# 저용량, 고화질 비디오 압축 브라우징에 대한 설계

강진석* , 김무영 , 김장형

제주대학교 정보공학과

# The Design of Video Compression Browsing for Low Capacity and High Quality

jin-suk kang* , moo-young Kim , jang-hung Kim

*Dept. of Information Engineering, Cheju National Univ.

SUMMARY

In the 21th century, everyone feels that the multimedia system is close at hand in real life due to the rapid advance of the computer processing ability and high speed and high guality of communication services. Also the limited frequencies resource will be optimized due to rapid advances in digital video technology which is believed superior to analogue technology in information engineering. MEPG-2 has been introduced for broadcasting use such as digital TV Thus it features the high-definition and hyper-low bit rate. But, because of much throughput it has been implemented by high-priced private ASIC chip and is not in general use yet. But in this research, noticing the rapid enhancement of PC processor performance comparing with the price. MPEG-2 was developed by real time software MPEG-2 had been known impossible to implement with S/W, but the research proved the possibility of the S/W implementation and below are the pictures also in the research was improved 'Motion Vector and Compensation' Algorithm which requires the most operations and DCT was made possible real time process. Multimedia Info Society has settled and accompanied by the rapid advance of image-processing technology and lots of standards.

## I. 서 론

21세기는 컴퓨터의 처리 능력의 향상 및 통신의 고속화, 고신뢰도 및 고도의 서비스 등 디지털의 발전을 통한 멀티미디어 시스템이 현실적으로 가까워졌다. 특히 멀티미디어의 압축 처리부의 중점적으로 확대 및 필요성으로 인하여, 데이터의 비트율이 점점 더 낮아져 가는 권고안을 토대로 연구가 진행되었으며, 이러한 발전을 토대로 현재 정보 자원으로써 비디오의 실제 이용은 쉽고 효과적으로 사용할 수 있게 되었다. 현재 상태의 기술은 logging 처리를 사용하여 데이터베이스에 입력된 구조화된 텍스트 방법으로 비디오 소스를 취급하고 있다. 따라서 이러한 전반적인 내용을 토대로 살펴볼 때, 효과적인 Tool이 없는 상태이다. 이러한 어려움을 극복하기 위해서 적어도 세 가지 타스크를 제공하는 툴이 필요하다[5, 8].

첫째, 비디오를 클립이라 불리는 적당한 인덱싱 단위로 분할한다.

둘째, 클립을 식별하기 위해서 저차원 영상 특성과 의미론적인 특성을 알아야 한다.

셋째, 질의로부터 결과를 비디오 자원으로부터 선택하는 검색과 브라우징이 필요하다.

이러한 내용으로 볼 때 지금까지의 연구는 압축되지 않은 비디오를 대상으로 했으며 일반적으로 아날로그 소스를 디지털화하여 왔다. 그러나 이 논문에서는 비디오 데이터를 압축된 디지털 형태로 저장하고 분류하는 기초 연구이며, 본 논문에서는 MPEG을 사용하여 압축된 비디오 데이터로 비디오 파싱, 검색, 브라우징을 지원하는 툴을 개발한 연구 내용을 발표한다. 먼저, 2장에서는 고화질, 저용량 매체인 MPEG-2 Video의 간략한 소개와, 3장에서는 비디오 브라우징을 설계하는데 있어서 가장 기본적이면서 핵심적인 MPEG-2 Video의 실시간 알고리즘에 대해 기술하며, 4장과 5장에서는 카메라를 통해 압축된 비디오의 내용을 기반으로 한 브라우징과 실험결과에 대해 고찰한다.

## II. MPEG-2 Video

MPEG-1 비디오가 CD등 디지털 축적 매체에 1.15Mbps의 저비트율로 동화상을 저장하는데 반해, MPEG-2 비디오는 보다 고비트율의 방송, 통신, 축적 미디어에서 고화질의 동화상을 전송하거나 저장하는데 사용된다. 응용분야가 다양해진 만큼 충족시켜야 할 요구조건도 많아졌다.

MPEG-2 비디오는 일종의 범용 압축 알고리듬으로, MPEG-1 비디오를 크게 확장 발전시키면서 많은 도구들을 마련하여 응용분야에 따라 이들을 적절히 선택 사용하도록 하고 있다. 압축효율의 향상을 위해 MPEG-2 비디오는 MPEG-1 비디오의 각 요소들을 재검토하여 조금씩 개선함으로써 전체적으로 향상됨을 알수 있고, 즉 필드단위의 처리, 움직임 추정과 보상방식, 양자화, DCT 계수와 주사방식 그리고 가변장 부호화 등 많은 부분들이 개선되었다. 또 기능에 따라서는 5개의 프로필로 나누어진다. 양방향 예측을 이용하는 B프레임을 제외하여 구현을 용이하게 한 Simple, 많은 기능을 포함하여 대부분의 응용분야에 채택되고 있는 Main, 계층 구조를 가지면서 보다 기능이 확장 된 SNR Scalable, Spatial Scalable, High 등이 그것이다. MPEG-2 비디오에서는 TV에서 사용되고 있는 비월주사 방식의 동화에 대해서도 많은 고려를 하고 있고, 프레임 구조로 부호화 할 수도 있다. 움직임이 많은 장면은 한 프레임의 두 필드간에도 큰 차이가 나므로 필드구조로 부호화하는 것이 효과적이고, 정지화에 가까울수록 두 필드간에 상관도가 높아 프레임 구조로 부호화하는 것이 유리하다. 또 프레임 구조의 부호화에 있어서도 각 매크로블록(16×16화소 단위)별로 필드단위의 처리가 가능하도록 하여 화면내의 부분적 움직임을 용이하게 처리할 수 있게 하고 있다.

## III. 압축 데이터를 기반으로 한 Video 파싱

이 장에서는 JPEG과 MPEG 비디오 압축 표준을 간단하게 검토하고, JPEG이나 MPEG으로 압축된 비디오를 분할하는 세 가지 다른 기술을 검토한다. : 두 가지 방법은 JPEG과 MPEG 모두에 사용된 DCT 계수를 사용하였고 세 번째 방법은 MPEG 움직임벡터를 사용하였다. 이에 따라서 다중-패스, 이중 비교법을 통합한 방법과 점진적인 변화와 카메라 동작을 검출할 수 있는 움직임 분석의 두 가지 접근법의 힘을 결합하기 위한 방법을 논의한다.

### 1 JPEG and MPEG 의 개요

본 논문의 알고리즘을 소개하기 전에 본 연구의 배경이 된 JPEG과 MPEG의 원리를 요약한다. Fig. 1은 정지 영상 압축 표준 JPEG의 기본 과정을 설명한다. 주요 개념은 공간적 여분을 줄이는 것이다. 압축은 이미지를 8×8 픽셀 블록들로 나누는데서 시작된다[2, 6, 4]. 각 블록의 세기 정도는 이산여현변환(DCT)에 의해 64 계수로 변환되고, 양자화와 run-length, 허프만 엔트로피로 부호화 된다.

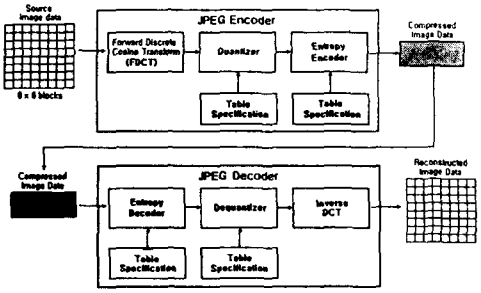


Fig. 1. 정지영상 압축 표준

MPEG, 동영상 압축 표준은 Moving Picture Experts Group에 의해 제안되었다. MPEG은 JPEG과 같이 공간적 여분을 줄이는 것이지만 시간적 여분을 줄이기 위해 움직임 보상을 사용한다. 그렇지만 이 방법은 계수 코딩과 빠른 랜덤 접근사이의 교환을 수반한다. 랜덤 접근을 용이하게 하기 위해서 어떤 프레임은 움직임 보상 부호화 없이 still JPEG 영상처럼 부호화된다: 이들 프레임을 *intra-pictures(I)*라 한다[2, 4]. 프레임의 두 가지 타입은 움직임 보상으로 부호화된다: 이들 프레임을 predictively coded (P) 프레임과 bi-directional predictively coded (B) 프레임이라 한다. Fig. 2.에 이 세 가지 타입을 설명하였다. 움직임벡터의 두 집합(매 8×8 블록중 하나)은 앞뒤이고, 각각의 B프레임과 각 P프레임의 단일 집합과 관련이 있다. P와 B프레임에서 각 블록의 움직임 보상과 관련이 있는 나머지 차이점은 JPEG 프레임에서 블록같이 같은 방법으로 DCT로 부호화된다. 본 논문에서는 비디오 파싱을 위해서 DCT와 움직임 보상 데이터를 이용하는 방법을 연구한다.

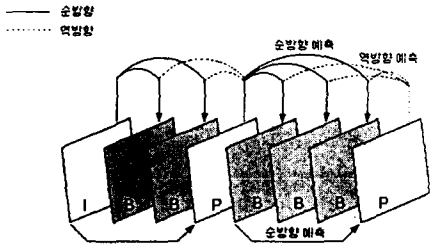


Fig. 2. 시간적 픽쳐 구조

### 2 DCT를 기반으로 한 알고리즘

공식화된 알고리즘은 Arman등에 의해 개발되었다. [1]은

JPEG으로 압축된 비디오의 연속된 프레임의 DCT 계수사이의 관계에 기반을 두었다. 또한 보통의 영상인 경우에는 통계적으로 고주파보다는 저주파가 많기 때문에 화상을 주파수별로 분해하면 값이 저주파에 몰리게되고 고주파에는 거의 존재하지 않아서 작은 값을 갖는 고주파를 무시해도 화상에는 별로 영향을 미치지 않는다. 이러한 압축이 수행되는 것이 DCT이고, 이때 부호화하는 것이 DCT이고, 복호화하는 것이 역DCT (Inverse DCT, IDCT)이다. N×N의 2차원 DCT의 수식을 보면 다음과 같다.

■ DCT 수식

$$F(u,v)=\frac{2}{N}C(u)\,C(v)\sum_{x=0}^{N-1}f(x,y)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \quad (1)$$

- $u,v,x,y=0,\ 1,\ 2,\ \cdots N-1$
- $x,\ y=$ 샘플 영역에서의 공간좌표
- $u,\ v=$ 변환 영역에서의 공간좌표

$$C(u),\ C(v)=\begin{cases}\frac{1}{2} & u,\ v=0 \text{ 일때}\\ 1 & \text{그렇지않을때}\end{cases} \quad (2)$$

■ IDCT 수식

$$F(u,v)=\frac{2}{N}\sum_{x=0}^{N-1}\sum_{y=0}^{N-1}C(u)C(v)F(u,v)\cos\frac{(2x+1)u\pi}{2N}\cos\frac{(2y+1)v\pi}{2N} \quad (3)$$

또한, 각 프레임의 백터 표시는 프레임안에 블록의 부분집합에 의해 구성된다.

$$V_f=\{c_1,c_2,c_3,\ldots,c_k\} \quad (4)$$

프레임간의 차이값은 정규화된 내적에 의해 다음과 같이 정의된다:

$$\Psi=1-\frac{|V_f\cdot V_{f+\varphi}|}{|V_f||V_{f+\varphi}|} \quad (5)$$

비교되는 두 프레임사이의 프레임의 수가 $\varphi$이다.

화소단위비교법의 접근[4]에 따르면 연속되는 비디오 프레임에 대응되는 블록의 DCT계수를 비교하는 화소단위 블록 알고리즘이 개발되었다. 보다 엄밀히 말해서, $c_{l,k}(i)$를 프레임 $i$에서 블록 $l$의 DCT계수라고 하면, $k$의 범위는 1에서 64까지이고 $l$은 프레임의 크기와 상관있다. $\varphi$ 프레임 떨어져 있는 두 프레임에서 블록 $l$의 내용 차이는 다음과 같은 식으로 측정할 수 있다.

$$Diff_l=\frac{1}{64}\sum_{k=1}^{64}\frac{|c_{l,k}(i)-c_{l,k}(i+\varphi)|}{\max[c_{l,k}(i),c_{l,k}(i+\varphi)]}\cdot 100\% \quad (6)$$

만일 이 차이가 주어진 경계값 $t$를 초과하면:

$$Diff_l > t \quad (7)$$

두 프레임사이에 변화가 있다는 것을 알 수 있다. 만일 $D(i,\ i+\varphi)$이 변경된 블록의 퍼센트로 정의된다면 분할 경계는 다음과 같이 선언된다.

$$D(i,\ i+\varphi) > T_b \quad (8)$$

여기서 $T_b$는 카메라 브레이크에 대한 경계값이다. 이 차이값은 픽셀의 세기 대신에 DCT계수를 사용한 화소단위비교법과 유사한 방법이다.

### 3 움직임 벡터에 기반 한 분할 방법

세기 값과 분포는 별도로 하고, 객체와 카메라의 동작은 비디오 내용의 중요한 요소가 된다. 일반적으로 단순한 카메라 샷 내에서 움직임벡터는 비교적 계속적인 변화를 보인다. 이 연속성은 다른 샷과 교차하는 프레임을 분열시킨다. 그러므로, 움직임벡터 영역의 순서에 대한 연속적인 값은 경계값을 검출하는 또 하나의 기준으로 제공될 수 있다.

Fig. 2.에서 설명한 MPEG 데이터 스트림에서 각각의 B 프레임과 관련된 앞 뒤 움직임벡터의 두 집합과 각각의 P 프레임과 관련된 움직임벡터의 한 집합이 있다. 이것은 각각의 B프레임이 움직임 보상에 의한 전후의 I/P 프레임으로부터 예측되고 수정됨을 의미한다. 그리고 각각 P프레임은 앞선 I/P프레임으로부터 유사하게 예측된다. 이 두 경우에서 움직임 보상 후에 오차 에러는 DCT계수로 변환되고, 부호화 된다. 그러나, 만약 이 오차 에러가 어떤 블록에 대해 주어진 경계값을 초과한다면, 움직임 보상 예측은 버려지고, 그들 블록은 I 프레임과 같이 DCT계수로 표현된다. 그 결과로서, B/P프레임에서 블록과 관련된 움직임벡터는 없게 될 것이다. 이와 같이 높은 오차 에러 값은 카메라 샷 경계를 교차하는 블록에서의 많은 경우에 비슷하게 발생될 수 있고, 이 정보를 이용하는 알고리즘을 개발하였다.

$M$을 각 P프레임에 대한 확실한 움직임벡터의 수와 각 B프레임에 대한 확실한 전후 움직임벡터의 수를 작게 하고, $T_b$는 0에 가까운 경계값이다.

$$M < T_b \quad (9)$$

(삽입되기 전인지 후인지에 따라) B/P 프레임의 카메라 앞 뒤 경계를 효과적으로 표시할 수 있다. 이 경우에 카메라 브레이크는 1-17까지의 번호를 매긴 경계 레벨이하 계곡 모양으로 모두 정확하게 표시된다. MPEG으로 압축된 다큐멘터리 비디오로부터 모든 연속하는 I 프레임들간의 차이 값의 순서는 DCT 계수와 화소단위 비교법으로 정의되고 이 특별한 순서는 뚜렷한 브레이크, 점진적 변화, 카메라 동작과 객체 움직임을 포함하고 실험한 결과 우수했다. 약 1.3분 동안, 25개의 샷을 17개의 뚜렷한 브레이크와 7개의 점진적 변화로 분리해냈다. 또한 4개의 샷에 카메라 패닝이 포함되어 있고, 17개의 뚜렷한

브레이크는 1-17까지 번호가 붙여있고 모든 것은 브레이크 경계값 $T_b$의 적당한 값을 초과하는 것을 검출한다. 소개한 결과를 표로 만들어 요약하였다. 먼저 카메라 브레이크의 검출을 고려한다.

Table. 1. 카메라 브레이크 검출요소

| | 검출된 카메라 브레이크 | 검출하지 못한 카메라 브레이크 | 잘못 검출된 카메라 브레이크 |
|---|---|---|---|
| DCT1 | 16 | 1 | 4 |
| DCT2 | 17 | 0 | 4 |
| 움직임벡터 | 17 | 0 | 0 |
| 복합적인 방법 | 17 | 0 | 0 |

이 표에서 DCT2가 본 논문 자체의 화소단위 블록 비교법이라면, DCT1은 [1]의 DCT 계수와 비교하여 설명한다. 본 논문에서는 다음과 같이 점진적인 변화의 검출을 요약한다.

Table. 2. 점진적인 변화의 검출

| | 검출된 변화 | 검출되지 않은 변화 | 잘못 검출된 변화 |
|---|---|---|---|
| DCT1 | 3 | 5 | 1 |
| DCT2 | 4 | 4 | 1 |
| 움직임벡터 | 0 | 0 | 0 |
| 복합적인 방법 | 7 | 0 | 0 |

마지막으로, 설명한 카메라 움직임 검출에 대한 본 논문에서 제안한 방법은 테스트 데이터에서 카메라 작동의 4가지를 모두 검출했다.

## 4 실시간 MPEG-2 비디오 압축을 위한 알고리듬

MPEG-2 Video을 부호화하고 복호화 과정은 시스템의 많은 부하를 초래하는 것을 피할 수 없다. 이에 본 연구에서 실시간 비디오 브라우징 시스템을 목적으로 MPEG-2 비디오를 이용하기 위해서 MPEG-2 비디오를 실 시간적으로 부호화하고 복호화가 가능하도록 연구하였다. MPEG-2 부호화 시스템에서 움직임 추정 과정은 가장 핵심적인 과정일 뿐만 아니라 연산이 복잡하여 지연이 가장 심한 과정이기도 하다. 그 원인으로는 속도를 고려하지 않은 전역탐색 블록매칭 기법을 이용하였기 때문이다. 이러한 이유로 본 연구에서 MPEG-2 인코딩을 위한 실시간 움직임 추정 알고리듬을 개발하기 위해 블록매칭 기법과 그래디언트 기법을 혼합한다. 이를 간단하게 블록도로 표시하면 Fig 3과 같다.

또한, 그래디언트 기법을 이용한 움직임 추정에서는 탐색 영역이 불필요하게 된다. 반면에 모든 벡터들이 추정되어진 후에 벡터 범위가 계산되어진다는 문제점을 야기 시킨다. 그러므로 움직임 추정에서의 그래디언트 기법이 한 픽쳐의 코딩 지연을 암시한다는 것을 이용하여 벡터의 범위(f_code)들은 어떤 매크로블록이전에 전송되어져야만 한다.

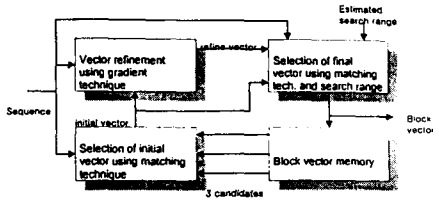


Fig 3. 하이브리드 매칭 그래디언트 기법을 이용한 움직임 추정 알고리듬의 블록도

이러한 문제점을 피하기 위해, 본 연구에서 탐색 범위를 결정하는 적응된 스키마를 연구해야 한다. 이 스키마는 움직임 벡터 필드의 후 분석(post-analysis)에 기반하며, 그 개념은 연속되는 픽쳐들 사이에 움직임의 연속성을 이용하여 현재 픽쳐의 움직임 벡터 필드를 분석함으로써 코딩 순서에 있어서 다음 픽쳐에 대한 탐색영역을 결정하는 것이다. Fig. 4.는 움직임 추정을 위한 탐색 범위를 결정하는 적응된 스키마를 도식한다. 적응된 스키마는 다음과 같은 4가지 단계로 구성된다:

① 탐색 범위의 적응된 결정
② 결정된 탐색 범위의 정규화
③ 정규화된 탐색 범위의 스케일링
④ 스케일된 탐색 범위의 표준화

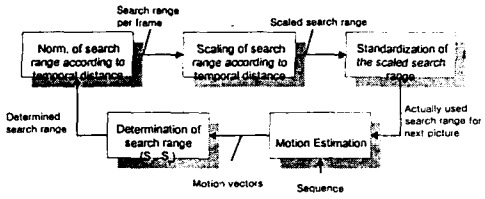


Fig. 4 움직임추정에서 적응된 탐색범위 결정기법

수평 탐색 범위 $S_x$와 수직 탐색 영역 $S_y$는 독립적으로 결정된다. 절대적 움직임 벡터들이 이 결정에 이용된다.

$MB_{cnt}$를 픽쳐내의 총 매크로블록 수로 놓고, { $(V_x^j, V_y^j) \mid j = 0, 1, 2, ..MB_{cnt}-1$ }이 현재 픽쳐에서 적당치로 이용된 움직임 벡터라 할수 있으므로, 탐색 범위는 다음과 같이 정의된다:

$$\begin{cases} S_x = E(|V_x|) + 2 \cdot D(|V_x|) \\ S_y = E(|V_y|) + 2 \cdot D(|V_y|) \end{cases}$$

$$\begin{cases} E(|V_x|) = \dfrac{1}{MB_{cnt}} \displaystyle\sum_{j=0}^{MB_{cnt}-1} |V_x^j| \\ E(|V_y|) = \dfrac{1}{MB_{cnt}} \displaystyle\sum_{j=0}^{MB_{cnt}-1} |V_y^j| \end{cases}$$

$$\begin{cases} D(|V_x|) = \dfrac{1}{MV_{cnt}} \displaystyle\sum_{j=0}^{MB_{cnt}-1} \| V_x^j - E(|V_x|) \| \\ D(|V_y|) = \dfrac{1}{MV_{cnt}} \displaystyle\sum_{j=0}^{MB_{cnt}-1} \| V_y^j - E(|V_y|) \| \end{cases}$$

탐색 범위가 결정되어진 후에, 결정된 탐색 범위는 참조 프레임과 현재 프레임사이의 시간적인 거리를 이용하여 프레임당 탐색 범위(search range per frame)로 정규화된다. 그리고 나서, 다음 픽쳐와 그 참조 픽쳐(들)사이의 시간적인 거리에 따르는 정규화된 탐색 범위를 스케일링함으로써 다음 픽쳐의 움직임 추정을 위한 탐색 범위가 얻어지게 된다.

마지막으로, 다음 픽쳐를 위한 스케일된 탐색 범위(순방향/역방향)이 표준화된다. S를 스케일된 탐색 범위(수평/수직/순방향/역방향)로 놓고, 탐색 범위는 Table. 3. 에 보여진 것처럼 표준화된다.

Table. 3. 스케일된 탐색범위

| range of S | [1:8] | [9:16] | [17:32] | [33:64] | [65:128] | [129:256] | [256:512] | [512:1024] | [1025:2048] |
|---|---|---|---|---|---|---|---|---|---|
| std. S | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 |
| f_code | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

## V. 실험 결과 및 고찰

본 논문에서는 카메라을 통해 저장된 브라우징을 통해 재생 및 고화질 속도개선을 위해 사용되는 도구이며, 브라우징에서 비디오 파싱으로부터 얻을 수 있는 정보를 이용하는 방법을 검토한다. 비디오 정보 검색을 쉽게 하기 위한 비디오 자료인 MPEG-2 Video 파일을 브라우징을 통해 재생하며, 이러한 브라우징은 시간적 요소의 두 단계(대충 보기와 자세히 보기)가 있으며, 또한 이러한 두 단계 사이의 동작을 용이하게 한다. 상세한 브라우징은 VCR과 같은 기능, 특히 빨리 감기와 되감기에 기반을 두었다. 이러한 기능은 계산 집약적이지만 내용 기반이 아닌 압축된 소스의 모든 프레임의 압축을 푸는 일이 필요하다. 반면에 대충 보기 위한 브라우징은 압축데이터를 기반으로 한 알고리즘으로부터 얻어진 내용 변화 정보에 기반을 두었다. 또한, 초당 최대 25 Frame(320×240 해상도) 저장 성능을 가지게 되며, 움직임 검출 기능을 개발하여 움직임이 발생할 시에만 저장이 될 수 있도록 함으로서 저장 용량을 최소한으로 줄일 수 있다. 동영상 녹화/재생/관리 기능으로 카메라로부터 입력된 영상들은 디렉토리에 MPEG 파일로 저장되어 재생과 관리에 편리하며, 디지털 기술을 활용하여 모든 화상처리와 화면 전환을 컴퓨터로 처리함으로서 여러 가지 다양한 모드의 기능을 제공하고 뛰어난 화면 Capture기능으로 선명한 화질을 제공한다. 대부분의 시스템을 살펴볼 때 카메라에서 들어온 영상을 비디오 테이프 방식은 화질과 대용량인 면에서 본다면 많은 양의 데이터와 저 화질등을 개선하기 위해 적은 양의 데이터 저장과 화질의 질을 높일 수 있는 측면을 고려하여 본 시스템을 제안하고 구현하였다.

1. 시스템 설계

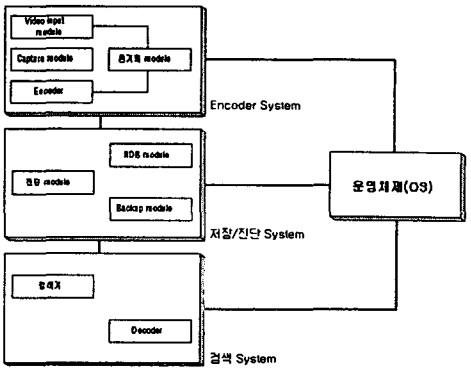


Fig. 5. 시스템 설계 구성도

■ 부호화 시스템

비디오들의 입력을 부호화 시스템을 통해 MPEG DIB (Device Independent Bitmap)으로 변환하여 저장한다.

a) Video Input Module : 카메라를 제어하고 영상을 화면에 출력한다.
b) Capture Module : 카메라의 아날로그 신호를 YUV 4:2:2 형식으로 변환한다.
c) 부호화 Module : YUV 4:2:2 형식의 신호를 MPEG DIB으로 변환한다.
d) 동기화 Module : 각각의 모듈간의 독립적 수행을 위한 동기화 기능을 수행한다.

■ 저장/진단 시스템

부호화 시스템을 통해 생성되는 MPEG 파일을 저장/운용하고, 또한 Backup 기능을 수행하며,. 세부사항을 다음과 같다.

a) RDB(Relational DataBase) Module : 카메라 ID와 그 움직임 그리고 생성시간에 따른 인덱스를 구축하고 각 테이블을 구성하여 관리
b) 진단 Module : 각 카메라의 상태를 항상 감시하며 문제 발생 시에 이를 통보하는 기능
c) Backup Module : DB에 저장된 MPEG파일을 저장 또는 외부 장치로의 이동

■ 검색 시스템

RDB에 의해 생성된 각각의 인덱스에 의해 자료를 검색하고, 검색된 MPEG 정보를 복원하는 기능

a) 검색기 : DB에 저장된 데이터를 인덱스(카메라 ID, 움직임, 생성시간)을 통해 자료를 검색하는 기능
b) 디코더 : 인덱스에 의해 검색된 MPEG 정보를 복원하는 기능

2 구현 결과

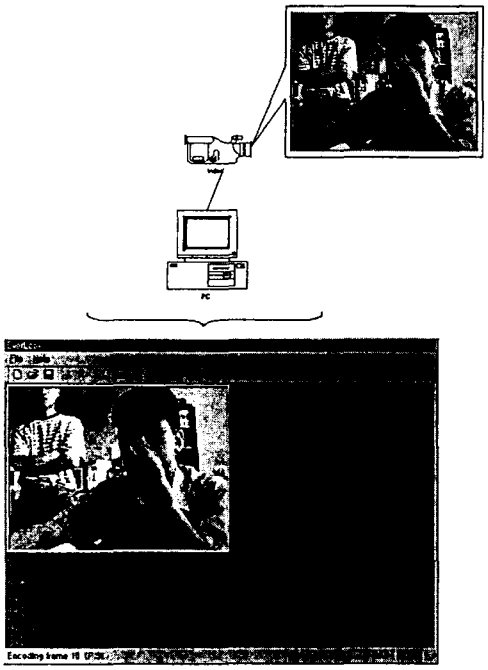

Fig. 6 MPEG-2 Video Browsing

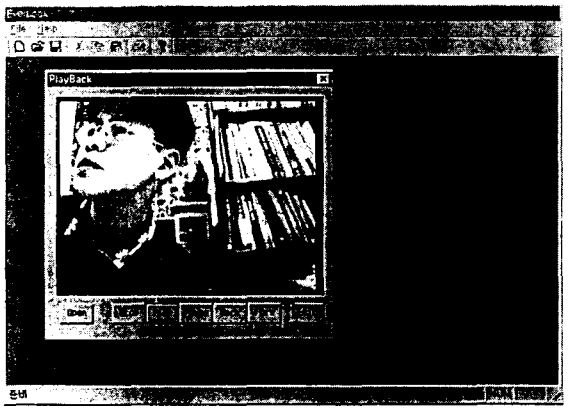

Fig. 7. MPEG-2 Video 재생기능

## Ⅵ. 결 론

본 논문에서 압축된 MPEG-2 Video 스트림의 내용을 하드웨어적인 요소에서 벗어나 실시간 소프트웨어 MPEG 기법을 제안하였다. MPEG 비디오는 샷들로 분해(Parse)되고 그 샷들을 대표하는 각각의 키 프레임들을 선택하여 공간적인 정보와 시간적인 정보로 그 프레임을 인덱스한다.

MPEG-2는 디지털 TV(또는 HDTV)와 같은 방송용을 목표로 제안되었다. 따라서 매우 고화질이며, 초저비트율이다. 그러나 처리량이 많아 고가의 전용 ASIC칩으로 구현되어 있으며, 그나마 아직 보편화되지 않았다. 그러나 본 개발에서는 PC 프로세서의 가격대 성능비가 급격히 상승하고 있는 점에 주목하여 실시간 S/W로 MPEG-2를 구현하였다. MPEG-2는 S/W 구현이 불가능한 것으로 알려져 있으나, 본 연구를 통하여 그 가능성을 입증하였다. 주요한 기술은 MPEG에서 가장 많은 연산을 요구하는 '움직임 추정과 보상'의 알고리즘을 개선하였고, DCT도 실시간 처리가 되도록 하였다. 차후 실제 구현된 브라우저을 통하여 실시간 처리 화상회의 시스템과 기타 감시 작동 시스템에서 용이하게 사용되어질 수 있다.

## [참고문헌]


[1] F. Arman, A. Hsu, and M.Y. Chiu, "Image Processing on Compressed Data for Large Video Databases", Proc. ACM Multimedia 93, Anaheim, CA, pp. 267-272, 1993.

[2] D. Le Gall, "MPEG: A Video Compression Standard for Multimedia Applications", Communications of the ACM, 34(4), pp. 46-58, 1991.

[3] A. Nagasaka and Y. Tanaka, "Automatic Video Indexing and Full-Video Search for Object Appearances", Visual Database Systems II, E. Knuth and L. M. Wegner, editors, North-Holland, pp. 119-133, 1991.

[4] R. Steinmetz, "Data Compression in Multimedia Computing-Standards Systems", Multimedia Systems 1(4), pp. 187-204, 1994.

[5] D. Swanberg, C. F. Shu, and R. Jain, "Knowledge Guided Parsing in Video Databases", Proc. IS&T/SPIE Conf. on Storage and Retrieval for Image Video Databases, San Jose, CA, 1993.

[6] G. K. Wallace, "The JPEG Still Picture Compression Standard", Communications of the ACM, 34(4), pp. 30-44, 1991.

[7] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic Partitioning of Full-motion Video", Multimedia Systems, 1(1), pp. 10-28, 1993.

[8] H. J. Zhang, S. W. Smoliar, "Developing Power Tools for Video Indexing and Retrieval", Proc. IS&T/SPIE Conf. on Storage and Retrieval for Image and Video Databases II, San Jose, CA, pp. 140-149, 1994.