

Normalized Cuts을 이용한 그래프 기반의 하이레벨 모션 분할

(Graph-based High-level Motion Segmentation using Normalized Cuts)

윤성주[†] 박안진[†] 정기철^{**}
(Sungju Yun) (Anjin Park) (Keechul Jung)

요약 모션 캡처 장비는 사람의 자연스러운 행동이나 동작 정보를 정밀하게 얻기 위해 널리 이용되며, 영화나 게임과 같은 콘텐츠 제작 시에 자주 활용된다. 하지만 모션 캡처 장비가 고가이고 사용하기 불편하기 때문에 대부분의 경우 한번 입력받은 데이터를 모션별로 분할하고 상황에 맞게 재결합하여 이용하며, 입력 데이터를 모션별로 분할하는 작업은 대부분 수동으로 이루어진다. 이 때문에 캡처된 모션 데이터를 자동으로 분할하기 위한 연구들이 최근 다양하게 시도되고 있다. 기존의 연구들은 크게 각 프레임의 전역적 특징을 고려하는 오프라인 방법과 이웃하는 프레임 사이의 유사도를 고려하는 온라인 방법으로 나누어진다. 본 논문에서는 온라인과 오프라인 방법을 통합한 그래프 기반의 하이레벨 모션 분할 방법을 제안한다. 하이레벨 모션은 모션 내에서 반복되는 프레임을 포함하는 특징을 가지고 있다. 우리는 이 특징을 기반으로 이웃하는 프레임뿐만 아니라 일정시간내의 모든 프레임 사이의 유사도를 고려하는 그래프를 생성하며, 그래프의 정점(vertex)에는 프레임 정보를 간선(edge)의 가중치는 두 프레임 사이의 유사도를 반영한다. 그래프를 분할하기 위해 분할된 간선의 가중치를 전역적으로 최소화할 수 있는 normalized cuts을 이용하며, 분할된 정점의 집합은 하이레벨 모션을 의미한다. 결과적으로 제안된 방법은 이웃하는 프레임뿐만 아니라 일정시간내의 모든 프레임 사이의 유사도를 반영하는 그래프를 전역적으로 최소화함으로써 온라인과 오프라인 방법을 동시에 고려할 수 있으며, 실험에서 제안된 방법은 기존의 오프라인 방법 중 하나인 GMM과 온라인 방법 중 하나인 PCA를 이용한 방법보다 좋은 결과를 보였다.

키워드 : 모션 캡처 장비, 모션 분할, Normalized Cuts

Abstract Motion capture devices have been utilized in producing several contents, such as movies and video games. However, since motion capture devices are expensive and inconvenient to use, motions segmented from captured data was recycled and synthesized to utilize it in another contents, but the motions were generally segmented by contents producers in manual. Therefore, automatic motion segmentation is recently getting a lot of attentions. Previous approaches are divided into on-line and off-line, where on-line approaches segment motions based on similarities between neighboring frames and off-line approaches segment motions by capturing the global characteristics in feature space. In this paper, we propose a graph-based high-level motion segmentation method. Since high-level motions consist of repeated frames within temporal distances, we consider similarities between neighboring frames as well as all similarities among all frames within the temporal distance. This is achieved by constructing a graph, where each vertex represents a frame and the edges between the frames are weighted by their similarity. Then, normalized cuts algorithm is used to partition the constructed graph into several sub-graphs by globally finding minimum cuts. In the

· 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음

[†] 학생회원 : 송실대학교 미디어학과
lay-wing@nate.com
anjin.park@gmail.com

^{**} 종신회원 : 숭실대학교 미디어학부 교수
kcjung@ssu.ac.kr
논문접수 : 2008년 8월 7일
심사완료 : 2008년 10월 16일

Copyright©2008 한국정보과학회 : 개인 목적이거나 교육 목적인 경우, 이 저작물의 전체 또는 일부에 대한 복사본 혹은 디지털 사본의 제작을 허가합니다. 이 때, 사본은 상업적 수단으로 사용할 수 없으며 첫 페이지에 본 문구와 출처를 반드시 명시해야 합니다. 이 외의 목적으로 복제, 배포, 출판, 전송 등 모든 유형의 사용행위를 하는 경우에 대하여는 사전에 허가를 얻고 비용을 지불해야 합니다.

정보과학회논문지: 소프트웨어 및 응용 제35권 제11호(2008.11)

experiments, the results using the proposed method showed better performance than PCA-based method in on-line and GMM-based method in off-line, as the proposed method globally segment motions from the graph constructed based similarities between neighboring frames as well as similarities among all frames within temporal distances.

Key words : Motion Capture, Motion Segmentation, Normalized Cuts

1. 서론

모션 캡처 장비는 사람의 자연스러운 행동이나 동작 정보를 정밀하게 얻기 위해 주로 이용되며, 영화나 게임과 같은 콘텐츠에서 3D 캐릭터에 부드러운 움직임을 부여하기 위해 자주 활용되고 있다. 그러나 모션 캡처 장비가 고가이고 사용하기 불편하기 때문에 대부분의 경우 한번 입력 받은 데이터를 모션별로 분할하여 저장하고, 필요한 모션을 상황에 맞게 재결합하여 사용하였다. 이 경우 캡처된 데이터를 모션별로 분할하는 단계가 매우 중요하지만, 대부분의 경우 사람이 수동으로 분할하기 때문에 많은 시간이 소요되는 단점이 있다. 이를 해결하기 위해 최근 캡처 데이터를 모션별로 자동으로 분할하기 위한 방법들이 다양하게 연구되고 있다[1-8].

캡처 데이터를 모션별로 분할하는 기존의 방법들은 크게 오프라인 방법과 온라인 방법으로 분류된다[1]. 오프라인 방법은 특징공간에서 각 프레임의 전역적 특징들을 분석하는 방법으로 K-means와 GMM과 같은 군집화 방법을 주로 이용하였다. Barbic 등[1]은 GMM을 이용하여 캡처 데이터를 모션별로 분할하는 방법을 제안하였다. Kim 등[2]은 실시간으로 모션을 분할/합성하는 방법을 제안하였으며, K-means를 이용하여 캡처 데이터를 분할하였다. Lee와 Elgammal[3]은 캡처 데이터를 모션별로 분할하고 새로운 모션 시퀀스를 생성하는 방법을 제안하였으며, SOM을 이용하여 캡처 데이터를 학습하고 K-means를 이용하여 분할하였다. Sakamoto 등[4]은 고차원의 캡처 데이터를 시각화하기 위해 SOM을 이용한 모션 맵(motion map)을 생성하였으며, 이를 이용하여 캡처 데이터를 모션별로 분할하고 검색하였다. 이와 같은 오프라인 방법은 특징공간에서 모션들의 전역적 특징을 고려하여 분할하는 방법으로써, 확실하게 분할될 수 있는 특징값을 가진 캡처 데이터에서 좋은 성능을 보이지만, 대부분의 캡처 데이터는 잡음과 같은 외부 요인에 의해 확실하게 분할될 수 있는 특징값을 가지기 어렵다.

이를 해결하기 위해 이웃하는 프레임(Frame) 사이의 유사도를 고려하는 온라인 방법이 다양하게 제안되고 있다. Fod 등[5]은 팔의 움직임을 모션별로 분할하는 방법을 제안하였으며, 연속된 프레임에서의 팔의 4가지 관절의 자유도(degree of freedom) 방향을 이용하였다.

Kwon과 Shin[6]은 걷는 동작과 뛰는 동작을 분할하는 방법을 제안하였으며, 무게중심의 연속되는 두 최고점의 간격을 기반으로 순차적으로 분할하였다. Yamasaki와 Aizawa[7]는 모션 캡처 장비가 아닌 여러 대의 카메라를 이용하여 캡처한 3D 비디오에서 이웃하는 두 프레임 사이의 유사도 차이를 계산하며, 계산 결과에서 지역 최소점을 찾아 3D 비디오 데이터를 모션별로 분할하였다. Barbic 등[1]은 PCA(principal component analysis)를 이용하여 캡처 데이터를 모션별로 분할하는 방법을 제안하였으며, PCA에 의해 계산된 연속되는 각 프레임의 손실값의 누적 변화량을 기반으로 분할하였다. Bouchard와 Bardler[8]는 캡처 데이터를 신경망으로 학습하였으며, 연속 프레임에서 신경망 결과가 지역 최고점이 될 때 분할하였다. 위에서 언급한 온라인 방법들이 최근 오프라인 방법보다 다양하게 제안되고 활용되고 있지만 이웃하는 프레임의 유사도만을 고려하기 때문에 너무 작은 단위의 모션으로 분할하는 문제가 발생한다.

본 논문에서는 온라인과 오프라인 방법을 통합한 그래프 기반의 하이레벨 모션 분할 방법을 제안한다. 하이레벨 모션은 로우레벨 모션의 집합인 걷기, 뛰기와 같은 의미있는 개별적인 행위를 뜻하며, 로우레벨 모션은 왼손 올리기 등의 작은 단위의 행위로서 각 프레임을 뜻하는 부분동작들로 이루어져 있다. 하이레벨 모션의 가장 큰 특징은 모션 내에서 반복되는 프레임은 포함한다는 것이다. 우리는 이 특징을 기반으로 이웃하는 프레임 뿐만 아니라 모든 프레임 사이의 유사도를 함께 고려하는 그래프를 생성하며, 그래프의 정점(vertex)에는 프레임 정보를 간선(edge)의 가중치에는 두 프레임 사이의 유사도를 반영한다. 그러나 서로 다른 2개의 하이레벨 모션에서 동일한 프레임이 반복되었을 경우 하나의 하이레벨 모션으로 잘못 분할되는 문제점이 있다. 이를 해결하기 위해 우리는 시간축에서 프레임 사이의 거리가 가까울수록 유사도가 증가하고 거리가 멀어질수록 유사도가 감소하는 시간 항(time term)을 가중치를 설정할 때 추가적으로 고려하며, 이로 인해 모든 프레임사이의 유사도가 아닌 일정시간내의 프레임사이의 유사도만을 고려하였다. 그래프를 분할하기 위해 본 논문에서는 분할된 간선의 가중치 합을 전역적으로 최소화하고 분할된 정점의 내부 가중치의 합을 최대화하는 normalized cuts[9]을 이용하며, 이때 분할된 정점의 집합은 하이레

벨 모션을 의미한다. 결과적으로 본 논문에서 제안한 방법은 이웃하는 프레임뿐만 아니라 일정시간내의 모든 프레임을 반영하는 그래프를 전역적으로 최적화함으로써 온라인과 오프라인 방법을 동시에 고려할 수 있으며, 실험에서 기존의 오프라인 방법 중 하나인 GMM기반의 방법과 온라인 방법 중 하나인 PCA기반의 방법보다 좋은 성능을 보였다.

2장에서는 모션 분할에 이용할 특징값과 그래프의 생성에 대하여 설명한다. 3장에서는 본 논문에서 이용하는 normalized cuts에 대해 설명하고 4장에서는 제안된 방법과 기존의 방법인 GMM, PCA기반의 방법을 비교하여 성능평가를 한다. 마지막으로 5장에서는 최종 결론과 앞으로의 연구계획에 대해서 논의한다.

2. 모션 분할을 위한 유사도 그래프

2.1 특징값

실험에서 이용한 모션 캡처 데이터는 1초에 120개 프레임으로 이루어져 있으며, 각 프레임마다 114차원의 데이터를 가지고 있고 114차원의 데이터는 38개 관절의 3차원 좌표이다. 본 논문에서는 위치정보와 회전정보에 영향을 받지 않는 특징값을 추출하기 위해, 중앙에 있는 관절 Hip에서 각 관절 사이의 거리를 이용하였으며 k 번째 프레임의 특징값은 다음과 같이 정의할 수 있다.

$$F^k = \{P_1, P_2, \dots, P_n, \dots, P_{37}\} \quad (1)$$

여기서 P_n 는 Hip과 n 번째 관절 사이의 거리를 나타내며, 우리는 유클리디언 거리를 이용하여 계산하였다. n 의 최대 수는 총 관절 수 38개에서 Hip 관절을 뺀 개수이며, 결론적으로 본 논문에서 이용한 특징값은 각 프레임 당 37차원의 데이터를 가진다.

2.2 시간축을 고려한 프레임 유사도 그래프

그림 1은 실험에서 이용한 캡처 데이터의 일부를 보여주며, 하이레벨 모션은 그림 1에서 사각형으로 표시한 것과 같이 걷기와 차기 같은 의미있는 행위로 구성되어 있다. 하나의 하이레벨 모션은 여러 개의 로우레벨 모션(그림 1의 타원)으로 이루어져 있고 로우레벨 모션은 여러 개의 부분동작(그림 1의 알파벳)으로 구성되어있으며, 여기서 부분동작은 프레임을 뜻한다.

그림 1에서 하나의 모션 안에 포함된 부분동작임에도 불구하고 두 프레임 사이의 유사도가 높지 않다. 예를 들어 발차는 동작(그림 1의 High-Level Behaviors 2)은 동작이 크기 때문에 각 프레임 사이의 유사도가 걷는 동작(그림 1의 High-Level Behaviors 1)에 비해 낮다. 하지만 각각의 부분동작 위에 표시된 알파벳과 같이 하나의 하이레벨 모션안에는 부분동작이 주기적으로 반복되며, 우리는 이 특징을 기반으로 이웃하는 프레임뿐만 아니라 모든 프레임의 유사도를 고려한 가중치 그래프 $G = \langle \nu, \epsilon \rangle$ 를 구축한다. 그래프에서 정점 ν 는 프레임의 집합을, 간선 ϵ 는 각 프레임 사이 연결의 집합을, 간선의 가중치는 두 프레임 사이의 유사도를 나타낸다.

그러나 서로 다른 2개의 하이레벨 모션에서 동일한 프레임이 반복되는 경우 하나의 하이레벨 모션으로 잘못 분할되는 문제가 발생한다. 예를 들어 그림 1에서 부분동작 A는 걷는 행위 안에서만 반복되는 것이 아니라 차는 행위에서도 반복되며, 이때 걷는 행위에서 차는 행위 사이에 있는 부분동작 A까지 하나의 하이레벨 모션으로 분할되는 잘못된 결과가 나올 수 있다. 이 문제를 해결하기 위해 가중치를 계산할 때 시간축에서 프레임 사이의 거리가 가까울수록 높은 비용(cost)을 거리가 멀어질수록 낮은 비용을 할당하는 시간 항 $t(i, j)$ 를 추가적으로 고려하며, $t(i, j)$ 는 식 (2)와 같이 계산되고, 두 프레임 사이의 거리에 반비례하여 값을 설정하였다.

$$t(i, j) = e^{-\left| \frac{i-j}{\sigma} \right|} \quad (2)$$

여기서 σ 는 한 동작에 대한 길이를 반영하는 값으로 우리는 수동으로 설정한다. 그림 2는 A라는 부분동작이 반복될 때 두 프레임의 거리에 따라 $t(i, j)$ 의 값을 보여주며, x축은 프레임을 나타내고 σ 는 150으로 설정하였다.

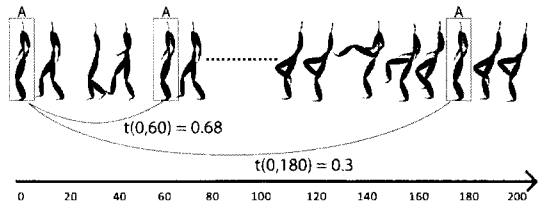


그림 2 프레임 시간축

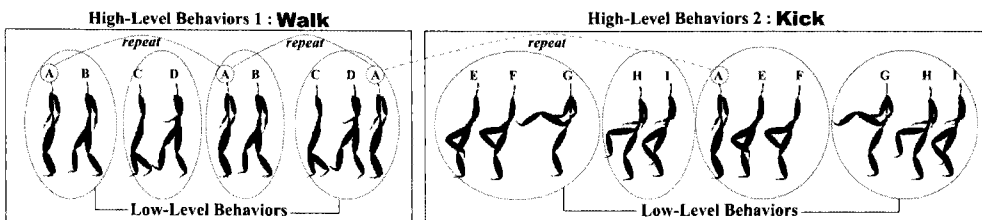


그림 1 모션 데이터

결론적으로 본 논문에서 이용한 그래프의 가중치는 다음과 같이 계산된다.

$$W(i, j) = w(i, j) \times t(i, j) \quad (3)$$

여기서 $w(i, j)$ 는 프레임 i 와 j 사이의 유사도를 나타낸다. $W(i, j)$ 는 이웃하는 프레임뿐만 아니라 $t(i, j)$ 에 의해 일정시간내의 모든 프레임을 반영하는 그래프의 가중치이며, 값이 높을수록 두 프레임이 하나의 동작에 포함될 가능성이 높다.

3. Normalized Cuts

우리는 구축된 그래프를 분할하기 위하여 같은 모션 내의 유사도 합을 최대화하고 다른 모션 사이의 유사도 합은 최소화할 수 있는 normalized cuts[9]를 이용하였다. 그래프 $G=(V,E)$ 를 2개의 부분 그래프로 분할한다고 가정할 때, 두 개의 부분 그래프 $G'=(V', E')$, $G''=(V'', E'')$ 는 $V' \cup V'' = V$ 와 $V' \cap V'' = \emptyset$ 를 만족한다. 이때, 두 부분 그래프 G' , G'' 를 연결하고 있는 간선들의 가중치의 합은 두 부분 그래프 사이의 유사도를 반영하며, 이를 그래프 이론에서 cut이라고 하며, cut은 다음과 같다.

$$\text{cut}(V', V'') = \sum_{i \in V', j \in V''} W(i, j) \quad (4)$$

그래프를 최적으로 분할하기 위해서는 cut의 값이 최소가 될 때 분할해야 한다. 그러나 이 방법은 항상 최적의 분할을 고려하지 못한다. 예를 들어 그래프에서 홀로

고립된 (멀리 떨어져있는) 정점의 경우 cut의 값이 최소 값을 가질 수 있기 때문에 잡음임에도 불구하고 분할되는 문제점을 가지고 있다. 이 방법을 모션 분할에 사용할 때 동작이 크거나 빠른 움직임일 가진 하이레벨 모션의 경우 하이레벨 모션을 이루는로우레벨 모션 사이의 유사도가 낮기 때문에 하나의 모션으로 분할되는 문제가 발생한다. 그림 3은 본 논문에서 제안한 모션 유사도 그래프에서 최소값만을 이용하여 분할한 결과의 예를 보여주며, 그림 3에서 보는 바와 같이 빠른 움직임을 가진 주먹을 뺀 모션 내에서 잘못 분할되는 결과를 보였다.

본 논문에서는 이러한 결점을 해결한 normalized cuts을 이용하였으며, normalized cuts은 다음과 같다[9].

$$N\text{-cut}(V', V'') = \frac{\text{cut}(V', V'')}{\text{assoc}(V', V')} + \frac{\text{cut}(V'', V')}{\text{assoc}(V'', V')} \quad (5)$$

여기서 $\text{assoc}(V', V) = \sum_{i \in V', j \in V} W(i, j)$ 는 집합 V' 에 속하는 모든 정점에서 V 에 속하는 모든 정점까지의 가중치 합을 의미한다. 따라서 N-cut값은 cut을 $\text{assoc}(V', V)$ 와 $\text{assoc}(V'', V)$ 로 정규화하기 때문에 홀로 떨어져 있는 정점을 가진 부분 그래프에서 최소값을 가지지 않는다[9]. 다음 식 (6)은 그래프 내부의 유사도를 가중치의 합으로 나타내는 식이다.

$$\text{Nassoc}(V', V'') = \frac{\text{assoc}(V', V')}{\text{assoc}(V', V)} + \frac{\text{assoc}(V'', V'')}{\text{assoc}(V'', V)} \quad (6)$$

$\text{assoc}(V', V')$ 와 $\text{assoc}(V'', V'')$ 는 V' 과 V'' 의 내부 정점간의 가중치 합이며, Nassoc값이 클수록 내부 정점간의 유사성이 크다. 식 (4)에 의해 $\text{cut}(V', V'')$ 은 $\text{assoc}(V', V) - \text{assoc}(V', V')$ 또는 $\text{assoc}(V'', V) - \text{assoc}(V'', V'')$ 로 표현할 수 있으며, 식 (7)에서 보는 바와 같이 N-cut과 Nassoc은 서로 역수 관계이다. 이는 N-cut의 값을 최소화하면 두 그래프 사이의 유사도는 최소화하며, 그래프 내부의 유사도는 최대화함을 뜻한다[9].

$$N\text{-cut}(V', V'') = 2 - \left(\frac{\text{assoc}(V', V')}{\text{assoc}(V', V)} + \frac{\text{assoc}(V'', V'')}{\text{assoc}(V'', V)} \right) = 2 - \text{Nassoc}(V', V'') \quad (7)$$

4. 실험 및 결과

본 실험에서 우리는 Carnegie Mellon University Graphics Lab의 모션 캡처 데이터를 이용하였으며[10], 하나의 모션 캡처 데이터는 걷기와 뛰기 등의 여러 동작으로 구성되어 있다. 실험의 속도를 향상시키기 위하여 초당 120프레임의 데이터를 30프레임으로 줄여서 이용하였으며, 제안된 방법을 분석하기 위해 기존의 오프라인 방법 중 하나인 GMM기반의 방법[1]과 온라인 방법 중 하나인 PCA기반의 분할 방법[1]을 비교하여 성능을 평가하였다. 그림 4는 실험에서 이용한 6개의 모션 캡처 데이터를 100프레임당 하나씩 뽑은 영상이며, 설명

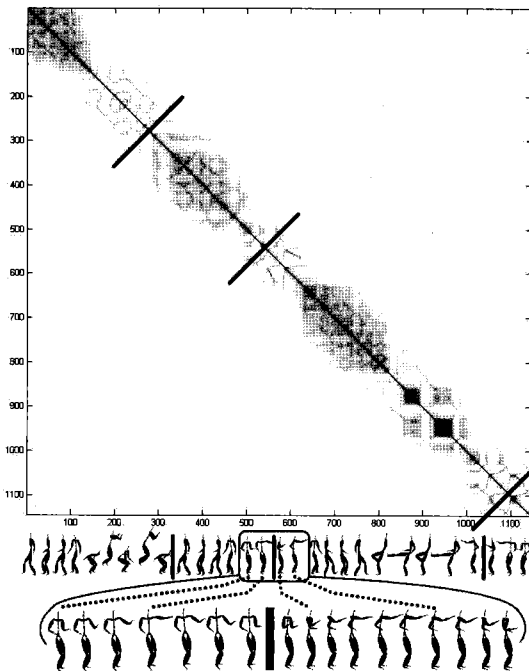
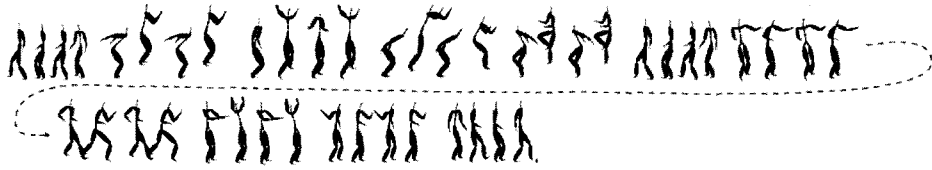


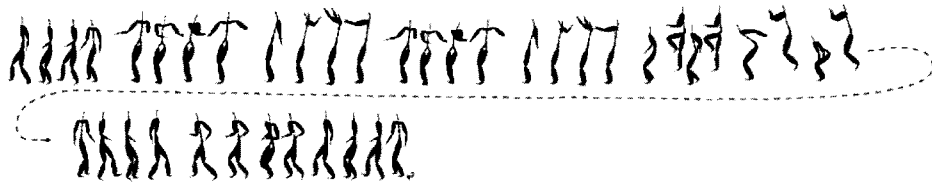
그림 3 모션 유사도 그래프에서 최소값만을 이용하여 분할할 때 발생하는 문제점



(a)



(b)



(c)



(d)



(e)



(f)

그림 4 논문에서 이용한 모션 캡처 데이터: (a) walking, jumps, kicks, and punches (86-01), (b) walking, jumping, jumping jacks, jumping on one foot, punching, chopping (86-05), (c) walking, swinging arms, stretching, jumping on one leg, and jumping (86-07) walking, sitting, looking, stand up (86-09), (e) walk around, sit, stand up, and running (86-10), (f) walking, stretching, walking and turning (86-11).

부분(caption)에 구성 동작과 모션 캡처 데이터[1]에서 제공하는 인덱스(index) 정보를 기술하였다.

그림 5는 6개의 모션 캡처 데이터의 유사도 그래프를 제안된 방법을 이용하여 분할한 결과와 사람이 직접 분할한 결과를 나타내며, 그래프의 x축과 y축은 프레임을 뜻한다. 제안된 방법은 normalized cuts을 기반으로 입력 캡처 데이터를 모션 별로 분할한다. Normalized cut은 기본적으로 하나의 그래프 G 를 두 개의 부분 그래프 G' 와 G'' 로 분할하는 방법이며, 두 개 이상의 부분 그래

프로 분할할 경우 분할될 개수가 사용자가 입력한 개수와 일치할 때까지 탐-다운(top-down)으로 반복 수행한다. 그림 6은 두 개의 부분 그래프로 분할하는 방식을 탐-다운으로 반복 수행하는 단계를 보여주며, 그림 6(b)에서 두 개의 부분 그래프 중 왼쪽 부분 그래프가 분할된 이유는 오른쪽 분할 결과보다 가중치의 합이 작기 때문이다. 여기서 분할된 하나의 부분 그래프는 하나의 하이레벨 모션을 뜻한다. 그림 5에서 설정한 클러스터의 수 K 는 각 모션 캡처 데이터의 분할될 동작의 수이며

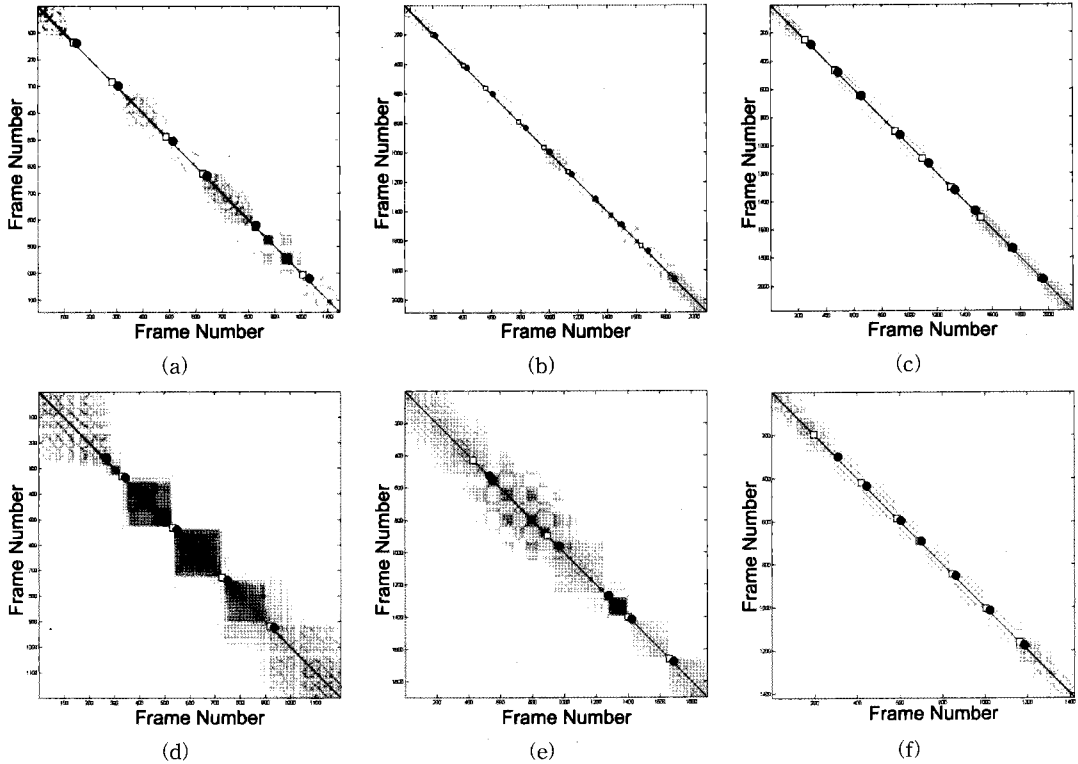


그림 5 Normalized cuts을 이용한 그래프 분할 결과: (a) 86-01(시간축의 길이=150, 클러스터의 수 K=7), (b) 86-05 ($\sigma=150$, K=11), (c) 86-07($\sigma=150$, K=10), (d) 86-09($\sigma=450$, K=6), (e) 86-10($\sigma=450$, K=6), (f) 86-11($\sigma=150$, K=8)

Graphics lab[10]에서 지정한 값으로 설정하였으며, σ 는 실험에서 가장 좋은 결과를 보이는 값으로 설정하였다. 제안된 방법에서 그래프를 만들 때 σ 값에 의해 그래프의 모양이 바뀌고, 결과 역시 크게 영향을 받는다. 이를 평가하기 위해 각 실험 데이터를 σ 의 값에 따라 Precision과 Recall를 이용하여 평가하였다. 그림 7은 Precision과 Recall에 대한 그래프를 보여주며, 여기서 Precision과 Recall의 식은 다음과 같다.

$$\text{Precision} = \frac{\text{reported correct cuts}}{\text{total number of reported cuts}} \times 100 \quad (8)$$

$$\text{Recall} = \frac{\text{reported correct cuts}}{\text{total number of reported cuts}} \times 100$$

그림 7의 a,b,c,f와 같이 한 동작의 길이가 짧을 경우 σ 값이 작을수록 정확도가 높아지는 반면 나머지 d,e와 같이 한 동작의 길이가 길 경우 σ 가 커질수록 정확도가 증가하는 것을 볼 수 있다. 결론적으로 매개변수 σ 는 하이레벨 모션의 평균 길이에 비례해서 설정할 때 가장 좋은 성능을 보였다.

그림 8은 실험에 이용한 총 6개의 모션 캡처 데이터를 Barbic 등[1]이 제안한 GMM, PCA기반의 방법과

본 논문에서 제안된 방법을 이용하여 분할한 결과를 보여준다. 그림 8의 x축은 프레임을 뜻하고 y축은 3가지의 알고리즘을 나타내며, 점선 사각형은 사람이 수동으로 분할한 영역을 나타낸다. 오프라인 방법인 GMM[1]은 특징공간에 표현된 특징값에 따라 성능이 크게 차이가 난다. 실험 데이터 중 하나인 86-01의 경우 확실하게 분할될 수 있는 특징값을 가지고 있기 때문에 좋은 성능을 보이지만, 대부분의 경우에는 정확한 분할개수보다 많이 분할되거나 적게 분할되는 문제점을 보였다. 온라인 방법인 PCA[1]는 그림 8의 d,e,f와 같이 정확한 분할개수보다 많이 분할되는 문제점이 발생하였으며, 이는 온라인 방법의 가장 큰 단점인 이웃하는 프레임 사이의 관계성만을 고려하여 지역적으로만 분할하기 때문이다. 반면 제안된 방법은 대부분 안정적인 분할결과를 보였으며, 그림 8의 분할결과에서 보는 바와 같이 GMM, PCA와 비교해서 정확한 성능을 보여주었다. 하지만 86-11과 같이 수동으로 설정한 σ 의 값보다 긴 동작이 캡처 데이터에 포함되어있을 경우 동작이 끝나기 이전에 분할되는 문제점을 가지고 있다. 그림 9는 각 모션캡처 데이터에서 GMM, PCA, 제안된 방법의 분할 성

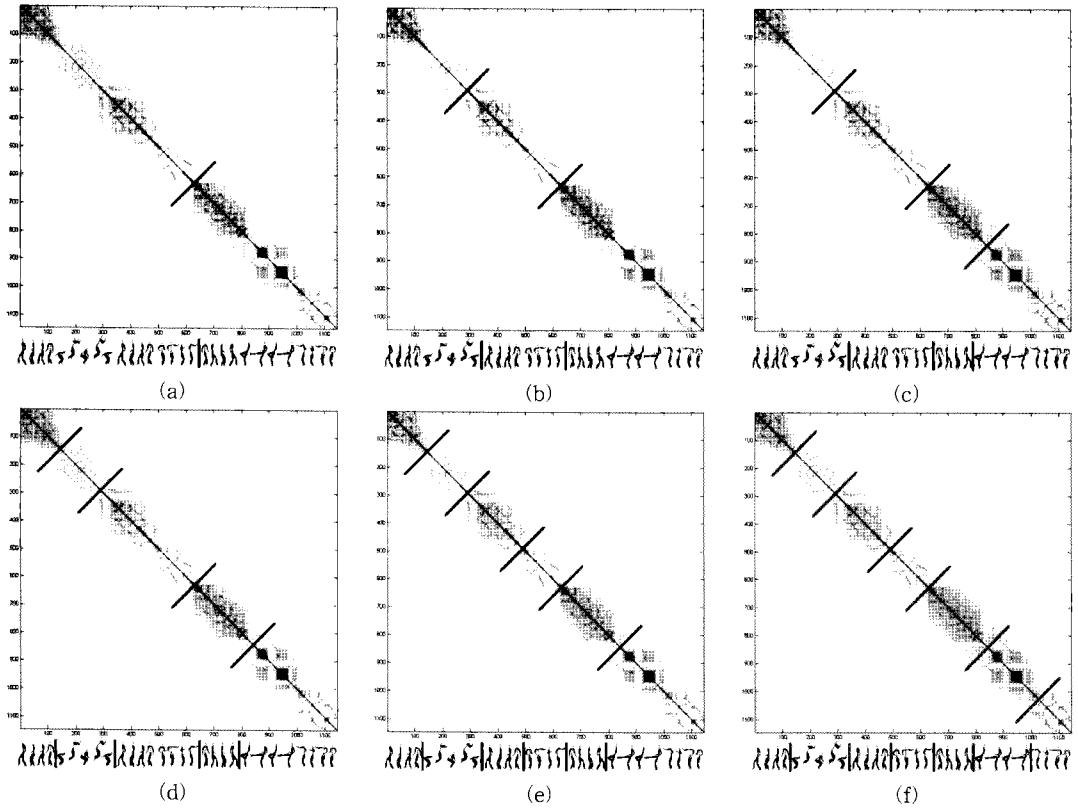


그림 6 그림 5(a)에 대한 normalized cuts을 이용한 다중 모션 분할의 단계별 결과

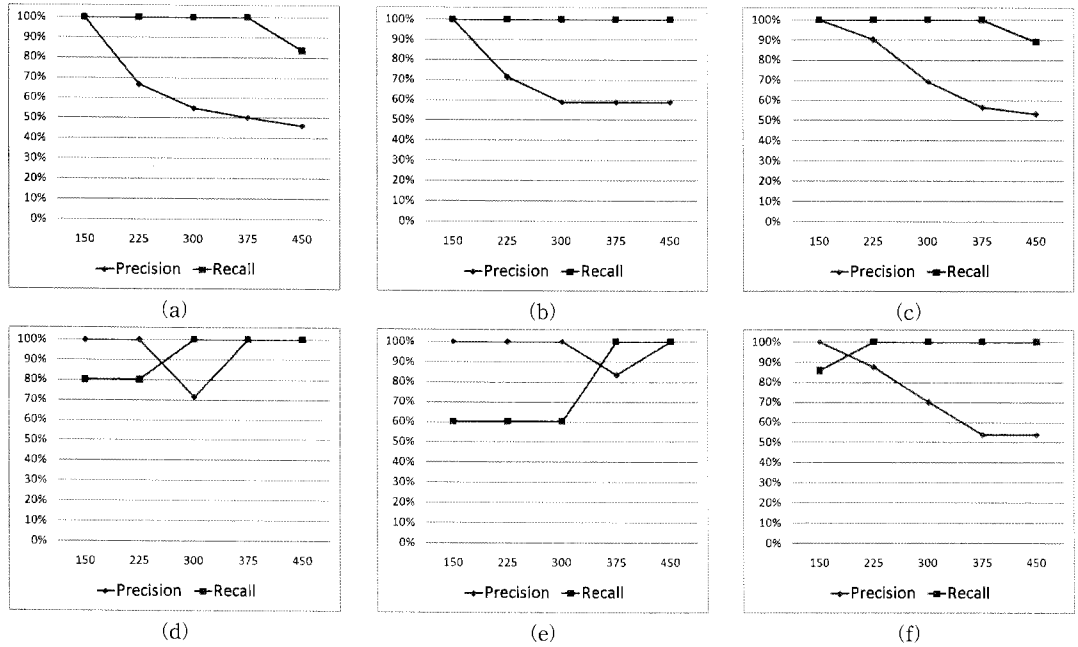


그림 7 시간축의 길이에 따른 Precision과 Recall평가: (a) 86-01(x축은 시간축 길이 σ , y축은 백분율), (b) 86-05, (c) 86-07, (d) 86-09, (e) 86-10, (f) 86-11

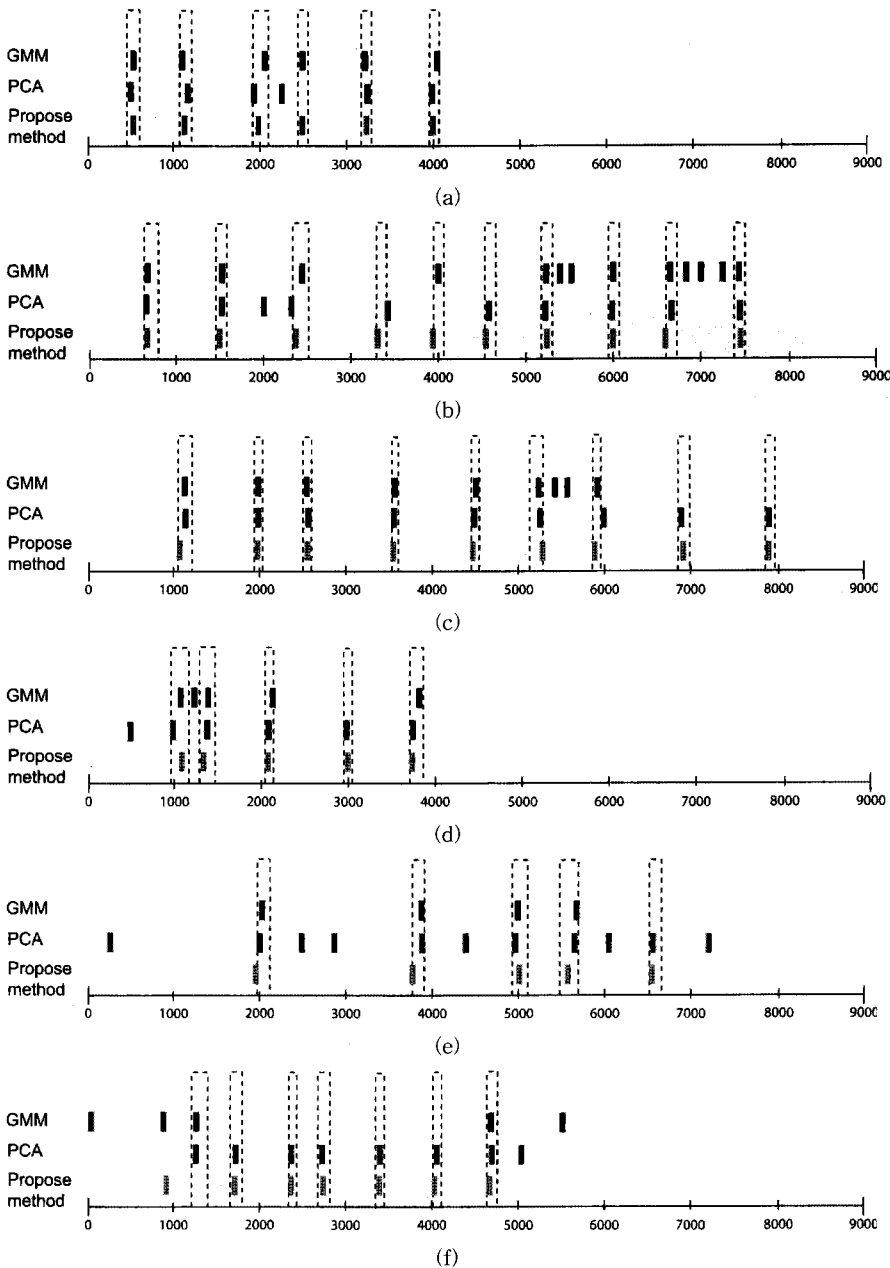


그림 8 GMM, PCA, 제안된 방법의 모션분할 결과: (a) 86-01(X축은 프레임, y축은 3가지 알고리즘, 점선 사각형은 정확한 분할의 범위), (b) 86-05, (c) 86-07, (d) 86-09, (e) 86-10, (f) 86-11

능 평가를 나타내며, 평가를 나타내는 요인으로 위에서 설명한 Precision과 Recall로 평가하였다.

표 1은 실험에서 이용한 모션 캡처 데이터 별 제안된 방법의 수행시간을 나타낸다. 각 데이터의 총 수행시간은 프레임 길이와 시간축 길이에 따라 크게 차이가 나며, 이는 그래프를 구축할 때 하나의 프레임에서 일정시

간내의 모든 프레임 사이의 유사도를 추가적으로 계산하기 때문이다. Normalized cuts의 수행시간(N-cut 시간)은 프레임의 수에만 영향을 받기 때문에 상대적으로 빠른 수행시간을 보인다.

그림 10은 제안된 방법을 이용해 개발된 모션분할 어플리케이션을 보여준다. A는 이웃하는 프레임 별로 모

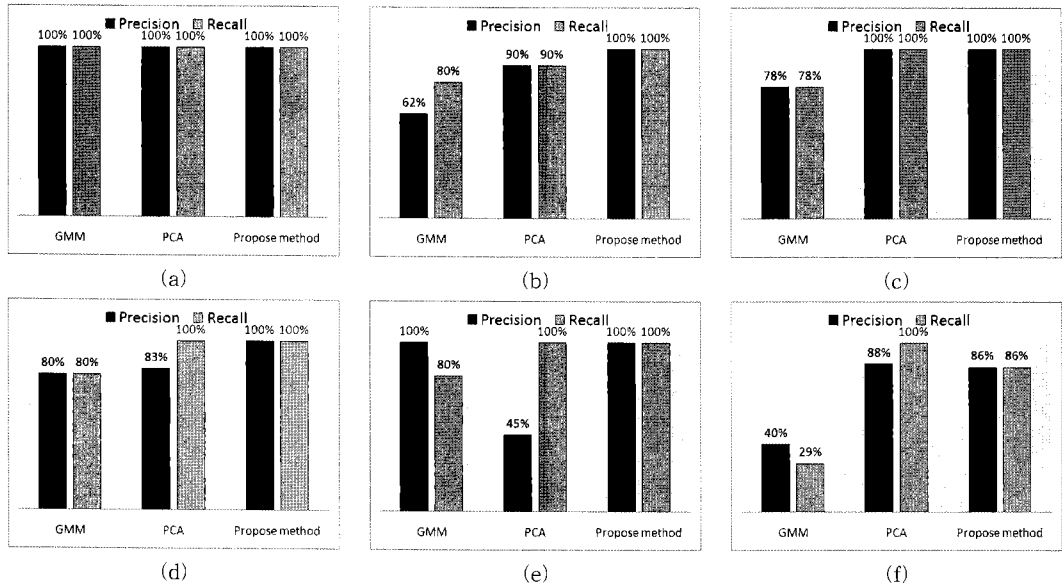


그림 9 GMM, PCA, 제안된 방법의 Precision과 Recall 평가: (a) 86-01(x축은 알고리즘, y축은 백분율), (b) 86-05, (c) 86-07, (d) 86-09, (e) 86-10, (f) 86-11

표 1 각 단계별 수행시간

모션 캡처 데이터	86-01	86-05	86-07	86-09	86-10	86-11
프레임	4579	8340	8702	4794	7583	5674
시간축 길이(σ)	150	150	150	450	450	150
그래프 구축 시간(sec.)	14	42.9	46.8	14.2	36.3	19.8
N-cut 시간(sec.)	1.2	1.79	1.8	1.5	1.74	1.3
총 수행시간(sec.)	15.2	44.69	48.6	15.7	38.04	21.1

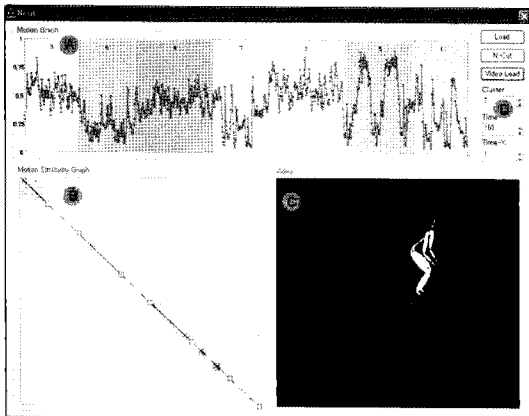


그림 10 모션 분할 프로그램

선의 유사도를 표현한 꺾은선 그래프와 색상 별로 분할된 영역을 표시하고 B는 시간축을 고려한 프레임 유사도 그래프이며, C는 모션이 분할된 뒤에 각각의 결과를 볼 수 있는 창이고 마지막으로 D에서 클러스터의 수(K)와 시간축의 값(σ)을 지정 한다.

5. 결론

본 논문에서는 온라인과 오프라인 방법을 통합한 그래프 기반의 캡처 데이터 분할 방법을 제안하였다. 캡처 데이터 분할 방법으로 이웃하는 각 프레임 사이의 유사도뿐만 아니라 시간축을 기반으로 하는 일정시간내의 프레임들의 유사도를 모두 고려하였으며, 분할된 같은 프레임내의 유사도 합을 최대화 하고 다른 프레임간의 유사도 합은 최소화 할 수 있는 normalized cuts을 이용해 모션분할을 하였다. 오프라인 방법중 하나인 GMM은 특징값에 따라 분할된 결과의 정확도가 크게 달라졌으며, 대체적으로 낮은 정확도를 보였다. 온라인 방법인 PCA의 분할결과는 이웃하는 프레임 사이의 유사도만을 고려하기 때문에 몇 개의 실험 데이터에서 정확한 분할개수보다 많이 분할되는 문제점을 보였다. 본 논문에서 제안한 normalized cuts는 GMM과 PCA보다 정확도가 높은 안정적인 실험결과를 보였다.

제안된 방법의 문제점으로 클러스터의 수와 σ 값을 수동으로 설정하는 불편함과 한 동작이 σ 의 값보다 길어

질 경우 한 하이레벨 모션이 끝나기 전에 모션이 분할 되는 점이다. 이를 해결하기 위한 방법을 연구할 예정이며, 앞으로 모션의 동작을 검색하는 방법에 대해서도 연구를 계속 이어갈 것이다.

참 고 문 헌

- [1] J. Barbic, A. Safonova, J.-Y. Pan, C. Faloutsos, J. K. Hodgins and N. S. Pollard, "Segmenting Motion Capture Data into Distinct Behaviors," Proceedings of ACM International Conference on Graphics Interface, Vol. 62, pp. 185-194, 2004.
- [2] T. Kim, S. Park, and S. Shin, "Rhythmic-Motion Synthesis Based on Motion-Beat Analysis," ACM Transactions on Graphics, Vol. 22, pp. 392-401, 2003.
- [3] C. Lee and A. Elgammal, "Human Motion Synthesis by Motion Manifold Learning and Motion Primitive Segmentation," Proceedings of Articulated Motion and Deformable Objects on International Conference, Lecture Notes in Computer Science, Vol. 4069, pp. 464-473, 2006.
- [4] Y. Sakamoto, S. Kuriyama, and T. Kaneko, "Motion Map: Image-based Retrieval and Segmentation of Motion Data," Proceedings of Eurographics/ACM SIGGRAPH Symposium on Computer Animation, pp. 259-266, 2004.
- [5] A. Fod, M. J Mataric, and O. Jenkins, "Automated Derivation of Primitives for Movement Classification," Autonomous Robots, Vol. 12, No. 1, pp. 39-54, 2002.
- [6] T. Kwon and S. Shin "Motion Modeling for On-Line Locomotion Synthesis," Proceedings of ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp. 29-38, 2005.
- [7] T. Yamasaki and K. Aizawa, "Motion Segmentation and Retrieval for 3D Video Based on Modified Shape Distribution," EURASIP Journal on Advances in Signal Processing, Vol. 2007, No. 2, pp. 1-11, 2007.
- [8] D. Bouchard and N. Badler, "Semantic Segmentation of Motion Capture Using Laban Movement Analysis," Proceeding of Intelligent Virtual Agents on International Conference, Lecture Notes in Computer Science, Vol. 4722, pp. 37-44, 2007.
- [9] J. Shi and J. Malik, "Normalized cuts and Image Segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 8, pp. 888-905, 2000.
- [10] <http://mocap.cs.cmu.edu>



윤 성 주

2008년~현재 숭실대학교 IT대학 미디어학과 석사과정. 관심분야는 컴퓨터비전, 패턴인식, Interactive Contents

박 안 진

정보과학회논문지 : 소프트웨어 및 응용 제 35 권 제 9 호 참조

정 기 철

정보과학회논문지 : 소프트웨어 및 응용 제 35 권 제 9 호 참조