

파편화된 다수 동영상 파일의 동기화 재생 제어

김현구, 권정웅, 남황우, 전중남
 충북대학교 소프트웨어학과
 e-mail:joongnam@cbnu.ac.kr

Synchronized Play Control for Multiple Fragmented Video Files

Hyeon Gu Kim, Jung Ung Kwon, Hwang Woo Nam, Joong Nam Jun
 Dept of Software, Chung-buk University

요 약

이 논문에서는 손상에서 복원 되었거나 기타 이유로 DVR 시스템의 관리에서 벗어난 파편화된 다채널 영상 파일들에 대한 다채널 영상 재생 기법을 제안한다. 이를 위해 영상에 매핑 된 메타데이터에 기반 하여 촬영 시간에 대응되는 올바른 시간적 위상에서 재생될 영상 파일을 실시간으로 불러오고, 재생이나 탐색 중에 시간을 동기화시키는 과정을 수행 한다. 그리고 오픈소스 영상 재생 API인 VLC Player를 활용하여 시스템을 설계 하였고, 구현한 시스템은 상용 DVR 시스템을 사용하지 않고도 다채널 영상 파일을 직관적으로 관독할 수 있는 환경을 제공한다.

1. 서론

실시간 지역 감시에서 녹화된 영상의 사후 관독은 아주 중요하다. CCTV 영상은 대체로 영상의 길이가 길고, 여러 채널의 영상을 비교해야하기 채널별로 표시성을 보장하는 다채널 영상 재생 시스템을 활용해 관독하는 것이 일반적이다. 하지만 이러한 영상 관독 시스템은 보통 비교적 고가의 상용 DVR 솔루션의 일부로써 제공되며 영상 파일의 메타데이터 구조가 제조사별로 호환되지 않는다. 때문에 수시로 여러 종류의 DVR에서 취득된 영상을 관독해야하는 상황에서는 기존의 상용 DVR을 사용하는 것이 불리하다.

이 논문에서는 오픈소스 영상 재생 API인 VLC Player를 활용하여 비교적 단순한 구조로 DVR 시스템의 관리를 받지 않는 다채널 영상파일에 대해 활용 가능한 다채널 영상 관독 시스템을 제안한다. 이를 위해 영상 파일이 재생되는 시간을 동기화시키는 구체적인 방법을 설명하고 실제 시스템을 설계 구현하는데 초점을 둔다.

2. 관련 연구

시간 동기화를 이루는 다채널 영상 재생 방법은 기존의 DVR 시스템에서 일반적으로 사용되고 있다. 하지만 다채널 영상재생 기술에 대한 연구는 실시간 원격 감시[1], 영상의 캡처 및 관리하는 과정과 통합되어 이뤄져 왔다. 때문에 영상의 스트리밍 재

생, 캡처 성능 개선에 대한 연구는 있었지만[2] 단순히 영상의 관독만을 담당하는 독립된 시스템은 없는 상황이다.

비영리단체 VideoLAN는 오픈소스 방식으로 영상 재생 API인 VLC Player를 개발 했는데 플랫폼의 제약이 없고 다양한 포맷의 영상 파일을 재생 할 수 있다. 때문에 이를 활용하여 영상 재생기능을 포함하는 어플리케이션을 비교적 단순한 구조로 개발 할 수 있다. 또한 VLC Player의 컴포넌트는 영상의 재생을 다중쓰레드로 처리하기 때문에 동시에 여러 개를 동작시키더라도 성능이 보장되며, 시간을 정확히 동기화 시킬 수 있는 장점이 있다.[3]

3. 동기화 재생 제어 방법

3.1. 메타데이터 구조와 실시간 영상 선별

이 시스템에서 다루는 영상 파일은 영상의 '녹화 시작 시각'과 '녹화 종료 시각', 영상이 취득된 카메라의 '채널 번호' 정보를 메타데이터로 가지고 있어야 한다. 이 때 각 영상 파일의 주요 메타데이터인 '녹화 시작 시간' 과 '녹화 종료시간', '채널 번호'를 해당 영상파일의 이름에 실는 것으로 한다.

<표 1> 주요 메타 데이터를 싣는 파일 이름의 형식

파일 이름 형식
CAM<채널번호>[시작 년, 월, 일, 시, 분 초]~ [종료 년, 월, 일, 시, 분, 초]
예시
CAM01_[2014년 8월 29일 8시 0분 1초] ~ [2014년 8월 29일 8시 0분 12초]

영상파일은 재생 및 탐색이 일어나는 동안 실시간으로 선별되어 VLC Player에 로드된다. 이 때 선별되는 영상파일은 재생, 탐색이 이루어지는 임의의 '재생 시각'을 '녹화 시작 시각'과 '녹화 종료시각' 사이에 포함하는 영상으로 결정한다. 또한 영상은 채널별로 할당된 화면 영역에서 동시에 복수로 재생될 수 있다. (그림 1) 과 같이 배치 될 수 있는 채널별로 각기 다른 영상파일 A, B, C, D, E, F 가 존재 한다고 가정하면, 시각 13과 시각 6이 재생시각일 때 채널별로 각각의 영역에서 재생되는 영상은 다음 (그림 2) 와 같이 선택된다.

시각 \ 채널	4	5	6	7	8	9	10	11	12	13	14
1			A				B				
2			C				D				

(그림 1) 임의의 영상파일 A, B, C, D 을 촬영 시각을 바탕으로 타임라인에 배치한 모습

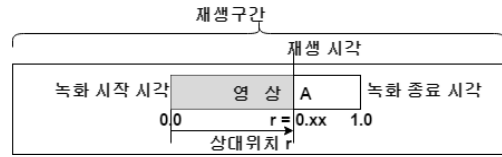
a)	채널 1	채널 2	b)	채널 1	채널 2
	-	D		A	C

(그림 2) 시각 13에서 영역별로 로드되는 영상 a) 과 시각 6에서 영역별로 로드되는 영상 b)

3.2. 상대 위치와 탐색 중 시간 동기화

상대 시각은 '재생 시각'에 대응되는 로드된 영상에서의 위치이다. 여기서 상대위치 r 은, '재생 시각'이 t 일 때 '촬영 시작 시각'이 s , '녹화 종료 시각'이 e 인 영상에 대해 다음과 같이 정의한다.

$$r = (t - s) / (e - s) \quad (1)$$



(그림 3) 상대위치의 개념

사용자가 트랙 바를 스크롤 하면 탐색이 시작되고, 사용자에게 의해 변경된 트랙 바의 핸들의 위치에 의해 '재생 시각'이 결정된다. 채널별로 로드된 각각의 영상에 대해 '재생 시각'에 대응되는 상대위치를 VLC Player의 position 속성으로 세트 해주면 '재생 시각'에 대응되는 위치에서 부터 재생이 준비되고 해당 위치의 프레임이 디스플레이 된다.

3.3. 재생 중 시간 동기화

재생이 일어나는 동안 '재생 시각'은 타이머의 시간 경과에 영향을 받는다. '재생 시각' t 는 마지막으로 일시정지가 일어났던 '재생 시각' p 에서 재생이 재개된 이후 시스템 타이머 상으로 경과한 시간 s 의 합으로 결정되고, 트랙 바 핸들의 위치를 변경하는데 활용한다.

$$t = p + s \quad (2)$$

영상 파일의 '녹화 시작 시각'과 '녹화 종료시각'의 차로 결정되는 영상의 '녹화 시간'은 영상파일의 실제 런 타임과 다를 수 있다. 때문에 영상 재생이 '촬영 시간'에 걸처서 이뤄질 수 있도록 재생 속도의 비율을 조정해야한다. 이 때 보정된 재생 속도의 비율 c 는 '녹화 시작 시각' l , 영상의 런 타임 r 에 대해 다음과 같이 결정된다.

$$c = \frac{r}{l} = \frac{r}{e - s} \quad (3)$$

VLC Player의 rate 속성을 위에서 계산한 c 로 세트하면 해당 비율만큼 FPS가 변경되어 재생 속도가 조정된다. VLC Player의 재생역시 타이머의 시간경과에 동기화되기 때문에 재생 속도를 조절하는 것으로도 '재생 시각'에 맞춰 시간이 동기화가 이루어지는 것으로 간주할 수 있다.

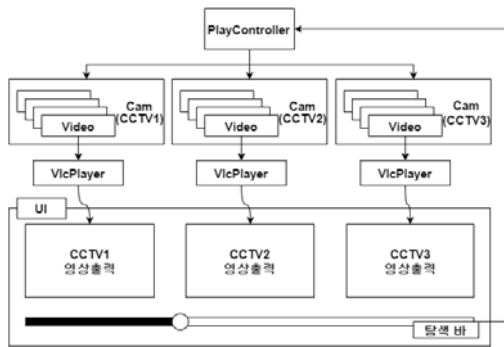
4. 시스템 설계 및 구현

재생될 각각의 영상들의 파일 경로와 메타데이터 들은 불러 오기를 통해 Video 객체에 저장된다. 활

영된 채널번호가 동일한 Video 객체들은 동일한 Cam 객체에 저장된다. Cam 클래스는 채널 별로 Video 객체를 분류하고, 동시에 특정 시각에서 재생 되어야 하는 내부의 Video 객체를 찾아내는 역할을 수행한다.

트랙바의 스크롤, 시스템 타이머의 시간 경과, 재생 버튼 클릭과 같은 이벤트를 수신하고 '재생 시각'에 VlcPlayer가 재생해야 할 영상의 경로를 Cam 클래스에서 전달받아 VlcPlayer에 할당하는 역할은 PlayController 클래스가 담당한다. PlayController는 이를 위해 내부에 모든 Cam 객체와 각각의 Cam 객체에 대응되는 VlcPlayer 객체를 포함한다.

PlayController 객체는 추가적으로 Time 객체를 포함한다. 이 Time 객체는 '재생 시각'과 타이머 객체를 저장하며 트랙바가 스크롤 되거나 시스템 타이머 시간 변화에 따라 값이 변하게 되며 시간 동기화를 위해 사용된다.



(그림 4) 시스템의 모듈 및 UI 구조

5. 실행 결과

구현된 시스템은 실제로 다채널 영상을 시간의 흐름에 따라 직관적으로 관독할 수 있는 환경을 제공한다. VLC Player에서 지원하는 MPEG-2, Divx, H.264, MKV, WMV 같은 형식의 영상 파일을 불러와 재생 할 수 있다. 다중 스레드를 통해 영상 재생이 일어나기 때문에 UI의 반응성을 보장 하면서도 시간의 동기화가 정확히 이루어진다.

하지만 <표 1>과 <표 2>와 같은 테스트 환경에서 약 20채널 정도의 영상을 동시에 재생할 경우 영상들의 재생 과 탐색 성능이 떨어지는 현상이 발생한다. 이에 따라 또한 UI가 사용자 입력에 따른 이벤트역시 제 시간에 처리하지 못하게 되면서 UI의 반응성 역시 현저히 나빠지는 문제가 있다.

<표 2> 테스트가 이루어진 시스템 환경

테스트 시스템 성능 및 환경	
CPU	i5-3570
운영체제	Windows 8.1 64bit
RAM	8 Gb
그래픽카드	Nvidia GTX-660

<표 3> 테스트한 영상들의 정보

테스트 영상 정보	
코덱	H264
영상 크기	704 x 240
비트 레이트 범위	250 ~ 500kbps
오디오	없음

6. 결론

제안한 시스템을 활용하면 기존의 DVR 시스템을 사용하지 않고도 파편화된 다 채널 영상 파일을 손쉽게 관독 할 수 있다. 파일 구조에 대한 제약이 없고 메타 데이터의 형식이 단순하기 때문에 임의의 영상 파일을 이 시스템에 맞춰서 다채널 방식으로 관독할 수 있도록 수정하는 것이 용이하다.

또한 PC의 성능이 충분하다면 동시 재생 가능한 채널의 개수에 제한 없이 다수의 영상을 동시에 재생 하는 것이 가능하다. 하지만 재생되는 영상의 개수가 많아질 때 재생 및 탐색의 성능이 저하되는 현상이 확인되었다. 이러한 문제는 영상을 디코딩하는 과정에서 발생하는 부하가 여러 영상에 걸쳐 누적된 결과로써 고성능의 CPU를 사용해 해결 할 수 있을 것으로 판단된다.

감사의 글

이 연구는 IIPT의 2015년도 서울 어코드사업의 지원을 받아 수행되었습니다.

참고 문헌

[1] 김규석, 지미경, 김천석, "Full HD H.264 카메라를 이용한 감시 시스템의 다채널 Client Viewer 구현", 한국방송공학회, 한국방송공학회 하계학술 발표대회 논문집, 7. 2010, pp. 357-360.
 [2] 정선태, "DirectX 기반 다채널 영상 감시 시스템 구현 및 성능 분석", 한국콘텐츠학회논문지 제5권 제1호, 2. 2005, pp. 217-227.
 [3] VideoLAN organization, "Developer zone", 2015, Available <http://www.videolan.org/developers/> (accessed on Sept. 9. 2015)