DBpia

방송영상 편집을 위한 고속 자동장면 전환검출 방식의 설계 및 구현

정회원 송 한 춘*, 김 진 용**

Development of High speed Automatic shot Boundary Detection for broadcasting video editing

Song Han Chun*, Kim Jin Young** ***Regular Members***

요 약

대용량의 방송 동영상 데이터를 효율적으로 색인하고 편집하기 위한 고속의 자동 장면전환검출 시스템이 요구되고 있다. 본 논문에서는 압축된 방송영상 데이터를 선택적으로 추출하여 8×8 비트 맵 배열구조에 최소 복호화하고 각 장면의 색상 값의 합을 구하고 다음 장면과의 차이의 정도를 가지고 장면전환을 검출하는 방식의 시스템을 제안하고, 제안한 장면전환검출 방식의 시스템을 설계 및 구현하여 실제 동영상 데이터를 적용하여 장면검출성능을 평가하였다. 제안한 장면전환 검출방식은 다양한 형태의 동영상 압축포맷에서 고속으로 프레임의 체크가 가능하였고, 기존 검출방식에 비하여 4배 이상 빠른 검출성능을 나타내었다.

Key Words : Scene change detection, Video editing

ABSTRACT

Scene change detection systems, which is very important for video indexing and editing to a large broadcasting video data. In this paper, we propose fast and accurate scene change detection method. The process of proposed scene change detection method is like this, first selective data extraction from compressed video, and the data is applied to 8*8 bitmap array structure with minimal decoding, second, calculated the sum of color space value for each scene, third, scene change is detected with difference of color space value in each scene. We also design and implement that proposed scene change detection system, and evaluate the performance of that system. Proposed method system appears 4 times higher frame check speed than existing method system, and can be applied various types of video compression format data.

Ⅰ. 서 론

최근 지상파방송을 포함하여 케이블방송, 위성디지털방송, 인터넷방송 등 다양한 형태의 멀티미디어 방송서비스가 등장하여 제공되고 있다. 이들 방송을 위한 각종 뉴스, 영화, 오락프로그램 등 다양한 형태의 대용량 방송영상물도 급속히 증가하고 있으며, 이러한 대용량의 방송영상물을 효율적으로 관리하고 저장하고 편집하기 위한 아카이빙 시스템이 요구되고 있다[1-3].

대용량의 방송영상물을 효율적으로 관리하기 위해서는 고속으로 장면전환을 검출하여 편집할 수 있는 기술이 요구된다. 일반적인 비디오 데이터의 색인 및 검색을 위해서는 먼저 비디오 시퀀스를 시

※ 본 논문은 2007년도 서일대학 학술연구비에 의해 연구되었음.
* 서일대학 정보통신과 부교수(sanho@seoil.ac.kr), ** (주)케레스인포 기술연구소 소장
논문번호: 08073-1122, 접수일자: 2008년 11월 22일

www.dbpia.co.kr

간적으로 동일한 성질을 갖는 부분 즉 샷(shot)으로 분할하고 분할된 샷에서 대표 프레임인 키 프레임(key frame)을 추출하여 키 프레임의 유사성을 판단하여 장면전환을 검출하게 된다[4]. 장면전환은 장면사이의 전이가 발생하는 것을 말한다. 장면전환이 발생하기 전과 후의 영상은 서로 다른 장면으로 구성된다. 장면전환은 동영상 자료의 편집으로 인해 생성되는 것으로 컷은 인접한 영상사이에서 장면이 급변하는 전환을 말한다[5-10].

본 논문에서는 압축된 방송영상 비디오의 일부를 선택적으로 추출하여 8*8 비트맵 배열구조에 적용하는 최소 복호화 방법과 장면 사이의 색상 값의 합의 차이를 이용한 장면전환(Scene change detection)방식을 적용하여 방송 영상물을 보다 빠르고 정확하게 장면전환을 검출하는 방식을 제안하였고, 제안한 장면전환 검출방식을 적용한 시스템을 설계 구현한 다음 방송 동영상 데이터를 가지고 그 성능을 평가 분석하였다.

Ⅱ. 장면 간의 색상 값 합의 차이를 이용한 장면전환 검출방식 고찰

장면전환 검출방법으로는 비트열을 모두 디코딩하여 완전한 이미지를 복원한 후 검출 알고리즘을 적용하는 비압축 영역에서의 장면전환 검출방법과 필요한 만큼의 비트열만을 선택적으로 디코딩한 후 얻어진 데이터에 검출알고리즘을 적용하는 압축영역에서의 장면전환 검출방법이 있다[6]. 압축영역에서의 장면전환 검출방법은 비트열을 디코딩하는데 소요되는 시간을 줄일 수 있고, 다루게 되는 데이터량이 적어지기 때문에 효율적으로 적용할 수 있는 장점이 있다[7].

본 논문에서는 장면전환이 일어날 때 장면의 컬러나 밝기가 이전 프레임과는 매우 큰 차이가 나타나는 점을 추출하여 장면전환을 검출하는 방식을 적용한다. 본 논문에서는 장면전환을 나타내는 특징을 추출하기 위하여 샷(shot) 사이의 색상 값의 합의 변화를 사용한다. 보통 칼라영상은 RGB 칼라로 제공된다. 이러한 칼라영상에 대한 영상처리로는 RGB칼라모델을 그대로 사용하는 방법과 HSI 칼라 모델로 변환하여 처리하는 방법이 있다. 칼라영상에 대한 경계선 검출도 RGB 칼라 공감에서 수행할 수도 있고 HSI 칼라 공간에서 수행할 수도 있다. RGB 칼라공간에서의 경계선을 검출하기 위해서는 RGB 각 성분에 대하여 경계선 검출을 위한 회선 마스크를 수행한 다음 그 결과 값을 다음 식(1)과 같이 합하면 된다. HSI 칼라공간에서의 경계선 검출은 단지 RGB 칼라를 HSI 칼라로 변환한 다음에 명도 성분에 대하여 회선을 수행하면 된다[8].

$$E(x,y) = \sqrt{Ered^2(x,y) + Egreen^2(x,y) + Eblue^2(x,y)} \tag{1}$$

장면전환 중 컷은 한 장면에서 다른 장면으로 급전하는 변환이므로 컷을 기준으로 이전 장면과 다음 장면 사이의 유사성은 매우 낮다. 따라서 장면사이의 색상, 명암, 채도의 변화는 매우 심하게 나타난다. 특히 서로 다른 장면일수록 명암이나 채도에 비해 색상의 변화가 심하다. 이런 특성을 기반으로 컷 장면전환에 대한 특징은 식(2)와 같이 정의할 수 있다[6].

$$F_{corr} = \alpha^* H_{corr} + \beta^* I_{corr} \tag{2}$$

$$H_{corr} = \frac{1}{2}\left(1 + \frac{\sum_{i=1}^{N}(BHM_{t-1}^{i} - HM_{t-1})(BHM_{t}^{i} - HM_{t})}{\sqrt{\sum_{i=1}^{N}(BHM_{t-1}^{i} - HM_{t-1})}\sqrt{\sum_{i=1}^{N}(BHM_{t}^{i} - HM_{t})}}\right)$$

$$I_{corr} = \frac{1}{2}\left(1 + \frac{\sum_{i=1}^{N}(BIM_{t-1}^{i} - IM_{t-1})(BIM_{t}^{i} - IM_{t})}{\sqrt{\sum_{i=1}^{N}(BIM_{t-1}^{i} - IM_{t-1})}\sqrt{\sum_{i=1}^{N}(BIM_{t}^{i} - IM_{t})}}\right)$$

$$where \alpha, \beta : weights(0 < \alpha \le \beta < 1, \alpha + \beta = 1)$$

식(2)에서 컷 특징 F_{corr}은 현재 시점과 이전 시점 영상사이의 색상과 명암 값의 상관관계를 나타내는 H_{corr}과 I_{corr}으로 정의하며, 0에서 1사이의 범위를 갖도록 정규화 한다. 컷 특징 F_{corr}이 0에 근접할수록 컷일 확률이 높고, 반대로 1에 근접할수록 컷일 확률이 낮다. 식(2)에서 BHM_t^i와 BIM_t^i는 현재 시점의 I번째 블록의 색상 평균과 명암값 평균을 의미하며, HM_t^i과 IM_t^i는 현재 시점 영상의 색상 평균과 명암 평균을 의미한다.

장면전환 중 페이드인은 명암 값이 증가하면서 고정영상을 점진적으로 장면화 하는 전환이다. 그리고 페이드아웃은 명암 값을 감소하면서 장면을 고정 영상으로 점진적으로 변화시키는 장면전환이다. 따라서 페이드 장면전환이 발생할 경우에는 인접한 영상 사이의 색상이나 채도의 변화보다는 명암값의 변화가 현저히 발생한다. 이런 사실을 근거로 식(3)과 같은 명암 값의 차이의 비율을 이용하여 페이드인과 페이드아웃의 특징을 정의할 수 있다.[6]

www.dbpia.co.kr

$$F_{ratio} = \frac{I_{diff}}{I_{Adiff}} \quad (3)$$

$$I_{diff} = \frac{\sum_{t=1}^{N} w_{i.} \left(\sum_{k=1}^{n} \sum_{l=1}^{n} BI_t^i(x+k, y+l) - BI_{t-1}^i(x+u_i+k, y+v_i+l) \right)}{n^2 . N . I_{\max}}$$

$$I_{Adiff} = \frac{\sum_{t=1}^{N} w_{i.} \left(\sum_{k=1}^{n} \sum_{l=1}^{n} |BI_t^i(x+k, y+l) - BI_{t-1}^i(x+u_i+k, y+v_i+l)| \right)}{n^2 . N . I_{\max}}$$

식(3)에서 페이드의 특징 F_{ratio}는 현재 시점과 이전 시점영상 사이의 명암 값의 차이와 명암 값의 절대 값의 차이의 비율로 정의되며, -1에서 1사이의 값을 가진다. 페이드 특징 F_{ratio}가 -1에 근접할수록 페이드 아웃일 확률이 높으며, 반대로 F_{ratio}가 1에 근접할수록 페이드인일 확률이 높다[6].

Ⅲ. 고속 자동장면전환 검출구조의 설계 및 구현

3.1 제안한 장면전환검출 방식설계

본 논문에서 제안한 장면 간의 색상 값 합의 차이 정도를 이용한 장면전환검출방식의 성능을 평가하기 위하여, 제안한 시스템을 설계 및 구현하였다. 제안한 고속 자동장면전환검출 시스템을 설계함에 있어서 MS windows 운영체제 환경에서 현재 유통되고 있는 모든 형태의 동영상 파일에 적용될 수 있도록 Microsoft 사에서 제공하는 Direct-Show 필터를 활용하도록 설계하였다. 또한 MS Windows 플랫폼 상 Media Player의 구동에 있어서, 컴퓨터의 그래픽카드(VGA)의 메모리에 직접 액세스 하는 대신에 다양한 필터 인터페이스를 통하여 액세스 할 수 있게 설계하여, 영상 코덱을 자유롭게 제작하여 사용할 수 있도록 하였으며, 다양한 형태의 영상 코덱을 지원할 수 있도록 설계하였다. 또한 프레임에 필요한 크기로 가상 디바이스를 만들어 압축하여 플레이함으로서 메모리의 절대 사용량을 줄이고, 영상자료의 분석에 소요되는 시간을 최소화함으로서 고속으로 장면전환의 검출이 되도록 설계하였다.

3.2 제안한 장면전환검출 방식의 검출과정

본 논문에서 제안한 방식의 자동 장면전환검출 과정은 다음과 같다. 먼저 사용자가 콘솔을 이용하여 영상파일과 장면전환 결정을 위한 색상 값의 합의 차이인 임계값을 전달하면, 방송영상 자료를 Direct-Show 필터를 이용하여 플레이 하게 된다. 이때 플레이는 실제 창이 아니라, 8x8의 가상 디바이스(Device) 상에서 플레이함으로써 실제 화면 출력을 위한 오버헤드를 최소화 하고, 가상 디바이스 상에서 플레이되는 영상을 버퍼(Bitmap Array)에 프레임별로 저장한다.

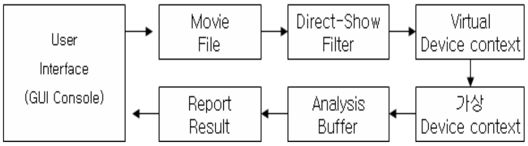


그림 1. 제안한 자동장면전환검출방식 처리과정

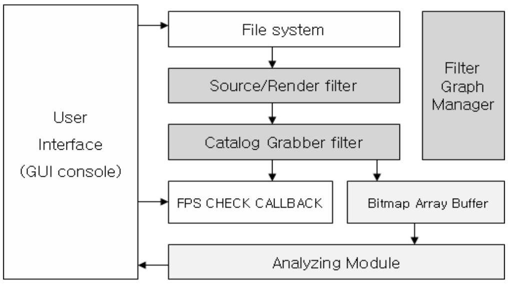


그림 2. 제안한 장면전환 검출시스템 기능모듈 구조

플레이가 완료된 이후 버퍼를 분석하여 영상의 장면변환을 검출하여 그 결과 값을 사용자 인터페이스에 반환하면 처리과정이 완료가 된다.

본 논문에서 제안한 장면검출시스템의 기능구조는 그림 2와 같이 사용자 인터페이스모듈, 분석모듈, 미디어필터모듈의 세 개의 모듈로 구성된다.

사용자 인터페이스(User Interface) 모듈은 모든 인터페이스를 담당하는 모듈이다. 인터페이스는 CLI (Command Line Interface), GUI (Graphic User Interface)를 통하여 직접 입력하는 인터페이스 또는 공유 메모리(Shared Memory Map)를 이용하여 다른 프로그램과 직접 통신 가능하도록 설계하였다. 사용자 인터페이스 모듈은 파일시스템(File System)을 이용하여 영상 파일을 미디어 필터 모듈로 전달하고, 분석(Analyzing)모듈을 거쳐 분석된 결과 데이터를 사람이 읽을 수 있는 값으로 반환하거나 호출한 프로그램에 전달하는 역할을 수행한다. 위의 그림3-2에서 필터들로 구성되는 미디어필터 모듈은 Direct-Show 필터를 이용하여 데이터를 프레임(frame) 별로 비트 맵 어레이(Bit-map array)버퍼에 복제하는 일종의 파스(Parse)모듈이다. 미디어필터 모듈은 영상이 입력되면, 파일 시스템을 이용하여 데이터를 읽어 들이면서, 소스(Source)필터와 렌더(Render)필터를 이용하여 영상의 파스(Parse)에 필요한 인터페이스로 자동으로 연결한다. 예를 들어 MPEG2 영상이 입력된 경우, 시스템을 검색하여 MPEG2를 Parse할 수 있는 코덱이 있는지 확인한

www.dbpia.co.kr

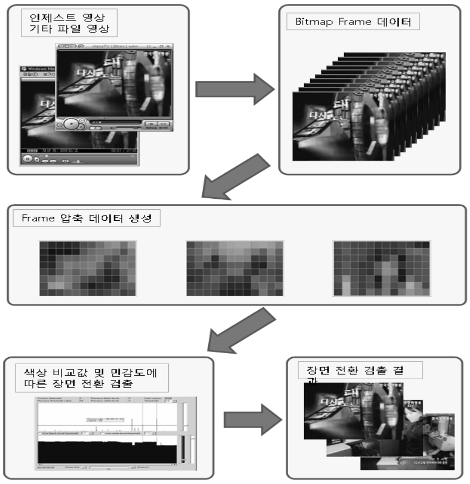


그림 3. 제안한 장면전환검출 방식의 동작과정

후, 이를 연결하여 MPEG2 영상을 디코딩한다. 또한 디코딩됨과 동시에 Grabber 필터에서는 이 디코딩 되고 있는 영상을 비트 맵 어레이(Bit-map Array)버퍼에 계속 복제한다. 이 과정에서는 영상을 플레이할 수 있는 최대한의 속도로 플레이하면서, 동시에 가상 VGA 디바이스 내에서 최소한의 영역(8x8)에서 영상을 플레이함으로써 메모리의 사용을 최소화한다. 분석(Analyzing)모듈에서는 미디어 필터모듈에서 복제된 비트 맵 어레이 버퍼를 분석하는 작업을 수행한다. 분석 작업은 전후 1개의 프레임을 분석하지 않고, 전후 5개의 프레임을 변경 사항을 분석함으로써, 장면전환의 정확도를 높이게 된다. 본 논문에서 제안한 장면전환검출의 처리절차를 그림3에 나타낸 바와 같이 각 영상 파일은 비트맵 프레임 데이터로 나누어져서 프레임 압축 데이터를 생성하게 되고 이것에 장면전환 검출 알고리즘을 적용하여 장면전환의 검출이 이루어지게 된다.

3.3 제안한 장면전환 검출 동작원리

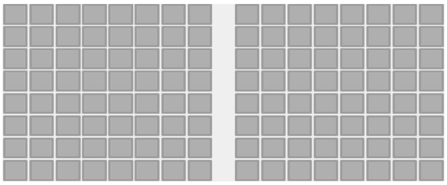
본 논문에서는 장면전환 검출성능 향상을 위하여 첫째, 영상을 어느 정도의 크기로 압축하였을 때, 장면전환검출을 수행할 수 있는 최소한의 크기를 만들 수 있을까에 대한 것과 둘째, 장면전환 검출 시간에서 성능을 어떻게 향상시킬 수 있는가에 중점을 두고 제안시스템을 설계 및 구현하였다.

먼저, 영상을 압축(Shrink)하기 위해서, 가상의 Device Context를 선언하고, 선언한 Device Context의 영역에 영상을 플레이 되도록 하였다. Device Context의 영역은 여러 번의 결과 값에 대한 비교 실험을 통하여, 8pixel * 8pixel의 크기로 설계하였다. 디스플레이 시에 영역을 작게 했을 때에 나타나는 장점은 플레이할 수 있는 배속을 초당 1200 프레임까지 늘릴 수 있었으며, 이를 바탕으로 비트맵을 추출함과 동시에 전체 영역을 플레이하면서 얻을 수 있는 수치에 비해 훨씬 높은 수치를 얻을 수 있었다. 또한 압축(Shrink)된 영상을 기본으로 영상자료를 분석 시에 영상을 저장하는 영역에서 이미 비트맵의 크기가 8*8로 압축되어 있으므로, 비디오 메모리에 저장되어있는 비트맵을 복제하였으므로, 압축에 걸리는 시간을 최소화할 수 있었다.

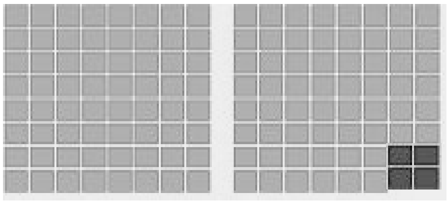
다음으로는, 빠른 장면전환검출을 위해서 8*8 비트 맵 배열 구조를 사용하였으며, 장면전환을 판단하는 임계값은 여러 형태의 동영상 압출파일에 대하여 반복실험을 실시한 결과, 색상 합의 최대값을 255로 할 때, 앞의 샷(shot)의 색상 값의 합과 다음 샷(shot)의 색상 값의 합의 차이가 20이상이 되면 장면전환이 발생하는 것을 확인할 수 있었다. 따라서 본 실험에서 임계값은 20으로 설정하였다.

본 논문에서 동영상 파일에 장면전환을 검출하는 동작원리는 8*8비트맵을 구성하고 있는 각 유닛에 대한 색상 값을 합하여 총 합을 구한다음, 뒤의 프레임의 색상 값의 합과 비교했을 때 임계값(20)을 넘으면 이 지점은 영상이 전환되는 지점으로 판단하였다.

그림 4의 경우에는 앞과 뒤의 두 프레임의 색상 값의 합이 같거나 설정한 임계치보다 작아서 장면전환이 일어나지 않았다고 판단할 수 있다. 하지만 그림4(b)의 경우에는 색상 값의 합에 대한 자이가



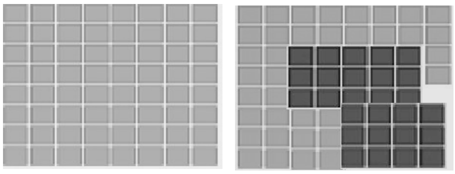
(a) 장면전환이 일어나지 않은 경우



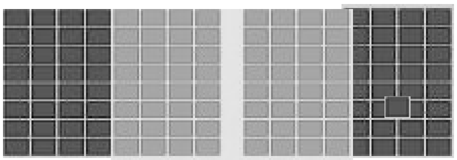
(b) 색상 값 합이 임계치보다 작은 경우

그림 4. 장면전환이 일어나지 않은 경우

www.dbpia.co.kr



(a) 두 장면 간의 색상 값 합의 차이가 임계값 보다 큰 경우



(b) 두 장면 간의 색상 값 합의 차이는 임계값 보다 작지만 픽셀 위치정보가 다르게 나타난 경우

그림 5. 두 프레임 간에 장면전환이 발생한 경우

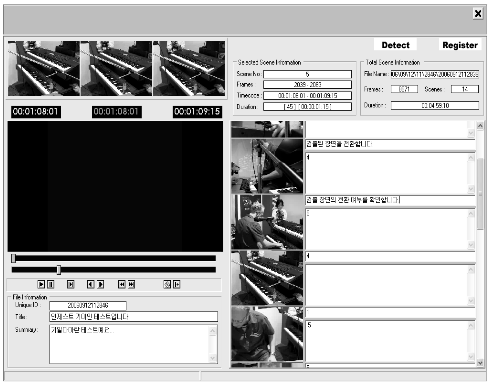
크지 않으므로 하나의 장면 내에 다른 어떤 물체가 등장했다고는 판단할 수 있다.

위의 그림5(a)의 경우 전후 프레임 간의 색상 값이 임계값보다 크게 나타나므로 장면전환이 발생했다고 볼 수 있다. 또한 그림5(b)의 경우처럼 전후 프레임의 색상 값의 합은 임계값(Threshold)을 넘지 않으나, 실제는 장면이 전환된 경우가 있을 수 있다. 이때에는 각 프레임에서 각 값의 위치 정보의 변경을 확인한다. 이와 같은 경우, 전혀 다른 장면이지만 색상 값은 동일하게 나타날 수도 있다. 이러한 경우를 대비하여, 각 색상 값의 합이 임계 값 내에 포함될 경우, 각 픽셀 위치마다의 색상의 차이를 비교하게 된다. 색상 값의 합은 임계 값 내에 포함되나, 픽셀의 위치를 포함했을 경우, 서로 다른 픽셀들로 구성되어 있음을 알 수 있다.

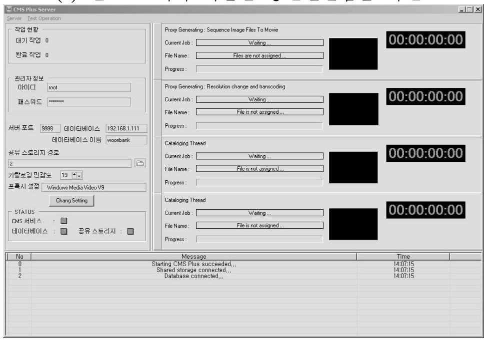
3.4 제안한 장면전환방식의 시스템 구현

3.4.1 개발환경

본 논문에서는 영상의 플레이와 캡처를 위해서 Microsoft 사의 DirectX 9.0 버전에 포함되어 있는 Direct-Show 개발환경을 사용하였다. 또한 성능시험 및 개발에서 공통적으로 사용된 그래픽 카드는 NVIDIA 사의 GeForce 6200 (VRAM : 128MB)을 사용하였다. 프로그램의 효율적인 재사용을 위하여 DLL 형태로 관련 프로그램을 구현하였으며, GUI로 연결되는 테스트 프로그램도 함께 개발하였다. 공통의 라이브러리 (DLL) 제작을 위하여 Microsoft Visual C++ 6.0을 사용하였다. 연결되는 GUI 프로그램은 Borland 사의 C Builder 6.0을 이용하였다.



(a) 본 논문에서 개발한 장면전환검출 화면



(b) 장면전환검출 결과를 위한 서버화면

그림 6. 개발한 장면전환검출 시스템 화면

Ⅳ. 성능시험 및 결과고찰

4.1 시험방법

본 논문에서 제안하고 구현한 장면전환검출방식의 시스템에 대한 성능시험은 마이크로소프트 Windows 2003 R2 버전과 1GB RAM을 이용하는 시스템에서 실시하였다. 시험방법으로는 첫째, CPU와 하드디스크(HDD)를 각각 변경하면서 각 CPU와 하드디스크에서의 얼마나 빠르게 구동되는 동영상 파일에 대하여 프레임을 체크할 수 있는지의 성능 시험하였다. 하드디스크는 대용량 방송동영상 파일의 입출력처리 속도에 영향을 줄 수 있으므로 하드디스크 속도를 달리하면서 시험을 실시하였고, CPU는 대용량 방송동영상의 처리속도에 영향을 줄 수 있으므로 CPU를 달리하면서 제안한 장면전환검출 방식의 시스템을 적용하면서 프레임 체크 속도를 측정하였다. 또한 동영상 파일은 동영상 압축 기술에 따라 지원하는 해상도, 칼라 수 등이 다양하게 나타날 수 있으므로 코덱을 변경하면서 적용시험을 실시하였다. 둘째, 본 논문에서 개발한 고속 장면전환검출 시스템의 처리성능을 분석하기 위하여 기존의 장면

www.dbpia.co.kr

구 분	HDD 5600 rpm	HDD 7200 rpm	HDD 10,000 rpm
Celeron 1GHz	46 fps	51 fps	67 fps
Pentium 3 1GHz	96 fps	105 fps	175 fps
Xeon 3.2Ghz	512 fps	590 fps	629 fps

(a) MPEG1, 29.97fps, 352*240 5:02초 적용 시험

구 분	HDD 5600rpm	HDD 7200rpm	HDD 10,000rpm
Celeron 1GHz	28 fps	31 fps	39 fps
Pentium 3.1GHz	63 fps	85 fps	125 fps
Xeon 3.2Ghz	392 fps	457 fps	525 fps

(b) MPEG2, 29.97fps, 720*480 7:02초 적용 시험

그림 7. 개발한 장면전환검출 시스템의 동영상 프레임 체크 성능 시험결과

전환검출 시스템과 성능 비교시험을 실시하였다.

4.2 시험결과 및 결과고찰

1) CPU 및 HDD 프레인 체크 성능시험 현재 사용되고 있는 펜티엄 4급에서는 성능이 기존의 장면전환검출 소프트웨어에 비해서 높게 나타나고 있다. MPEG2와 같이 해상도가 높은 경우에는, 장면 검출 처리속도가 저하되었다. 특히, Microsoft 사의 DirectShow 필터를 사용하므로, Windows Media Video를 이용한 포맷의 경우, 가장 탁월한 성능을 발휘했으며, 10,000 rpm 스토리지의 경우 최대 24배속을 넘는 속도를 나타내었다.

하드 디스크의 성능과 관련해서는 하드디스크 성능이 약 2배 가까운 향상을 보인다고 할지라도 실제 성능은 10~20% 상승에 그치고 있는 것으로 나타나고 있다. 이는 하드 디스크의 성능 보다는 실제 CPU와 포맷에 따른 성능의 차이가 급격하게 나타나는 것을 확인할 수 있었다. CPU의 경우 1GHz에서 3GHz로 약 3배 증가할 때, 실제 장면전환 검출에 걸리는 속도는 약 5~10배의 성능 차이가 나타나고 있다. CPU 성능이 향상되면서, CPU 캐시 메모리의 증가와 실제 계산에서 나타나는 속도의 차이, MMX 기능의 추가 등이 장면전환 검출 속도에 전반적인 영향을 미치고 있었다.

표 1. 제안한 시스템의 프레임 체크 성능

구 분	기존 시스템 (MBNT)	제안한 개발 시스템	비 고
프레임체크 속도	6 배속	24 배속	일반적인 초당 30프레임 기준
정확도 (오류 수)	2	0	MPEG1파일을 3분 동안 구동 시
지원 영상포맷	MPEG 1,2 특정 포맷	MPEG 1,2,4, AVI,WMV, MOV	-

2) 기존 시스템과의 성능비교 결과

본 연구에서는 제안하고 개발한 장면전환검출 시스템의 성능을 평가하기 위하여 다양한 포맷을 가지는 대용량 동영상을 적용하여 시험을 실시하였다. 그 결과 본 제안 시스템은 다양한 형태의 동영상 포맷에서도 빠른 속도로 장면전환이 이루어짐을 확인할 수 있었다. 또한, 기존에 시판되고 있는 M사 멀티미디어 아카이빙 제품의 장면전환 검출 시스템과 3분 정도의 MPEG 1 파일을 가지고 장면검출 속도를 비교 실험했을 때, 다음의 표1에서 보는 바와 같이 오류가 없이 4배 이상 더 빠르게 프레임 체크를 함을 알 수 있었고, 다양한 영상포맷에 대하여 장면전환검출 가능함을 확인하였다.

Ⅳ. 결 론

지상파 방송을 포함하여 케이블방송, 위성디지털 방송, 인터넷방송 등 다양한 방송서비스가 등장하였다. 이를 위하여 각종 뉴스, 영화, 오락프로그램 등 다양한 형태의 대용량 동영상 방송영상물도 급속히 증가하고 있다. 따라서 이러한 대용량의 방송영상물을 효율적으로 관리하고 저장하고 편집하기 위한 시스템이 요구되고 있다. 대용량의 방송영상물을 효율적으로 관리하기 위해서는 고속으로 장면전환을 검출하여 편집할 수 있는 기술이 필요하다.

본 논문에서 제안하고 구현한 장면 간의 색상값의 합의 차이정도를 이용한 장면전환 시스템을 CPU를 달리하고 하드디스크를 달리하면서 초당 프레임 처리성능을 시험하였다. 제4장의 시험결과에서 보듯이 초당 프레임 체크 성능을 기존의 시스템 보다 4배 (24배속)의 성능을 나타냈으며, 오류도 없이 정확도하게 검출되었다.

본 논문에서 개발한 고속 자동장면전환 시스템은

www.dbpia.co.kr

윈도우 미디어플레이어 에서 구동 되는 모든 미디어 영상을 구동하여 장면 전환을 고속으로 검출할 수 있으며, 다양한 동영상 코덱을 지원함으로 다양한 포맷을 가진 대용량의 방송영상 자료에 대하여 고속 자동 장면전환 검출 시스템으로 활용이 가능하다.

참 고 문 헌

〔1〕 Haitao Jiang, A. Helal, Ahmed K, Elmagarmid and Anupam Joshi, "Scene Change Detection techniques for video data-base systems" Multimedia System, pp. 186-195. 1998.

〔2〕 Rainer Lienhart."Comparison of Automatic Shot Boundary Detection Algorithms" Microcomputer Research Labs, Intel Corp.

〔3〕 B. L. Yeo, and B. Liu, "Rapid scene analysis on compressed video," IEEE Trans. on Circuits and Systems for Video Technology, Vol.5, No.6, pp.533-544, Dec.1995.

〔4〕 Nilesh V. Patel, Ishwar K. Sethi, "Video Shot Detection and Characterization for Video Databases," Pattern Recognition, Vol.30, No.4, pp.583-592, 1997.

〔5〕 Dan Lelescu, Dan Schonfeld, "Statistical Sequential Analysis for real time Video Scene Change Detection on compressed Multimedia Bitstream," IEEE Transaction on multimedia," Vol5, No1, pp.106-117, 2003.

〔6〕 김강욱 외3, "MPEG으로 압축된 비디오에서 최소 복호화에 의한 빠른 장면전환검출 알고리듬", 정보처리학회논문지 B 제9-B권 제3호, pp.343-346, 2002.

〔7〕 강응관, "MPEG 비디오 시퀀스에서 비디오 요약을 위한 장면 전환 검출 및 대표 프레임 추출 알고리즘", 멀티미디어학회 논문지 제6권 제5호, pp.797-802, 2003.

〔8〕 정성태,"Visual C++을 이용한 실용영상처리" 생능출판사, 2005.

〔9〕 장석우 외1,"카메라의 동작을 보정한 장면전환 검출",정보처리학회논문지 제12-B권 제4호, 2005.

〔10〕 김중헌 외1,"MPEG에서 B 프레임의 특징을 이용한 급진적인 장면전환 검출에 관한 연구", 정보처리학회논문지 제12-B권 제5호, 2005.

〔11〕 신성윤,"장면전환검출과 사용자 프로파일을 이용한 비디오 학습평가 시스템", 정보처리학회 논문지 제11-D권 제1호, 2004.

〔12〕 김성국 외4,"MPEG-2 압축영역의 TV 스포츠 뉴스 색인을 위한 효율적인 장면전환 및 기사 검출", 정보처리학회논문지 제6권 제6호, 1999.

송 한 춘 (Song Han Chun) 정회원

1990년 성균관대학교 공대전자 공학과 졸업<공학사>

1994년 연세대학교 공학대학원 전자공학전공 졸업 <공학석사>

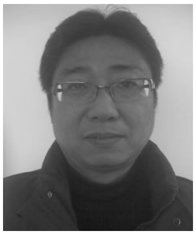
1998년 8월 성균관대학교 대학원 통신공학전공 졸업 <공학박사>

1990년 1월~1997 2월 (주) 데이콤 근무

1998년 3월~현재 서일대학 정보통신과 부교수

<관심분야> 정보통신시스템, 인터넷 네트워크

김 진 용 (Park Myeong Hwan) 정회원



1983 인하대학교 이과대 물리학과 졸업 <이학사>

1988년~1998년 현대정보기술(주) 미디어사업 본부 책임연구원

1998년~2002년 서울일렉트론 방송 SI팀 팀장

2004년~2008년 케레스인포 부설기술연구소 연구소장

<관심분야> 디지털방송 편집, 멀티미디어통신

www.dbpia.co.kr