

# 비디오 데이터베이스에서 지원 사용자 수를 증가시키기 위한 저장시스템 (Storage System for increasing the number of Users supported in Video Database)

김 근 형\*  
(Keun-Hyung Kim)

## 요 약

비디오 데이터베이스는 비디오 정보를 저장하고 있다가 사용자가 요구한 비디오 정보를 검색해 주는 기능을 갖는다. 따라서, 비디오 데이터베이스의 목적은 비디오정보를 저장하고 검색하기 위한 편리하고도 효율적인 환경을 제공해야 할 뿐만 아니라 보다 많은 사용자들에게 보다 신속하게 요구한 정보들을 검색해 줄 수 있어야 한다.

본 논문에서는 비디오 데이터베이스에서 빠른 응답을 유지하면서 지원할 수 있는 사용자들의 수를 증가시킬 수 있는 저장시스템을 제안한다. 다중 디스크환경하의 저장시스템에서 다중 비디오 스트림들을 읽을 때, 각 디스크헤드들의 이동을 비동기화 시킴으로써 빠른 응답을 유지할 수 있고, 한쪽 방향으로만 이동하게 함으로써 디스크 대역폭을 절약할 수 있다. 또한, 더블버퍼링을 기반으로 비디오 스트림의 시간제약 특성을 만족시키고 더블버퍼링의 단점을 최소화할 수 있는 기법을 제안한다. 새로운 저장시스템에 대한 타당성은 이론적으로 분석 및 증명되고 시뮬레이션을 통하여 평가된다.

## ABSTRACT

Video server which stores video streams should serve more users with the requested streams rapidly, satisfying their time constraints. For this objective, the storage system of the video server needs large disk bandwidth and buffer capacity. But, because given disk bandwidth and buffer capacity are limited, technologies that utilize effectively the given disk bandwidth and buffer capacity without waste might be needed.

In this paper, we propose the policies that not only reduce the startup latency but also support more users by utilizing the disk bandwidth and buffer efficiently, which also considering time constraint characteristics of video stream. In order to reduce the startup latency of the service requests, it induces the effect of random access of disk head by moving asynchronously disk heads in multiple disk environments. Also, there are two technologies to support more users. First, the proposed policy reduces the seek latency by moving the disk head sequentially so that it saves the disk bandwidth. Second, it determines the segment size through considering both given buffer capacity and given disk bandwidth by heuristic approach in order to overcome the constraints of the number of streams supported due to limitation of buffer capacity. The segment size makes it utilize given buffer optimally with given disk bandwidth. The correctness of the proposed policies is not only analyzed theoretically but also evaluated through simulations.

---

\* 정회원 : 제주산업정보대학 컴퓨터정보계열 조교수

논문접수 : 2001. 3. 10.

심사완료 : 2001. 3. 31.

## 1. 서론

컴퓨터와 통신 기술의 발전으로 대용량의 멀티미디어 데이터를 저장, 접근, 처리하는 것이 가능하게 되었고 현실 세계에서 이를 응용하기 위한 활동들이 활발히 진행되고 있다. 특히, 비디오 데이터는 교육, 오락등과 같은 분야에서 인터넷 기술등과 연동되었을 때 상업적 잠재력이 크므로 비디오 데이터 관련 연구들이 많이 이루어지고 있다. 비디오 데이터의 양은 현재 전 세계적으로 600만 시간에 달하고 있고 매년 10% 이상씩 증가하는 추세이므로 이러한 많은 양의 비디오 데이터를 체계적으로 관리하기 위한 비디오 데이터베이스의 중요성은 날로 더해가고 있는 실정이다[1].

비디오데이터베이스시스템(Video DataBase System, VDBS)은 비디오 정보를 저장하고 있다가 사용자가 요구한 비디오 정보를 검색해 주는 기능을 갖는다. 따라서, VDBS의 주요 목적은 비디오 데이터들을 쉽게 저장할 수 있게 하는 것과 저장된 비디오 데이터들 중에서 원하는 부분을 신속하게 검색할 수 있게 하는 것, 보다 많은 사용자들을 서비스할 수 있게 하는 것이다[1].

VDBS와 관련된 주요 연구들은 원시 비디오데이터(raw video data)에 대한 메타데이터(meta data)들을 효율적으로 생성하여 요구된 비디오 정보를 보다 정확하고 신속하게 검출할 수 있는 방안들에 집중되어 왔다[2, 3]. 그러나, 요구한 정보의 신속한 검출은 효율적인 메타데이터의 구축과 더불어 실제로 원시 비디오데이터에 접근하여 요구한 부분을 읽어주는 하부저장시스템의 역할도 중요하다. VDBS의 하부 저장시스템의 관점에서 볼 때 원하는 비디오부분을 신속하게 읽기 위하여 디스크헤드들의 임의적 이동을 허용해야 한다[4]. 특히, 하부 저장시스템의 성능은 VDBS가 동시에 지원할 수 있는 사용자들의 수에 영향을 끼친다. 향후 통신 기술의 발전으로 네트워크 대역폭이 충분할 경우 동시적으로 지원할 수 있는 사용자들의 수는 VDBS에 대한 중요한 성능평가 요인이 될 수 있다. VDBS의 하부 저장시스템이 동시적으로 보다 많은 사용자들을 지원하기 위하여 한정된 디스크대역폭을 절약할 수 있어야 한다. 디스크대역폭의 절약방안은 디스크헤드의 순차적 이동을 통하여 스트림 검색사이의 탐색지연시간을 단축시켜

야 한다[5].

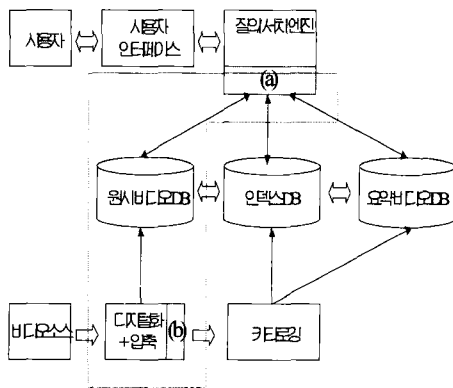
본 논문에서는 빠른 응답을 유지하면서 보다 많은 사용자들을 지원할 수 있는 VDBS의 하부 저장시스템을 제안한다. 다중 디스크환경하의 하부 저장시스템에서 각 디스크의 디스크헤드의 이동을 비동기화시킴으로써 임의 이동의 효과를 유도하고 또한, 한쪽방향으로만 이동할 수 있게 함으로써 순차이동의 효과를 유도한다. 이때, 비디오 스트림의 시간제약 특성을 만족시키기 위하여 더블버퍼링을 사용하고 더블버퍼링의 단점을 최소화할 수 있는 방안도 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 비디오 데이터베이스에 대해서 설명하고 3장에서는 비디오 데이터베이스의 하부 저장시스템과 관련한 여러 가지 정책들을 고찰하며 기존의 관련 연구들 및 문제점을 분석한다. 4장에서는 VDBS의 성능을 향상시킬 수 있는 하부 저장시스템의 새로운 검색정책을 제안하고 이에 대한 타당성을 검증한다. 5장에서는 모의 실험을 통하여 제안된 정책을 평가하고 분석하며, 마지막으로 6장에서 결론을 맺는다.

## 2. 비디오데이터베이스

VDBS의 주요 목적은 비디오 정보를 저장하고 검색하기 위한 편리하고도 효율적인 환경을 제공하는 것이다. 좀 더 구체적으로 말하면, 순차적인 비디오 데이터의 특정 부분에 대한 임의접근을 제공하기 위하여 비디오 스트림을 의미있는 데이터단위로 분할하여 각 데이터단위들을 인덱싱(indexing)하고 보다 쉬운 브라우징(browsing)과 검색(retrieval)을 할 수 있게끔 그 인덱스들을 표현하는 것이다[1]. 그리하여 VDBS는 원시 비디오데이터들에 대한 인덱스들의 데이터베이스라고 할 수 있다. 이러한 것들과 더불어 VDBS는 전통적인 DBMS(DataBase Management System)의 요구사항들 즉, 데이터 공유성, 보안성, 데이터 독립성, 중복배제성, 일치성, 무결성 등의 부분들도 고려하여야 한다. [그림 1]은 전형적인 VDBS의 블록다이어그램[2]을 나타내고 있다. [그림 1]에서 VDBS의 각 부분의 개략적인 기능명세는 다음과 같다.

- ① 사용자 인터페이스 : 편리하고도 시각적인 방법으로 데이터베이스와 상호작용하면서(interactively) 질의(querying) 및 브라우즈(browse) 할 수 있게 하고 선택된 비디오 클립(video clip)들을 볼 수 있게 한다.
- ② 질의/서치 엔진(query/search engine) : 사용자가 제공한 파라미터들에 따라 데이터베이스를 서치한다.
- ③ 원시 비디오DB : 디지털화되고 압축된 비디오 데이터들의 저장소
- ④ 시각적인 요약 : 간결하면서도 계층적인 방법으로 비디오데이터들을 표현할 수 있는 요약된 비디오정보
- ⑤ 인덱스DB: 비디오 데이터단위들에 대한 포인터들
- ⑥ 디지털화+압축 : 아날로그 비디오데이터를 압축된 디지털 포맷으로 변환하기 위하여 필요한 하드웨어와 소프트웨어
- ⑦ 카타로그킹(cataloguing) : 원시 비디오데이터로부터 의미있는 데이터단위를 추출하고 대응하는 인덱스들을 생성



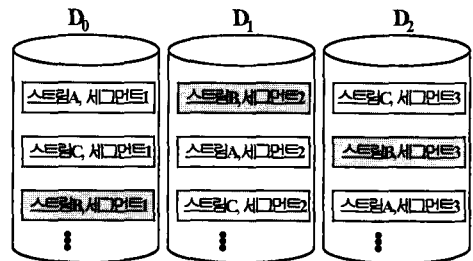
[그림 1] VDBS의 블록다이어그램  
[Fig.1] Block diagram of VDBS

VDBS의 하부 저장시스템은 비디오데이터들을 저장장치에 저장하고 사용자요구에 따라 원시 비디오 데이터의 어떤 부분을 읽어주는 기능을 갖는다고 할 수 있다. 이러한 관점에서 볼 때, [그림 1]에서 하부

저장시스템을 구성하는 부분은 「원시비디오DB」와 「디지털화+압축」 그리고 「질의/서치 엔진에서 원시비디오 DB에 접근하여 요구 비디오 데이터를 읽는 부분」이라고 할 수 있다. 「원시비디오DB」를 위한 저장공간이 다중디스크 환경이라고 가정할 때 VDBS에서 하부 저장시스템의 주요 기능은 원시 비디오 데이터들 즉, 비디오 스트림들을 다중 디스크상에 적절하게 배치하고 요구된 비디오 데이터들을 효율적으로 읽는 것이라고 할 수 있다. 하부저장시스템 기능은 그림1에서 점선으로 둘러싸인 영역이라고 할 수 있다. VDBS에 도착한 서비스요구에 대하여 인덱스DB를 탐색하는 과정이 선행되고 여기에서의 결과를 바탕으로 하부 저장시스템은 원시 비디오DB를 검색하여 최종적인 결과를 도출한다. 다음 절에서 VDBS의 하부 저장시스템과 관련하여 기존에 연구된 주요 결과들을 고찰하고 분석해 본다.

### 3. 하부저장시스템을 위한 기존 연구들의 고찰

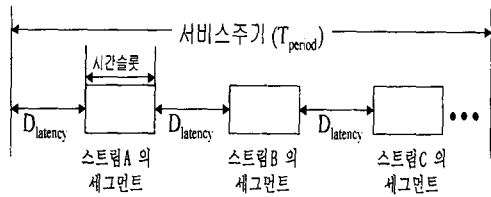
저장 시스템이 비디오 스트림에 물리적으로 접근하는 단위를 세그먼트라고 부른다. 즉, 스트림은 세그먼트들로 분할되어 다중 디스크들 상에 분산 배치되고 스트림 검색 시 세그먼트 단위로 읽어들인다. [그림 2]는 비디오 스트림들 A, B, C가 각각 세그먼트들로 분할되어 디스크 D<sub>0</sub>, D<sub>1</sub>, D<sub>2</sub> 상에 배치된 상황을 나타내고 있다.



[그림 2] 비디오스트림의 분산배치 모델  
[Fig.2] Placement model of video streams

비디오 스트림을 검색하여 재생할 때는(서비스할 때) 그 스트림의 연속적인 세그먼트들을 세그먼트 단

위로 하나씩 읽고 재생하는 과정을 반복한다. 서비스 주기는 하나의 세그먼트가 재생되는 시간이다. 다중 스트림들을 서비스하기 위하여 각 스트림 당 하나의 세그먼트씩을 서비스주기 내에 모두 읽어들이어야 한다. [그림 3]에서 나타내는 바와 같이 서비스주기 ( $T_{period}$ ) 동안 사용될 데이터를 자기에게 할당된 시간 슬롯(time slot) 내에 모두 읽어들이어야 한다. 그리하여, 현재 서비스를 받고 있는 스트림들은 종료되기까지 특정한 서비스주기를 두고 반복적으로 처리되어야 한다. [그림 3]은 단일 디스크 환경을 가정한 검색 서비스 모델이고  $D_{latency}$ 는 디스크 전 영역을 대상으로 어떤 스트림의 세그먼트를 읽기 위한 탐색지연 시간이다.



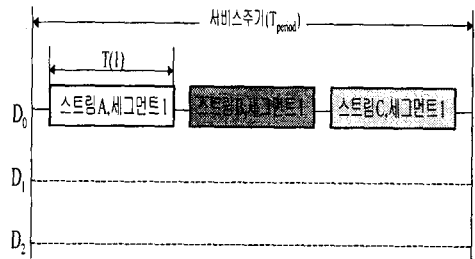
[그림 3] 단일 디스크 환경에서의 다중 스트림 서비스모델

[Fig.3] Multiple streams service model in single disk environment

서비스 주기는 세그먼트가 소모되는 시간보다 작아야 한다. 서비스 주기를 길게 하려면 세그먼트 크기를 크게 해야 한다. 서비스 주기가 세그먼트 소모 시간 보다 길어지면 스트림 재생 시 깜빡거림(hiccup) 현상이 일어난다. 따라서, 서비스 주기는 세그먼트 크기에 의하여 결정되고 식 (1)을 만족해야 한다. 이후부터 주로 사용될 기호들 중에서  $T_{period}$ 는 서비스주기,  $S$ 는 세그먼트 크기,  $R_{med}$ 는 스트림소모율,  $R_{disk}$ 는 디스크 전송률,  $D_{latency}$ 는 디스크 내에서의 탐색지연시간,  $m$ 은 서비스주기동안 검색될 세그먼트 개수를 의미한다.

$$m * \left( \frac{S}{R_{disk}} + D_{latency} \right) \leq T_{period} \leq \frac{S}{R_{med}} \quad (1)$$

[그림 4]는 다중 디스크 환경에서 비디오 스트림들이 분산배치되었을 때의 스트림 검색서비스 모델이다. 어떤 서비스주기의 임의의 시간슬롯에서 다중 디스크 환경의 각 디스크는 상이한 스트림들의 세그먼트를 검색한다. [그림 4]에서 시간슬롯  $T(i)$  동안 디스크  $D_0$ 에서 스트림A의 세그먼트1을 읽는동안에 디스크  $D_1$ 과  $D_2$ 에서는 다른 스트림들의 세그먼트들을 읽는다.



[그림 4] 다중 디스크환경에서 다중 스트림 서비스 모델  
[Fig.4] Multiple streams service model in multiple disk environment

각 서비스 주기동안 각 스트림 당 하나의 세그먼트들이 버퍼로 읽혀지고 읽혀진 세그먼트는 재생된다. 단일 버퍼링(single buffering)을 가정하면 세그먼트가 버퍼로 읽혀지는 동안 동일버퍼에서 데이터 소모도 동시에 일어난다[6]. 따라서, 버퍼에 데이터가 읽혀지는 속도는  $R_{disk} - R_{med}$ 가 된다.  $T(i)$ 가 서비스주기( $T_{period}$ )동안 스트림B에 할당된 시간 슬롯이라고 한다면 스트림B를 위하여 요구되는 버퍼량은  $(R_{disk} - R_{med}) * T(i)$ 이고 이는 세그먼트의 크기 $S$ 와 같다.

현재  $m$ 개의 스트림들이 서비스중일 때 필요로 되는 버퍼 용량은 식 (2)처럼 표현될 수 있다[6].

$$B = \sum_{i=1}^m [(R_{disk} - R_{med}) * T(i)] = \sum_{i=1}^m [R_{med} * (T_{period} - T(i))] = m * S \quad (2)$$

식 (2)에서 볼 수 있는 것처럼 단일 버퍼링은 버퍼 용량을 적게 요구하는 대신 버퍼로 읽어들이는 데이터량이 스트림소모율 만큼 감소되고 있음을 알 수 있다. 또한, 단일 버퍼링은 각 서비스주기에서 스

트림의 검색순서를 유지하여야 스타베이션(starvation) 현상이 발생하지 않는다[6].

더블 버퍼링(double buffering)은 세그먼트가 버퍼로 읽혀지는 동안 다른 버퍼에서 데이터 소모가 일어난다. 따라서, 버퍼에 데이터가 읽혀지는 속도는 디스크 전송속도 $R_{disk}$ 를 유지할 수 있다.  $T(i)$ 가 서비스주기동안 스트림 $B$ 에 할당된 시간 슬롯이라고 한다면 스트림 $B$ 를 위하여 요구되는 버퍼량은  $2 * R_{disk} * T(i)$  이고 이것은 세그먼트 크기 $S$ 의 2배이다. 현재  $m$ 개의 스트림들이 서비스중일 때 필요로 되는 버퍼 용량은 식 (3)처럼 표현될 수 있다.

$$B = \sum_{i=1}^m [2 * R_{disk} * T(i)] = \sum_{i=1}^m [2 * R_{med} * T_{period}] = 2 * m * S \quad (3)$$

식 (3)에서 볼 수 있는 것처럼 더블 버퍼링은 버퍼 용량을 많이 요구하는 대신 세그먼트를 버퍼로 읽어들이는 속도는 디스크 전송 속도를 그대로 유지할 수 있음을 알 수 있다. 또한, 더블 버퍼링은 각 서비스주기에서 스트림의 검색순서를 유지하지 않아도 된다.

다중 디스크 환경에서 비디오 스트림의 저장 및 검색정책과 관련하여 기존에 연구된 주요 결과들은 RR(Round Robin), VSP(Virtual Sequential Pumping) 등이 있다. 각 정책들은 비디오 스트림  $V_i$ 를 세그먼트들  $V_i^0, V_i^1, V_i^2, \dots$  으로 분할하여 각 세그먼트들을  $X$ 개의 디스크들  $D_0, D_1, D_2, \dots, D_{X-1}$  상에 분산 배치한다.

RR정책에서 임의의 비디오 세그먼트  $V_i^j$ 가  $X$ 개의 디스크들  $D_0, D_1, D_2, \dots, D_{X-1}$  상에 배치될 때 디스크  $D_{(i+j) \bmod X}$ 의 임의의 위치에 배치된다[7]. RR정책은 각 디스크에서 디스크헤드의 임의이동에 기반하여 다중 세그먼트 서비스를 하므로 세그먼트 검색사이의 탐색 시간 등으로 디스크 대역폭을 낭비한다. 그러나, 스트림 요구에 대한 응답은 즉각적으로 이루어진다.

VSP 정책은 각 디스크 영역을 동일한 수의 블록 데이터를 갖고 동일한 디스크 전송률을 갖는 서브영역 즉, 존(zone) 들로 분할하고 비디오 스트림의 세그먼트들을 특정 디스크의 특정 존에 배치한다. 디스크 배열에서  $X$ 개의 디스크  $D_0, D_1, D_2, \dots, D_{X-1}$ 이 있고 각 디스크는  $Y(=X+1)$ 개의 존들  $Z_0, Z_1, Z_2, \dots,$

$Z_{Y-1}$ 로 나누어져 있다고 할 때 임의의 비디오 세그먼트  $V_i^j$ 의 전역 세그먼트 번호가  $g$ 라면 이 세그먼트는 디스크  $D_{g \bmod X}$ 의 존  $Z_{g \bmod Y}$ 에 배치된다. 전역세그먼트 번호는 세그먼트들에 부여된 일련번호인데 다른 스트림들의 세그먼트들도 고려하여 순서대로 부여된 세그먼트 번호이다. 예를 들어 스트림  $V_0$ 와  $V_1$ 의 세그먼트들에 대한 전역 세그먼트 번호를 구할 때 스트림  $V_0$ 가 6개의 세그먼트들  $V_0^0, V_0^1, V_0^2, V_0^3, V_0^4, V_0^5$ 로 이루어질 경우 스트림  $V_1$ 의 첫 번째 세그먼트  $V_1^0$ 의 전역세그먼트번호는 7이 된다. 데이터 검색 시 다중 디스크 환경의 모든 디스크들의 디스크헤드들은 서비스주기마다 존 단위로 동기화(synchronization)를 유지하면서 안쪽 존부터 바깥쪽 존으로 순차적 이동을 하면서 요구 세그먼트들을 읽는다. 특정 서비스주기동안에 각 디스크들은 특정 존 내의 세그먼트들만을 검색하므로 탐색지연시간은 RR정책의 경우보다 짧아진다. RR정책의 탐색지연을  $D_{latency}$ 라고 했을 때 VSP의 탐색지연 $Z_{latency}$ 는 식 (4)와 같이 표현될 수 있다.

$$Z_{latency} \approx \frac{D_{latency}}{Y} \quad (4)$$

( $Y$  : 단일 디스크에서의 존 개수)

$Z_{latency}$ 는  $D_{latency}$ 보다 짧으므로 VSP정책의 디스크 대역폭은 RR정책에 비해서 절약되고 절약된 디스크 대역폭만큼 더 많은 스트림들을 서비스할 수 있다. 그러나, VSP정책에서 시작 지연은 RR정책보다 더 길다. RR정책에서는 디스크 헤드의 임의이동을 기반으로 함으로 요구 스트림의 시작 세그먼트를 즉각적으로 읽을 수 있지만 VSP정책에서는 디스크 헤드가 다른 디스크들의 헤드들과 존 단위의 동기화를 유지하면서 요구 스트림의 시작 세그먼트가 있는 존으로 올 때 까지 기다려야 한다. 또한, 요구 스트림들의 세그먼트들이 특정 존 영역으로 집중될 경우 시작 지연은 더 길어질 수 있다.

### 4. AVSP 정책

VDBS의 하부저장시스템 성능을 향상시킬 수 있는 새로운 스트림 검색정책을 제안한다.

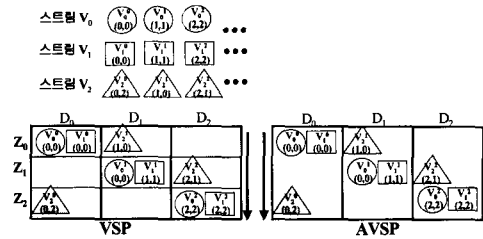
#### 4.1 AVSP 검색정책 개요

AVSP(Asynchronized VSP)정책은 스트림들을 배치할 때는 VSP정책을 이용하지만 스트림들의 검색 시에는 시작지연을 단축시키고 서비스 가능한 스트림 수를 많게 하기 위하여 기술적인 검색을 한다. AVSP 검색정책은 시작지연을 단축시키기 위하여 다중 디스크환경의 디스크시스템에서 각 디스크의 디스크헤드는 독립적으로 작동하고 존 단위의 동기화가 해제된다. 또한, 서비스 가능한 스트림 수를 많게 하기 위하여 AVSP검색정책은 각 디스크의 디스크헤드들을 한쪽 방향(안쪽 존에서 바깥쪽 존)으로만 이동하면서 세그먼트 검색을 함으로써 VSP만큼 탐색지연시간을 절약한다. AVSP정책은 전통적인 디스크 스케줄링 정책중의 하나인 SCAN 개념과 비슷하지만 비디오 스트림의 시간제약 특성을 만족시킨다. 그러나, AVSP 정책은 더블 버퍼링에만 적용된다. 더블버퍼링은 단일버퍼링보다 많은 버퍼량을 요구하기 때문에 버퍼용량의 제약으로 인하여 제공되는 디스크 대역폭을 충분히 활용할 수 없을 때 AVSP검색정책은 제공된 버퍼 용량과 제공된 디스크 대역폭을 동시에 고려하여 휴리스틱(heuristic) 방법에 의해서 세그먼트 크기를 결정함으로써 요구 버퍼량을 적게 한다.

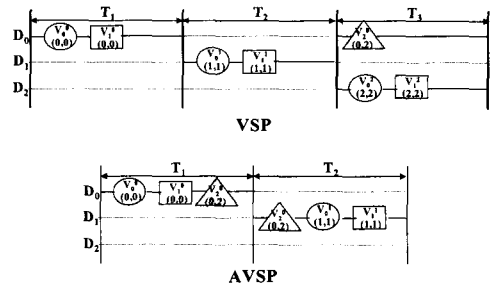
#### 4.2 AVSP 검색정책 및 예제

AVSP 검색정책은 세그먼트들에 적절한 우선순위(priority)를 부여하고 각 디스크에서는 우선순위 순으로 세그먼트들을 읽는다. 세그먼트의 우선순위는 순서쌍 (ASN, Z#)가 낮은 순으로 높은 우선순위가 부여된다. 여기서, ASN(Assigned Serial Number)은 스트림의 세그먼트들에 부여되는 번호인데 스트림내에서 그 세그먼트의 검색순서를 의미한다. 예를 들면, [그림 5]의 (a)에서 세그먼트 V01의 ASN은 2이고 V10의 ASN은 1이다. Z#는 해당 세그먼트가 위치한 존 번호를 의미한다. [그림 5]의 (a)에서 세그먼트

V01의 Z#는 1이고 V1<sup>0</sup>의 Z#는 0이다. 우선순위 순서쌍 (ASN, Z#)에서 ASN은 스트림 내에서의 세그먼트 검색 순서를 유지하기 위하여 필요하고 Z#는 디스크 헤드의 순차 이동을 유도하기 위하여 필요하다. 우선순위 순으로 세그먼트들을 읽으면 디스크 헤드가 순차적 이동을 하면서 세그먼트들을 읽는 것과 같은 효과를 유도하면서 시작지연은 단축시킬 수 있다. 즉, 세그먼트들 사이의 탐색지연시간이 D<sub>latency</sub> 대신에 Z<sub>latency</sub>+ β가 되고 시작 지연은 임의접근시의 시작 지연과 비슷하게 단축된다. 다음 부분에서 AVSP검색 정책의 예를 들면서 설명한다.



(a) 스트림 배치상태



(b) 스트림 검색 서비스

[그림 5] VSP와 AVSP의 검색 정책 비교  
[Fig.5] Retrieval policy comparison of VSP and AVSP

[그림 5]의 (a)는 3개의 스트림들 V<sub>0</sub>, V<sub>1</sub>, V<sub>2</sub>들이 VSP배치 정책을 이용하여 디스크의 존 상에 배치된 상태를 나타내고 있다. [그림 5]의 (a)에서 세그먼트 V<sub>i</sub><sup>j</sup>(a,b)는 세그먼트V<sub>i</sub><sup>j</sup>의 ASN이 a이고 Z#가 b임을 의미한다. 즉, 세그먼트 V<sub>1</sub><sup>0</sup>(0,0)은 V<sub>1</sub>의 ASN이 0이고 존 Z<sub>0</sub>에 배치되어 있음을 의미한다. [그림 5]의

(b)는 3개의 스트림들  $V_0, V_1, V_2$  들에 대한 요구가 발생했을 때 AVSP검색 정책과 VSP검색 정책이 각 스트림의 세그먼트들을 서비스하는 상황을 나타내고 있다. [그림 5] (b)에서 점선 부분은 다른 스트림들의 세그먼트들을 서비스하고 있음을 의미한다. VSP검색 정책은 존을 기준으로 디스크 헤드들 사이의 동기화를 유지하여야 하므로 서비스주기  $T_1$  동안에 존  $Z_0$  에서 스트림  $V_0$ 와  $V_1$ 의 세그먼트들  $V_0^0(0,0), V_1^0(0,0)$ 을 검색하고 디스크 대역폭의 여분이 있다고 할 지라도 스트림  $V_2$ 의 시작 세그먼트  $V_2^0(0,2)$ 를 서비스할 수 없다. 반면, AVSP검색 정책에서는 존들 사이의 동기화를 유지하지 않으므로 서비스 주기  $T_1$  동안에 스트림  $V_0$ 와  $V_1$ 의 세그먼트들  $V_0^0(0,0), V_1^0(0,0)$ 를 검색하고 디스크 대역폭의 여분이 있으면 스트림  $V_2$ 의 시작 세그먼트  $V_2^0(0,2)$ 를 서비스할 수 있다. 즉, VSP검색 정책은 스트림  $V_2$ 의 서비스를 서비스 주기  $T_3$ 에서 시작하지만 AVSP는 서비스 주기  $T_1$ 에서 시작할 수 있다. 여기서, AVSP정책은 각 서비스 주기동안 스트림들의 서비스 순서가 달라지지만 더블 버퍼링을 적용할 것이므로 문제가 되지 않는다.

### 4.3 AVSP 검색정책의 순차처리 성질

앞에서 제시한 우선순위 순으로 세그먼트들을 검색한다면 VSP정책의 성능과 비슷한 순차적 접근 효과를 가져온다는 성질을 살펴본다.

(정리1) 어떤 디스크의 Y개의 세그먼트들이 Z#의 순서에 의하여 서비스될 때 그 세그먼트들이 특정 존에 집중되어 있는 경우의 검색 시간( $RT_a$ )과 Y개의 존 상에 골고루 분포되어 있는 경우의 검색시간( $RT_b$ )은 동일하다(이때, Y는 존 수이고 디스크 헤드는 최종적으로 가장 바깥쪽 존 까지 이 동한다).

<증명>  $T_{move}$ 는 존 사이의 이동시간이고  $Z_{latency}$ 는 존에서의 탐색지연,  $T_{trans}$ 는 세그먼트 전송 시간이라 하자.

(Case1) Y개의 세그먼트들이 Y개의 존 들  $Z_0, Z_1, Z_2, \dots, Z_{Y-1}$ ,에 골고루 분포된 경우  
 $RT_a = Y * (T_{move} + Z_{latency} + T_{trans})$

(Case2) Y개의 세그먼트들이 임의의 존 들  $Z_a, Z_b, Z_{Y-1}$ 에 집중되어 있는 경우

Y개의 세그먼트들 중 p개의 세그먼트들은 존  $Z_a$ 에, q개의 세그먼트들은  $Z_b$ 에, r개의 세그먼트들은  $Z_{Y-1}$ ( $Y = p + q + r, 0 < a < b < Y$ )에 있다고 하자.

$$\begin{aligned} RT_b &= a * T_{move} + p * (Z_{latency} + R_{trans}) + (b(a) * T_{move} \\ &+ q * (Z_{latency} + T_{trans}) + (Y(b) * T_{move} + r \\ &* (Z_{latency} + T_{trans})) \\ &= (a+b * a + Y(b) * T_{move} + (p+q+r) \\ &* (Z_{latency} + T_{trans})) \\ &= Y * (T_{move} + Z_{latency} + T_{trans}) \\ &= RT_a \\ \therefore RT_a &= RT_b \end{aligned}$$

AVSP검색 정책에서 우선순위 순으로 세그먼트들을 검색하면 서비스 주기 동안에 검색할 세그먼트들이 항상 동일한 존 내에 존재하지는 않는다. 그러나, 단일 디스크에서 검색할 세그먼트들의 수가 존 갯수 이상이 되면 AVSP검색정책의 (정리1) 성질에 의하여 검색할 세그먼트들이 임의의 존 들에 존재한다 할 지라도 연속적인 존에 존재하는 경우로 간주할 수 있다. 따라서, AVSP검색 정책은 디스크 헤드가 연속적으로 존 들을 이동하면서 순차적으로 세그먼트 검색을 할 수 있으므로 탐색 지연 시간은 VSP정책에서 보다 약간 길어지지만 RR정책보다는 짧으므로 서비스주기 동안 RR정책 보다 더 많은 세그먼트들을 읽을 수 있다.

### 4.4 더블버퍼링의 보완방안

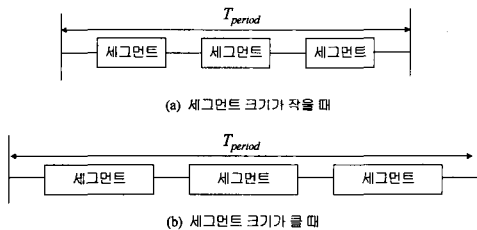
이 정책은 단일 버퍼링에서 요구되는 버퍼량으로 더블 버퍼링을 적용했을 때에도 버퍼 제약으로 인한 서비스 가능한 스트림 수의 한계를 최소화시키고자 하는 목적에서 제안한다.

세그먼트 크기가 크면 많은 버퍼량을 요구하지만 서비스 주기가 길어져서 서비스 주기동안 많은 세그먼트들을 검색할 수 있을 것이라고 기대할 수 있다. 그러나, 세그먼트가 커짐에 따라 세그먼트 전송시간도 증가하게 되고 세그먼트 검색 사이의 탐색지연시간이 세그먼트 전송시간에 비하여 무시될 수 있는 상

황이 발생한다. 이 때, 세그먼트 크기를 계속 증가시켜서 서비스 주기가 계속 길어진다 할 지라도 그 서비스주기동안 읽을 수 있는 세그먼트들의 수는 [디스크전송률/스트림소모율] 이상 증가하지 않는다. 따라서, AVSP검색정책에서 디스크 처리율을 최대로 하는 세그먼트 크기  $S_{max}$ 는 식 (1)로부터 식 (5)처럼 계산할 수 있다.

$$S_{max} = \frac{Z_{latency} + \beta}{\left( \frac{1}{\left[ \frac{R_{disk}}{R_{med}} \right] * R_{med}} - \frac{1}{R_{disk}} \right)} \quad (5)$$

세그먼트 크기가  $S_{max}$  이상이 되면 세그먼트 크기가 커져도 서비스 주기동안 읽을 수 있는 세그먼트들의 수는 증가하지 않는다. 비슷한 개념으로, 세그먼트 크기가 작아짐에 따라 서비스 주기는 짧아지는 반면 그와 비례해서 읽어야 할 세그먼트들의 크기도 작아지므로 서비스 주기동안 읽을 수 있는 세그먼트들의 수는 세그먼트 크기가 작아지는 것과 비례해서 감소하지는 않는다. [그림 6]은 이러한 개념을 도식적으로 나타내고 있다.



[그림 6] 서비스 주기와 세그먼트 크기와의 관계  
[Fig.6] Service period and segment size

크기가  $S_{max}$ 인 세그먼트가 재생되는 시간동안 단일 디스크에서  $M_{max}$ 개의 세그먼트들을 읽을 수 있다고 하면 X개의 디스크들로 구성된 다중 디스크환경에서의 최대 동시허용 스트림 수는  $M_{max} * X$  (X는 디스크 수)가 되지만 이것은 디스크 대역폭만을 고려하여 계산된 것이고 요구 버퍼량은  $2 * S_{max} * M_{max} * X$  (더블 버퍼링 일 때)가 된다. 제공된 버퍼 용량이 이 보다

적으면 디스크 대역폭이 지원할 수 있는 동시 허용 스트림 수보다 더 적은 스트림들을 지원한다.

[그림 6]에서 보여주는 개념에 따라 세그먼트 크기가 작아지더라도 허용 스트림 수가 그에 비례해서 적어지는 것은 아니므로 버퍼 용량을 고려하여 세그먼트 크기를 휴리스틱(heuristic) 방법에 의하여 조절할 필요가 있다. [그림 7]은 제공된 버퍼 용량을 고려하여 세그먼트 크기를 계산하는 알고리즘을 보여준다.

Procedure ADJUST

```

Input : Rdisk, Rmed, B, X, Y
      // B: 제공된 버퍼크기,
      X: 디스크 수, Y : 존 수 //
      GOPsize //MPEG에서 1 GOP의 크기 //
Output : SEGsize // 조절된 세그먼트 크기 //
Begin
(1) Mmax ← [ Rdisk / Rmed ] ;
(2) Smax ← (Zlatency + (β) / (1 / (Mmax * Rmed) - 1 / Rdisk ) ;
(3) strip ← [ Smax / GOPsize ] ;
(4) Btemp ← Mmax * 2 * GOPsize * strip * X ;
(5) while (Btemp >= B)
      strip ← strip - 1;
      Btemp ← Mmax * 2 * GOPsize * strip * X ;
End while
(6) SEGsize ← strip * GOPsize ;
end
    
```

[그림 7] 세그먼트 크기를 조절하는 알고리즘

[Fig.7] ADJUST algorithm

[그림 7]에서 단계(5)는 버퍼 용량을 고려하여 세그먼트 크기를 조절하기 위한 과정이다. 제공된 버퍼 크기가 충분하지 않으면 세그먼트 크기를 GOP단위로 줄이는 과정을 반복한다.

5. 실험 및 결과 분석

이 장에서는 제안한 AVSP 정책을 동시허용 스트림 수와 평균 시작지연시간을 기준으로 RR 및 VSP 정책과 비교하여 성능평가 및 분석을 한다. 실험을 위한 몇 가지 가정을 설정하였다. 첫째, 임의의 수

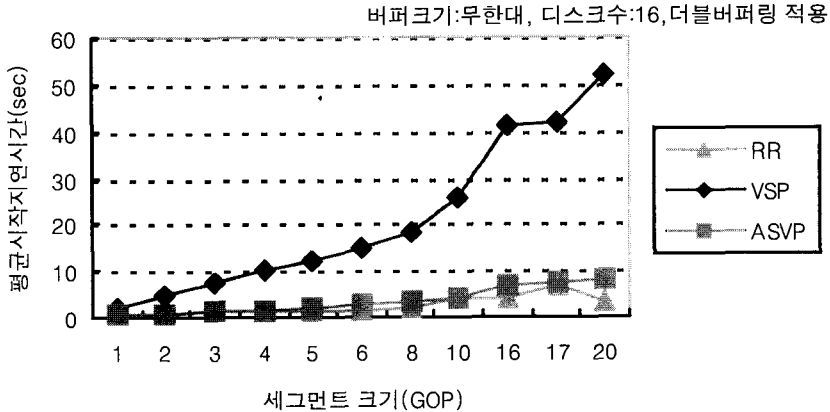


의 비디오 스트림들은 MPEG-I으로 압축되어 관련 배치 정책을 기반으로 다중 디스크 상에 배치되어 있다고 가정하고 랜덤 함수를 사용하여 요구스트림의 시작 세그먼트가 있는 디스크와 존이 결정된다. 둘째, 요구 스트림의 도착 시간은 평균이 3초인 Poisson분포를 따른다. 셋째, 시작지연시간은 요구 비디오데이터의 위치를 인덱스DB로부터 계산한 후부터 요구비디오의 첫 번째 세그먼트를 읽을때까지의 시간이다. 넷째, 시작지연시간 측정 시 동시에 (concurrently) 허용 가능한 스트림 수까지의 평균 시작 지연 시간만을 고려한다.

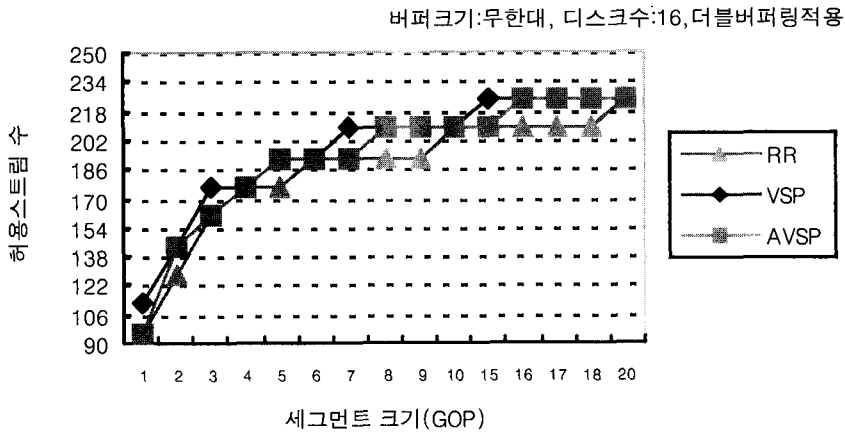
실험을 위한 디스크 모델은 Seagate Wren8 ST41650N[6]를 사용한다. 시뮬레이터(Simulator)는 Visual SLAM의 PC 버전인 AweSim2.0[7]과 C언어를 이용하여 구현되었고 펜티엄III의 윈도우98 환경에서 수행된다.

이상과 같은 실험 환경하에서 시뮬레이션은 다중 디스크 환경 및 버퍼 용량을 변화시켜 가면서 동시 허용 스트림 수 및 평균 시작 지연 시간을 측정한다.

[그림 8]은 제공되는 버퍼 크기를 무한대로 하고 다중 디스크 환경의 디스크 수를 16으로 하였을 때 세그먼트 크기 당 평균 시작 지연 시간을 보여주고

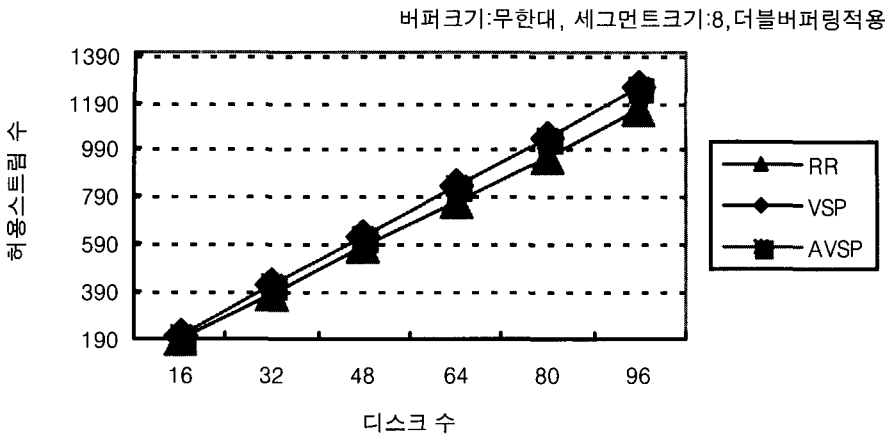


[그림 8] 최대 허용 스트림 수까지의 평균 시작 지연



[그림 9] 세그먼트 크기 당 허용 스트림 수

[Fig.9] The # of the admitted streams per segment size



[그림 10] 디스크 수 당 허용 스트림 수(세그먼트 크기 : 8)

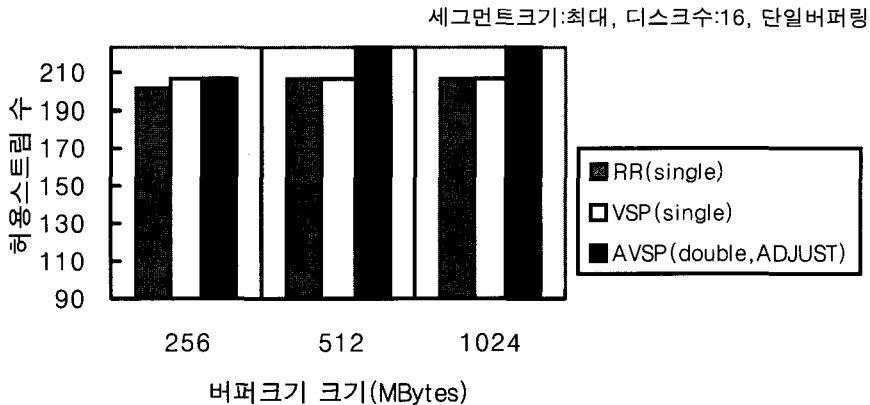
[Fig.10] The #of the admitted streams per the # of disks (segmen size : 8)

있다. 세그먼트 크기는 MPEG의 GOP크기의 배수이다. AVSP는 시작지연시간이 VSP보다 짧고 RR과 거의 비슷함을 알 수 있다. 전체적으로 세그먼트 크기가 작을 때는 세그먼트 크기가 클 때에 비하여 평균 시작지연시간이 짧다.

[그림 9]는 제공되는 버퍼 크기를 무한대로 하고 다중 디스크 환경의 디스크 수를 16으로 하였을 때 세그먼트 크기 당 최대의 동시 허용 스트림 수를 보여주고 있다. 디스크 처리율을 최대로 하는 세그먼트

크기(이때, 최대로 서비스 가능한 스트림 수는 224개 이다)는 RR 정책의 경우 20이고 VSP정책은 15, AVSP정책은 16임을 알 수 있다. 이때, 세그먼트 크기가 작을수록 버퍼용량을 절약할 수 있으므로 최대 스트림 수를 지원하기 위하여 요구되는 세그먼트 크기가 작을수록 좋다. 따라서 RR보다는 VSP나 AVSP가 좋은 정책임을 알 수 있다.

[그림10]은 세그먼트 크기가 8일 때 디스크 수당 허용 스트림 수를 보여주고 있다. VSP와 AVSP정



[그림 11] ADJUST알고리즘의 효과

[Fig.11] The effect of ADJUST algorithm

책이 타 정책에 비하여 7% 정도 더 많은 스트림들을 허용하고 있음을 알 수 있다.

[그림 11]은 ADJUST알고리즘(더블버퍼링의 보완 방안)의 효과를 나타내고 있다. 여기서는 디스크 처리율을 최대로 할 수 있는 세그먼트크기를 가정하고 있다. ADJUST알고리즘은 단일 버퍼링에서 요구되는 버퍼량으로 더블 버퍼링을 적용할 때에도 서비스 가능한 스트림들의 수를 많게 하기 위한 정책이다. [그림11]에서 보여주듯이 버퍼 용량이 256MB이면 RR 정책에서 단일 버퍼링을 사용한다 할 지라도 버퍼 제약으로 인하여 디스크 대역폭이 지원할 수 있는 최대 스트림 수를 제공할 수 없다. 반면, AVSP정책에서는 더블버퍼링을 사용하면서도 ADJUST알고리즘의 효과에 의하여 RR정책보다 더 많은 스트림들을 서비스하고 있음을 알 수 있다.

## 6. 결론

비디오 데이터는 시각적 효과를 극대화할 수 있기 때문에 컴퓨터와 관련된 다양한 분야에서 상업적 잠재력이 매우 크다. 또한, 전세계적으로 많은 양의 비디오 데이터가 존재하고 생성되고 있기 때문에 이러한 비디오 데이터를 체계적이고 효율적으로 관리할 수 있는 VDBS의 필요성 및 중요성은 날로 증대되고 있다.

본 논문에서는 VDBS의 하부 저장시스템과 관련한 새로운 스트림 검색정책을 제안하였다. 제안한 정책은 사용자들의 신속한 스트림 검색요구를 만족시키면서 VDBS가 지원할 수 있는 사용자들의 수를 많게 할 수 있는 방안이다. 제안한 AVSP검색정책은 스트림들의 세그먼트들에 우선순위를 부여하여 우선 순위순으로 세그먼트검색을 함으로써 디스크헤드의 순차적 이동효과를 유도하고 세그먼트검색 사이의 탐색 지연 시간을 단축시킬 수 있다. 또한, 다중 디스크환경에서 각 디스크헤드들의 이동을 비동기화시킴으로써 하부저장시스템으로의 서비스요구에 대한 시작지연을 단축시킬 수 있다. AVSP 검색정책이 더블 버퍼링을 기반으로 수행되므로 타 정책에 비하여 더 많은 버퍼를 필요로 한다. 이를 보완하기 위하여 제공된 버퍼 용량과 디스크 대역폭을 고려하여 최적의 세그먼트 크기를 결정할 수 있는 휴리스틱 알고

리즘 ADJUST를 제안하였다. ADJUST는 단일버퍼 관리기법에서 요구되는 버퍼량으로도 AVSP검색정책을 효율적으로 지원할 수 있음을 알 수 있었다.

## ※ 참고문헌

- [1] Oge Marques and Borko Furht, Issues in Designing Contemporary Video Database Systems, *Proceedings of IASTED International Conference IMSA '99*, Oct, 1999, pp43-47.
- [2] H.-J.Zhang, "Content-based video browsing and retrieval", in *Handbook of Internet and Multimedia Systems and Applications*, B.Furht(ed). Boca Raton:CRC Press,1999.
- [3] B.-L. Yeo and M.M.Yeing,"Retrieving and Visualizing Video", *Communications of the ACM*, Vol.40, No.12, December 1997.
- [4] Edward Chang and Hector Garcia-Molina, BubbleUp:Low Latency Fast Scan for Media Servers, *Proceedings of ACM Multimedia 97*, 1997, 87-98.
- [5] Gin-Kou Ma, Chiung-Shien Wu, Mei-Chian Liu, and Bao-Shuh P. Lin, Efficient Real-time Data Retrieval Through Scalable Multimedia Storage, *Proceedings of ACM Multimedia 97*, 1997, pp165-172.
- [6] Huang-Jen Chen, Thomas D.C. Little, Storage Allocation Policies for Time-Dependent Multimedia Data, *IEEE Transactions on Knowledge and Data Engineering*, Vol.8, No.5, Oct, 1996, pp855-864.
- [7] Taeck-Geun Kwon, Sukho Lee, Data Placement for Continuous Media in Multimedia DBMS, *Proceedings of IEEE Multimedia DBMS*, August 28-30, 1995, pp110-117.
- [8] Patterson, D., Gibson, G., and Katz, R., A case for Redundant Arrays of Inexpensive Disks (RAID), *Proceedings of SIGMOS'88*.
- [9] A. Alan and B. Prisker, Simulation with Visual SLAM and AweSim, Systems Publishing Corporation, 1997.

- [10] D.James Gemmel, Harrick M.Vin, Dilip D.K andlur, P.VenkatRangan, Multimedia Storage Server: A Tutorial and Survey, *IEEE Computer*, Vol. 28, No.5, May, 1995, pp40-49.
- [11] Steven Berson, Shahram Ghandeharizadeh, Staggered Striping in Multimedia Information Systems, *Proceedings of ACM SIGMOD*, 1994, pp79-90.
- [12] B.Ozden, R.Rastigi and A.Siberschatz, Disk Striping in Video Server Environments, *Data Engineering*, 18(4), 1995, pp4-16.
- [13] S.Park, KeunHyung.Kim, Efficient Storage and Retrieval Scheme of Video Streams in VOD Server, *Proceedings of IASTED International Conference IMSA '99*, 1999, pp163-169.
- [14] M.S. Chen, D. Kandlur, P. Yu, Support for Fully Interactive Playout In Disk-Array-Based Video Server, *Proceedings of ACM Multimedia 94*, 1994, pp391-398.

김근형



1986. 3~1990. 2 서강대학교  
컴퓨터학과 졸업(학사)  
1990. 3~1992. 2 서강대학교  
대학원 컴퓨터학과(DB전공)  
졸업(석사)  
1995. 3~2001. 2 서강대학교  
대학원 컴퓨터학과(DB 전공)  
졸업(박사)  
1992. 2~1994. 2 현대전자  
소프트웨어 연구소 연구원  
1994. ~ 현재 제주산업정보대학  
컴퓨터정보계열 조교수