논문-00-5-1-11

칼라 및 모션 특징 기반 비디오 씬 분할 기법

송 창 준*,**, 고 한 석**, 권 용 무*

Video Scene Segmentation Technique based on Color and Motion Features

Chang-Jun Song*,**, Hanseok Ko**, and Yong-Moo Kwon*

요 약

기존의 비디오 구조화 기법은 주로 샷 또는 샷 그룹 레벨에서 이루어져 왔다. 그러나 이러한 샷 레벨 구조는 사용자에게 의미(semantics)를 충분히 전달할 수 없는 단점이 있다. 이런 단점을 극복하기 위해, 최근 샷 보다 상위 레벨 구조인 비디오 씬 분할에 관한 연구가 진행되고 있다. 본 논문에서는 이러한 샷 레벨 구조의 단점을 극복하기 위해서 칼라와 모션 특징을 기반으로 한 비디오 씬 분할 기법을 제안한다. 샷 내의 다양한 칼라 분포를 반영하기 위해서 각 샷을 sub-shot으로 재분할하고, 이를 이용해 대표 프레임을 추출한다. 샷 내의 모션 특징은 MPEG-1 비디오 내의 모션 벡터를 이용한다. 유사한 컨텐트를 가지고 있는 샷을 찾기 위해서 탐색 구간내의 모션 특성을 반영한 적응적 가중치를 칼라와 모션 특징에 적용한다. 실험 결과 비교를 통해 씬의 과다 분할이나 의미 반영 면에서 기존의 씬 분할 기법보다 우수함을 보였다. 제안된 기법은 비디오를 의미 있는 계층 구조로 분할해서, 사용자에게 의미를 반영하는 씬 단위로의 브라우징이나 검색을 가능케 한다.

Abstract

The previous video structuring techniques are mainly limited to shot or shot group level. However, the shot level structure couldn't provide semantics within a video. So, researches on high level structuring are going on for getting over the drawbacks of shot level structure, recently. To overcome the drawbacks of shot level structure, we propose video scene segmentation technique based on color and motion features. For considering various color distribution, each shot is divided into sub-shots based on color feature. A key frame is extracted from each sub-shot. The motion feature in a shot is extracted from MPEG-1 video's motion vector. Moreover adaptive weights based on motion's property in search range are applied to color and motion features. The experiment results of proposed technique show the excellence in view of the over-segmentation and the reflection of semantics, comparing with those of previous techniques. The proposed technique decomposes video into meaningful hierarchical structure and provides video browsing or retrieval based on scene.

I. 서 론

최근 들어서 저장 매체의 가격 하락, 향상된 압축 기법, 그리고 인터넷 사용의 증가로 인해 디지털 비디오의 사용은 놀랄만한 속도로 증가되고 있다[1][2][3]. 그러나 디지털 비디오는 데이터 양이 방대하고, 특정한 구조가 없기 때문에, 사용자가 브라우징 또는 검색을 위해 비디오에 접근하는 일은 그다지 쉬운 일이 아니다.

사용자로 하여금 효율적이면서 쉽게 비디오에 접근할 수 있게 하기 위해서 가장 근본적이고 중요한 작업은 비디오로부터 구조를 추출하는 비디오 분석과정이다. 초기의

* 한국과학기술연구원 영상미디어연구센터,
Imaging Media Research Center, Korea Institute of Science and Technology

** 고려대학교 전자공학과
Dept. of Electronic Engineering, Korea University

비디오 분석은 주로 샷 경계 검출, 대표 프레임 추출, 샷 그룹(shot group) 추출에 초점을 맞추어 왔다[4][5][6]. 비디오 시퀀스에 대해 우선 샷 경계 검출을 하고, 각각의 샷 내에서 대표 프레임을 추출한다. 여기서 샷은 검색 또는 브라우징의 기본 단위가 되며, 이때 각 샷의 내용은 대표 프레임을 이용해 나타낸다. 여기에 각 샷의 대표 프레임을 이용해서 칼라 히스토그램과 같은 특징을 바탕으로 유사한 샷끼리 묶어서 사용자로 하여금 비디오에 접근할 수 있도록 제시하였다.

그러나 이러한 샷 또는 샷 그룹 레벨의 비디오 구조화로는 비디오내의 사건이나 의미를 충분히 반영하지 못한다는 단점이 있기 때문에 최근 몇 년 전부터 이런 점을 극복하려는 연구가 진행되고 있다. 이들은 공통적으로 모두 비디오 내의 의미를 반영하기 위한 비디오 씬 분할에 초점을 맞추고 있다. 기존의 씬 분할 기법은 칼라 특징만을 이용하기 때문에, 특히 비디오내의 화면진행이 빠르게 일어나는 동적인 구간에서는 좋은 씬 분할 결과를 보여주지 못했다. 동적인 구간에서는 실제 하나의 씬이 여러 씬으로 나뉘는 씬 과다 분할 현상이 일어나기 때문이다. 또한 비디오 씬 분할 시, 모든 샷 시퀀스에 대해서 유사도를 측정하기 때문에, 많은 비디오에 적용 시 많은 시간이 요구되는 단점이 있다.

따라서 본 논문에서는 아래와 같은 내용에 초점을 맞춘다.

- 칼라 특징뿐만 아니라 모션 특징을 복합적으로 고려함으로써, 비디오 내의 의미를 충실히 반영하고 씬 과다 분할 현상을 줄이려고 하였다.
- 비디오 씬 분할 시 Improved Overlapping Links(IOL) 기법을 이용해서 연산 시간을 단축하려고 하였다.
- 비디오 브라우징 및 검색에 이용하기 위해서 샷과 씬의 계층적 구조를 추출하려고 하였다.

본 논문의 구성은 다음과 같다. II장에서는 기존의 씬 분할 기법에 대한 고찰을, III 장에서는 칼라와 모션을 이용한 비디오 씬 분할 기법, IV장에서는 실험 결과 및 고찰을, V장에서는 결론을 맺는다.

II. 기존의 씬 분할 기법

기존에는 주로 샷 경계 검출이라는 용어 대신에 장면 전환 검출(Scene Change Detection)의 용어가 널리 사용되어 왔다. 그러나 이 개념은 본 논문에서 추출하려는 씬(scene)의 개념과는 거리가 있다. 따라서, 본 논문에서는 비디오 구조화 시 사용되는 용어의 의미를 다음과 같이 기술한다.

- 대표 프레임(key frame) : 각 샷의 내용(content)을 대표할 수 있는 프레임을 의미한다.
- 서브 샷(sub-shot) : 일정 프레임(시간) 이상의 연속적인 프레임으로 구성된 칼라 분포가 유사한 그룹을 의미한다. 각 샷의 모션이나 칼라의 변화에 따라서, sub-shot은 하나의 샷 당 한 개 이상이 될 수 있다.
- 샷(shot) : 하나의 카메라에 의해서 연속적으로 기록된 프레임을 샷이라 한다.
- 샷 그룹(shot group) : 샷과 씬의 중간 단계로서 유사한 샷들의 모임을 의미한다.
- 씬(scene) : 시간적으로 이웃해 있는 샷들로 이루어져 있고, 이들이 내용 면에서 서로 관련이 있을 때, 이런 샷들의 모임을 씬이라 한다.

샷이 비디오를 구성하는 기본 단위인 반면에, 씬은 사용자에게 비디오의 내용 또는 스토리를 전달할 수 있는 구조이다. 비디오 내에서 하나의 씬은 같은 장소(배경)에서 일어나거나 같은 주인공(얼굴, 의상 등)이 나오기 때문에 씬 내부에서는 유사한 컨텐트(칼라, 모션, 오브젝트 등)를 가지는 샷들이 연속되어 나오거나 반복되어 나온다. 따라서, 컨텐트 분석을 통한 샷간의 유사도를 구함으로써 비디오 씬을 분할할 수 있다.

M.M. Yeung and B.L. Yeo[8]은 시간 구속 조건(time-constrained clustering)방법에 의한 스토리 단위로의 비디오 분할을 제안하였다. Alan Hanjalic[9]는 한 편의 영화를 분할해서 자동으로 Logical Story 단위로 묶는 방법에 대해 제안하였으며, Yong Rui[10]는 비디오 시퀀스에 대해서 T.O.C(Table-of-Content)를 만드는 방법을 제안하였다.

위 논문 모두 하나의 사건이나 내용을 담고 있는 씬 레벨의 구조 추출이 주된 목적이다. 또한 씬 분할을 위해서, 하나의 씬 내에는 유사한 칼라 분포를 가지는 샷들로 이루어져 있다는 가정과 유사 샷을 찾는 과정에 있어서 시간 제약성을 이용한다. 시간 제약성을 사용하는 이유는, 현재의 씬에 있는 샷과 다른 씬의 내부에 있는 샷간에도 유사한 칼라 분포를 가질 수 있기 때문에 두 샷간의 유사도 측정 시 두 샷간의 시간 간격을 고려하는 것이다. 이때 유사한 칼라 분포를 지니고 있는 샷들이 검출되면 이 두 샷의 내부에 있는 중간 샷들은 모두 하나의 씬 내에 포함된다.

위 방법들은 샷간의 유사도를 측정하기 위해서 칼라 또는 각 샷 내의 칼라 히스토그램 차의 평균인 칼라 액티버티 특징을 사용해 왔다. 칼라 특징만을 가지고서는 실제 하나의 씬 내에서도 유사한 칼라 분포를 지니고 있지 않는 경우에 있어서는 좋은 씬 분할 결과를 보여 주지 못한다. 이런 경우는 특히 화면 전환이 빠르게 일어나는 액션 영화에서 심하게 나타난다. 화면 전환이 빠르게 일어나는 구간 내 에서는 배경의 칼라 분포가 빠른 속도로 바뀌기 때문에, 유사한 칼라 분포를 지니는 샷들이 존재하지 않을 수 있다. 따라서 실제 하나의 씬이 여러 씬으로 분할되는 씬의 과다 분할 현상과, 이런 현상에 의해서 검출된 씬이 사건을 충분히 반영하지 못하는 문제점을 지니고 있다.

III. 제안하는 방법

본 장에서는 칼라 및 모션 정보를 복합적으로 고려한 씬 분할 기법을 제안한다. 그림 1은 제안하는 알고리즘에 대한 전체 구조를 나타내고 있다.

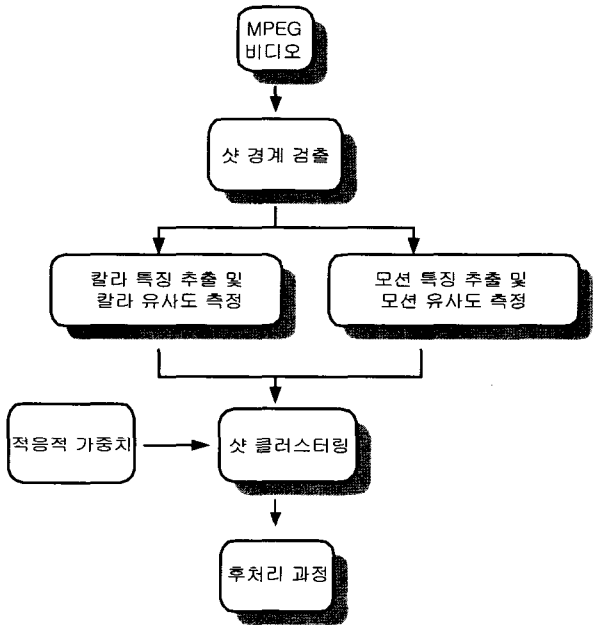

그림 1. 제안하는 전체 알고리즘
Fig. 1. The structure of the proposed algorithm

1. 샷 경계 검출

비디오 분석을 위해서 가장 중요한 초기 작업은 비디오 시퀀스를 샷 단위로 나누는 일이다. 여기서 검출된 샷을 기반으로 대표 프레임 추출, 샷 클러스터링 그리고 씬 분할 등에 이용되기 때문이다.

여기서 샷이란 하나의 카메라 움직임에 의해 찍힌 연속적인 비디오 시퀀스를 의미한다. 또한 비디오의 한 샷은 가장 작은 검색 또는 찾으려는 단위가 된다. 급격한 샷 경계 검출과 점진적인 샷 경계 검출은 기존에 많은 연구가 되어 왔기 때문에 본 논문에서의 샷 경계 검출은 Boon-Lock Yeo[7]가 제시한 방법을 이용한다. 급격한 샷 경계 검출은 이웃한 두 프레임의 히스토그램 차를 이용하고, 점진적인 샷 경계는 여러 프레임에 걸쳐서 일어나기 때문에 K 프레임 윈도우를 사용하여 검출한다.

2. 샷간의 칼라 유사도 측정을 위한 대표 프레임 추출

샷간의 칼라 유사도를 측정하기 위해서, 각 샷 내의 모든 프레임에 대해서 유사도를 측정할 수 없기 때문에 각 샷을 대표할 수 있는 대표 프레임을 추출하여 이들간의 유사도를 측정함으로써 샷간의 유사도를 대신할 수 있다.

기존의 씬 분할 기법에서는 샷간의 유사도 측정 시, 처음과 마지막 프레임을 대표 프레임으로 추출하고 이들간의 유사도를 샷간의 유사도로 대신하였다. 그러나 샷 내의 칼라 분포의 변화가 심한 경우는 처음과 마지막 프레임을 이용해서 샷 내의 다양한 칼라 분포의 변화를 충분히 반영할 수 없다.

따라서 본 논문에서는 샷간의 유사도 측정 시 샷 내의 칼라 분포의 변화를 충분히 반영하기 위해서, sub-shot을 추출하고 대표 프레임을 추출하는 방법을 제안한다. 여기서 sub-shot은 하나의 샷내에서 일정 프레임(시간) 이상의 연속적인 프레임으로 구성된 칼라 분포가 유사한 그룹을 의미한다. sub-shot을 추출하기 위해, 프레임간 간격을 증가시키면서 비교하는 방법은 그림 2에 잘 나타나 있다.

실제 하나의 샷 내에서도 카메라 모션이나 오브젝트의 모션이 커지면, 샷 내의 칼라 분포도 이에 따라서 많이 변화하게 된다. 하지만 이런 경우에 있어서도 이웃하는 프레임끼리는 상당한 유사성을 지닌다. 이 방법은 샷 내에서 칼라 분포가 변하는 경계를 찾기 위해서 프레임간의 시간 간격을 증가시키면서 두 프레임간 비유사도를 비교하는 방법이다. 만약에 이 때의 두 프레임간 비유사도가 정해진 임계값보다 커진다면, 이 지점에서부터 새로운 sub-shot이 시작된다.

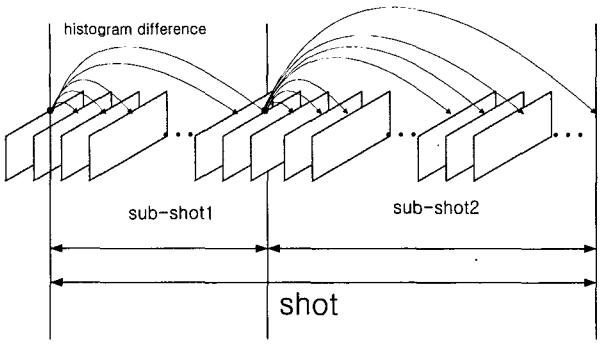

그림 2. sub-shot 추출을 위한 알고리즘도
Fig. 2. The algorithm for extracting sub-shot

위의 알고리즘을 통해서 하나의 샷 내에서 칼라 분포가 많이 변화할 때, 샷을 하나 이상의 sub-shot들로 나누고, 각 sub-shot 내에서 시간적으로 가운데에 위치해 있는 프레임을 대표 프레임으로 추출한다. 이 알고리즘의 장점은 적은 연산량을 가지고 샷 내의 칼라 분포를 충분히 반영할 수 있다는 점이다. 또한 각 sub-shot의 최소 지속 시간 T를 고려함으로써, 하나의 샷 내에서 검출될 수 있는 대표 프레임의 최대 개수를 제한할 수 있다.

3. 칼라, 모션 특징을 이용한 샷간의 유사도 측정

샷간의 유사도를 측정하기 위해서, 본 논문에서는 두 개의 특징인 칼라와 모션 특징을 이용한다. 칼라를 이용한 샷간의 유사도는 위에서 추출한 각 샷의 대표 프레임간의 유사도를 구한 후, 이중 가장 큰 유사도 값을 채택한다. 아래 식 (1), (2)는 샷간의 칼라 유사도를 측정하는 방법을 나타낸다.

$$ShotColorSim_{i,j} = \max(FrameColorSim_{r_i, r_j}) \quad (1)$$

$$FrameColorSim_{r_i, r_j} = \sum_{k=1}^{N_{Bin}} \min(Hist_{r_i}(k), Hist_{r_j}(k)) \quad (2)$$

여기서 r_i 와 r_j는 i, j 번째의 각 샷의 대표 프레임의 인덱스를 나타낸다. N_{Bin}은 칼라 유사도 측정 시 사용된 칼라 빈의 개수를 의미한다.

각 샷 내의 모션 특징은 샷 내에 있는 P 프레임으로부터 추출한다. MPEG-1 동영상의 I, B, P 프레임 중 P 프레임은 앞단의 I나 P 프레임과의 움직임 특징이 매크로 블록 단위의 벡터 형태로 저장되어 있다. 따라서 P 프레임내의 16 ×16 매크로 블록의 모션 벡터를 추출하여 x, y 방향에 대해서 절대 변위를 구한다. 또한 각 방향에 대해서 구한 절대 변위는 16으로 나누어 정규화(normalization)함으로써 0 ~1 사이의 값을 갖게 된다. 각 매크로 블록의 정규화된 절대 모션 크기는 식 (3)을 이용해서 구할 수 있다.

$$NAbsMotion_k = \sqrt{\frac{NM_{x_k}^2 + NM_{y_k}^2}{2}} \quad (3)$$

여기서 NM_{x_k}, NM_{y_k}는 샷 내에 존재하는 P 프레임 중에서 k번째 순방향 매크로 블록의 x, y 방향에 대한 정규화된 절대 변위를 의미한다. 마지막으로 식 (4)를 이용해서 샷 내의 정규화된 절대 평균 모션 크기를 구할 수 있다.

$$NAvgMotion = \frac{1}{N_{MB}} \sum_{k=1}^{N_{MB}} NAbsMotion_k \quad (4)$$

여기서 N_{MB}는 샷 내의 순방향 매크로 블록의 총 개수를 의미한다. 또한 i, j번째의 두 샷간의 모션 유사도는 아래 식 (5)와 같다.

$$ShotMotionSim_{i,j} = 1 - |NAvgMotion_i - NAvgMotion_j| \quad (5)$$

그림 3은 각 샷 내에서의 모션 특징 추출 과정을 나타내고 있다. 모션 추출을 위해 추가적인 연산을 하지 않고, MPEG-1 비디오내의 이미 존재하는 모션 벡터를 이용하기 때문에 연산 시간 면에서 상당한 이점이 있다. 위에서 구한 두 샷간의 칼라와 모션 유사도를 이용하여, 전체적인 샷간의 유사도를 아래 식 (6)과 같이 구할 수 있다.

$$ShotSim_{i,j} = ShotColorSim_{i,j} * W_C + ShotMotionSim_{i,j} * W_M \quad (6)$$

단, $W_C + W_M = 1$이다.

만약 샷 i와 j번째의 전체 유사도가 임계값 α보다 크다면, 두 샷은 같은 컨텐트를 가지고 있다고 볼 수 있기 때문에 하나의 씬에 포함시킬 수 있다. 칼라와 모션 유사도에 가중치를 적용하는 방법은 아래 5절에서 자세히 다룰 것이다.

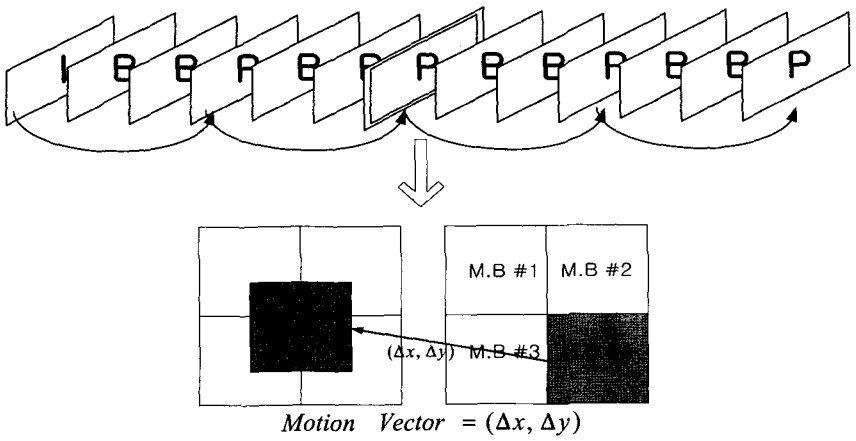

그림 3. 각 샷 내에서의 모션 특징 추출
Fig. 3. The motion information extraction in each shot

4. 비디오 씬 분할 과정

본 논문에서는 비디오 씬 분할을 위해, Improved Overlapping Links(IOL) 기법을 제시한다. 기존의 Overlapping Links(OL) 기법[9]은 유사 샷 검색을 위해 그림 4와 같이 탐색 구간내에서 순차적 검색을 한다. 여기서 탐색 구간은 기준 샷과 유사한 샷이 존재하는 지 여부를 비교하는 구간이다. 본 논문에서 제안하는 IOL 기법에서는 유사 샷 검색을 위해 그림 5와 같이 탐색 구간의 가장 마지막 샷에서부터 역으로 검색한다. 예를 들어 유사 샷 탐색 구간의 마지막 샷이 유사 샷이라면, 제안하는 방법은 한번의 유사도 비교 과정이 필요하지만 기존의 방법은 탐색 구간 내에 있는 모든 샷들에 대해 유사도를 측정해야 한다. 그런데 비디오 씬 분할 시 연산 시간을 줄이기 위해서는, 기준 샷으로부터 가장 멀리 떨어져 있는 유사 샷을 가능한 짧은 시간에 찾을 필요가 있다. 따라서 제안하는 방법이 연산 시간 면에서 기존의 방법에 비해 더 효율적이다. 또한 탐색 구간의 길이가 길어질수록 비교 연산 시간 감소

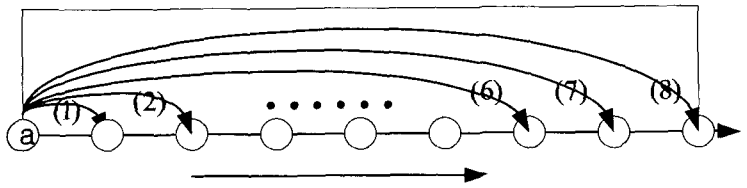

그림 4. 기존의 유사 샷 탐색 과정
Fig. 4. The previous process of finding similar shot

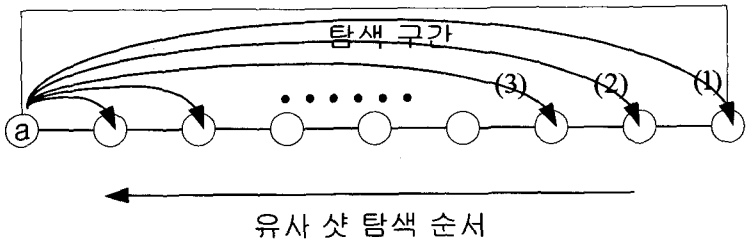

그림 5. 제안하는 유사 샷 탐색 과정
Fig. 5. The proposed process of finding similar shot

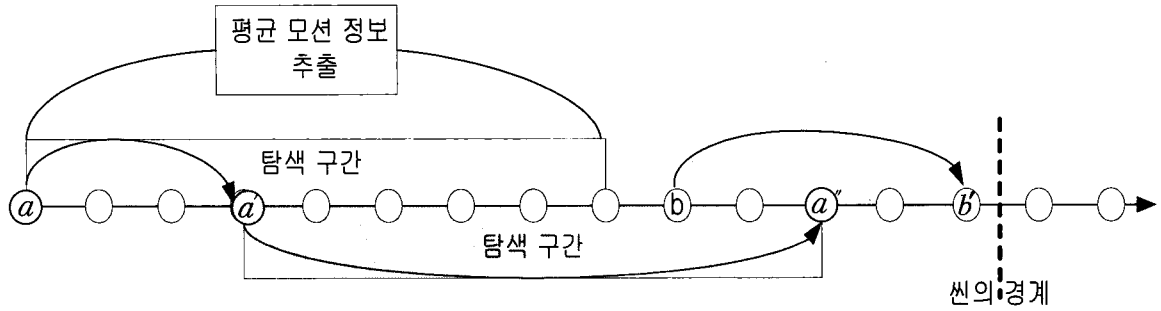

그림 6. 비디오 씬 분할 과정
Fig. 6. The procedure of video scene segmentation

효과는 증가한다.

IOL 기법은 기존의 OL 기법과 마찬가지로 유사 샷을 연결시켜 나감으로써 비디오 씬을 분할한다. 위 그림 6은 IOL 기법을 이용한 비디오 씬 분할 과정을 나타내고 있다. 탐색 구간에서부터 역 검색 과정을 통해 기준 샷과 최초로 주어진 임계값을 넘는 유사 샷이 발견되면, 두 샷은 서로 연결(link)된다. 이 방법의 특징은 기존의 전체 샷 시퀀스의 샷 레이블링을 통한 비디오 씬 분할[9]이 아닌, 유사한 샷끼리 계속 연결시켜 나감으로써 씬의 경계를 확장시킨다는 점이다. 그림 6에서 a와 a', a'와 a''는 유사 샷으로 검출되었기 때문에 서로 연결(link)되었다. 유사 샷이 더 이상 나오지 않을 때는 그 이전 샷들로 이동하면서 유사 샷을 검색해나감으로써, 씬의 경계를 확장시킨다. 그림 6에서 a''는 탐색구간 내에서 유사 샷은 없지만, 이전 샷인 b는 샷 b'와 유사하기 때문에 서로 연결(link) 되고, 비디오 씬은 계속 확장된다. 이전 샷들에 대해서도 더 이상의 유사 샷이 검출되지 않는 경우에는, 다음 샷에서부터 새로운 씬이 시작되고 똑같은 알고리즘이 적용된다. 그림 6에서 샷 b'는 탐색 구간 내에서 유사 샷이 없고, b' 이전 샷들과도 유사 샷이 없기 때문에 b'에서 씬이 분할된다.

이러한 방법을 통해서 추출된 연속된 샷들의 집합을 SceneUnit이라고 명명할 것이다. 실제로 하나의 씬이 여러 개의 씬으로 추출되는 과다 분할 현상이 일어나기 때문에 검출된 씬은 실제의 씬과는 다른 용어를 사용할 필요가 있기 때문이다. 검출된 각각의 SceneUnit은 한 씬의 부분 집합이 된다.

5. 샷간의 유사도를 구하기 위한 적응적 가중치 적용

샷간의 유사도를 구하기 위해서는 식 (6)에서 나타낸 바와 같이 칼라와 모션을 통해 구한 샷간의 유사도에 각각 가중치를 주어야 한다. 그림 4와 같이 탐색 구간 내에서 유사 샷을 찾기 때문에, 탐색 구간의 특성에 따라서 칼라와 모션에 가중치를 적용한다. 이를 위해 본 논문에서는 탐색 구간내의 평균 모션을 추출하고, 이 특성에 따라 가중치를 적응적으로 조정한다. 이러한 적응적 가중치를 주지 않는다면 씬 분할을 하려는 모든 비디오에 대해서 실험을 통한 최적의 가중치를 찾아야만 하고, 고정된 가중치가 비디오 전체에 대해 적용되기 때문에 탐색 구간 내의 특성을 충분히 반영할 수 없다. 따라서, 현재의 샷과 유사 샷을 찾기 위해서 탐색 구간 내에 있는 샷들에 대해서 식 (7)을 이용해 평균 모션을 구하고, 이 평균 모션을 이용해서 탐색 구간 내의 특징으로 사용한다. 이때 각 샷의 모션 특징은 3절에서 구한 정규화된 절대 평균 모션 크기(NAvgMotion)를 이용한다.

만약 탐색 구간 내의 평균 모션이 크다면, 이 구간은 액션 씬과 같이 화면이 빠르게 진행되는 동적 구간이기 때문에 칼라보다는 모션에 더 큰 가중치를 주게 되고, 반대로 탐색 범위내의 평균 모션이 작다면 이 구간은 정적인 구간이기 때문에 모션보다는 칼라에 더 큰 가중치를 적용하게 된다.

$$W_M = \frac{1}{SR+1} \sum_{i=I_{CS}}^{I_{CS}+SR} NAvgMotion_i \qquad (7)$$

여기서 I_{CS}는 샷 시퀀스에서 현재 샷의 인덱스를 의미하며, SR(Search Range)는 I_{CS}로부터 탐색할 샷의 개수를 의미한다. 실제로 수식 (6)에서 모션 유사

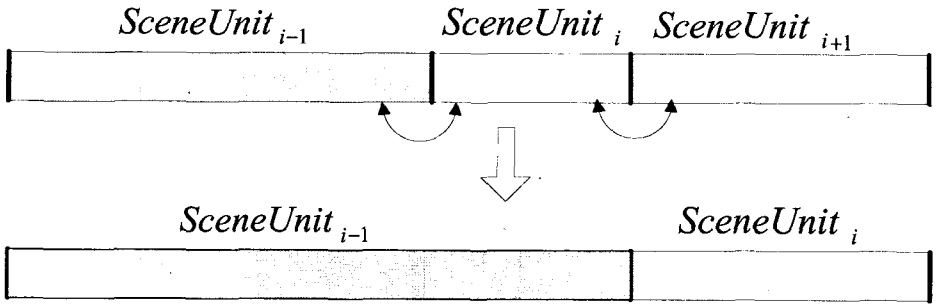

그림 7. 이웃하는 SceneUnit간의 모션 특징을 이용한 SceneUnit 병합
Fig. 7. SceneUnit Merging using motion feature between adjacent two SceneUnits

도에 대한 가중치 W_M은 샷 검색 범위 내의 평균 모션을 이용하였고, 칼라 유사도에 대한 가중치 W_C는 1에서 모션 적용 가중치를 뺌으로써 자동적으로 얻을 수 있다.

6. 모션 특징을 이용한 후처리 과정

실제로 비디오를 위의 씬 분할 알고리즘을 적용해 보면, τ시간 보다 적은 시간 간격을 가지고 있는 SceneUnit이 상당수 검출된다. 여기서 τ 는 SceneUnit 의 시간 길이를 나타낸다. 그러나 이러한 SceneUnit들은 실제로 사용자에게 제대로 된 사건(내용)을 전달할 수 없기 때문에 이웃하는 SceneUnit과 서로 병합함으로써 내용을 충실히 전달할 수 있게 된다.

이렇게 과다 분할된 SceneUnit들은 유사 샷 탐색 구간 내에서 칼라 특징의 유사도가 낮기 때문에 생긴다. 따라서 하나의 씬 내에 있는 샷들은 유사한 모션 특성을 가진다는 점을 이용한다. 현재의 SceneUnit의 시간 간격이 τ보다 적다면, 이웃하는 SceneUnit과의 모션 특징을 비교해서 두 SceneUnit을 병합할 수 있다. 각 SceneUnit의 모션 특징은 식 (8)을 이용해서 구한다.

$$M_{SU}=\frac{1}{I_{ES}-I_{SS}+1}\sum_{i=I_{SS}}^{I_{ES}} NAvgMotion_i \qquad (8)$$

여기서 I_{SS}와 I_{ES}는 각 SceneUnit의 처음과 마지막 샷의 인덱스를 나타낸다. 만약 서로 인접해 있는 두 SceneUnit의 모션 유사도가 주어진 임계값보다 크다면, 두 SceneUnit은 서로 병합된다.

이렇게 SceneUnit을 묶어나감으로써 씬의 과다 분할 현상을 줄일 수 있으며, 병합된 SceneUnit은 더욱 더 충실히 씬 내의 의미를 반영할 수 있게 된다.

IV. 실험 결과 및 고찰

1. 실험 환경

비디오 씬 분할을 위해서 실험은 터미네이터 II MPEG 1 비디오의 후반부(약 1시간)에 대해 이루어졌다. 초당 30프레임으로 이루어져 있고, 영상 사이즈는 352 ×240이다. 샷 경계 검출 결과 1348개의 샷들이 검출이 되었으며, 여기에 제안된 대표 프레임 추출 방법을 적용하였다. 대표 프레임 추출 시 sub-shot의 최소 지속 시간을 각 샷의 지속시간(duration)의 1/5로 정해 놓음으로써, 추출될 수 있는 대표 프레임의 수를 5개로 제한하였다. 아래 표 1은 샷 경계 검출 및 대표 프레임 추출 결과로 얻어낸 샷 레벨에서의 비디오 구조를 나타낸다. 유사 샷 탐색 범위는 10으로 하였다. 또한 샷간의 유사도가 0.66이상이면 두 샷은 유사하다고 보고 두 샷 내에 있는 샷들은 하나의 SceneUnit에 포함된다. 여기서 탐색 범위 및 샷간의 유사도 값은 변수화 되어야 하며 대상으로 하는 비디오에 적합한 값을 선택해야 한다. 후처리 과정에서는 5초(τ=5)이하의 SceneUnit에 대해서, 이웃하는 SceneUnit과의 모션 유사도가 0.66이상이면 두 SceneUnit을 합쳤다. 또한 RGB 각 성분을 8개의 bin으로 양자화 하였고, 칼라 히스토그램을 이용해서 두 샷간의 칼라 유사도를 측정하였다.

표 1. 샷 레벨에서의 비디오 구조
Table 1. The video structure in shot level

샷 넘버	시작 프레임	마지막 프레임	대표 프레임 #1	대표 프레임 #2	대표 프레임 #3	대표 프레임 #4	대표 프레임 #5
1	0	22	11				
2	23	457	66	249	423		
3	458	721	518	628	699		
4	722	827	774				
5	828	1,112	894	1,036			
6	1,113	1,296	1,161	1,238	1,281		
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
613	62,305	62,329	62,317				
614	62,330	62,476	62,403				
615	62,477	62,556	62,500	62,539			
616	62,557	62,594	62,575				
617	62,595	62,743	62,611	62,643	62,675	62,708	62,733
618	62,744	62,771	62,757				
619	62,772	62,798	62,785				
620	62,799	62,876	62,808	62,825	62,841	62,862	
621	62,877	62,912	62,894				
622	62,913	62,941	62,927				
623	62,942	62,983	62,962				
624	62,984	63,110	63,015	63,059	63,091		
625	63,111	63,189	63,125	63,164			
626	63,190	63,555	63,226	63,309	63,430	63,530	
627	63,556	63,610	63,583				
628	63,611	63,735	63,673				
629	63,736	63,785	63,760				
630	63,786	63,812	63,799				
631	63,813	63,832	63,822				
632	63,833	63,870	63,838	63,848	63,862		
633	63,871	63,919	63,895				
634	63,920	63,998	63,959				
635	63,999	64,057	64,028				
636	64,058	64,168	64,113				
637	64,169	64,264	64,178	64,197	64,216	64,245	
638	64,265	64,302	64,274	64,293			
639	64,303	64,426	64,323	64,356	64,381	64,410	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
1,343	110,308	110,476	110,370	110,454			
1,344	110,477	110,516	110,496				
1,345	110,517	110,734	110,539	110,605	110,692		
1,346	110,735	110,764	110,749				
1,347	110,765	110,902	110,779	110,847			
1,348	110,903	110,996	110,949				

2. 실험 결과 및 고찰

본 알고리즘의 비디오 씬 분할 성능을 알아보기 위해서 기존의 칼라만을 사용했을 경우, 적응적 가중치를 적용한 경우, 적응적 가중치에 후처리 과정을 한 경우에 대한 실험 결과를 비교하였다.

2.1 SceneUnit의 장소별 사건 반영 비교

비디오의 씬 구간을 나누는 기준은 개개인마다 다르기 때문에 검출된 씬에 대한 정확한 평가가 쉽지 않다. 따라서, 객관적인 평가를 위해 전체 비디오를 장소별로 나누고 검출된 SceneUnit이 얼마나 장소 내에서 일어나는 사건 또는 의미를 잘 반영하는지 평가하였다. 실제 터미네이터 II는 장소의 이동에 따라 주로 사건의 내용이 바뀌는 경향을 보여준다. 표 2는 각 장소에 대하여 여러 알고리즘에 의해 검출된 SceneUnit과의 실제 사건 반영 관계를 나타낸다. 칼라만을 사용해서 씬 분할을 한 경우는 씬 과다 분할 현상 때문에, 검출된 각 SceneUnit은 제대로 사건을 반영할 수 없음을 알 수 있다. 또한 적응적 가중치를 적용한 경우와 적응적 가중치에 후처리 과정을 한 경우로 갈수록 훨씬 실제 장소에 일어나는 사건을 잘 반영함을 알 수 있다. 가장 씬 분할이 잘되지 않는 11번째 장소를 살펴보면, 이 사실을 확인할 수 있다. 실제 1개의 씬으로 구성되어 있음에도 불구하고 칼라만을 이용했을 경우에는 130개의 SceneUnit이 검출되었고, 모션과 칼라 특징을 이용한 적응적 가중치를 적용했을 경우에는 30개, 마지막으로 적응적 가중치에 후처리 과정을 적용한 경우는 14개의 SceneUnit이 검출되었다. 이를 통해 각각의 알고리즘에 의해 검출된 씬에 대한 사건 반영도에 있어서 제안하는 알고리즘의 우수함을 알 수 있다. 왜냐하면 하나의 씬 임에도 불구하고, 검출된 씬의 개수가 많다면 검출된 씬은 그 만큼 사건을 반영할 수 있는 확률이 떨어지기 때문이다. 표 2를 통해서

표 2. 장소별 SceneUnit의 의미 반영
Table 2. The reflection of semantics about each location

Location Number	Manually Segmentation 단위:샷	칼라만 사용 (C)	제안하는 방법 #1 (A)	제안하는 방법 #2 (AP)
1	1-83	C1-C6 (1-83)	A1-A2 (1-76)	AP1 (1-76)
2	84-105	C7-C12 (84-106)	A3-A4 (77-105)	AP2-AP3 (77-105)
3	106-122	C13-C15 (107-118)	A5-A6 (106-118)	AP4 (106-118)
4	123-128	C16 (119-182)	A7 (119-183)	AP5 (119-183)
5	129-285	C17-C22 (183-285)	A7-A8 (119-285)	AP5-AP6 (119-285)
6	286-290	C23 (286-290)	A9 (286-290)	AP7 (286-290)
7	291-351	C24-C38 (291-351)	A10-A12 (291-342)	AP8 (291-342)
8	352-357	C39-C44 (352-357)	A13 (343-398)	AP9 (343-398)
9	358-638	C45-C86 (358-625)	A14-A24 (399-560)	AP10-AP15 (399-560)
10	639-943	C87-C106 (626-943)	A25-A29 (561-943)	AP16-AP18 (561-943)
11	944-1348	C107-C236 (944-1348)	A30-A59 (944-1348)	AP19-AP32 (944-1348)

칼라 특징을 이용한 경우의 SceneUnit : C
칼라+모션 특징에 적응적 weight 적용한 경우의 SceneUnit : A
적응적 weight 적용 + 후처리(post-processing)를 이용한 경우의 SceneUnit : AP
주: 본 실험 결과는 SR=10 인 경우에 대한 것이다.

적응적 가중치와 후처리 과정을 동시에 사용했을 경우는 장소 9와 11을 제외하면 대부분 검출된 SceneUnit이 각 장소의 사건을 충실히 반영하고 있음을 알 수 있다.

2.2 씬의 과다 분할 비교

표 3을 통해서 씬의 과다 분할(over-segmentation)정도와 분할된 전체 SceneUnit의 개수를 알 수 있다. 실험 결과 본 논문에서 제안하는 방법이 가장 적은 씬 과다 분할 결과를 나타냄을 알 수 있다. 칼라만을 이용해 SceneUnit을 검출했을 경우는 하나의 씬 당 평균 21.45가 나온 반면에 비디오 내의 특성을 고려한 적응적 가중치를 적용한 경우에 있어서는 5.36으로 상당히 줄어듬을 알 수 있다. 또한 적응적 가중치에 후처리 과정을 통해서는 2.9로 나왔으며, 이는 칼라만을 이용했을 경우보다 씬의 과다 분할이 약 1/7로 줄어들었음을 알 수 있었다.

표 3. 각 알고리즘에 대한 과다 분할 비교
Table 3. The comparison of over-segmentation about each algorithm

	칼라만 사용	제안하는 방법 #1 적응적 가중치 적용	제안하는 방법 #2 적응적 가중치+후처리
전체 SceneUnit의 개수	236	59	32
한 씬 당 검출된 SceneUnit의 평균 개수	21.45	5.36	2.9

2.3 씬 분할 시 연산 시간 비교

비디오 씬 분할 시, 기존의 overlapping links 기법과 본 논문에서 제안하는 improved overlapping links 기법과의 연산 시간을 측정하였다. 아래 표 4는 이 결과를 나타낸다. 표 4를 통해서 IOL 기법이 OL 기법에 비해 30% 정도의 연산 시간이 감소되었음을 알 수 있다.

표 4. 비디오 씬 분할 시 연산 시간 비교
Table 4. The comparison of processing time for video scene segmentation

	overlapping links	improved overlapping links
연산 시간	7분 42초	5분 21초

V. 결 론

본 논문에서는 비디오 내의 사건 또는 의미를 반영할 수 있는 새로운 씬 분할 기법을 제안하였다. 샷간의 정확한 칼라 유사도를 측정하기 위해서, 각 샷 내의 다양한 칼라 분포를 충분히 반영할 수 있는 대표 프레임을 추출하여 샷간의 유사도 측정에 이용하였다.

씬 분할 시 기존의 칼라만을 사용했던 방법에 모션 특징을 추가적으로 사용하였고, 탐색 구간내의 모션 특성을 이용해서 적응적 가중치를 적용하였다. 이를 통해서 칼라만으로 씬 분할을 했을 경우보다 씬 과다 분할 현상을 줄일 수 있었으며, 또한 비디오내의 의미를 충실히 반영할 수 있었다. 후처리 과정을 통해 아주 짧은 시간으로 이루어진, 실제로 하나의 사건이나 의미를 반영할 수 없는 SceneUnit을 이웃 SceneUnit과 병합함으로써 보다 충실하게 의미를 반영할 수 있었다. Improved overlapping links 기법을 이용하여 기존의 overlapping links 기법에 비해서, 연산 시간을 줄일 수 있었다.

추출된 계층적인 비디오 구조는 샷 레벨의 구조와는 달리 사용자가 비디오를 브라우징하거나 검색할 때 비디오의 씬, tit, 키 프레임 단위의 접근을 가능하게 한다.

참 고 문 헌

[1] H. Okamoto, M. Nakamura, Y. Hatanaka, and S. Yamazaki, "A Consumer Digital VCR for Advanced Television," *IEEE Trans. Consumer Electronics*, vol. 39, pp. 199-204, Aug. 1993.

[2] P. H. N. de with, "Data Compression Systems for Home-use Digital Video Recording," *IEEE J. Select. Areas Commun.*, vol. 10, pp. 97-121, Jan. 1992.

[3] N. Yanagihara, C. Siu, K. Kanota, and Y. Kubota, "A Video Coding Scheme with a High Compression Ratio for Consumer Digital VCR's," *IEEE Trans. Consumer Electronics*, vol. 39, pp. 192-198, Aug. 1993.

[4] G. Ahanger and T. D. C. Little, "A Survey of Technologies for Parsing and Indexing Digital Video," *Journal of Visual Communication and Image Representation*, vol. 7, no. 1, pp. 28-43, Mar. 1996.

[5] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic Partitioning of Full-motion Video," *Multimedia Systems*, vol. 1, pp. 10-28, Jul. 1993.

[6] D. Zhong, H. Zhang, and S. F. Chang, "Clustering Methods for Video Browsing and Annotation," *in Proc. SPIE Storage and Video Retrieval for Still Image and Video Databases IV*, vol. 2670, pp. 239-246, Feb. 1996.

[7] Boon-Lock Yeo. Efficient Processing of Compressed Images and Video. Ph.D Thesis, Princeton University, 1996.

[8] M. M. Yeung and B. L. Yeo, "Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 7, no. 5, pp. 771-785, Oct. 1997.

[9] Alan Hanjalic, Reginald L. Lagendijk, "Automated High-Level Movie Segmentation for Advanced Video-Retrieval Systems," *IEEE Transactions on Circuits & Systems for Video Technology*, vol. 9, no. 4, pp. 580-588, Jun. 1999.

[10] Yong Rui, Thomas S. Huang, and Sharad Mehrotra, "Constructing Table-of-Content for Videos," ACM Multimedia Systems, vol. 7, no. 5, pp. 359-368, Sep. 1999.

저 자 소 개

송 창 준

1998년 2월 : 고려대학교 전자공학과 졸업 (공학사)
2000년 2월 : 고려대학교 전자공학과 대학원 졸업 (공학석사)
2000년 2월 ~ 현재 : 삼성 전자 중앙연구소 연구원
주관심분야 : 영상 및 비디오 분할/요약/검색, MPEG 4, MPEG 7

고 한 석

1982년 5월 : Carnegie-Mellon Univ. Electrical Engineering 졸업 (공학사)
1986년 5월 : Univ. of Maryland System Engineering 졸업 (공학석사)
1988년 5월 : Johns Hopkins Univ. Electrical Engineering 졸업 (공학석사)
1992년 5월 : Catholic Univ. of America Electrical Engineering 졸업 (공학박사)
1983년 9월 ~ 1995년 2월 : Univ. of Maryland Part-time Ass. Professor
1995년 3월 ~ 현재 : 고려대학교 전기.전자.전파공학부 부교수
주관심분야 : 음성신호처리, 이미지 데이터 융합, 표적신호 탐지/추정/추적

권 용 무

1980년 2월 : 한양대학교 전자공학과 졸업 (공학사)
1983년 2월 : 한양대학교 대학원 전자공학과 졸업 (공학석사)
1992년 8월 : 한양대학교 대학원 전자공학과 졸업 (공학박사)
1983년 1월 ~ 현재: KIST 영상미디어 연구센터 책임연구원
주관심분야 : 영상검색, 영상압축, 입체영상, 가상환경