 인사 자세를 취한 것으로, 마찬가지로 사용자의 다소 일그러진 자세를 입력하더라도 예제 자세를 참조하여 자연스럽게 보정된 자세가 생성되는 것을 확인할 수 있다.

7. 실험 결과

이 장에서는 본 연구의 실시간 상호작용 애니메이션 시스템의 작동 및 성능에 관한 실험 내용 및 그 결과를 소개한다. 본 애니메이션 시스템을 개발 및 작동 실험을 수행하는 데에는 32GB RAM, Intel® Core™ i7-10700k CPU(8 Cores) 3.80GHz, GeForce RTX 2080 Ti로 구성된 PC 1대를 사용하였다. 사용자의 자세(동작) 입력 장치로는 대표적인 보급형 동작 인식 장비인 마이크로소프트사의 키넥트 2012년 모델(Kinect V1)을 사용하였고, 소프트웨어 개발 도구로는 NuitrackSDK를 사용하였다. 예제 동작 데이터는 120fps로 캡처되었고 총 프레임의 길이는 18,547이었으나 Kinect V1의 스펙에 맞추어 30fps로 샘플링하여 사용했다. 내용은 두 연기자가 90도 인사, 악수, 손 흔들어 인사 등 상호작용 동작을 연출하는 장면을 포착한 2인 동작 데이터로 구성되어 있다.(첨부 동영상 참고) 애니메이션에 사용된 캐릭터 모델은 이전에 언급한 Figure 4와 같은 구조를 갖고 있다.

Figure 8은 사용자가 직접 키넥트를 통해 실시간으로 동작을 입력하고 이에 본 상호작용 애니메이션 시스템이 실시간으로 작동한 결과 화면을 보여준다. 결과 품질은 6.1절에서 언급된 프레임 t의 범위에 따라 차이를 보였는데 현재 프레임 t만 고려하는 첫 번째 경우의 품질이 가장 낮았다. 해당 Figure는 품질이 가장 좋았던 과거~미래 프레임을 고려하는 세 번째 경우의 결과이다. 각 화면에서 중앙에 있는 캐릭터는 사용자 캐릭터로 사용자의 키넥트 입력 동작에 의해 제어되며, 오른쪽에 있는 캐릭터는 상대 캐릭터로 사용자 캐릭터의 동작에 따라 반응 동작을 수행한다. 좌측에 있는 캐릭터는 본 시스템의 작동 성능 확인을 위한 실험용 캐릭터로 사용자의 키넥트 입력 정보를 그대로 출력한다. 즉 좌측의 캐릭터는 사용자의 입력 동작을 그대로 모사하며, 중앙의 캐릭터는 사용자의 입력 동작이 본 애니메이션 시스템에 의해 보정된 최종 결과 동작을 수행하는 셈이다. 또 우측의 캐릭터는 사용자 캐릭터의 동작에 따라 결정된 반응 동작을 수행하는데, 사용자 캐릭터가 특정 동작을 취하면 상대 캐릭터가 대응하는 반응 동작을 취함으로써 결과적으로 두 캐릭터가 상호작용 동작을 연출한다.

Figure 8(a)은 사용자 캐릭터가 상대 캐릭터에게 허리를 숙여 정중하게 인사하자 상대 캐릭터 또한 함께 허리를 숙여 인사하는 장면이다. Figure 8(b) 및 (c)는 사용자 캐릭터가 손을 들고 흔들어 인사하자 상대 캐릭터 또한 함께 손을 들고 흔들어 인사하는 장면이다. (b)는 사용자 캐릭터가 손을 낮게 들어 흔들고, (c)는 손을 높이 들어 흔드는데, 상대 캐릭터는 사용자 캐릭터가 손을 흔드는 높낮이에 맞춰서 유사하게 반응한다. Figure 8(d)는 두 캐릭터가

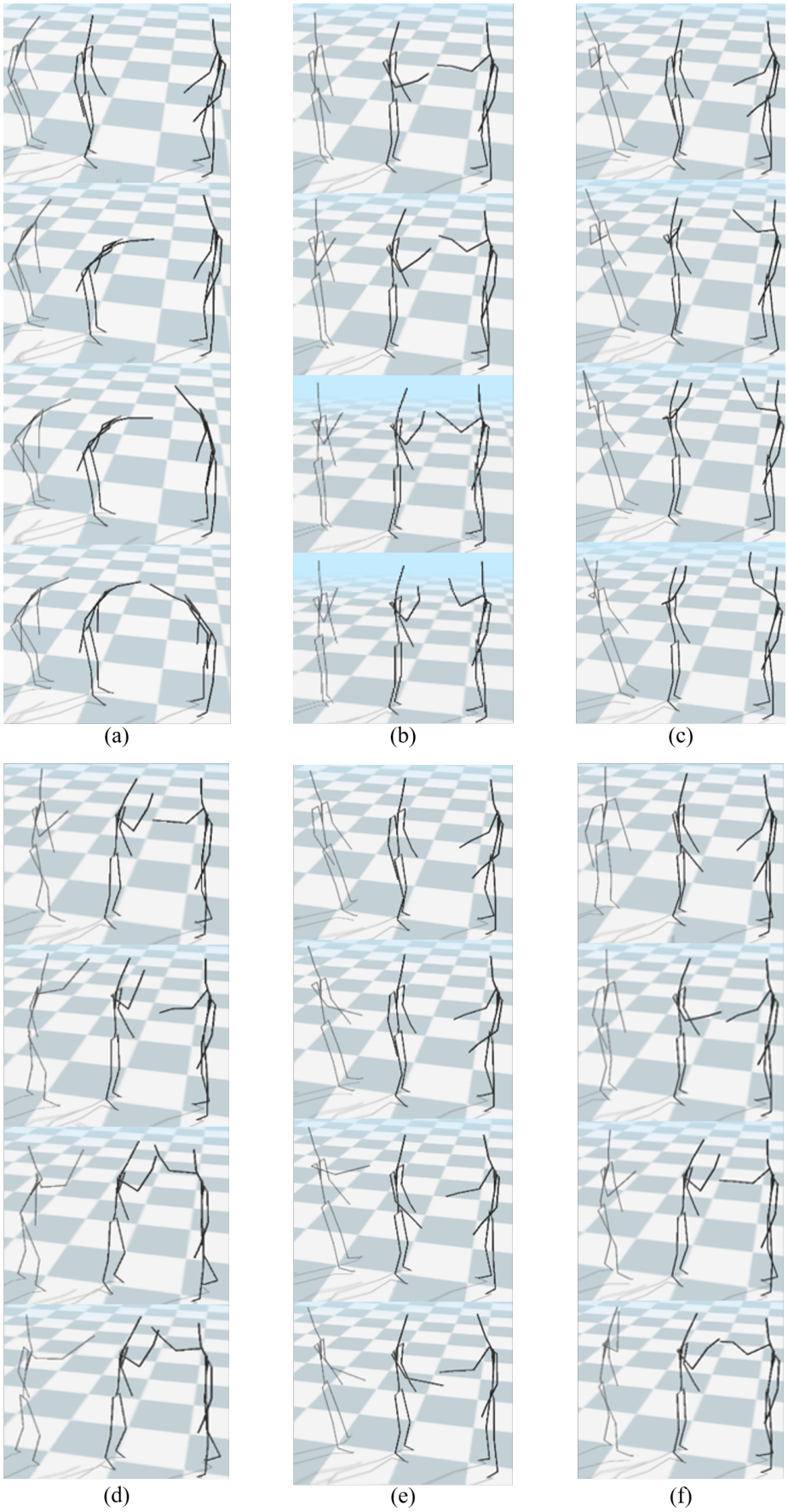


Figure 8: (a)∼(f) show the experimental results for various input motions.

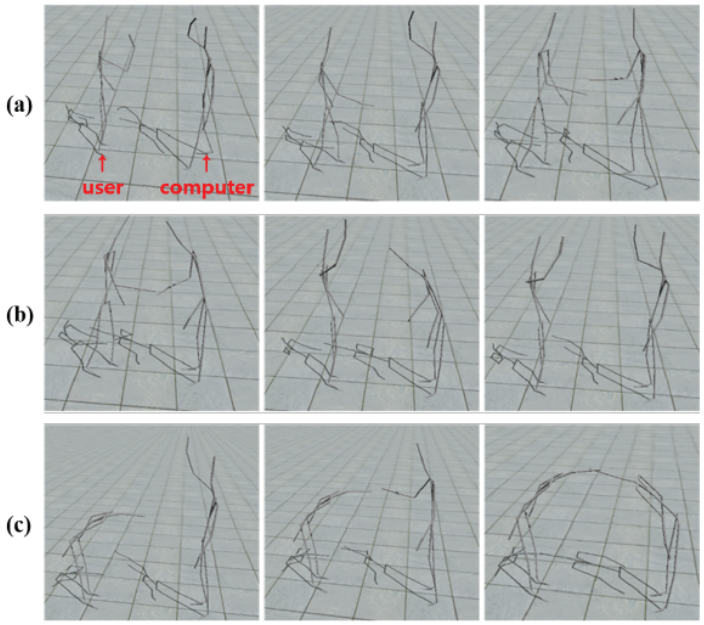


Figure 9: Counterpart character motions responding to the user character motion that changes in real time.

함께 하이파이브(손바닥 맞대기)를 하는 장면, (e)는 악수하는 장면, (f)는 주먹 인사(주먹 맞대기)를 하는 장면이다. (d)~(f)는 두 캐릭터가 각각 한 손을 서로 접촉시키는 상호작용 동작을 수행하는데, 동작의 종류에 따라 두 손을 접촉하는 높낮이 및 방법이 달라진다. 실험용 캐릭터(키넥트 입력)와 사용자 캐릭터는 전반적으로는 비슷하지만 미세하게 다른 포즈를 취하고 있는데, 이는 사용자의 입력 자세가 본 애니메이션 시스템에 의해 가상 환경에 맞게 보정된 것을 보여준다. 또 사용자 캐릭터와 상대 캐릭터는 상호 접촉이 포함된 상호작용의 경우에도 관통이나 미끄러지는 현상 없이 올바르게 접촉하는 상호작용 동작을 보여준다. 이러한 결과 화면을 통해 본 실시간 상호작용 애니메이션 시스템은 사용자의 입력 동작에 맞추어 상대 캐릭터는 적절하고 대응 동작을 수행하고, 화면상의 두 캐릭터가 서로 상호작용 동작을 연출하는 것을 확인할 수 있다.

Figure 9는 사용자가 실시간으로 특정 동작의 입력 도중 다른 동작으로 바꾸었을 때 시스템의 작동 결과 화면을 보여준다. 각 화면에서 좌측의 캐릭터는 사용자 캐릭터이고, 우측의 캐릭터는 상대 캐릭터이다. 상대 캐릭터는 사용자 캐릭터의 동작을 확인한 후 적절한 대응 동작을 취하는데, 해당 Figure은 사용자 캐릭터가 동작 수행 도중 다른 동작으로 바꾸어도 상대 캐릭터는 기존 동작의 대응 동작을 수행하다가 바뀐 동작의 대응 동작으로 바꾸어 수행하는 것을 확인할 수 있다. Figure 9(a)는 사용자 캐릭터와 상대 캐릭터가 함께 손을 들어 흔들며 인사를 나누던 도중, 사용자 캐릭터가 손을 내려 악수를 청하자 상대 캐릭터도 마찬가지로 손을 내려 사용자 캐릭터와의 악수를 위해 손을 뻗는 장면이다. (b)는 사용자 캐릭터와 상대 캐릭터가 함께 악수를 하는 도중, 사용자 캐릭터가 손을 들어 올려 인사를 하자 상대 캐릭터도 악수를 하던 손을 들어 사용자 캐릭터와 인사를 나누는 장면이다. (c)는 사용자 캐릭터와 상대 캐릭터가 함께 손을 들어 흔들며 인사를 하는 도중, 사용자 캐릭터가 허리를 숙여 정중하게 인사하자 상대 캐릭터도 인사를 하던 손을 내리고 허리를 숙여 인사를 하는 장면이다. 이러한 결과 화면을 통해 본 실시간 상호작용 애니메이션 시스템은 사용자의 실시간 입력에 따라 상대 캐릭터의 대응 동작 또한 실시간으로 변화하며, 동작이 변화할 때마다 동작이 부드럽게 연결되어 자연스러운 연속 동작을 연출하는 것을 확인할 수 있다.

8. 결론

본 논문에서는 대표적인 보급형 장비인 키넥트를 활용하여 실시간으로 사용자 캐릭터의 자세를 제어하고, 상대 캐릭터와 함께 자연스러운 상호작용 동작을 수행하는 실시간 상호작용 애니메이션 시스템을 소개하였다. 해당 상호작용 애니메이션 시스템은 실시간으로 두 캐릭터의 상호작용 동작을 연출하는 시스템으로, 사용자는 키넥트를 이용한 자세 입력을 통해 사용자 캐릭터의 동작을 제어하고 상대 캐릭터는 사용자 캐릭터의 동작에 따라 반응하는데 이 반응 동작은 시스템에 의해 자동으로 결정된다. 전처리 과정은 예제 동작 데이터 정보를 사전에 관측 및 분석하여 맵핑 모델을 생성하고, 실시간 처리 과정에서는 사용자의 실시간 입력에 맞는 두 캐릭터의 자세(동작)을 실시간으로 생성 및 보정 후 최종 결과 애니메이션을 화면에 출력한다. 실험 결과를 통해 본 해당 시스템은 사용자의 입력 동작에 맞추어 상대 캐릭터는 적절하고 대응 동작을 수행하고, 화면상의 두 캐릭터가 서로 상호작용 동작을 연출하는 것을 확인할 수 있다.

본 연구는 보급형 동작 인식 장비 및 관련 소프트웨어 시장의 성장에 발맞추어 보급형 장비의 성능적 한계를 보완하고 실시간으로 상호작용 애니메이션을 연출하는 시스템의 구현을 시도하였다. KNN 알고리즘을 이용한 동작 탐색 및 기타 동작 합성 및 보정 기술을 통해 애니메이션 동작 품질을 높이되, 실시간 작동을 위하여 최적화된 시스템 구성을 설계하였다. 이러한 시스템을 통해 사용자는 직접 동작을 입력하고 사용자 캐릭터의 동작에 따라 상대 캐릭터가 실시간으로 반응 동작을 수행하는 것을 확인할 수 있다. 사용자와의 실시간 상호작용을 통해 두 캐릭터가 인사, 악수, 하이파이브 등의 상호작용 동작을 수행할 수 있으며, 자세의 높낮이나 각도 변화에도 자연스럽게 대응할 수 있다. 본 논문에서 제안하는 기술 및 아이디어는 응용하여 실제 사용자 상호작용 소프트웨어 개발에 적용할 수 있고, 이를 통해 사용자에게 더 나은 몰입감을 제공할 수 있을 것이다.

시스템의 성능을 실험하는 과정에서 몇 가지 한계를 발견하였는데, 첫 번째는 사용자의 급격한 입력 동작 변화에 대응하지 못한다는 것이다. 악수 동작의 경우, 일반적으로 사용자는 우선 손을 앞으로 내민 후에 손을 잡는 흉내를 내고 (허공에서 멈춤) 흔드는 듯한 동작을 취할 것이다. 그러나 손을 내밀어 허공에서 멈춘 동작을 수행하다가, 손을 흔드는 대신 갑자기 팔을 들어 올

려 손을 흔드는 인사 동작으로 전환하는 상황을 가정해보자. 이러한 동작 입력에 대해 본 시스템은 초반의 손을 내밀어 허공에서 멈춘 동작을 토대로 악수 동작을 수행할 것이라고 추론하게 된다. 따라서 후반에 동작 입력이 인사 동작으로 바뀌는 것을 노이즈로 인식하여 무시하고 사용자 캐릭터에 악수 동작을 합성하여 출력한다. 입력의 동작의 변환이 천천히 이루어질 경우에는 사용자의 캐릭터 또한 동작을 바꾸어 출력하지만, 변환 속도가 급격할 경우에는 이러한 입력 변화가 무시되는 경향이 있다. 두 번째 한계점으로는 본 시스템을 통해 구현된 상호작용 동작의 난이도와 종류가 한정적이라는 점을 들 수 있다. 시스템 구현에는 인사, 악수 등의 간단한 상호작용 동작이 포함된 데이터베이스를 사용하였고, 실험 결과의 품질 또한 만족스러운 수준이다. 그러나 인사와 악수는 두 캐릭터가 마주보고 같은 동작을 취하는 모양새의 상호작용으로, 상대 캐릭터가 적절한 반응 동작을 선출하는 것이 아니라 단순히 사용자 캐릭터를 모방하고 있는 듯한 의구심을 자아낸다. 본 시스템의 성능을 보다 명확하게 증명하기 위해서는, 보다 다양한 상호작용 동작들을 포함한 데이터베이스를 활용하여 실험 난이도를 높일 필요가 있다. 마지막으로, 커플 댄스나 싸우는 동작 등의 관절의 움직임이 복잡한 상호작용 동작에 대해서는 결과 애니메이션의 품질이 매우 낮아진다. 이는 동작 정보 분석 및 단순화 과정에서 자세 정보를 축약하기 때문에 발생하는 문제로, 계산량을 축소함으로써 실시간 애니메이션 시스템 구동을 위한 것이다. 추후에 보다 복잡한 동작이 연출 가능한 시스템으로 발전시키기 위해서는, 적은 정보로 복잡한 동작을 묘사할 수 있는 새로운 캐릭터 모델의 고안이나 효율적인 알고리즘 설계가 추가적으로 필요할 것이다.

감사의 글

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(NRF-2019R1A4A1029800, NRF-2020R1A2C1012847) 및 정보통신기획평가원의 지원(No.2021-0-00320, 실 공간 대상 XR 생성 및 변형/증강 기술 개발)을 받아 수행된 연구임.

References

[1] Z. Zhang, “Microsoft kinect sensor and its effect,” *IEEE multimedia,* vol. 19, no. 2, pp. 4-10, 2012.

[2] A. Zabatani, V. Surazhsky, E. Sperling, S. B. Moshe, O. Menashe, D. H. Silver, Z. Karni, A. M. Bronstein, M. M. Bronstein, and R. Kimmel, “Intel® realsense™ sr300 coded light depth camera,” *IEEE transactions on pattern analysis and machine intelligence,* vol. 42, no. 10, pp. 2333–2345, 2019.

[3] T.-h Kim, S. I. Park, and S. Y. Shin, “Rhythmic-motion synthesis based on motion-beat analysis,” *ACM Transactions on Graphics (TOG),* vol. 22, no. 3, pp. 392-401, 2003.

[4] M. R. Jones and M. Boltz, “Dynamic attending and responses to time,” *Psychological review,* vol. 96, no. 3, p. 459, 1989.

[5] E. Hsu, S. Gentry, and J. Popović, “Example-based control of human motion,” in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics symposium on Computer animation,* pp. 69-77, 2004.

[6] T. Kwon, Y.-S. Cho, S. I. Park, and S. Y. Shin, “Two-character motion analysis and synthesis,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 14, no. 3, pp. 707-720, 2008.

[7] H. P. Shum, T. Komura, and S. Yamazaki, “Simulating competitive interactions using singly captured motions,” in *Proceedings of the 2007 ACM symposium on Virtual reality software and technology,* pp. 65-72, 2007.

[8] H. P. Shum, T. Komura, and S. Yamazaki, “Simulating interactions of avatars in high dimensional state space,” in *Proceedings of the 2008 Symposium on interactive 3D Graphics and Games,* pp. 131-138, 2008.

[9] H. P. Shum, T. Komura, M. Shiraishi, and S. Yamazaki, “Interaction patches for multi-character animation,” *ACM transactions on graphics (TOG),* vol. 27, no. 5, pp. 1-8, 2008.

[10] H. P. Shum, T. Komura, and S. Yamazaki, “Simulating multiple character interactions with collaborative and adversarial goals,” *IEEE Transactions on Visualization and Computer Graphics,* vol. 18, no. 5, pp. 741-752, 2010.

[11] K. Wampler, E. Andersen, E. Herbst, Y. Lee, and Z. Popović, “Character animation in two-player adversarial games,” *ACM Transactions on Graphics (TOG),* vol. 29, no. 3, pp. 1-13, 2010.

[12] J. Lee and K. H. Lee, “Precomputing avatar behavior from human motion data,” *Graphical models,* vol. 68, no. 2, pp. 158-174, 2006.

[13] E. S. Ho and T. Komura, “A finite state machine based on topology coordinates for wrestling games,” *Computer Animation and Virtual Worlds,* vol. 22, no. 5, pp. 435-443, 2011.

[14] E. S. Ho and T. Komura, “Character motion synthesis by topology coordinates,” *Computer Graphics Forum,* vol. 28, no. 2. pp. 299-308, 2009.

[15] E. S. Ho, J. C. Chan, T. Komura, and H. Leung, “Interactive partner control in close interactions for real-time applications,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM),* vol. 9, no. 3, pp. 1-19, 2013.

[16] E. S. Ho, T. Komura, and C.-L. Tai, “Spatial relationship preserving character motion adaptation,” *ACM Transactions on Graphics (TOG),* vol. 29, no. 4, pp. 1-8, 2010.

[17] R. Kulpa, F. Multon, and B. Arnaldi, “Morphology-independent representation of motions for interactive human-like animation,” *Computer Graphics Forum,* vol. 24, no. 3, pp. 343-351, 2005.

[18] L. Kovar, M. Gleicher, “Automated extraction and parameterization of motions in large data sets,” *ACM Transactions on Graphics (TOG),* vol. 23, no. 3, pp. 559-568, 2004.

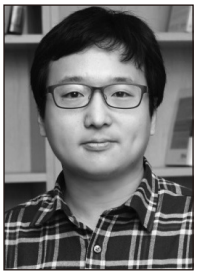
[19] J. Kim, Y. Seol, H. Kim, and T. Kwon, “Interactive character posing with efficient collision handling,” *Computer Animation and VirtualWorlds,* vol. 31, no. 3, p. e1923, 2020.

〈저자소개〉

김정호
- 2016-2020 한성대학교 IT응용시스템공학과 학사
- 2020-2022 한양대학교 컴퓨터소프트웨어학과 석사
- 관심분야: Character animation, Physics-based character control
- https://orcid.org/0000-0002-2973-0563

강다은
- 2010-2014 숙명여자대학교 컴퓨터과학 학사
- 2014-현재 한양대학교 컴퓨터소프트웨어학과 석박사 통합과정
- 관심분야: Character animation, Physics-based simulation
- https://orcid.org/0000-0003-2362-7669

이윤상
- 1999-2007 서울대학교 기계항공공학부 학사
- 2007-2014 서울대학교 컴퓨터공학부 박사
- 2014-2016 삼성전자 소프트웨어센터 책임연구원
- 2016-2018 광운대학교 소프트웨어학부 조교수
- 2018-현재 한양대학교 컴퓨터소프트웨어학부 부교수
- 관심분야: Deep control policies for virtual characters / physical robots, Deep motion synthesis, Simulating human body / movement mechanisms
- https://orcid.org/0000-0002-0579-5987



권태수
- 1996-2000 서울대학교 전기컴퓨터공학부 학사
- 2000-2002 서울대학교 전기컴퓨터공학부 석사
- 2002-2007 한국과학기술원 전산학전공 박사
- 관심분야: Physics-based models, Machine learning techniques.
- https://orcid.org/0000-0002-9253-2156