

# 사용자 멀티 터치 입력을 통한 캐릭터 이동 모션 생성

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

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

## Synthesizing Character Locomotion Animation Using User Multi-touch Input

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

### 요약

본 논문에서는 캐릭터 제어를 위해 일반적으로 사용되는 방식인, 사용자가 입력한 캐릭터의 이동 방향과 속도 등의 정보에 따라 캐릭터의 전신 동작을 생성하는 방법 대신, 사용자가 멀티 터치 디바이스를 통해 입력한 캐릭터의 발걸음에 따라 캐릭터의 전신 동작을 생성하는 방법을 제안한다.

### 1. 서론

일반적으로 사용자가 3차원 캐릭터의 이동을 조작하기 위해 사용하는 컨트롤러로 키보드와 마우스 조합 또는 게임 패드가 사용된다. 키보드와 마우스의 경우 주로 마우스로 캐릭터의 방향을, 키보드로 캐릭터의 전후좌우 이동을 조작하며 추가적인 키 입력을 통해 속도를 제어한다. 게임 패드의 경우 조이스틱형 장치를 기울여 캐릭터의 방향과 이동을 제어하며 장치를 기울이는 각도를 조절하여 속도를 비교적 연속적으로 제어할 수 있다. 하지만 이러한 방식들은 이동 방향과 속도 등의 정보를 받아 이에 해당하는 전신 동작을 생성하기에 사용자가 직접 캐릭터의 발걸음을 제어할 수 없다.

우리는 사용자가 멀티 터치 디바이스로 입력한 캐릭터의 발걸음에 따라 캐릭터의 전신 동작을 생성하는 방법을 제안한다. 사용자의 터치 입력은 다음 시점의 고정 발(stance foot)의 캐릭터의 루트에 대한 상대적인 수평 위치로 해석된다. 이를 통해 직관적인 방식으로 캐릭터의 발걸음을 직접 제어할 수 있어, 사용자가 캐릭터의 발동작을 보다 세부적으로 묘사하는 것이 가능해진다. 제안하는 시스템은 사용자가 입력한 발걸음과 이전 1초 동안의 동작 정보를 받아 포즈 네트워크가 다음 프레임에서의 캐릭터의 자세를, 변환 네트워크가 캐릭터의 이동을 예측한다.

### 2. 네트워크 구조

제안하는 네트워크는 과거 및 현재 캐릭터 상태 정보와 사용자 입력 정보를 받아 다음 시점의 자세와 이동을 예측한다.

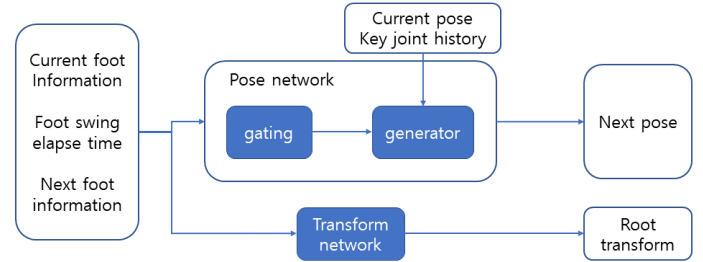


그림 1: 네트워크 구조

네트워크는 현재 및 다음 시점의 발 정보  $F_t, F_{t+1}$  과 발이 뜬 상태로 경과한 시간  $f_t^e$  및 현재 캐릭터 포즈 정보  $P_t$ 를 입력으로 받는다. 시점  $t$ 에서의 발 정보  $F_t$ 는 각 발의 지면 접촉 여부  $f_t^c$ , 위치 및 속도  $f_t^v, f_t^p$ 로 구성된다.  $f_t^e$ 는 각 발에 대해 발이 떠 있는 때 프레임마다 1씩 증가하고 해당 발이 지면에 접촉하면 0으로 초기화된다.  $P_t$ 는  $t$ 시점에서 각 관절의 회전  $j_t^r$ 와 일부 주요 관절의 과거 1초 동안의 위치 및 속도를  $n$ 개 frame에 걸쳐 표현한  $K_{t-n:t}^p, K_{t-n:t}^v$ 로 구성된다. 위의 입력을 받아 네트워크는 각 관절의 회전  $j_{(t),t+1}^r$ , 주요 관절의 속도  $K_{t+1}^v$  및 루트의 상대적 변환  $T_{t+1}$ 을 출력한다.  $T_{t+1}$ 은 다음 시점에서의 루트 위치와 회전으로 구성된다. 네트워크의 입력과 출력은 캐릭터 좌표계에 대해 표현되며 이는 왼쪽 고관절에서 오른쪽 고관절을 지나는 벡터를 지면에 투영하여 x축으로, 전역 좌표계 위쪽 방향을 y축으로, 이 둘의 외적을 z축으로 가지며, 캐릭터의 루트(골반)의 위치를 지면에 투영한 지점을 원점으로 하는 좌표계이다.

네트워크는 역할에 따라 다음 시점의 포즈를 생성하는 포즈 네트워크와 다음 시점의 캐릭터 위치를 예측하는 변환 네트워크로 구성된다. 포즈 네트워크는 보다 안정적인 동작을 생성할 수 있도록 mixture-of-expert

\* 구두 발표논문, 요약논문 (Extended Abstract)  
\* 본 논문은 요약논문 (Extended Abstract) 으로서, 본 논문의 원본 논문은 현재 타 학술대회 (논문지)에 제출 준비중임.  
\* 본 연구는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행되었음 (NRF-2019R1C1C1006778, RS-2023-00222776).

구조 [Zhang et al. 2018]를 사용했다. Gating 네트워크는  $F_t$ 와 사용자가 입력한 발 정보  $F_{t+1}$ 을 입력 받아 generator 네트워크의 파라미터를 결정하며 generator 네트워크는  $F_t$ ,  $F_{t+1}$  및  $P_t$ 를 입력 받아  $j_{\{t,t+1}^r$  및  $K_{t+1}^p$ 을 출력한다. 변환 네트워크는 다음 시점의 캐릭터 루트의 위치와 회전을 예측하는 완전연결 계층 구조를 가지며 gating 네트워크와 동일한 입력을 받아  $T_{t+1}$ 을 출력한다.

### 3. 네트워크 학습

학습 과정에는 FK(forward kinematics) 계층 [Villegas et al. 2018]을 사용하여 네트워크가 예측한 루트의 이동과 각 관절의 회전을 다음 시점의 각 관절의 위치로 변환하여 손실 함수 계산에 사용했다.

학습에 사용된 전체 식은 다음과 같다.

$$r = L_{gen} + L_{fk} + L_{contact} + L_{next\_frame} \quad (4)$$

위의 식에서  $L_{gen}$ 은 네트워크 출력과 원본 모션에서 구한  $K_t^v$ ,  $j_t^r$ 의 평균 제곱 오차이며  $L_{fk}$ 는 FK 계층을 통해 구한 다음 시점에서의 관절들의 예측 위치와 실제 다음 시점 관절들의 위치 사이의 평균 제곱 오차이다.

$L_{contact}$ 는 다음과 같이 표현된다.

$$L_{contact} = (c_t * c_{t+1}) * MSE(f_{fk}^p, f_t^p) \quad (4)$$

여기서  $c_t$ 는  $t$ 시점에서의 접촉 유무를 표현하며 접촉 시 1, 반대의 경우 0으로 표현된다.  $f_{fk}^p$ 는 FK 계층을 통해 구한 발의 위치이다.

$L_{next\_frame}$ 는 다음과 같이 표현된다.

$$L_{next\_frame} = (c_t * c_{t+1}) * MSE(f_t^p, f_{\{t,t+1}^p) \quad (4)$$

여기서  $f_{\{t,t+1}^p$ 는  $t$  시점의 캐릭터 좌표계로 표현된  $t+1$  시점의 발의 위치로 FK 계층으로 계산된 관절 위치들을 통해 구한  $t$  시점 기준  $t+1$  시점의 캐릭터 좌표계 행렬로 네트워크 입력으로 들어오는  $f_{t+1}^p$ 을 변환하여 얻어진다.

### 3. 실험 결과

네트워크는 LaFAN [Harvey et al. 2020] 데이터들 중 걷기 및 달리기 동작을 선정하여 약 3시간 학습되었다.

실험에서 멀티 터치 모니터를 통해 사용자의 입력을 받았다. 사용자가 손가락을 끌어 시간에 따른 발의 지면 접촉 위치를 표현하면 시스템은 실시간으로 입력된 발의 상대 위치에 맞는 동작을 생성한다.

실험 결과 입력된 발의 위치를 최대한 만족하는 동작이 생성되는 것을 확인했으며 임의의 보폭에 맞춰 캐릭

터의 이동이 표현되는 것을 확인할 수 있었다.

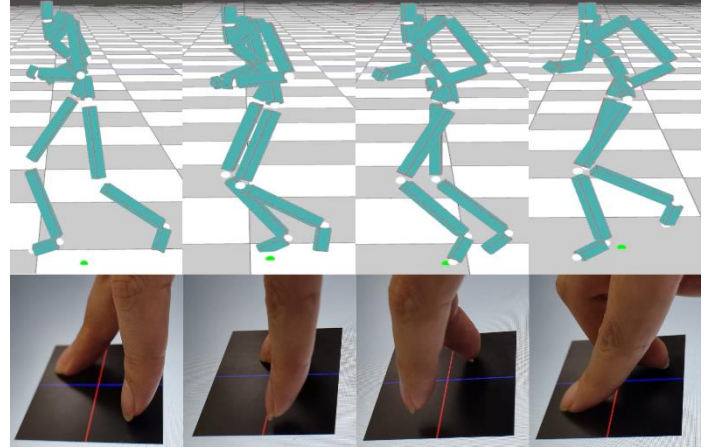


그림 2: 사용자 입력에 따른 모션 생성 예 (위: 생성 모션, 아래: 사용자 입력)

### 4. 결론

우리는 사용자의 터치 입력으로 발걸음 정보를 받아 캐릭터의 진신 동작을 생성하는 방법을 제안하였다. 터치 입력을 발걸음으로 해석하는 기존 연구로 [Lockwood and Singh. 2012]가 있으나, 해당 연구는 전역 공간에서의 발자국 위치를 입력 받으므로 큰 터치 영역을 필요로 하는 반면, 우리의 연구는 루트에 대한 상대적인 발의 위치를 입력 받으므로 모바일 디바이스 같은 작은 화면에서도 적용 가능하다는 차이점이 있다. 현재 시스템에서는 캐릭터가 바라보는 방향이나 루트의 높이 조절이 잘 되지 않는데, 이는 발의 상대적인 움직임만으로는 정보가 부족한 것이 원인으로 생각되며 추가적인 정보를 입력 받거나 생성 모델을 통해 상황에 알맞은 보충 정보를 생성하여 해결할 수 있을 것이다.

### 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. <https://doi.org/10.1145/3197517.3201366>
- [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). IEEE Computer Society, Washington, DC, USA, 8639–8648.
- [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. <https://doi.org/10.1145/3386569.3392480>
- [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). Eurographics Association, Goslar, DEU, 43–52.