



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원 저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리와 책임은 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)



공학석사학위논문

물리 기반 캐릭터 시뮬레이션을 이용한
낙상 감지 용도의 가상 관성 데이터 생성

Generation of Virtual Inertial Data for Fall Detection
using Physics-based Character Simulation

2023년 2월

서울대학교 대학원

컴퓨터 공학부

정민효

물리 기반 캐릭터 시뮬레이션을 이용한
낙상 감지 용도의 가상 관성 데이터 생성

Generation of Virtual Inertial Data for Fall Detection
using Physics-based Character Simulation

지도교수 이영기
이 논문을 공학석사 학위논문으로 제출함
2022년 11월

서울대학교 대학원
컴퓨터 공학부
정민효

정민효의 석사 학위논문을 인준함
2022년 11월

위 원 장 _____ (인)
부위원장 _____ (인)
위 원 _____ (인)

요약

낙상은 어떠한 환경에서 연령을 가리지 않고 발생할 수 있으며 특히 노년층의 경우 사망에까지 이를 수 있다. 낙상 감지 기술에 대해서는 영상 또는 관성 센서 등과 같은 다양한 정보를 이용하여 감지하는 연구가 되었고 최근에는 인공지능을 이용하여 이 정보들을 분석하는 연구가 활발히 진행되고 있다. 모바일 기기의 발전으로 관성 센서를 이용한 낙상 감지 기술이 많이 연구되었으나, 데이터의 중요성에 비해 낙상 환경의 데이터를 충분히 얻기 어려운 문제점이 있다. 본 논문에서는 가상 물리 시뮬레이션 환경에서 인간 형태의 캐릭터를 사용하여 자연스러운 낙상 동작을 생성하고 3축 가속도, 속도와 같은 가상 관성 센서 정보 또한 생성하고자 한다. 강화학습을 이용하여 학습시킨 시뮬레이션 환경에서는 캐릭터의 신체 정보 및 마찰력과 같은 물리적인 정보를 이용하여 낙상 감지에 필요한 데이터를 생성할 수 있다. 본 연구를 통하여 최신의 캐릭터 애니메이션 기술을 낙상 감지 기술에 접목 함으로써 기존에 얻기 어려웠던 낙상의 관성 센서 데이터를 쉽게 생성하여 낙상 감지 기술에 활용할 수 있다.

주요어: 컴퓨터 애니메이션, 물리 시뮬레이션, 물리기반 제어, 데이터 기반 애니메이션, 딥 러닝, 강화 학습, 낙상 감지, IMU 센서, 장단기 메모리

학번: 2021-21985

Contents

요약	i
Contents	ii
List of Figures	v
List of Tables	vi
Chapter 1 서론	1
Chapter 2 관련연구	5
2.1 낙상	5
2.1.1 낙상의 정의	5
2.1.2 낙상의 원인	5
2.2 낙상 예방 및 감지 시스템	6
2.2.1 Non-wearable 시스템	6
2.2.2 Wearable 시스템	6
2.3 낙상 데이터 수집 방법	7
2.3.1 사용자 모집 기반	8
2.3.2 시계열 데이터 증강을 통한 생성	8

2.3.3	비디오를 통한 생성	8
2.3.4	3D Environment를 통한 생성	9
2.4	물리 기반의 캐릭터 동작 학습 및 제어	9
2.5	외력에 반응하는 동작 합성 방법	11
Chapter 3	시뮬레이션을 통한 낙상 동작 생성	12
3.1	심층 강화 학습 환경	12
3.1.1	정책 학습	12
3.1.2	상태	13
3.1.3	Action	13
3.1.4	보상	14
3.1.5	End of episode	17
3.2	낙상 동작 생성 기법	17
3.2.1	다양한 외력과 낙상 환경	17
3.2.2	Fall-VAE	18
3.2.3	보상 전환	20
Chapter 4	낙상 감지 시스템	21
4.1	LSTM을 이용한 낙상 분류기	21
4.1.1	입출력	22
4.1.2	학습 데이터와 테스트 데이터	22
4.1.3	전이 학습	22
4.1.4	활성화 함수, 학습 파라미터 외	22
Chapter 5	실험	24
5.1	생성된 낙상 동작 분석	24
5.2	낙상 감지 시스템 구성	26
5.2.1	데이터세트	26

5.3 낙상 감지 시스템 결과	31
5.4 가상 데이터세트 분류 결과	32
5.5 전이 학습 결과	32
5.6 종합 결과	34
Chapter 6 고찰 및 결론	35
Bibliography	38
Abstract	42

List of Figures

Figure 3.1	신체의 주요한 부위: 얼굴, 목, 가슴	16
Figure 3.2	Fall-VAE를 통해 생성된 자세	18
Figure 3.3	보상 전환 예시	20
Figure 5.1	생성된 낙상 동작	25
Figure 5.2	Fall-VAE가 적용되지 않은 낙상 동작	25
Figure 5.3	SmartFall 낙상 데이터 예시	27
Figure 5.4	합성 낙상 데이터 예시	28
Figure 5.5	센서 부착 위치	30
Figure 5.6	LSTM을 이용한 SmartFall 낙상/비낙상 학습 결과	32
Figure 5.7	LSTM을 이용한 전이 학습 결과	33

List of Tables

Table 5.1	전이학습을 실시하지 않고 가상데이터를 분류한 Confusion Matrix	32
Table 5.2	전이 학습을 통한 Confusion Matrix.	33
Table 5.3	전이 학습을 통한 가상데이터세트가 가장 좋은 성능을 보이 고 있다.	34

Chapter 1

서론

낙상은 넘어지거나 떨어져서 몸을 다치는 것을 말한다. 전 연령에서 발생 가능하지만 특히 노인층에 있어서 신체의 심각한 손상을 야기하고 심한 경우 사망에 이를 수도 있다. WHO에서 작년에 발행한 자료에 따르면 매년 약 684,000명이 낙상으로 인한 사망에 이르고 있다. 웨어러블 기기를 이용한 낙상 감지 시스템이 현재 상용화 되어있으나 아직은 충분하게 기술이 성숙하지 않아 보다 발전이 필요한 상황이다. 이를 위해서 많은 데이터를 필요로 하고 있다.

딥러닝의 발전으로 많은 분야에서 기존의 문제들을 해결할 수 있었지만 여전히 관성 센서를 이용한 분야에서는 데이터를 생성하는 일이 어렵다. 이 분야는 실험실 환경에서 수 시간 정도의 데이터를 생성하는데 그치는 경우가 많다. 관성 센서 데이터를 생성하는 일은 비싸고 수작업을 요하는 경우가 많다. 사람의 수작업이 포함되는 만큼 레이블링에 오류가 발생할 가능성도 높다. 또한 실험실 환경에 국한하여 데이터를 생성하는 경우가 많아 실제 환경에서 데이터셋트가 잘 작동하지 않을 수 있다. 그리고 동작과 인체가 다양하여 적절한 데이터셋트를 확보하는 것의 어려운 부분으로 데이터 수집 조건을 잘 설계한 경우에도 모든 경우에 맞는 데이터를 생성할 수 없다.

특히 낙상의 경우 실제 낙상 데이터를 만들기가 매우 어렵다. 이는 낙상 동작의 특성상 심한 경우 죽음이나 회복 불가능한 부상에 이를 정도로 위험하기 때문이다. 그러므로 낙상은 실험실 환경에서 생성한 데이터가 절대 다수를 차지하고 있다. 낙상의 또 다른 어려움으로는 낙상이라는 정의 아래 다양한 형태의 동작이 포함되어 있다는 것이다. 예를 들면, 바닥이 미끄러워 미끄러지듯이 넘어지는 경우, 신경의 작용으로 기절하는 경우, 관절의 가동 범위가 줄어들어 보행이 불안해져 넘어지는 경우, 환경적인 이유로 넘어지는 경우 등이 있을 수 있다. 모두 다 낙상의 정의에는 부합하지만 관성 센서와 사람이 보는 동작의 입장에서 보면 각기 다른 특성을 지니는 동작일 수 있다.

실제 사람을 이용한 낙상 데이터의 획득이 매우 어렵기 때문에 본 연구에서는 가상의 물리 환경에서 사람 형태의 캐릭터를 사용하여 자연스러운 낙상 동작을 생성하고 이를 통해 합성 관성 데이터를 생성하고자 한다. 일반적인 캐릭터 애니메이션 분야에서는 물리 환경에서 낙상에 대한 연구가 널리 이루어지지 않아 본 연구에서 자연스러운 낙상을 정의하고 생성하는 방법에 대해 연구하였다. 최근 컴퓨터 비전의 발전으로 비디오를 통해 스켈레톤을 추출해 관성 데이터를 얻는 방법이 많이 연구되고 있으나 마찬가지로 낙상의 영상은 쉽게 획득할 수가 없어 어려움이 있다. 물리 환경에서 시뮬레이션이 가능하다면 낙상 데이터 획득에 큰 도움이 될 것이다. 그런 후에 이 합성 관성 데이터가 어떻게 쓰일 수 있는지를 보이고자 한다.

캐릭터 애니메이션은 적은 모션 클립을 학습시켜 물리 환경에서 캐릭터를 학습 할 수도 있고, 모션 클립과 같은 주어지는 데이터가 없어도 어떤 보상 체계를 통해 움직임을 만들어 낼 수 있다. 즉, 목적에 맞는 자연스러운 동작을 다양하게 생성할 수 있으며, 이를 이용해서 어떠한 도메인에 적용할 수 있는 것이다. 게임에서 캐릭터 컨트롤을 하거나, 영화에서 가상 캐릭터의 동작을 생성할 때 주로 사용되지만 본 연구에서는 가상 물리 환경에서 생성한 동작을 이용하여 물리량들을 추출하는 것에 중점을 두었다. 물리량의 예시로는 가속도, 각속도, 속도, 에너지, 엔트로피, 모멘텀 등이 있으며, 낙상 동작에서 추출한 이 물리량들을 이용 데이터세트를 구

축하고 이를 낙상 감지 분야에 적용해보고자 한다.

자연스러운 낙상 동작 생성을 위해서는 사람의 내재적인 보상 시스템을 모델링하였다. 이러한 동작을 만드는 것에는 절대적인 레퍼런스 동작이 존재하지 않고, 사람은 몸에 가해지는 충격을 최소화하도록 내재적인 동기가 있다는 가정을 이용하였다. 접촉하는 힘이 최소화되도록 하는 보상을 주어 심층 강화학습을 사용해 넘어지는 동작을 생성하였다. 그리고 신체 부위별로 같은 충격량이 주어지더라도 다른 가중치를 부여하였다. 머리와 장기가 있는 몸통은 가중치를 높게 부여하였고, 다른 신체 부위들은 상대적으로 낮게 조절하였다. 그리고 머리와 몸통에는 충격량을 기준점을 정하여 과도한 충격이 가해지는 경우는 그러한 동작을 학습하지 못하도록 학습을 종료하였다.

생성한 동작에 자연스러움을 추가하기 위해 Fall-VAE를 사용하였다. 스턴트 배우에 의해 의도된 낙상 동작들의 모션 캡쳐 데이터를 수집하여 인코더를 만들어 강화학습 시에 그 잠재 변수를 학습하도록 하였다. 또한 보상 전환 기법을 이용해 다양한 동작을 생성하였다. 최초에 걷는 동작을 모방학습을 통해서 동작을 학습한 후에 이후에 걷는 동작에 대한 보상을 배제하고 외력을 주어 낙상 상황을 생성하고, 낙상 상황에서는 앞서 말한 충격을 최소화하는 보상을 사용하도록 학습하였다. 초기 상태를 고정시키지 않고 걷는 동작의 다양한 시간 단계 중에 외력과 보상 전환으로 다양한 상황에서 낙상 동작이 생성될 수 있도록 하였다. 외력은 그 세기와 방향과 외력이 주어지는 신체를 다양하게 설정하였다.

가상 낙상 동작을 이용하여 가상 관성 데이터세트를 생성하였다. 관성 데이터는 머리, 척추, 양 손, 양 발목에 센서를 부착해 수집하였다. 가능한 물리량 중에서는 선가속도와 각속도를 추출하였다. 그리고 낙상과 낙상이 아닌 경우를 구분하여 레이블을 자동으로 생성되도록 하였다. 낙상의 정의에 기반하여 발 이외의 신체 부위가 바닥에 닿는 경우를 낙상으로 지정하였다.

결과적으로 낙상에 관련된 동작을 다양하게 생성할 수 있었다. 우선 어떠한 보상도 주지 않고 외력을 가했을 때는 캐릭터가 어떠한 규칙도 없이 주어진 힘과 중력에 의해 바닥에 떨어진 래그돌 같이 보였다. 그러나 신체 중요 부위에 대한

충격 최소화를 조건으로 학습하여 생성한 낙상 동작의 경우에는 낙상으로 부터 신체의 중요한 부위를 지키려는 동작이 생성되었다. 다만 몸통과 머리를 지키기 위해 과도하게 바닥으로 부터 멀리하는 동작이 생성되는 경우가 있어 자연스러움이 부족한 경우가 있었다. Fall-VAE를 적용한 경우에는 팔로 바닥을 앞뒤로 짚고 넘어지는 등 충격을 최소화하는 동작들이 나타났다. 그리고 환경에 의한 낙상 동작도 생성하였다. 바닥이 미끄러운 경우나, 바닥이 움푹 패인 곳을 지나가는 경우의 낙상도 자연스럽고 그럴듯한 동작이 나타났다.

이렇게 생성된 동작과 데이터를 이용해 낙상 감지 시스템을 구현하고 성능을 검증하였다. 낙상 감지 시스템은 LSTM을 이용해 구현되었으며, 실제 낙상 데이터세트인 SmartFall의 데이터로 우선 학습하고, 전이 학습을 통해 가상 데이터세트를 다시 학습하였다. 총 데이터의 Window를 40개로 정해서 그 구간에서 낙상 데이터를 25개 전체를 포함하는 경우 낙상으로 판정 정의 하였으며, 이 경우 가상 데이터세트의 낙상과 비 낙상을 구분하는 정확도가 96.9%로 나타났다.

캐릭터 애니메이션 분야에서 최근 각광 받기 시작한 물리 환경에서의 동작 생성 기술을 이용해서 낙상 감지 분야에 필요한 합성 데이터세트를 생성하는 문제에 적용해 보았다. 가상 환경에서 동작을 만드는 일은 어려운 일로 여겨졌으나 심층 학습과 강화학습의 발전, 그리고 모방학습 등 다양한 연구를 통해 점점 해당 분야 이외의 사람들도 사용 가능한 기술으로 발전하고 있다. 본 연구는 낙상 감지 분야에서 그러한 가능성을 발견하였다. 그리고 가상 환경에서 낙상에 대처하는 사람의 고유한 보상을 모델링하여 자연스러운 동작으로 생성해내고, 이를 통해서 가상의 낙상 데이터 세트를 구축하였다.

Chapter 2

관련연구

2.1 낙상

2.1.1 낙상의 정의

낙상이란 사고나 실신, 경련, 마비 등의 여러 가지 원인에 의해 의도하지 않게 바닥이나 아래쪽의 표면으로 신체가 이동하는 것을 말한다. 또는 낙상의 사전적 의미는 떨어지거나 넘어져서 다침, 또는 그런 상태를 말한다. WHO의 정의에 따르면 본인의 의사와 상관없이 발바닥 이외의 신체 일부가 바닥면에 접지한 경우를 말한다 [1]. 또한 낙상은 주요한 내인성 사고나 위험한 사고의 결과가 아닌, 안정 상태에서 바닥이나 낮은 곳으로 비의도적으로 발생되는 사고로 [2] 서 있거나 앉아있거나, 누워 있는 상태에서 갑자기 떨어지는 것을 의미한다.

2.1.2 낙상의 원인

낙상은 매우 다양한 원인으로 인해 발생한다. 주로 외부적인이나 생리적인 요인으로 발생할 수 있고 이는 가상 시뮬레이터에서 낙상을 만들기 어렵게하는 요인이 된다. 주요한 낙상의 원인에는 연령이나, 낙상의 공포와 같이 모델링하기 어려운

요인도 있으나, 생리적인 원인의 경우 하체 근육의 약화나 균형의 상실 같은 근골격계 시뮬레이터를 사용하는 경우 모사가 가능한 원인도 있다. 본 연구에서는 토크 기반의 캐릭터 시뮬레이터를 사용하므로 다양한 외력을 주어 낙상을 유도하였다. 이는 직관적인 방법으로 특정한 동작이 가능한 캐릭터에게 Perturbation을 가함으로써 낙상을 유도하는 방법이다. 임의의 힘을 전후좌우 등 임의의 방향으로 다양한 신체 부위에 외력을 가하여 낙상을 유도하고 이를 통해 발생가능한 낙상 동작을 생성하였다.

2.2 낙상 예방 및 감지 시스템

다양한 방법으로 낙상 예방 및 감지 시스템을 만들 수 있으나 크게 Non-wearable 시스템과 Wearable 시스템의 두 범주로 나눌 수 있다.

2.2.1 Non-wearable 시스템

Non-wearable system은 (1) 비전 기반의 센서와 (2) 바닥 기반의 센서로 나눌 수 있다. 비전 기반의 센서로는 RGB-D 카메라, 적외선 센서, 레이저 범위 스캐너(LRS) 등이 있다. 이러한 센서를 통해 얻은 정보를 바탕으로 이미지 프로세싱 등을 사용해 분석할 수 있다. 예를 들어, 비디오 감시 카메라의 경우 지속적으로 포착한 이미지를 입력으로 이용하여 어떠한 알고리즘이 낙상이 발생했는지 아닌지 판단할 수 있을 것이다. 바닥 기반의 센서의 경우 지면반력(Ground Reaction Force, GRF) 센서나 사람의 발에 의해 추출된 힘을 관찰하는 압력 센서를 이용해 낙상 여부를 알 수 있다. Non-wearable system의 경우 사무실, 집 등에 별도로 설치가 필요한 경우가 많아 적용 범위에 한계가 있다는 단점이 있다.

2.2.2 Wearable 시스템

Wearable system은 인체에 부착된 센서나 기기를 이용해 데이터를 수집한다. 주로, 가속도계, 자이로스코프, 자기계, Inertial Measurement Unit(IMU) 등을 센서로

활용한다. 가속도계는 축을 따라 물체의 가속을 측정하는 장비이다. 자이로스코프는 방향을 고려하여 회전의 변화를 감지한다. 그러므로 세 축을 따라 각속도를 계산해낼 수 있다. IMU는 가속도계, 자이로스코프, 자기계로 구성된 센서이다. 이러한 종류의 센서들은 스마트폰, 스마트워치에 일반적으로 내장되어 있어, 별도의 시스템 구축이나 기기 구입을 필요로 하지 않으며, 사람의 몸에 부착하는 것이므로 실험실 밖 실제 환경에서도 이용이 가능하다. 다만 모바일 기기의 경우 전력의 제한이 있어 사용 시간이 제약되는 경우가 있고, Wearable을 통해 얻은 데이터는 추가적인 가공을 필요로 한다.

2.3 낙상 데이터 수집 방법

데이터 수집은 낙상 방지 시스템의 가장 중요한 부분이다. 현재 대부분의 데이터세트는 실험실 환경에서 통제된 방법으로 획득되었으며, 참여자들은 실험자의 지시에 따라 해당 동작을 수행했다. Usami [3]의 33개의 낙상 방지 관련 논문 조사 결과 데이터 획득을 위한 기기로는 IMU Sensors 73%, 스마트폰 기반의 센서 15%, sEMG 센서 3%, 카메라/비디오프레임 6%과 같은 비율로 사용된 것으로 조사되었다. 센서의 부착 위치는 허리가 31%로 가장 높았고, 손목 13%과, 척추 13% 가 그 뒤를 이었다. 센서의 갯수는 1개를 사용한 논문이 11개로 가장 많았고, 2개를 사용한 논문이 4개였다. 뒤이어, 10개 센서 사용한 논문이 2개이고, 센서 3,4,6개를 사용한 논문이 각각 하나였다. 실제 데이터를 얻는 방법은 매우 시간 소모적이고, 데이터를 분류하는 작업은 노동집약적이다. 또한 영상의 경우에는 사생활 침해의 문제가 발생할 수 있다. 이러한 문제들로 인해 충분히 큰 규모의 데이터를 얻기는 어렵다. 사용자 행동 인식(Human Activity Recognition) 문제에서 Wearable 센서 데이터세트는 그 규모가 PAMAP2의 경우에는 약 20명의 참가자와 십 수 시간에 달하는 녹화 데이터로 이루어져있다. 또한 수집한 데이터의 대표성 또한 문제가 될 수 있다. 적은 수의 샘플이기 때문에 실제 수많은 사람들과는 맞지 않을 수 있다. 특히 낙상 데이터의 경우 실제 데이터를 얻는 방법이 매우 위험하고, 또 실제

위험군에 있는 피실험자를 사용하기 보다는, 위험이 낮은 군의 실험자를 이용하는 경우가 많았다. 즉, 낙상의 위험은 65세 이상에서 특히 두드러지고 빈번히 발생하는데 낙상 데이터 자체는 20-30대에서 획득하는 경우가 있었다. 이러한 데이터의 부족은 또한 이 분야에 있어 딥러닝을 이용한 발전을 더디게 하였다.

2.3.1 사용자 모집 기반

실제 피실험자들에게 IMU센서를 착용시킨 후 특정한 동작을 지시 아래 행동하게 하는 방법이다. 센서를 찬 사용자들이 하는 행동을 기록하고 관찰하여 이를 모아 데이터 세트로 만든다.

2.3.2 시계열 데이터 증강을 통한 생성

Gonzales [4]는 wearable 기기에서 생성하는 3축 가속 센서를 대상으로 하여 낙상 감지 모델에서 시계열 데이터를 증강하고 드롭아웃을 적용하여 더 강건하고 정화한 모델을 만들 수 있음을 보였다. 기계학습을 이용해 낙상 감지와 분류의 문제를 푸는 주요한 방법으로 삼았고, 특징 추출 단계에서는 서포트 벡터 머신을 이용해 시계열 윈도우를 분류하고 시계열 데이터 증강을 하였다. 그리고 Pita [5]의 연구에 따르면 낙상 감지 문제에서 영상 데이터 증강을 이용하는 경우에도 유효함을 알 수 있다.

2.3.3 비디오를 통한 생성

Kwon [6]은 IMUTube를 통해 기존에 온라인에 존재하는 비디오들을 신체에 부착된 Wearable 센서 기반의 사용자 행동 인식 데이터세트으로 변환시킬 수 있음을 보였다. 컴퓨터 비전의 발전으로 영상에서 사람의 신체를 찾고 추적해낼 수 있으며 이를 비디오에서 IMU 데이터로 추출 하는 것이다. 영상의 제목과 설명을 이용해 자동으로 레이블 된 IMU 데이터를 얻을 수 있으며, IMUTube는 영상을 첫번째로 2D 포즈로 변경하고 이를 입력으로 하여 다시 3D 포즈를 생성한다. 그리고 각 관절의 IMU 데이터로 추출하는 것이다. 벤치마크 테스트를 통해서 그 성능

을 입증하였다. 랜덤 포레스트 기법을 이용한 실험에서 실제 데이터 세트와 가상 데이터 세트로 학습한 경우를 각각 비교하였을 때, Realworld 데이터 세트를 기준으로 98%의 성능을 보이고, Opportunity 기준으로 94%, Pamap2 (locomotion) 을 기준으로 96%, 그리고 Pamap2 (complex) 를 기준으로 80%의 성능을 보였다. 특히 실제 데이터를 가상 데이터와 혼합해 사용하는 경우 뛰어난 성능을 발휘하였다. 그리고 위의 2.3.2 절의 시계열 데이터 증강을 통해 생성하는 것과 마찬가지로 영상을 이용해 인식하는 경우를 위해 영상 데이터를 증강해서 데이터를 생성하는 경우도 있었다 [7].

2.3.4 3D Environment를 통한 생성

디지털 인간의 물리 모델을 이용해 낙상을 시뮬레이션하고 환경과 상호작용하도록 구현하여 합성 데이터를 생성한 연구가 있다. Zherdev [8]은 다양한 텍스처와 메시와 골격을 이용해 Unreal Engine 4를 이용해 3차원 장면을 생성하고 이 속에서 디지털 인간의 다양한 행동을 구현한 후, 충돌하는 경우를 생성하였다. 그리고 디지털 사람과 환경 사이의 충돌 마스크를 생성해 낙상 데이터를 생성하였다. 이처럼 컴퓨터를 이용한 시각화와 렌더링을 이용하면, 레이블이 완벽한 데이터를 얻을 수 있다.

2.4 물리 기반의 캐릭터 동작 학습 및 제어

최근 컴퓨터 그래픽스 중 캐릭터 애니메이션 분야의 발전으로 물리 기반으로 캐릭터의 동작을 학습하고 제어를 할 수 있게 되었다. 적은 분량의 모션 캡쳐 기반의 Kinematic 데이터를 바탕으로 학습하여 물리 환경에서 동작하는 캐릭터를 컨트롤 할 수 있게 된 것이다. 이 경우에는 물리 기반으로 학습한 캐릭터는 더 복잡한 환경과 상호작용할 수 있게 되고, Perturbation에 강건할 수 있다. 예를 들면, 걷다가 블록이 날아오는 경우 부딪혀도 스스로 걷기 동작을 복원해서 지속할 수 있게 된다. 그리고 걷기 방향을 갑자기 바꾸는 경우에도 자연스러운 모션을 이용해 동작을

변화하는 것이 가능하다. Peng [9]은 Deep Mimic 논문에서 캐릭터와 레퍼런스 동작과 태스크를 제공해 정책을 학습하고 하고 이 정책으로 레퍼런스 모션을 따라하면서 물리 시뮬레이션된 캐릭터가 특정한 태스크를 수행하도록 하였다. 여러개의 모션 클립을 합한 정책이 나타날 수도 있고 다른 시뮬레이션 캐릭터나 환경에도 리타게팅이 가능하다. 그리고 사자와 공룡과 같이 모션 캡쳐가 어려운 동물 캐릭터의 경우에도 애니메이터가 수작업으로 생성한 키 프레이밍 동작도 모방을 할 수 있다. 이 연구에서 보상 함수는 두 개의 항으로 이루어져있다. 모방 목적함수는 캐릭터가 레퍼런스 모션을 흉내내도록 한다. 태스크 목적 함수는 캐릭터가 특정한 목적을 만족하도록 한다. 이를 통해 뒤공중제비, 옆으로 재주넘기, 손 안 짚고 옆으로 재주 넘기, 구르기와 같은 난이도가 높은 동작도 물리 환경에서 수행할 수 있음을 보였다. 또한 동작을 수행하는 도중 날아오는 박스와 충돌하는 Perturbation을 발생시켜도 강건한 모습을 보였다. 하지만 레퍼런스 동작이 없는 경우에 이상하게 행동함을 보였다.

앞선 방법은 참고할 수 있는 모션 클립이 있는 경우에만 유효하다. 하지만 그러한 모션 클립이 없이도 어려운 동작을 생성해 목표를 수행하는 프레임워크가 존재한다. Yin [10]은 높이 뛰기 동작의 전략을 발견하는 프레임워크를 제시하였다. 이 프레임워크의 결과로 캐릭터는 학습한 모션 클립에 존재하지 않지만, 현실에서 가장 효율적인 높이뛰기 동작인 배면뛰기(Fosbury flop) 동작으로 1.7cm의 캐릭터가 2.0m 의 높이뛰기를 자연스러운 동작으로 뛰어 넘는데 성공하였다. 이 연구에서 학습은 두 단계로 이루어진다. 우선 모션 클립을 활용해 달리는 모션을 학습하여 넘어지지 않고 달릴 수 있는 컨트롤러를 학습한다. 그리고 도움닫기 상태에서 높이뛰기를 시도하는 점프 컨트롤러로 변경해 달성해야 할 태스크와 Sparse 보상을 가지고 학습하였다. 뛰기 동작에서 이어지는 낮은 단계의 높이 뛰기부터 시작해, 점차적으로 높이를 높이는 커리큘럼 학습을 사용하였다. 세부적으로 첫번째 단계에서는 Bayesian Diversity Search를 통해서 높이 뛰기 전략을 찾고, Take-off State를 찾아 새로운 정책을 계속 찾도록 학습 하였다. 그리고 다양한 Pose variational autoencoder를 사용해 자연스러운 모션을 학습하였다. 본 연구도 이

와 유사하게 모션 클립에 있는 데이터를 일부 활용하지만 다양한 낙상의 경우에는 물리적 환경에 따라 동작이 생성될 수 있도록 하였다.

2.5 외력에 반응하는 동작 합성 방법

주어진 외력에 반응하는 동작 합성 방법 또한 다양하게 연구되었다. Ye [11]는 가상의 캐릭터가 예측하지 못한 외부의 Perturbation에 대해 반응하는 동작을 합성하였다. Arikhan [12]는 외부의 원천에서 의해 밀리는 캐릭터의 동작을 합성하였다. 녹화된 데이터의 집합에서 동작을 골라 합성하는 방법을 사용하였다. Hoyet [13]은 캐릭터간 물리적인 접촉을 시뮬레이션하는데 발생하는 에러를 두 캐릭터를 이용해 몸통을 미는 동작과 반응을 모션캡쳐하여 일련의 인지적인 실험을 수행하였다. Zordan [14]은 충격에 대한 반응과 같은 예측되지 않은 상황에 대해 모션 캡쳐 기반의 동작 시스템을 접촉 힘에 반응하도록 구현하였다. 모션 캡쳐 세그먼트에 연결하여 물리 기반의 반응을 생성하도록 했다. Lee [15]의 경우에는 사용자의 입력과 임의로 주어지는 외부적인 방해에 의해 기민하게 반응하는 캐릭터의 모션 데이터를 표현하였다. 모션 필드라고 불리는 벡터 필드를 이용해 실시간에 모션을 합성하는 방법이다. 하지만 이런 경우 대부분의 연구는 강건하게 현재의 상태를 유지할 수 있거나, 그에 따른 자연스러운 반응을 생성해내는 목적을 중심으로 하여 자연스러운 넘어지는 동작에 대해서는 많이 연구가 되어있지 않다.

Chapter 3

시뮬레이션을 통한 낙상 동작 생성

3.1 심층 강화 학습 환경

하나의 제어기를 통해 심층 강화학습을 실시하였다. 그리고 두 가지 구간을 두었는데 외력인 Perturbation을 가하는 순간을 기점으로 (1) 모방 구간 (2) 낙상 구간으로 구분 하였다.

3.1.1 정책 학습

학습 알고리즘은 Proximal Policy Optimization (PPO) [16] 방법을 사용하였다. PPO는 정책 네트워크와 가치 함수 네트워크를 학습한다. PPO는 17년 등장했을 당시에는 최첨단 (State-of-the-Art) 알고리즘이었고 현재에도 널리 사용되고 있다. 사람의 동작에 대해서 구현과 결과에 있어서 가장 좋은 결과를 내고 있다.

3.1.2 상태

본 제어기의 상태는 다음과 같이 정의된다. 각 항들은 위치, 속도, 시간, 그리고 perturbation이 발생 여부를 나타낸다.

$$s = (s_p, s_v, s_{\text{phase}}, s_{\text{perturbation}})$$

$$s_p \in \mathbb{R}^{217}$$

s_p 은 캐릭터의 무게중심(Center of Mass)을 기준으로 캐릭터 body node 의 상태 위치와 캐릭터 body node 의 6d global orientation으로 표현되어 있다.

$$s_v \in \mathbb{R}^{147}$$

s_v 은 캐릭터의 무게중심의 선형 속도를 기준으로 캐릭터 body node 의 상태 선형 속도를 나타내는 항과 캐릭터 body node의 global 3d의 각속도로 표현되어 있다.

$$s_{\text{phase}} = [0, 1]$$

s_{phase} 은 perturbation이 들어오기 전 캐릭터가 수행하는 동작에 대한 phase 값을 나타낸다.

$$s_{\text{perturbation}} = 0, 1$$

$s_{\text{perturbation}}$ 은 외력 및 환경이 변하기 전인지 후인지를 나타내는 항이다. 전(모방 구간)이면 0, 후(낙상 구간)이면 1로 표현된다.

3.1.3 Action

강화학습의 Action은 다음과 같이 정의된다.

$$a = (a_{\text{skeleton}}, a_{\text{vae}})$$

a_{skeleton} 은 PD controller 에 들어갈 PD Residual 목표를 나타낸다. a_{vae} 은 Fall-VAE 의 디코더에 들어가여 PD 목표 자세를 생성할 잠재 공간 변수를 나타낸다.

3.1.4 보상

강화학습의 보상함수는 다음과 같이 정의된다.

$$r = \begin{cases} r_{\text{imitation}} r_{\text{residual}} r_{\text{latent}} & \text{if } s_{\text{perturbation}} = 0 \\ r_{\text{fall}} r_{\text{residual}} r_{\text{latent}} & \text{if } s_{\text{perturbation}} = 1 \end{cases} \quad (3.1)$$

Perturbation 전후에 따라 보상 함수는 달라진다. Perturbation 전, 즉 동작을 모방하는 구간 일 때, $r_{\text{imitation}} r_{\text{residual}} r_{\text{latent}}$ 이고, perturbation 이 들어와 모방동작을 할 때는 $r_{\text{fall}} r_{\text{residual}} r_{\text{latent}}$ 이다. r_{residual} 은 PD Residual 목표의 L1 노름을 최소화 하는 것이다.

$$r_{\text{imitation}} = r_{\text{position}} r_{\text{endeffector}} r_{\text{comvelocity}} r_{\tau} \quad (3.2)$$

$$r_{\text{fall}} = r_{\text{contactforce}} r_{\tau} \quad (3.3)$$

보상의 항들은 다음과 같이 정의된다.

$$r_{\text{position}} = \exp(-w_p \|q - q_{\text{target}}\|^2) \quad (3.4)$$

자세 보상인 r_{position} 은 캐릭터가 매 단계마다 관절의 위치를 레퍼런스 모션에 맞추도록 한다. 시뮬레이션 된 캐릭터의 관절 q Orientation Quaternion이며, 레퍼런스 모션인 q_{target} 간의 차를 계산한다.

$$r_{\text{ee}} = \exp(-w_{\text{ee}} \|p^{\text{ee}} - p_{\text{target}}^{\text{ee}}\|^2) \quad (3.5)$$

end-effector 보상인 r_{ee} 는 캐릭터의 손이나 발이 레퍼런스 모션에 맞추도록 한다.

$$r_{\text{comvelocity}} = \exp(-w_{\text{com}} \|v^{\text{com}} - v_{\text{target}}^{\text{com}}\|^2) \quad (3.6)$$

속도 보상인 $r_{\text{comvelocity}}$ 은 무게중심의 속도가 레퍼런스의 속도와 같도록 한다.

$$r_\tau = \exp(-w_{\text{com}} \|\boldsymbol{\tau}\|^2) \quad (3.7)$$

r_τ 는 토크에 관한 보상이다. 여기서 $\boldsymbol{\tau}$ 는 각 관절에 가해지는 토크를 가리키며 토크가 줄어들도록 학습하였다.

$$r_{\text{contactforce}} = \exp(-w_{\text{contactforce}} \|W \cdot F\|_1^1) \quad (3.8)$$

$r_{\text{contactforce}}$ 는 접촉력에 대한 보상이다. 각 신체부위별로 가중치를 더해 합하고 이것이 최소화 되도록 학습하였다. Figure 3.1.에 표시된 부분은 모두 W 를 1로 설정하여 충격량을 온전히 반영하였고, 이외의 신체 부위는 W 를 0.1로 설정하여 실제 받는 접촉력의 10%만 반영이 될 수 있도록 하였다. r_{fall} 에서는 접촉력의 합이 작은 방향으로 학습할 수 있도록 하였다. r_{latent} 는 Fall-VAE의 입력인 잠재 변수 z 값이 특정 임계값, c 를 넘어가는 것을 최소화하는 것이다.

$$r_{\text{residual}} = \exp(-w_{\text{residual}} \|a^{\text{skeleton}}\|) \quad (3.9)$$

r_{residual} 은 PD residual 목표 자세의 L1 노름을 최소화하는 보상이다.

보상 종합 다양한 낙상 동작을 생성하기 위해서 보상을 서로 다른 두 단계로 구분하였다. 모방 보상과 낙상 보상이다. 모방 보상 단계에서는 모방 보상, residual 보상, latent 보상으로 나누어 이를 모두 곱해서 보상을 계산하였고 모방 보상은 위치 보상, end-effector 보상, 무게중심 속도 보상, 토크 보상으로 구할 수 있다. 낙상 보상에서는 피해량을 최소화하기 위한 동작을 낙상 순간에 취할 수 있다고 가정하였고 접촉력을 기준으로 피해량을 정의하고 토크 보상과의 곱이 작아지도록 하여 보상을 정의하였다.

신체가 바닥에 부딪히는 경우 접촉력이 발생하게 되는데 이 크기가 크면 클수록 피해가 크다고 정의하는 것이다. 접촉력이 작아지는 경우 캐릭터는 더 많은 보상을 받게 된다. 그리고 사람의 신체 부위 별로 중요도를 다르게 하였다. 머리와

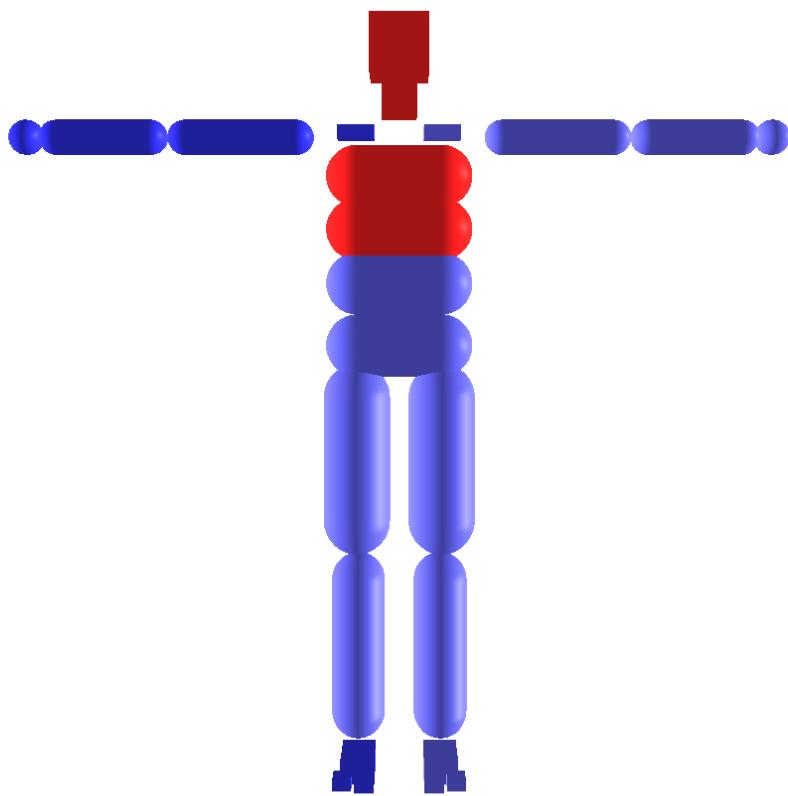


Figure 3.1: 신체의 주요한 부위: 얼굴, 목, 가슴

몸통 같은 신체 부위는 상대적으로 중요한 장기가 많이 포함되어있거나 충격을 받았을 때 고통을 느끼는 정도가 크다고 할 수 있다. 따라서 본 연구에서는 머리, 목, 가슴 부위는 충격량의 가중치를 1로 계산하였다. 이외의 부위의 가중치는 0.1로 계산하여 힘이 반영되도록 하였다.

3.1.5 End of episode

End of episode 조건으로는 중요한 부위의 경우에는 충격량이 일정 힘을 넘어가는 경우로 하여 학습을 종료시키도록 하였다. 본 연구에서는 중요한 부위는, 머리, 목, 가슴 부위의 접촉력(Contact force)의 합이 80N의 되는 경우 학습을 종료하도록 하였다. 낙상 할 때 의식이 있다면 신체에 위험이 크게 가는 것을 피하도록 본능적(intrinsic)으로 사람이 행동할 것이라는 가정으로 만든 종료 조건이다.

3.2 낙상 동작 생성 기법

3.2.1 다양한 외력과 낙상 환경

낙상의 원인은 사람 신체를 기준으로 내외부로 다양하게 발생할 수 있다. 본 연구에서는 외부적인 요소를 사용하여 낙상을 유도한다. 가상 시뮬레이션 환경에서 신체의 다양한 부위에 외력을 가하거나, 바닥의 상태를 변화시켜 낙상을 발생시켰다. 외력은 100N에서 600N 사이의 범위로 손과 팔을 제외한 임의의 신체요소의 질량중심에 가해진다. 외력의 방향은 바닥과 평행한 평면 상에서 임의로 주어진다. 바닥의 상태는 두 가지로 변화된다. 첫번째는 바닥의 마찰계수가 0이 되도록 하여 마찰력을 감소시키는 것이고, 두번째는 바닥의 높이를 20 cm 낮추는 것이다. 바닥의 상태 변화와 외력이 임의의 확률로 각각 혹은 동시에 가해지도록 하여 캐릭터의 낙상을 유도한다. 이 규칙은 제어기의 학습 단계와 데이터 생성 단계에서 동일하게 사용된다.

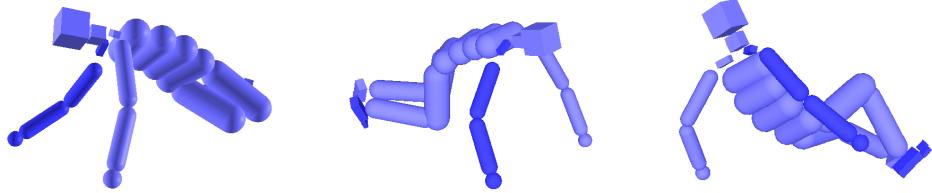


Figure 3.2: Fall-VAE를 통해 생성된 자세

3.2.2 Fall-VAE

자연스러운 모션 생성을 위해서 낙상 자세를 생성하는 Variational Autoencoder를 사용하였다. Yin [10]은 높이뛰기 동작을 생성할 때 보다 자연스러운 동작을 위해 Pose-VAE를 사용하였는데 이와 유사하지만 보다 적은 데이터를 이용한 방식을 사용하였다. 사람의 모션 클립들로부터 추출된 자세들을 바탕으로 생성하여, 자연스러운 1차 PD 목표 자세를 생성할 수 있는 모델을 사용한 것이다. 본 연구에서는 자연스러운 자세의 잠재 공간을 학습할 수 있도록 Fall-VAE를 제안한다.

Motion Clips Fall-VAE의 학습을 위하여 Mixamo [17]에서 낙상 관련 데이터 동작을 수집하였다. 넘어지거나, 점프하거나, 땅을 짚거나 하는 등의 동작을 다양하게 수집하였다. 전문적인 모션 캡쳐 연기자에 의해 수행된 동작으로, FBX의 형태로 제공된다. FBX binary를 기준으로 BVH 파일로 변환하여 사용하였다. BVH 파일 변환은 Blender에서 script를 이용하여 수행하였다.

낙상 관련 동작의 경우 8개의 넘어지는 동작과 1개의 낙상 상태에서 일어나는 동작으로 구성하였다. 일어나는 동작의 경우 역재생 하는 경우 넘어지는 것과 유사하여 포함하였다. VAE의 특성상 시각적으로 만족스럽고, 사람의 자연스러운 동작과 유사한 특정 자세를 생성하도록 하므로, 양자를 모두 포함하였다. 그리고 낙상 동작 이외에 걷기, 뛰기 등의 약 900개의 포즈 데이터를 사용하였으며, 중복이 되는 자세 데이터는 제외하고 사용하였다.

Fall-VAE 구조 및 학습 Fall-VAE는 표준 Variational Autoencoder 를 기반으로 한다. 인코더는 자세 q 를 저차원의 잠재 공간 Z 로 투영하며, 다음 손실 함수를 최소화하면서 학습을 진행하였다.

$$\mathcal{L} = \mathcal{L}_{\text{MSE}}(q, q') + \beta \cdot \text{KL}(\mathcal{N}(\mu_z, \Sigma_z), \mathcal{N}(0, 1)) \quad (3.10)$$

\mathcal{L}_{MSE} 항은 평균 제곱 오차를 재건하는 항이다. 그리고 두번째 항은 잠재 변수 분포가 표준 가우시안 분포를 따르도록 쿨백-라이블러 발산을 사용했다. 입력 피쳐 q 는 모든 관절의 각도로 이루어진 벡터이다. 각 관절의 각도는 6D Rotation representation 으로 표현되었다.

인코더와 디코더는 3개의 히든 레이어를 가지는 완전 연결 신경 네트워크 (Fully-Connected Neural Layer)를 사용해 구현하였다. 각각 레이어들은 512, 256, 128개의 LeakyReLU 활성화 함수를 가지고 있다. d_{latent} 는 잠재 변수의 차원으로 d_{latent} 변수의 갯수는 임의의 갯수인 6으로 설정하였다. 이를 바탕으로 전체 입력된 자세 데이터를 다룰 수 있도록 하였다. 잠재변수를 입력으로 하여 잠재 공간에 있는 자연스러운 낙상 자세를 출력으로 하도록 학습하였다. 시스템에서는 주어진 상태에서 action인 a_{vae} 를 생성하면 이것으로 동작을 생성하는 PD 목표 자세를 만들게 된다.

PD 목표 자세 합성 PD 제어기는 오류에 따라 동작이 발동할 수 있도록 (actuation) 한다. 그러므로 원하는 자세를 만들기 위해서는 실제 목표 자세가 특정한 분량 만큼의 오프셋을 필요로 한다. 외력에 의하거나, 혹은 내부적인 이유로 균형을 잃고 넘어지는 경우 역시 오프셋을 필요로 한다.

본 연구에서는 정책 $\pi_{\theta}(a|s)$ 에서 추출된 $a = (a_{\text{skeleton}}, a_{\text{vae}})$ 에서 자세를 추출하는 함수 $D_{\text{pose}}(\cdot)$ 에 a_{vae} 를 입력하여 1차적인 목표 자세를 구한다. 그리고 a_{skeleton} 을 더하여 최종적인 PD 목표자세를 구한다.

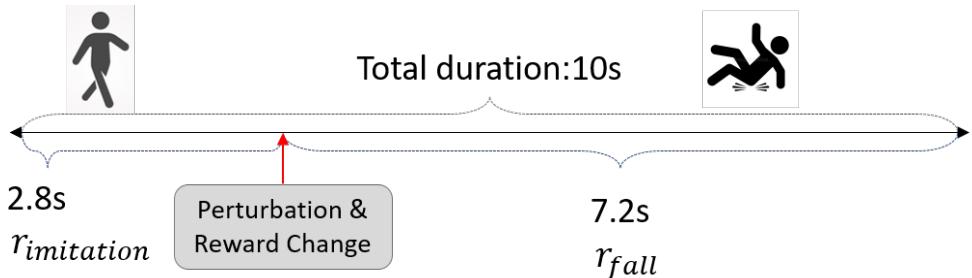


Figure 3.3: 보상 전환 예시

3.2.3 보상 전환

마지막으로 낙상 동작을 학습하고 생성하는 방법으로 보상 전환 기법을 사용하였다. 보상 전환 기법은 캐릭터가 특정한 모션을 잘 수행하는 $r_{imitation}$ 을 학습 시킨 후 n초의 시간동안 이를 수행한 후 외력이 주어지면서 동시에 이를 위의 낙상에 대한 r_{fall} 로 전환하는 기법이다. 앞서 설명한 기법들로 낙상 동작을 생성했을 때에는 기본적으로 정지 상태에서의 낙상 동작들이므로 낙상 동작이 자연스럽게 나타나더라도 동작의 다양성에서 충분한 결과를 얻지 못하였다. 낙상 전의 동작을 걷기나 뛰기 같이 동적으로 설정하여 이 단계에서 외력이 주어졌을 때 발생하는 다양한 동작을 관찰하고 이를 학습하였다. 본 연구에서 $r_{imitation}$ 은 걷기, 뛰기 동작으로 학습하였다. 이 동작의 경우 강화학습을 이용하고 모방학습 [9]을 통해 학습하였다. 이 동작에서 약 임의의 시간 이 후에 다양한 외력을 가함과 동시에 r_{fall} 로 전환해서 발생하는 동작들을 확인하였다. 외력을 가하는 경우 이외에도 마찰력이나 지형을 변화시켜 환경 변화에서 발생하는 다양한 동작들을 학습하고 이를 통해 데이터를 생성하였다.

Chapter 4

낙상 감지 시스템

4.1 LSTM을 이용한 낙상 분류기

낙상과 비 낙상을 감지하기 위해서, RNN 중 LSTM 모델이 사용되었다. 순환신경망 (RNN)은 심층학습 모델로 주로 시계열 속성을 가진 데이터를 처리하는데 사용된다. 동작 데이터를 처리하는데에도 순환신경망이 주로 사용된다. 하지만 RNN의 경우 입력과 출력의 간격이 길어질수록 장기 의존성 문제가 생긴다. RNN 모델은 경사하강법을 학습하는데에 있어 기울기가 소멸하거나 폭주하여 낮은 학습 성능이 문제가 된다. 장단기메모리(Long Short-Term Memory)는 RNN의 이러한 장기 의존 문제를 해결하기 위해 고안되었다. LSTM은 RNN의 은닉 상태에 cell-state 를 추가한 구조로 이루어져 있다. 이 특징으로 상태가 오래 경과하여도 기울기가 비교적 잘 전파되며 따라서 앞서 언급한 기울기가 소멸하는 문제에 대한 대응이 가능하다.

4.1.1 입출력

한 번에 40개의 샘플을 입력 윈도우으로 하였다. SmartFall의 경우 31.25Hz로 샘플링 되어 있으며, 낙상 구간의 데이터는 25개의 샘플을 포함하고 있다. 따라서 매 구간마다 25개의 낙상 데이터가 모두 포함되면 낙상으로 판정하도록 해당 윈도우의 레이블을 추가하였다. 만약 낙상 데이터의 전 구간을 포함하지 않으면 낙상 아님으로 레이블 하였다. [18]

4.1.2 학습 데이터와 테스트 데이터

학습 데이터는 SmartFall에서 기본으로 제공하는 학습 데이터와 트레이닝 데이터를 사용하였다. 그리고 본 연구에서 합성한 데이터세트의 성능을 평가하기 위해서는 합성 데이터세트의 10%를 기존 SmartFall을 이용해 학습한 모델을 다시 합성 데이터를 위한 학습 데이터로 사용하고 합성 데이터의 90%는 테스트 데이터로 사용하였다. 합성 데이터의 경우 100Hz로 생성하여 33.3Hz로 다운 샘플 하였다. 그리고 낙상의 갯수를 낙상이 발생한 시점에서 25개의 데이터를 포함하도록 정하였다. 낙상 동작은 1초 이내에 그 특징을 보이는 경우가 많아 해당 샘플로 충분하였다. 그리고 때에 따라 낙상이 발생한 이후에 비 낙상으로 주석이 달린 데이터의 경우 낙상으로 후처리를 진행하였다.

4.1.3 전이 학습

또한 SmartFall 데이터세트로 사전 훈련된 모델을 합성 데이터를 통해 재학습하였다. SmartFall 데이터로 학습하고 테스트하고, SmartFall 데이터로 학습한 모델에 전이 학습이 없이 합성 데이터세트를 적용했을 때와 전이 학습을 적용 후에 테스트를 각각 실시하였다.

4.1.4 활성화 함수, 학습 파라미터 외

tanh 함수를 활성화 함수로 사용하였다. 학습 파라미터들은 다음과 같다. 배치 크기는 128, hidden dimension은 256 차원으로 하고, 40 epoch 동안 학습을 진

행하였다. 15회 동안 정확도의 개선이 없는 경우 해당 회차의 학습은 진행하지 않았다. 그리고 Adam Optimizer와 Cyclic learning rate를 적용하였다.

Chapter 5

실험

우리는 앞서 사람의 본원적인 보상 체계를 바탕으로 낙상 동작을 생성하였다. 기술 컴포넌트가 추가될 때마다 점점 자연스러운 낙상을 학습해가는 것을 볼 수 있었다. 특히 각 단계별로 사람의 행동의 근원적인 주요한 부분을 모사하는 것으로 보였다.

5.1 생성된 낙상 동작 분석

충격 최소화 보상을 적용한 경우에는 팔 등을 이용해서 충격을 최소화 하려는 동작이 나타났다. 충격 최소화 보상을 적용하지 않은 경우에는 다음과 같은 평균 충격량을 보이며 낙상 동작이 생성되었다. 사람의 신체를 보호하는 본질적인 반응이 학습이 되었다고 할 수 있다.

충격 최소화 보상만을 적용했을 때는 다리가 과도하게 벌어지거나, 낙상 중에도 팔을 쭉 뻗치는 동작이 나타났다. 충격 최소화의 측면에서는 타당할 수 있으나, 시각적으로 자연스럽지 못하다 할 수 있다. Fall-VAE를 적용한 경우에는 팔과 무릎을 굽혀서 몸의 충격을 줄이고 자연스러운 동작이 생성 되었다.

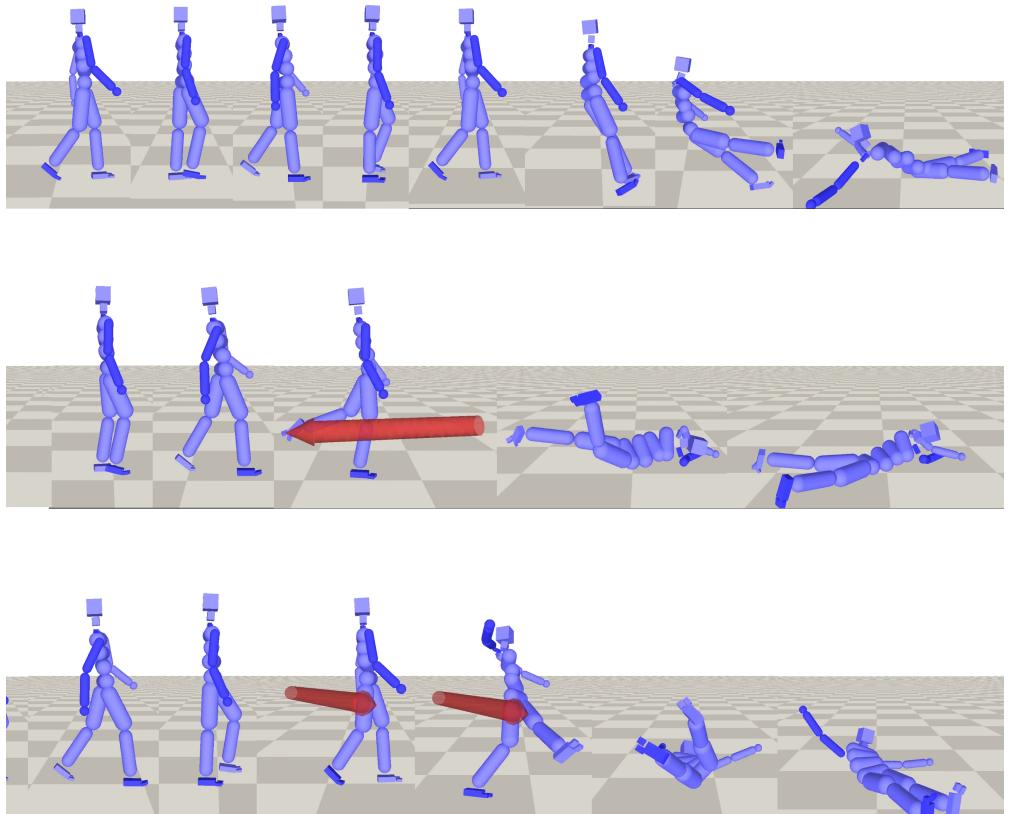


Figure 5.1: 생성된 낙상 동작

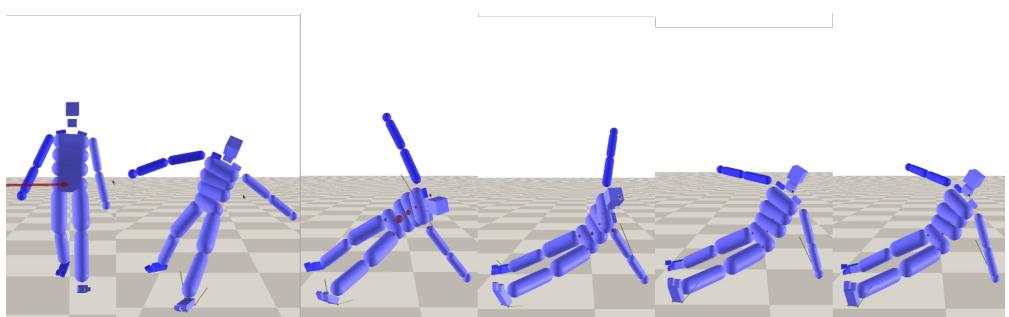


Figure 5.2: Fall-VAE가 적용되지 않은 낙상 동작

5.2 낙상 감지 시스템 구성

5.2.1 데이터세트

앞에서 생성한 데이터세트를 사용하여 낙상 감지 시스템을 구성하였다. 낙상 감지 시스템은 총 세 개의 데이터세트을 사용하였다. 두 개는 기존에 존재하던, 실제 피실험자들을 이용해 생성한 낙상 데이터세트이다. 하나는 물리 환경에서 시뮬레이션을 이용해서 생성한 데이터세트이다.

SmartFall

Mauldin [19]은 스마트워치 기반의 낙상 감지 시스템을 구현하면서 이를 위한 데이터를 수집하고 공개하였다. 이 데이터는 총 두 가지의 서로 다른 기기로 수집하였다. 한 가지는 마이크로소프트의 밴드 워치를 사용하였고 다른 것은 노치를 사용하였다.

SmartFall 스마트워치 데이터세트 31.25Hz로 샘플링 되어있다. 낮은 주파수에서도 잘 동작한다면, 저전력 모델인 스마트워치에서도 잘 동작할 것이다. 시뮬레이션 환경은 480Hz로 수행되므로 이보다 낮은 주파수는 쉽게 획득할 수 있으며, 이 이상의 주파수는 현재까지는 이 도메인에서는 사용할 일이 없다. Smartwatch 데이터세트는 7명의 지원자들이 MS Band watch를 착용한 데이터를 수집하였다. 지원자 모두는 신체 건강한 자들로 모집되어 ADL을 수행하였다. 나이는 21-55세 사이이며, 키는 5피트에서 6.5 피트 사이이다. 스마트워치는 왼손에 차고 미리 세팅된 조깅, 앓기, 물체를 던지기, 손을 흔들기의 ADL을 수행하였다. 데이터세트는 기본적으로 “비 낙상”으로 레이블되고, 12인치 높이의 매트리스에 앞, 뒤, 왼쪽, 오른쪽으로 떨어지는 낙상을 수행하였다. 각각 유형의 낙상을 10회씩 수행하였다. 샘플링 레이트는 4Hz, 1.25Hz, 62.5Hz가 지원이 되고 31.25 Hz로 정하였다. 4Hz는 전력 소모가 적으나 낙상의 중요한 정보를 놓치는 경우가 많다. 레이블링은 스마트폰 애플리케이션을 구현해 버튼을 누르면 낙상 그렇지 않은 경우 비 낙상으로 레이블 되도록 했다. 연구자에 의해 실시간으로 수집되나, 낙상 시점에 비해

SmartFall

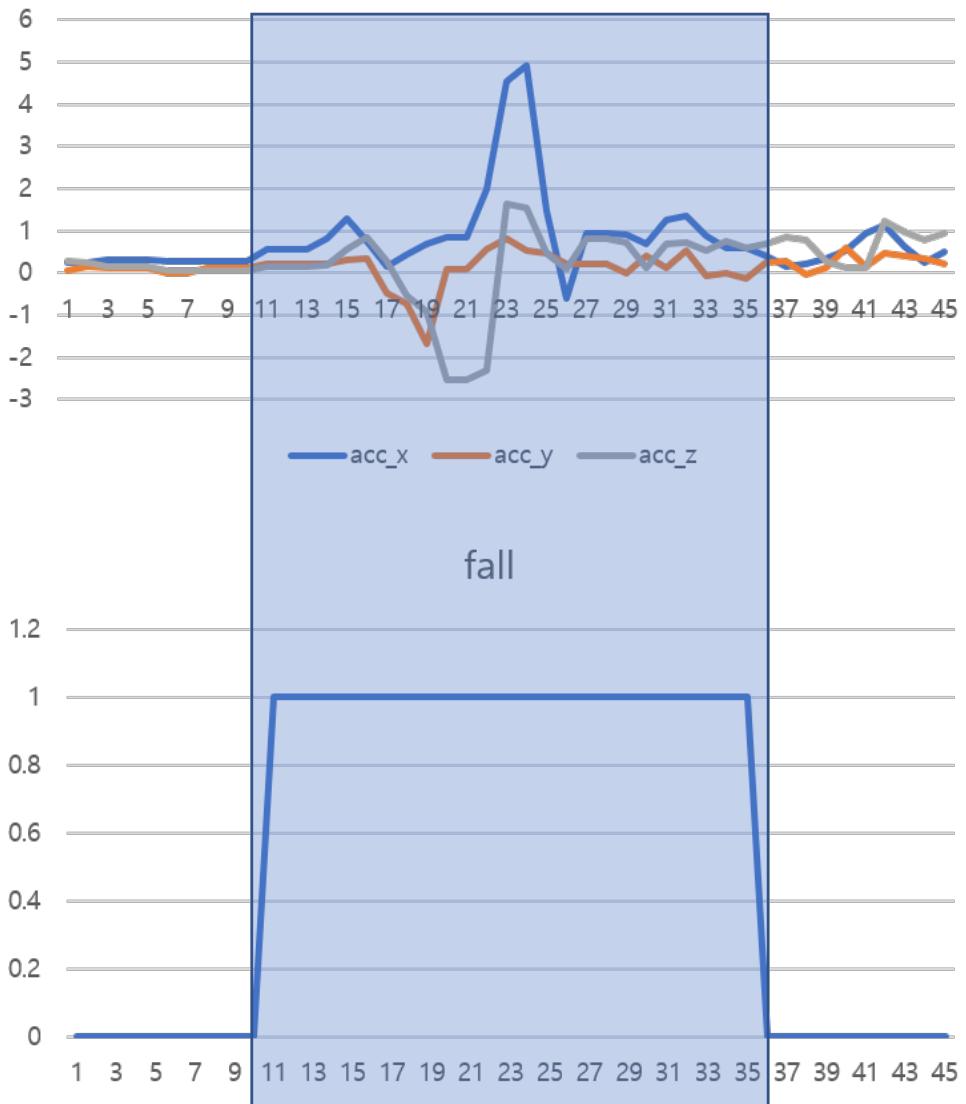


Figure 5.3: SmartFall 낙상 데이터 예시

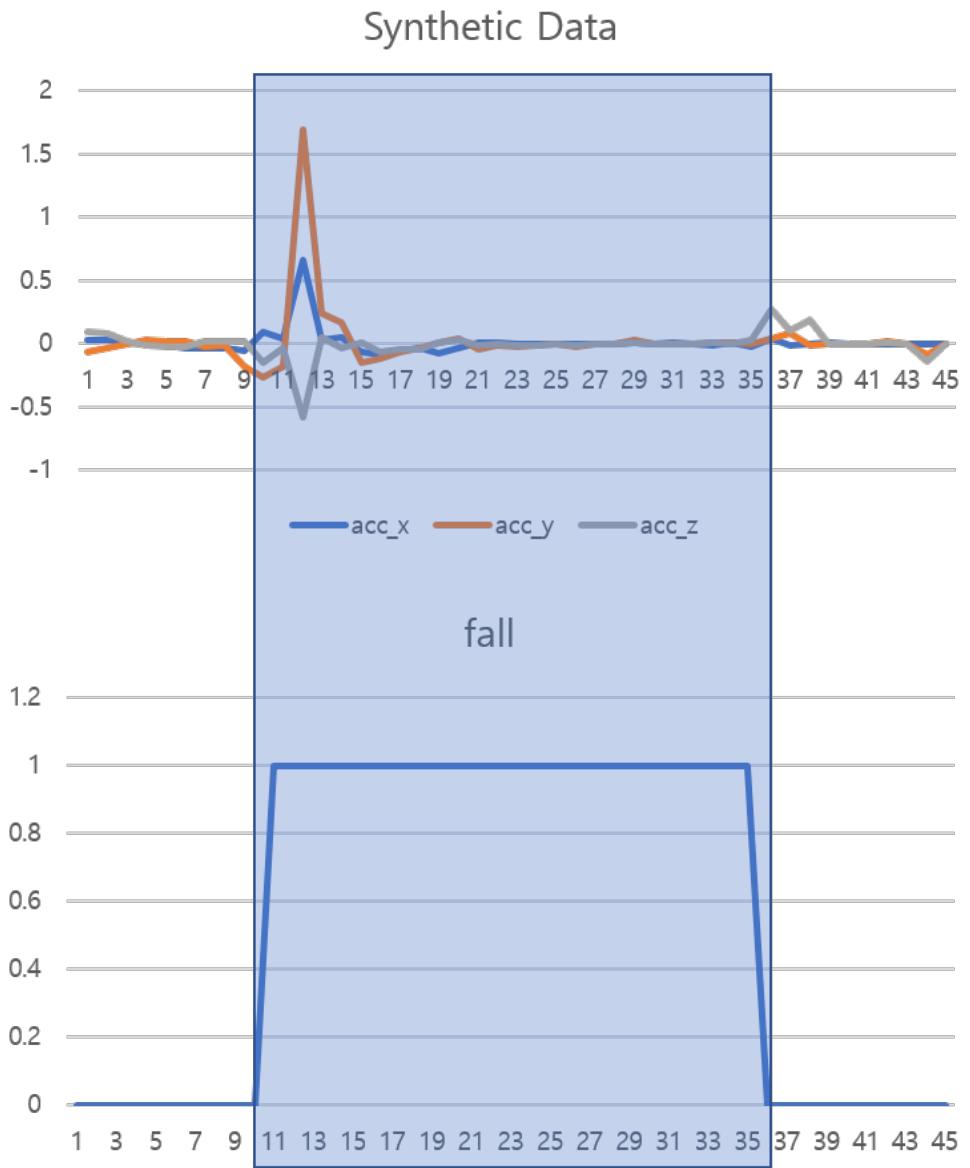


Figure 5.4: 합성 낙상 데이터 예시

버튼을 누르는 것이 너무 빠르거나, 늦거나 아니면 길거나 하는 오류의 가능성이 있다. 이러한 에러를 경감하기 위해 후처리를 통해 낙상의 주요한 단계만을 낙상으로 라베링하여 처리하였다. 모든 낙상 데이터의 가장 가속도가 높은 지점과 그 앞뒤를 낙상으로 지정하였다.

가상 데이터 세트

합성 데이터세트는 물리 환경에서 구축한 낙상 동작들을 이용하여 생성하였다. 낙상 동작들은 다음 위치에서 센서 정보를 통해 획득 되었다. 센서의 위치는 [Figure 5.5]와 같다. 관성 센서는 머리, 왼쪽 손목, 오른쪽 손목, 척추, 복사뼈 왼쪽, 복사뼈 오른쪽에 설치하였다. 가상 관성 센서는 해당 위치에서 얻은 물리량을 기록하도록 하였다. 관성 센서를 통해서 생성한 정보는 100Hz로 획득하였다. 시뮬레이션 시간 단계 (Time Step)는 480Hz로 고정되어있다. 이보다 낮은 주파수를 갖는 관성 센서의 정보는 손쉽게 획득할 수 있으며, 이 기술이 적용될 대부분의 모바일 기기의 특성상 높은 주파수가 필요하지 않다. 대부분의 실제 낙상 관성 데이터는 100Hz 보다 낮은 주파수로 수집되어있으며, 모바일 기기를 사용하는 경우 전력 사용량에 따라 이를 초과하는 주파수는 드물게 있었다. 오히려 주파수가 높은 경우 많은 데이터양을 처리할 수 있는 연산 능력이 부족한 경우가 보고되었다. 따라서 필요에 따라 합성 데이터세트를 실제 기기에서 수집하는 주파수에 맞게 샘플링 하여 사용 할 수 있도록 100Hz로 추출하였다. 센서에서 추출한 정보는 선가속도와 각속도를 추출하였다.

데이터 전처리 SmartFall과 마찬가지로 가상에서 획득한 데이터를 유사한 샘플링 비율로 수정해주었다. 33.3Hz로 수정한 데이터세트에 SmartFall과 같은 기준을 적용하기 위해서 Fall은 25개의 타임스텝의 연속으로 하였다. 낙상은 짧은 시간 내에 이루어지므로 최초의 낙상이 발생한 순간부터 총 25개의 열 데이터를 낙상으로 사용하고 이후는 사용하지 않았다. 이후의 데이터의 경우 주로 누워있는 동작으로 몸이 바닥에 닿아있어 낙상으로 레이블링 되지만, 누워있음과 특별한 차이가 없는

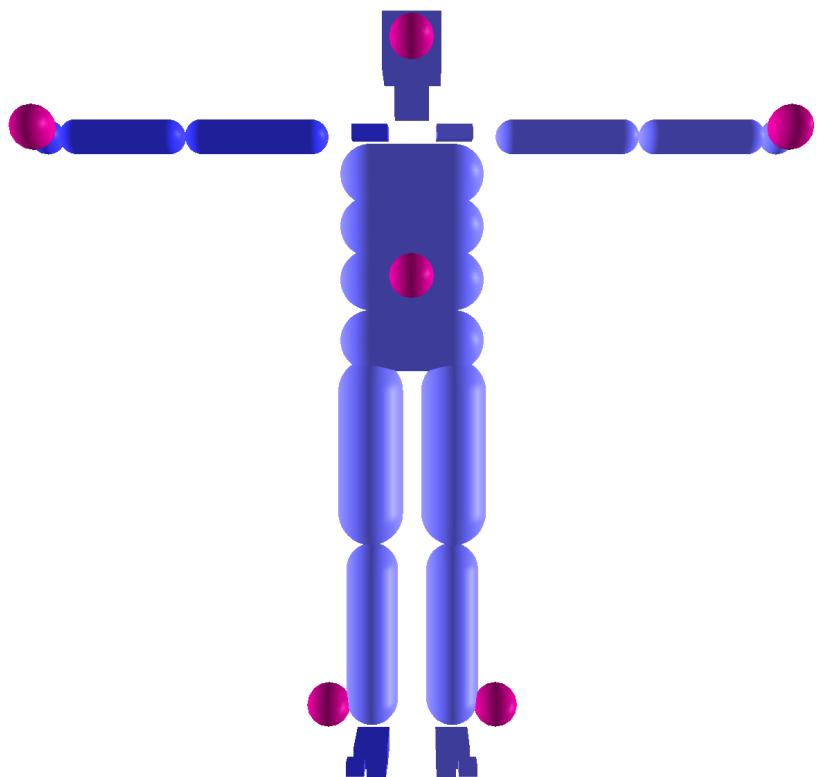


Figure 5.5: 센서 부착 위치

경우이다.

레이블링 매 타임스탬프마다 낙상과 비낙상 상태를 구분하는 레이블을 두었다. 낙상과 낙상이 아닌 것을 구분하는 것은 자동적으로 설정하였다. 실제 데이터세트의 경우 수작업을 통해서 낙상을 결정하는 경우가 많다. 이에 따라 데이터의 오류가 발생할 수 있다. 낙상 동작에서 생성된 시계열 관성 데이터의 레이블링을 측정하는데 오류는 연구자마다 다를 수 있으며, 같은 연구자도 매 시행마다 오차가 발생할 수 있다. 이에 따라 데이터의 품질이 낮을 수 있으며 후처리를 요하는 경우도 있다. 합성 데이터세트는 이러한 우려 없이 낙상에 대한 레이블링을 수행할 수 있다. 본 연구에서는 낙상의 정의에 따라 두 발이 땅에 닿은 경우에는 일반 동작으로 레이블링하고, 그 이외의 다른 부분이 닿은 경우 낙상으로 레이블링 하였다. 낙상의 정의는 “바닥이나 아래쪽의 표면으로 신체가 이동하는 것”이나 “발바닥 이외의 신체 일부가 바닥면에 접지한 경우”이므로 이를 시뮬레이터에서 구현하였다.

ADL 데이터 추가 가상 데이터세트을 이용하는 경우에는 ADL 데이터를 SmartFall 데이터세트에서 ADL에 해당하는 데이터를 추가로 추출해 사용하였다. 해당 데이터세트에서 낙상에 해당하는 데이터를 모두 제외하고 생성하였다. 가상데이터세트를 분류하는 문제를 풀 때에 위 데이터세트를 이용해 학습과 분류에 사용하였다.

5.3 낙상 감지 시스템 결과

SmartFall Training/Test SmartFall 데이터를 이용해서 LSTM을 학습하였다. SmartFall 데이터세트를 원래 학습용과 테스트용 데이터로 나누어진 그대로 사용하였다. 학습을 통해 SmartFall의 원래 검출 정확도는 92.5%로 f1-스코어는 약 0.87, recall은 약 0.79, precision은 약 0.97으로 나타났다. 이를 베이스 라인으로 사용하였다. 학습 결과는 [Figure 5.4]와 같이 나타났다. 하이퍼파라미터는 다음과 같이 사용하였다. 학습률은 0.001, hidden dimension은 256 차원, 에포크의 수는

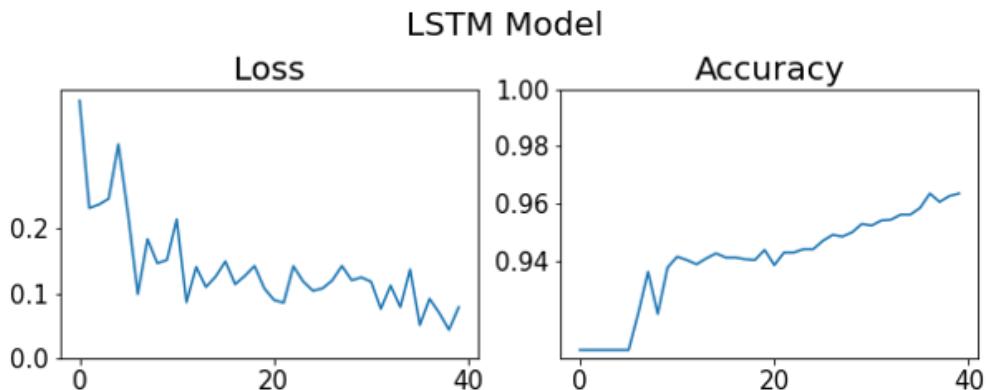


Figure 5.6: LSTM을 이용한 SmartFall 낙상/비낙상 학습 결과

	0	1
0	3144	72
1	18962	222

Table 5.1: 전이학습을 실시하지 않고 가상데이터를 분류한 Confusion Matrix

40으로 하였다.

5.4 가상 데이터세트 분류 결과

SmartFall Training/합성 데이터 Test SmartFall의 학습 데이터를 사용해 만든 모델에 합성 데이터세트를 분류하였을 경우 약 15.0%의 인식 성능을 보여 인식률이 아주 낮았다.

5.5 전이 학습 결과

전이학습을 적용하여 SmartFall로 학습된 모델을 재학습하였을 때, 96.9%의 정확도를 보였다.

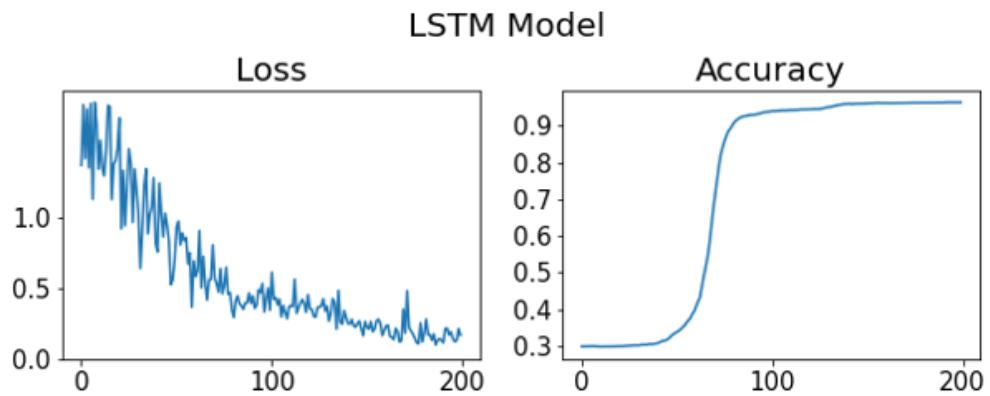


Figure 5.7: LSTM을 이용한 전이 학습 결과

	0	1
0	6298	574
1	179	17114

Table 5.2: 전이 학습을 통한 Confusion Matrix.

Dataset for Training	SmartFall Smartwatch	SmartFall Smartwatch	SmartFall Smartwatch (transfer) Virtual Dataset
Dataset for Test	SmartFall	가상 데이터세트	가상 데이터세트
Precision Score	0.96	0.67	0.96
Recall Score	0.78	0.15	0.98
F1 Score	0.86	0.06	0.97
Accuracy	0.93	0.15	0.97

Table 5.3: 전이 학습을 통한 가상데이터세트가 가장 좋은 성능을 보이고 있다.

5.6 종합 결과

이와 같이 낙상 감지 시스템을 통해 생성한 가상 데이터의 타당성을 검증해보았다. SmartFall 데이터세트를 통해 구한 정확도의 베이스라인을 92.5%라고 하였을 때 가상 데이터세트를 통해서 전이학습 하고 낙상 분류한 모델의 결과는 96.9%로 4.4%p 높다. 이를 통해 낙상 감지와 같은 문제에서 가상 데이터세트를 사용한 방법이 타당하게 쓰일 수 있다는 점을 알 수 있다. 다만 SmartFall 이외의 다른 낙상 데이터세트에 대해서도 검증이 이루어지면 더욱 더 강력한 결과를 획득할 수 있을 것이다.

Chapter 6

고찰 및 결론

본 논문에서는 가상 환경에서 관성 데이터를 생성하고 검증할 수 있는 시스템을 구축하였다. 이를 위해 가상 환경에서 다양하고 자연스러운 낙상 동작을 생성하고 이 가상의 낙상 동작에서 가속도와 속도와 같은 가상의 관성 센서 데이터를 획득하였다. 그리고 순환 신경망 모델을 이용한 낙상 감지 시스템을 구현하고 이를 실제 낙상 데이터 세트와 비교 실험하여 이 가상 데이터의 타당함을 보였다.

낙상 동작 생성을 위해서는 가상의 물리 환경에서 강화학습을 이용하여 학습을 하였다. 낙상이 발생하는 조건에서 사람이 취할 수 있는 전략을 모델링하여 낙상 동작을 학습하였다. 그 전략은 넘어질 때 충격량을 줄이고, 신체 부위 별로 충격량에 대한 가중치를 달리 주는 것이다. 이를 통해 실제 사람과 유사한 동작을 생성하였다. 충격량에 대한 조건으로 학습하고 그리고 넘어지는 동작을 포함한 모션 캡쳐 데이터로 학습한 Fall-VAE를 이용해 동작의 품질을 향상 시켰다. 그리고 걷기 등과 같은 동작을 수행하는 도중에 낙상으로 정책을 전환해서 학습과 데이터생성을 진행하는 경우에 더 다양한 동작을 생성해 낼 수 있었다. 별도의 미리 입력된 정책이 없이 학습을 하는 경우, 낙상 동작이 외력에 보다 의존적으로 생성되는 것으로 관찰되었고 이와 달리 기본적인 동작을 학습하고 수행하는 도중에

외력이 발생한 경우는 수행 중인 동작에서 낙상으로 이어지는 과정에서 다양한 동작을 생성할 수 있었다.

데이터 생성 단계에서는 학습 단계에서 존재하는 외력이나 혹은 존재 하지 않았던 외력을 캐릭터에게 가하여 낙상을 유도하였다. 이렇게 생성한 낙상 동작은, 별도의 동작 클립을 입력으로 하지 않고 생성된 동작이며, 동작에서 관성 데이터와 해당 데이터 마다의 레이블을 추출할 수 있다. 가상의 센서 위치는 임의로 지정 할 수 있으며 본 연구에서는 비교의 데이터세트에서 생성한 것과 동일한 위치의 물리값을 추출하였다.

낙상 감지 시스템은 순환 신경망 모델을 이용해 구현하였다. 연속된 정보를 손실없이 처리할 수 있는 모델로 LSTM(Long-Short Term Memory)를 사용하였다. 모델 최적화를 위해서 Cyclic Learning Rate를 파라미터로 갖는 경사 하강법(Stochastic Gradient Descent) 함수를 쓰고, 교차 엔트로피(cross entropy)를 이용해 모델의 손실을 최소화 하였다. SmartFall 데이터세트와 교차로 검증하였을 때, SmartFall 데이터로 학습하고 가상 데이터세트로 인식하였을 때는 정확도가 96.9% 이었다.

관성 데이터를 활용한 사용자 행동 인식 연구는 데이터의 확보가 어려운 것을 중요한 도전으로 한다. 현실에서 이러한 데이터를 생성하는 것은 비용이 비싸고, 수작업을 많이 요하는 작업이다. 이러한 한계가 있어, 데이터 증강, 비디오를 이용한 사용자 행동 인식 연구, 컴퓨터 비전을 이용한 데이터 생성 방법도 많이 시도되고 있으나, 각각 한계를 가지며 특히 카메라를 이용한 방식은 카메라를 설치해야하는 어려운 점 또한 존재한다. 관성 센서는 비교적 저렴하고 휴대가 간편하며 신체에 부착할 수 있다. 그리고 사생활 침해에 대한 우려가 현저히 적다. 관성 센서 데이터 수집과 레이블링이 용이해진다면, 더욱 다양한 연구와 응용이 일어날 것으로 예상된다. 이 연구가 낙상 이외의 다른 동작으로도 응용을 할 수 있다면 그러한 데이터 수집 방법에 진일보를 기록할 수 있을 것이다.

자연스러운 낙상 동작 생성은 이전에 이와 같은 방법으로 시도된 적이 없으며, 컴퓨터 그래픽스 중 캐릭터 애니메이션 분야의 강화학습을 이용한 캐릭터 동작 생

성 및 제어 방법들이 발전하면서 가능하게 되었다. 가상 물리 환경과 실제 환경을 연결하는 보다 많은 연구가 양쪽 분야를 모두 발전 시킬 수 있을 것이다. 캐릭터 애니메이션 분야에서는 자연스러운 동작에 대한 정의를 요구하는 경우가 많다. 이러한 관성 정보가 실제 데이터셋과 비교해서도 동등하거나 뛰어난 결과를 보인다면 한 가지 동작의 자연스러움에 대한 증거가 될 수 있을 것이다. 또한 사용자 행동 인식이나 낙상 감지와 같은 분야에서도 기존에 생성하기 어려웠던 데이터를 생성할 수 있는 하나의 방법으로 자리 잡을 수 있을 것이다.

본 연구의 한계로는 낙상 정의의 어려움으로 인한 것이다. 낙상의 원인을 정의함에 있어 가능한 많은 형태의 동작을 생성하였으나, 실제 낙상 동작에 미치는 요인은 근력이 약해지거나 균형을 상실하는 것 이외에 개인의 기질이나 환경적인 요인과 같이 물리 환경 시뮬레이터에서 다루기 어려운 수많은 요인들이 존재한다. 이러한 부분을 모델링하고 포함할 수 있는 가상 환경이 있다면 더욱 풍부한 데이터를 생성할 수 있을 것이다. 최근에는 입력된 관성 데이터를 바탕으로 사람의 동작과 그에 따른 지형을 생성하는 연구 등이 발표되었다. 혹은 유튜브 영상을 이용하여 3 차원의 자세를 추출하고 또 관성 데이터를 생성해내는 연구도 발표되었다. 이러한 연구들이 축적이 되고 본 연구의 방법과 응용해서 사용된다면 가상 물리 환경에서 낙상 동작이 더욱 풍부해질 수 있을 것이다.

또한 가상의 관성 센서로 생성한 낙상 데이터의 정확성에 대한 의문이 제기될 수 있다. 가상에서 생성한 값이 정확하더라도 실제 센서 데이터는 센서 기기 자체에서 나오는 값에 잡음이 있거나, 기기 자체에서 후보정이 있을 수 있다. 이와 같은 차이는 널리 알려진 것으로 가상의 관성 데이터 뿐 아니라 가상 환경, 합성 방법을 사용하는 모든 방법론에서 제기될 수 있는 문제이다. 이러한 문제에 대해서는 여러 분야에서 논의된 것과 같이 전이학습 등을 통해서 대응할 수 있으며, 높은 성공률을 보이고 있다. 이렇게 생성하는 방식은 특정한 형태의 인간 형태의 아바타에만 적용할 수 있는가 하는 의문이 제기될 수 있다. 본 연구에서는 한 가지 형태의 아바타를 이용해 다양한 동작을 생성하는 방법에 대해서 초점을 맞추고 이 데이터가 얼마나 타당한지 검증하는 방식을 이용했다. 하지만 같은 방법을 이용하

여, 아바타의 신체적인 조건을 변경할 수도 있고, 심지어는 사람이 아닌 동물이나 상상 속의 생명체의 형태도 학습이 가능하다.

동작의 다양성에 대한 의문이 제기될 수 있고 혹은 자연스러운 다양한 동작을 생성하기 위해서는 개별적으로 학습을 하고, 보상을 설정하는 등의 작업이 동작마다 요구되는지 의문이 될 수 있다. 현재로서는 자연스러운 동작을 생성하는 방법이 아직 충분히 성숙되지 않은 단계로 추후에는 훨씬 적은 노력으로 그런 동작을 생성할 수 있을 것이라 생각한다. 본 연구에서는 낙상을 주제로 사람의 동작을 생성하여 이를 실제의 데이터셋과 비교하여 검증하는 방식을 사용하였다. 낙상은 아주 다양한 형태로 발생할 수 있다. 즉 그 원인과 동작에 있어서 정형화하여 분석하기 어려운 부분이 존재한다. 하지만 정의가 가능한 부분에 대해서는 다양한 동작을 생성가능하고, 이를 물리량으로 추출해낼 수 있는 시스템을 구축하였다. 낙상 이외의 획득하기 어려운 동작을 생성하여 관련 연구를 발전 시킬 수 있을 것이다. 또한 인간 이외의 생물이나 가상의 생물에 대한 동작도 생성이 가능하다. 심층학습의 발전으로 여러 연구 분야들이 기존의 문제들을 해결하고 더 크고 차원이 높은 문제에 도전하고 있다. 많은 연구와 실험이 기반이 되었지만, 아주 많은 데이터들이 그 기반에 있다는 것은 부정할 수 없다. 사람의 동작 분야에 대해서는 아직까지 동작 데이터 획득이 어려워 이미지 처리나 생성 또는 자연어 처리와 같이 대형의 데이터 세트를 확보하지 못했다. 본 연구의 방식을 발전시켜 나가면 사람 동작의 현실세계에 적용이 가능한 가상의 데이터셋을 확보할 수 있을 것으로 보인다.

Bibliography

- [1] K. R. M. Gibson MJ, Evans JG, “Falls in later life.,” *Improving the Health of Older People: A World View.*, 1990.
- [2] M. E. Tinetti, M. Speechley, and S. F. Ginter, “Risk factors for falls among elderly persons living in the community,” *New England journal of medicine*, vol. 319, no. 26, pp. 1701–1707, 1988.
- [3] S. Usmani, A. Saboor, M. Haris, M. A. Khan, and H. Park, “Latest research trends in fall detection and prevention using machine learning: a systematic review,” *Sensors*, vol. 21, no. 15, p. 5134, 2021.
- [4] E. G. González, J. R. Villar, and E. d. l. Cal, “Time series data augmentation and dropout roles in deep learning applied to fall detection,” in *International Workshop on Soft Computing Models in Industrial and Environmental Applications*, pp. 563–570, Springer, 2020.
- [5] M. S. U. Pita, A. S. Alon, P. M. B. Melo, R. M. Hernandez, and A. I. Magboo, “Indoor human fall detection using data augmentation-assisted transfer learning in an aging population for smart homecare: A deep convolutional neural network approach,” in *2021 IEEE 19th Student Conference on Research and Development (SCoReD)*, pp. 64–69, IEEE, 2021.

- [6] H. Kwon, C. Tong, H. Haresamudram, Y. Gao, G. D. Abowd, N. D. Lane, and T. Ploetz, “Imutube: Automatic extraction of virtual on-body accelerometry from video for human activity recognition,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 4, no. 3, pp. 1–29, 2020.
- [7] Y. Zhao, Y. Gao, J. Zhai, and D. Li, “A data augmentation strategy for skeleton-based fall detection,” in *2021 China Automation Congress (CAC)*, pp. 7188–7193, IEEE, 2021.
- [8] D. Zherdev, L. Zherdeva, S. Agapov, A. Sapozhnikov, A. Nikonorov, and S. Chaplygin, “Producing synthetic dataset for human fall detection in ar/vr environments,” *Applied Sciences*, vol. 11, no. 24, p. 11938, 2021.
- [9] X. B. Peng, P. Abbeel, S. Levine, and M. van de Panne, “Deepmimic: Example-guided deep reinforcement learning of physics-based character skills,” *ACM Trans. Graph.*, vol. 37, no. 4, 2018.
- [10] Z. Yin, Z. Yang, M. Van De Panne, and K. Yin, “Discovering diverse athletic jumping strategies,” *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–17, 2021.
- [11] Y. Ye and C. K. Liu, “Synthesis of responsive motion using a dynamic model,” *Computer Graphics Forum*, vol. 29, no. 2, pp. 555–562, 2010.
- [12] O. Arıkan, D. A. Forsyth, and J. F. O’Brien, “Pushing people around,” in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’05, (New York, NY, USA), p. 59–66, Association for Computing Machinery, 2005.

- [13] L. Hoyet, R. McDonnell, and C. O’Sullivan, “Push it real: Perceiving causality in virtual interactions,” *ACM Trans. Graph.*, vol. 31, jul 2012.
- [14] V. B. Zordan, A. Majkowska, B. Chiu, and M. Fast, “Dynamic response for motion capture animation,” *ACM Trans. Graph.*, vol. 24, p. 697–701, jul 2005.
- [15] Y. Lee, K. Wampler, G. Bernstein, J. Popović, and Z. Popović, “Motion fields for interactive character locomotion,” *ACM Trans. Graph.*, vol. 29, dec 2010.
- [16] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017.
- [17] Adobe Systems Inc., “Mixamo,” 2018.
- [18] S. K. J. L. InKyung Kim, SeongSil Heo, “전이 학습을 통한 gru와 lstm의 낙상 검출 성능 향상,” *Korea Institute Of Communication Sciences(KICS)*, 2020.
- [19] T. R. Mauldin, M. E. Canby, V. Metsis, A. H. Ngu, and C. C. Rivera, “Smartfall: A smartwatch-based fall detection system using deep learning,” *Sensors*, vol. 18, no. 10, p. 3363, 2018.

Abstract

Falls can occur in any setting at any age and can have serious consequences, particularly among the elderly. As a result, the development of fall detection technology that uses various information sources, such as images or inertial sensors, has been actively researched using artificial intelligence. In recent years, with the advancement of mobile devices, many fall detection technologies that use inertial sensors have been developed. However, obtaining sufficient data on the fall environment remains a challenge for effective fall detection. This study proposes a solution to this challenge by utilizing human-shaped characters in a virtual physics simulation environment to recreate natural falling motions. Virtual inertial sensor information, including 3-axis acceleration and velocity, is also generated. Reinforcement learning is applied in the simulation environment to generate the data needed for fall detection, such as body information and the character's frictional force. By leveraging the latest character animation technology, this approach enables the rapid generation of fall-related inertial sensor data, which was previously difficult to obtain, and its application to fall detection technology.

Keywords: Computer Animation, Physics Simulation, Physics-based Control, Data-Driven Animation, Deep Learning, Reinforcement Learning, Fall Detection, IMU Sensor, Long Short-Term Memory

Student Number: 2021-21985