

Normalized Cuts을 이용한 그래프 기반의 모션 분할

윤성주⁰ 박안진 정기철
송실대학교 IT대학 미디어학과
{yuyu04, anjin, kcjung}@ssu.ac.kr

Graph-based Motion Segmentation using Normalized Cuts

Sungju Yun⁰ Anjin Park Keechul Jung
Department of Digital Media, College of IT Soongsil University

요 약

모션 캡처 장비는 사람의 자연스러운 행동이나 동작 정보를 정밀하게 얻기 위해 널리 사용되며, 영화나 게임과 같은 콘텐츠에서 자주 활용되고 있다. 하지만 모션 캡처 장비가 고가이기 때문에 한번 입력받은 데이터를 모션별로 분할하고 상황에 맞게 재결합하여 사용할 필요가 있으며, 입력 데이터를 모션별로 분할하는 것은 대부분 수동으로 이루어진다. 이 때문에 캡처된 데이터를 자동으로 분할하기 위한 연구들이 다양하게 시도되고 있다. 기존의 연구들은 크게 전역적 특성에 대한 고려없이 이웃하는 프레임만을 고려하는 온라인 방식과 데이터를 전역적으로 고려하나 이웃하는 프레임 사이의 관계를 고려하지 않는 오프라인 방식으로 나누어진다. 본 논문에서는 온라인과 오프라인 방식을 병합한 그래프 기반의 모션 분할 방법을 제안한다. 분할을 위해 먼저 모션데이터를 기반으로 그래프를 생성하며, 그래프는 이웃하는 각 프레임사이의 유사도뿐만 아니라 시간축을 기반으로 일정시간내의 프레임들의 유사도를 모두 고려하였다. 이렇게 생성된 그래프를 분할하기 위해 분할된 모션내의 유사도 합을 최소화하고 각 모션간의 유사도는 최대화할 수 있는 normalized cuts을 이용하였다. 실험에서 제안된 방법은 기존의 오프라인 방식 중 하나인 GMM과 온라인 방식 중 하나인 국부최소값 분할 방법보다 좋은 결과를 보였으며, 이는 각 프레임 사이의 유사도뿐만 아니라 일정시간내의 유사도를 전역적으로 고려하기 때문이다.

1. 서 론

모션 캡처 장비는 사람의 자연스러운 행동이나 동작 정보를 정밀하게 얻기 위해 널리 사용되며, 영화나 게임과 같은 콘텐츠에 자주 활용되고 있다. 하지만 모션 캡처 장비가 사용하기 불편하고 상당히 고가이기 때문에 한번 입력받은 데이터를 걷기, 뛰기와 같은 모션별로 분할하고 상황에 맞게 재결합하여 많이 사용하며, 대부분의 경우 입력받은 데이터를 사람이 수동으로 분할하였다. 이 때문에 최근 모션을 자동으로 분할하기 위한 다양한 방법들이 연구되고 있다.

입력 데이터를 모션별로 분할하는 방법은 크게 온라인과 오프라인으로 분류한다. 온라인은 순차적으로 들어오는 데이터를 이웃하는 프레임의 유사도 차이를 기반으로 분할하는 방법이다. Yamasaki와 Aizawa[1]는 3D Video에서 동작들에 대한 유사도를 포함한 그래프의 지역 최소점을 기준으로 모션 데이터를 분할하였으며, Kwon과 Shin[2]은 걷는 동작의 합성을 위해 사람의 무게중심점의 궤도와 걸음걸이의 패턴을 이용하여 모션 데이터를 분할하였다. Barbic 등[3]은 PCA를 이용하여 모션 데이터를 저차원으로 줄이고 각 프레임간의

유사도 변화에 따라 분할하는 방법을 제안하였다. 온라인 방식의 장점은 모션을 프레임 단위로 읽어와 분할하기 때문에 메모리에 모든 모션 데이터를 가져올 필요가 없으며, 이 때문에 메모리를 줄일 수 있다. 단점은 동작을 분류할 때 이웃하는 프레임의 유사도만을 고려하기 때문에 지역 최소화에 빠져 너무 작은 단위의 모션으로 나누는 문제가 발생한다.

오프라인 방법은 특징공간에서 각 프레임의 전역적 특징들을 분석하는 방법으로 K-means와 GMM을 주로 사용하였다. Barbic 등[3]은 클러스터 방식인 GMM을 이용하여 모션을 분할하였으며, Fod 등[4]과 Kim 등[5]은 K-Means를 이용한 모션 분할 방법을 제안하였다. Sakamoto 등[6]은 SOM을 사용하여 모션 맵(Motion Map)을 만들어 동작을 분류하고 검색하는 방법을 제안하였다. 오프라인 방식은 동작의 전역적 특징을 고려하여 분할할 수 있는 장점이 있다. 그러나 전체의 모션 데이터를 메모리로 가져와서 분할하기 때문에 사용하는 메모리가 커지며, 이웃하는 프레임 사이의 유사도를 전혀 고려하지 않는다.

분할된 모션들은 크게 로우레벨(Low-Level)과 하이레벨(High-Level)동작으로 나눌 수 있다. 로우레벨

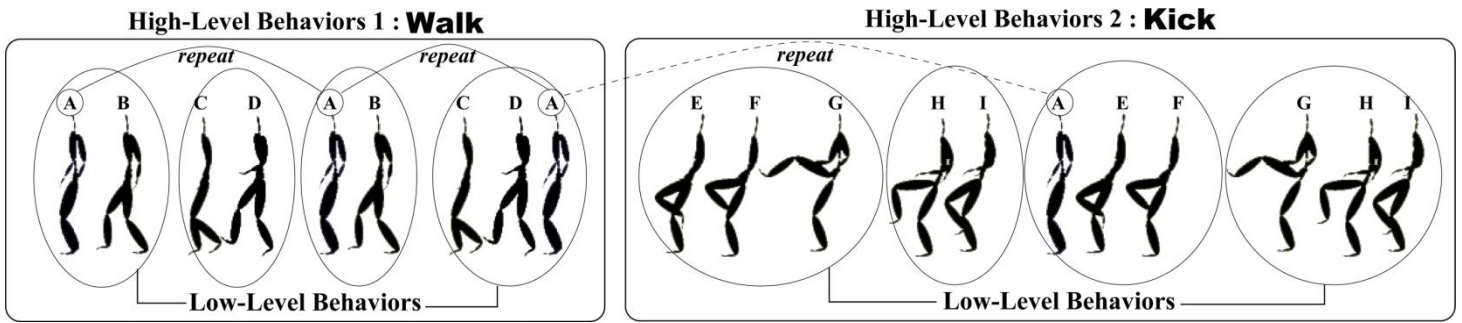


그림 1. 모션 데이터.

동작은 프리미티브(primitive)라고도 말하며, 그림 1에서 타원으로 표시된 것처럼 차기 동작에서 발을 올리고 내리는 각각의 동작들을 의미한다. 하이레벨 동작은 많은 로우레벨 동작들이 모여서 하나의 동작으로 표현된 것으로 사람이 생각하는 뚜렷한 하나의 동작으로 볼 수 있으며, 예를 들어 걷기, 뛰기, 차기 같은 동작들을 말한다. 로우레벨 동작은 온라인 방식에서 하이레벨 동작은 오프라인 방식에서 주로 분할한다.

본 논문에서는 모션데이터를 하이레벨 동작으로 분할하기 위해 온라인과 오프라인을 병합한 그래프 기반의 모션 분할 방법을 제안한다. 그래프를 구축할 때 이웃하는 프레임의 유사도 뿐만 아니라 일정 시간내의 프레임 사이의 유사도를 가중치로 설정하였으며, 이는 이웃하는 프레임만을 고려하는 온라인방식과 모든 프레임을 고려하는 오프라인방식을 적절히 병합한 방식이다. 그림 1에서 보는 바와 같이 구축된 그래프에서 하이레벨 동작으로 분할할 때 걷는 동작에서 부분동작 A가 차기 동작에서도 반복되는(repeat) 경우가 생기며, 이때 잘못된 분할결과가 나올 수 있다. 이를 해결하기 위한 방법으로 프레임의 거리에 따라 유사도가 감소하는 시간축을 고려하였으며, 그래프를 분할할 때 사용하는 알고리즘으로 같은 프레임내의 유사도 합을 최대화하고 다른 프레임간의 유사도 합을 최소화 할 수 있는 normalized cuts[7]을 이용하였다. 그래프를 구축하는데 사용한 특징값은 3차원 좌표에 상관없이 각 프레임에 대한 특징값만을 표현하기 위하여 중앙에 있는 관절 Hip에서의 각 관절과의 거리의 집합을 이용하였으며, 프레임당 총 37차원의 데이터를 갖는다. 실험에서 기존의 오프라인 방식중 하나인 GMM과 온라인 방식중 하나인 국부 최소값 분할 방법 보다 좋은 결과를 보였으며, 이는 각 프레임 사이의 유사도뿐만 아니라 일정시간 내의 유사도를 전역적으로 고려하기 때문이다.

2장에서는 모션 분할에 사용할 특징값과 프레임 유사도 그래프에 대하여 설명한다. 3장에서는 본 논문에서 사용하는 알고리즘인 normalized cuts에 대해 설명하고 4장에서는 normalized cuts으로 자동 분할한

결과와 기존의 방법과의 성능평가를 하며, 5장에서는 최종 결론과 앞으로의 연구계획에 대해서 논의한다.

2. 제안된 기법

2.1 특징 값

실험에서 사용한 모션 캡처 데이터는 1초에 120개 프레임으로 이루어져 있으며, 각 프레임마다 114차원의 데이터를 가지고 있고 114차원의 데이터는 38개 관절의 3차원 좌표이다. 본 논문에서 좌표에 상관없이 각 프레임의 특징값을 표현하기 위해서 중앙에 있는 관절 Hip에서 각 관절사이의 거리를 이용하였으며 k번째 프레임의 특징값은 다음과 같이 정의할 수 있다.

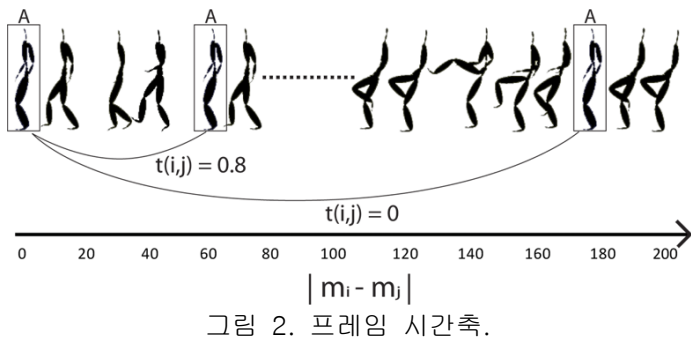
$$F^k = \{P_1, P_2, \dots, P_{37}\} \quad (1)$$

여기서 P_n 는 Hip과 n번째 관절사이의 거리를 나타내며 유클리드안 거리를 이용하였다. 37은 총 관절 38개에서 Hip 관절을 뺀 개수이다. 결론적으로 본 논문에서 사용한 특징값은 각 프레임 당 37차원의 데이터를 가진다.

2.2 시간축을 고려한 프레임 유사도 그래프

본 논문에서 우리는 그래프 기반의 모션 분할 방법을 제안하며, 이를 위한 첫 번째 단계로써 각 프레임의 특징값을 기반으로 가중치 그래프 $G=(V, E)$ 를 구축한다. 그래프에서 정점 V 는 프레임의 집합이며, 간선 E 는 각 프레임 사이의 연결을 집합으로 나타낸 것이다.

그림 1은 실험에서 사용한 모션 데이터의 일부를 보여주며, 걷는 동작(High Level Behavior 1)과 차는 동작(High Level Behavior 2)으로 구성되어 있다. 그림 1에서 보는 바와 같이 같은 동작이지만 두 프레임 사이의 유사도가 높지 않음을 알 수 있다. 예를 들어 발차는 동작은 동작이 크기 때문에 각 프레임 사이의 유사도가 걷기 동작에 비해 낮다. 하지만 각각의 동작위에 표시된 숫자와 같이 한 동작안에 반복되는 부분동작이 있으며, 우리는 이점을 이용하여 그래프를



구축할 때 이웃하는 프레임뿐만 아니라 주위 일정 범위내의 프레임을 같이 고려하였다. 다음 식 2는 본 논문에서 사용할 그래프에 가중치를 계산하는 식이다.

$$W(i, j) = w(i, j) \times t(i, j) \quad (2)$$

여기서 $w(i, j)$ 는 프레임 i 와 j 사이의 유사도를 나타내며, 유사도가 낮을수록 두 프레임이 하나의 동작에 포함될 가능성이 높다. 그림 1에서 설명한 것처럼 전체 프레임에서 생기는 반복문제를 피하기 위해 일정 시간내의 프레임 사이의 유사도만을 고려해야 하며, 이를 위해 우리는 $t(i, j)$ 항을 추가하였다. $t(i, j)$ 의 계산 방법은 식 3과 같이 표현되며, 두 프레임 사이의

거리에 반비례하여 값을 설정하였다.

$$t(i, j) = e^{-\left\| \frac{m_i - m_j}{\sigma} \right\|} \quad (3)$$

여기서 m_i 와 m_j 는 전체 프레임에서 각 프레임 i 와 j 의 순서를 나타내며, 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 보여준다. 본 논문에서는 두 항의 곱으로 가중치를 설정하며, 이는 프레임의 거리에 따라 유사도가 감소하고, 모든 프레임이 아닌 일정 시간 내의 프레임 사이의 유사도만을 고려한다.

3. Normalized Cuts

우리는 그래프를 분할하는 기술로 normalized cuts[7]을 이용한다. 입력 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 집합 G' , G'' 를 연결하고 있던 간선들의 가중치의 합은 두 부분 그래프 사이의 유사도를 반영하며, 뿐만 아니라 두 개의 부분 집합의 차이점도 반영한다. 이를 그래프 이론에서 cut이라고

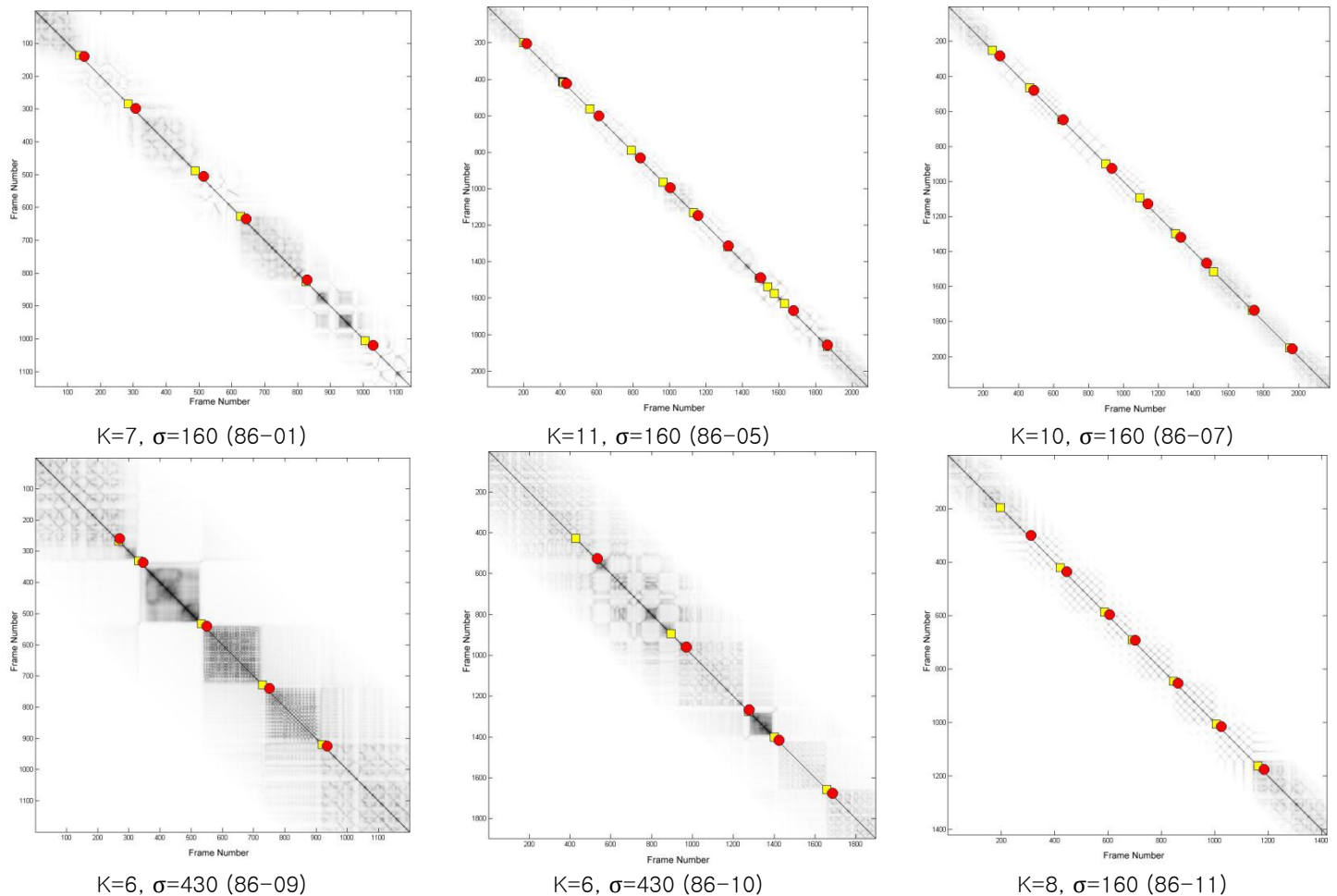


그림 3. 실험 결과.

한다. cut은 다음과 같다.

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

그래프를 최적으로 분할하기 위해서 cut의 값이 최소가 될 때 분할해야 한다. 그러나, 이 방법은 항상 최적의 분할을 고려하지 못한다. 예를 들어 그래프에서 홀로 떨어져있는 정점에서 cut이 최소값을 가지기 때문에, 잡음이지만 분할되는 문제가 있다. 이러한 결점을 해결한 normalized cuts[7]을 사용하였으며, normalized cuts은 다음과 같다.

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

여기서 $\text{assoc}(V', V) = \sum_{i \in V', j \in V} W(i, j)$ 는 집합 V' 에 속하는 모든 정점에서 그래프에 존재하는 모든 정점까지의 가중치 합을 의미한다. N-cut의 값을 $\text{assoc}(V', V)$ 와 $\text{assoc}(V'', V)$ 로 정규화하기 때문에 홀로 떨어져 있는 정점을 가진 부분 그래프에 대해 최소값을 가지지 않는다. 다음 식 6은 그래프 내부의 연관 정도를 나타내는 식이다.

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

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

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

4. 실험

실험에 사용된 모션 캡처 데이터는 Carnegie Mellon 대학교 Graphics Lab의 모션 캡처 데이터를 사용하였으며[3], 걷기와 뛰는 동작 등의 동작으로 구성된 하나의 긴 모션 데이터이다. 실험의 속도를 향상시키기 위하여 초당 120개의 프레임을 30개 프레임으로 줄여서 사용하였다. 제안된 방법을 분석하기 위해 기존의 오프라인 방식인 GMM과 온라인 방식인 국부최소값 분할 방식을 비교하여 성능을 평가한다.

그림 3은 각각 총 6개의 긴 모션 시퀀스 데이터를 normalized cuts을 이용해 그래프에서 분할된 결과와 사람이 직접 분할된 결과를 나타낸다. 그래프에서 가로축과 세로축은 프레임의 수를 나타내며 원은 사람이 직접 분할한 결과를 나타내고 사각형은 normalized cuts의 결과를 나타낸다. 본 논문에서

사용자로부터 입력받는 매개변수는 한 동작에 대한 길이를 반영하는 σ 와 클러스터의 수를 뜻하는 K이며, 실험에서 가장 좋은 결과를 보이는 값으로 σ 를 설정하였고, 클러스터 수는 Graphics lab에서 각 모션 캡처 데이터에 지정한 동작의 수로 설정하였다.

그림 4는 실험에 사용된 총 6개의 모션 캡처 데이터를 이용해 국부최소값, GMM, N-Cut의 분할 성능 평가를 나타낸다. 평가를 나타내는 요인은 2가지로 Precision과 Recall로 평가 하였으며, 다음과 같다.

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

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

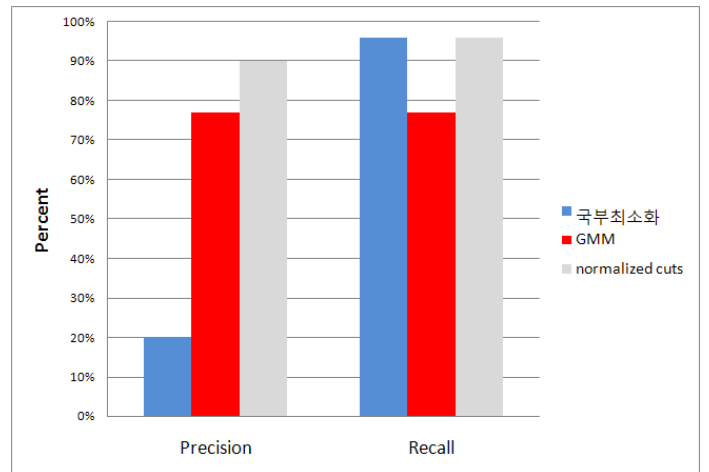


그림 4. 국부 최소화, GMM, normalized cuts의 Precision, Recall의 평가.

다음 그림 5는 normalized cuts을 사용해 개발된 어플리케이션을 보여준다.

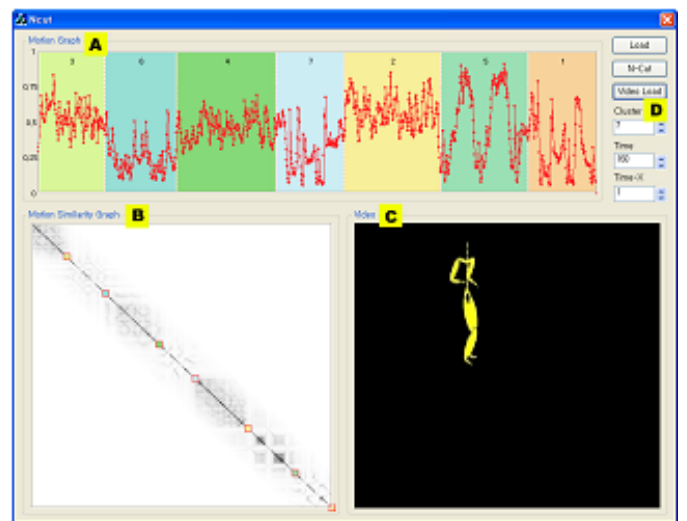


그림 5. 모션 분할 프로그램.

그림 5에서 A는 이웃하는 프레임 별로 모션의

유사도를 표현한 꺾은선 그래프와 분할된 영역을 표시하고 B는 시간축을 고려한 프레임 유사도 그래프이며, C는 모션이 분할 된 뒤에 각각의 결과를 볼 수 있는 창이고 마지막으로 D에서 클러스터의 수와 시간축의 값을 지정 할 수 있다.

5. 결 론

본 논문에서는 온라인과 오프라인 방식을 병합한 그래프 기반의 모션 데이터 분할 방법을 제안하며, 자동으로 모션분할을 위한 어플리케이션 프로그램을 제작하였다. 모션 데이터 분할 방법으로 이웃하는 각 프레임 사이의 유사도뿐만 아니라 시간축을 기반으로 하는 일정 시간내의 프레임들의 유사도를 모두 고려하였으며, 분할된 프레임내의 유사도 합을 최소화 하고 각 프레임간의 유사도는 최대화 할 수 있는 normalized cuts을 이용해 모션분할을 하였다. 온라인 방식인 국부 최소화의 실험결과는 Recall은 96%로 높지만 Precision은 20%로 낮은 결과를 보였고 오프라인 방식중 하나인 GMM은 Precision과 Recall 모두 77%로 낮은 결과를 보였다. 본 논문에서 제안한 normalized cuts는 Precision은 90%와 Recall은 96%로 모두 안정적인 높은 실험결과를 보였다.

제안된 방법의 문제점으로 클러스터의 수와 σ 값을 수동으로 설정하는 불편함과 한 동작이 σ 의 값보다 길어질 경우 한 하이레벨 동작이 끝나기 전에 모션이 분할되는 점이다. 이를 해결하기 위한 방법을 연구할 예정이며, 앞으로 모션의 동작을 검색하는 방법에 대해서도 연구를 계속 이어갈 것이다.

Acknowledgment: 이 논문은 한국과학재단(KOSEF)의 특정기초연구 No.(R01-2006-000-11214-0)의 지원에 의하여 연구되었음

참고문헌

- [1] 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.
- [2] T. Kwon and S. Shin "Motion Modeling for On-Line Locomotion Synthesis," Proceedings of 2005 ACM Siggraph/Eurographics Symposium on Computer Animation, pp. 29-38, 2005.
- [3] 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 Graphics Interface, Vol. 62, pp. 185-194, 2004.
- [4] 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.
- [5] T. Kim, S. Park, and S. Shin "Rhythmic-Motion

Synthesis Based on Motion-Beat Analysis," ACM Transactions on Graphics, pp. 392-401, 2003.

[6] 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.

[7] 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.