한국컴퓨터그래픽스학회
Korea Computer Graphics Society Vol. 27, No. 1, P. 1~8

주 관절 경로의 변형을 통한 걷기 동작 수정

김미진° 이석원*

한국전자기술연구원

{mj_kim1023, sukwonlee}@keti.re.kr

Deforming the Walking Motion with Geometrical Editing

Meejin Kim° Sukwon Lee*

Korea Electronics Technology Institute

요 약

본 논문에서는 캐릭터의 걷기 동작 데이터를 변형하는 방법을 제안한다. 이를 위하여 주 관절(root joint)의 이동 경로를 그래프로 분석하고 라플라스 연산자를 이용해 변형하는 방법을 사용한다. 주 관절의 경로는 동작 데이터의 각 프레임별 위치와 방향을 정점으로 하고 이를 인접 프레임의 정점과 연결한 그래프 형태로 나타낸다. 주 관절 경로를 라플라스 연산자를 이용하여 좌표계를 변환하고 이를 목표 위치 및 방향에 맞도록 반복적인 방법으로 해를 구하여 변형한다. 이 방법을 이용하여 동작 데이터의 걷기 스타일을 유지하면서 다양한 경로의 걷기 동작을 얻을 수 있게 되며 많은 비용이 드는 동작 데이터 취득을 최소화할 수 있다. 최종 모션은 변형된 주 관절 경로를 기준으로 기존 모션의 다른 관절을 위치시키고 후처리하여 생성한다. 본 논문에서 제안한 방법을 응용함으로써 적은 모션 데이터로도 복잡한 환경에서 캐릭터의 걷는 동작을 생성하는 것을 보인다.

Abstract

This paper proposes a simple deformation method for editing the trajectory of a walking motion with preserving its style. To this end, our method analyzes the trajectory of the root joint into the graph and deforms it by applying the graph Laplace operator. The trajectory of the root joint is presented as a graph with a vertex defined the position and direction at each time frame on the motion dataThe graph transforms the trajectory into the differential coordinate, and if the constraints are set on the trajectory vertex, the solver iterative approaches to the solution. By modifying the root trajectory, we can continuously vary the walking motion, which reduces the cost of capturing a whole motion that is required. After computes the root trajectory, other joints are copied on the root and post-processed as a final motion. At the end of our paper, we show the application that the character continuously walks in a complex environment while satisfying user constraints.

키워드: 캐릭터 애니메이션, 모션캡처, 모션 변형, 그래프 라플라시안
Keywords: character animation, motion capture data, motion editing, graph Laplacian

1. 서론

모션 데이터는 캐릭터의 자연스런 동작을 만들기 위해 필수적인 요소이다. 하지만 모션 데이터를 취득하는 데에는 높은 비용이 든다. 이에 따라 학계에서는 기존 모션 데이터를 분석하여 새로운 모션 데이터를 생성하는 방법으로 이 비용을 줄이고자 하였다. 모션 데이터를 확장하기 위한 방법으로는 여러 데이터 중 유사한 자세를 분석하여 이 관계로부터 새로운 동작을 생성하거나[1, 2] 동작에 대한 수학적 가정을 통하여 새로운 동작으로 변환[3, 4]하는 방법이 있다. 본 논문에서는 후자의 방법으로, 이미 얻어진 모션 데이터를 이용하여 새 데이터를 생성한다. 특히 주관절 경로를 그래프로 해석하고 이를 라플라스 연산자를 이용한 변환으로 스타일을 유지한채 경로를 변형한다. 최종 모션은 변형된 주 관절 경로에 각 프레임의 자세를 합성하고 후처리하여 생성한다. 2장에서는 모션 변형을 위한 기존 연구들과 기하학적 변형을 위

*corresponding author: Sukwon Lee/Korea Electronics Technology Institute(sukwonlee@keti.re.kr)

Received : 2020.10.27./ Review completed : 1st 2020.11.19. / Accepted : 2020.12.30.
DOI : 10.15701/kcgs.2021.27.1.1
ISSN : 1975-7883(Print)/2383-529X(Online)

한 방법들을 소개하고, 3장에서는 제안하는 방법의 정의와 과정을 자세히 소개하고, 4장에서는 모션 그래프 방법을 이용한 확장 응용을 제안한다. 마지막 5장에서는 결론과 논의점을 이야기할 것이다.

2. 관련 연구

2.1 동작 변형

하나의 모션 데이터를 이용하여 모션을 변형하는 방법은 프레임 간의 관계를 이용한다. [5] 는 캐릭터의 계층적인 구조를 이용하여 모션을 변형하였고, [3, 6] 은 모션을 주변 환경이나 다른 캐릭터와의 관계를 이용하여 변형하였다. 또 [7] 은 물리 모델을 도입하여 모션을 변형하였다. 여러 모션 데이터의 각 프레임 사이의 관계를 이용하여 변형 및 합성 방법으로 새 모션으로 만드는 방법이 있다. [1, 8] 은 모션 데이터의 자세간 차이를 비교하여 유사한 자세를 간선으로 정의한 그래프로 해석하였다. 캐릭터의 상태는 자세와 위치로 정의되며 모션을 생성하는 것은 하나의 상태에서 다른 상태으로 이동하는 것에 대응된다. 그러므로 원하는 상태에 도달하는 모션을 생성하기 위해서는 시작 상태에서 그래프 탐색을 수행하여 정점 리스트를 구하여야 한다. 이 방식은 그래프의 연결성에 따라 성능이 결정되는데, 각 정점과 연결되는 간선의 수가 적으면 목표 상태로의 탐색이 어렵다는 것과 간선의 수가 많으면 탐색 시간이 느려진다는 점이 서로 상충된다. 이 점을 해결하기 위해 유사한 간선 사이에 연속적인 새 간선을 보간하는 연구 [9, 10] 가 진행되었다. 본 연구는 기존의 간선으로 표현되는 모션을 변형하여 새 간선을 만드는 방법이다. 학습을 통해 입력에 따라 연속된 자세를 생성하는 방법 [11, 2, 12, 13, 14] 들도 연구되었다. 이들은 비선형 공간으로 모션 데이터들을 변환하고 이 공간에서 자세 간 관계를 추출하였다. 원하는 목표 동작을 만들기 위해 비 선형 공간에서의 이동을 학습하여 새 동작을 생성하였다. 또한 한정된 크기의 동작 데이터에서 목표 상태까지의 이동을 보장하기 위해서는 추가 학습을 필요로 하였다. 본 논문이 제안하는 방법은 추가적인 동작 데이터나 학습 없이, 걷는 동작에 한해 원하는 목표 상태를 달성할 수 있다. [15]에서는 환경에 따른 3차원 경로를 수정하여 모션을 보간하는 연구를 진행하였다. 위 연구에서 주 관절의 경로를 변형하기 위해 3차원의 전역 좌표계를 사용하였는데, 이와 다르게 이 연구에서는 전 프레임으로 정의되는 지역 좌표계를 사용함으로써 경로의 회전에도 직관적인 스타일을 유지할 수 있도록 하였다.

2.2 기하학적 변형

[3, 16] 에서는 모션 변형을 위하여 캐릭터 상태의 각 조인트의 위치와 환경과의 위치를 연결한 메쉬로 표현하였고, 이를 그래프 라플라시안으로 변형하였다. 본 논문 또한 모션 변형을 위하여 그래프 라플라시안 연산자를 이용하며 [17] 에서 제안하는 메쉬 변형법을 이용하고 있다. 이 연구는 메쉬를 그래프의 형태로 나타내고 여기에 행렬을 곱하여 그래프 정점을 주변 정점과의 관계로 표현하는 공간(Differential coordinate)으로 변환한다. 이 변환된 좌표를 다시 3차원 공간(Cartesian coordinates)으로 변환하기 위해서는 제약 조건이 필요하며 이를 최소자승법(least square method)으로 복원하게 된다. 여기에서 구해진 해는 제약조건에 따라 달라지는데 주변 정점과의 관계를 최대한 유지한 채로 제약조건을 만족하게 되므로 세밀한 정보는 유지하고 전체적인 정보를 변형하는 결과를 얻게 된다. 또한 이 연구에서는 정점의 위치만 고려하지 않고 법선 벡터를 함께 고려하므로 곡선에 대한 변형도 고려할 수 있게 되었다. 본 논문은 이러한 변형 방법을 주관절 이동경로에 적용하였다. 각 프레임의 주관절 위치를 정점으로 하고 주변 프레임의 위치를 이어주는 간선을 가진 그래프로 표현하였고 이를 같은 라플라스 행렬로 공간 변환을 수행하였다. 또한 주관절의 방향(캐릭터가 향하는 곳)을 함께 고려하여 동작 변형 시 목표 캐릭터 방향도 지정할 수 있게 하였다. 목표 방향에 따라 중간 이동 경로 또한 함께 변형함을 확인하였다.

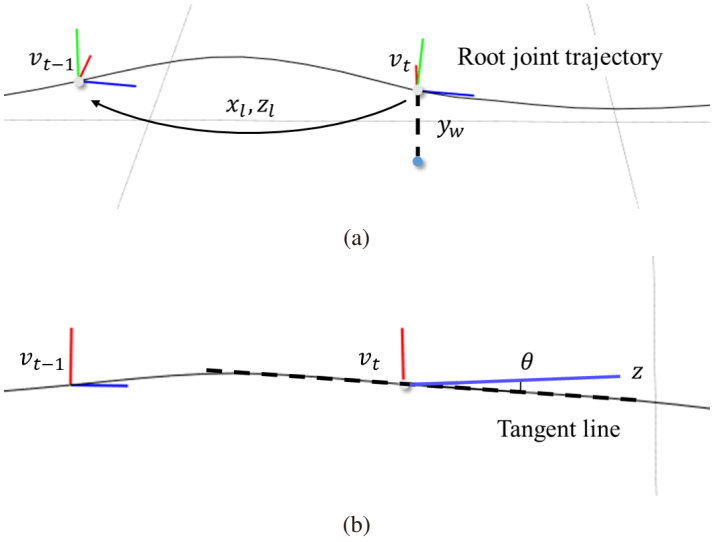


Figure 1: The state of root joint trajectory. y_w represents a height from the ground and x_l, z_l are coordinates seen from the root frame at the previous frame. To deal with the direction of the trajectory, θ is the difference between the tangent of the trajectory and z. (a) and (b) shows the same points at the different viewpoints for the explanation.

3. 주 관절 경로 변형

3.1 주 관절의 상태 벡터 및 선형 변환

이 장에서는 제안하는 방법을 설명하기 위해 먼저 주관절 경로를 정의한다.

$$\chi_t = \{x_w, y_w, z_w\} \quad (t = 0 \cdots T) \tag{1}$$

χ_t 는 프레임 t 의 경로의 전역 좌표계 w의 좌표값들로 정의된다. 모션 M에는 각 χ_t를 기준으로 캐릭터 주 관절의 자세가 기록되어 있다. 이 자세들은 y 축이 캐릭터의 위쪽 방향(중력 반대 방향)으로, z 축은 캐릭터의 앞 방향으로 정렬되어 있다. 여기에 우리는 각 χ_t마다 상태 벡터 v_t 를 정의하였다 (Figure 1).

$$v_t = \{y_w, x_l, z_l, \theta\} \tag{2}$$

상태 벡터에서 y_w 는 전역 좌표계의 값이고, x_l과 z_l은 지역좌표계에서 보는 x, z 값이며, θ 는 주 관절 경로의 접선벡터와 주 관절의 z 벡터사이의 각도이다. 이 각도는 관성좌표계의 y 축의 값으로만 표현한다. 모든 프레임의 상태 벡터는 행렬 V로 표현 가능하다.

$$V = \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_T \end{pmatrix} \tag{3}$$

v_t 의 지역좌표계는 $t-1$ 프레임의 주 관절 자세와 같다. 우리는 상태 벡터로부터 그래프 G를 정의할 수 있다. 각 정점은 상태 벡터 v 이며 간선은 주변 프레임의 상태 벡터를 연결한다.

$$G = (v_{1:T} \mid \{v_t, \mathbf{neighbor}(v_t)\}_{t=1:T}) \tag{4}$$

$$\mathbf{neighbor}(v_t) = \{v_{t+1}, v_{t-1}\} \tag{5}$$

우리는 정의된 그래프로부터 도출할 수 있는 변이 값(differential coordinates, δ)을 정의하였다.

$$\delta = v_t - \frac{v_{t+1} + v_{t-1}}{2} \tag{6}$$

이 값은 한 정점 v 를 그의 주변 프레임 $\mathbf{neighbor}(v)$으로 기술하는 것이며, 또한 주변 정점과의 관계를 나타내어 지역적 모양 정보를 포함한다. 예를 들면, v가 메쉬(mesh) 위의 정점을 나타낸다면 δ는 v의 주변 정점들의 중점과 v의 상대 위치로 표현된다. 이는 δ 가 v 의 곡면 정보(법선 벡터 및 곡률)를 추정한다고 볼 수 있다. V 로부터 δ를 구하기 위해서 다음과 같은 선형변환 L (그래프 라플라시안 연산자)을 정의할 수 있다.

$$L = I - 0.5D \tag{7}$$

I 는 단위행렬이고 D는 띠행렬(band matrix) 로 대각 성분의 위아래의 값이 1로 채워져 있다.

$$D_{ij} = \begin{cases} 1, & \text{if } j = i \pm 1 \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

구해진 L을 이용한다면

$$\begin{aligned} \delta_{1:T} &= LV \\ &= \begin{pmatrix} \delta_0 \\ \vdots \\ \delta_T \end{pmatrix} \end{aligned} \tag{9}$$

로 표현할 수 있다. 여기에서 L의 역행렬을 이용하면 δ로 부터 상태 벡터 v를 다시 얻을 수 있지만 L의 계수(rank)가 부족하므로 이를 위해선 추가적인 항이 필요하다. 우리는 이 부분에 목표하는 제약 조건c을 추가함으로써 계수를 증가시키고, 원하는 형태로 변형된 상태 벡터 v'을 구할 수 있다. 만약 추가되는 제약 조건이 1개 이상이라면 이는 해가 없는 상태 (over-determined) 가 되는데 이를 해결하기 위해 최소자승법(weighted least-square method)을 이용하여 V를 구한다. c 를 추가하여 v'를 얻는 방법은 다음 항에서 설명한다.

3.2 목표 상태에 따른 경로의 변형

여기에서는 제약 조건을 이용하여 주관절 경로를 구하는 것을 보여준다. 여기에서 제약 조건은 주관절의 상태 v 위에 설정된다.

$$c = \{y_w, x_l, z_l, \theta\} \tag{10}$$

제약 조건을 고려한 해를 구하기 위해 행렬 A 를 정의한다.

$$A = \begin{pmatrix} L \\ S_0 \\ S_T \end{pmatrix} \tag{11}$$

$S_t \in \mathbb{R}^{1\times T}$ 는 선택 행렬로 t 열에 1 이고 이외의 열에는 모두 0을 가진다. 여기에서 우리는 0, T 번째 프레임에 목표를 설정할 것이다.

$$s_j = \begin{cases} 1, & \text{if } j = t \\ 0, & \text{otherwise} \end{cases} \tag{12}$$

L과 함께 S는 연립 선형 방정식을 이룬다. 위의 행렬 A는 제약 조건을 0과 T 프레임에 설정하였다. 이를 이용하여 해 v^* 를 구하려면

$$v^* = A^\dagger b(\delta, c) \tag{13}$$

의 식을 이용한다. $A^\dagger$는 A행렬의 의사역행렬(pseudo inverse)이다. $b(\delta, c)$ 는 복원하기 위한 원 모션의 δ 값과 A에서 설정한 0, T 번째 프레임의 목표 상태 c_0, c_T 를 쌓은 형태의 행렬이다.

$$b(\delta, c) = \begin{pmatrix} \delta_l \\ c_0 \\ c_T \end{pmatrix} \tag{14}$$

위와 같이 목표하는 경로 v^*를 구할 수 있다. 여기에서 상태벡

터의 x_l, z_l 는 지역좌표계로 정의되어 있다. 지역좌표계는 이전 프레임으로 정의되어 있으므로 x_l, z_l 의 값 또한 이전 프레임에 종속된 함수로 생각할 수 있다. 이로인해, 직관적으로 역행렬을 이용하여 해를 구하는 것으로 정확한 해를 구할 수 없게 된다. 이를 해결하기 위해 우리는 지역적인 범위에서는 선형이라는 조건을 추가하여 반복적인 방법(iterative method)로 비선형의 해를 구하였다. 우선 목표 경로 χ^*에 근접하기 위해 제약조건 c를 구하여야 한다. 이전 상태 벡터를 v^-라고 정의하고 이와 대응되는 경로는 χ^-라 하자. 이 경로를 구하기 위해서는 지역좌표계로 정의된 x_l, z_l를 전역 좌표계 x_w, z_w로 변환해야 한다. χ^-와 χ^*로부터 c를 구하는데, 비선형성을 고려하여 χ^-의 목표를 최대 α 값 만큼만 차이 나도록한다.

$$c = \mathbf{max}\left(|\chi^* - \chi^-|, \alpha\right)(\chi^* - \chi^-) + \chi^- \tag{15}$$

위에서 구해진 c를 식 13에 대입하여 새로운 해 v'를 구한다. 이 과정을 v'가 목표에 도달할 때까지 반복한다.

3.3 경로로 부터 모션 생성

변형된 주 관절 경로에 적합한 모션을 생성하기 위해서 우선 모션 데이터의 환경과의 접촉점 정보를 추출한다 [18]. 이는 차후 모션 변경시 접촉점과 환경과의 접촉시 밀림 현상을 제거하기 위한 정보로 사용된다. 모션 생성을 위해 우선 모션 데이터의 주관절 경로를 구해진 해(V^*)로 대체한다. 그리고 접촉점 최적화 기법[16] 을 이용하여 생성된 각 관절의 경로를 부드럽게 유지하면서 접촉점을 유지하는 모션을 생성한다. 위의 과정을 다음의 알고리즘으로 정리하였다.

Algorithm 1 Deforming the root joint trajectory with start and end constraints

M : initial motion data.
χ^* : the target state.
α : maximum distance threshold

function SOLVE(M, χ^*)
  $\chi \leftarrow$ GETTRAJECTORY(M)
  $V \leftarrow$ GETSTATE(X)
  $L \leftarrow$ MAKELAPLACIAN(M) ▷ eq. (7)
  $S \leftarrow$ DETECTCONTACT(M) ▷ [18]

  $\delta^* \leftarrow LV$ ▷ the style of an input motion.
  $V' \leftarrow V$ ▷ initialization

  while |DISTANCE(V', χ^*)| $\leq \epsilon$ **do**
    $\chi' \leftarrow$ RECONSTRUCTTRAJECOTRY(V')
    $C' \leftarrow$ GETCONSTRAINTS(χ', χ^*) ▷ eq. (15)
    $V' \leftarrow A^{\dagger}b(\delta^*, C')$
  end while

  $M' \leftarrow$ RECONSTRUCTMOTION(X', S) ▷ [16]
  return V'
end function

4. 실험

이번 절에서는 제안된 방법을 이용하여 걷기동작을 원하는 위치 및 방향으로 전환하는 결과를 보이고 이를 모션 그래프(Motion graph)와 결합하여 기존에는 찾기 힘들었던 복잡한 환경에서도 걷기 동작을 만들어 낼 수 있음을 보인다.

4.1 모션 변형

변형에 사용한 동작 데이터는 직선으로 걷기와 옆으로 걷기 동작을 이용하여 실험을 진행하였다. Figure 2 는 직선으로 걷는 동작이며 데이터의 끝 (프레임 T)에 변형 조건을 설정하였다. Figure 2(a)는 도착 위치는 같지만 방향은 90도 회전을 하였다. 주관절 경로의 스타일을 유지하면서 마지막 회전 조건을 만족하기 위해 중간 경로가 회전 반대 방향으로 휘어져 있는 것을 확인할 수 있다. 만약 경로가 후반의 적은 프레임만 조건을 만족시키기 위해 회전한다면 δ에 의해 발생되는 오차값이 커진다. 우리는 자승법에 의해 오차를 줄이므로 오차값이 경로 전반으로 퍼져나가게 되어 위와 같은 형태로 최종 수렴한다. Figure 2(b)에서는 회전 방향을 고정한 채로 위치를 옮겼을 때의 결과를 보여준다. 또는 회전과 이동 조건을 동시에 만족하는 경로를 구할 수도 있다. Figure 3 는 옆으로 걷는 동작의 변형을 보여준다. 직선 걷기와는 다르게 주관절의 방향이 걷는 방향과 서로 수직에 가깝다. 우리는 주관절 상태 벡터 v를 정의하면서 경로와 주관절 방향의 차이 θ를 정의했으므로 이러한 동작도 변형가능하다. Figure 3(a) 의 경우에는 최종 프레임의 주관절 위치만 이동시킨 결과이다. 주관절 방향은 유지한채 경로만 변형된 것을 확인할 수 있다. 또한 관절 경로의 형태가 변형후에도 비슷한 것을 보이는데 δ 값으로 스타일이 유지되었다는 것을 알 수 있다. Figure 3(b) 는 위치와 방향이 동시에 변형된 결과를 보여준다. 좀더 길어지고 방향도 틀어졌는데, 이럴 경우 발이 바닥에 쓸리는 현상(Foot skate)이 발생할 가능성이 높다. 하지만 후처리에서 접촉점을 유지하기 위해 접촉점 최적화 기법[16]을 이용하므로 이런 이상현상을 최소화하였다.

실험에는 Intel i5-9500F CPU 가 장착된 컴퓨터를 사용하였으며, 각 실험에 데한 데이터는 Table 1 에 정리하였다. 종료 조건인 ϵ은 0.01로 설정하였다. 위치에 대한 조건만 있을 경우 1번에 만족하게 되지만 방향에 대한 조건이 추가되면 여러번의 반복이 필요한 것을 볼 수 있다.

4.1.1 강건성에 대한 실험

이 절에서는 제시하는 변형 방법의 강건성을 실험하기 위해 목표 지점을 기존 도달 지점과 180도 회전하여 도달하도록 하였다. 또한 올바른 결과를 얻기 위하여 도달 위치는 시작점에서 좌측으로 이동하였다. 실험을 통해 Figure 4(a)에서 강조된 영역과 같이 주관절 경로가 급격히 변형되는 모습을 볼 수 있다. 이는 $|\delta|$ 의 값과

experiment	# of iterations	# of frames	time (ms)
straight (pos)	1	100	3.49
straight (pos, dir)	6	100	7.87
side (pos)	1	450	12.10
side (pos, dir)	3	450	18.13

Table 1: The summary of experiments. (pos) and (dir) stands for position and direction, respectably.

연관되어 있으므로 Figure 4(a)의 경로에 $|\delta|$ 값을 막대 그래프로 표현하여 $|\delta|$ 값이 급격히 변화하는 부분을 붉은 색 막대 그래프로 구분하였다. 이에 따라 주 관절 경로가 급격히 변화하는 부분과 $|\delta|$ 값이 급격히 변화하는 부분이 유사하게 나타난다는 것을 알 수 있다. 따라서 δ 값이 급격히 증가하는 것을 방지하기 위하여 B-스플라인 보간법 [19]을 사용하여 δ의 값을 경로 전체에 걸쳐서 완화하였다. 이러한 결과 기존에 기대했던 Figure 4(b)가 됨을 확인하였다.

또한 동작을 변형하기 위한 Algorithm 1 단계별 소요 시간을 측정하였다. 실험에 사용된 동작은 총 316프레임으로 약 2.6초의 데이터를 담고 있다. 이 실험에서 해를 구하기 위해서 14라인의 반복문을 3번 수행하였다. Algorithm 1 의 6번 줄부터 9번 줄은 초기화 부분으로 약 4.5 밀리초가 소요 되었다. 반복문에서 15번 줄은 경로를 복구하는 것으로 약 3.1 밀리초가 소요되었으며 16번 줄은 제약 조건을 설정하는 것으로 약 0.1밀리초 소요되었다. 17번 줄은 희소행렬의 역행렬을 구하여 해를 구하는 것으로 약 0.8 밀리초가 소요되었다. 이는 초기화를 제외하고 많은 시간을 경로 재 구축하는 것에 쓰임을 알 수 있다.

4.2 응용: Motion Graph

본 논문에서 제안한 동작 변형 방법을 모션 그래프 탐색에 적용하여 복잡한 환경에서도 목표 동작을 만족하는 경로를 구할 수 있게 한다. 2장에서 언급했듯, 모션 그래프는 다른 상태로 전이할 수 있는 간선이 충분해야 목표에 도달하는 경로를 구할 수 있지만, 반대로 많은 간선을 탐색하기 위해선 탐색시간이 늘어나므로 해결 방법이 필요하게 된다. 제안한 모션 변형을 이용하면 적은 모션으로도 넓은 범위의 모션을 가진 것 같은 효과가 있으므로 간선을 늘리지 않아도 해를 찾을 수 있게 해준다. 이는 특히 해를 찾기 어려운 환경 (예, 복잡하고 좁은 통로)에서도 경로를 찾을 수 있게 해준다.

4.2.1 모션 데이터

실험에 사용된 동작 데이터는 약 90초 정도의 걷기 행동이 담겨 있으며 (Figure 5(b)) 캐릭터는 65개의 관절과 총 162개의 관절 자유도(Degrees of Freedom)를 가지고 있다. 이는 손가락 관절과 같은 세밀한 움직임을 나타내는 모든 관절을 포함한 것이다. Figure 5(b) 과 같이 동작 데이터는 일정 넓이의 공간을 무작위로 걷는

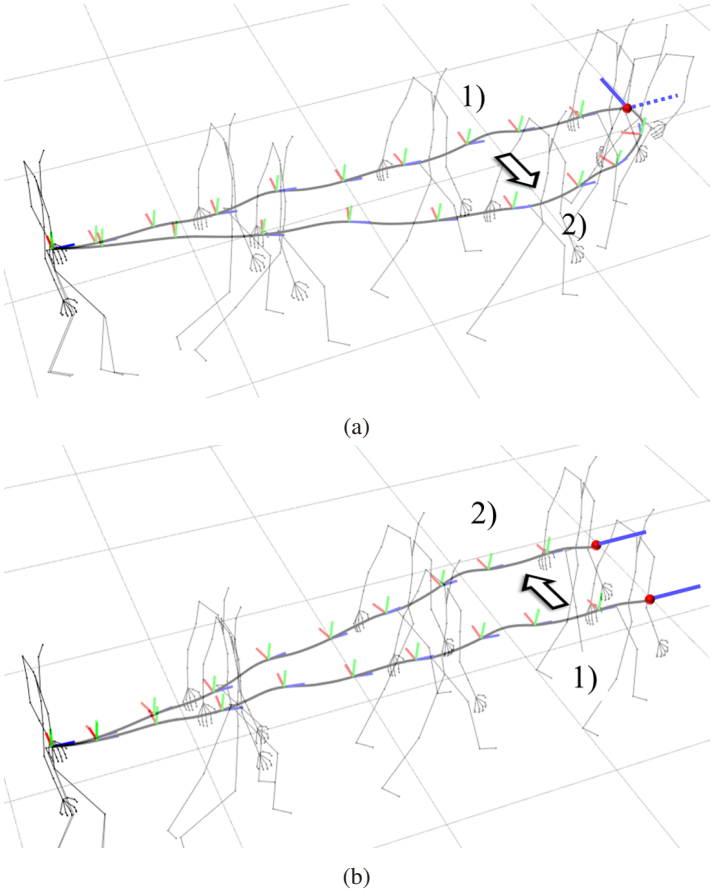
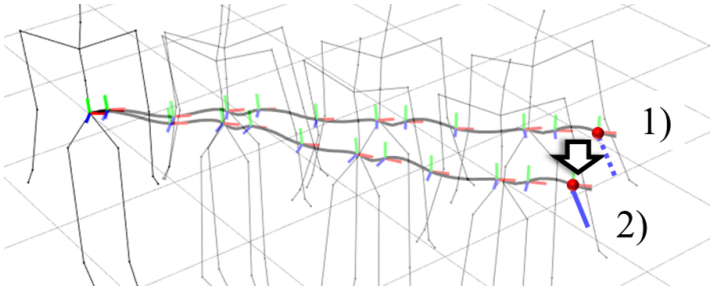


Figure 2: The result of the editing that deforms the walking motion in the straight. (a) shows when the final pose is rotated about 90°, and (b) shows when the final position is moved.

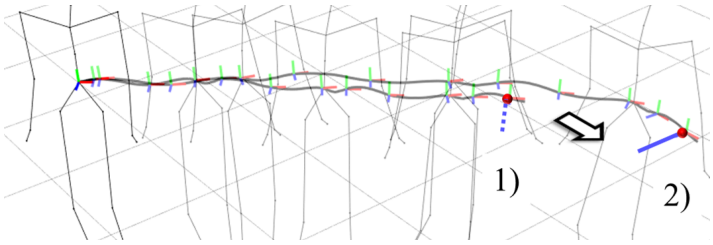
동작이 촬영되어 있으며 이를 30Hz 로 처리하여 사용하였다. 이 데이터에서 모션 그래프를 생성하였으며 그래프 탐색은 A* 알고리즘을 이용하였다. 캐릭터가 걸어다닐 환경은 Figure 5(a)와 같이 방과 방이 연결된 주거 공간으로 방과 방 사이 통로는 약 1.2m의 너비를 가진다. 우리는 모션 그래프를 이용하여 방 1(Room 1)에서 방 2(Room 2)로 이동하는 동작을 생성한다. 캐릭터는 방1에서 의자에서 일어난 후에 걷기 모션을 통해 방 2로 이동한 후 침대에 눕는다. 이 사이 걷기 동작을 생성해야 하므로 동작의 양 끝의 위치와 자세는 지정되어 있다.

4.2.2 그래프 탐색과 모션 변형 방법의 적용

방1에서 캐릭터가 의자에서 일어나는 동작(동작 1)과 방 2에서 침대에 눕는 동작(동작 2)을 연결시키기 위해서 그 사이의 걷는 동작을 생성시키고자 한다. 그러기 위해서 동작 1의 마지막 위치와 자세를 캐릭터의 시작 상태 v_s로 정의하고 동작 2의 시작 위치와 자세를 종료 상태 v_t 로 정의한다. 우선 모션 그래프를 탐색하여 v_s를 시작 상태로 하여 목표 상태 v_t 까지의 경로를 구할 수 있는지 확인한다. 탐색은 v_t 에 충분한 범위 내에 있을 때 종료하게 된다. 실험에서 탐색 종료 조건은 위치와 방향이 50cm, 45° 이하로 설정하였다. 탐색 결과는 갑자기 캐릭터가 이동하는



(a)



(b)

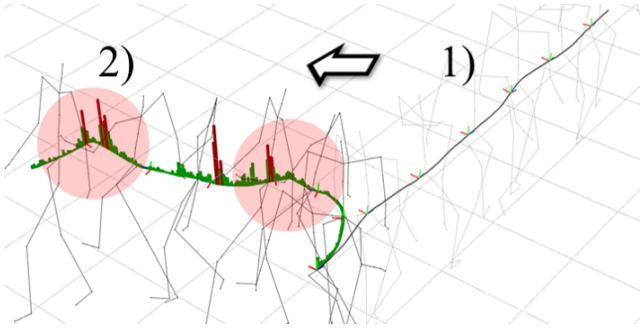
Figure 3: The result of the editing that deforms the side walking motion. (a) shows when the final pose is moved, and (b) shows when the final position and the direction are modified.

부자연스러운 동작이 된다(Figure 6(b)). 이를, 제안한 동작 변형 방법을 적용하여 부드럽게 연결되는 최종 동작을 생성한다. 이 방법은 단순히 위치뿐만 아니라 방향도 보정할 수 있으며 δ에 의해 경로 전체에 오차값이 전달되어 부드러워 지면서도 스타일을 보존 하는 동작이 생성된다. 만약 경로가 더욱 복잡할 경우, v_s와 v_t 사이에 중간 목표 지점을 추가하여 여러 경로로 나누어 탐색하여 해결할 수 있다.

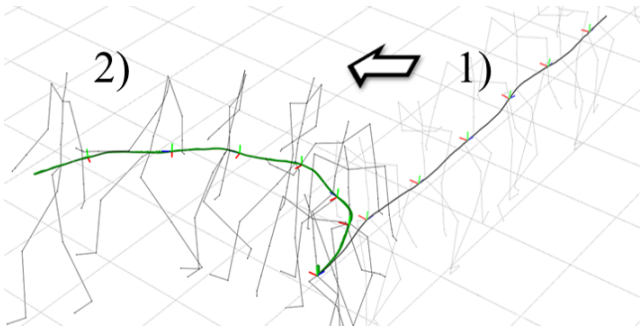
5. 결론 및 논의점

이 논문에서는 기존의 걷기 동작 데이터를 분석하여 새로운 걷기 동작으로 재구성하는 방법을 제안한다. 이를 위하여 각 프레임 별로 주 관절의 이동 경로를 그래프로 해석하였는데, 이 그래프는 위치와 방향을 정점으로 하고 인접 프레임의 정점과 연결된 형태로 구성하였다. 이와 같이 정의된 주 관절의 경로 그래프에 라플라스 연산자를 통해 좌표계를 변형하여 목표 위치와 방향에 맞춰 동작을 형성한다. 변형된 주 관절 경로에서 기존 모션 데이터에서 추출한 환경과의 접촉점을 유지하는 모션을 생성함으로써 새로운 모션 데이터를 얻을 수 있다.

평면 지형을 이용한 걷기 동작은 연속적이고 반복적인 동작으로 모션 데이터를 생성하였을 때 큰 위화감 없이 연결할 수 있다. 그러나 이 방법을 장애물이 있거나 고르지 않은 지형에 적용하였을 때에 위화감 없이 동작하기는 어렵다. 동작의 스타일을 유지한다는 것은 그 만큼 유연성이 적다는 말에 가깝다. 따라서 지형물에 따라 동작의 스타일을 변형시킨다면 더 다양한 모션 데이터를 생성하는 데에 이 논문을 이용할 수 있을 것이다.



(a)



(b)

Figure 4: (a) is the result of the deformation when the target is turned to 180°, which makes the motion more deforming than other experiments. We highlight the artifacts on the trajectory with the red circle. The bar on the trajectory represents $|\delta|$ where a higher bar is a higher value. To handle this problem, we interpolate the trajectory as the B-Spline, and apply the deformation. (b) shows the result after the interpolation.

감사의 글

이 논문은 산업통상자원부가 지원한 '5G 연계 증강현실 기기시스템 기술개발기술개발사업'으로 지원을 받아 수행된 연구 결과입니다. [과제명: 제조현장의 통합 작업지원을 위한 산업용 AR 지원 플랫폼 기술개발 / 과제고유번호: 20011076]

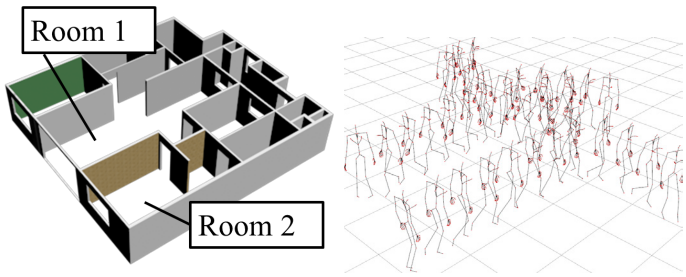
References

[1] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *ACM SIGGRAPH 2008 classes*, 2008, pp. 1–10.

[2] 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.

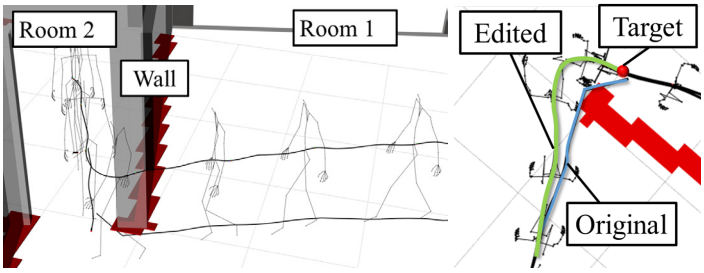
[3] E. S. Ho, T. Komura, and C.-L. Tai, "Spatial relationship preserving character motion adaptation," in *ACM SIGGRAPH 2010 papers*, 2010, pp. 1–8.

[4] C. Kang and S.-H. Lee, "Multi-contact locomotion using a contact graph with feasibility predictors," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, p. 1, 2017.



(a) (b)

Figure 5: (a) The house model as the environment, in which the searching is performed. (b) The source motion for generating the motion graph.



(a) (b)

Figure 6: The result of the motion graph that combined with our deforming method. The path starts from room 1 and ends at room 2 that the character must pass through the door, which is narrow. (a) is the perspective view of the motion. Also, (b) is the top view, and the original trajectory is not continuously connected to the target. Our method can connect smoothly two ends despite the mismatch between angles.

[5] J. Lee and S. Y. Shin, "A hierarchical approach to interactive motion editing for human-like figures," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 1999, pp. 39–48.

[6] M. Kim, K. Hyun, J. Kim, and J. Lee, "Synchronized multi-character motion editing," *ACM transactions on graphics (TOG)*, vol. 28, no. 3, pp. 1–9, 2009.

[7] Y.-Y. Tsai, W.-C. Lin, K. B. Cheng, J. Lee, and T.-Y. Lee, "Real-time physics-based 3d biped character animation using an inverted pendulum model," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 2, pp. 325–337, 2009.

[8] 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.

[9] H. J. Shin and H. S. Oh, "Fat graphs: constructing an interactive character with continuous controls," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*. Eurographics Association, 2006, pp. 291–298.

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

[11] S. Levine, J. M. Wang, A. Haraux, Z. Popović, and V. Koltun, "Continuous character control with low-dimensional embeddings," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

[12] 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.

[13] 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.

[14] J. Hwang, M. Yang, I. H. Suh, and T. Kwon, "Real-time grasp planning based on motion field graph for human-robot cooperation," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 1025–1032.

[15] M. G. Choi, M. Kim, K. L. Hyun, and J. Lee, "Deformable motion: Squeezing into cluttered environments," in *Computer Graphics Forum*, vol. 30, no. 2. Wiley Online Library, 2011, pp. 445–453.

[16] S. Lee and S.-H. Lee, "Projective motion correction with contact optimization," *IEEE transactions on visualization and computer graphics*, vol. 25, no. 4, pp. 1746–1759, 2018.

[17] Y. Lipman, O. Sorkine, D. Cohen-Or, D. Levin, C. Rossi, and H.-P. Seidel, "Differential coordinates for interactive mesh editing," in *Proceedings Shape Modeling Applications, 2004.* IEEE, 2004, pp. 181–190.

[18] L. Kovar, J. Schreiner, and M. Gleicher, "Footskate cleanup for motion capture editing," in *Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation*, 2002, pp. 97–104.

[19] G. D. Knott, *Interpolating cubic splines*. Springer Science & Business Media, 2000, vol. 18.

〈저자소개〉

이 석 원

- 2013년 광주과학기술원 (공학석사)
- 2019년 한국과학기술원 (공학박사)
- 2019년~현재 한국전자기술연구원 선임연구원
- 관심분야 : 캐릭터 애니메이션, 증강/가상 현실
- ORCID: 0000-0002-3005-3707

김 미 진

- 2019년 한국과학기술원 (공학석사)
- 2020년~현재 한국전자기술연구원 연구원
- 관심분야 : 증강/가상 현실, 텔레프레즌스
- ORCID: 0000-0002-5484-7886