

# 윈도우즈 기반 다채널 영상 감시 시스템의 실시간 재생

양 정 훈, 정 선 태  
승실대학교 정보통신전자공학부  
전화 : 02-817-5987/핸드폰 : 016-282-4966

## Real-time Playback of a Windows based Multichannel Visual Monitoring System

### Abstract

In this paper, we present a DirectShow-based retrieval and playback subsystem of DVR(Digital Video Recorder), which supports real-time playback of stored video data and synchronized playback among several video channel data. The effectiveness of our proposed design is verified through experiments with a DVR system implementing the proposed design.

### I. 서론

DVR(Digital Video Recorder)은 CCTV 카메라들로부터 획득한 아날로그 영상 신호를 디지털 영상 데이터로 변환하여 모니터 화면에 실시간 디스플레이하고, 추후 검색을 위해 스토리지 장치에 영상 데이터를 저장하며 저장된 데이터를 쉽게 검색 할 수 있는 새로운 디지털 영상 감시 시스템이다 [1].

DVR은 보통 다채널(최대 16채널) CCTV 카메라 영상을 디스플레이하고, 16 채널 영상 데이터를 저장한다. 그런데, 현재 시판되고 있는 DVR 시스템에서 다채널 검색시에 실시간 검색의 지원은 미흡하다. 즉, 검색 및 재생시에, 녹화된 각채널의 영상이 녹화되었던 당시의 시간 및 속도에 맞게 재생되지 않고 CPU 및

운영체제 성능에 따라 재생속도가 결정된다. 예를 들어, 10초에 1장씩 저장되었던 영상 데이터가 1초에 수십 장씩 재생되는 경우가 발생한다. 이 경우, 실제 녹화시의 시간에 맞지 않게 각 프레임이 재생되어 버린다. 또한, 다중 채널 검색 및 재생 시에 각 채널의 영상이 서로 시간 동기가 맞지 않게 재생되므로, 각 채널의 영상을 시간에 따라 비교할 필요가 있는 경우에는 문제가 발생한다.

따라서, 검색 및 재생시에 시간 동기가 맞는 실시간 검색을 위해서는 영상 데이터 저장시에 저장 실시간 관련 정보가 삽입되어야 하고, 이 실시간 저장 시간 관련 정보를 이용하여 실시간 검색 및 재생이 되어야 한다. 현재 시판되고 DVR에서는 영상의 실시간 재생에 필요한 저장 실시간 관련 정보를 잘 지원하고 있지 못해, 실시간 검색 및 재생이 어렵다.

DVR은 PC 기반 시스템과 임베디드 시스템으로 구별되는 데, 고기능 및 고성능의 DVR은 PC 기반 시스템이며, 사용되는 운영체제로 대부분 윈도우즈를 채택하고 있다. 윈도우즈에서는 멀티미디어 스트리밍 처리를 위해 DirectShow를 지원하고 있다. DirectShow는 멀티미디어의 시간 동기를 위해 타임스탬프가 처리되는 멀티미디어 스트림에 기록되도록 지원하고 있다. 따라서, DirectShow 기반으로 DVR의 저장, 검색 및 재생 시스템을 구성하면, 실시간 재생 구현이 용이하게 된다. 현재, 출시되는 윈도우즈 기반 DVR 시스템에서는 아직까지는 DirectShow를 채택하고 있지 못하며, 따라서 동영상 스트림의 시간 정보 처리에 충분한 기능을 제공하는 데 미흡하다.

본 논문에서는 영상 데이터 저장시에 저장 실시간

관련 정보를 갖도록 설계·구현한 DirectShow 기반 DVR에서 실시간 검색 및 재생, 그리고 다채널간 실시간 동기 재생이 이루어지도록 한 연구 결과를 설명한다.

## II. 기존 영상 감시 시스템에서의 저장, 검색 및 재생 구조

기존의 대부분의 DVR 제품은 전용 코덱을 채택하고 있다. 영상 감시 시스템에서는 특정 장면의 감시 및 기록 등이 필요하므로 프레임 단위의 처리가 요구되며, 이에 따라 프레임 단위 처리에 편리한 MJPEG(Motion JPEG)을 전용 코덱으로 많이 사용한다. MJPEG는 JPEG 프레임 또는 그 변형 형태의 프레임의 연속으로 구성된다. 물론 경우에 따라서는 H.263 또는 MPEG 등의 표준 동영상 코덱을 채용하기도 한다. MJPEG 코덱을 사용하는 경우, 보통의 MJPEG 파일은 프레임 영상 데이터에 대해, 프레임 번호, 영상 데이터의 크기 등을 기록한 헤더를 갖으며, 전체 MJPEG 파일의 헤더에는 기록된 총 프레임수의 정보 등을 보유하도록 설계된다. 검색 키워드인 시간, 채널, 이벤트 등을 지원하기 위해, 시간대별로 단일 인덱스 파일을 구성하고, 해당 인덱스 파일은 레코드들의 모임으로 구성된다. 레코드에는 해당 프레임이 저장된 시간(년,월,일,시,분,초), 카메라 채널 번호, 센서 번호, 현재 저장하고 있는 파일 이름, 현재 저장되는 프레임 번호 등의 정보를 보유하도록 설계되고 있다. 이때, DVR에 요구되는 검색 중 가장 중요시되는 시간별 검색을 위해서 레코드가 보통 시간순서로 저장된다. 이러한 저장 구조에서는 예를 들어 시간별 검색 및 재생시에, 인덱스 파일에서 해당 채널, 해당 시간에 해당되는 레코드를 찾고 이 레코드의 정보로부터 해당 프레임들을 찾아 처음부터 차례로 한 프레임씩 해당 채널에 따라 재생하게 된다. 따라서, 각 채널별로 재생되는 영상 프레임의 시간 동기는 어느 정도 이루어진다고 볼수있다. 그러나, 실시간 재생은 아니고 CPU와 운영체제의 성능에 맞추어 헛헛 재생하는 것이 되기 때문에 실시간 30분에 걸쳐 저장된 영상이 10분 이내로 재생될 수도 있다는 단점이 있다. 더구나, 이러한 인덱스 파일 구조에서는 모든 채널의 영상 프레임에 대한 레코드가 시간 순서대로 배치되어 있기 때문에, 영상 데이터 저장시에 각 카메라 채널별로 쓰레드를 두어 각 채널별 저장이 이루어 지도록 하기 힘들다. 이는 각 쓰레드별로 해당 영상 프레임을 저장하고 해당 프레임에 대한 레코드 정보를 단일한 인덱스 파일에 저장할 때, 시간 순서대로 인덱스 파일이 구성되기

어렵기 때문이다. 이러한 단일 인덱스 파일 구조에서는 저장시 뿐만 아니라, 검색·재생 시에도 멀티쓰레딩 작업이 힘들게 되어, 저장 및 검색·재생 성능을 제고하는 데 제약을 갖게 된다.

한편, 기존 구조에서도 채널별로 인덱스 파일을 만드는 것을 고려해 볼 수도 있다. 이 경우, 채널별 저장을 위한 멀티쓰레딩 저장 작업은 가능해진다. 그러나, 검색후 재생이 채널간에 시간 동기에 맞추어 재생되어야 하는 데, 멀티쓰레딩을 이용하는 경우는 이 작업이 단일 인덱스 파일을 사용할 때에 비해 복잡해지며, 싱글 쓰레딩을 이용하는 경우에는 단일 인덱스 파일을 사용하는 것 보다 검색 및 재생시간이 더 걸린다.

## III. 실시간 검색 지원을 위해, 제안된 다채널 영상 감시 시스템의 저장, 검색 및 재생 구조

### 3.1 제안된 DVR 시스템 개요

새롭게 설계된 DVR 시스템은 DirectX 기반으로 설계·구현되었다. 실시간 디스플레이 부분은 DirectX의 DirectDraw를 이용하였고, 저장, 재생 부분은 DirectX의 DirectShow를 이용하였다[2,3]. 영상 캡처를 위해서 WDM 기반 영상 캡처 장치 드라이버를 설계하고 구현하였으며, 영상 캡처와 연동되는 DirectShow 소스 필터를 설계·구현하였다. 영상 저장의 경우, 이 소스 필터, 움직임 탐지 모드를 지원하도록 설계·구현된 트랜스폼(Transform) 필터, 압축 코덱을 지원하는 또 다른 트랜스폼 필터(윈도우즈 제공) 및 파일저장 필터들이 필터 그래프를 형성하여 영상 저장 서버 시스템을 구성하도록 설계·구현되어 있다. 영상데이터 검색 및 재생을 위해 DirectShow 구조, RIFF AVI 파일 형식 [4] 및 다중 쓰레드 처리 구조를 고려하여 인덱스 파일 및 영상 파일 구조가 설계·구현되었다. 제안된 새 DVR의 구조에 대한 보다 자세한 설명에 대해서는 본 논문의 저자들의 이전 논문[3]을 참조하라.

### 3.2 제안된 DVR의 영상 파일 저장 구조, 인덱스 파일 구조 및 저장 구조

#### 1) 영상 파일 저장 구조

영상 데이터의 저장은 채널별로 이루어지며, 연속 저장의 경우에는 매시간 새로운 파일에 저장하도록 하며, 녹화 기능이 정지되었다가 새로 녹화할 경우에도 새로운 파일에 저장하도록 하였다. 따라서, 이러한 정

책에 의해 파일 저장 디렉토리 구조와 파일 이름은 다음과 같이 설계가 되었다.

../Data/년도월일/시간/채널번호분초.avi

2) 인덱스 파일 구조

인덱스 파일 작성은 채널과 이벤트를 고려하여 설계하였다. 만일 채널별로 한 개의 인덱스 파일이 구성되는 경우 이벤트를 키워드로 하는 검색은 각 채널의 인덱스 파일을 모두 검색하여야 하므로 비효율적이다. 따라서, 시간 키워드에 대한 검색에 대해서 하루에 채널별로 한개의 시간 인덱스 파일을 작성하고, 이벤트 검색을 위해서는 하루에 한 개의, 사건 인덱스 파일을 작성하도록 설계하였다.

DirectShow에서는 RIFF AVI 파일의 시간별 및 프레임 번호별 검색을 지원하므로, 종래 DVR 의 인덱스 파일처럼 프레임별로 해당 프레임이 저장된 시간 정보 등을 기록할 필요가 없다. 따라서, 시간 인덱스 파일은 RIFF AVI 파일 마다의 레코드(Time Index Block) 로 구성된다.

기존 영상 감시 시스템의 경우 영상 데이터의 구조를 비표준형 구조를 사용함으로써 프레임의 Time stamp에 영상의 녹화 시간을 기록할 수 있다. 그러나 RIFF AVI구조에서는 시간 정보가 녹화 시작 시점이 0부터 시작하는 Stream Time으로 Time stamp가 기록이 되어 있다. 따라서 검색을 위해서는 영상 데이터가 기록된 년월일과 같은 실제 시간에 대한 정보를 별도로 기록을 하여야 한다. 따라서 별도로 인덱스 파일에 이러한 실제 녹화 시간에 대한 시간 정보의 기록이 필요하다. 제안된 시스템에서 저장된 영상 데이터에 대한 실제 녹화 시간은 시간에 의해 분류된 디렉토리 정보를 통해서도 획득 될 수 있다. 그러나 이러한 시간 정보 외에 채널에 대한 정보나 Motion Detection 사용 유무와 같은 추가적인 정보의 기록의 필요성에 의해 별도의 인덱스 파일에 정보를 저장하는 것이 필요하다.

시간 인덱스 파일의 각 레코드(Time Index Block)는 해당 RIFF AVI 파일에 대한 검색 정보, 즉 기록 시작 시간, 기록 정지 시간, 파일 이름, 채널 정보, 기록 모드(연속하여 기록, 또는 움직임이 탐지되는 경우에만 기록) 등의 정보를 보유하도록 설계하였다.

3) 인덱스 파일 저장 구조

제안된 시간 인덱스파일의 저장구조는 다음과 같다.

시간인덱스파일:../Data/년도월일/Index/채널번호.idx

인덱스 파일은 각 채널별로 저장이 되며 하나의 동영

상 파일에 대한 정보를 하나의 Index Block에 저장한다. 검색 시에는 동영상 저장 디렉토리 구조에 대한 전체적인 검색 없이 인덱스 파일의 정보만을 검색함으로써 빠르게 원하는 동영상 데이터의 존재 유무와 동영상 파일로의 접근을 가능하게 한다. 하나의 Index Block은 하나의 동영상 파일과 대응하므로, 기존의 인덱스 구조에 비해 인덱스 파일 자체의 크기가 훨씬 작아져 메모리의 효율적인 사용이 가능하고 인덱스 파일 저장을 위한 파일 I/O 작업의 양도 줄어들게 되어 성능을 개선시킨다.

3.3 실시간 재생 및 채널간 동기화 문제 해결

1) 실시간 재생

영상 획득시에 영상 획득 장치 드라이버에 연동되는 소스 필터에서 획득된 영상 한 프레임을 미디어 샘플로 만들 때에, 획득 실시간과 그 영상 프레임 재생시간 구간을 스트림 시간으로 기록하도록 설계·구현하였다. 이후 소스 필터와 연결된 트랜스폼 필터, 파일 저장 필터 등은 미디어 샘플에 기록된 스트림 시간을 참조하며, 파일 저장 필터는 영상 스트림을 AVI 파일로 저장할 때, 이 스트림 시간을 고려하여 저장한다. 따라서, 재생시에, DirectShow 재생 필터는 각 AVI 에 저장된 저장 실시간 정보에 관련된 정보를 이용하여 재생하므로, 저장시의 실시간에 맞게 실시간 재생을 수행한다.

2) 채널간 시간 동기화 문제 해결

기존 구조에서는 인덱스 파일에 기록된 순서에 따라 한 프레임씩 재생을 하지만 제안된 구조에서는 각 채널별 재생이 다중 쓰레드에 의해 동시에 동작을 하므로 채널간의 시간 동기화 문제의 해결이 필요하다.

다음 그림 1과 같이 동영상 데이터가 저장이 되어 있을 경우를 가정하자.

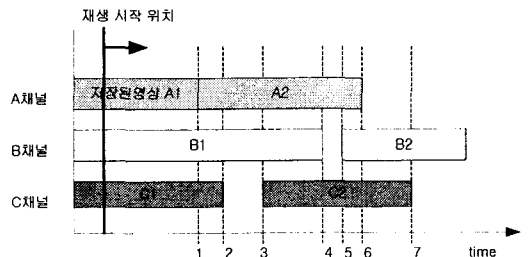


그림 1. 채널별 동영상데이터의 시간에 따른 저장 예

- time 1 : A1 파일의 종료, A2 파일의 시작
- time 2 : C1 파일의 종료
- time 3 : C2 파일의 시작

- time 4 : B1 파일의 종료
- time 5 : B2 파일의 시작
- time 6 : A2 파일의 종료
- time 7 : C2 파일의 종료

위와 같이 저장되어 있을 경우, 재생시에 시점 1에서 A1파일의 종료와 동시에 A2 파일에 대한 FilterGraph를 형성하여 재생을 시작하여야만 파일이 분할되어 있더라도 사용자에게는 하나의 스트림으로 보이게 된다. 시점2에서는 C1 파일이 종료되므로 C2에 대한 FilterGraph를 형성하지만 재생 시점이 아니므로 시점 3까지 대기한 뒤 시점 3부터 재생을 시작한다. 이러한 방식으로 시간에 대해 각 채널별로 동기화를 유지하는 것이 사용자에게 시간에 대한 채널별 녹화영상의 상황을 올바르게 전달 할 수 있다. 따라서 재생 Manager는 이러한 동기화를 유지하기 위한 기능을 갖추어야 한다.

따라서 이러한 기능을 담당하는 재생 Manager는 각 FilterGraph의 상태를 항상 알고 있어야 한다. 시점 1과 같은 경우 A채널 FilterGraph의 재생이 종료되었다는 상태를 알고 있어야 그에 따른 동작 등을 수행 할 수 있기 때문이다. DirectShow FilterGraph는 어플리케이션 측에 상태 변화에 따른 이벤트 메시지를 전달하고 재생 Manager에서는 이러한 이벤트 발생시 이벤트를 분석하여 이벤트 내용에 따른 오퍼레이션을 취하게 된다. 즉, 시점1에서는 A채널의 FilterGraph로부터 이벤트 통지 메시지가 발생하게 되고 재생 Manager는 FilterGraph의 Event Queue로부터 이벤트를 분석하여 FilterGraph의 상태가 재생 종료 되었다는 것을 알 수 있다. 그에 따라 인덱스 정보를 분석하여 다음 동영상 데이터에 대한 재생 위한 FilterGraph를 새로 생성하고 디스플레이 할 시점에 도달하면 재생을 하게 된다.

또한 재생 Manager는 전체적인 재생의 흐름을 기준 시간을 통해 제어하게 된다. 기준 시간은 영상 데이터의 재생 스트림에 대한 시간이라고 할 수 있다. 만약 사용자가 2배속 재생을 하게 된다면 기준 시간은 실제 시간보다 2배로 빠르게 흘러가게 되며 저속 재생을 하게 된다면 느리게 흘러간다. 이러한 기준 시간은 시점 3이나 시점 5와 같이 특정 FilterGraph의 재생 시작 시점을 파악하기 위해 필요하다. 재생 Manager에서는 재생 중인 FilterGraph중에 현재 재생 시점부터 해당 FilterGraph의 재생 종료 시점까지의 기간이 가장 긴 채널을 유동적으로 선택하여 기준 시간 정보를 얻어오고 이를 이용하여 전체적인 재생의 흐름을 제어 한다.

이렇게 설계된 검색 및 재생 서버 시스템의 구조는 다음 그림2 와 같다.

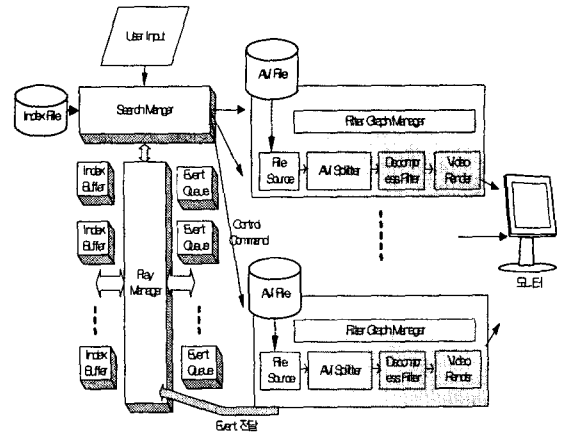


그림 2. 설계된 검색 및 재생 서버 시스템의 구조

#### IV. 결론

본 논문에서는 현재 출시되고 있는 윈도우즈 기반 DVR 시스템에서 미흡한 실시간 재생 및 각 채널간 시간 동기 재생을 지원하기 위해, DirectShow에 기반한 저장, 검색 및 재생 구조를 제안하였다. 제안된 구조에 기반하여 실제 구현한 결과 실시간 재생 및 채널간 시간 동기 재생이 완벽하게 지원됨을 확인하였다. 현재, 이러한 결과를 바탕으로 상업화 버전을 구현하고 있다.

#### 참고문헌

- [1] DVR 시스템 개요, 웹문서, <http://www.digitalplaza.co.kr>
- [2] "DirectShow SDK Documentation", [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9\\_c/directx/htm/directshow.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/directx9_c/directx/htm/directshow.asp)
- [3] 정연권, 하상석, 정선태, "DirectShow 기반 다채널 영상 감시 시스템의 검색 및 재생", 대한전자공학회 추계 학술대회 논문집, 2002.
- [4] "MSDN Library" [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/w98ddk/hh/w98ddk/mmedia\\_4ylv.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/w98ddk/hh/w98ddk/mmedia_4ylv.asp)