

# 멀티 터치 장치를 통한 캐릭터의 이동 동작 제어\*

박근태<sup>0</sup>, 이지원, 이윤상  
한양대학교 컴퓨터소프트웨어공학과

qkrmsxo01@hanyang.ac.kr, babap8514@gmail.com, yoonsanglee@hanyang.ac.kr

## Interactive Character Locomotion Control via Multi-touch Device

Geuntae Park<sup>0</sup>, Jiwon Yi, Yoonsang Lee  
Dept. of Computer Science, Hanyang University

### 요약

멀티 터치 장치상에서 둘 또는 세 손가락의 터치 정보를 이용하여 캐릭터의 이동 동작을 제어하는 방법을 제안한다. 시스템은 기본적으로 사용자가 두 손가락으로 입력한 위치 정보의 변화로부터 적절한 전진 동작을 생성하며 추가로 다른 한 손가락의 입력 정보에 따라 캐릭터의 정면 방향 회전을 결정한다.

### 1. 서론

Finger-walking은 두 손가락을 이용하여 사람의 발과 다리의 움직임을 묘사하는 방법이다. 이는 걷기, 달리기 등 다양한 종류의 움직임뿐만 아니라 보폭, 발걸음 속도 등 발동작의 스타일도 직관적으로 표현할 수 있다.

우리는 캐릭터의 이동 및 동작을 제어할 때 자주 사용되는 속도 및 목표 경로 등의 정보 대신 지면에 접촉 중인 발의 위치 정보로부터 전진 동작을 생성하는 방법을 제안한다. 먼저, 사용자는 멀티 터치 장치상에서 Finger-walking을 수행하여 매 시점 지면에 접촉 중인 발의 위치 정보를 시스템에 입력한다. 시스템에 입력된 발의 위치는 각 시점에서의 캐릭터에 상대적인 위치 변화로 해석된다. 그런 다음 시스템에서는 위치 변화 정보를 통해 사용자가 입력한 발의 상대 위치 정보를 지키면서도 이전까지의 동작과 이어지는 다음 자세를 만들어낸다. 따라서 사용자는 캐릭터의 자세한 발걸음을 직접적으로 제어 및 조작할 수 있다.

두 발의 위치 변화 정보만으로는 캐릭터의 정면 방향을 제어하기 어려우므로 이를 제어하기 위한 하나의 터치 입력을 시스템에 추가했다. 사용자는 다른 한 손가락을 수평으로 이동시켜 캐릭터의 왼쪽 및 오른쪽 회전을

제어할 수 있다.

### 2. 네트워크

제안하는 시스템의 네트워크는 크게 이동 네트워크와 포즈 네트워크로 구성되며 각각 다음 시점에서의 캐릭터의 이동과 자세를 출력한다. 자연스럽게 이어지는 동작을 만들어내기 위해 이동 및 포즈 네트워크 모두 Gated Recurrent Units (GRU)를 사용하여 과거의 정보들도 효과적으로 활용하도록 했다.

#### 2.1. 이동 네트워크

이동 네트워크는 2개의 GRU 계층으로 구성된다.

이동 네트워크는 최근  $n+1$ 개 프레임에서의 루트의 수평 위치 이력  $\mathbf{p}_{t-n:t}$ 와 최근  $n$ 개 프레임과 다음 프레임에서의 발 접촉 정보 이력  $\mathbf{f}_{t+1-n:t+1}$ 을 입력 받아 다음 프레임에서의 루트의 수평 위치  $\mathbf{p}_{n+1}$ 을 출력한다.  $\mathbf{p}_{t-n:t}$  및  $\mathbf{p}_{n+1}$ 은 모두 현재 시점의 캐릭터 좌표계 기준으로 표현되며  $\mathbf{f}_{t+1-n:t+1}$ 은 각 정보마다 대응되는 시점에서의 캐릭터 좌표계를 기준으로 표현된다.

#### 2.2. 포즈 네트워크

포즈 네트워크는 [1]과 같이 Mixture of Experts (MoE) 구조를 가지나 과거 정보를 활용하기 위해 gating 네트워크에 GRU를 적용했다. Gating 네트워크는 1개의 GRU 계층으로 구성되며 generator는 2개의 완전 연결 은닉 계층을 가진  $K$ 개 expert들로 구성된다 (실험에서는  $K=8$  사용).

Gating 네트워크는 발 접촉 정보 이력  $\mathbf{f}_{t+1-n:t+1}$ 만을 입력으로 받아 expert들에 대한 혼합 가중치를 출력한다. Generator의 매개변수는 gating 네트워크가 출력한 가중치에 따라 expert들의 매개변수들을 합한 값으로 결정되며 다음 프레임에서의 발 접촉 정보  $\mathbf{f}_{t+1}$ 과 정면 방향  $\mathbf{d}_{t+1}$  및 캐릭터 상대 이력  $\mathbf{s}_{t-n:t}$ 을 입력 받아 다음 프레임에서의 캐릭터 상대 정보  $\tilde{\mathbf{s}}_{t+1}$ 를 출력한다.  $\mathbf{d}_{t+1}$ ,  $\mathbf{s}_{t-n:t}$  및  $\tilde{\mathbf{s}}_{t+1}$ 은 현재 시점의 캐릭터 좌표계 기준으로 표현되며 나머지는 이동 네트워크와 동일하게 표현된다.

### 3. 학습

\* 구두 발표논문

\* 본 논문은 요약논문 (Extended Abstract) 으로서, 본 논문의 원본 논문은 현재 타 학술대회 (논문지)에 제출 중임.

\* 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원(RS-2023-00222776) 및 문화체육관광부 및 한국콘텐츠진흥원의 2024년도 문화체육관광 연구개발사업 지원을 (RS-2024-00399136) 받아 수행되었음.

학습 시 이동 및 포즈 네트워크가 동시에 학습된다.  
 학습에 사용된 전체 손실 함수는 다음과 같다.

$$L = w_1 \cdot L_r + w_2 \cdot L_{FK} + w_3 \cdot L_d + w_4 \cdot L_c + w_5 \cdot L_{ct} \quad (1)$$

위 식에서  $L_r$  은 네트워크의 출력과 원본 데이터 간의 평균 제곱 오차로 계산된다.

$L_{FK}$  는 forward kinematics(FK) 계층[2]을 사용하여 네트워크가 예측한 포즈 및 이동으로부터 각 관절의 다음 시점에서의 예측 위치를 구하고 실제 다음 시점 관절들의 위치와 평균 제곱 오차를 계산한 값이다.

$L_d$  은 생성된 자세의 정면 방향이 입력 받은 목표 방향을 따르도록 학습시키는 손실함수로 입력으로 들어온 목표 정면 방향과 네트워크가 예측한 포즈의 정면 방향 사이의 평균 제곱 오차를 계산한다.

앞서 설명한 손실함수들을 통해 입력에 따른 모션 생성을 학습시킬 수 있으나 생성되는 모션의 발 미끄러짐 현상이 심각한 것을 확인했다. 이에 우리는 발 미끄러짐을 방지하기 위해 추가로  $L_c$ 와  $L_{ct}$ 를 학습에 적용했다.

$L_c$ 는 다음과 같이 표현된다.

$$L_c = \|(\mathbf{f}_t^c \odot \mathbf{f}_{t+1}^c) \odot \text{MSE}_{\text{feet}}(\mathbf{f}_{t+1}^{FK}, \mathbf{f}_t^p)\|_1 \quad (2)$$

여기서  $\odot$ 은 요소별 곱을 나타내며  $\text{MSE}_{\text{feet}}$ 은 두 발 각각에 대한 2차원 수평 위치 정보를 받아 각 발에 대한 평균 제곱 오차를 출력한다.  $\mathbf{f}_{t+1}^{FK}$ 은 FK layer를 통해 얻은 현재 시점의 캐릭터 좌표계로 표현된 다음 시점에서의 예상 발 위치를 나타내며  $\mathbf{f}_t^c$ 와  $\mathbf{f}_t^p$ 는 각각 입력 받은 두 발의 접촉 상태와 위치를 나타낸다.

$L_{ct}$ 는 다음과 같다.

$$L_{ct} = \|(\mathbf{f}_t^c \odot \mathbf{f}_{t+1}^c) \odot \text{MSE}_{\text{feet}}(\hat{\mathbf{f}}_{t+1}^p, \mathbf{f}_t^p)\|_1 \quad (3)$$

여기서  $\hat{\mathbf{f}}_{t+1}^p$ 은  $\mathbf{f}_{t+1}^p$ 를 네트워크가 예측한 이동 및 포즈로부터 구한 캐릭터 좌표계를 역연산하여  $t$  프레임에서의 캐릭터 좌표계 기준으로 변환한 값이다.  $L_c$ 와  $L_{ct}$ 는 이동 네트워크 및 포즈 네트워크의 출력이 입력된 발의 상태 변화를 보다 잘 반영하도록 학습시킨다.

#### 4. 실험 결과

실험을 위해 LaFAN[3] 데이터들 중 걷기, 달리기, 점프 같은 평평한 지면에서의 동작이 담긴 데이터를 선정하여 약 3시간 네트워크를 학습했다.

실험에서 멀티 터치 모니터를 사용하여 다양한 스타일의 finger-walking들과 목표 방향 정보를 입력했으며 시스템에서 장치 상에 입력한 발의 위치 변화 및 목표 방향 입력을 반영하여 동작들을 생성하는 것을 확인했다 (그림 1 참조).

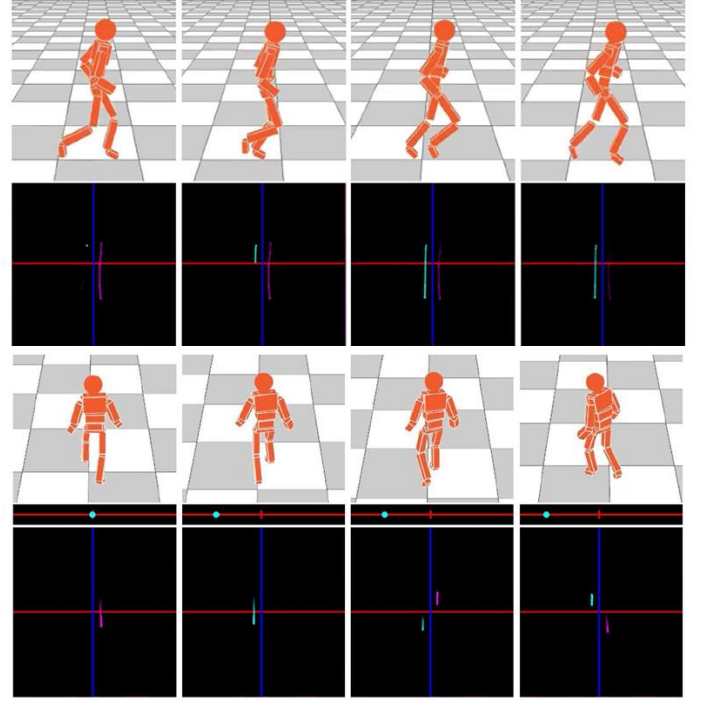


그림 1: 터치 입력에 따라 생성된 결과 모션. 접촉 해제 시에도 입력 경로의 시각화를 위해 이전 입력 위치가 점차 연한 색으로 렌더링 됨. 초록색은 왼발, 붉은 색은 오른발 터치 입력 궤적을 의미. (상단: 달리는 동작, 하단: 목표 방향 입력에 따라 회전하며 걷는 동작)

#### 5. 결론

우리는 멀티 터치 장치상에 사용자가 입력한 finger-walking으로부터 캐릭터의 전신 동작을 생성하는 방법을 제안하였다. Finger-walking으로 전신 동작을 생성하는 기존 연구[4]는 장치상의 터치 위치를 전역 공간의 좌표 값으로 지정했기 때문에 캐릭터의 활동 영역이 터치 장치의 넓이에 제한되는 반면 우리의 연구는 캐릭터 좌표계에 대한 상대적인 발 위치로 표현함으로써 작은 영역에서도 캐릭터의 이동을 제어할 수 있다.

#### 참고문헌

- [1] He Zhang, Sebastian Starke, Taku Komura, and Jun Saito. 2018. Mode-adaptive Neural Networks for Quadruped Motion Control. *ACM Trans. Graph.* 37, 4, Article 145 (July 2018), 11 pages.
- [2] Ruben Villegas, Jimei Yang, Duygu Ceylan, and Honglak Lee. 2018. Neural kinematic networks for unsupervised motion retargetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '18)*.
- [3] Félix G. Harvey, Mike Yurick, Derek Nowrouzezahrai, and Christopher Pal. 2020. Robust Motion In-Betweening. *ACM Trans. Graph.* 39, 4, Article 60 (July 2020), 12 pages.
- [4] Noah Lockwood and Karan Singh. 2012. Finger walking: motion editing with contact-based hand performance. In *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '12)*.