
DirectX 기반 다채널 영상 감시 시스템 구현 및 성능 분석

Implementation and Performance Analysis of a Multichannel Visual Monitoring System based on DirectX

정선태

승실대학교 정보통신전자공학부

Sun-Tae Chung(cst@ssu.ac.kr)

요약

본 논문은 성능이 우수한 DirectX 기반 다채널 영상 감시 시스템을 설계·구현하고, 성능 분석한 결과를 제시한다. 본 논문의 시스템은 크게 디스플레이 서브시스템, 저장 서브시스템, 검색 및 재생 서브시스템으로 구성된다. 디스플레이 서브시스템은 빠른 디스플레이, 채널 간 디스플레이 동기 및 tearing effect 개선을 위해, DirectDraw의 H/W 가속 기능과 overlay, flip 기능을 효율적으로 이용하여 설계되었다. 저장 서브시스템은 저장 속도 개선을 위해, DirectShow 필터 기반 다중 쓰레드 구조로 설계되어 캡처 장치에서 획득되는 다채널의 영상 스트림이 채널별로 압축 저장되고 채널 간 데이터 병목 현상이 최소화되도록 하였다. 검색 및 재생 서브시스템에는 효율적인 인덱스 파일 구조와 영상 데이터 저장 구조, 재생 시에 채널별 동시 재생이 가능하도록 한 효율적인 재생 구조를 설계·구현하여 보다 빠른 재생 및 검색이 가능하도록 하였다. 실험결과, 제안된 시스템이 기존 시스템보다 저장 속도는 최고 2배 까지, 검색 및 재생 속도는 최고 3.5배 정도까지 개선되었음이 확인되었다.

■ 중심어 : | 다채널 영상감시 | 비디오 스트림 | DirectX, DirectShow | DVR |

Abstract

In this paper, we present the result of an efficient implementation of DirectX-based multichannel visual monitoring system and the performance analysis of it. Our proposed system mainly consists of three subsystems: display, storage, and retrieval/playback. The display subsystem is designed to utilize efficiently H/W acceleration, overlay and flip of DirectX for faster real-time display, display synchronization among channels, and improvement of tearing artifact. For the performance improvement of storage speed, the storage subsystem is designed and implemented in DirectShow Filter-based multithreading architecture so that it can store multichannel video streams efficiently in each channel with minimizing data bottleneck among channels. In the retrieval and playback subsystem, efficient index file architecture and video data storage architecture, efficient playback architecture which can make playback be processed parallel among channels are designed and implemented for faster retrieval and playback. Through experiments, our proposed system is shown to be maximally 2 times as fast in storing speed and maximally 3.5 times as fast in retrieval and playback speed as the previous system.

■ Keyword : | Multichannel visual monitoring | Video stream | DirectX | DVR |

* 본 연구는 승실대학교 교내연구비 지원으로 이루어졌음

접수번호 : #041208-001

접수일자 : 2004년 12월 08일

심사완료일 : 2005년 02월 06일

교신저자 : 정선태, e-mail : cst@ssu.ac.kr

I. 서론

영상 감시 시스템은 카메라에서 획득한 영상을 통하여 현장을 실시간으로 감시하고, 추후 검색을 위해 획득된 비디오 데이터를 저장한다. 영상 감시 시스템의 하나인 DVR(Digital Video Recording System)은 종래의 아날로그 기반 시스템과는 달리, CCTV 카메라에서 획득한 아날로그 영상을 디지털 영상 데이터로 변환하고 이 디지털 영상 데이터를 기반으로 디스플레이, 저장, 검색 및 재생 등을 처리하는 디지털 영상 감시 시스템이다. DVR에서는 디지털 영상 데이터를 처리하므로 발전된 디지털 신호 처리 기술의 적용이 가능하며 이에 따라 다채널 실시간 디스플레이, 다채널 영상 데이터 압축·저장, 저장된 다채널 영상 데이터 검색 및 재생 등의 다양한 고급 기능들이 지원되고 있다. 이에 따라, DVR은 새로운 기술의 영상 감시 시스템으로 각광받고 있다 [1]. DVR의 운영체제로는 마이크로소프트 윈도우즈와 리눅스가 주로 채택되어 사용되고 있다.

윈도우즈에서는 효율적인 멀티미디어 응용 구현을 지원하기 위해 DirectX 플랫폼을 제공하고 있으며, 이에 따라 게임 및 영상 감시 등 여러 멀티미디어 응용 분야에서 DirectX 기반 구현에 관한 연구와 제품 개발에 대한 관심이 많이 이루어지고 있다[2,3,4,5].

그런데, 영상 감시 시스템의 경우, 주로 처리하여야 할 데이터는 동영상 스트림 데이터로, 용량이 크고 시간 제약이 존재하는(따라서 동기화가 요구되는) 연속 미디어 데이터이다[6]. 더구나, DVR과 같은 다채널 영상 감시 시스템에서는 다채널 실시간 영상 디스플레이/저장/검색 및 재생 등의 기능 지원이 이루어져야 하므로, 주어진 시간 제약(동기화)를 갖는 대용량 멀티미디어 스트림 데이터의 원활한 처리가 요구된다. 따라서, 다채널 영상 감시 시스템에서 다채널 실시간 디스플레이/저장/검색 및 재생 등 여러 부분에 걸쳐 만족할 만한 성능을 달성하기 위해서는, 단순한 DirectX 기술 적용만의 설계로서는 곤란하고 보다 효율적인 다채널 비디오 데이터 스트림 처리가 가능한 구조의 설계가 필요하다. 현재 다채널 실시간 디스플레이/저장/검색 및 재생에 있어서의 효율적인 설계 구조에 대한 연구는 별로 보고되어

있지 않다[7,8,9,10].

본 논문에서는 윈도우즈 기반 다채널 영상 감시 시스템의 디스플레이, (녹화)저장, 그리고 재생의 각 부분에 대한 본 저자의 이전 연구[7,8,9,10]에 이어, DirectX 기반 다채널 영상 감시 시스템을 보다 효율적으로 설계·구현한 연구 결과를 기술한다. 또한, 본 논문에서 설계·구현한 시스템의 우수성을 보이기 위해 수행한 성능 분석의 결과를 제시한다.

본 논문의 구성은 다음과 같다. 제2절에서는 본 논문의 배경 지식이 되는 DirectX에 대해 간단히 소개되며, 제3장에서는 제안된 DirectX 기반 다채널 영상 감시 시스템의 전체 구조, 각 서브시스템의 설계 및 구현에 대해 기술된다. 제4장에서는 실험 및 성능 분석 결과가 설명되며, 마지막으로 제5장에서는 결론이 기술된다.

II. DirectX 개요

마이크로소프트 윈도우즈에서는 멀티미디어 데이터의 효율적인 처리를 위하여, DirectX 플랫폼을 제공하고 있다. DirectX는 DirectDraw, Direct3D, DirectX Audio, DirectInput, Direct Play, DirectShow, DirectSetup 등으로 구성되어 있으며, 현재 버전은 9.0이다[11]. 이 가운데, 본 논문에서 필요한 Direct Draw와 DirectShow에 대해서 간단히 소개한다.

1. DirectDraw

윈도우즈 운영체제 환경에서는 영상의 빠른 처리를 위해 직접 프레임 버퍼로의 작업을 지원하는 DirectDraw를 제공하고 있다. DirectDraw는 GDI(Graphic Device Interface)가 지원하는 상위레벨 드로잉 함수(문자 출력, 2D 그리기 함수 등) 등을 지원하지 못하지만, 응용이 직접 디스플레이 메모리에 접근할 수 있도록 허락하며, GDI에서는 지원하지 못하는 최신 그래픽 하드웨어의 고급 특징들(하드웨어 블리팅(blitting), 스트레칭, 오버레이 등)을 이용할 수 있도록 지원하고 있다. 따라서 빠른 디스플레이 처리가 필요 하는 응용에서는 DirectDraw를 사용할 필요가 있다. 본 논문에서 DirectDraw를 이용하여 설계·구현한 디스플레이 서브

시스템의 연구내용을 이해하기 위해 필요한 디스플레이 및 DirectDraw 관련 개념 및 용어는 [11,12]를 참조하기로 한다.

2. DirectShow

DirectShow는 윈도우즈 환경의 스트리밍 미디어 아키텍처로, 현재 DirectX 패키지에 통합되어 제공되고 있다[11]. DirectShow는 멀티미디어 스트림 처리 기능이 필터들의 조합으로 유연하게 구현되도록 지원하고 있다(그림 1 참조).

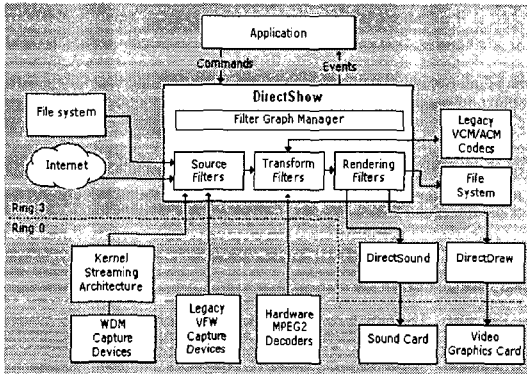


그림 1. 응용, DirectShow 구성요소, 장치(드라이버)들의 관계[11]

DirectShow 필터는 소스 필터(Source Filter), 변형 필터(Transform Filter), 렌더러 필터(Renderer/Rendering Filter) 등의 세 가지 종류로 구분된다. 파일 혹은 캡처 장치로부터 멀티미디어 데이터를 획득하는 역할을 수행하는 필터는 소스필터, 미디어 데이터를 가공 처리(인코딩, 디코딩, 압축, 파싱, 기타 가공 처리 등)하는 역할을 수행하는 필터는 변형필터, 미디어 데이터를 외부장치로 디스플레이를 하거나 디스크에 저장 혹은 네트워크를 통해 전송하는 등의 역할을 수행하는 필터는 렌더러 필터라 한다.

이러한 필터들은 연결이 되어, 하나의 필터그래프를 구성하며, 연결된 필터들은 필터그래프 매니저(Filter Graph Manager)에 의해 제어된다. 연결되어진 필터 사이에는 미디어 샘플(Media Sample)이라는 데이터 구조체의 전달을 통해, 미디어 데이터 포맷, 미디어 데

이터에 대한 시간 정보, 미디어 데이터의 버퍼 포인터가 전달되도록 되어 있다.

멀티미디어 스트림 처리에서는 처리하여야 할 데이터 량이 많으므로 가급적 불필요한 데이터 복사가 생기지 않도록 주의할 필요가 있는 데, DirectShow에서는 미디어 샘플을 이용하여, 다음 단계의 필터로 불필요하게 데이터 버퍼의 내용이 복사되지 않고 데이터 버퍼 포인터만이 전달되도록 지원하고 있다. 또한, DirectShow에서는 기본적으로 다양한 영상/음성 코덱을 변형 필터 형태로 지원하고 있다.

따라서 DirectShow를 이용하면, 유연하고 효율적인 멀티미디어 스트림그림 1. 응용, DirectShow 구성요소, 장치(Driver)들의 관계[11] 처리가 가능하고 다양한 영상/음성 코덱 이용도 가능하므로, 현재 윈도우즈 멀티미디어 스트림 응용 구현에 많이 사용되고 있다.

DirectShow 에 대한 보다 구체적인 내용은 [11]을 참조하기로 한다.

III. 제안된 DirectX 기반 다채널 영상 감시 시스템 설계 및 구현

본 절에서는 본 논문에서 연구된, 성능이 우수한 DirectX 기반 다채널 영상 감시 시스템의 설계 및 구현 내용을 기술한다.

1. 시스템 기능 사양 및 H/W 환경

현재 다채널 영상 감시 시스템에 요구되는 주요 기능들은 다음과 같다[1].

- 다중 카메라 채널(최대 16채널) 동시 실시간 디스플레이 지원
- 영상 데이터 실시간 녹화 저장(최대 16채널)
- 움직임이 있는 경우에만 녹화하는 모드(움직임 탐지 기능) 지원
- 다양한 화면 분할 모드(1채널 Full Screen, 4, 9, 10, 13, 16 채널 모드) 디스플레이
- 저장된 영상의 시간대별/이벤트별 검색 지원
- 카메라 Pan/Tilt/Zoom 제어 및 디지털 입출력(보통 8~16개) 지원

본 논문에서 구현한 시스템도 상기 기능을 기능 사양으로 갖는다. 성능 사양은 출시된 다채널 영상 감시 시스템(DVR)마다 다르다. 다채널 영상 감시 시스템의 성능은 S/W 구조뿐만 아니라, 어떤 H/W 사양(캡처카드, CPU, 메모리, HDD 등)을 사용했느냐에 따라 다르기 때문에, 본 논문에서 구현한 시스템과 H/W 사양을 정확히 알 수 없는 기존의 다른 상용 시스템(DVR)과의 단순한 성능 비교는 곤란하다.

제안된 다채널 영상 감시 시스템의 H/W는 CCTV 카메라 아날로그 영상을 획득하여 디지털 영상으로 변환하는 캡처 카드와 PC 머더보드, 모니터 등 크게 3부분으로 구성된다. 본 논문에서 구현한 시스템은 2개의 캡처 카드를 장착하여, 1개의 캡처 카드당 8개 카메라 채널, 전체 16 카메라 채널 영상 캡처를 지원한다. 사용한 영상 캡처 카드와 해당 장치 드라이버는 본 논문의 연구를 위해 본 저자의 연구실에서 제작하고 구현한 것으로, 캡처 카드의 비디오 디코더는 Conexant 사의 BT878A 칩을 이용하였다(그림 2).

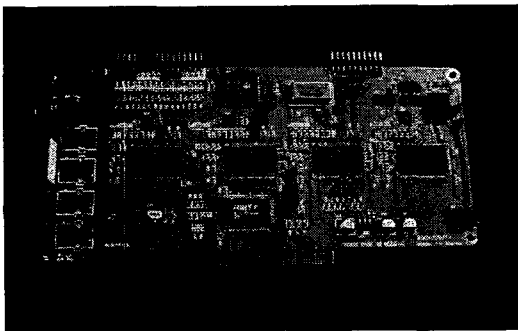


그림 2. 개발된 캡처/오버레이 보드

한편, 위에서 기술한 다채널 영상 감시 시스템의 기능 요구 사항을 보면, 다채널 영상 감시 시스템의 주요 기능이 크게 '다채널 실시간 디스플레이', '다채널 실시간 녹화 저장', '다채널 검색 및 재생' 등의 3가지임을 알 수 있다. 따라서 본 논문에서는 이러한 주요 3가지 기능 중심으로 연구 결과를 기술한다.

2. 전체 시스템 S/W 구조

제안된 DirectX 기반 다채널 영상 감시 시스템은 크

게 디스플레이 서브시스템, (녹화)저장 서브시스템, 검색 및 재생 서브시스템 등의 3부분으로 구성된다. 다음 그림 3은 본 논문에서 제안된 DirectX 기반 다채널 영상 감시 시스템의 S/W의 전체 구조를 나타낸다.

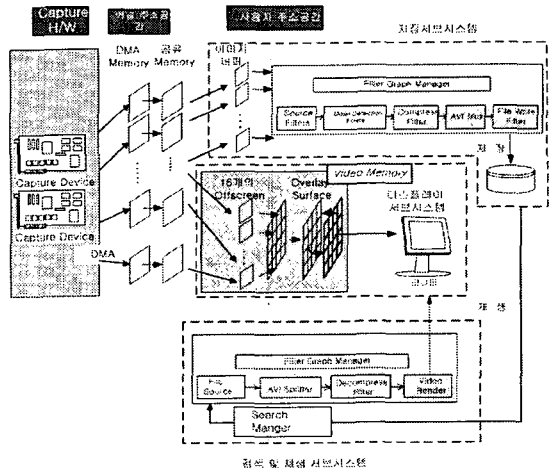


그림 3. 제안된 DirectX 기반 다채널 영상 감시 시스템의 전체 S/W 구조

영상 캡처카드에서 획득한 영상 데이터(최대 16 채널)는 DMA 되어 커널 공간에 위치하게 되고(DMA 메모리), 장치 드라이버는 이를 다시 사용자 공간과 공유되는 공유 메모리로 복사한다. 이후, 디스플레이 서브시스템은 여러 채널 카메라 영상 데이터를 조합하여, 요구되는 디스플레이 분할 모드로 화면에 실시간 디스플레이 시킨다. 디스플레이 속도 성능을 위하여 DirectDraw의 H/W 가속 기능과 overlay 기능을 이용하였다. 또한, tearing 현상 제거를 위하여 DirectDraw의 overlay surface와 flip 기능을 효율적으로 이용하여 구조를 설계하였다.

저장 서브시스템은 획득된 영상 데이터를 압축 저장하며 검색을 위한 추가적인 인덱스 정보를 기록한다. 동시에 여러 채널 압축 저장을 지원하기 위해, 채널별 DirectShow 저장 필터그래프를 설계하였다.

DirectShow 저장 필터그래프는 소스 필터, 움직임 탐지 기능을 지원하기 위한 Motion Detection 필터, 압축 필터, 윈도우즈가 지원하는 AVI 파일 포맷으로의 저장을 위한 AVI Mux 필터, AVI 파일 저장을 위한

File Writer 필터로 구성된다.

검색 및 재생 서브시스템은 사용자가 입력한 검색 키워드에 일치하는 영상 데이터를 찾아내는 검색 부분과 영상 데이터를 화면에 디스플레이 하는 재생부분으로 구성된다. 검색 부분은 사용자가 요청한 시간, 채널, 이벤트 등의 검색 키워드 정보를 기반으로 해당 인덱스 파일을 찾아 이를 통해 키워드에 일치하는 동영상 파일의 위치와 파일 내에서의 재생을 시작할 위치를 연도록 하며, 재생 부분은 이후 재생을 위해 채널별로 재생 필터그래프를 구성하고, 이후 재생하여 디스플레이 하는 역할을 수행한다.

3. 디스플레이 서브시스템 설계 및 구현

3.1 디스플레이 서브시스템 S/W 구조

DirectShow 에서는 영상 디스플레이를 위하여, Video Renderer 필터를 지원하고 있으며, 이를 이용하면 쉽게 영상 디스플레이 구현이 가능하다. 그러나 다채널 영상 디스플레이의 경우, 채널별(스트림별)로 Video Renderer 필터를 사용하는 경우, 각 채널별 디스플레이가 독립적으로 이루어지게 되므로 채널간 시간 동기화를 이루기가 힘들다. 또한, 채널별 Video Renderer 필터의 사용은 채널별 별도의 쓰레드 생성과 자원의 중복 사용을 초래하며 이로 인해 불필요한 CPU 사용율이 증가되어 원하는 성능을 얻기가 어렵게 된다. 따라서 제안된 시스템의 디스플레이 서브시스템은 빠른 디스플레이, 채널간 디스플레이 동기 및 tearing effect의 개선을 위해 DirectX의 DirectDraw 기능을 효율적으로 이용하여 설계·구현하였다. 구현된 디스플레이 서브시스템의 보다 자세한 S/W 구조는 아래 그림 4와 같다.

캡처카드에서 획득된 영상(카메라 채널 번호 포함)은 시스템 커널 주소 공간(DMA 메모리)으로 DMA 되어 가져오며, 장치 드라이버는 이를 다시 커널 공간과 사용자 공간이 공유하는 공유 메모리로 복사한다. 공유 메모리를 둔 이유는 장치가 DMA 하는 동안, 응용에서도 작업이 이루어지도록 하기 위해서다. 사용자 주소공간의 디스플레이 처리 모듈(디스플레이 쓰레드)은 카메라 채널 정보를 읽어 각 공유 메모리에 담겨진 영상 데이터가 어느 채널용인가를 판단하여, 해당 카메라 채널별 오

프스크린 표면의 버퍼에 복사한다. 이후, 디스플레이 모드에 따라 오프스크린 표면은 카메라 채널 영상이 최종 다중 채널 영상이 구성될 임시 오프스크린 표면의 해당 구역으로 블리팅 된다.

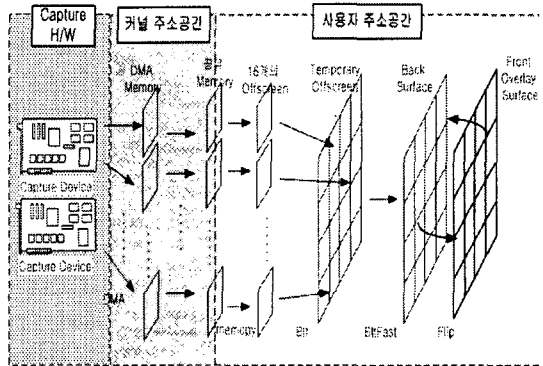


그림 4. 디스플레이 서브시스템의 S/W 구조

블리팅 목적지 영역의 크기가 소스 영상 영역보다 크기보다 큰 경우(예, 4채널 모드) 블리팅 시에 스케일링이 이루어진다. 이후, 임시 오프스크린 표면 전체는 오버레이 후방 표면에 블리팅 된다. 임시 오프스크린 표면의 영상 크기가 스케일링되지 않고 그대로 블리팅 되게 되므로 속도가 빠른 BltFast 메소드를 이용하였다. 플립 수행 시에 후방 오버레이 표면은 전방 오버레이 표면으로 전환되고 이 전환 시기는 수직 공백 기간에 동기화되도록 하였다[12]. 이 후, 전방 오버레이 표면의 내용이 화면 주사 주기에 따라 화면에 갱신되게 되며, 이때 DirectX의 H/W 가속 기능과 overlay 기능이 이용된다.

3.2 Tearing 개선 S/W 구조

프레임 버퍼의 영상 데이터 갱신 시간과 아날로그 모니터 화면에서의 영상 갱신 시간의 불일치가 발생하면, 이에 따라 화면에 선이 가거나 영상의 상하가 불일치하는 tearing 현상(artifact)이 발생하게 된다[8,12]. 다채널을 지원해야 하는 영상 디스플레이 시스템의 경우, 프레임 버퍼의 갱신 비율이 단일 채널 디스플레이 경우보다 높기 때문에(본 논문의 영상·디스플레이 시스템의 경우, 최대 초당 240 프레임), tearing 현상이 더 심하게 나타난다.

본 논문에서는 tearing 현상 개선을 위하여, 임시 오프스크린 표면과 오버레이 플리핑 체인(후방 표면 및 전방 표면)을 추가로 설계하고, 플리핑 체인의 플립(Flip) 주기를 스크린 주사 주기를 고려하여 선정하였다. 본 논문에서 제안한 tearing 현상 개선 방법의 효율성의 확인은 [8]을 참조하기로 한다.

4. 저장 서브시스템 설계 및 구현

영상 감시 시스템의 저장 서브시스템은 대용량의 멀티미디어 스트리밍 데이터를 실시간 압축 저장해야 하는 고성능의 작업을 요하는 부분이다. 빈번한 I/O 작업과 영상의 압축 작업이 일어나므로 효율적인 구조 설계를 위해서는 최대한 I/O 작업을 줄이며 효율적인 스트리밍 데이터의 처리 구조가 필요하다.

본 논문에서는 DirectShow 필터그래프를 도입하여, DirectShow의 향상된 멀티미디어 스트리밍 처리 능력을 활용하고, 다중 쓰레드 구조의 저장 시스템으로 구성하여 캡처 장치에서 획득되는 다채널의 영상 스트림을 각 채널별로 압축 및 저장되도록 지원하여 채널간의 데이터 병목현상을 최소화하도록 설계하였다.

본 논문에서 연구된 DVR의 저장 서브시스템은 크게 각 채널별로 획득된 영상을 압축 저장하는 필터그래프와 구성된 필터그래프를 제어하며 인덱스 정보를 기록하는 저장 매니저로 구성된다.

4.1 저장 서브시스템의 필터그래프 구성

각 채널별 필터그래프는 소스 필터, 움직임 탐지(Motion Detection) 필터, 압축(Compress) 필터, AVI Mux 필터, 파일 기록(File Write) 필터로 구성이 된다. 본 논문의 DVR 구현을 위해 제작된 비디오 캡처 장치 드라이버는 WDM 커널 스트림 클래스 구조를 사용하지 않고, 독자의 monolithic 드라이버 구조를 이용하여 개발한 것이다. 따라서 커널 스트림 클래스 장치 드라이버와의 연동을 위해 DirectShow에서 지원하는 소스 필터는 사용할 수 없으므로, 연동을 지원하는 소스 필터의 구현이 필요하며 이에 따라 이를 따로 구현하였다. 소스 필터는 전체 데이터의 흐름을 제어하게 되므로 시스템의 자원 활용을 고려한 설계가 필요하다. 또한, 움직임

탐지 기능 지원을 위한 추가적인 움직임 탐지 필터를 구현하였다. 그 밖의 압축 필터, AVI Mux 필터, 파일 기록 필터 등은 DirectShow가 제공하는 것을 사용하였다. 압축 필터는 프로그램 코드 레벨에서의 수정 없이 유동적으로 압축 코덱 변경이 가능하도록 하였다.

각 채널별로 필터그래프의 구성이 가능하므로 보다 효율적인 다중 쓰레드 구조의 저장 서브시스템 동작이 가능하며, 채널별로 서로 다른 영상 압축 코덱의 사용이 가능하다. 또한 필터그래프에 다른 필터가 추가될 수 있도록 유연한 구조를 지원함으로써 다양한 포맷으로의 저장 및 기능적 확장이 가능하다. 제안된 시스템에서는 기본적으로 AVI 포맷으로 영상 데이터를 저장한다. AVI 포맷으로의 저장 방식을 통해 채널간이나 시간별로 서로 다른 압축 코덱으로 영상 데이터가 저장이 되어 있다하더라도 일관된 재생 구조를 유지할 수 있다.

4.2 저장 매니저

저장 매니저는 녹화하고자 하는 채널, 압축 코덱, 움직임 탐지 기능사용 유무 등 사용자가 선택한 정보를 바탕으로 각 채널별로 유동적으로 필터그래프를 구성하며 이를 제어한다. 또한 추후 검색을 위한 인덱스 정보를 인덱스 파일에 기록하며, 필터그래프의 이상 발생 등의 상태에 따른 유동적인 처리를 담당하며 사용자의 녹화 유무 선택과 같은 설정의 변경에 따라 필터그래프를 제어하는 역할을 수행하도록 설계·구현하였다.

5. 검색 및 재생 서브시스템

5.1 설계 고려 사항

현재의 다채널 영상 감시 시스템(DVR) 제품들은 16 카메라 채널까지 지원하고 있다. 따라서 검색 및 재생 서브시스템이 효율적으로 설계되지 않으면, 다채널에 대해 지정한 검색 키워드에 해당하는 영상 데이터를 찾는 성능이 제대로 얻어지기 어렵게 된다.

검색에 있어서 가장 중요한 검색 키워드는 시간이다. 이는 영상 감시 시스템에서 감시 장면의 발생 시간 정보가 매우 중요하며, 영상 데이터는 시간에 연속적인 동영상 데이터 이므로 채널 또는 이벤트별 검색인 경우라도 재생 시에는 해당 채널 또는 해당 이벤트별로 시간

에 따른 영상 장면 재생이 요구되기 때문이다.

원하는 영상 데이터를 검색하고 재생할 때에 먼저 해당 영상 데이터를 갖는 영상 파일을 먼저 찾아야 하므로, 영상 데이터 저장을 위한 저장 구조가 검색·재생 성능에 영향을 미치게 된다. 따라서 다채널 영상 데이터 저장 구조는 영상 감시 시스템에서의 검색 구조를 고려하여 설계되어야 한다.

또한 검색 시에 필요한 영상 프레임에 대응량의 영상 파일에서 효과적으로 찾기 위해서는 효율적인 인덱스 파일의 설계가 필요하다. 인덱스 파일의 설계에는 다채널 영상 감시 시스템의 검색 요구 사항과 영상 파일 구조가 고려되어야 한다. 또한, 검색 및 재생 서브시스템 설계 시에 본 논문의 다채널 영상 감시 시스템이 기반하고 있는 DirectShow 구조 또한 고려되어야 한다.

5.2 검색 및 재생 서브시스템의 S/W 구조

본 논문에서 설계·구현한 DirectShow 기반 영상 감시 시스템의 검색 및 재생 서브시스템의 보다 자세한 S/W 구조는 다음 그림 5와 같다.

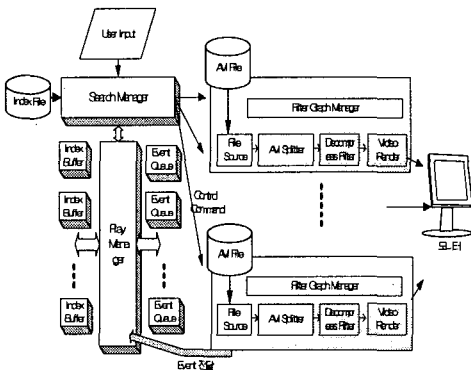


그림 5. 설계된 검색 및 재생 서브시스템의 동작 구조

검색 및 재생 서브시스템은 사용자가 입력한 검색 키워드에 일치하는 영상 데이터를 찾아내는 검색 부분과 영상 데이터를 화면에 디스플레이 하는 재생 부분으로 구분된다. 검색 부분(서치 매니저)은 사용자가 요청한 시간, 채널, 이벤트 등의 검색 키워드 정보를 기반으로 해당 인덱스 파일을 찾아 시스템의 메모리(인덱스 버

퍼)로 적재한 후 인덱스 정보를 분석한다. 이를 통해 키워드에 일치하는 동영상 파일의 위치와 파일 내에서의 재생을 시작할 위치를 얻게 되면 재생을 위해 채널별로 재생 필터그래프를 구성한다. 따라서 다채널 선택시 필터그래프는 선택한 채널의 수만큼 구성된다.

재생 부분에서는 구성된 필터그래프에 명령을 전달함으로써 동영상 데이터를 재생하여 디스플레이 하는 역할을 한다. 그런데, 재생이 구성된 각 채널별 필터그래프에 의해 각기 독립적으로 동작을 하므로, 각 채널에서 재생되는 영상 데이터의 시간 동기 제어가 필요하다. 재생 매니저(Play Manager)는 각각의 필터그래프를 제어하고 관리하며, 채널간의 시간 동기화 문제의 해결하도록 하기 위해 설계되었다. 이에 대한 보다 자세한 설명은 [10]을 참조하기로 한다.

제안된 시스템 구조에서의 검색 성능은 인덱스 파일 구조와 영상 파일 저장 구조의 단순함(뒤에 자세히 기술)과 DirectShow가 지원하는, 영상 파일에 있어서의 해당 영상 데이터의 효율적 탐색 때문에 기존 시스템 구조에 비해 많이 개선된다. 또한, 재생이 채널별로 작업이 진행되는 다중 스레드 재생 구조를 가지고 있어 재생이 매우 빠르게 진행된다.

5.3 영상 데이터 저장구조 및 인덱스파일 구조 설계

1) 영상 데이터 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. 또한, 영상 파일 저장 구조와 파일 이름은 다음과 같이 설계되었다.

/Data/년도월일/시간/채널번호분초.avi

예) 2002년 2월 3일 4시05분06초에 녹화를 시작한 0 채널의 영상 파일의 경우

/Data/20020203/04/000506.avi

2) 인덱스 파일 포맷 및 저장 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 한 개의 시간 인덱스 파일을 작성하고, 이벤트 검색을 위해서는 하루에 한 개의 사건 인덱스 파일을 작성하도록 설계하였다.

DirectShow에서는 AVI 파일의 시간별 및 프레임 번호별 검색을 지원하므로, 종래 영상 감시 시스템의 인덱스 파일처럼 프레임별로 해당 프레임이 저장된 시간 정보 등을 기록할 필요가 없다. 따라서 시간 인덱스 파일은 AVI 파일마다의 '시간 인덱스 블록'으로 구성되며, 시간 인덱스 블록에는 해당 AVI 파일에 대한 검색 정보, 즉 기록 시작 시간, 기록 정지 시간, 파일 이름, 채널 정보, 기록 모드(연속하여 기록, 또는 움직임이 탐지된 경우에만 기록) 등의 정보를 보유하도록 설계하였다(그림 6 참조).

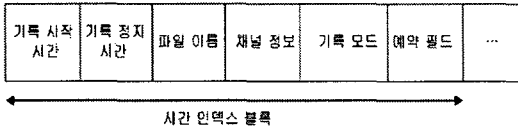


그림 6. 시간 인덱스 파일 구조

이벤트 인덱스 파일의 경우, 이벤트별 정보 구조로 구성되며(이벤트 인덱스 블록), 이벤트 인덱스 블록에는 해당 이벤트 정보(센서 번호 등), 해당 이벤트와 연동되어 녹화되는 카메라 채널 번호, 해당 프레임 정보(프레임 번호 및 프레임 기록 시간) 등의 정보를 보유한다. 인덱스 파일의 저장구조는 다음과 같다.

- 시간 인덱스파일 ;
 /Data/년도월일/Index/채널번호.idx
- 사건 인덱스파일 ;
 /Data//년도월일/Index/events.idx

V. 실험 및 성능 분석

본 논문에서 제안된 시스템을 구현하여 이미 구현되어 있는 기존 시스템과의 성능 비교 실험을 수행하였다.

본 논문에서 제안된 시스템의 디스플레이 서브시스템의 성능 실험 및 확인 결과는 [8,9]을 참조하기로 하며, 여기서는 저장 서브시스템과 검색 및 재생 서브시스템의 성능 분석 결과만을 기술한다.

1. 실험 환경

실험에 사용한 환경은 다음 표 1과 같다.

표 1. 실험 환경

| 항 목 | 사 양 |
|---------|---|
| CPU | Intel Pentium IV 1.7GHz |
| Memory | SDRAM PC800 256M |
| OS | Microsoft Windows2000 SP2 |
| 기타 | DirectX 9.0 Windows Media Codec 9.0 |
| 영상 캡처카드 | 제작된 캡처 보드 2개 성 능 : 8ch Max. 240 Frames/sec 16ch Max. 160 Frames/sec |

캡처 카드는 320x240 해상도의 YUY2(YUV422) 형태의 영상을 획득하도록 설정하였다. 캡처 카드는 각 카드당 4개의 영상 디코딩 칩(BT878)이 내장되어 있고 칩당 1개 채널의 처리 시에 최대 240 Frames/sec의 영상을 획득하고 MUX Switching을 사용하여 칩당 2개의 채널의 처리시는 160 Frames/sec의 영상을 얻을 수 있는 성능을 가지고 있다.

2. 저장 서브시스템 성능 분석

본 논문에서 제안된 시스템과 기존 시스템과의 저장 성능을 비교 측정하는 실험을 수행하여, 본 논문에서 제안된 시스템의 개선된 저장 구조의 성능 개선을 확인하였다. 압축 코덱의 설정은 기존 영상 감시 시스템의 MJPEG 코덱과 유사한 PIC MJPEG 코덱[13]을 사용하고 추가적으로 MS MPEG4 Part2를 이용한 실험을 병행하였다. 성능 측정은 1분간의 영상을 저장하였고 총 30회의 실험을 반복하여 이에 대한 평균값을 구하였다. 저장 성능을 측정한 결과는 아래 그림 7과 같다.

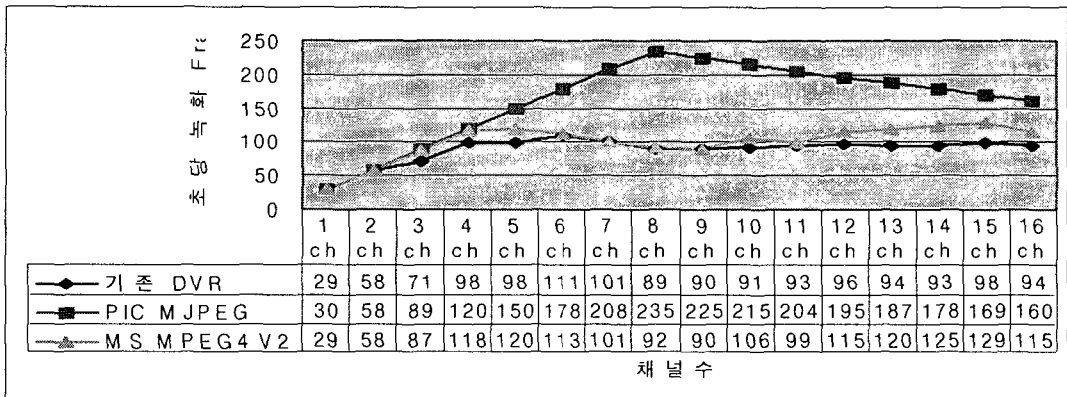


그림 7. 저장 성능 측정 실험 결과

그림 7의 실험 결과는 시스템의 부하가 적게 걸리는 1~5 채널만을 녹화하는 경우, 기존 시스템과의 성능 차이가 크게 나지 않지만, 시스템 부하가 더 많이 요구되는 6~16 채널 녹화 시에는 제안된 시스템이 훨씬 개선된 성능을 보여주어, 기존의 영상 감시 시스템에 비해 최고 200%이상(MJPEG 사용의 경우), 저장 성능 향상을 보였다. 이러한 결과의 이유는 저장 서브시스템의 구조 개선에 의한 것이라 볼 수 있다. 기존의 단일 스트림 구조에서는 스트리밍 데이터의 양이 증가함에 따라 성능의 저하가 더 심해지는데 이러한 이유는 데이터 처리 작업의 병목 발생으로 데이터 처리 작업으로 진입을 하지 못하고 버퍼에서 대기하거나 드롭 되어 버리는 데이터의 양이 증가하기 때문이다. 개선된 구조에서는 스트림의 흐름을 각 채널별로 분산함으로써 스트리밍 데이터의 흐름을 좀더 원활히 분배하고 있다.

MS MPEG4와 같이 데이터 처리에 있어 MJPEG 보다 훨씬 많은 리소스가 사용되는 코덱의 경우에는 기존 영상 감시 시스템과 마찬가지로 성능저하가 발생하지만 스트리밍 데이터 처리의 분산을 통해 기존과 비슷한 성능을 나타내는 결과를 보여 주었다.

또한 추후 검색을 위한 인덱스 정보 기록 과정에 있어서, 기존 시스템의 경우 영상 프레임의 획득이 발생 할 때 마다 인덱스 정보를 기록함으로써 빈번한 I/O 작업이 발생하게 되었지만, 제안된 시스템의 구조에서는 하나의 영상 파일에 대해 단 하나의 인덱스 블록을 기록함으로써 인덱스 정보 기록을 위한 I/O 작업의 발생을

감소시킬 수 있다. 이러한 구조의 개선이 전체적인 데이터 처리 성능을 개선시키는 결과를 이끌어 낸다.

3. 검색 및 재생 서브시스템 성능 분석

본 논문에서 제안된 시스템과 기존 시스템과의 검색 및 재생 성능을 비교 측정하는 실험을 수행하였다. 성능 측정은 24시간 동안 녹화된 16채널의 영상 및 인덱스 데이터에 대해 무작위로 검색시간을 지정하고, 이때의 검색 시작부터 각 채널별 최초 영상 프레임이 화면에 디스플레이 되기까지의 시간을 측정하는 것으로 하였다. 이러한 실험을 30회 반복하여 평균 시간을 구한 결과가 다음 그림 8과 같다.

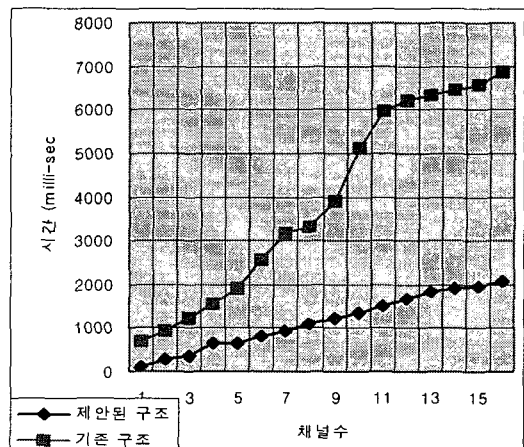


그림 8. 검색 및 재생 성능 측정 결과

측정 결과 제안된 구조가 기존 시스템 구조에 비해, 검색 및 재생 성능이 개선되었음을 볼 수 있었다. 특히 채널수가 증가할수록 성능의 차이가 커져 기존 시스템 보다 최고 3.5 배정도 까지 성능 차이를 보임을 알 수 있었다. 기존 시스템의 경우, 인덱스 구조에 프레임 단위로 정보가 기록되어 있어 인덱스 파일의 데이터의 양이 상당히 많아지게 된다. 따라서 기존 시스템에서는 인덱스 정보 분석에 시간이 많이 소요되며 검색하고자 하는 시간이 인덱스 파일에 뒤쪽에 있을수록 검색시간은 더 증가하게 된다.

또한 기존 시스템에서는 검색 및 재생이 채널별로 병렬 처리되지 않고, 단일 쓰레드에서 순차적으로 처리된다. 따라서 검색 및 재생하고자 하는 채널의 수가 증가하면, 상대적으로 성능의 저하가 더 커지게 된다. 제안된 구조의 경우 인덱스 데이터의 양이 상대적으로 적어 채널의 수가 증가해도 인덱스를 분석하고 영상 데이터에 접근하는 시간이 기존 시스템에 비해 상당히 적게 걸린다. 그러나 파일 재생을 위한 필터그래프 형성에 상대적으로 많은 시간이 걸리게 된다.

따라서 검색하고자 하는 채널이 증가함에 따라 검색 시간은 비교적 선형적으로 증가하는 것을 볼 수 있다. 그런데 필터그래프 형성에서 첫 프레임의 디코딩까지의 처리는 다중 쓰레드 구조로 동작함으로 인해 기존의 단일 쓰레드에 의한 순차적 처리에 비해서는 효율적임을 알 수 있다.

VI. 결론

본 논문에서는 윈도우즈의 멀티미디어 스트림 처리 지원구조인 DirectX에 기반을 둔 다채널 영상 감시 시스템의 구조를 제안하고, 구현한 결과 및 성능 분석 결과를 기술하였다. 기존의 구조와 비해, 디스플레이, 저장, 검색 및 재생 성능이 개선되었음을 실험을 통해 확인하였다.

본 논문의 연구 결과는 향후 영상 감시 시스템뿐만 아니라, 성능 좋은 멀티미디어 처리 시스템의 설계 및 구현에 일조할 수 있을 것으로 기대된다.

참고문헌

- [1] DVR 시스템 개요, <http://www.rogertowne.com/DVR.htm>
- [2] Microsoft DirectX Home page, <http://www.microsoft.com/windows/directx/default.aspx>
- [3] L. Cline et al., "DirectShow RTP support for Adaptively in Networked Multimedia Applications," *Proc. of IEEE Int. Conf. on Multimedia Comp. and Sys.*, pp. 13-22, 1998.
- [4] 한국 DVR 산업협회, <http://www.koreadvr.com>
- [5] 게임개발자 온라인 커뮤니티, <http://www.game-dev.net/reference/list.asp?categoryid=24>
- [6] R. Steinmetz and K. Nahrstedt, *Multimedia: Computing, Communications & Applications*, PTR, 1995.
- [7] 정연권, 하상석, 정선태, "다채널 영상 감시 시스템을 위한 다중 포맷 동영상 저장 DirectShow Filter 설계 및 구현", *대한전자공학회 하계 학술대회 논문집*, 2002.
- [8] 정연권, 이동학, 정선태, "윈도우즈 기반 영상 감시 시스템에서의 Tearing 현상 개선", *한국통신학회 논문집*, 제27권, 제11C호, pp. 1097-1105, 2002.
- [9] 정연권, 정선태, "다채널 및 다중 화면 분할 모드를 지원하는 영상 감시 장치의 디스플레이 시스템 설계 및 구현", *IEEK Summer Conference*, 2002.
- [10] 양정훈, 정선태, "윈도우즈 기반 다채널 영상 감시 시스템에서의 실시간 재생", *2003 대한전자공학회 하계 학술대회 논문집*, 제26권, 제1호, pp. 2116-2119, 2003.
- [11] DirectX SDK Documentation, <http://www.microsoft.com/windows/directx/default.aspx>
- [12] Damon Chandler and Michael Ftsch, *Windows 2000 Graphic API Black Book*, Coriolis, 2001.
- [13] 페가수스 이미지징 코덱, <http://www.jpg.com/>

저 자 소개

정 선 태(Sun-Tae Chung)

정회원



- 1983년 : 서울대학교 전자공학과 (공학사)
- 1990년월 : 미시간대학(미) 전자 및 컴퓨터공학과(공학박사)
- 1991년~현재 : 숭실대학교 정보통신전자공학부 교수

<관심분야> : 영상처리, 컴퓨터 비전, 영상감시 임베디드 시스템, 실시간 시스템