

즉흥 음악 연주에 따른 춤 동작 자동 생성

Automatic Synthesis of Dancing Motions Matching to Ad-lib Musical Performances

강경규 Kyung-kyu Kang, 정유진 Yujean Choung, 김정아 Jung-A Kim,
리현희 Xianji Li, 김동호 Dongho Kim
숭실대학교 미디어학과 MAGIC Lab.

요약 본 연구에서는 사용자가 즉석에서 연주한 곡에 대응하는 춤 동작을 자동으로 생성하는 기술을 제안한다. 본 기법은 먼저 댄서로부터 모션 캡처 받은 춤 시퀀스를 분절화한 후 사용자가 신디사이저를 통해 직접 연주하여 얻은 MIDI 데이터를 분석해 음악에서의 특징점을 추출한다. 그리고 분절화 한 모션의 세그먼트들을 음악에 맞춰 다시 배열하여 새로운 춤 시퀀스를 생성함으로써 사용자가 연주한 음악과 어울리는 춤 동작을 자동으로 생성한다. 이를 위해 세 단계의 작업을 수행하게 되는데, 첫 번째 단계에서 모션 캡처를 통해 얻게 된 데이터에서 캐릭터의 위치와 자세를 기준으로 하여 긴 시퀀스를 의미 있는 작은 춤 동작으로 분절화한다. 두 번째 단계에서는 사용자의 연주를 통해 획득한 MIDI를 분석하여 특징점을 추출하고, 마지막 단계에서는 이를 바탕으로 음악에 기반한 음악과 동작을 합성한다. 본 연구는 음악과 댄스의 리듬감이 파괴되지 않도록 합성함으로써 우리의 연구는 기존 연구에 비해서 훨씬 자연스러운 춤 시퀀스를 결과물로 만들어 낸다.

핵심어: *Character animation, motion analysis, motion segmentation, motion synthesis, music analysis*

1. 서론

캐릭터 애니메이션 데이터를 게임, 영화, 가상현실 등의 분야에서 활용하는 연구가 최근 활발히 진행 중이다. 애니메이션 데이터를 모션 캡처를 통해서 획득하거나 수작업으로 제작하는 일은 비용이 많이 들어가기 때문에, 기존의 애니메이션 데이터를 활용하여 새로운 환경에서 재사용하는 기술들이 개발되어왔다.

최근 배경음악과 어울리는 춤 동작을 자동으로 생성하려는 연구들이 활발하게 진행되고 있다. 하지만, 이들 연구는 춤 동작 사이의 일관성이 부족하고 세그먼트의 단순 반복이 많아 만족스럽지 못한 결과를 얻고 있다. 또한, 음악데이터나 동작데이터의 변화로 인하여 원본 데이터의 고유의 특징들을 잃어버리는 경우도 많이 발생한다.

본 연구에서는 사용자가 직접 연주하여 얻은 MIDI 데이터를 분석하고 댄서로부터 모션 캡처 받은 캐릭터의 춤 동작을 분석하여 새로운 춤동작을 만드는 기법을 제안한다. 즉, 분석한 음악에 특징점들마다 강도(intensity)를 계산하고 그것과 가장 매칭되는 춤 동작을 찾아주는 시스템으로

원본 동작 데이터의 고유의 특징이 거의 손상되지 않으면서 음악과 모션이 자연스럽게 매칭되는 결과를 얻을 수 있다.

춤 시퀀스는 전문 댄서로부터 모션 캡처를 하여 획득하고, 댄서의 무대 위에서의 위치와 춤 자세를 기준으로 춤으로서 의미 있는 작은 동작인 세그먼트(segment)로 분절화한다. 분절 후에는 세그먼트마다 강도를 계산하여 저장한다. MIDI 데이터는 연주가 끝나면 바로 분석하여 마디마다 강도를 계산한다. 춤 합성 단계에서는 곡이 연주되는 동안 음악의 강도 변화와 매칭되도록 춤 세그먼트를 찾아서 합성 후 재생한다.

본 논문은 2장에서 관련연구를 소개하고, 3장에서 즉흥 연주곡에 어울리는 춤 동작을 자동 생성하는 알고리즘을 소개하며, 4장에서 실험 결과를, 5장에서 결론과 향후 연구를 소개한다.

2. 관련연구

본 연구와 관련된 이전 연구들은 크게 3가지 분야로 나눌 수 있다. 첫째로는 모션 분석 및 분절화에 관한 연구가 있

고, 둘째, 음악 분석 및 특징점 추출에 관한 연구, 셋째로 음악과 모션의 합성에 관한 연구들이 있다.

먼저 모션 분석 및 분절화에 관한 연구로서 Alankus[12]는 유저가 기계학습을 위한 학습 데이터를 수동으로 입력하고 Support Vector Machines(SVM) 기계학습 방법을 이용하여 자동으로 동작을 분절화 한다. Barbic[11]는 일상 생활(locomotion)의 동작을 Probabilistic Principal Component Analysis 혹은 Gaussian Mixture Model로 분석하여 구별되는 동작인 걷기, 뛰기, 앉기 동작 등으로 분절화 하는 연구를 보여준다. Shiratori[10]는 춤 동작에 특화된 분절화 방법을 연구하였는데 동작을 분석하여 분절 후보를 찾고, 배경음악의 박자(beat)와 일치하는 최종 분절을 찾는다. 반면에, 본 연구에서는 캐릭터의 위치변화와 자세의 변화에 주목하여 분절화를 수행한다.

MIDI 관련 연구로서 C. McKay[14]에서는 MIDI 데이터에서 악기, 전체적 기초(texture), 리듬, 강약법(dynamics), 음의 높이, 멜로디, 그리고 화음(chord)등을 통하여 음악의 특징을 추출하고 그 특징을 이용하여 주로 음악의 장르를 구분하고 있다. Xu, C.[15]는 SVM에 기초한 multi-layer classifier를 통하여 음악장르를 구분하는 연구이다.

음악과 모션의 동기화와 관련된 연구로서 먼저 Lee[7]는 컴퓨터 애니메이션의 배경음악과 모션을 동시에 수정하는 방법을 제안한다. 그리고 Kim[6], Alankus[12]는 작은 단위의 춤 동작 데이터베이스를 만들고 배경음악의 특징에 맞추어 춤 동작을 재조합(합성)하는 연구이다. 또한 춤동작 생성을 연구한 Shiratori[10]은 춤 동작을 리듬(rhythm)과 강렬함(intensity)을 기준으로 하여 분절화한 후에 배경음악에서도 같은 기준으로 특징을 찾아내어 춤을 합성한다. 이 연구에서는 춤에 대한 사람의 감정적인 면을 고려한 점이 특이하다.

기존 연구들은 모션과 음악을 동기화 하는 과정에서 원본 데이터의 손실이 이뤄져서 원래의 데이터가 가지고 있는 고유의 특징이 손실될 수 있다. 본 논문에서는 사용자가 직접 연주한 음악을 분석하여 얻은 강도값과 분절화된 춤 동작을 분석하여 얻은 강도값을 매칭시키는 시스템을 제안하여 원본데이터의 손실을 막고 기존 연구보다 자연스러운 결과를 얻는다.

3. 알고리즘

본 시스템에서 음악과 춤을 동기화 하는 과정은 총 3가지 단계로 나뉜다(그림 1). 첫 번째 단계에서 모션 캡처 통해 획득한 데이터를 분석하여 세그먼트로 분절화한다. 두 번째 단계에서는 사용자가 연주한 MIDI를 분석하여 특징점을 추출하고 마지막으로 음악과 동작을 각각의 특징에 매칭되게

자동으로 합성하여 자연스러운 춤 시퀀스 결과를 얻어 낸다.

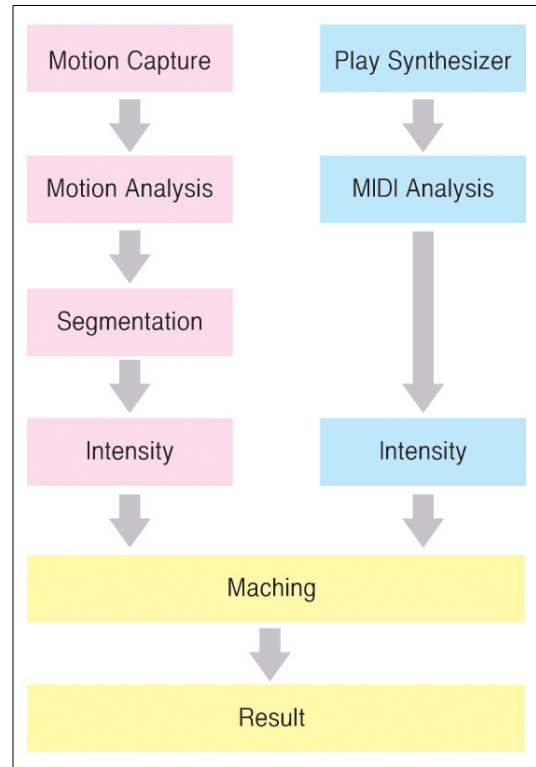


그림 1. 시스템 구조도

3.1 춤 동작 분절화

우리는 안무가(choreographers)가 일반적으로 배경음악에 맞춰 짧은 길이의 춤 동작을 만들고, 이것들을 이어 붙여서 긴 춤 동작을 만든다고 가정한다. 이러한 가정에서 우리는 모션 캡처를 통해서 획득된 길이가 긴 춤 동작 데이터는 춤으로서 의미 있는 짧은 길이의 춤 동작(세그먼트)으로 분절화하여, 또 다른 배경음악에 어울리는 춤을 생성할 때 안무가가 만들어 낼 짧은 춤 동작으로 이용한다.

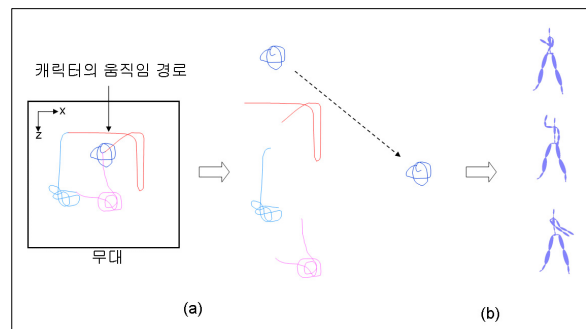


그림 2. 두 단계로 진행되는 분절화. (a) 위치 벡터를 이용한 분절 단계. (b) 자세 벡터를 이용한 분절 단계. 긴 춤 동작은 (a) 단계에서 대충의 세그먼트로 분절되고, (a)의 결과 세그먼트를 (b)단계에서 더 작은 세그먼트로 분절된다.

분절화 방법은 캐릭터의 위치 벡터(position vector, E)를 이용해서 길이가 긴 세그먼트로 나눈 후에, 캐릭터의

자세 벡터(posture vector, E_P)를 이용해서 앞 단계의 긴 세그먼트를 더 작은 세그먼트로 분절한다[그림 3]. 위치 벡터는 월드 좌표계에서 캐릭터의 위치를 의미하며 월드 좌표계의 원점은 캐릭터가 춤을 추는 무대 중심과 일치한다. 캐릭터 계층 구조에서 루트가 되는 엉덩이(hip)관절의 위치가 위치 벡터로 변환된다. 자세 벡터는 모든 관절들의 회전 벡터의 크기로 이루어진 벡터이다. 위치 벡터는 3차원 벡터이며 자세벡터는 관절 개수만큼의 원소를 갖는다.

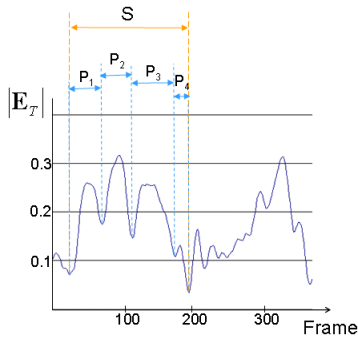


그림 3. 위치 벡터 크기 그래프를 이용한 분절의 예. S는 세그먼트(segment)를 의미하며 P는 세그먼트를 구성하는 프리미티브(primitive)이다.

위치 벡터를 이용한 분절과 자세 벡터를 이용한 분절, 모두 동일한 방법으로 분절화를 진행한다. [그림 4]와 같이 벡터의 크기를 그래프로 그려보면 증가와 감소가 반복적으로 나타나는 것을 볼 수 있다. 위치 벡터와 자세 벡터의 크기 그래프에서 모두 반복성을 볼 수 있기 때문에 이를 이용하여 분절화한다.

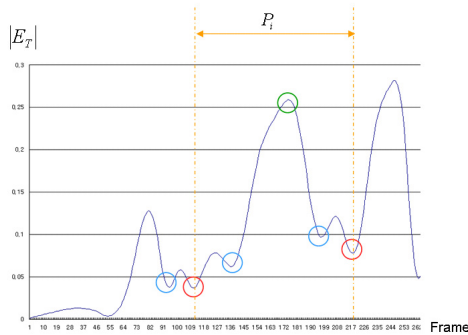


그림 4. 그래프에서 프리미티브 찾기. 빨간색 원들이 지역 최저값으로 프리미티브의 시작과 끝을 나타낸다. 파란색 원들은 지역 최저값 후보이며, 녹색 원은 프리미티브 내의 지역 최대값이다.

분절화는 두 단계로 이뤄진다. 첫 번째 단계는 그래프의 반복성을 작은 단위인 프리미티브(primitive)로 나누는 것이다. 먼저 일정한 구간 내에서 지역 최소값(local minimum)을 찾고 지역 최대값(local maximum) 이후의 영역에서 지역 최소값을 찾아 이전 지역 최소값과의 사이 구간을 하나의 프리미티브로 나눈다[그림 5]. 프리미티브는 [그림 4]

에서의 P와 같이 두 개의 지역 최소값 사이에 지역 최대값을 하나 갖는 형태가 된다. 지역 최소값 근처에는 지역 최소값 후보들이 존재하는데 일정 구간내의 후보들 중에서 가장 작은 최소값인 지역 최소값을 선택한다. 이 구간의 길이가 짧아지면 프리미티브의 길이가 짧아지고, 구간의 길이가 길면 그 반대 효과가 있다.

분절화의 그 다음 단계는 첫 번째 단계의 프리미티브들을 그룹화 하는 것이다. 그룹화에서는 하나의 프리미티브에서 시간상으로 앞에 이웃한 프리미티브들의 지역 최대값의 평균을 계산하고, 뒤에 위치한 프리미티브들의 지역 최대값의 평균 계산하여 이 두 평균의 차이를 저장하여 이용한다. 프리미티브마다 이 평균 차 값을 모두 계산하고, 이 평균의 차이가 제일 큰 프리미티브를 그룹을 두 개로 나누는 경계로 고려할 수 있다.

그룹화는 재귀적으로 이뤄지는데 평균의 차이가 가장 심한 프리미티브를 기준으로 두 개의 그룹으로 나누고, 다시 나눠진 그룹들에서 평균의 차이가 큰 프리미티브를 기준으로 두 개의 그룹으로 나눈다. 이렇게 재귀적으로 프리미티브를 그룹화 하다가 그룹의 길이가 너무 짧아지지 않도록 재귀의 깊이를 조정하여 재귀적인 그룹화를 중지한다[그림 6]. 최종으로 그룹화된 프리미티브들이 하나의 세그먼트가 되는 것이다. 재귀의 깊이의 조절과 최소 세그먼트길이의 제한을 두어서 세그먼트의 길이가 지나치게 짧아지는 것을 방지할 수 있다.

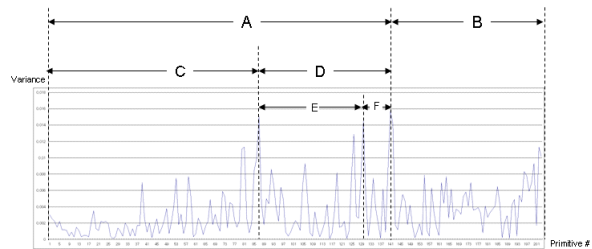


그림 5. 프리미티브 그룹화. 그룹화는 재귀적으로 이뤄지며, 재귀 호출의 깊이를 조절하여 한 그룹을 이루는 프리미티브의 개수를 조절할 수 있다. 첫 번째 그룹화에서 전체 프리미티브는 A와 B 그룹 두 그룹으로 나누어진다. 두 번째 그룹화에서 A는 C와 D 그룹으로 된다. 다시 D는 E와 F 그룹으로 나뉜다. 여기서 그룹화를 멈추면 E와 F는 각각 세그먼트가 된다.

위치 벡터를 이용한 분절화에서는 그룹화의 깊이를 알게 하여 세그먼트의 길이를 길게 한다. 이 세그먼트들을 또 다시 자세 벡터를 이용한 분절화를 통해서 적당한 길이의 세그먼트를 최종으로 얻을 수 있다. 이 세그먼트들이 다음 춤 동작 생성 단계에서 이용된다.

3.2 배경음악의 특징점 추출

이 장에서는 배경음악과 동작을 자연스럽게 합성하기 위해 배경음악의 특징점을 추출해내는 방법에 대하여 소개한

다. 이 단계에서 추출된 특징점의 값은 해당 시간 값과 함께 모션단계로 전달되어 작동하게 된다. 본 논문에서는 음악에서 다음의 요소들을 분석하여 강도값을 추출한다.

- 소리의 크기 (Note Velocity, 건반을 누르는 속도)
- 음의 높낮이 (Chords, 코드 사이의 높이차)
- 템포 (Tempo, 4분음표의 빠르기)

위의 요소들은 음악의 분위기나 특징을 크게 반영하는 기본적인 요소로서 음악의 장르 구분에도 많이 쓰인다[14]. 해당 요소들을 분석하여 최종적으로 동작에 보내지는 강도값을 비교하는 시간 단위는 한 마디이다. 그 이유는 한 가지 동작을 시작하여 완료하는 시간이 짧지 않으므로, 짧은 시간 내에 많은 강도값을 보내면, 한 가지 동작이 완료되기 전에 다른 동작을 해야 하므로 부자연스러운 동작을 만들어 낼 수 있기 때문이다. 따라서 동작을 완료할 수 있는 최소 단위의 시간을 한 마디로 정하였다.

각 마디의 강도값(M)을 구하는 식은 다음과 같다.

$$M(m_i) = B + v(m_i) + c(m_i), \quad i = 1, 2, 3, \dots, n \quad (1)$$

m 은 한 마디를 나타내고, i 는 현재 마디의 번호를 나타내며, n 은 노래의 전체 마디 수이다. B 는 모든 마디에 동일하게 주어지는 기본 값으로, 각 마디에 주어지는 기본 값 1과, 곡 전체의 템포(tempo)를 연산한 값을 더하여 계산한다. 템포의 연산방법은 곡의 템포를 받아 기준템포인 88을 빼서, -1와 1 사이의 값으로 정규화한 값이다[표1]. 템포가 빠를수록 큰 수를 가지기 때문에 빠른 곡일수록 큰 강도값을 가질 수 있다.

표 1. 빠르기 별 템포(Tempo Marks)

빠르기 말	Tempo	빠르기
Grave	42이하	아주 느리게
Largo, Lento	46~52	느리게
Larghetto	56	Largo보다 약간 빠르게
Adagio	60	적당히 느리게
Andante	66	건는 빠르기
Andantino	69	Andante보다 조금 빠르게
Moderato	88	보통 빠르기
Allegretto	108~120	Allegro보다 조금 느리게
Allegro	132	빠르게
Vivace	160	활기있게
Presto	184	아주 빠르게
Prestissimo	208이상	아주아주 빠르게

$v(m_i)$ 는 한 마디 내에 있는 음표(note)들이 가지는 음의 세기 값(velocity)값의 차이를 평균 낸 값이다. $c(m_i)$ 는 한 마디 내에 있는 음표들이 가지는 코드를 마디의 음계번호 값을 통해 수치화하여 계산한다. 코드 값의 평균과 코드 사이 높이차의 평균값을 각각 가중치를 주어 계산해낸 값으로

계산식은 다음과 같다.

$$c(m) = (w_1 A_c + w_2 A_g) / (w_1 + w_2), \quad (w_1 < w_2) \quad (2)$$

A_c 와 A_g 는 각각 코드 값의 평균과 코드 사이의 간격의 평균값을 나타낸다. w_1 과 w_2 는 A_c 와 A_g 의 비율을 조절하기 위한 가중치이다.

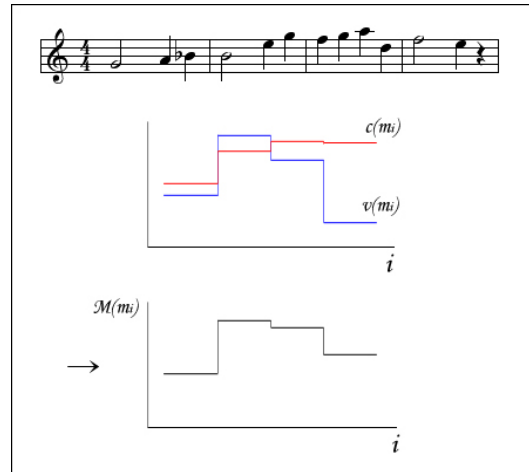


그림 6. 마디 별 강도값을 구한 예. $v(m_i)$ 와 $c(m_i)$ 를 통해 $M(m_i)$ 을 구해낸다.

위의 식들을 토대로 획득한 각각의 소절 값을 이용하여 최종 추출 값 $I(m)$ 을 다음과 같은 순서로 얻어낸다. 먼저 각 마디가 가지는 강도값의 차이에 대한 평균을 얻어낸다.

$$A_m = \frac{\sum_{i=0}^n |M(m_i) - M(m_{i+1})|}{n} \quad (3)$$

m 은 한 마디를 나타내고, $M(m_i)$ 은 i 번째 마디의 강도값을 나타낸다. 다음으로 이 평균값을 기준 치로 두어 획득하는 최종 추출 값 $I(m)$ 은 $|M(m_i) - M(m_{i+1})| > A_m$ 을 만족하는 $M(m_{i+1})$ 과 그 마디의 시작 시간이 된다. 즉, $I(m)$ 에 포함되는 데이터 정보는 앞에 놓인 마디와의 차이 값이 평균보다 큰 마디의 강도값 $M(m_{i+1})$ 과 그 마디의 시작시간이다.

3.3 춤 동작 생성

우리는 음악 분석에서 얻어진 시간에 따른 강도값의 변화에 맞추어 적당한 세그먼트를 찾아서 캐릭터의 춤 동작에 적용한다. 춤 세그먼트의 강도는 모든 프레임에서 자세 벡터 크기들의 평균을 이용한다. 음악과 춤의 강도의 범위를 0과 10사이의 값으로 조정하였고, 음악의 강도와 가장 비슷한 강도를 갖는 춤 세그먼트를 찾아 재생하게 된다[그림 7].

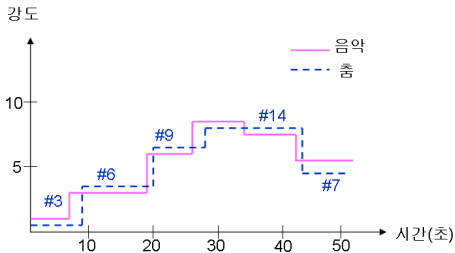


그림 7. 음악의 강도값에 따른 춤 세그먼트 찾기. 시간의 흐름에 따라 음악의 강도가 변화하는데 유사한 강도값을 갖는 세그먼트를 찾아서 재생한다. 이 음악에서는 3번, 6번, 9번, 14번, 7번 춤 세그먼트 순으로 재생된다.

찾아진 다음 춤 세그먼트는 바로 캐릭터에 적용되지 않고 재생되고 있는 세그먼트가 끝나기 1초(30프레임) 전에 다음 세그먼트와 보간되면서 재생된다[그림 8]. 이렇게 하면 춤 세그먼트와 세그먼트가 부드럽게 이어지는 것을 위한 것이다. 보간 단계에서는 캐릭터의 각 관절과 위치가 프레임마다 가중치가 이전 세그먼트에서 천천히 다음 세그먼트로 커지는 형태의 선형 보간을 수행한다.

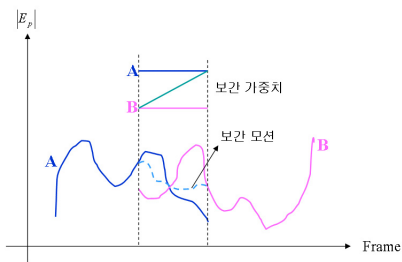


그림 8. 두 세그먼트의 보간. 두 세그먼트는 30 프레임(1초) 정도 겹쳐 재생된다. 이 구간에서 자세는 A세그먼트에서 B 세그먼트로 서서히 선형보간 된다.

이 과정을 통해 이미 분석된 음악의 강도값과 분절화 된 춤 세그먼트의 강도값이 서로 어울리도록 춤 세그먼트를 실시간으로 찾아준다.

4. 실험결과

본 연구에서는 세 개의 춤 데이터- 'Yoo', 'Hug', 'Diva' -를 전문 댄서의 춤으로부터 획득해서 실험에 이용하였다. 사용된 가상 캐릭터의 관절은 모두 28개이며 이들은 3자유도(회전)를 갖고 있다. 단, 캐릭터 계층구조의 최상위 관절은 3자유도(이동)가 추가된다. 모션 데이터는 초당 30프레임으로 구성되는데, 'Yoo'는 5,583프레임, 'Hug'는 6,826프레임 그리고 'Diva'는 6,410프레임으로 구성되었다.

춤 시퀀스의 분절화 결과로 'Yoo'는 68개의 세그먼트로

, 'Hug'는 70개의 세그먼트로, 'Diva'는 64개의 세그먼트로 분절되었다. 분절화는 음악이 연주되기 전에 미리 실행하며 세그먼트 단위로 모션 파일에 저장하고 세그먼트의 강도를 미리 계산하여 함께 저장한다. 합성에 이용된 배경음악은 사용자가 즉흥에서 미디 건반을 연주하여 얻은 28초 분량의 MIDI 데이터를 이용하였다.

사용자가 연주한 임의의 곡에 대한 춤 동작을 합성한 춤 시퀀스는 [그림 9]와 같다. [그림 10]의 결과에서도 알 수 있듯이 곡의 강도가 높아지면 춤 강도가 높은 춤 세그먼트가 재생되는 것을 볼 수 있다. 곡의 강도 변화가 없을 때는 해당 강도를 갖는 다른 춤 세그먼트를 찾아서 재생하게 하여 동일한 세그먼트가 연속되어 여러 번 반복 재생되는 것을 방지하였다. 곡의 강도에 해당하는 다른 춤이 없을 경우 가장 강도가 비슷한 다른 춤을 찾아 재생한다.

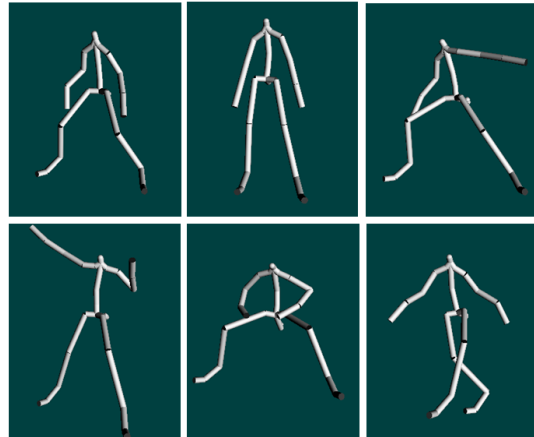


그림 9. 연주된 곡의 강도에 의해서 합성된 춤 동작

5. 결론 및 향후 연구

본 연구에서는 사용자가 즉석에서 연주한 곡에 대응하는 춤 동작을 자동으로 생성하는 기술을 제안하였다. 사용자가 즉석으로 신디사이저를 연주하여 획득한 MIDI 데이터를 분석하여 특징점을 추출하고 특징점의 특성에 따라 강도를 부여한 후 모션 캡처를 통해서 얻어진 모션 데이터를 자동으로 분절화하여 얻어진 세그먼트들의 특징에 따라 부여된 강도를 음악과 매칭하여 최종으로 자연스럽게 음악과 모션이 합성된 결과를 얻었다.

본 연구는 사용자의 참여를 유발 할 뿐 아니라 자신의 연주에 맞는 춤의 자동생성을 통한 재미요소를 가지고 있기 때문에 향후 게임이나 인터랙티브 아트 분야에 적용될 수 있는 장점을 가지고 있다. 또한 값비싼 모션 캡처 장비를 이용하여 춤 동작 데이터를 얻는 방법 대신 기존의 모션 데이터를 통하여 데이터 베이스를 만들고 이를 재활용하여 새로운 콘텐츠를 제작 하는데 이용할 수 있을 것이다.

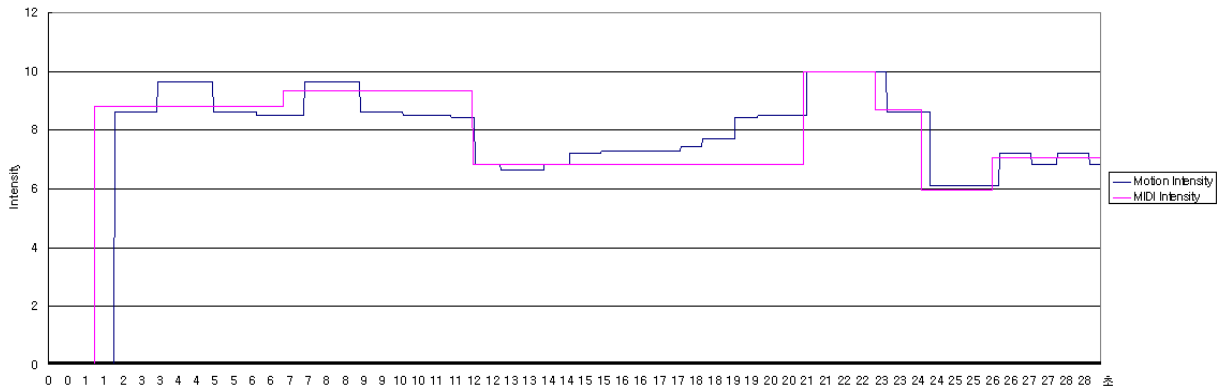


그림 10 음악과 모션의 강도값 매칭 결과. 분홍 선은 미디 분석 후 얻어낸 강도값이고 파란 선은 모션의 강도값이다. 이 그래프 결과와 같이 모션과 음악의 강도값에 따라 매칭되고 있음을 알 수 있다.

향후 연구로는 춤동작을 사용자가 직접 합성할 수 있도록 하는 인터페이스가 개발되어야 할 것이다. 또한 사용자가 연주하면서 실시간으로 음악을 분석하여 춤동작과 매칭할 수 있도록 하면 사용자로 하여금 현재보다 더 흥미를 느낄 수 있게 될 것이다.

감사의 글

본 연구는 서울시 산학연 협력사업의 지원으로 이루어졌습니다.

참고문헌

- [1] M. Gleicher, "Retargetting Moiton to New Characters", In Proceedings ACM SIGGRAPH 98, 33-42, 1998.
- [2] O. Arikan, D. Forsyth, J. O'Brien, "Motion Synthesis from Annotations", ACM Transactions on Graphics, Vol. 22, No.3, 402-408, 2003.
- [3] L. Kovar, M. Gleicher, F. Pighin, "Motion Graphs", ACM Transactions on Graphics, Vol. 21, No.3, 473-482, 2002.
- [4] S. L. Park, H. J. Shin, S. Y. Shin, "On-line Locomotion Generation Based on Motion Blending", In Proceedings ACM SIGGRAPH Symposium on Computer Animation, 105-111, 2002.
- [5] J. Lee, J. Chai, P.S.A. Reitsma, J. K. Hodgins, N. S. Pollard, "Interactive control of avatars animated with human motion data", ACM Transactions on Graphics Vol. 21, No. 3, 491-500, 2002.
- [6] T. H. Kim, S. I. Park, S. Y. Shin, "Rhythmic motion synthesis based on motion beat analysis", ACM Transactions on Graphics, Vol. 22, No. 3, 392-401, 2003.
- [7] H. C. Lee, I. K. Lee, "Automatic Synchronization of Background Music and Motion in Computer Animation", EUROGRAPHICS 2005, Computer Graphics Forum Vol. 24, No.3, 353-362, 2005.
- [8] K. H. Lee, M. G. Choi, J. Lee, "Motion Patches: Building Blocks for Virtual Environments Annotated With Motion Data", ACM Transactions on Graphics, Vol. 25, No. 3, 898-906, 2006.
- [9] T. Shiratori, A. Nakazawa, K. Ikeuchi, "Detecting dance motion structure through music analysis", International Conference on Face and Gesture Recognition , 857-862, 2004.
- [10] T. Shiratori, A. Nakazawa, K. Ikeuchi, "Dancing-to-Music Character Animation", EUROGRAPHICS 2006, Computer Graphics Forum, Vol. 25, No. 3, 449-458.
- [11] J. Barbic, A. Safonova, J. Pan, C. Faloutsos, J. K. Hodgins, N. Pollard, "Segmenting Motion Capture Data into Distinct Behaviors", Graphics Interface 2004, 185-194.
- [12] G. Alankus, A.A. Bayazit, O.B. Bayazit, "Automated Motion Synthesis for Dancing Characters", Journal of Computer Animation and Virtual Worlds , 259-271, 2005.
- [13] A. Nakazawa, S. Nakaoka, K. Ikeuchi, K. Yokoi, "Imitating Human Dance Motions through Motion Structure Analysis", In Proceedings of International Conference on Intelligent Robots and Systems, 2539-2544, 2002.
- [14] C. McKay and I. Fujinaga, "Automatic genre classification using large high-level musical feature reach-sets", In Proceedings of ISMIR, 31-38, 2004.
- [15] Xu, C., N. C. Maddage, X. Shao, F. Cao, and Q. Tian, "Musical genre classification using support vector machines", Proceedings of the International

