

# 다채널 및 다중 화면 분할 모드를 지원하는 영상 감시 장치의 디스플레이 시스템 설계 및 구현

정 연권, 정선태  
숭실대학교 정보통신전자공학부  
전화 : 02-817-5987/ 핸드폰 : 011-9765-4738

## Design and Implementation of A Display system for a Visual Monitoring system supporting multichannel display in multi screen division display modes

Yeon-Gweon Jeong, Sun-Tae Chung  
Dept. of Electronics Engineering, Soongsil University  
E-mail : jygyjg@syscon.soongsil.ac.kr

### Abstract

DVR(Digital Video Recorder), a recently popular visual monitoring system, is required to support multi camera channel display(upto 16 channels) with various display modes in real-time. For such a multichannel display system, it is well known that tearing artifacts become more serious. Thus, one needs to design a display system for DVR so that it does not show tearing artifacts as much as possible, but keeps real-time display speed. In this paper, we present our efforts in designing and implementing a display system which rarely ever shows tearing artifacts, but without degradation of required real-time display speed, and which displays character information stably without blinking.

### I. 서론

영상 감시 시스템은 카메라에서 획득한 영상을 통하여 현장을 실시간으로 감시하고 추후 검색을 위해 획득된 비디오 데이터를 저장하며, 원격 감시를 지원한다.

현재 영상 감시 시스템으로 각광을 받고 있는 DVR(Digital Video Recorder)은 종래의 아날로그 기반 시

스템과 달리 CCTV카메라에서 획득한 아날로그 영상을 디지털 영상으로 변환하고 이를 컴퓨터 시스템으로 받아들여 모니터에 디스플레이하고 원격감시를 위해서 전송하며, 추후 검색을 위해 저장하는 새로운 기술의 영상 감시 시스템으로 각광을 받고 있다[1]. 현재 출시되고 있는 DVR은 PC 기반으로 설계 제작된 제품과 독립적인 임베디드 시스템으로 설계 제작된 제품으로 대별된다. PC 기반 제품의 경우, 운영체제로 윈도우즈(윈도우즈 95, 98, me, NT, 2000, XP 등)를 사용하는 경우와 리눅스를 사용하는 경우로 대별되며, 임베디드 DVR은 임베디드 리눅스, 실시간 운영체제 등을 사용한다.

DVR은 TV 수신 시스템이나 기타 영상 시스템과는 달리 다채널(보통 최대 16채널) 및 다중 화면 분할 모드(전화면, 4분할, 6분할, 9분할, 10분할, 13분할, 16분할 모드 등) 디스플레이 등을 지원해야 하는 요구 조건을 가지고 있다. 또한, 다채널 다중 화면 분할 모드 디스플레이가 실시간으로 이루어져야 하고, 또한 디스플레이시에 tearing 현상이 없으며, 디스플레이되는 카메라 영상 화면에 찍혀나오는 문자 정보는 깜박이지 않고 안정적으로 보여야 한다.

본 논문은 윈도우즈 운영체제 기반에서 다채널 다중 화면 분할 모드의 효율적인 실시간 디스플레이 시스템을 설계하고 구현한 결과를 기술한다.

## II. DirectDraw 및 DVR 디스플레이 시스템에서의 tearing 현상

### 2.1 DirectDraw

윈도우즈 운영체제 환경에서는 영상의 빠른 처리를 위해 직접 프레임 버퍼로의 작업을 지원하는 DirectDraw를 제공하고 있다. DirectDraw는 GDI(Graphic Device Interface)가 갖는 상위레벨 드로잉 함수(문자 출력, 2D 그리기 함수 등)를 제공하지 못하지만, 응용이 직접 디스플레이 메모리에 접근할 수 있도록 허락하며 GDI에서는 가능하지 않은 현대 그래픽 하드웨어의 고급 특징(하드웨어 브리팅(blitting), 스트레칭, 오버레이 등)을 이용할 수 있도록 지원한다. DirectDraw는 DirectX API 세트의 일부분이며, 모든 윈도우 운영체제에서 지원된다. 현재 버전은 8.1이다.

DirectDraw를 이용하여 구현한 디스플레이 시스템을 이해하기 위해 필요한, 디스플레이 및 DirectDraw 관련 용어를 여기에 정리한다.

#### 1) 스캐닝(scanning)

- 그래픽 어댑터의 RAMDAC은 프레임 버퍼에 있는 영상 데이터를 읽어 아날로그 신호로 변환하여 이를 아날로그 모니터에 전송한다. CRT 모니터의 경우, 이 아날로그 신호에 따라 CRT의 전자총에서 전자빔(Cathode Ray)의 강도 및 방향을 조정하고, 이에 따라 전자빔이 스크린에 코팅된 형광물질을 발광시켜 컬러가 보이도록 한다. 그래픽 어댑터는 프레임 버퍼의 영상 데이터를 왼쪽 상단에서부터 오른쪽 하단까지 한줄 단위로 읽게 되는 데, 이를 래스터 주사(raster scan) 방식이라 한다.

#### 2) VBI(Vertical Blank Interval)

- 래스터 스캐닝시, 주사선이 오른쪽하단에 도달하여 다시 왼쪽 상단까지 돌아가는 데 걸리는 시간 간격

#### 3) Tearing artifacts

- 프레임 버퍼의 영상정보 갱신 시간과 아날로그 모니터 화면에서의 영상 갱신 시간의 불일치가 발생하여 영상 화면에 선이 그어지거나, 한 프레임에서 영상의 상하가 불일치하게 되는 현상을 말한다.

#### 4) DirectDraw Surface

- 디스플레이 메모리의 선형 메모리 영역을 말하며, 시스템 메모리나 비디오 메모리에 존재한다.

#### 5) Blit

- Blit는 Bit block transfer의 약자로써, 메모리의 한 주소 공간에서 다른 주소공간으로 데이터 블록을 전송하는 오퍼레이션을 말한다. 보통 영상 블록 데이터를

이동할 때 사용되는 오퍼레이션이다.

#### 6) Primary Surface

- 모니터에 디스플레이되는 표면을 말하며, 그래픽 어댑터는 프라이머리 표면의 영상 데이터를 스캐닝하여 모니터에 디스플레이한다.

#### 7) Paging flipping

- 연속적인 표면들의 체인을 구성하여 디스플레이하는 방법으로써, 처음 표면은 front surface, 그 뒤의 표면들은 back surface라 하며, 플립시에 전방 표면 뒤의 후방 표면이 전방 표면으로 바뀌게 된다. 따라서, 전방 표면이 디스플레이되는 동안 후방 표면에 디스플레이될 영상을 준비시킨 후에, 전방 표면의 디스플레이가 끝나면 플립하여 후방 표면이 디스플레이 되도록 할수 있어 빠른 영상 디스플레이 처리가 가능해진다. 또한 DirectDraw에서는 플립시에 수직 공백 구간(VBI; Vertical Blank Interval) 동기를 달성하도록 지원하고 있다. VBI 동기는 주사선이 VBI에 주사하기 시작하는 시점을 기다려서, 이때부터 영상정보를 갱신하는 것을 말하며, 이 VBI 동기는 tearing 현상을 해결하는 중요한 해법이다[4].

#### 8) Overlay surface

- 오버레이 표면은 프라이머리 표면에 오버레이하는 데 사용된다. 오버레이가 셋팅되는 경우, 프라이머리 표면위에 오버레이될 사각형 영역이 정의된다. 그래픽 어댑터가 모니터에 디스플레이하기 위해 프라이머리 표면에 있는 영상 데이터를 스캐닝할 때, 오버레이로 정의된 사각형 영역에 도달하게 되면, 영상 데이터를 프라이머리 표면에 있는 영상 데이터가 아니라 오버레이 표면에 있는 영상 데이터를 스캐닝한다. 이러한 작업은 그래픽 어댑터 하드웨어로 지원되기 때문에 디스플레이 속도에 저하가 없다.

DirectDraw는 오버레이 표면 지원하며 다양한 오버레이 효과를 제공한다. 오버레이 표면은 비디오 메모리에서만 생성된다.

### 2.2 DVR 디스플레이 시스템에서의 tearing 현상

Tearing 현상은 프레임 버퍼에 갱신되는 영상 데이터의 갱신 속도가 모니터의 화면 갱신 속도보다 빠른 경우에 발생한다[3] 그런데, 보통 한 채널 풀모션 비디오가 초당 30 프레임의 변경이 이루어지는 데 반해, 보통 16 채널 지원 DVR의 경우 최대 초당 240프레임 이상의 디스플레이를 지원하여야 한다. 따라서, 만약 갱신된 카메라 채널 영상 데이터를 직접 프레임 버퍼에 갱신하는 경우에는, 프레임 버퍼의 갱신 비율이 스크린의 주사선 갱신 주기(보통 초당 50 ~ 120)에 비해 훨씬 높기 때문에 빨리 변하는 동영상 디스플레이의 경우

tearing 현상이 심하게 나타난다. 따라서, DVR 디스플레이에서는 한 채널만을 사용하는 영상시스템이나 오버레이 하드웨어를 사용한 디스플레이 방식에 비해 tearing 현상 해결이 더 어렵다는 것을 알 수 있다.

Tearing 현상을 막는 방법의 하나는 프레임 버퍼로의 갱신을 주사선이 수직 공백 구간(vertical blank interval)에 주사하기 시작하는 시점을 기다려서, 이때부터 주사선이 좌측 상단에 갈 때까지의 시간에 영상정보를 프레임 버퍼에 갱신하여 스크린 주사 주기와 프레임 갱신 주기를 동기화시킴으로써, tearing 현상을 제거하는 것이다. 윈도우즈의 DirectDraw에서는

"IDirectDraw7::WaitForVerticalBlank", "IDirectDraw7::Flip" 와 같은 수직 공백 구간에 동기화를 달성할 수 있는 기능을 제공한다[4]. 그러나 단일 채널이 아닌 다채널의 경우 이러한 동기화에 의한 지연으로 성능의 감소를 초래할 수 있다.

### III. 윈도우즈 기반 DVR의 디스플레이 시스템 설계 및 구현

#### 3.1 DVR 디스플레이 시스템의 H/W

본 논문에서 설계하고 구현한 DVR 디스플레이 시스템은 CCTV 카메라 아날로그 영상을 획득하여 디지털 영상으로 변환하는 캡처 카드와 머더보드, 모니터 등 크게 3부분으로 구성된다. 캡처카드는 PCI 타입 카드로서, PCI 슬롯에 장착하도록 되어 있다. 캡처카드에는 아날로그 영상을 받아 디지털 영상으로 변환하는 비디오 디코더와 이를 PCI 버스와 연동시키는 PCI 브릿지, 1개의 비디오 디코더에 2개의 카메라 영상을 먹싱하는 믹스 스위치 등으로 구성된다. 비디오 디코더는 Conexant의 BT878A를 사용하고 PCI Bridge는 Hint의 HBI PCI-PCI bridge를 아날로그 믹스 스위치로는 8x16 Array Switch인 Mitel의 MT8816을 사용하였다. 구현한 DVR 디스플레이 시스템은 2개의 캡처카드를 장착하여, 1개의 캡처카드당 8개 카메라 채널, 전체 18 카메라 채널 영상을 지원한다.

캡처 장치드라이버는 1개의 장치마다 존재하며, 1개의 장치가 2개의 카메라 영상을 처리하므로, 16채널 지원시에 1개 장치 드라이버가 2개의 카메라 채널을 먹싱하여 처리하게 된다.

#### 3.2 DVR 디스플레이 시스템의 S/W 구조

##### 1) 메모리 구조

구현된 DVR 디스플레이 시스템의 S/W의 메모리 구

조는 다음 그림 [1]과 같다.

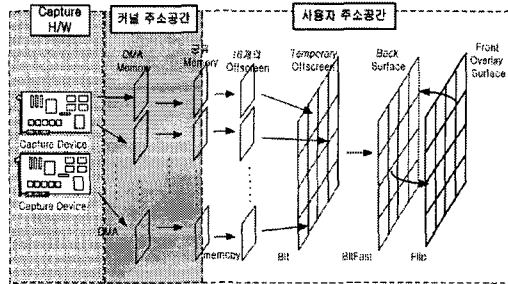


그림 3. 디스플레이 시스템의 S/W 구조

커널 공간에는 캡처 장치가 획득한 영상 데이터가 DMA되는 DMA 메모리 8개(장치당 1개씩)와 장치드라이버와 사용자 공간 디스플레이 쓰레드가 공유하는 공유 메모리 8개가 존재한다. 장치가 먹싱을 지원하는 경우, 공유 메모리에는 획득한 디지털 영상 데이터와 카메라 채널 정보등이 추가된다.

사용자 공간에는 디스플레이 쓰레드가 사용하는 오프스크린 표면 16개, 임시 오프스크린 표면, 오버레이 전방 표면, 후방 표면, 프라이머리 표면이 설계되어 있으며, 이들은 모두 비디오 메모리에 구현하였다. 임시 오프스크린 표면, 오버레이 전방표면, 후방 표면의 버퍼의 크기는 오프스크린 표면 버퍼의 16배 크기이며, 임시 오프스크린 표면부터 최종 다중 채널 영상 구성이 이루어진다. 공유 메모리에서 직접 임시 오프 스크린 표면에 복사하지 않고 중간에 오프스크린 표면에 복사하도록 한 것은, 직접 복사하는 경우 화면 모드에 따라 스케일링이 필요하게 되는 데 공유 메모리에서 같은 크기의 비디오 메모리에 있는 오프스크린 표면 버퍼에 복사한 후, 그래픽 어댑터의 하드웨어 가속 능력을 지원하는 DirectDraw 블링팅의 스케일링 기능을 이용하여 확대 복사하는 경우 시스템에 부하를 주지 않고 빠른 속도로 처리 할 수 있기 때문이다. 임시 오프스크린 표면과 오버레이 플립 체인을 둔 이유는 tearing 현상 개선을 위해서이다. 이에 대한 보다 자세한 내용은 [2]을 참조하라.

##### 2) 디스플레이 시스템 동작 구조

캡처카드에서 획득한 영상은 비디오 디코더 BT878 의 DMA 능력을 이용하여, 시스템 메모리(DMA 메모리) 공간으로 DMA 하여 가져오며, 디바이스 드라이버는 이를 다시 커널공간과 사용자 공간이 공유하는 공유 메모리장소로 복사한다. 공유 메모리를 둔 이유는 장치가 DMA 하는 동안, 응용에서도 작업이 이루어지도록 하기 위해서이다.

사용자 주소공간의 디스플레이 처리 모듈(디스플레이 쓰레드)은 카메라 채널 정보를 읽어 각 공유 메모리에 담겨진 영상 데이터가 어느 채널용인가를 판단하여, 해당 카메라 채널별 오프스크린 표면의 버퍼에 복사한다. 이후, 디스플레이 모드에 따라 오프스크린 표면은 카메라 채널 영상이 최종 다중 채널 영상이 구성될 임시 오프스크린 표면의 해당 구역으로 필요에 따라 스케일링이 되며 블리팅을 한다. 이후, 임시 오프스크린 표면 전체는 오버레이 후방 표면에 블리팅된다. 임시 오프스크린 표면의 영상 크기가 스케일링되지 않고 그대로 블리팅되게 되므로 속도가 빠른 BltFast 메소드를 이용하였다. 플립 수행시에 후방 오버레이 표면은 전방 오버레이 표면으로 전환되고 이 전환 시기는 수직 공백 구간에 동기화된다[4]. 이후, 전방 오버레이 표면의 내용이 화면 주사 주기에 따라 화면에 갱신되게 된다.

3) Tearing 현상 개선 구조

임시 오프스크린 표면과 오버레이 플립핑 체인(후방 표면 및 전방 표면)을 추가로 설계한 것은 tearing 현상을 개선시키기 위해서다. 플립 메소드 호출 주기는 25ms로 설정하였다. 디스플레이를 수행하는 쓰레드의 동작이 다른 쓰레드의 동작에 영향을 받기 때문에, 호출주기의 지터가 발생하게 되며, 실험적으로는 플립 메소드 호출이 25 ~ 30 ms 마다 이루어 지고 있음을 관찰되었다. 이런 시간 간격 (25 ~ 30 ms)의 경우, 임시 오프스크린 표면에 각 카메라 채널 영상이 평균 1회 이상 갱신될 수 있으므로, 결국 화면에는 각 카메라 채널 영상이 충분한 속도로 디스플레이 되게 된다.

4) 쓰레드 및 동기화 구조

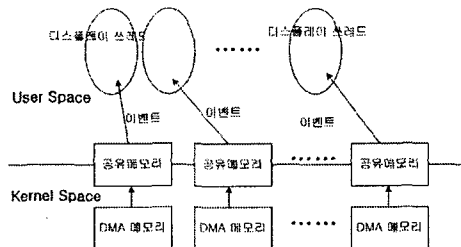


그림 4. 쓰레드 및 동기화 구조

2)에서 설명된 사용자 공간에서의 수행 내용은 디스플레이 쓰레드에서 수행된다. 장치 드라이버와 디스플레이 쓰레드의 동기는 커널공간과 사용자공간이 공유하는 이벤트를 통해 이루어 진다. 장치 드라이버는 DMA 메모리 내용과 추가 정보를 공유 메모리에 복사한 후, 이벤트를 통해 디스플레이 쓰레드를 깨어주며, 깨어난 디스

플레이 쓰레드는 공유메모리에 있는 카메라 채널 정보를 확인한 후에, 공유 메모리에서 영상 데이터를 해당 오프스크린 버퍼로 복사한다. 이후의 쓰레드의 동작은 플립시까지 계속 진행된 후, 다시 장치 드라이버로부터의 이벤트 도착을 기다리며, 다시 잠들게 된다.

5) 문자 정보 디스플레이

영상 감시시에 관리를 위하여 모니터 화면에 문자 정보(카메라 채널 번호, 현재시간 및 설치 장소 이름 등)가 디스플레이 될 필요가 있다.

감시 카메라 영상에 대한 문자 정보(카메라 채널 번호, 현재시간 및 설치 장소 이름 등)는 영상 감시 장치 동작중에는 화면 모드가 변경될 때나 모니터 화면이 가려졌다 다시 나타나는 경우를 제외하고는 다시 출력 내용이 갱신될 필요가 없는 고정 그래픽이므로, 문자정보가 사용하는 색상값과 다른 색상값을 프라이머리 기본 색으로 하여 오버레이가 이 특정 색상에만 투영되도록 하는 컬러 키 기법을 이용하도록 설계 구현하였다.

IV. 결 론

본 논문에서는 최근 영상 감시 시스템으로 각광을 받고 있는 DVR 시스템에서 tearing 현상을 개선하고, 문자 정보 출력이 안정적으로 디스플레이되며, 디스플레이 성능이 실시간으로 이루어지는 DVR 의 디스플레이 시스템을 설계하고 구현한 결과를 기술하였다. 구현 결과, 캡처 장치의 영상 캡처 성능 (먹싱하지 않는 경우 30fps, 먹싱하는 경우 채널당 10fps) 에 부합하는 실시간 디스플레이 성능을 보여줌과 tearing 현상이 크게 개선되었음을 확인 할 수 있었다.

현재, 리눅스 기반의 영상 감시 장치의 디스플레이 시스템을 구현하고 있으며, 추후 이에 대한 보고가 이루어 질 것이다.

참고문헌(또는 Reference)

[1] DVR 시스템 개요, <http://www.digitalplaza.co.kr/library/cctv-dvr.htm> :웹문서  
 [2] 이동학, 정연권, 정선태, “윈도우즈 기반 영상 감시 시스템에서의 Tearing 현상 개선”, 제12회 통신 정보 합동 학술 대회, 제주, 2002.4.  
 [3] Damon Chandler, Michael Ftsch, Windows2000 Graphic API Black Book, Coriolis (2001)  
 [4] DirectX7Platform SDK , <http://download.microsoft.com/download/win98SE/DXSDK/7.0a/W9X/EN-US/dx7adxf.exe>