

ESP : 녹화 및 검색 기능 향상을 위한 DVR전용 파일 포맷

박재경[†] · 양승민^{**}

요 약

DVR(Digital Video Recorder)시스템은 기존의 아날로그 테이프 저장 시스템과는 달리 영상 입력을 디지털 신호로 변환하여 압축 저장하는 장치로서 기존의 실시간 감시, 녹화, 검색 기능의 개선은 물론 워터마킹, 무선 감시 등의 부가기능 추가 및 응용이 가능하게 되어 아날로그 테이프 저장 장치를 대체해 가고 있다.

DVR 시스템에서 녹화 및 검색용 동영상 파일 포맷으로는 AVI 포맷이 가장 널리 쓰이고 있다. 그러나 AVI 포맷은 구조적인 문제로 DVR의 녹화와 검색을 위해 최적이라 할 수 없다. 일부 DVR 업체에서는 AVI 포맷을 대체하기 위한 자체 포맷을 개발하여 사용하고 있지만 공개된 파일 포맷은 없는 실정이다.

본 논문에서는 DVR 전용 파일 포맷인 ESP 포맷을 제안한다. ESP 파일 포맷은 AVI 포맷의 문제점을 해결하면서 AVI 포맷의 장점들을 최대한 적용하였다. ESP 포맷은 다중 스트림 녹화 및 이벤트 녹화도 지원하여 기존 DVR 시스템의 녹화 및 검색 기능을 개선시킨다.

키워드 : DVR, 파일 포맷, 녹화, 검색, ESP, AVI

ESP : A DVR File Format for Enhanced Recording and Searching

Park Jae-kyung[†] · Yang Seung-min^{**}

ABSTRACT

DVR(Digital Video Recorder) system stores video inputs in compressed digital formats and retrieve them. DVR system has several advantages over traditional analog tape recorder system which are (1) improved real-time monitoring, recording and searching capability, and (2) other capabilities such as watermarking and remote monitoring through network.

AVI format is the most popular format used for DVR systems. However, AVI format has drawbacks in recording and searching due to structural problem. Some vendors develop and use their own format, do not open the format to the public.

In this paper, ESP format is proposed. ESP format solves the drawbacks of AVI format, and the advantages of AVI format apply to ESP format. Moreover, ESP format provides multistream recording/replay and event-recording. In result, ESP format enhances functionality of recording and searching in DVR system.

Keywords : DVR, File Format, Recording, Seeking, ESP, AVI

1. 서 론

기존의 영상 감시 보안 시스템인 CCTV(Closed-Circuit Television)는 아날로그 데이터를 테이프로 녹화하고 이를 VCR(Video Cassette Recorder)로 판독하는 방식이다. 이 방식은 아날로그의 특성으로 인하여 녹화된 영상 데이터 관리와 판독이 불편한 단점을 가지고 있다.

DVR(Digital Video Recorder)은 언급된 기존 CCTV시스템의 단점과 디지털 영상 기술의 발달로 인해 등장하게 되

었다. DVR은 기존 CCTV시스템과 마찬가지로 실시간 감시, 녹화, 검색 등의 기능을 수행하는 내장형 시스템으로서 디지털의 장점이 그대로 적용되므로 빠른 검색 및 판독이 가능하며 녹화된 데이터의 전송 및 복제가 자유롭다. 원격 감시/검색, 워터마킹(Watermarking)등의 고급기능들도 추가가 가능해 짐에 따라 현재 아날로그 테이프 저장 장치를 대체해 나가고 있다.

DVR의 가장 핵심적인 기능으로는 감시 상황을 녹화하는 기능과 녹화된 데이터를 검색하는 기능을 들 수 있다. 녹화 기능은 실시간으로 캡처되고 있는 영상을 압축하여 디스크에 파일로 저장하는 기능이며 검색 기능은 이미 녹화되어 있는 파일들을 재생하여 원하는 영상을 찾는 기능이다. 이 두 가지 기능은 높은 프로세서 점유율과 극심한 디스크 접근, 기능의 복잡성으로 인해 구현이 매우 까다로운 실정이다.

※ 이 연구는 2007년도 숭실대학교 교내 연구비의 지원으로 연구되었음.

† 정 회 원 : (주)마크애니 주임연구원

** 정 회 원 : 숭실대학교 컴퓨터학부 교수

논문접수 : 2007년 12월 27일

수정일 : 1차 2008년 9월 12일, 2차 2008년 11월 10일

심사완료 : 2008년 11월 12일

DVR의 녹화와 검색에 쓰이는 파일 포맷에 관하여 아직 어떠한 표준같은 것은 정의된 것이 없으며 업계에서는 대부분 AVI(Audio Video Interleave) 포맷을 사용하고 있다. 이 파일 포맷은 PC용 동영상 파일 포맷으로는 가장 널리 쓰이고 있지만 DVR에서 녹화와 검색용 파일 포맷으로 쓰이기에는 여러 가지 한계점이 있다. 인덱스 리스트가 파일을 단을 때 기록되는 구조로 인해 녹화 중인 파일을 재생할 수 없으며 또한 녹화 중인 파일이 DVR의 예외 상황으로 인해 비정상 종료된다면 역시 복구 작업을 수행하기 전에 재생을 할 수 없는 문제점이 있다.

본 논문에서는 DVR의 녹화와 검색 기능의 향상을 위한 DVR 전용 파일 포맷인 ESP(Enhanced Streaming Packet)를 설계한다. DVR의 녹화와 검색 기능을 고려하여 AVI 파일의 단점은 개선시키고 장점은 적용하도록 하였다. 청크(Chunk)라는 고정 사이즈 파일 접근 단위를 도입함으로써 인덱스 리스트 없이 AVI의 무작위 탐색 성능에 근접할 수 있도록 하였다. 프레임별 타임스탬프를 가지도록 하여 녹화와 검색 기능의 프레임 레이트와 프레임 카운트에 대한 의존성을 없앴다. DVR의 특징인 다중 채널을 지원하기 위해 하나의 파일로 다중 채널을 녹화 및 재생 수행이 가능하도록 하였으며 DVR 이벤트 정보/사용자 정의 데이터를 파일에 추가할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장은 논문의 적용 대상인 DVR 시스템에 대해 소개하고 현재 쓰이고 있는 파일 포맷인 AVI 포맷에 대해서 분석한다. 3장에서는 이 논문에서 제안하는 ESP 파일 포맷의 설계를 다룬다. 4장에서는 테스트 프로그램으로 측정한 ESP 파일 포맷과 AVI 파일 포맷의 성능을 비교 분석하고 향후 연구 방향을 제시하며 결론을 맺는다.

2. 관련 연구

2.1 DVR 시스템

DVR 시스템은 영상 감시가 필요한 장소의 상황을 실시간 감시, 녹화, 검색, 저장/인쇄, 백업 등을 가능하게 해주며 기존의 아날로그 테이프 보안 시스템을 대체하는 디지털 보안 감시 시스템이다. 기존의 보안 시스템인 CCTV 시스템은 아날로그 테이프를 저장 매체로 사용하고 VCR로 검색/판독하는 방식의 시스템으로 아날로그의 특성상 많은 단점을 가지고 있다. DVR은 화상을 디지털로 처리함에 따라 디지털 기술의 장점이 그대로 적용되어 기존 아날로그 보안 감시 장비의 문제점을 해결하였다^[1,2].

DVR을 비롯하여 영상 보안 감시 장비의 핵심 기능은 녹화 기능과 녹화 데이터를 검색하는 기능이라고 할 수 있다. DVR에서 녹화 기능은 디지털로 변환된 카메라 영상을 MPEG이나 MJPEG 코덱(Codec)을 이용하여 압축한 후 파일 형태로 하드디스크에 저장하는 작업을 수행한다. 보안 감시 장비는 특성상 1년 365일 가동되어야 하므로 녹화 기능은 한시도 쉬지 않고 여러개의 카메라에 해당하는 녹화와

일을 동시에 열어 디스크에 쓰는 작업을 수행하여야 한다. 녹화 기능의 시스템 부하는 하드디스크 불량 등의 DVR 시스템 오류를 일으키는 주된 원인이 되고 있다.

검색 기능은 하드디스크에 녹화된 영상을 재생하여 사용자가 원하는 영상을 찾을 수 있도록 하는 기능이다. 동영상 재생기의 기본적인 기능인 재생, 멈춤, 한 프레임씩 이동, 무작위 탐색을 복수개의 카메라에 대해서 수행하며 제품에 따라서 역방향 재생이나 객체 검색(Museum Search) 등의 고급기능 까지 지원한다. 검색 기능은 녹화가 진행되고 있는 상태에서 동시에 수행되어야 하며 여러 개의 파일에 대해서 디스크 읽기를 수행하여 동시에 여러 개의 비디오/오디오 프레임을 복호화 하여 출력해야 한다. 이런 부하가 높은 상태에서 다중 채널을 실시간 영상처럼 자연스럽게 보여줘야 하기 때문에 구현이 가장 까다로운 기능이다.

그리하여 DVR 전용 파일 포맷을 설계하여 사용하는 것이 DVR의 녹화와 검색 기능을 소프트웨어로 향상시킬 수 있는 방법으로 제안되었다. 기존 범용 동영상 파일에서는 고려되지 않았던 DVR의 녹화와 검색 등의 기능을 고려하여 디스크 접근 횟수나 디스크에 유지되는 녹화 파일의 개수를 최소화 할 수 있는 구조로 설계가 가능하다. 또한 이벤트 저장과 같은 보안 감시 장비의 특수 기능을 파일 포맷에서 고려하여 DVR의 기능 확장과 성능의 향상을 꾀할 수 있기 때문이다.

2.2 AVI 파일 포맷

AVI 포맷은 멀티미디어 데이터에 대한 캡처, 편집, 재생 등의 목적으로 1992년에 마이크로소프트사에서 개발된 파일 포맷이다^[3-5]. RIFF(Resource Interchange File Format)라는 포맷에 기반을 두었으며 비디오, 오디오와 같은 멀티미디어 타입의 스트림을 하나 이상 포함한다^[6-7]. 동영상 감상 목적으로 PC에서는 가장 일반적으로 쓰이는 파일 포맷이며 DVR에서는 영상과 음성을 압축하여 저장하는 녹화용 파일 포맷으로 가장 널리 쓰이고 있지만 DVR에서 녹화용 포맷으로 적용하기에는 다음과 같은 문제점을 가지고 있다.

- 녹화 중인 파일 재생 불가능 - AVI 파일은 헤더에 데이터 사이즈와 프레임 카운트를 포함하고 있기 때문에 파일에 데이터를 다 쓰고 단을 때 헤더를 기록하는 구조로 되어 있다. 그러므로 현재까지 녹화된 데이터를 검색하기 위해서는 녹화 중인 파일을 닫고 새로 녹화를 시작해야 하는 불합리한 점이 있다. 검색 중에 현재 진행 중인 녹화 데이터를 반영하기 위해 녹화 재시작을 자주 해주는 것은 시스템에 큰 부하를 주므로 실제로 검색에서 녹화 데이터의 실시간 반영은 불가능하다고 할 수 있다.
- 비정상 종료된 파일 재생을 위한 복구 작업 필요 - AVI 파일을 단을 때 위에서 언급한 헤더 정보 외에도 인덱스 정보가 기록된다. AVI 파일에서는 'movi' 청크에 프레임 데이터들을 다 기록되고 난 후에 인덱스 리스트인 'idx1' 청크를 'movi' 청크 뒤에다 기록하도록 되어 있다^[8].

그러므로 녹화 도중에 에러 또는 정전 등의 상황에 의해 비정상 종료가 된 경우 헤더와 인덱스 정보를 복구해 주는 작업이 필요하다. 이 작업은 AVI 파일의 모든 프레임에 접근을 해야 하는 작업이기 때문에 많은 시간이 소요된다. 인덱스는 AVI 파일에서 생략이 가능하지만 검색에서는 반드시 필요하다.

- FPS에 의존적인 타임스탬프 계산 - AVI 포맷은 프레임 레이트에 따라 프레임 간격이 정해지도록 되어있어 앞에서부터 프레임 간격만큼 대기했다가 다음 프레임을 디스플레이하는 방식으로 재생하도록 되어있다. 즉 프레임의 타임스탬프는 "(60 / FPS x 프레임 번호 (초))"가 된다. 이런 방식은 프레임 개수가 프레임 레이트에 맞춰서 존재 한다는 가정 하에서만 가능하다. 현재 위치의 이전에 존재하는 프레임들의 개수가 현재 위치의 타임스탬프 계산에 쓰이기 때문이다. 그러므로 한 프레임이 빠지는 경우가 발생하면 그 위치에다가 더미 프레임(Dummy Frame)을 생성해서 기록해야 재생 시에 프레임들이 제시간에 출력될 수 있다.

AVI 포맷의 장점으로는 가장 널리 쓰이므로 지원하는 재생기가 풍부하다는 점을 들 수 있다. AVI 포맷의 단점으로 여겨지는 인덱스 리스트 또한 한편으로는 장점이 되기도 한다. 한 번 읽어 들인 인덱스 리스트를 이용하면 이후로는 디스크 접근 없이 모든 프레임의 위치를 알 수 있으므로 무작위 탐색시 지연시간을 최소화 할 수 있기 때문이다.

3. ESP 파일 포맷의 설계

3.1 ESP 파일 포맷의 특징

ESP 파일 포맷은 설계시 초점을 DVR의 녹화와 검색 기능 지원에 맞추었으며 2장 관련 연구에서 다른 AVI 포맷의 단점을 최대한 보완하여 상용 DVR에 적용할 수 있도록 설계되었다. ESP 파일 포맷의 특징 및 개선된 사항은 다음과 같다.

- 녹화시 순방향 진행 - ESP 파일을 생성한 이후부터 닫힐 때까지 파일 포인터가 순방향만으로 이동한다. 녹화가 진행되면서 파일포인터를 이전으로 이동시키는 작업이 생기지 않도록 설계되었다. 이렇게 설계한 가장 큰 목적은 순간 전원 차단 시 손실을 최소화하는 것이다. 메타데이터의 손실로 인해 해당 데이터가 기록되었더라도 사용할 수 없게 되는 일을 방지하여 녹화가 중단되기 이전까지 기록된 모든 데이터를 그대로 이용할 수 있도록 하였다.

- 고정 블록 사이즈 - ESP 파일 전체를 청크라는 동일한 크기의 블록으로 나누도록 하였다. 청크의 크기를 디스크의 블록 크기에 맞추고 ESP 파일을 읽고 쓸 때 청크 단위로 읽고 쓰게 하여 디스크 입출력 성능을 향상 시켰다. 청크의 맨 첫 바이트에서는 반드시 패킷이 시작되도록 하여 무작

위 탐색을 할 때 청크의 크기 단위로 파일 포인터를 이동하여 지정된 위치를 찾을 수 있도록 하였다.

- 프레임별 타임스탬프 - 프레임 레이트에 의존하여 타임스탬프를 구하는 방식 대신에 모든 프레임이 자신의 타임스탬프를 가지는 방식을 도입 하였다. 이러한 방식은 프레임 개수를 신경 쓸 필요가 없으므로 더미 프레임을 신경 쓸 필요도 전혀 없다. 무작위 탐색을 수행할 경우 목표 시간에 해당하는 파일 내에서의 위치를 예상하여 그 위치에서부터 목표 프레임을 찾아가는 과정이 필요하다. ESP 포맷에서는 어느 위치에서도 해당 위치의 프레임이 가지고 있는 시간 정보를 이용하여 목표 프레임까지 찾아갈 수 있다.

- 프레임별 키프레임(Key Frame) 정보 - 키프레임이란 I-Frame(Intra Frame)이라고도 불리며 앞/뒤 프레임과 상관없이 그 프레임 자체만으로 디코딩이 가능한 프레임이다^[9]. 무작위 탐색 성능을 최대한 향상시키기 위해 모든 비디오 프레임 패킷에 이전 키프레임 패킷의 위치를 가지도록 하였다. 키프레임 위치는 무작위 탐색 수행 시에 반드시 알아 내야 하는 정보로서 ESP 포맷에서는 목표 프레임 패킷만 찾으면 바로 키프레임 위치로 이동이 가능하게 하여 무작위 탐색을 용이하게 하였다.

- 이벤트 및 부가 정보를 스트림 형태로 저장 - DVR에서 감지한 이벤트 정보들이 비디오/오디오 데이터와 마찬가지로 스트림의 형태로 저장할 수 있다. 움직임 감지, 센서 입력 등의 이벤트 정보가 프레임 데이터와 같은 형태로 저장되며 맵핑된 비디오/오디오 스트림과 같은 파일 내에 존재할 수 있다. 이벤트 정보 외에도 DVR의 녹화 부가 정보를 스트림 형태로 사용자가 정의해서 같은 파일 내에 포함시키는 것이 가능하다.

- 하나의 파일에 다중 스트림 저장 - 하나의 파일에 비디오/오디오/이벤트 스트림을 각각 256개까지 녹화할 수 있다. DVR은 일반적으로 4개 이상의 비디오 채널과 하나 이상의 오디오 채널을 녹화한다. 이런 여러 채널의 스트림으로 구성된 녹화 데이터들을 하나의 파일로 저장할 수도 있고 서로 다른 파일로도 나누어서 저장할 수도 있다. 각 스트림들은 다른 스트림에 대한 참조 정보를 가지고 있어 오디오 데이터나 이벤트 데이터가 어떤 영상과 맵핑 되는지를 알 수 있다.

반면 ESP 파일 포맷은 AVI 파일 포맷에 비교하여 다음과 같은 취약점을 가지고 있다.

- 무작위 탐색 시간 - 프레임 위치를 알아내는데 메모리에 적재된 인덱스 리스트만 접근 하면 되는 AVI 파일 포맷과는 달리 ESP 파일 포맷은 직접 파일 내에서 위치를 이동해가면서 헤더의 내용을 읽어들이야 프레임의 위치를 찾을 수 있는 오버헤드를 가진다. 그러나 ESP 파일 포맷은

무작위 접근시 디스크 접근을 최소화 할 수 있도록 프레임 별로 타임스탬프와 키프레임 정보를 추가하였으며 실제로 사용자가 느끼는 무작위 탐색 시간에는 키프레임부터 목표 프레임 까지의 모든 프레임들을 디코딩 하는 시간이 포함 되므로 체감상으로는 성능의 차이를 느낄 수 없다고 할 수 있다.

- 파일 크기 - 프레임별 타임스탬프, 키프레임 정보와 청크 분할에 따른 추가적인 패킷 헤더로 인해 같은 데이터를 녹화한 AVI 파일과 비교하여 파일 크기가 약간 더 크다. 이 추가적인 정보들로 인해 녹화중 재생이 가능하며 또한 무작위 탐색 성능이 AVI 파일에 근접하게 되었으므로 감수할만하다 하겠다.

3.2 ESP 파일 포맷의 전체 구조

다음 (그림 1)는 ESP 파일의 전체 구조를 도식화 한 그림이다.

ESP 파일의 기본 구성단위는 패킷으로 (그림 1)의 ESP_FILE, ESP_STREAM, ESP_FRAME, ESP_FRAME_SEGMENT 는 패킷의 종류를 나타내는 것이다. AVI 포맷과 마찬가지로 파일의 앞부분에는 파일에 대한 헤더를 나타내는 ESP_FILE 패킷과 파일이 포함하고 있는 스트림에 대한 정보를 가지고 있는 스트림 헤더인 ESP_STREAM 패킷이 위치하고 그 뒷부분부터 프레임 데이터들이 존재한다. 프레임 데이터들은 ESP_FRAME패킷과 ESP_FRAME_SEGMENT 패킷들로 저장되어 지며 기본적으로 (그림 1)과 같이 스트림 종류는 상관하지 않고 시간 순서대로 번갈아 저장되며 같은 스트림끼리 연속된 형태로 저장될 수도 있다.

하나의 ESP 파일은 청크라는 단위로 나뉘어 지며 한 파일 내의 청크들은 모두 같은 사이즈를 가진다. 각 청크의

시작은 반드시 패킷의 첫번째 바이트 값으로 시작해야 한다. 그러므로 하나의 패킷이 청크의 경계에 걸릴 경우 두개의 패킷으로 나뉘어 지게 된다. (그림 1)의 빨간 선은 청크의 경계에 의해 패킷이 분할되는 것을 보여준다. 한 청크에 포함될 수 있는 프레임 패킷 개수에 대한 제한은 없다.

3.3 ESP 파일 포맷의 세부 구조

ESP 파일 포맷은 패킷으로 이루어지며 패킷들은 각 패킷들의 종류에 따라 필요한 정보들을 가지고 있다. 헤더와 프레임 모두 하나 이상의 패킷으로 구성된다. 패킷의 사이즈는 4바이트의 배수이어야 하며 65540바이트를 초과할 수 없다.

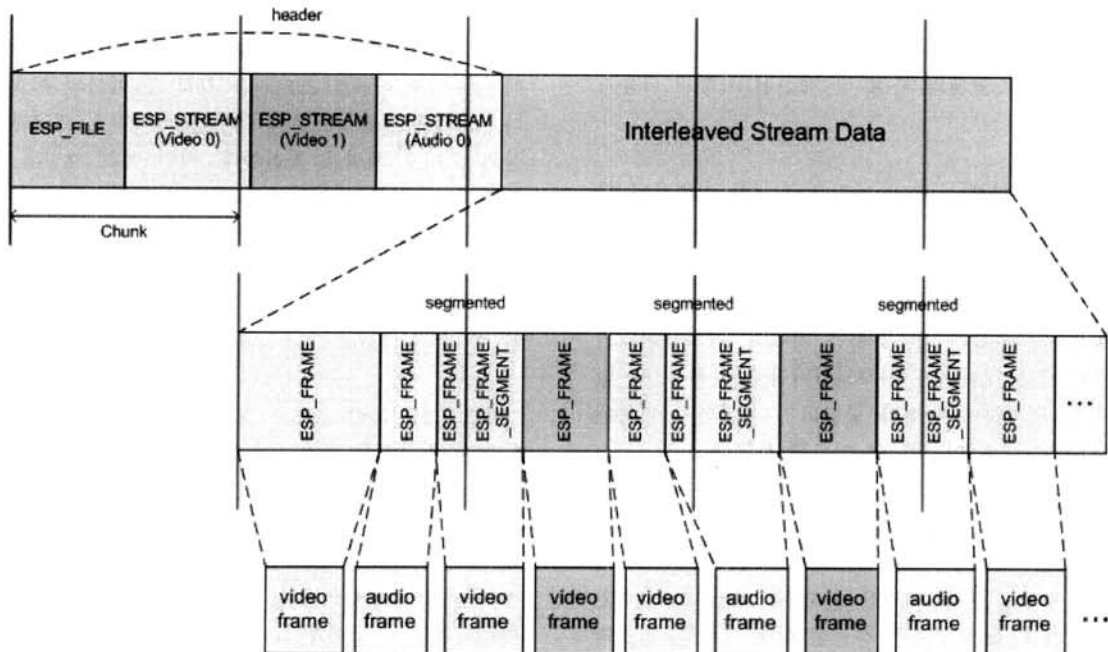
3.3.1 패킷의 구조

패킷의 구조는 다음 <표 1>과 같이 패킷의 속성을 나타내는 부분과 패킷의 데이터 부분으로 나뉘어져 있다.

- type : 패킷 종류를 나타냄.
- flag : 패킷 속성 또는 세부 종류. type에 따라서 다른 의미를 가진다.
- data_size : 패킷에 포함된 data필드의 크기. ESP_FRAME 패킷이 청크의 경계에 걸려 분할되는 경우 이 값은 (frame_size - 16)의 값과 다르다. 최대값은 65535이며

<표 1> 패킷의 구조

fields	size(bytes)
type	1
flag	1
data_size	2
data	data_size
padding	4 - (data_size % 4)



(그림 1) ESP 파일의 전체 구조

이 값보다 보다 큰 사이즈를 가지는 프레임은 패킷화 할 경우 65535를 초과하는 부분은 새로운 프레임 패킷에 해당한다.

- data : 실제 데이터. type값에 따라서 다른 종류의 데이터를 포함하게 된다.
- padding : 한 패킷을 4바이트 정렬(align)시키기 위해 덧붙여지는 패딩(Padding) 공간.

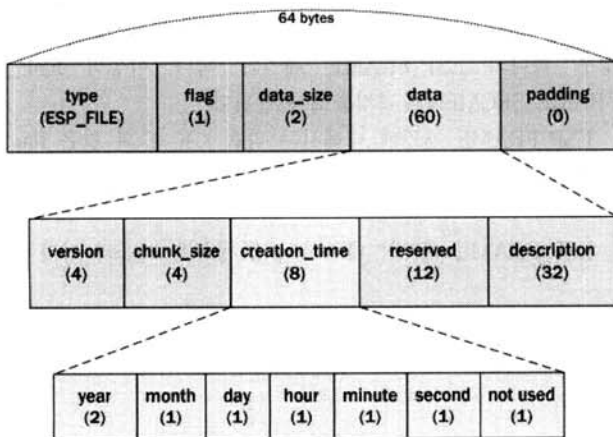
<표 2>는 현재 정의된 4가지의 패킷의 종류를 보여준다. 각 패킷의 종류별 기능과 구조에 대한 상세 명세는 다음 장부터 설명한다.

<표 2> 패킷의 종류

type	value	
ESP_FILE	0x03	파일 헤더 패킷
ESP_STREAM	0x02	스트림 헤더 패킷
ESP_FRAME	0x01	프레임 패킷
ESP_FRAME_SEGMENT	0x00	프레임 패킷에서 분할되어져 나온 패킷

3.3.2 ESP_FILE 패킷

파일의 맨 앞에 존재하여 ESP 파일에 대한 정보를 가지는 헤더 패킷이다. 이 패킷은 다음 (그림 2)와 같이 구성된다.



(그림 2) ESP_FILE 패킷의 구조

ESP_FILE 패킷의 데이터 구성 항목은 다음과 같다.

- version : 현재 파일의 ESP 파일 포맷 버전. 버전에 따라 달라지는 구조를 미리 알기 위해 맨 처음에 존재함.
- chunk_size : 이 파일의 녹화시 적용된 청크 사이즈. 최소값은 256.
- creation_time : 파일의 생성 일자.
- reserved : 추후 확장을 위해 예약된 공간
- description : 제목이나 이름 등의 파일에 대한 부가 정보

3.3.3 ESP_STREAM 패킷

스트림 헤더 정보를 가지고 있는 패킷이다. 이 패킷은

ESP_FILE 패킷 다음부터 존재하며 ESP 파일이 포함하고 있는 스트림 개수만큼 존재한다. 녹화시 파일에 포함될 모든 스트림의 정보를 각각의 ESP_STREAM 패킷으로 기록한다. ESP_STREAM 패킷을 다 기록하고 프레임 데이터를 기록하게 된다. ESP_STREAM 패킷의 구조는 다음 (그림 3)와 같다.

ESP_STREAM 패킷에 대해서는 다음 <표 3>과 같은 flag들이 정의되어 있다.

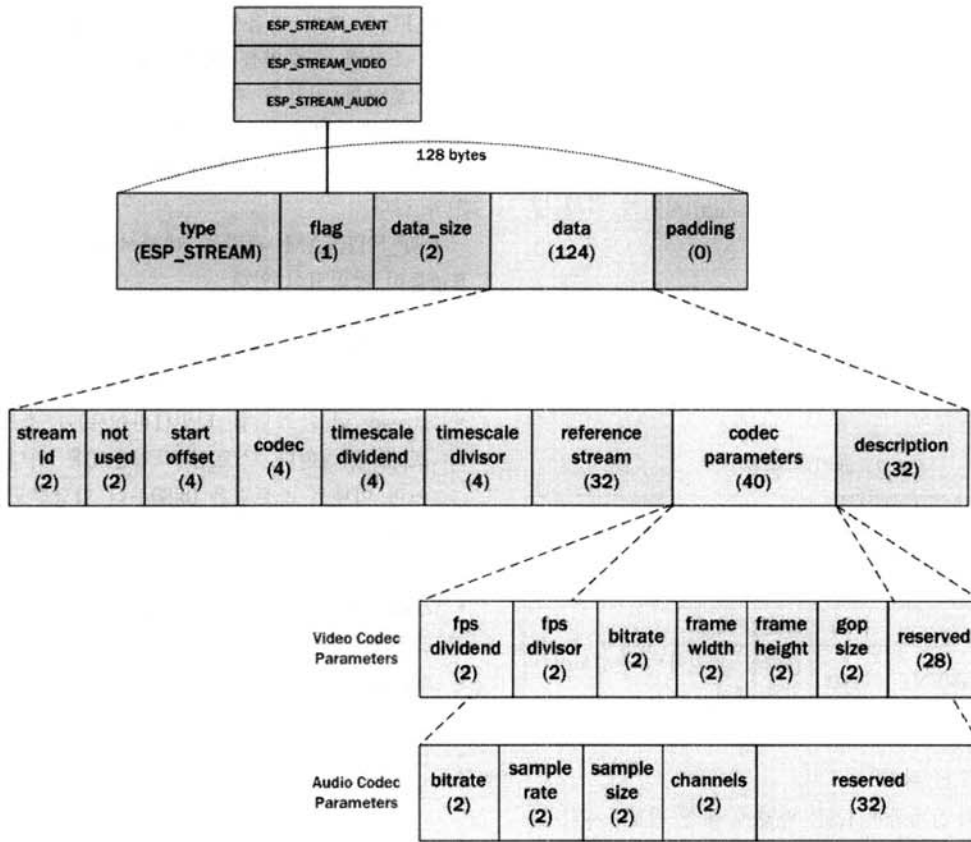
ESP_STREAM 패킷의 데이터 구성 항목은 다음과 같다.

- stream_id : 스트림 타입(1바이트)과 스트림 번호(1바이트)로 구성된다. 스트림 타입은 0은 비디오, 1은 오디오이며 이벤트 스트림은 0x80부터 할당받도록 되어있다.
- start_offset : 해당 스트림에 속하는 첫번째 프레임 패킷의 위치 값임.
- codec : 코덱을 나타내는 FourCC 값^[11], 비디오/오디오 스트림일 경우에만 쓰임.
- timescale_dividend : 프레임이 표시되는 시간을 구하기 위해 타임스탬프에 곱해지는 값. 단위는 마이크로초.
- timescale_divisor : 프레임에 표시되는 시간을 구하기 위해 타임스탬프를 나누는 값. 단위는 마이크로초. timescale_dividend와 timescale_divisor는 프레임의 타임스탬프 계산에 직접적으로 영향을 미치므로 정밀하게 지정할 수 있도록 분자와 분모로 나누었다.
- reference_stream : 이 스트림이 비디오 스트림인 경우 참조하는 스트림들을 2바이트 스트림 ID 16개가 붙어 있는 형태로 지정한다.
- codec_parameters : 디코더(Decoder) 생성에 필요한 정보들을 가진다. 스트림 종류 별로 구성 필드들이 다르며 현재 비디오 스트림과 오디오 스트림에 대해서 정의되어 있다. 스트림이 비디오일 때 사용되는 codec_parameters의 항목들은 다음과 같다.
 - fps_dividend, fps_divisor : 프레임 레이트를 지정하기 위한 값. 정수로 표현되지 않는 프레임 레이트를 지정할 수 있도록 하기 위해서 분자와 분모 값으로 나누어짐. 프레임 레이트는 $(FPS = fps_dividend / fps_divisor)$ 식에 의해 계산되어 진다.
 - bitrate : 데이터 레이트. 단위는 bps
 - frame_width : 비디오 프레임의 가로 크기
 - frame_height : 비디오 프레임의 세로 크기
 - gop_size : 키프레임 간격

오디오 프레임일 경우 사용되는 codec_parameters의 항

<표 3> ESP_STREAM 패킷에 사용되는 flag

flag	value	
ESP_STREAM_VIDEO	0x01	이 스트림은 비디오 스트림임.
ESP_STREAM_AUDIO	0x02	이 스트림은 오디오 스트림임.
ESP_STREAM_EVENT	0x04	이 스트림은 이벤트 스트림임.



(그림 3) ESP_STREAM 패키지의 구조

목들은 다음과 같다.

- bitrate : 데이터 레이트. 단위는 bps
- sample_rate : 샘플 레이트
- sample_size : 샘플 사이즈. 단위는 bit
- channels : 오디오 채널 개수
- description : 스트림에 대한 부가 정보

청크의 경계에 걸친 경우나 데이터 크기가 65535를 초과할 경우 하나의 ESP_FRAME 패키지와 하나 이상의 ESP_FRAME_SEGMENT 패키지들로 분할된다.

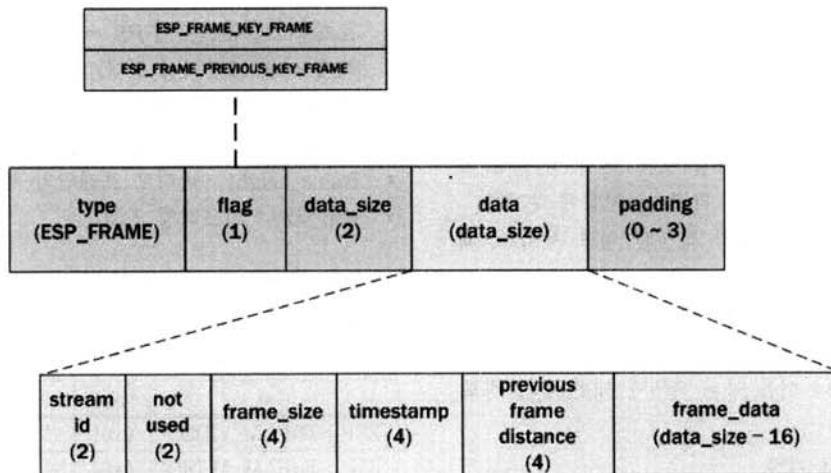
ESP_FRAME 패키지에 대해서는 다음 <표 4>와 같은 flag 들이 정의되어 있다.

ESP_FRAME 패키지의 데이터 구성 항목은 다음과 같다.

3.3.4 ESP_FRAME 패키지

스트림을 구성하는 프레임 을 나타낸다. 보통 하나의 ESP_FRAME 패키지는 하나의 프레임에 대응되지만 패키지는

- stream_id : 이 프레임이 속한 스트림의 ID
- frame_size : 전체 프레임 데이터의 사이즈. 프레임이 분할



(그림 4) ESP_FRAME 패키지의 구조

<표 4> ESP_FRAME 패킷에 사용되는 flag

flag	value	
ESP_FRAME_KEY_FRAME	0x01	이 프레임은 키프레임임.
ESP_FRAME_PREVIOUS_KEY_FRAME	0x02	비디오 프레임일 경우 <i>previous_frame_distance</i> 가 최근의 키프레임과의 거리를 지정함.

- 되는 경우 분할되기 이전 프레임의 전체 크기를 나타냄.
- **timestamp** : 프레임의 타임스탬프. 실제로 프레임이 보여지는 시간은 ($PTS = timestamp * timestamp_dividend / timestamp_divisor$) 식에 의해 계산되어 진다.
 - **previous_frame_distance** : 같은 스트림에 소속된 프레임 중 가장 최근 프레임 패킷 위치와의 거리를 나타낸다. 거리는 현재 프레임 패킷의 시작 위치와 이전 프레임 패킷의 시작 위치간의 거리로 한다.

비디오 프레임 패킷에 ESP_FRAME_PREVIOUS_KEY_FRAME 플래그가 설정되어 있는 경우 같은 스트림의 프레임 중 가장 최근 키프레임 패킷 위치와의 거리를 나타낸다.

- **frame_data** : 프레임 데이터. ESP_FRAME 패킷의 다른 필드와는 달리 이 필드가 체크의 경계에 걸리게 되는 경우에 체크 경계의 뒷부분에 해당하는 프레임 데이터는 ESP_FRAME_SEGMENT 패킷으로 할당되게 된다. frame_data 이외의 부분에서 체크의 경계에 걸리는 경우는 체크 경계의 앞부분에 해당하는 공간은 무시하고 새로운 체크에서 ESP_FRAME 패킷이 시작하도록 한다.

3.3.5 ESP_FRAME_SEGMENT 패킷

프레임이 분할되는 경우 분할된 부분을 포함하는 패킷이다. 프레임이 체크의 경계에 걸리는 경우 분할된 뒷 부분의 데이터 부터는 ESP_FRAME_SEGMENT 패킷에 포함되어 새로운 체크에 기록된다.

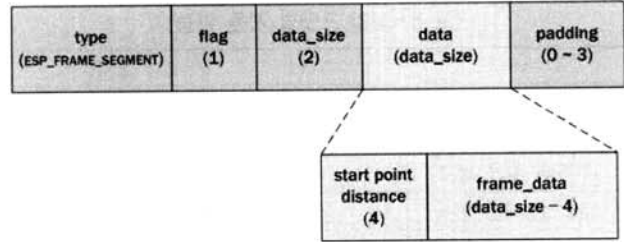
ESP_FRAME_SEGMENT 패킷의 데이터 구성 항목은 다음과 같다.

- **start_point_distance** : 원래 프레임에서 분할된 패킷들 중 첫번째 패킷인 ESP_FRAME 패킷 위치와의 거리를 나타낸다. 거리는 현재 프레임 패킷의 시작 위치와 ESP_FRAME 패킷의 시작 위치간의 거리로 한다.
- **frame_data** : 프레임 데이터. ESP_FRAME 패킷과 마찬가지로 이 영역이 경계에 걸린 경우에만 경계 뒷부분의 프레임 데이터가 또 다시 새로운 ESP_FRAME_SEGMENT 패킷으로 분할된다.

4 성능 평가

4.1 성능 평가 방법 및 제한

3장 연구 내용에서 언급했듯이 ESP 파일 포맷은 기존의



(그림 5) ESP_FRAME_SEGMENT 패킷의 구조

AVI 파일 포맷 장점은 유지시키면서 단점은 보완하고 거기에 DVR을 위한 추가 기능이 포함된 파일 포맷이다. 그러므로 기존 파일 포맷에서 지원하지 않는 기능들이 존재하게 되는데 이런 수치적인 성능 비교 평가가 불가능한 기능들에 대해서는 지원 여부를 비교하는 것으로 평가를 수행하였다. 성능 측정치에 대한 비교 평가는 파일 사이즈와 무작위 탐색 소요시간에 대해서 성능 평가를 하였다. 파일 사이즈는 DVR의 저장 가능한 기간을 결정짓는 중요한 요소이므로 성능 평가에 포함 시켰으며 무작위 탐색 소요시간은 DVR의 핵심 기능인 검색 기능의 성능 척도 중 하나이므로 평가에 포함시켰다.

성능 평가를 위해 네트워크를 통해 DVR로부터 MPEG4로 압축된 영상을 수신하여 AVI, ESP 두개의 파일로 녹화하는 프로그램을 작성하였다. 녹화 시간은 10분으로 하였으며 ESP 파일의 경우 체크 사이즈를 1, 2, 4, 8, 16Kbytes로 다르게 녹화하여 비교하였다.

무작위 탐색 소요시간은 무작위 탐색 수행시 키프레임 까지 찾아가는데 소요되는 시간까지만 측정하였다. 키프레임으로 이동 후 목표 지점까지 프레임을 읽으면서 복호화하는데 걸리는 시간은 파일 포맷과는 상관없는 부분이기 때문에 측정에서 제외하였다. 무작위 탐색 대상 파일은 녹화 테스트 프로그램을 이용하여 동일한 녹화 설정을 사용하여 동일한 영상을 녹화한 것을 사용하였다.

4.2 성능 평가 결과 및 분석

다음 <표 5>는 AVI, ESP 파일 포맷의 기능 지원 여부를 비교한 표이다.

다중 스트림 지원 여부는 하나의 녹화 파일에 DVR의 복수개의 채널을 녹화할 수 있는지를 말해 주는 것이다. DVR용 부가 정보는 DVR에서 쓰이는 특수한 정보(움직임 감지 같은 보안 관련 이벤트의 정보)가 녹화 파일에 포함될 수 있느냐 하는 것이다. 녹화 중 재생 여부는 현재 녹화 중인 파일에 대해서 재생이 가능한가 하는 것이다. 파일 전송중 재생 여부는 별도의 스트리밍 프로토콜 없이 녹화 파일을 다운로드 받으면서 다운로드 받은 만큼의 데이터에 대해서 재생이 가능한가를 나타내는 것이다.

<표 5>에서 알 수 있듯이 언급한 기능들에 대해서는 ESP 파일 포맷은 AVI 파일 포맷에 비해서 향상되었다는 것은 이미 연구 내용에서 언급하였다. 성능 측정이 가능한 비교 항목인 파일 사이즈와 무작위 탐색 성능에 대해서는 측정 결과를 가지고 다음 장에서 비교 평가를 하였다.

〈표 5〉 파일 포맷별 지원 기능

	AVI	ESP
다중 스트림 지원	파일별 256개 맵핑 정보 없음	파일별 65535개 맵핑 정보 존재
DVR용 부가 정보	X	O
녹화 중 재생 여부	X	O
파일 전송중 재생 여부	O	O

4.2.1 파일 사이즈

파일 포맷에서 프레임 데이터를 제외하고 파일 사이즈에 가장 영향을 미치는 부분은 각 프레임별 부가 정보의 크기와 인덱스 정보의 크기이다. 파일 포맷별로 추가되는 정보의 크기는 〈표 6〉과 같다. 파일과 스트림 헤더 정보의 크기는 다른 부분에 비하면 매우 작고 변화폭도 거의 없으므로 무시하였다.

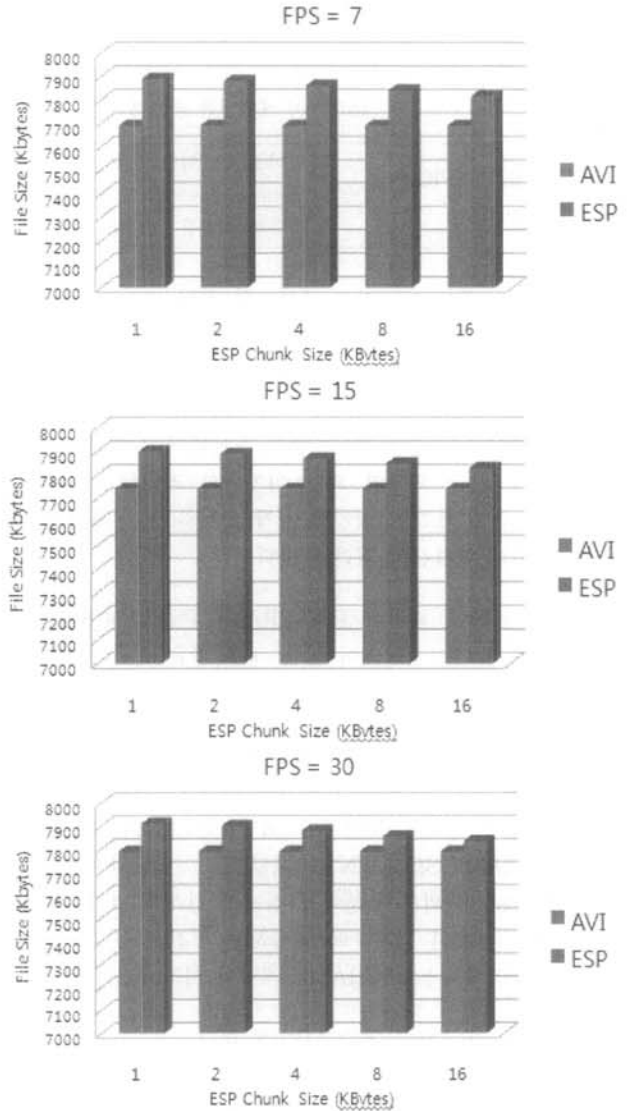
동일한 영상에 대해서 녹화를 수행한다고 가정하면 AVI 파일의 파일 사이즈는 실제로 측정을 해보지 않아도 계산만으로 비교 평가가 가능하다. ESP 파일의 프레임 패킷간의 패딩은 프레임을 녹화하려는 지점부터 다음 청크까지 남은 공간이 ESP_FRAME 패킷이나 ESP_FRAME_SEGMENT 패킷의 프레임 관련 정보를 기록하기에 모자라는 경우 그 청크의 남은 공간이 무시되는데, 이 남은 공간을 가리키는 것이다. 이 공간의 크기를 미리 알기 위해서는 녹화되는 모든 프레임의 사이즈를 알아야만 하므로 결국 녹화를 실제로 수행해봐야 파일 사이즈를 알 수 있다.

〈표 6〉의 식에 의하면 AVI 파일과 ESP 파일의 사이즈를 결정하는 가장 큰 요소는 프레임 개수와 청크 개수임을 알 수 있다. 파일 사이즈 성능 평가시 프레임 개수를 결정짓는 요소인 FPS와 청크 개수를 결정하는 청크 사이즈를 변화시켜가며 녹화 테스트를 수행하였다. 다음 (그림 6)이 녹화 테스트를 통해 측정된 파일 사이즈를 나타낸 것이다. 고정 비트레이트 100Kbps, 프레임 간격 50의 녹화 설정으로 동일한 영상을 10분 동안 녹화하였다.

성능 측정 결과 ESP 파일의 사이즈가 AVI 파일의 사이즈보다 약 1~2%정도 크다는 것을 알 수 있다. 그러나 이 차이는 AVI 파일에서 문제점으로 지적되었던 FPS에 의존적인 시간계산 문제 해결을 위한 타임스탬프와 녹화 중 재생이 가능하면서도 AVI 파일에 근접한 무작위 탐색 성능을 위한 키프레임 정보가 포함되어 있는 크기이며, 접근성 향상과 디스크 입출력 속도 향상을 위한 청크의 개념을 도입함으로써 생긴 오버헤드이므로 충분히 감수할만한 수치이다.

〈표 6〉 추가 파일 사이즈 계산식

파일 포맷	추가되는 정보 (Bytes)
AVI	프레임 정보 = 총 프레임 개수 * 8 인덱스 정보 = 총 프레임 개수 * 16
ESP	프레임 정보 = 총 프레임 개수 * 20 청크 경계에서 분할된 패킷 헤더 ≤ 총 청크 개수 * 4 프레임 패킷간의 패딩 = ?



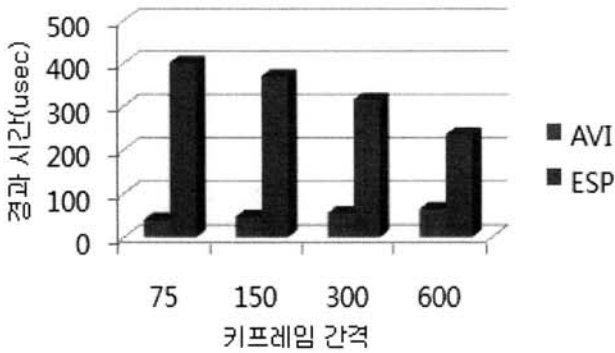
(그림 6) 파일 사이즈 측정 결과

4.2.2 무작위 탐색 소요시간

다음 (그림 7)은 AVI, ESP 파일들에 대하여 무작위 탐색을 각각 10000번씩 수행하여 측정된 시간의 평균을 그래프로 나타낸 것이다. 각 파일은 고정 비트레이트 100Kbps, 7 FPS의 녹화 설정으로 녹화되었다.

AVI 파일은 메모리에 존재하는 인덱스 리스트에서 파일 내 접근할 위치를 알오므로 파일의 녹화 설정이나 탐색 목표 위치에 영향을 받지 않고 일정하게 아주 빠른 속도로 탐색이 수행된다.

ESP 파일의 경우 인덱스 리스트가 존재하지 않기 때문에 사용자가 원하는 지점을 파일 내 임의의 위치에서부터 찾아가야 한다. 이번 성능 측정시 사용한 방법은 목표 지점까지 경과한 시간을 입력받아 총 녹화 시간과의 비율을 파일 사이즈에 곱하여 파일 내 탐색 시작 위치를 계산하도록 하였다. 키프레임 간격이 커질수록 일정한 간격으로 프레임이 존재하게 되므로 그만큼 탐색 시작 위치가 목표 위치에 인접할 확률이 높아지게 되어 목표 지점에 해당하는 프레임은



(그림 7) 무작위 탐색 소요시간 측정 결과

찾는데 소요되는 시간이 줄어든다. ESP 파일은 모든 비디오 프레임 패킷에 키프레임 위치가 저장되어 있어 목표 지점의 프레임을 찾으면 바로 키프레임 위치로 이동할 수 있으므로 키프레임 간격이 늘어나면 탐색 소요시간이 줄어들음을 알 수 있다.

프레임별 키프레임 정보와 체크 단위 접근을 활용하여 최대한 무작위 탐색 성능을 향상시켰으나 AVI 파일에 비해서는 떨어진다. 그러나 모든 프레임의 인덱스를 따로 유지하는 것은 이상 AVI 파일의 무작위 탐색 성능을 따라 잡을 수는 없다. 오히려 인덱스를 따로 유지하지 않음으로서 녹화중인 파일과 비정상 종료된 파일에 대한 재생이 가능해졌으며 무작위 탐색 성능도 체감상으로는 차이를 느끼지 못하므로 DVR의 검색 기능에 있어서는 AVI 파일보다 우월하다고 평가할 수 있다.

5. 결론 및 향후 연구 과제

DVR의 기능 중에서 녹화와 검색 기능은 가장 기본적인 면서 핵심적인 기능이라고 할 수 있으며 프로세서와 디스크 자원을 가장 많이 사용하는 기능이다. 게다가 대부분의 DVR에서는 AVI 파일 포맷을 녹화 파일 포맷으로 사용하고 있는데 AVI 파일 포맷은 범용 멀티미디어 파일 포맷이라서 DVR에서 녹화용 파일 포맷으로 쓰이기에는 여러 가지 취약점을 가지고 있다.

본 논문에서는 DVR의 녹화와 검색 기능에 초점을 뒀서 AVI 파일 포맷의 문제점을 개선하고 장점만을 적용시킨 ESP 파일 포맷을 설계하였다. ESP 파일 포맷은 체크 단위의 접근 구조를 사용하여 AVI 포맷의 가장 큰 문제점이었던 녹화 중 재생을 가능하게 하였다. 그리고 DVR의 녹화와 검색 기능을 고려하여 부가 정보를 파일에 같이 추가할 수 있도록 하였다.

향후 과제로는 사용자 PC에서도 ESP 녹화 파일을 재생해 볼 수 있는 재생기 개발이 이루어져야 한다. 그리고 성능 평가에서 AVI 포맷 보다 다소 낮은 성능을 보여주었던 무작위 탐색 성능에 대해서 보완이 이루어져야 한다.

참 고 문 헌

- [1] "Sentry24", <http://www.emstone.com>
- [2] "보안용 DVR 산업 동향 및 전망", 산업자원부 디지털전자사업과, 2004.
- [3] "OpenDML AVI File Format Extensions," <http://jmcgowan.com>
- [4] "Microsoft Video for Windows Development Kit, Programmer's Guide", Microsoft, Ch. 4, 1993.
- [5] "DV Data in the AVI File Format Specification, Version 1.01," Microsoft, 1997, <http://www.microsoft.com/whdc/archive/dvavi.msp>
- [6] "Encyclopedia of Graphics File Formats," <http://www.oreilly.com/catalog/gff>
- [7] "New Multimedia Data Types and Data Techniques Standards Update," Microsoft, pp.5-13, 1994.
- [8] Roger Zimmermann, "Streaming of DivX AVI Movies," Department of Computer Science, University of Southern California, pp.1-2, 2003.
- [9] "ISO/IEC 11172-2: Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s -- Part 2: Video."
- [10] 문동규, 이좌형, 김병길, 김윤, 정인범, "고화질 WMV 파일 포맷 분석을 통한 Key-Frame 추출 알고리즘 설계 및 구현", 한국정보과학회 2004년도 가을 학술발표논문집, 제31권 제2호 (III), Oct. 2004.
- [11] "Video Codecs and Pixel Formats," <http://www.fourcc.org>
- [12] 조용주, 홍진우, 김진웅, "MPEG-21 기반 멀티미디어 파일 포맷", 한국방송공학회, 방송공학회논문지, 제8권 제4호, pp.392-398, Dec. 2003.



박재경

e-mail : c992699@gmail.com

2004년 숭실대학교 컴퓨터학부 졸업(학사)

2006년 숭실대학교 대학원 컴퓨터공학과 (공학석사)

2002년~2008년 (주)엠스톤

2008년~현 재 (주)마크애니 주임연구원

관심분야 : DVR, DRM



양 승 민

e-mail : smyang@ssu.ac.kr

1978년 서울대학교 전자공학과(학사)

1983년 Univ. of South Florida 전산학과
(공학석사)

1986년 Univ. of South Florida 전산학과
(공학박사)

1978년~1981년 (주)삼성전자 연구원

1987년 Univ. of South Florida 조교수

1988년~1993년 Univ. of Texas at Arlington 조교수

1996년~1998년 국회도서관 정보처리국장 파견

1993년~현 재 숭실대학교 컴퓨터학부 교수

2001년~현 재 (주)엠스톤 대표이사

관심분야 : 실시간 내장 시스템, 운영체제