한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 26, No. 2, P. 11~19

무게중심을 활용한 모션 생성 기술

박근태[1○] 손채준[2] 이윤상[3*]

한양대학교 컴퓨터 전공

{qkrrmsxo01[1], yoonsanglee[3]}@hanyang.ac.kr, thscowns@gmail.com[2]

Motion generation using Center of Mass

Geuntae Park○ Chae Jun Sohn Yoonsang Lee*

Dept. of Computer Science, Hanyang University

요 약

캐릭터의 자세가 변할 때 마다 캐릭터의 무게 중심(COM) 위치도 변하게 된다. 이 때 무게 중심의 위치 변화는 걷기, 뛰기, 쭈그려 앉기 등 다양한 동작 각각에 대응되는 독자적인 패턴을 가지므로 이를 이용하면 원래 동작의 정보를 알아낼 수 있다. 본 논문에서는 캐릭터의 무게 중심의 위치 변화를 토대로 동작을 예측하는 모션 생성 기법을 제안한다. 이 방법을 이용하면 무게 중심 정보를 통해 원래 동작의 유형에 대한 별도의 라벨 없이도 다양한 동작을 생성할 수 있다. 그러므로 네트워크의 학습 및 실행을 위한 데이터셋을 만들 때 사람의 손을 거칠 필요 없이 전처리를 비롯한 모든 과정을 자동으로 진행할 수 있다. 본 논문에서 제안하는 신경망 모델은 캐릭터의 모션 이력(history) 정보와 무게 중심 정보들을 입력 받아 현재 프레임에서의 포즈 정보를 출력하며, 연속적인 시계열 모션 데이터를 다루기 위해 1차원 Convolution을 수행하는 간단한 형태의 Convolutional Neural Network(CNN)를 사용하여 학습되었다.

Abstract

When a character's pose changes, its center of mass(COM) also changes. The change of COM has distinctive patterns corresponding to various motion types like walking, running or sitting. Thus the motion type can be predicted by using COM movement. We propose a motion generator that uses character's center of mass information. This generator can generate various motions without annotated action type labels. Thus dataset for training and running can be generated full-automatically. Our neural network model takes the motion history of the character and its center of mass information as inputs and generates a full-body pose for the current frame, and is trained using simple Convolutional Neural Network(CNN) that performs 1D convolution to deal with time-series motion data.

키워드: 무게 중심, Convolutional Neural Network(CNN), 캐릭터 애니메이션
Keywords: Center of mass, Convolutional Neural Network(CNN), Character animation

1. 서론

자신 또는 주변의 환경에 따라 자연스러운 캐릭터 애니메이션을 만들 수 있는 모션 생성기법은 컴퓨터 그래픽스 분야에서 중요한 과제이다. 모션 생성 모델은 환경에 대한 정보 및 사용자 입력을 받아 캐릭터의 포즈를 결정한다. 일반적으로 사용자의 입력에 따라 자연스럽게 반응하는 캐릭터 동작 생성을 목표로 하기에 이러한 모션 생성 기법은 게임과 같이 사용자가 상호작용을 통해 캐릭터를 제어하는 응용 프로그램에서 특히 중요하게 다뤄지는 부분이기도 하다. 이런 시스템에서 효과적으로 모션을 생성할 수 있는 매개변수를 선정하는 것은 매우 중요하다. 매개변수의 구조가 복잡하지 않을 수록, 차원이 작을수록 사용되는 자원이 줄어들며, 해당 매개변수로 다룰 수 있는 동작의 다양성이 높을 수록 매개변수를 통해 만들어 낼 수 있는 행동이 더 자연스럽고 다양해질 수 있기에 실용적이면서도 좋은 성능을 가진 생성 모델을 만들어 내기 위해서는 바람직한 매개변수를 찾는 것이 필수적이다.

*corresponding author: Yoon-Sang Lee/Hanyang University(yoonsanglee@hanyang.ac.kr)

Received : 2020.02.26./ Review completed : 1st 2020.03.19. / Accepted : 2020.03.20.
DOI : 10.15701/kcgs.2020.26.2.11
ISSN : 1975-7883(Print)/2383-529X(Online)

이에 본 논문에서는 무게 중심이라는 캐릭터 정보에 주목했다. 무게 중심 정보는 캐릭터의 동작을 결정하는 매개변수로 사용하는데 필요한 장점을 충분히 가지고 있다. 우선 무게중심은 매우 작은 차원만으로 표현이 가능하다. 좌표계에서 무게중심의 위치를 나타내기 위한 x, y, z축의 위치 값만 있으면 된다. 또한 무게중심의 위치 변화만 가지고도 많은 동작을 예측할 수 있다. 예를 들면, 걷는 동작의 경우 무게 중심은 y축을 지면에 수직인 방향이라 가정했을 때 x, z평면에서의 위치 값이 변화하며, y축 방향으로는 아주 작은 폭으로 진동하는 움직임을 보인다. 달리는 동작의 경우 걷는 동작과 유사하나, x, z 평면에서의 속력에 차이가 생긴다. 뛰어오르는 동작과 쭈그려 앉는 동작에서는 y값의 변화가 커진다. 본 논문에서는 무게 중심 정보에 추가로 좀 더 자연스러운 결과물을 만들기 위해 모션 이력 정보를 추가로 활용하여 무게중심을 통한 포즈 예측의 정확성을 높였다.

캐릭터 애니메이션 분야에 다양한 신경망을 활용한 연구가 활발히 진행되고 있다. 이러한 연구에서는 Convolutional Neural Network(CNN)[1], Recurrent Neural Network(RNN)[2] 같이 기본적인 형태의 신경망을 비롯하여 Phase-Functioned Neural Network(PFNN)[3]이나 Mode-Adaptive Neural Network(MANN)[4] 같이 성능 향상을 꾀하기 위해 기존 신경망을 개선시킨 새로운 네트워크를 활용했다. 이 논문에서 보일 모션 생성 모델은 그 중 CNN을 활용하여 모델을 학습했다. 여기서 CNN을 사용하여 연속적인 시계열 모션 데이터를 처리하기 위해 Holden et al. 2015[1]가 제안한 방법을 사용했다.

본 논문에서는 무게 중심 정보를 통해 현재 프레임의 전신 포즈를 예측하는 모델을 제안한다. 무게 중심 정보는 각 유형의 동작에 대응되는 독자적인 패턴을 나타내므로 이 특성을 활용하면 네트워크의 입력으로 들어가는 데이터셋에 각 동작 유형의 라벨을 따로 명시하지 않아도 다양한 유형의 동작들을 모델 스스로가 판단하고 처리할 수 있다. 이를 통해 사용자는 전처리 과정에서 각 동작 유형에 대해 수동으로 라벨링 해 줘야 하는 부담을 덜어 편의성과 효율성을 얻을 수 있으며 다양한 동작에 대한 잘못된 라벨링이 발생할 위험을 피할 수 있다. 본 논문에서 제안하는 기법은 사용자가 무게중심을 직접 조작하면 모델에서 이를 입력받아 적합한 모션을 예측하는 형태로 활용될 수 있을 것이다.

2. 관련 연구

컴퓨터 그래픽스에서 모션을 만들 때 사용되는 대표적인 방법으로 데이터-기반 접근법(data-driven approach)과 물리-기반 접근법(physics-based approach)이 있다. 물리-기반 접근법은 물리 법칙에 따라 수식을 계산하여 캐릭터의 모션을 생성한다. 반면 데이터-기반 접근법은 모션 캡쳐 장비를 이용해 얻어낸 실제 사람의 여러 모션 데이터를 통해 계산 모델을 구축하여 대상의 행동 구조를 모델링하는 기술이다. 본 논문은 데이터-기반 접근법에 기반하여 연구를 수행했다. 데이터-기반 접근법의 주요 목표는 모션 캡쳐 데이터셋과 같은 고품질의 데이터 집합을 활용하여 임의의 환경에서 캐릭터의 자연스러운 동작을 생성하는 것이다. 이에 Lee et al. 2002[5], Kovar et al. 2002[6], Arikan and Forsyth 2002[7]에서는 모션 그래프(motion graph)라는 개념을 제안하고, 여러 모션 캡쳐 데이터를 연결하여 연속적인 캐릭터의 움직임을 합성하는 방법을 연구했다. 나아가 Lau and Kuffner 2005[8]는 보다 간단하고 모션간의 연결 상태가 명확한 구조의 유한 상태 머신(finite-state machine)을 통해 상호작용을 위한 캐릭터의 반응성을 향상시켰으나, 유한 상태 머신의 표현력에는 한계가 있어 다양한 행동 중 일부만 묘사 가능하다는 단점이 있다. Hyun et al. 2016[9]에서는 복잡한 행동 구조를 모델링하기 위한 방법으로 motion grammar를 제안했다.

모션 캡쳐 데이터는 각 프레임에서의 동작 정보가 시간 순서에 따라 나열 될 때 의미를 가지는 일종의 시계열 데이터로 볼 수 있으며, 이러한 관점에서 계산 모델을 구현하려는 다양한 연구들 또한 있었다. 연속적으로 나열된 데이터의 의미를 추출하고 이를 통해 결과를 만들어 내는 계산 모델을 제안하기 위해 선형 회귀(linear regression)[10], K-Nearest Neighbor(KNN)[11], 커널 기반 접근법[12]과 같은 학습 방법을 사용하여 시계열 데이터의 의미를 학습하는 연구가 여럿 진행된 바 있다.

근래 들어 신경망을 활용한 연구가 활발히 진행되고 있다. 신경망은 학습을 수행할 때 데이터의 양에 상관 없이 연구자가 정한, 작은 크기의 메모리 공간만을 사용하며 비 선형적인 모델링이 가능하다. 또한 학습이 완료된 모델을 실행할 때에는 빠른 시간 안에 적은 양의 메모리만을 사용하여 좋은 품질의 결과물을 출력하므로 실시간으로 데이터를 처리해야 하는 시스템에 효과적이다. 모션 생성을 위해 다양한 종류의 신경망들이 활용되었는데, Taylor et al. 2009[13], Taylor et al. 2011[14]은 conditional Restricted Boltzmann Machine(cRBM)으로 모션 데이터를 학습시켰으며, Holden et al. 2015[1]는 Convolutional Neural Network(CNN)을 활용하여 시계열 데이터를 처리하고 학습하는 방법을 제안했다. Mordatch et al. 2015[2], Lee et al. 2018[15]은 캐릭터 애니메이션에 Recurrent Neural Network(RNN)를 활용했다. 본 논문에서는 Holden et al. 2015[1]에서 제안한 방법과 비슷하게 CNN모델을 구성했고 이를 사용하여 모션 데이터를 학습시켰다. 이에 대한 자세한 내용은 Section 3.3에서 설명할 것이다.

3. 무게중심을 활용한 모션 생성 모델

3.1 Overview

본 논문에 사용된 모델을 학습시키는데 필요한 데이터셋을 준비하기 위해 가공되지 않은 모션 캡쳐 데이터를 전처리하고 입력 및 출력 벡터를 만들었다. 네트워크는 이 학습 데이터셋을 이용하여 end-to-end 방식으로 학습되었다.

학습 된 모델은 이전 프레임까지의 모션 이력과 현재 프레임의 무게 중심 위치로 이루어진 입력 값 $\mathbf{x}$를 받아 현재 프레임의 캐릭터의 포즈 $\hat{\mathbf{y}}$을 예측하는 시계열 모델이다. 모션 이력은 일정 시간

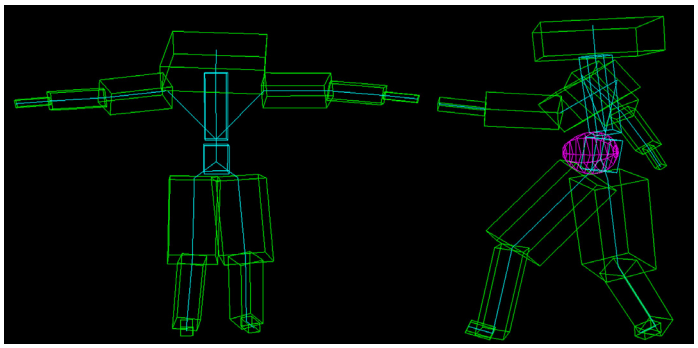


Figure 1: Left : The skeleton structure of body model used in our experiments. It has 15 body parts. Right : Visualization of COM. The purple ball represents character COM position.

동안의 연속된 포즈 데이터들로 구성되며 시간이 흐름에 따라 자동으로 갱신된다. 생성 모델은 CNN과 fully-connected network로 구성된 간단한 형태의 신경망으로 구성 되었다.

3.2 데이터 준비

본 논문의 학습 데이터는 가공되지 않은 모션 캡쳐 데이터를 사용하여 만들어진다. 다양한 스타일의 걷기, 제자리 운동 및 쭈그려 앉아 걷기로 구성된 모션 데이터가 사용되었다. 모션 캡쳐 데이터셋은 네크워크의 학습 데이터셋에서 ground truth 출력값에 해당되는 전신 포즈 정보만을 가지고 있으므로, 입력값은 따로 계산을 통해 만들어야 한다. 학습 데이터셋을 만들기 위한 전처리 과정에 동작 유형에 대한 별도의 라벨링은 필요 없으므로 학습 데이터셋은 전자동으로 제작된다.

실험에 사용된 캐릭터의 골격은 모션 캡쳐 데이터에 맞춰 15개의 관절부로 구성했다(Fig. 1 왼쪽 참조). 따라서 캐릭터 신체는 각 관절들의 자유도(15×3)에 root의 위치 정보(x,y,z 좌표)가 더해져 총 48만큼의 자유도를 가진다. 임의의 프레임 n에서의 캐릭터 포즈 정보는 본 논문에서 m_n으로 표현된다.

원본 모션 캡쳐 데이터들은 다양한 환경에서 수집된 데이터들이므로 서로 fps 및 초기 위치와 초기 방향이 다르다. 본 연구에서는 먼저 각 모션 캡쳐 데이터들을 30fps로 샘플링하는 과정을 수행하였다. 이 후 전역 좌표계 기준 $(0, 0, 0)$ 지점으로 초기 위치를 통일 시켰으며, z축 방향 $(0, 0, 1)$을 바라보도록 방향성을 통일 시켰다. 본 논문에서 사용된 모델은 캐릭터 신체를 기준으로 하는 좌표계의 값을 사용하기 때문에 좌표와 방향성을 통일하는 작업이 필요한 것은 아니지만, 실험 시 다양한 모션 데이터에 대한 ground truth와 모델의 출력을 간단히 육안으로 확인할 수 있는 환경을 만들기 위해 수행되었다.

다음으로, 전처리 완료된 데이터에서 각 프레임별 캐릭터 전신의 무게 중심 좌표를 구한다. 원본 모션 캡쳐 데이터에는 캐릭터의 신체 부위별 질량에 대한 정보는 없으므로 본 연구에서는 각 신체 부위에 임의의 질량 값을 설정한 후 캐릭터 전신의 무게 중심 좌표를 구했다(Fig. 1 오른쪽 참조). 본 논문에서 설정한 각 신체 부위별 질량은 Table 1에서 확인할 수 있다. 이렇게 만들어진 데이터는 네트워크를 학습할 때 모션 이력 데이터와 합쳐져 입력 값으로 사용된다.

Table 1: Name and mass value setting per body part

부위(명칭)	설정 값(질량)
Hips	10
Spine	20
Head	5
LeftArm	2
LeftForeArm	1.5
LeftHand	0.5
RightArm	2
RightForeArm	1.5
RightHand	0.5
LeftUpLeg	5
LeftLeg	4
LeftFoot	0.5
RightUpLeg	5
RightLeg	4
RightFoot	0.5
total	62

입력 값은 무게 중심 정보 외에도 현재 프레임 t를 기준으로 바로 이전 프레임부터 과거 i번째 프레임까지의 모션 데이터 각각을 t의 신체의 root 지역 좌표계(local coordinate)에 맞춰 정렬한 모션 이력 $M_t = \{m_{t-i}...m_{t-1}\}$도 포함한다. 이는 캐릭터가 과거 임의의 시간 동안 수행한 모션 정보를 네트워크에 추가로 제공하여 무게 중심 정보만을 사용했을 때의 모호한 결과를 보완하고 동작 예측이 좀 더 정확해 지도록 보조한다. 이 모션 이력 정보는 전처리된 모션 캡쳐 데이터에서 간단히 추출할 수 있다.

지도 학습을 위한 출력값에는 ground truth 값을 사용한다. end-to-end 네트워크를 통해 만들고자 하는 값은 현재 프레임 t의 모션 정보이므로 가공된 모션 캡쳐 데이터에서 현재 프레임 t의 모션 정보를 추출하여 학습을 위한 출력값으로 사용한다.

최종적으로, 프레임 t에서의 입력 벡터 $\mathbf{x}$는 $\{M_t, C_t\}$로 표현되며, 여기서 C_t는 t에서의 무게 중심 위치 정보이며 M_t는 과거 $t-i$ 시점까지의 모션 이력 정보이다. 학습을 위한 출력 벡터 $\mathbf{y}$는 m_t의 형태로, 가공된 모션 캡쳐 데이터에서 추출해 낸 현재 프레임 t의 ground truth 모션 정보를 의미한다.

3.3 Network

실험에 사용된 생성 모델은 시계열 데이터를 처리하기 위한 3개 계층(layer)으로 구성된 CNN과 현재 모션을 예측하기 위한 2개 계층으로 구성된 fully-connected network가 결합된 간단한 구조를 가진다(Fig. 2 참조). 전체 네트워크는 현재 프레임 t에서 무게 중심이 위치하는 3차원 위치 값 C_t와 과거 i 프레임 동안의 모션 정보 값 M_t을 입력 받아 현재 모션 정보를 예측한 값 $\hat{m}_t$를 출력한다.

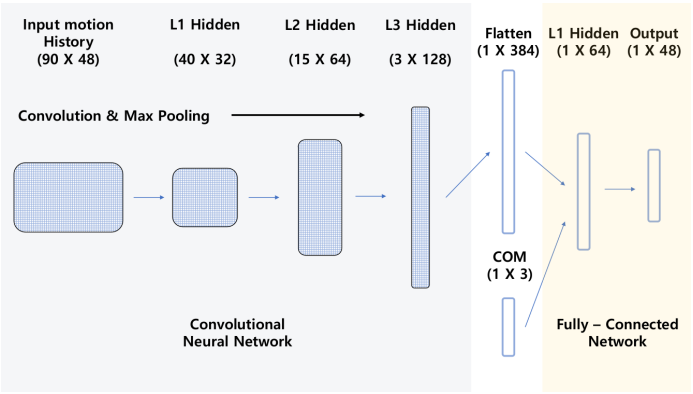


Figure 2: The architecture of our motion generator composed of 3 layered Convolutional Neural Network(CNN) and 2 layered fully-connected network. In CNN, the network deals with time-series data. In fully-connected network, the network predicts current motion information.

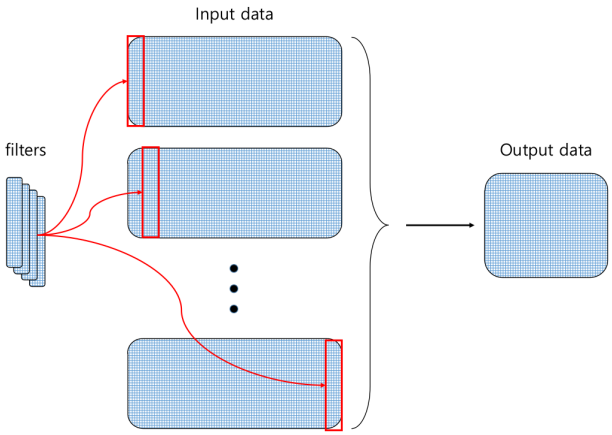


Figure 3: Simple description of 1D Convolution process.

CNN은 RNN같은 복잡한 네트워크에 비해 구현이 간단하고 학습 속도가 빠르다. 또한 CNN은 이미지 분류[16], 음성 인식[17] 등의 분야 뿐만 아니라 모션 데이터의 매니폴드(manifold)를 학습하는 데 적용해도 좋은 성과를 나타낸다는 것이 알려졌다[1]. 이에 본 논문에서는 Holden et al. 2015[1]가 제안한 방법으로 CNN을 사용하여 시계열 데이터를 처리하는 네트워크를 만들었다.

모션 예측 네트워크(Motion prediction network)는 모션 이력의 특징을 추출하는 CNN $N(\cdot)$과 현재 프레임의 모션을 예측하는 fully-connected network $P(\cdot)$로 구성된다. $N(\cdot)$은 다음과 같은 형태를 가진다.

$$N(M) = \phi(\text{ReLU}(\phi(\text{ReLU}(M * \mathbf{F}_0 + \mathbf{b}_0)) * \mathbf{F}_1 + \mathbf{b}_1)) * \mathbf{F}_2 + \mathbf{b}_2$$

여기서 *은 Convolution 연산을 나타내며 $\mathbf{F}_n$은 n번째 계층의 필터를 의미한다. $\mathbf{b}_n$은 n번째 계층의 편향값(bias)을 의미한다. 또한 ϕ는 max pooling을 표현하는 기호이다. 각 CNN 계층에서 필터 $\mathbf{F} \in \mathbb{R}^{wh}$를 사용하여 1차원 Convolution을 수행한다(Fig. 3 참조). h는 1차원 Convolution을 위해 계층이 입력 받은 행렬의 행

Table 2: Filter and output of CNN layers

layer	filter 크기	filter 수	출력 값 크기
1	10×48	32	40×32
2	10×32	64	15×64
3	10×64	128	3×128

개수로 고정되며 w는 사용자가 임의로 정할 수 있는 값으로, 이 값을 조정하여 해당 계층에서 convolution 한 번당 탐색할 모션의 시간을 설정할 수 있다.

실험에 사용된 네트워크 각 계층에서의 필터 $\mathbf{F}$의 크기와 개수는 Table 2에서 확인할 수 있다. 각 계층의 출력 값이자 다음 계층의 입력 값인 중간 결과물 $\mathbf{S}$는 $\mathbf{S} \in \mathbb{R}^{ab}$를 따른다. b는 이전 계층의 필터 수로 정해진다. a는 convolution 연산을 통해 정해지며 계층을 거쳐갈수록 크기가 점차 줄어들게 된다(Fig. 2 참조). 각 계층의 중간 결과물 크기는 Table 2에 기술해 두었다. 최종적으로 임의의 시간대에서 연속된 모션은 CNN 계층들을 거쳐 1차원 벡터 $\mathbf{Z} \in \mathbb{R}^{384}$이 된다. 활성 함수로 사용한 Rectified Linear Units(ReLU)[18]는 다음과 같이 정의된다.

$$\text{ReLU} = max(x, 0)$$

$P(\cdot)$의 형태는 다음과 같다.

$$P(\mathbf{K}) = \mathbf{W}_1(\mathbf{W}_0\mathbf{K} + \mathbf{b}_0) + \mathbf{b}_1$$

여기서 $\mathbf{W}_n$은 n번째 계층의 가중치(weight)를 의미하며 $\mathbf{b}_n$은 n번째 계층의 편향값을 의미한다. 앞서 $N(\cdot)$으로 구한 벡터 $\mathbf{Z}$에 현재 프레임 t의 무게중심 3차원 위치 벡터 $C_t \in \mathbb{R}^3$을 이어 붙여 벡터 $\mathbf{K} \in \mathbb{R}^{387}$를 만든다. 이렇게 만들어진 $\mathbf{K}$는 2개의 FC 계층으로 구성 된 네트워크 $P(\cdot)$를 거쳐 t의 모션을 예측한 값 $\hat{m}_t$가 된다(Fig. 2 참조).

3.4 Training

네트워크는 가공된 모션 캡쳐 데이터로 만든 학습 데이터셋3.2을 사용하여 end-to-end 방식으로 학습된다. 즉 현재 프레임의 무게 중심 정보와 과거 일정 시점까지의 모션 이력이 담긴 데이터를 받아 현재 프레임의 모션을 예측하는 것이 네트워크의 학습 목표인 전형적인 추론 작업이다. 본 논문에서는 ground truth와 예측한 결과물 간의 평균 제곱 오차(mean squared error)를 비용 함수로 사용했다. 이는 다음과 같은 간단한 수식으로 표현할 수 있다.

$$\text{Cost}(M, \mathbf{C}, \mathbf{y}) = \|\mathbf{y} - P(\{N(M), \mathbf{C}\})\|_2^2$$

여기서 $\mathbf{y}$는 ground truth에 해당하는 현재 모션 값, M은 모션 이력 데이터이며 $\mathbf{C}$는 현재 프레임의 무게중심 위치이다. M과 $\mathbf{C}$는 입력 데이터 $\mathbf{x}$의 형태로 모델에 입력된다.

네트워크를 학습하기 위해 다양한 스타일의 걷는 동작, 쭈그려

Table 3: The Breakdown of Dataset

Motion type	data 수	평균 시간(초)	총 프레임
걷기	56	5.4	9097
쭈그려 걷기	6	10.2	1831
기타	10	11.2	3354

Table 4: Settings for training

History size	15	15	90	90
Kernel size	2	2	10	10
# of layer	1	3	1	3
# of out channel	32	8/16/32	128	32/64/128

걷는 동작 및 제자리 운동을 비롯한 기타 동작을 표현하는 모션 캡쳐 데이터를 사용했다. 학습이 더 잘 되도록 모션 데이터의 각 프레임마다 앞서 설정한 모션 이력 길이만큼의 하위 모션 클립을 만들어 간단한 증강(augmentation) 효과를 주었다. 이렇게 만들어진 모션 클립들을 임의의 배치(batch) 크기에 맞춰 무작위로 선택한 뒤 학습에 사용하였다(실험에 사용된 모션 클립당 설정한 프레임 수(모션 이력 길이)는 Section 4에서 자세히 설명할 것이다).

4. 실험 결과

본 연구는 AMD 라이젠 7 2700 CPU와 GeForce RTX 2070, 16GB RAM 등으로 구성된 장비에서 수행되었으며 Python ver. 3.6 환경에서 Pytorch ver. 1.3.1을 사용하여 모델을 구현하고 학습시켰다.

걷는 동작 56개, 쭈그려 걷는 동작 6개, 기타 동작 10개로 구성된 총 72개의 데이터가 실험에 사용되었다 (사용된 데이터의 자세한 정보는 Table 3참조). 학습시에 여러 환경 값을 설정하여 실험했으며 그 중 이력 크기 90, 커널 크기 10, 계층 수 3, 채널 수(또는 필터 수)를 첫 번째 계층은 32, 두 번째는 64, 세 번째는 128로 설정했을 때 가장 좋은 결과가 나타나는 것을 확인했다 (각 설정 값은 Table 4참조). 최적화를 위해 Stochastic Gradient Descent(SGD) 알고리즘을 사용했다. 이와 같은 환경에서 학습을 완료하기까지 약 10–12시간이 소요되었다.

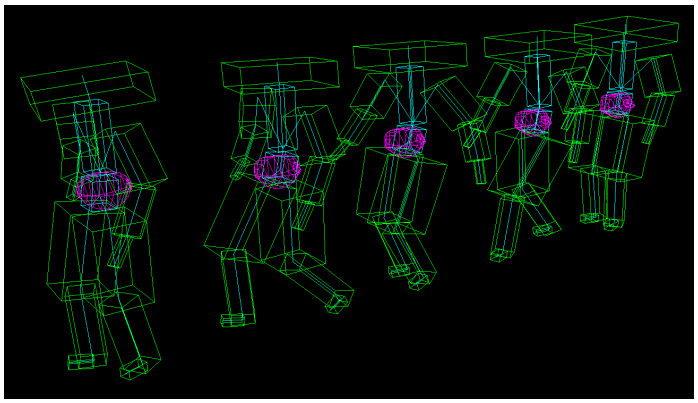
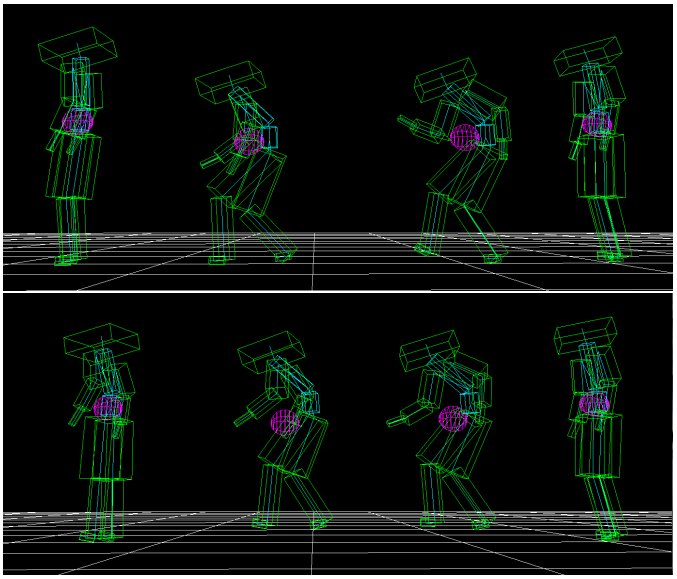
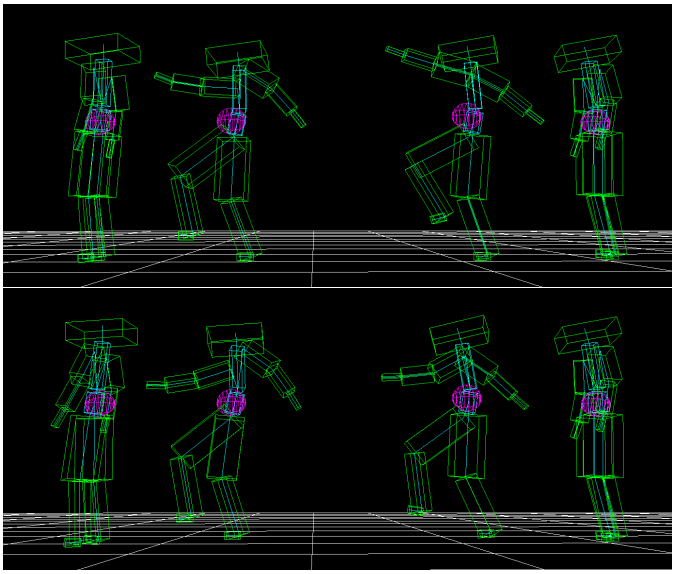


Figure 4: Simple walking motion that our model makes.



(a) Crouch walk motion. Ground truth(top) and generated model(bottom).



(b) Walking like soldier. Ground truth(top) and generated model(bottom).

Figure 5: Different type(a) and style(b) of motion than the simple walking motion(Fig. 4). In (a), as the COM shifted downward, crouch walk motion is generated. (b) is a kind of walking type motion, but unlike Fig. 4, it has large stride and arm motion. Model generates distinctive walking style well.

학습된 네트워크의 성능은 학습 데이터셋에서 학습에 사용하지 않고 제외해 둔 일부 데이터셋을 테스트 데이터셋으로 사용하여 평가했다. 네트워크가 작동 하려면 과거 일정 시간 동안의 모션 이력과 현재 시점에서의 무게중심 위치 정보가 있어야 하므로 0번째 프레임부터 시작하여 정해진 임의의 시간 동안의 이력이 담긴 모션 클립을 주고 그 이후부터의 모션을 예측하도록 했다. 모션 예측 실험은 두 가지 설정 하에 진행되었으며 네트워크 결과물의 품질은 결과물을 출력시킨 후 육안으로 평가하였다. 첫 번째 설정으로 네트워크에 입력하는 모션 이력을 갱신할 때 네트

Table 5: Prediction Error

Motion type	걷기	쭈그려 걷기	기타
result	0.415	0.614	1.567

워크의 출력값에 상관 없이 ground truth에서 현재 프레임의 모션 데이터를 연속적으로 나열한 이력 데이터에 추가했다. 실험 결과 예측(prediction) 시간에 상관 없이 비교적 안정적인 결과물을 출력하는 것을 확인했다(Fig. 5 참조).

Table 5는 원본 모션의 무게 중심 경로를 입력으로 넣었을 때 출력된 모션과 원본 모션과의 차이를 계산한 표이다. 학습 및 결과 출력에 각 관절의 각도를 이용했으므로 원본 모션과의 오차는 관절 각도의 차이를 기준으로 계산되었다. 비교 결과 가장 데이터가 많은 걷는 동작이 오차가 제일 낮았으며 비교적 데이터가 적은 쭈그려 걷는 동작도 잘 만들어 내는 것을 확인했다. 기타 동작의 경우 제자리 운동이나 뒤로 걷는 동작 등 소수의 여러 동작들로 구성되어 있기에 가장 낮은 정확도를 보였다.

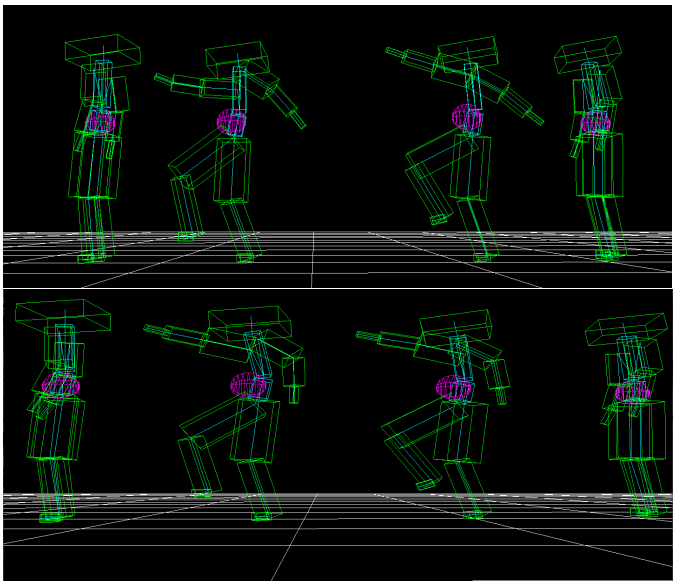
두 번째 설정으로는 연속적으로 나열된 이력 데이터에 네트워크의 예측 값을 추가하는 방법이었는데 모션 예측 시간이 길어질수록 모션 이력에 네트워크의 오차가 쌓여나가 결과물의 품질이 빠르게 악화되는 문제가 있었다.

5. 결론

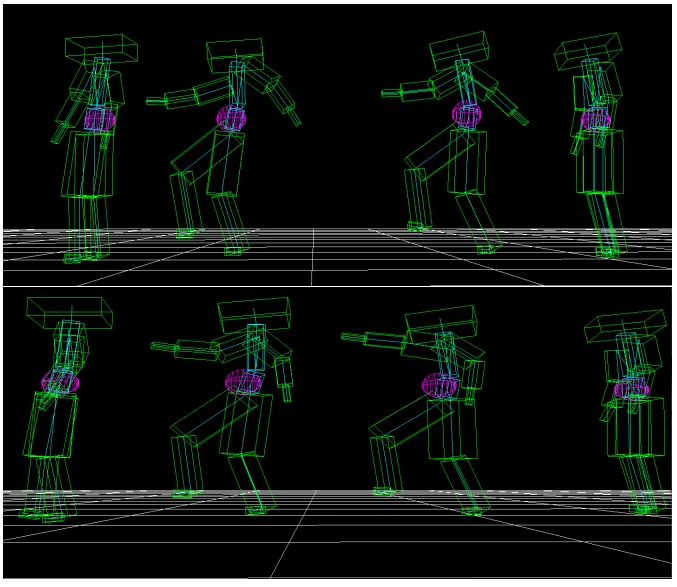
모션 생성 모델에 무게중심 정보를 매개변수로 활용하는 것은 여러 이점을 가지고 있다. 무게중심 정보는 매개변수화 했을 때 크기가 작다. 즉, 동작을 예측하기 위해 탐색해야 할 공간의 차원이 작기 때문에 학습 데이터셋의 양이 적어도 비교적 학습이 잘 되는 장점이 있다. 또한 무게 중심만을 통해서 여러 동작들을 구분할 수 있으며, 이를 원본 모션 캡쳐 데이터에서 간단한 계산을 통해서 쉽게 얻어낼 수 있다. 이 덕분에 네트워크를 학습시키기 위한 데이터셋을 준비할 때, 동작 유형 라벨을 기입해 주는 등 수동 작업을 해줄 필요가 없으며, 필요한 모든 데이터는 원본 데이터셋에서 자동으로 추출해 낼 수 있다. 무게 중심 정보가 root 좌표 정보에 비하여 움직임에 좀 더 민감한 정보라는 점 또한 모션 생성에 긍정적인 효과를 가지고 있다. 예를 들어 제자리에 멈춰서 팔만을 움직이는 동작의 경우 root의 좌표값에는 큰 변화가 없지만 무게중심의 좌표 값에는 큰 변화가 생긴다. 이 특성은 무게중심 정보가 보다 다양한 동작을 다룰 수 있도록 한다.

학습에 쓰인 모션 데이터셋이 걷기, 쭈그려 걷기 등 몇 가지 동작에 한정되어 있어 다양한 모션에 대한 결과를 확인하지 못했다는 한계점이 있다. 본 논문의 실험 결과로도 무게중심의 위치 정보가 다양한 유형의 모션을 다룰 수 있는 가능성을 보여줄 수 있으나 공중제비 같은 특수한 동작에 대한 학습 결과를 실질적으로 확인할 수 없었던 것은 아쉬운 점이다. 실험 결과에서 학습된 모델이 만들어 낸 결과물에서 발이 미끄러지듯 움직이는 문제가 발생하는 것이 관찰되었다. 이런 현상은 여러 캐릭터 애니메이션 연구들에서 꾸준히 언급된 현상이며, 학습 데이터가 부족한 경우 발생하기 쉽다는 것이 보고된 바 있다[4].

걷기와 쭈그려 걷기 같이 큰 범주로 구분할 수 있는 유형의 동작들 간에는 서로 구분할 수 있게 만들어지나 동일한 유형 동작간의 세밀한 차이는 제대로 만들지 못하는 한계도 있다. 예를 들어, 군인처럼 팔을 쭉 뻗고 앞으로 크게 흔들며 걷는 동작과 운동을 목적으로 힘차게 걷는 동작은 둘 다 걷는 유형의 동작이지만 팔의 움직임이나 발을 뻗는 방식에서 약간의 차이가 존재한다 (Fig. 6 참조). 이 현상은 모델 자체의 한계점으로, 서로 확연히 차이나



(a) Motions of ground truth. Soldier Walking (top). Exercise Walking (bottom).



(b) Generated motions using our model. Soldier Walking (top). Exercise Walking (bottom).

Figure 6: Two motions have slight difference in swinging arms and foot steps. The two generated results (b) shows similar motions with this difference excluded. In (b), the third motion image of each result shows this problem well.

는 동작들 간에는 무게중심의 위치 변화 패턴이 확연하게 다른

양상을 보이나 유사한 동작들 간에는 무게중심 변화 패턴이 크게 다르지 않아 학습 시 세세한 동작들이 섞여 평균화되므로 생기는 현상으로 보인다. 이 현상으로 인해 생성 모델이 만든 동작에서 팔을 올리다 마는 등 부자연스러운 모습을 보이는 경우도 있다.

본 논문의 시스템에 사용자의 조작을 입력 받아 무게중심을 원하는 위치로 변환시킬 수 있는 인터페이스를 추가하면 상호작용이 가능한 캐릭터 애니메이션 모델을 구현할 수 있다. 이 시스템에서 사용자는 마우스로 무게중심을 드래그 앤 드롭하는 것과 같은 간단한 조작만으로 달리기, 쭈그려 앉아 걷기 같은 다양한 동작을 구현해 낼 수 있다. 이는 사용자와의 상호작용이 중요한 비디오/컴퓨터 게임 분야에 효과적으로 적용하거나 관련 업계 종사자가 캐릭터 모션을 간단히 만들 수 있게 보조하는데 유용하게 사용될 수 있을 것이다.

본 논문에서 연구한 것은 무게 중심 정보의 활용 가능성을 확인하기 위한 기초적인 단계이므로 아직 해결해야 할 문제점과 앞으로 시도해 볼만한 흥미로운 연구거리들이 많이 있다. 앞으로의 주된 연구는 이 논문에서 보인 모델의 한계점을 극복하고 생성된 결과물의 품질을 향상시키는 방향이 될 것이다. 대표적으로 발이 미끄러지는 문제는 캐릭터의 모션을 부자연스럽게 만드는 치명적인 문제이며 자연스러운 동작을 위해서는 필수적으로 해결해야 할 문제이다. 여러 캐릭터 애니메이션 연구에서는 효과적으로 이를 해결할 다양한 방법들이 제안되었다. 한 가지 간단하고 적용하기 쉬운 해결책으로, Lee et al. 2018[15]에서는 이러한 문제를 발의 지면 접촉 정보를 매개변수화 한 뒤 입력 데이터에 추가하여 네트워크를 학습시키는 방법을 제안한 바 있다. 이 방법을 비롯하여 여러 해결 방법들을 본 연구에서 사용 된 모델에 접목시켜 개선된 결과를 확인할 수 있을 것이다. 사용자와 상호작용이 가능한 모델을 만들기 위해 현재 모델의 학습 및 실행 방식을 개선하는 것이 필요하다. 본 논문의 모델은 모션 이력에 ground truth가 들어가는 방식에서는 성공적으로 작동하는 것을 확인했다. 하지만 다르게 말해서 이는 ground truth가 계속해서 공급되어야 지속적으로 품질 높은 모션을 만들어 낼 수 있는 방식으로, 상호작용 가능한 응용 프로그램에 적합한 형태는 아니다. 상호작용 가능한 시스템을 위해 네트워크에서 출력 된 모션 데이터로 이력을 갱신하는 환경에서 안정적으로 작동하는 모델을 연구를 해 보는 것도 좋을 것이다. 또한 무게중심 값은 핵심 입력값이며 사용자의 직접적인 조작을 받는 값이므로 안정적인 성능을 보장할 안전장치가 필요하다. 하지만 현재 연구에 사용된 모델은 잘못된 입력값에 대한 안전장치가 없기에 입력으로 들어오는 무게중심 값에 갑작스럽고 과도한 변화가 생기는 경우 해당 프레임 뿐만 아니라 그 이후의 모든 프레임에서 의미 없는 결과를 만들게 될 것이다. 이 문제는 입력으로 들어오는 무게중심 값에 제한을 두어 이전 프레임 대비 일정 거리 이상 떨어진 좌표값이 들어오는 경우 방향성만 유지한 채 최대 제한 거리 내의 좌표값으로 수정해 주는 안전장치를 도입하는 것으로 해결할 수 있을 것이다. 세세한 동작을 표현하기 위해 추가적인 매개변수를 도입하는 연구도 해 볼 수 있을 것이다. 이를테면 몸 전체 무게중심 뿐만 아니라 손과 팔, 발과 다리, 몸통, 머리 같이 전신을 하위 부위로 나누어 각 부위별 무게중심 정보를 매개변수화하여 이용하는 방법을 생각해 볼 수 있을 것이다. 여러 유형의 동작들 간의 전환이 있는 학습 데이터셋을 만들 수 있는 데이터 증강 기법을 활용하고 이를 통해 네트워크를 학습시켜 볼 수도 있다. 그 결과 학습이 완료된 모델은 캐릭터가 다양한 모션간의 전환을 자연스럽게 할 수 있게 만들어 줄 것이다.

감사의 글

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 SW 중심대학지원사업의 연구결과로 수행되었으며 (2016-0-00023), 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2019R1C1C1006778, NRF-2019R1A4A1029800).

References

[1] D. Holden, J. Saito, T. Komura, and T. Joyce, “Learning motion manifolds with convolutional autoencoders,” in *SIGGRAPH Asia 2015 Technical Briefs*, 2015, pp. 1–4.

[2] I. Mordatch, K. Lowrey, G. Andrew, Z. Popovic, and E. V. Todorov, “Interactive control of diverse complex characters with neural networks,” in *Advances in Neural Information Processing Systems*, 2015, pp. 3132–3140.

[3] D. Holden, T. Komura, and J. Saito, “Phase-functioned neural networks for character control,” *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 1–13, 2017.

[4] H. Zhang, S. Starke, T. Komura, and J. Saito, “Mode-adaptive neural networks for quadruped motion control,” *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–11, 2018.

[5] J. Lee, J. Chai, P. S. Reitsma, J. K. Hodgins, and N. S. Pollard, “Interactive control of avatars animated with human motion data,” in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, 2002, pp. 491–500.

[6] L. Kovar, M. Gleicher, and F. Pighin, “Motion graphs,” in *Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH ’02. New York, NY, USA: Association for Computing Machinery, 2002, p. 473–482. [Online]. Available: https://doi.org/10.1145/566570.566605

[7] O. Arikan and D. A. Forsyth, "Interactive motion generation from examples," *ACM Transactions on Graphics (TOG)*, vol. 21, no. 3, pp. 483–490, 2002.

[8] M. Lau and J. J. Kuffner, "Behavior planning for character animation," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2005, pp. 271–280.

[9] K. Hyun, K. Lee, and J. Lee, "Motion grammars for character animation," in *Computer Graphics Forum*, vol. 35, no. 2. Wiley Online Library, 2016, pp. 103–113.

[10] E. Hsu, K. Pulli, and J. Popović, "Style translation for human motion," in *ACM SIGGRAPH 2005 Papers*, 2005, pp. 1082–1089.

[11] Y. Lee, K. Wampler, G. Bernstein, J. Popović, and Z. Popović, "Motion fields for interactive character locomotion," in *ACM SIGGRAPH Asia 2010 papers*, 2010, pp. 1–8.

[12] J. M. Wang, D. J. Fleet, and A. Hertzmann, "Gaussian process dynamical models for human motion," *IEEE transactions on pattern analysis and machine intelligence*, vol. 30, no. 2, pp. 283–298, 2007.

[13] G. W. Taylor and G. E. Hinton, "Factored conditional restricted boltzmann machines for modeling motion style," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 1025–1032. [Online]. Available: https://doi.org/10.1145/1553374.1553505

[14] G. W. Taylor, G. E. Hinton, and S. T. Roweis, "Two distributed-state models for generating high-dimensional time series," *Journal of Machine Learning Research*, vol. 12, no. Mar, pp. 1025–1068, 2011.

[15] K. Lee, S. Lee, and J. Lee, "Interactive character animation by learning multi-objective control," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 6, pp. 1–10, 2018.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[17] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[18] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[19] J. Chai and J. K. Hodgins, "Performance animation from low-dimensional control signals," in *ACM SIGGRAPH 2005 Papers*, 2005, pp. 686–696.

[20] J. Min and J. Chai, "Motion graphs++ a compact generative model for semantic motion analysis and synthesis," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 6, pp. 1–12, 2012.

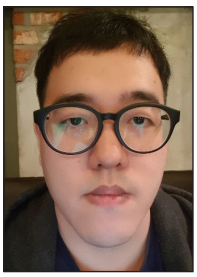
[21] A. Safonova and J. K. Hodgins, "Construction and optimal search of interpolated motion graphs," in *ACM SIGGRAPH 2007 papers*, 2007, pp. 106–es.

[22] K. Grochow, S. L. Martin, A. Hertzmann, and Z. Popović, "Style-based inverse kinematics," in *ACM SIGGRAPH 2004 Papers*, 2004, pp. 522–531.

〈저자소개〉

박근태

- 2013-2020 한양대학교 컴퓨터공학부 학사
- 2020-현재 한양대학교 컴퓨터소프트웨어학부 석사
- 관심분야: Data-driven Character Control
- https://orcid.org/0000-0002-6945-5930



손채준

- 2014-2020 한양대학교 컴퓨터공학부 학사
- 2020-현재 한양대학교 컴퓨터소프트웨어학부 석사
- 관심분야: Physically-Based Character Control, Deep Reinforcement Learning
- https://orcid.org/0000-0002-4118-2694

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 조교수
- 관심분야: Physically-Based Character Control, Robot Control Algorithm, Computational Desinge
- https://orcid.org/0000-0002-0579-5987