

■ '98 정보과학 논문경진대회 수상작

MPEG 비디오 스트림에서 줄거리 특성에 기초한 장면 경계 검출 방법

(A Scene Boundary Detection Scheme based on Story Line for MPEG Video Streams)

이 숙 경[†] 남 종 호^{**}

(Sook Kyoung Lee) (Jong Ho Nang)

요 약 비디오 데이터 스트림을 샷(shot) 혹은 장면(scene) 단위로 나누는 자동 비디오 인덱싱 기법은 VoD나 디지털 비디오 라이브러리와 같이 비디오 데이터를 이용하는 응용 분야를 개발하기 위하여 꼭 필요한 기능이다. 본 논문에서는 줄거리(story line)를 갖는 영화나 드라마와 같은 비디오 데이터에서의 장면 단위 인덱싱을 위해 장면의 경계를 자동으로 검출하는 방법을 제안한다. 일반적으로 영화에서 시간적으로 인접한 두 샷이 유사할 때 샷의 중간 부분은 움직임에 따라 변하는데 똑같은 움직임이 반복되지 않는다는 특성을 가진다. 제안한 검출 방법에서는 이러한 특성을 이용하여 각 샷의 처음과 마지막 프레임을 대표 프레임으로 선택하고 이들의 칼라 히스토그램으로 샷을 추상화한다. 이렇게 추상화한 샷에 대하여 영화에서의 장면은 일반적으로 비슷한 내용의 샷이 반복된다는 특성을 이용하여 장면의 경계를 검출하였는데, 하나의 장면에 해당하는 샷들을 묶어줄 때 시간적으로 인접한 이전 샷들과의 유사도만 측정하는 간단한 방법을 사용하였다. 본 논문에서 제안한 영화나 드라마에 대한 장면 경계 검출 방법은 MPEG 형식의 비디오 데이터를 이용한 디지털 라이브러리 등의 구축에 유용하게 사용될 수 있을 것이다.

1. 서 론

최근 멀티미디어 정보의 사용이 급속히 늘어나고 있는데 텍스트, 이미지, 그래픽, 오디오, 비디오와 같은 여러 멀티미디어 데이터 중에서 특히 비디오 데이터는 사람이 쉽게 인지할 수 있기 때문에 정보 전달이 효율적이다. 이러한 비디오를 기본 데이터로 사용하는 응용 시스템을 개발하고자 할 때는 압축 형식의 비디오를 저장한 데이터베이스(VDB : Video DataBase)를 우선 구축하여야 하며, 비디오 데이터에 대한 자동화된 비디오 인덱싱과 브라우징 기능이 필요하다. 이러한 기능을 제공하기 위해 비디오의 내용을 기반으로 하여 자동으로 인덱싱하는 기술이 필수적이라 할 수 있다.

비디오 데이터에 대한 인덱싱에 관한 지금까지의 연구는 샷 단위의 인덱싱 [9, 10, 11, 12, 13]이 대부분이

다. 그러나 비디오의 샷단위 인덱싱은 여전히 그 개수가 많기 때문에 샷의 상위 레벨로서 의미적 단위라고 할 수 있는 장면(scene)의 경계를 찾는 장면 단위 인덱싱이 비디오 인덱싱으로서 더 바람직하다. 하지만 샷의 경계는 카메라의 물리적인 움직임에 의해 정의되는 것이므로 쉽게 찾을 수 있지만, 장면은 같은 장면이라도 보는 사람에 따라 다르게 판단할 수도 있는 의미적인 단위이므로 장면의 경계를 찾는 것은 쉽지 않다. 장면 경계 검출에 대한 기존의 연구[2,3,4,5,7]는 뉴스나 스포츠와 같은 특정 비디오를 대상으로 하는 영역 종속적인 방법[2]과 영화나 드라마와 같은 일반적인 비디오를 대상으로 하는 영역 독립적인 방법[3,4,5,7]으로 나눌 수 있다. 영역 종속적인 방법은 특정 비디오를 대상으로 하기 때문에 정확도가 높지만 검출을 하기 전에 비디오에 대한 모델이 구성되어야 하고, 이러한 모델링 과정은 시간이 많이 소요되고 다른 영역에는 적용하지 못 한다는 단점이 있다. 영역 독립적인 방법은 영화에서 장면이 가지는 특성을 고려하지 않았기 때문에 필요 없는 정보를 많이 사용하고 있고 이로 인해 장면 경계를 검출할 때 잘못된 결과를 낳기도 한다.

본 논문에서는 줄거리(story line)를 갖는 영화나 TV

· 본 연구는 1999년도 정보통신부 대학 기초연구 지원사업에 의한 것임

† 비 회 원 : 서강대학교 컴퓨터학과

lskyung@hei.co.kr

** 종신회원 : 서강대학교 컴퓨터학과 교수

jhnang@ccs.sogang.ac.kr

논문접수 : 1998년 7월 4일

심사완료 : 1999년 7월 13일

드라마와 같은 비디오 데이터에 대해 장면을 이루는 샷들의 특성을 이용하는 새로운 장면 경계 검출 방법을 제안한다. 일반적으로 영화에서 시간적으로 인접한 두 샷이 유사할 때 샷의 중간에 나타나는 프레임들은 움직임의 변화에 따라 변하는데 똑같은 움직임의 변화가 반복되지 않는 특성을 가진다. 본 논문에서 제안한 샷의 추상화 방법에서는 영화 구성상의 이러한 특성을 이용하여 처음과 마지막을 나타내는 두 개의 프레임을 대표 프레임으로 선택하고 이들의 칼라 히스토그램을 사용하여 샷을 추상화한다. 이렇게 추상화된 샷에 대하여 각 샷들이 이전에 나타났던 샷인지를 검사하고 처음 나타나는 샷이면 뒤에 나타나는 샷들 중 현재의 샷 이전의 샷들과 유사한 샷이 없으면 새로운 장면의 시작으로 한다. 이러한 장면 경계 검출 방법은 여러 MPEG형식의 비디오 데이터를 사용하는 멀티미디어 응용 프로그램을 개발하는데 있어서 필요한 비디오 데이터베이스를 자동적으로 구축하는데 유용하게 사용될 수 있을 것이다.

본 논문은 다음과 같이 구성되어 있다. 2 장에서는 장면 경계 검출 방법에 대한 기존의 연구들에 대하여 살펴 보고, 3 장에서는 장면 경계 검출 방법을 샷의 추상화, 샷의 유사도 측정 방법 그리고 이를 이용하여 실제로 장면 경계를 검출하는 알고리즘에 대하여 설명한다. 4 장에서는 3 장에서 제안한 방법의 실험 결과를 토대로 성능 평가와 분석을 하고 마지막으로 5 장에서는 결론을 맺는다.

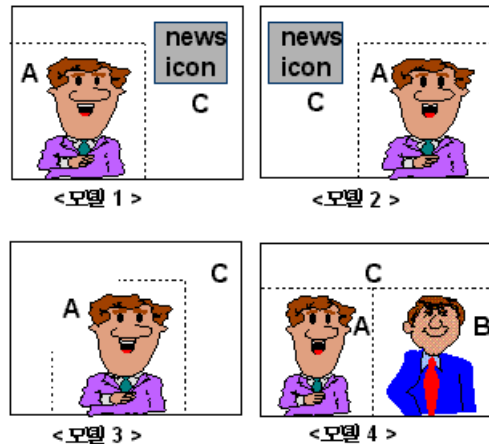
2. 연구배경

비디오 데이터는 프레임(frame)-샷(shot)-장면(scene)의 계층적인 구조를 가진다. 프레임은 비디오를 구성하는 기본 요소이며, 여러 개의 연속적인 프레임이 모여서 카메라의 물리적인 동작에 의한 샷을 구성하는데 이러한 샷은 비디오 인덱싱의 실질적인 기본 단위이다. 또한 여러 개의 샷이 서로간의 의미적 관계에 의해 장면을 구성한다. 본 장에서는 이러한 비디오 데이터의 계층적인 관계에 근거한 인덱싱으로 기존의 장면 경계 검출 방법을 영역 종속적인 방법과 영역 독립적인 방법으로 구분하여 설명하도록 한다.

2.1 영역 종속적인 방법

영역 종속적인 장면 경계 검출 방법은 특정 분야에 대해 그 분야의 특성에 맞는 모델을 미리 구성하고 그 모델에 맞는 장면의 경계를 정의한다. 이러한 방법의 예로 뉴스를 대상으로 한 Zhang [2]의 방법을 살펴 보면 뉴스 비디오는 앵커가 말하는 샷과 그 뉴스의 내용을 보여 주는 샷이 반복되어 나타나므로 각각의 샷을 앵커

샷과 뉴스 샷으로 분류한다. 이 중에서 뉴스 샷은 특정한 모델을 가지지 않지만 앵커 샷은 다음 <그림 1>과 같이 4가지 유형의 모델을 가지므로 주어진 샷이 모델링된 앵커 샷인지 판단하여 앵커 샷이면 새로운 장면의 시작으로 할 수 있다. 이 때, 앵커 샷의 모델링을 위한 모델 이미지를 선택하기 위해서 앵커 샷은 샷 내에서 움직임이 거의 없으므로 샷의 각 프레임 색깔에 대한 평균과 분산을 구하여 그 값이 주어진 임계값보다 작으면 4가지 모델 중 하나가 될 수 있는지 검사한다. 만약 <모델 4>이면 영역을 A, B, C로 나누는데 A와 B는 앵커가 등장하는 영역이므로 약간의 움직임이 있고 C는 배경이므로 움직임이 전혀 없다. 샷의 각 영역별로 이러한 특성을 만족하면 앵커 샷으로 하고 그 샷의 프레임들의 평균색을 <모델 3>의 모델 이미지로 선택한다. 앵커 샷에 대한 모델 이미지가 모두 선택되면 새로운 샷이 나타날 때마다 모델링된 앵커 샷과의 유사도를 검사해서 유사한 모델이 있으면 앵커 샷으로 하고, 그렇지



(1) 앵커 프레임의 4가지 모델



(2) <모델 1>의 실제 예

그림 2 뉴스 비디오에서 앵커 프레임의 실제 모델링

않으면 뉴스 샷으로 한다. 이러한 영역 종속적인 장면 경계 검출 방법은 특정 영역을 대상으로 하기 때문에 정확도가 높지만, 검출을 하기 전에 영역에 맞는 모델이 구성되어야 하고, 이러한 모델링 과정은 시간이 많이 소요되고, 다른 영역에는 적용하지 못 하는 단점을 가지고 있다.

2.2 영역 독립적인 방법

일반적으로 줄거리를 갖는 영화와 같은 비디오에 대한 장면 경계 검출 방법을 영역 독립적인 방법으로 분류한다.

2.2.1 Yeung의 방법

Yeung의 장면 인덱싱 방법 [4,5]에서는 우선 샷 내에 움직임이 많을수록 많은 프레임은 샷의 대표 프레임으로 선택하여 샷을 추상화한다. 다음은 추상화된 샷들에 대하여 장면의 경계를 찾기 위해 서로 유사한 샷들을 묶어 주는데 초기에는 N개의 샷에 대하여 각각 한 개의 샷만을 포함하는 클러스터를 N개 생성한다.

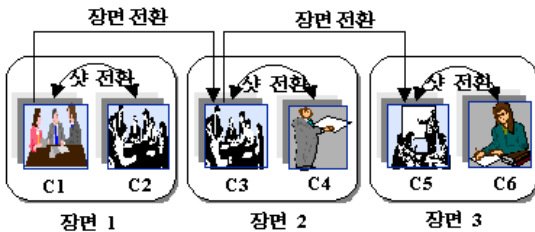


그림 3 STG와 이를 이용한 장면 검출의 예

이들 N개의 클러스터에 대하여 차이값이 가장 작은 클러스터의 쌍을 R과 S라 할 때 이 차이 값이 주어진 임계값 보다 작으면 R과 S는 하나의 클러스터로 묶는다. 이러한 과정을 클러스터가 한 개가 되거나 R과 S의 차이 값이 주어진 임계값 보다 크게 될 때까지 수행한다. 이렇게 생성된 클러스터들의 시간적인 전후 관계에 따라 화살표로 연결하면 <그림 2>와 같은 STG(Scene Transition Graph)를 생성할 수 있다. 여기에서 C1, C2, C3, C4, C5, C6는 클러스터를 의미하고, 두 클러스터를 연결하는 화살표는 시간 관계를 의미한다. 생성된 STG에서 두 클러스터가 <그림 2>의 C1과 C2처럼 양 방향으로 향하는 화살표를 가지고 있으면 이들이 포함하는 샷들은 서로 번갈아가며 나타난 것이므로 이 둘은 한 장면이라고 할 수 있다. 따라서 STG내의 cut edge들을 찾아서 이를 장면의 경계로 하는데, <그림 2>의 STG는 C1과 C3, C3와 C5사이의 edge를 제거하면 그래프가 별도의 연결된 서브그래프(connected subgraph)

를 생성하므로 cut edge이고, 이에 따라 3개의 장면을 찾을 수 있다.

2.2.2 Rui의 방법

장면 단위의 인덱싱을 위한 연구 중의 하나인 Rui의 방법[7]에서는 한 샷의 대표 프레임으로 시작 프레임과 마지막 프레임을 선택하고 그들을 칼라 히스토그램과 움직임 정도 값으로 추상화하였다. 또한 Yeung의 방법에서 클러스터링을 했던 것처럼 그룹화를 하는데 이를 위한 두 샷의 유사도 측정은 색의 유사도와 움직임 정도 유사도를 모두 적용한다. 실제 장면 경계를 검출하는 방법은 우선 N개의 샷 중 첫 번째 샷을 첫 번째 그룹과 첫 번째 장면에 포함시킨다. 그 다음 샷부터 마지막 샷까지에 대하여 현재 샷을 i라 할 때 샷 i와 기존의 그룹과의 유사도를 측정해서 가장 유사도가 큰 그룹과의 유사도가 주어진 임계값 보다 크면 샷 i를 그 그룹에 포함시키고 다음 샷 i+1을 검사한다. 그렇지 않으면 새로운 그룹을 생성하고 생성된 그룹과 기존의 장면과의 유사도를 측정해서 가장 유사도가 큰 장면과의 유사도가 주어진 임계값 보다 크면 그 장면에 포함시키고 그렇지 않으면 새로운 장면을 생성하고 다음 샷 i+1을 검사한다.

2.2.3 Kender의 방법

장면 단위의 인덱싱을 위한 연구 중의 하나인 Kender의 방법[3]에서는 coherence라는 개념을 이용하여 유사한 샷들을 그룹핑하였다. 다음 <그림 3>은 이 방법에서 사용한 coherence의 개념을 설명하는 것인데, 각 알파벳은 한 샷을 의미하고 같은 알파벳을 가지는 샷은 유사한 내용의 샷임을 의미한다. 여기에서 (1)은 과거 샷에 대한 재호출(recall)이 많으므로 coherence가 좋고, (2)는 coherence가 나쁘다고 할 수 있다.

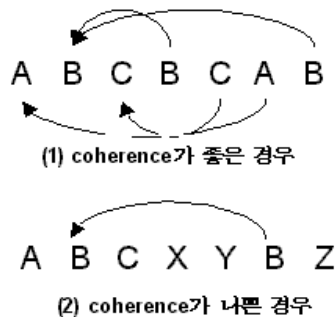


그림 4 Coherence의 의미

장면의 경계에서는 새로운 샷들이 나타나므로 장면

경계 뒤의 샷들이 경계 앞의 샷들을 재호출하는 정도가 주변보다 작게 된다. 이러한 원리를 이용하여 샷의 경계마다 그 경계 이후의 모든 샷들의 경계 이전의 모든 샷들을 얼마나 재호출하는지 계산하여 이 값을 coherence로 하는데 샷의 재호출값은 두 샷의 유사성이 크고 거리가 가까울수록 큰 값을 할당한다. 이러한 coherence의 결과값에 대하여 주변보다 coherence값이 특히 작은 샷의 경계를 장면의 경계로 선택한다.

다음 장에서는 장면 경계 검출을 위한 새로운 샷의 추상화 방법을 제안하고, 영화나 드라마에서 나타나는 일반적인 장면의 특성 즉, 장면의 시작 부분이 아닌 경우 대부분의 샷은 이전 샷 중에서 유사한 샷을 가진다는 특성을 이용하여 장면의 경계를 찾는 방법을 제안한다.

3. 새로운 장면 검출 방법

본 장에서는 칼라 히스토그램을 이용한 샷의 추상화 방법과 이를 이용한 장면 경계 검출 방법에 대하여 설명한다.

3.1 줄거리 특성을 이용한 샷의 추상화 방법

샷 추상화 방법의 목적이 내용 기반 검색인 경우 서로 독립적인 샷들을 비교하는 것으로 검색 조건에 따라 객체, 배경, 움직임 등을 비교해야 하므로 샷의 추상화는 이들에 대한 모든 정보를 표현해야 한다. 그러나 본 논문에서 사용할 장면 경계 검출을 위한 샷의 추상화는 두 샷의 비교시 서로 독립적인 샷들을 비교하는 것이 아니라 시간적인 관계에 있는 샷들을 비교한다. 즉, 비교하는 두 샷이 서로 유사하다면 한 샷이 다른 샷보다 시간적으로 앞선 내용이 된다. 따라서 한 사람이 실내에서 춤을 추는 샷이 앞선 샷이고 같은 사람이 같은 장소에 앉아서 쉬는 샷이 뒤의 샷이라면 이 두 샷은 내용 기반 검색을 할 때는 검색 조건에 움직임이 포함된다면 다른 내용이지만 장면 경계 검출을 위해서는 같은 샷으로 분류해야 장면의 경계를 찾을 수 있다.

샷의 대표 프레임을 선택하기 위해 샷을 카메라의 움직임과 객체의 움직임에 의해 다음과 같은 4가지로 분류하였다. 첫째는 <그림 4>의 (1)과 같이 카메라와 객체의 움직임이 많지 않은 것으로 샷 내에서 등장하는 객체가 사라지거나 새로운 객체가 등장하지 않고 일정한 경우이다. 둘째는 <그림 4>의 (2)와 같이 카메라가 정지되어 있고 객체만 움직이는 경우로 샷내에서 등장하는 객체가 사라지거나 새로운 객체가 등장하는 경우이다. 셋째는 <그림 4>의 (3)처럼 카메라는 움직이는데 객체는 정지되어 있는 경우로 페닝(panning)이나 틸팅

(tilting)과 같은 카메라 기법이 사용되는 경우로서 객체가 배경이라면 배경 내의 객체는 단지 배경의 일부분이므로 움직일 수도 있다. 마지막은 <그림 4>의 (4)와 같이 카메라와 객체가 모두 움직이는 경우로 대부분 어느 한 객체의 움직임을 따라서 카메라가 움직이게 된다.

다음의 4가지 유형 중에서 첫번째 유형은 하나의 프레임만으로 추상화할 수 있지만 나머지 3가지 유형들은 하나 이상의 프레임으로 표현해야 한다. 여기에서 두 번째 예를 보면 두 남녀가 문을 열면서 들어오는데 이 샷 뒤에 다시 이와 유사한 샷이 나타난다고 해도 똑같이 문을 열고 들어오는 샷은 나타나지 않고 두 남녀가 방안으로 들어오거나 서서 무슨 말을 하는 샷이 나타날 것이다. 따라서 이 샷의 추상화를 위해서 중간 부분은 불필요하고 앞부분과 뒷부분의 프레임만으로 충분하다. 즉, 카메라나 객체가 움직이는 샷의 처음 프레임은 움직임의 시작이고 마지막 프레임은 움직임의 끝을 보여 주는데, 이들의 앞이나 뒤에 이들과 유사한 샷이 있다면 대부분의 경우 움직임의 시작부분이나 움직임의 끝부분과 유사한 샷일 것이다. 이것은 영화나 드라마에서 샷의 중간 부분의 프레임들은 카메라와 객체의 움직임에 의해 내용의 변화가 나타나는데 그 샷 뒤에 유사한 샷이 나타나더라도 똑같은 내용의 변화가 반복되지 않는다는 특성에 의해 가능한 것이고 이러한 특성을 고려할 때, 장면 경계 검출을 위한 샷의 대표 프레임은 한 개나 두 개로 표현할 수 있다.

본 논문에서는 MPEG 비디오 스트림을 대상으로 할 때 복호화를 최소화하기 위해 샷 내의 I 프레임만을 대상으로 하여 한 샷이 가지는 I 프레임 중에서 맨 앞과 맨 뒤의 프레임을 대표 프레임의 후보로 선택한다. 이러한 두 개의 I 프레임의 DC값만을 얻고 이의 칼라 히스토그램을 이용하는데, MPEG이 YUV 모델을 사용하기 때문에 별도의 변환을 수행하지 않기 위해 YUV에 대한 칼라 히스토그램을 이용한다. 이렇게 선택된 두 개의 프레임의 유사도를 측정하여 유사도가 주어진 임계값보다 크면 두 프레임은 유사하다고 판단하여 첫 번째 I 프레임 하나만을 대표 프레임으로 선택하고, 그렇지 않으면 두 개의 프레임을 모두 대표 프레임으로 선택한다. 이 때 사용한 프레임의 유사도 측정 방법은 다음 절에서 샷의 유사도 측정 방법과 함께 설명한다.

일반적으로 본 논문에서 제안한 4가지 유형 중의 어느 하나로 표현될 수 없는 샷들도 있다. 예를 들어, 한 샷이 처음에는 정지되었다가 카메라나 객체가 움직이는 샷은 둘 이상의 유형이 한 샷에 같이 나타나는 경우이다. 그러나 이런 모든 조합을 고려하여 샷들을 추상화하

면 너무 복잡한 추상화 방법을 사용하여야 하기 때문에 본 논문에서는 모든 샷들을 이런 4가지 중의 하나로 분류하여 추상화하는 방법을 사용하였다.



(1) 카메라와 객체의 움직임이 많지 않은 경우



(2) 카메라는 정지되어 있고 객체만 움직이는 경우



(3) 카메라는 움직이는데 객체는 정지되어 있는 경우



(4) 카메라와 객체가 모두 움직이는 경우

그림 5 샷의 4가지 유형

3.2 샷의 유사성 비교방법

앞에서 제안한 샷의 추상화 방법을 이용하여 두 샷의 유사성을 검사하는 측정 방법이 필요한데, 두 샷이 서로 유사하다는 것은 두 샷의 내용이 같다는 것이고 이것은 두 샷의 배경 장소와 등장 인물 또는 등장하는 객체가 같은 샷이라는 것을 의미한다. 이것은 추상화 방법에서도 설명하였듯이 내용 기반 검색이 아닌 장면 경계 검출을 위한 것이므로 움직임이 달라도 같은 객체와 같은 배경을 가지는 경우는 유사하다고 할 수 있다.

3.1절에서 제안한 샷의 추상화 방법은 한 샷에 대하여 한 두 개의 대표 프레임을 사용하므로 샷의 유사도는 대표 프레임들의 유사도의 평균값과 최대값을 고려할 수 있다. 다음 <그림 5>-(1)은 <그림 4>-(2)가 나타내는 샷의 대표 프레임이고, <그림 5>-(2)는 <그림 4>-(2)와 유사한 샷의 대표 프레임으로, 이 샷은 <그림 4>-(2) 뒤에 나타난 샷이다. 이 두 샷의 내용을 보면 샷 (1)은 방문이 닫혀 있다가 열리면서 등장 인물 두 명이 차례로 모습을 보이는 샷이고 (2)는 등장 인물 두 명이 방 안을 바라보면서 무슨 말을 하는 샷이다. 여기

에서 샷 (1)의 첫 번째 대표 프레임과 (2)의 대표 프레임의 유사도를 측정하면 높지 않지만 (1)의 두 번째 대표 프레임과 (2)의 대표 프레임은 유사도가 높다. 이처럼 샷의 대표 프레임은 샷 내 움직임에 따라 한 두 개가 선택되고 두 샷의 유사성은 샷의 대표 프레임 모두가 유사하지 않더라도 <그림 5>처럼 어느 한 쌍이 유사하다면 유사한 샷이 된다. 따라서 대표 프레임 유사도의 평균값을 사용하면 유사하지 않은 두 대표 프레임에 의해 샷의 유사도가 낮아질 수 있으므로 대표 프레임의 유사도 중 최대값을 샷의 유사도로 선택한다.



(1) <그림 4>-(2)에 있는 샷의 키 프레임



(2) 다음 샷의 키 프레임

그림 6 인접한 두 샷의 키프레임

샷의 유사도 측정 방법에서 다음으로 고려해야 할 것은 두 프레임의 유사도를 측정하는 것이다. 2개의 정지 영상의 유사도를 측정하는 방법에 대해서는 많은 연구가 진행되어 왔으며, 각 응용 분야에 맞는 다양한 방법들이 제안되어 있는 상태이다. 그런데 본 논문의 주요 연구 내용은 이미지 사이의 유사성을 계산하는 것이 아니기 때문에 가장 간단하면서 만족할만한 성능을 낼 수 있는 Color Intersection 방법[6]을 사용하여 대표 프레임들 사이의 유사도를 측정하였다. 이 방법은 일반적으로 정지 영상의 유사도를 구하기 위하여 많이 사용되는 방법이며, 또한 I-프레임의 DC 값들을 이용하여 MPEG 형식의 비디오에서 쉽게 추출하여 빠른 유사도 측정이 가능하다는 장점을 가지고 있다. Color Intersection을 나타내는 다음의 식에서 I와 J는 각각 비교 대상이 되는 두 개의 프레임을 나타내고, I_k 와 J_k 는 각각 프레임 I와 J에서 색깔 k에 해당하는 픽셀의 개수, n은 정지 영상이 가질 수 있는 모든 색깔의 개수를 나타낸다.

$$Sim(I, J) = \frac{\sum_{k=1}^n \min(I_k, J_k)}{\sum_{k=1}^n I_k} \quad (1)$$

3.3 장면 경계 검출 알고리즘

추상화된 샷과 두 샷의 유사도 측정 방법을 이용하여 장면의 경계를 찾는 방법을 설명하기 위해 일반적으로 영화에서 장면이 갖는 특성에 대하여 설명하고 이를 이용하여 장면의 경계를 찾는 알고리즘을 설명한다.

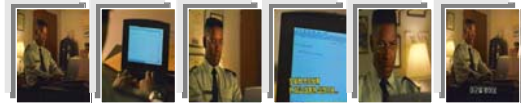
3.3.1 장면의 특성과 장면 경계 검출 원리

영화에서 장면을 [1]에서는 "하나의 시간과 공간 속에서 발생하거나 두 가지 이상의 동시적 행위를 보여주는 교차 편집을 이용하는 영화의 분절 단위"라고 정의하였다. 즉, 하나의 장면은 배경 장소와 시간이 일정한 경우도 있고, 배경 장소가 다른 곳에서 일어나는 장면을 동시에 보여 주고자 하는 경우, 또는 회상 장면을 포함하는 경우처럼 시간대가 다른 장면을 보여 주는 장면이 있다. 하지만 이러한 장면은 보는 사람에 따라 다르게 판단하는 경우가 있는 의미적인 단위이므로 컴퓨터가 인지할 수 있는 장면을 정의하기 위한 일정한 규칙이 필요하다. 따라서 이 절에서는 영화나 드라마에서 가장 일반적인 장면의 특성을 살펴 보고, 이를 위한 장면을 정의하도록 한다.

본 논문에서는 이런 장면의 구성 특성을 추출하기 위하여 많은 영화를 대상으로 장면 구성 방법을 분석하였으며, 분석 결과 다음과 같은 4가지 유형을 정의 하였다. 즉, 영화나 드라마에서 <그림 6>-(1)과 같이 등장 인물이 둘 이상인 경우 카메라는 이들이 하는 대화나 행동을 보여 주기 위해 등장 인물들을 번갈아 가면서 보여 주므로 둘 이상의 샷이 장면이 있을 수 있다. 또한, <그림 6>-(2)와 같이 등장 인물이 하나인 경우에도 그 인물을 보여 주는 샷은 여러 번 반복될 수 있는데 등장 인물이 둘 이상인 경우에도 그들을 함께 보여 주는 하나의 샷이 반복될 수도 있다. 특히 일반적인 드라마 장르의 영화는 대부분 (1)과 (2)와 같은 장면들로 이루어진다. 그러나 (1)과 (2)와 같이 샷의 내용이 반복되지 않는 경우도 있다. 이러한 예로는 장면이 어떤 사건 보다는 배경을 보여 주는 경우로 등장하는 인물이나 객체가 없거나 배경의 일부로서 작게 표현되는 경우와, 배경은 아니지만 한 장면이 한 두 개의 샷만을 포함하고 있는 경우, 그리고 등장 인물이 달리는 장면이나 차를 타고 가는 장면처럼 배경이 계속해서 바뀌기 때문에 유사한 샷이 등장하지 않는 장면 등이 포함된다. <그림 6>-(3)은 이 중에서도 장면의 내용이 배경을 나타내는 경우의 예이다.



(1) 유형 1: 둘 이상의 샷이 반복되는 예



(2) 유형 2: 하나의 샷이 반복되는 예



(3) 유형 3: 샷이 반복되지 않는 예

그림 6 장면 구성에 대한 세 가지 경우의 예

<그림 6>에서 둘 이상의 샷이 반복되는 장면을 유형 1, 하나의 샷이 반복되는 장면을 유형 2, 그리고 그 외 반복되지 않는 경우 중에서 배경을 나타내는 장면처럼 여러 개의 반복되지 않는 샷들로 구성된 장면을 유형 3으로 하였다. <표 1>은 반복되지 않는 장면 중 한 두 개의 샷으로 구성된 장면을 유형 4로 하여 여러 개의 비디오 데이터에서 각 유형에 해당하는 장면들의 개수를 정리한 것이다. <표 1>에 의하면 전체 장면의 60~80%의 장면이 샷이 반복되는 유형 1과 유형 2의 경우임을 알 수 있다. 또한 배경을 나타내는 장면이나 걸어서 뛰어서 또는 차로 이동하는 장면 등이 포함되는 유형 3의 경우와 한 두 개의 샷을 가지는 유형 4의 경우는 대부분이 길이가 짧은 장면들이다. 따라서 유형 1과 유형 2를 장면의 일반적인 경우라고 할 수 있고 이를 이용하여 장면이 가지는 규칙을 정의할 수 있다. 즉, 장면 내에서 등장하는 샷은 그와 유사한 샷이 장면 내에 존재할 가능성이 많기 때문에 장면을 서로 다른 내

표 1 비디오 데이터에서 각 유형에 해당하는 장면의 갯수

비디오종류	장면의 총개수	유형1	유형2	유형3	유형4
시트콤(17:38)	17	12	2	2	1
드라마(27:05)	24	17	0	2	5
영화1(168분)	115	79	10	14	12
영화2(104분)	75	48	2	12	13
영화3(102분)	93	55	8	11	19

용을 가지는 하나 이상의 샷이 반복되어 나타난 샷의

집합으로 정의한다.

이러한 장면의 특성을 이용하여 장면의 경계를 검출하는 방법에 대하여 예를 들어 설명하기 위해 하나는 ABABCACB...이고, 다른 하나는 ABABCDDCD...인 두 개의 스트림을 가정하자. 여기에서 알파벳은 각각의 샷을 의미하고 동일한 알파벳은 두 샷의 내용이 유사하다는 것을 의미한다. 두 스트림은 모두 다섯 번째 샷인 C까지는 동일하고 뒤에 나오는 샷들이 다른데 유사한 샷이 한 장면 내에 반복된다는 장면의 특성을 고려하면 각각 다음과 같다. 첫 번째 스트림은 C뒤의 샷 A, B가 C 이전에 유사한 샷을 가지므로 첫 번째 샷부터 마지막 샷 B까지는 모두 한 장면이 되고 따라서 다섯 번째 샷 C는 새로운 장면의 시작이 아니다. 그러나 두 번째 스트림은 C뒤의 샷들이 C 이전에 유사한 샷을 가지지 않으므로 다섯 번째 샷 C는 새로운 장면의 시작이 된다. 즉, 새로운 내용의 샷이 나타났을 때 그 샷이 새로운 장면의 시작인지 아닌지를 결정하기 위해서는 다음에 나타나는 샷들이 그 샷 이전의 샷과 유사한 것이 있는지를 검사한다. 만일 그 샷 이전에 유사한 샷이 있었다면 그 샷은 새로운 장면의 시작이 아니고 유사한 샷이 없었다면 새로운 장면의 시작으로 결정한다. 다음 절에서는 이러한 원리를 이용한 장면 경계 검출 알고리즘에 대하여 설명한다.

3.3.2 알고리즘

장면의 경계를 검출하기 위해 가장 먼저 해야 할 일은 샷 경계를 검출인데 본 논문에서는 본 연구의 선행 연구에서 제안한 방법[8]을 사용하였다. 이러한 샷 경계 검출 방법에 의해 얻을 수 있는 샷의 리스트를 이용하여 장면 경계 검출을 수행하는데 각 샷을 추상화하여 대표 프레임을 선택하고 대표 프레임의 칼라 히스토그램과 두 샷의 유사도 측정 방법을 이용한다. 3.3.1절에서 설명한 장면 경계 검출을 수행하는 알고리즘은 다음 <그림 7>과 같다.

<그림 7>의 첫 번째 for문은 현재 샷이 이전에 나타났던 샷인지 처음 나타나는 샷인지를 판단하기 위한 것으로 두 번째 샷부터 시작하여 현재 샷 i에 대하여 샷 i 이전에 나타난 샷 중에서 유사도가 가장 높은 샷을 찾는다. 찾은 샷의 유사도가 주어진 임계값보다 크면 두 샷은 유사하다고 판단하여 샷 i 이전의 샷 중 가장 유사한 샷의 번호를 갖는 배열 similar_shot[]의 i번째에 샷 번호를 할당하고, 임계값보다 작으면 유사한 샷이 없다는 의미로 -1을 할당한다. <그림 7>의 두 번째 for문은 첫 번째 for문의 결과로 얻어진 배열 similar_shot[]을

```

procedure SCENE_CHANGE_DETECTION()
  SHOT shot[N]; // 추상화된 샷의 리스트
  integer similar_shot[N];
  integer scene[N]; // 각 샷이 포함된 장면의 번호를 갖는 배열
  integer snum, sceneNum, index, i, j, flag;
  float max;
  // first for loop => 현재 샷과 가장 유사한 이전 샷을 찾는다.
  similar_shot[1] = -1;
  for snum <- 2 to TOTAL_SHOT do
    max = 0.0;
    for i<-snum-WINDOW to snum-1 do
      if ShotSim(shot[snum], shot[i])> max
        then max=ShotSim(shot[snum],shot[i]);
          index = i;
        endif
      endif
    endfor
    if max > Threshold
      then similar_shot[snum] = index;
        else similar_shot[snum] = -1;
      endif
    endif
  endfor
  // second for loop => 현재 샷이 새로운 장면의 시작여부 검사
  SceneNum = 1;
  for snum <- 2 to TOTAL_SHOT do
    flag = FALSE;
    if similar_shot[snum] = -1 then
      for i<-snum+1 to snum+1+WINDOW do
        if similar_shot[i] < snum
          then scene[snum] <- sceneNum;
            flag <- TRUE;
            break;
          endif
        endif
      endfor
    if flag = FALSE
      then scene[snum] <- sceneNum+1;
        sceneNum <- sceneNum+1;
      endif
    endif
  endfor
end SCENE_CHANGE_DETECTION

```

그림 7 제안한 알고리즘

이용하여 장면의 경계를 검출하는데 similar_shot[i]의 값이 -1이면 샷 i가 이전에 유사한 샷이 없는 새로운 샷이므로 새로운 장면의 시작일 가능성이 있다. 현재 샷과 이전 샷을 비교할 때 시간적으로 너무 떨어진 두 샷은 같은 장면이 아닐 확률이 크므로 이전의 모든 샷과 비교하지 않고 제한된 크기의 윈도우 크기 내의 샷과만 비교한다. 마찬가지로 scene[]의 값을 구하는 두 번째 for문에서도 similar_shot[i]의 값이 -1일 때 뒤에 나타나는 샷의 similar_shot[] 값을 검사할 때 제한된 크기의 윈도우 크기 내의 샷만을 검사한다. 이 알고리즘의 결과로서 scene[]은 각 인덱스 번호의 샷이 포함되는

장면번호를 갖는 배열이다.

3.3.3 알고리즘의 문제점 분석

앞에서 설명한 알고리즘에 의해 장면 경계를 검사했을 때 문제가 되는 경우의 예로 <그림 8>는 샷 #1부터 #8까지 한 장면이고 나머지 샷 #9와 #10은 다른 장면이다. 여기에서 첫 번째 장면의 샷 #8은 이전에도 등장하지 않았고 뒤에도 등장하지 않는 샷이지만 앞의 샷들과 배경이 같으므로 같은 장면이다. 그러나 이것을 위 알고리즘에 의해 장면 경계를 찾으면 샷 #1부터 #7까지 한 장면, 샷 #8을 한 장면, 나머지 샷 #9와 #10을 한 장면으로 찾아서 총 3개의 장면을 찾을 것이다.



그림 8 장면 경계를 잘못 찾은 예

본 논문에서 제안한 알고리즘은 <그림 8>에 나타난 예제와 같이 장면의 처음이나 마지막 부분에 다른 내용의 샷이 등장하는 경우 장면의 경계가 아닌데도 경계로 잘못 찾게 되는 문제가 발생한다. 앞 절에서 장면을 정의할 때 샷이 반복되어 나타난다고 가정하였으므로 한 장면이 세 개 이상의 샷을 포함한다고 제한할 수 있고, 실제로 일반적인 장면은 세 개 이상의 샷을 가지고 있다. 따라서 장면 내에 샷이 한 두 개인 경우에는 그 장면의 이전 장면과 이후 장면과의 유사도를 측정하여 유사도가 더 큰 장면에 포함시킴으로써 위 문제를 해결할 수 있다. 이 때 두 장면의 유사도 측정은 두 장면이 갖는 모든 샷끼리 유사도를 계산하여 이들의 평균값을 사용하였는데 이에 대한 식은 다음 <식 2>와 같다. 이 식에서 Scn_i 와 Scn_j 는 유사도를 측정할 두 장면이고, N_i 와 N_j 는 각각 Scn_i 와 Scn_j 가 포함하는 샷의 개수, $Scn_i(k)$ 와 $Scn_j(l)$ 은 각 장면 Scn_i 가 가지는 샷 중 k 번째 샷과 Scn_j 가 가지는 샷 중 l 번째 샷을 의미한다.

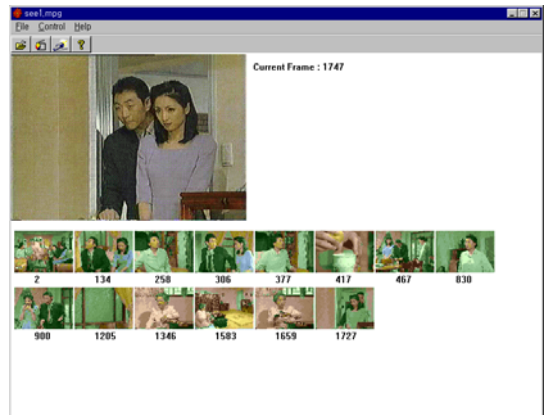
$$SceneSim(Scn_i, Scn_j) = \frac{1}{N_i N_j} \sum_{k=1}^{N_i} \sum_{l=1}^{N_j} ShotSim(Scn_i(k), Scn_j(l)) \quad (2)$$

이러한 과정은 장면의 경계가 아닌데도 장면의 경계로 잘못 찾은 장면들에 대해 원래 장면으로 묶는 과정이기 때문에 장면의 필터링이라고 할 수 있다. 또한 배경은 비슷하네 등장하는 객체가 반복되지 않는 장면에 대해 3.3.2절의 알고리즘을 수행하면 여러 개의 다른 장면으로 찾게 되는데 장면의 필터링을 수행하면 배경이 비슷하므로 같은 장면으로 찾을 수 있다.

4. 성능 평가 및 분석

4.1 시스템 구현

본 논문에서는 제안한 장면 경계 검출 알고리즘을 Windows95상에서 Visual C++를 이용하여 구현하였다. 구현한 장면 경계 검출기에서 샷 경계 검출은 [8]에서 제안한 방법을 사용하였는데 장면 경계 검출의 사용자 인터페이스는 <그림 9>와 같다. <그림 9>-(a)는 사용자의 입력으로 MPEG 스트림을 선택하고, 선택된 스트림을 상영하면서 검출된 샷의 첫번째 프레임들을 보여주는 수행 화면 예제이고, <그림 9>-(b)는 메뉴 중 Detect를 선택하면 각 장면별로 포함하는 샷들을 그림



(a) 샷 경계 검출기 수행 결과 화면



(b) 장면 경계 검출 (샷 그룹핑) 결과 화면

그림 9 장면 경계 검출 수행 화면

평하여 시간적 순서에 따라 사용자에게 보여주는 수행 화면 예제이다.

4.2 실험 및 분석

본 논문에서 구현한 장면 경계 검출 알고리즘의 테스트에서 대표 프레임 선택시, 샷의 유사도 측정시, 그리고 사용한 윈도우의 크기에서 3개의 임계값이 필요한데 이러한 임계값은 실험에 의해 결정하였다. 실험에 사용된 데이터는 모두 352x240 크기의 MPEG-1 시스템 스트림으로 시트콤, TV 드라마, 영화 3개 총 5개의 스트림을 가지고 하였다.

다음 <표 2>는 실험 데이터에 대해 가장 좋은 실험 결과를 나타내는 임계값을 적용하여 장면 경계 검출 알고리즘을 수행한 결과이다. 여기에서 실제 사용한 장면의 개수는 본 논문에서 제안한 알고리즘의 가정이 한 장면이 3개 이상의 샷을 가진다는 것이므로 한 두 개의 샷으로만 이루어진 장면은 제외한 개수이다. 실제로 비디오 데이터에서 한 두 개의 샷으로 구성된 장면의 개수는 3.2절의 <표 1>을 보면 알 수 있다. <표 2>의 리콜(recall)과 프리시전(precision)을 구하는 식은 다음 <식 3>과 같다. <식 3>을 보면 리콜은 찾지 못한 결과를 측정하는 기준이고, 프리시전은 잘못 찾은 결과를 측정하는 기준임을 알 수 있다.

이 실험 결과에서 장면의 경계가 아닌데도 장면의 경계로 잘못 찾은 경우는 <그림 10>-(a)처럼 같은 장면이지만 등장하지 않았던 새로운 샷 여러 개가 연속적으로 등장한 경우와, <그림 10>-(b)와 같이 앞에서 나타났던 샷과 유사한 샷이지만 객체의 크기가 달라져서 다른 칼라 히스토그램을 갖게 되어 Color Intersection 방법을 이용한 샷의 유사도가 낮게 나오는 경우 두 가지로 나눌 수 있다. <표 3>은 이 두 가지 경우에 대한 장면의 개수를 나타낸다. 이 중에서 첫 번째는 장면 경계 검출을 위해 가정한 일반적인 장면의 특성을 만족하지 않기 때문이고, 두 번째는 프레임의 유사도 측정 방법이 객체의 크기에 민감하기 때문이다. 이 중 객체의 크기 변화에 따라 프레임의 유사도가 낮게 나오는 문제는 단순한 칼라 히스토그램이 아니고 객체와 배경을 구별하여 객체의 칼라 히스토그램과 배경의 칼라 히스토그램을 따로 이용하면 해결할 수 있을 것이다. 또한 장면의 경계인데도 장면의 경계로 찾지 못한 경우는 <그림 10>-(c)와 같이 연속된 두 장면이 칼라 히스토그램이 비슷한 샷을 포함하고 있는 경우로서 이 두 샷이 어느 정도의 거리를 유지하고 있다면 그 거리보다 작은 윈도우를 사용함으로써 장면의 경계를 찾을 수 있을 것이다. 또한 이 경우는 객체와 배경은 다르지만 전체적인

프레임의 칼라 히스토그램이 유사해서 유사한 샷으로 찾은 경우도 있는데 이것은 객체의 배경의 칼라 히스토그램을 따로 이용하는 방법으로 해결할 수 있다.

표 2 장면 경계 검출기 수행결과

비디오 종류	프레임 갯수 (시간 m:s)	샷 갯수	장면의 갯수	실험결과				
				찾은 장면 갯수	잘못 찾은 갯수	찾지 못한 갯수	Recall	Precision
시트콤 (남자셋 여자셋)	31735 (17:38)	173	16	16	0	0	1.0	1.0
드라마 (보고 또 보고)	48724 (17:05)	289	19	21	1	1	0.95	0.95
영화1 (Dying Young)	38428 (21:22)	168	17	19	3	0	1.0	0.85
영화2 (Sliding Doors)	91843 (51:02)	725	48	53	11	6	0.86	0.79
영화3 (City of Angel)	89887 (50:00)	651	34	37	10	7	0.79	0.77

	Relavant	Not Relavant
Retrieved	A Correctly Retrieved	B Incorrectly Retrieved
Not Retrieved	C Missed	D Correctly Rejected

$$Recall = \frac{\text{relavant retrieved}}{\text{all relavant}} = \frac{A}{A + C}$$

$$Precision = \frac{\text{relavant retrieved}}{\text{all retrieved}} = \frac{A}{A + B} \tag{3}$$

표 3 장면을 잘못 찾은 경우의 구분

	시트콤	드라마	영화1	영화2	영화3
새로운 샷이 연속적으로 등장하는 경우	0	2	1	4	4
유사한 샷이지만 객체의 크기가 다른 경우	0	1	2	8	6
합계	0	3	3	12	10

다음 <표 4>는 본 논문에서 제안한 알고리즘에서 장면의 필터링 과정의 유용성을 보이기 위해 <표 2>에서 정리한 실험 결과와 필터링을 수행하지 않았을 때의 실험

험 결과를 비교한 것이다. 장면의 필터링은 장면의 경계가 아닌데도 장면의 경계로 잘못 찾는 경우의 수를 감소시키기 위한 것으로 찾지 못한 경우를 평가하는 리콜은 변하지 않고 잘못 찾는 경우를 평가하기 위한 프리시전에만 영향을 미친다. 따라서 <표 4>에서는 필터링을 한 경우와 안 한 경우의 프리시전을 비교하였다. 실제로 샷이 반복되는 유형의 장면이라고 해도 <그림 8>의 예처럼 반복되지 않는 부분을 포함하고 있는 경우가 많기 때문에 필터링을 안 한 경우의 프리시전이 훨씬 낮음을 알 수 있다.



(a) 새로운 샷이 연속적으로 등장하는 경우



(b) 유사한 샷이지만 객체의 크기가 다른 경우



(c) 두 장면이 서로 유사한 칼라 히스토그램을 갖는 샷을 포함한 경우

그림 10 잘못된 결과를 나타내는 예

표 4 필터링 유무에 따른 Precision 비교

	시트콤	드라마	영화1	영화2	영화3
필터링을 한 경우	1.0	0.947	0.85	0.792	0.768
필터링을 하지 않은 경우	0.612	0.546	0.513	0.418	0.489

4.3 관련 연구와의 비교

본 논문에서 제안한 방법은 샷을 추상화할 때 한 샷 내의 한 두 개의 프레임에 대한 칼라 히스토그램을 사

용하였기 때문에 샷 내에 여러 개의 프레임을 사용하거나 움직임 정보를 사용하는 기존의 방법[3,4,5,7]보다 더 적은 정보를 이용한다. 실제로 샷의 추상화를 Yeung의 방법과 Rui의 방법을 사용하여 제안한 장면 경계 검출 알고리즘에 적용한 실험 결과를 리콜(R)과 프리시전(P)을 이용하여 <표 5>에 정리하였다.

표 5 샷의 추상화 방법에 따른 실험결과

비디오 종류	Yeung의 추상화 방법		Rui의 추상화 방법		제안한 방법	
	Recall	Precision	Recall	Precision	Recall	Precision
시트콤	1.0	1.0	1.0	0.941	1.0	1.0
드라마	0.894	0.85	0.947	0.857	0.947	0.947
영화1	0.823	0.836	0.882	0.833	1.0	0.85
영화2	0.854	0.743	0.814	0.735	0.893	0.792
영화3	0.763	0.735	0.723	0.698	0.794	0.768

Yeung의 추상화 방법은 한 샷에 대해서 움직임이 많을수록 많은 프레임에 대표 프레임으로 선택하였고, 샷의 유사도 측정은 대표 프레임들간의 프레임 유사도 중 가장 큰 값을 사용하였다. 또한, Rui의 추상화 방법은 한 샷의 처음과 마지막 프레임을 대표 프레임으로 선택하고 샷의 움직임 정도를 사용하였으며 샷의 유사도 측정은 대표 프레임의 프레임 유사도값 중 가장 큰 값과 움직임 정도의 유사도값에 각각 가중치를 두어서 측정하였다. <표 5>의 결과를 보면 Yeung과 Rui의 추상화 방법을 이용하였을 때 리콜과 프리시전은 제안한 추상화 방법을 이용하였을 때보다 작으므로 더 좋은 결과를 가지지 못 한다. 따라서 움직임에 따른 정보는 장면을 묶는데 유용한 정보가 아님을 알 수 있다. 다음 <표 6>은 장면 경계 검출을 위해 본 논문에서 제안한 방법과 기존의 방법을 비교하여 정리한 것이다.

5. 결론

디지털 라이브러리나 VOD(Video on Demand)와 같은 디지털 비디오 데이터를 이용한 응용 프로그램을 개발할 때에는 비디오 데이터의 내용(Content) 정보를 입력하기 위해 먼저 비디오 데이터에 대한 인덱싱을 해야 한다. 본 논문에서는 MPEG 비디오 스트림에 대하여 장면단위 인덱싱을 수행하는 새로운 장면 경계 검출 알고리즘을 제안하였다. 제안한 장면 경계 검출 방법은 샷의 추상화를 위해 MPEG 스트림에서 I 프레임의 DC값

표 6 관련 연구와의 비교

연구자	샷의 추상화		샷의 유사성 비교방법		샷들을 장면으로 묶는 방법
	키프레임 갯수	움직임 정도	키 프레임 유사도	샷 유사도	우선 한 샷만 포함하는 그룹을 생성하고 전체 그룹 중 가장 유사한 두 그룹을 더 이상 유사한 그룹이 없을 때까지 묶는다 ⇒ 그룹들의 시간관 계를 이용하여 장면 경계 검출
Yeung [4,5]	여러개 사용	×	Color Intersection [6]	대표 프레임의 유사도의 최대값	Coherence를 사용하여 검출
Kender [3]					
Rui [7]	처음과 마지막 프레임	○		대표 프레임 유사도의 최대값 + 움직임 정도 유사도	처음 샷부터 순서대로 기존의 그룹과 유사하면 그 그룹에 포함시키고 그렇지 않으면 새로운 그룹을 생성하여 기존의 장면과 유사하면 그 장 면에 포함시키고 그렇지 않으면 새로운 장면을 생성하여 검출
제안한 방법	1-2개	×		대표 프레임의 유사도의 최대값	윈도우 크기 내에서 이전 샷에 비해 유사도가 낮고, 뒤에 나타난 샷들이 현재의 샷이후 샷들과만 유사하면 장면의 경계, 잘못 찾은 장면의 개수 를 줄이기 위한 필터링 수행

만을 이용함으로써 복호화를 최소화하고, 추상화된 샷을 이용한 장면 경계 검출을 위해서 기존의 방법보다 오버헤드가 적은 연산 과정을 수행한다. 본 연구에서 제안한 장면 단위의 인덱싱 방법은 MPEG 비디오 데이터 스트림에 대하여 최소한의 복호화를 통하여 장면 단위의 인덱싱을 수행할 수 있기 때문에 MPEG 형식의 압축 비디오 데이터를 이용한 응용 프로그램의 비디오 데이터 베이스 구축에 유용하게 사용될 수 있다.

앞으로의 연구에서는 샷을 추상화할 때 단순한 칼라 히스토그램이 아니고 키 프레임의 객체와 배경을 구별하여 객체의 칼라 히스토그램과 배경의 칼라 히스토그램을 따로 이용하면 더 정확한 장면 경계 검출이 가능하므로 이에 대한 연구가 이루어질 것이다. 이러한 연구는 두 샷의 유사도 측정 시에도 같은 내용의 샷이 포함하는 객체의 크기가 많이 다른 경우에는 유사도가 낮게 나오는 문제점을 해결할 수 있을 것이다. 그리고 지금까지는 비디오 데이터만을 이용하여 장면의 경계를 검출하였는데 여기에 오디오 정보를 이용한 장면 경계 검출 방법을 추가시킴으로써 더 정확하게 장면의 경계를 찾을 수 있을 것이다.

참 고 문 헌

- [1] D. Bordwell and K. Thompson, *Film Art : An Introduction*, McGraw-Hill, 1993.
- [2] H. J. Zhang, Y. Gong, and S. W. Smoliar, "Automatic Parsing of News Video," *Proceedings of IEEE Conference on Multimedia Computing Systems*, pp. 45-54, 1994.
- [3] J. R. Kender and B. L. Yeo, "Video Scene Segmentation Via Continuous Video Coherence," *Proceedings of Computer Vision and Pattern Recognition*, 1998.
- [4] M. Yeung, B. L. Yeo, and B. Liu, "Extracting Story Units from Long Programs for Video Browsing and Navigation," *Proceedings of International Conference on Multimedia Computing and Systems*, 1996.
- [5] M. Yeung, B. L. Yeo, and B. Liu, "Segmentation of Video by Clustering and Graph Analysis," *Computer Vision and Image Understanding*, 1998.
- [6] M. Swain and D. Ballard, "Color Indexing," *International Journal of Computer Vision*, 1991.
- [7] Y. Rui, T. S. Huang, and S. Mehrotra, "Exploring Video Structure Beyond the Shots," *Proceedings of International Conference on Multimedia Computing and Systems*, pp. 237-240, 1998.
- [8] 임영인, 남중호, "매크로블록 정보를 이용한 MPEG-1 비디오 스트림의 장면 변화 검출", *한국정보과학회 논문지 A*, 26권 4호, 1999년.
- [9] N. Gamaz, X. Huang, and S. Panchanathan, "Scene Change Detection in MPEG Domain," *Proceeding of 1998 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 12-17, 1998.
- [10] V. Kobla, D. Doermann and K. I. Lin, "Archiving, Indexing, and Retrieval of Video in the Compressed Domain," *Proceeding of SPIE Conference on Storage and Retrieval for Still Image and Video Databases*, SPIE Vol. 2916, pp. 78-89, 1996.
- [11] J. Meng, Y. Juan and S. F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," *Proceeding of SPIE Conference on Digital Video*

Compression: Algorithms and Technologies, SPIE Vol. 2419, pp. 14-25, 1995.

- [12] B. L. Yeo and B. Liu, "On the Extraction of DC Sequences from MPEG Compressed Video," *Proceedings of International Conference on Image Processing*, Vol. II, pp. 260-263, 1995.
- [13] H. J. Zhang, C. Y. Low and S. W. Smoliar, "Video Parsing and Browsing using Compressed Data," *Multimedia Tools and Applications*, Vol. 1, pp. 89-111, 1995.



이 숙 경

1997년 서강대학교 전자계산학과 졸업(학사). 1999년 2월 서강대학교 대학원 전자계산학과 졸업(석사). 1999년 2월 ~ 현재 현대 전자 연구원. 관심 분야는 멀티미디어 시스템, 멀티미디어 통신, 인터넷 프로그래밍 등



남 종 호

1986년 2월 서강대학교 전자계산학과 졸업(학사). 1988년 2월 한국과학기술원 전산과 졸업(석사). 1992년 2월 한국과학기술원 전산과 졸업(박사). 1992년 3월 ~ 1992년 8월 한국과학기술원 정보전자 연구소(연구원). 1992년 9월 ~ 1993년 8월 일본 Fujitsu연구소(방문연구원). 1993년 9월 ~ 현재 서강대학교 전자계산학과 부교수. 관심분야는 멀티미디어 시스템, 병렬 프로그램 언어 및 시스템, 인터넷 프로그래밍 및 응용.