

論文2003-40CI-6-13

DTV PVR에서 HD급 데이터의 실시간 지능형 검색을 위한 알고리즘 및 구현

(Algorithm and Implementation for Real-Time Intelligent Browsing of HD Bitstream in DTV PVR)

鄭壽云*, 張京勳**, 李東濤*

(Su-Woon Jung, Kyung-Hoon Jang, and Dong-Ho Lee)

요약

본 논문은 DTV PVR에 입력되는 HD급 방송 데이터에 대하여 영상의 특성에 따라 지능적으로 검색을 할 수 있는 저 복잡도의 알고리즘과 이의 실시간 구현에 관한 결과를 제시한다. MPEG-2 비디오 데이터에 대해 이를 복호하여 샷을 검출하고 이를 통해 신과 에피소드로 클러스터링하는 효율적인 알고리즘과 각 신의 복잡도를 계산하여 이를 기준으로 비선형적으로 검색하는 알고리즘을 제안한다. 또한 모의실험을 통해 제안하는 알고리즘 성능의 적절성을 검증하였다. 그리고 이의 실시간 구현을 위해 전체 알고리즘을 구현하는 데 있어서 대부분의 연산량을 차지하는 디코더와 기본 정보를 추출하는 부를 하드웨어적으로 구현하고 이를 이용하여 실제 검색을 위한 핵심적이고 확장이 요구되는 알고리즘을 소프트웨어로 구현하는 혼합 구조를 제시하고 실제 이를 구현하였다.

Abstract

This paper presents a low-complexity algorithm for browsing a HD bit stream in DTV PVR according to its characteristics and also presents its implementation results. We propose an efficient algorithm which detects shots using some information after decoding MPEG-2 data, clusters them into scene and episode, and intelligently browses them according to some criteria after calculating their complexity. Some simulation results are presented to show the performance feasibility of the proposed algorithm. To implement it in real time, we propose an efficient hybrid architecture which partitions the algorithm into two parts of hardware and software. The hardware covers decoding process and extraction of some basic information which take most complexity in implementing the algorithm. The software covers the heuristic part of the algorithm which has low complexity and needs to be expandable.

Keywords : PVR, MPEG, 브라우징, DTV, 샷 검출

* 正會員, 漢陽大學校 電子컴퓨터工學部

(School of Electrical and Computer Engineering)

** 正會員, LG電子 DTV 研究所

(LG Electronics, DTV Research Center)

※ 이 연구는 한국과학재단 특정기초연구과제(과제번호 2000-2-303-005-3)로 지원되었음.

接受日字:2003年4月1日, 수정완료일:2003年10月24日

1. 서론

1953년 미국의 National Television System Committee(NTSC)에서 제안되어 아날로그 방식으로 시작한 컬러 텔레비전의 시대는 21세기를 맞아 영화관급의 화질과 CD 수준의 음질을 제공하는 디지털 텔레비전(DTV)으로 대체되어 가고 있다. 이제 텔레비전은 단순

한 방송 시청의 기능 이외에 다양한 디지털 미디어 정보를 검색하고 저장할 수 있는 복합적인 기능이 요구되고 있다.

최근에는 DTV의 보급에 맞추어 기존의 VTR을 대체하는 새로운 저장매체를 이용한 PVR(Personal Video Recorder)이 선 보이고 있다. PVR의 주요 기능은 방송되는 프로그램의 저장과 재생으로, 특히 MPEG-7에서 논의되는 효율적인 검색 기능은 매우 중요한 기능이라 하겠다. 기존의 디지털 비디오 검색 방식은 전체 프로그램을 일정한 시간 간격으로 나누어 검색하거나 임의로 특정부분을 찾아가는 방법이거나, 이미 저장된 프로그램을 내부의 프로세서를 이용하여 비실시간으로 비디오 정보를 모두 복호 및 분석하여 검색에 필요한 정보를 만들어 저장하는 방식이 있다. 첫번째 방식은 사용자가 프로그램을 이해하기 힘들 뿐 아니라 원하는 장면을 찾아가기 위한 검색 과정에 많은 시간이 소요된다. 두 번째 방법은 그 정확도가 뛰어나나 스트림의 복호 처리과정에 많은 시간이 필요하고, 특히 HD 급 영상에 대해서는 실시간 구현이 어렵다.

본 논문에서는 HD 급 영상에 대해 실시간 구현을 고려하여 가능한 최소한의 복잡도를 가지면서 성능이 우수한 지능형 검색 알고리즘을 제안하고, 이를 하드웨어와 소프트웨어로 분할하여 실시간으로 구현하는 혼합 구조를 제안하여 구현하고 DTV PVR에 적용을 하였다. 검색 알고리즘은 먼저 MPEG-2 비디오 스트림을 복호하여 검색에 필요한 기본 정보를 추출하고, 이를 이용하여 장면 전환 단위인 샷(shot)을 찾는 알고리즘을 제안한다. 검출된 샷을 씬(scene)과 프로그램의 의미 있는 이야기 구조인 에피소드(episode)로 모으는 클러스터링(clustering) 방법을 제시하고, 각 씬 마다 화면의 복잡도를 계산하여 움직임이 적은 지루한 씬은 보다 고속으로 진행하고 움직임이 많은 씬 들은 상대적으로 저속으로 검색하는 지능형 검색 방법을 제시한다.

이러한 알고리즘을 구현하는 데 있어서 대부분의 연산은 입력되는 MPEG-2 비디오 데이터를 복호하여 필요한 기본 정보를 추출하는데 소요된다. 따라서 본 논문에서는 제안하는 검색 알고리즘을 입력되는 데이터에 대해 실시간으로 구현하기 위해 대부분의 연산이 요구되는 MPEG-2 비디오 데이터의 복호와 필요한 기본 정보를 추출하는 기능을 하드웨어로 구현을 하고, 이를 이용한 검색 알고리즘은 소프트웨어적으로 나누어 구현하여 실시간 구현이 가능한 구조를 제시하고

실제 구현을 하였다.

II장에서는 지능형 검색을 위한 용어와 검색을 위한 구체적인 방법 등을 설명하고 III장에서는 제안하는 검색 알고리즘에 대한 모의실험 결과를 제시한다. IV장에서는 실시간 구현을 위한 하드웨어 구조와 구현된 결과를 제시하고 V장에서 결론을 맺는다.

II. 지능형 브라우징 알고리즘

1. 용어의 정의

프로그램 중 카메라의 이동이나 다른 내용으로의 전환 등으로 갑작스러운 변화가 일어났을 때 이를 샷(shot)이라 부르며 이는 비디오 구조의 기초적인 단위로 하나의 카메라 장면이 끝날 때까지의 연속적인 장면을 대표하는 영상을 뜻한다. 이 때 한 개의 샷이 대표하는 연속적인 장면의 묶음을 씬(scene)이라 하며 이는 동영상을 분류하는데 있어 가장 작고 기본적인 분류 기준이다. 에피소드(episode)란 의미 있는 이야기 구조를 지니고 있는 단위로 하나의 씬으로 이루어지거나 유사한 두 개 이상의 씬으로 이루어 질 수도 있다. 이런 에피소드를 대표하는 샷을 키프레임(key frame)이라 한다. 이러한 키프레임의 정확한 검출은 브라우징에 있어서 매우 중요하며 각 영상을 씬 이나 에피소드 단위로 묶는 과정을 클러스터링(clustering)이라 한다.

2. 브라우징을 위한 화면 추출요소

기본적으로 각 화면마다 화면의 종류, 사이즈, 구조에 대한 정보가 필요하며 각 화면마다 필요한 정보는 <표 1>과 같다. I 화면은 모든 매크로블록이 움직임 보상이 표현되므로 모든 화면이 부호화 된다. 각 매크로블록을 구성하는 휘도(Y), 색도(Cb,Cr)의 평균 밝기인 DC 값을 이용할 수 있다. P 화면은 순방향 예측을 이

표 1. 각 화면에서 추출되는 정보
Table 1. Information extracted from each picture.

I picture	Y / Cb/ Cr DC coefficient , picture_coding_type , picture_structure, hor_size, ver_size
P picture	horizontal / vertical motion_vector, picture_coding_type, picture_structure, mb_type, fcode_fh, fcode_fv, hor_size, ver_size
B picture	mb_type, picture_coding_type, picture_structure, hor_size, ver_size

용하여 과거의 화면에서 움직임 보상을 하며, 이 화면에서는 각 매크로 블록의 타입과 움직임 보상이 사용된 매크로블록에서의 모션벡터와 움직임 보상의 범위인 f_code 를 추출해 낸다. B 화면은 양방향 예측을 이용하여 과거와 미래의 화면에서 움직임 보상을 한다. 따라서 매크로블록의 타입이 매우 중요한 정보가 되므로 11가지의 매크로블록 타입의 개수를 추출하게 된다.

3. 샷 검출 알고리즘

I 화면에서의 샷 변화를 판별하는데 가장 주요하게 사용되는 것은 두 장의 영상의 주요 DC성분의 변화를 비교하는 것이다⁶⁻⁷. 이때 샷의 신뢰성을 판단하는데 있어 중요한 것은 얼마나 많은 화소값을 비교하느냐에 달려있는데 가장 좋은 방법은 완전히 복호되어 원래의 값으로 복원된 화소를 비교하는 것이나 그 연산량 때문에 대부분은 매크로블록 단위로 휘도 신호 (Y)의 DC만을 비교하는 방법을 사용하며, 데이터를 더욱 줄이기 위하여 Y DC 값의 분포를 히스토그램화 하여 비교한다⁶. 히스토그램화는 데이터를 줄일 수 있는 장점 뿐 아니라 화면의 해상도가 바뀌어도 유연하게 대처할 수 있는 장점이 있다.

Y DC는 16~240의 값을 가지며 매크로블록 당 4개의 값이 나온다. 기존의 알고리즘^{4,6-7}은 매크로블록 당 4개의 값을 하나의 밝기 값으로 평균하며 HD급인 경우 전체 매크로블록에 대한 225개의 히스토그램을 계산하여 시간적으로 가장 인접하는 I 화면의 히스토그램과 비교를 통해 샷을 찾아낸다.

구체적인 방법은 식 (1)과 같은 절대적인 누적 에러인 CDS_A 를 계산하여 임계값보다 크게 되면 그 프레임을 샷으로 결정하는 방식을 택한다⁶.

$$CDS_A = \sum_{j=1}^G |H_i(j) - H_{i+1}(j)| \quad (1)$$

where G: number of gray level(225)
 i: number of frame
 H(j): value of the histogram for the gray level j

본 논문에서는 샷 검출 성능을 높이기 위해 휘도 신호 뿐 아니라 화면의 색상 값(Cb, Cr)도 사용하며, 히스토그램 데이터를 줄이기 위해 <표 2>와 같이 각각의 값을 225개가 아닌 4개로 양자화하여 히스토그램을 구하였다. 이 때 휘도와 색상 값을 각각 4단위로 양자화

하여 구한 3차원 히스토그램은 <그림 1>과 같다. 제안한 알고리즘에서 사용하는 히스토그램의 데이터량은 <표 3>에서와 같이 기존의 방법보다 세 배 이하로 떨어지지만 세가지 화면 값의 사용과 3차원 히스토그램으로 실제 구현한 결과 샷 검출 능력은 오히려 좋아지는 결과를 보였다.

표 2. 화면 밝기와 색상 값의 양자화
 Table 2. Quantization method of luminance and chrominance.

Y, Cb, Cr의 범위	Quantized index
~ < 72	0
71 < ~ < 128	1
127 < ~ < 183	2
182 < ~	3

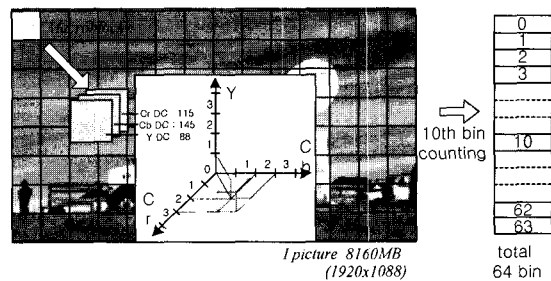


그림 1. Y, Cb, Cr값을 모두 이용한 DC 히스토그램 생성
 Fig. 1. Histogram generation of DC values using Y, Cb, and Cr.

션을 대표하는 샷을 결정할 때 식 (1)의 절대적인 CDS_A 는 씬 자체에 변화가 많고 적응과 같은 화면의 특성에 따라 그 차이가 매우 크게 달라지므로 CDS_A 만으로 샷을 결정하는 것은 좋은 성능을 기대할 수 없다. 따라서 본 논문에서는 CDS_A 뿐 만이 아니라 식 (2)와 같은 이전 CDS_A 와 현재 CDS_A 의 상대적인 비율을 계산한 CDS_R 를 계산하여 두 값 모두가 임계치를 넘어야 샷으로 간주하였다.

$$CDS_R = \frac{|CDS_A(i) - CDS_A(i+1)|}{CDS_A(i+1)} \times 100 (\%) \quad (2)$$

where $CDS_A(i)$: value of i_{th} CDS_A

P 화면이나 B 화면에서 샷이 일어났을 때에는 <표 4>와 같이 영상의 매크로블록 타입의 분포에 의하여

샷 여부를 결정한다. P 화면에서는 생략된 매크로블록 (skipped macroblock)를 제외한 전체 매크로 블록에 대한 인트라 매크로 블록의 비율을 참조하며, B 화면에서는 순방향/역방향 매크로블록의 비율을 이용하여 샷을 검출한다^{6, 10)}.

표 3. I 화면에서의 데이터 양의 비교(MP@HL급 기준)
Table 3. Comparisons of picture data at I frame.

	기존 알고리즘	제안 알고리즘
사용 DC	Only luminance(Y)	luminance(Y), chrominance(Cb, Cr)
Bin 개수	225 bin(16~240)	64bin(4×4×4)
정보량	3,600 bit	1,024 bit

표 4. 매크로 블록 타입의 양자화
Table 4. Quantization of macroblock type.

MB type 히스토그램 bin	P picture	B picture
10	--	Intra, quant
9	--	Backward, coded, quant
8	--	Forward, coded, quant
7	--	Interpolated, coded, quant
6	Intra, quant	Intra
5	No MC, coded, quant	Forward, coded
4	MC, coded, quant	Forward, not coded
3	Intra	Backward, coded
2	MC, not coded	Backward, not coded
1	No MC, coded	Interpolated, coded
0	MC, coded	Interpolated, not coded

4. 씬과 에피소드의 결정

본 논문에서 씬과 에피소드의 결정은 <표 5>에서와 같이 CDS_A와 CDS_R의 임계치에 따라 결정하였다. 임계치를 결정하는 정확한 기준은 없고 다양한 영상에 대한 실험을 통해 결정하였다.

표 5. 씬과 에피소드 결정의 임계치
Table 5. Threshold values for deciding scene and episode.

	씬(Scene)	에피소드(Episode)
CDS _A	3,000 이상	8,000 이상
CDS _R	40% 이상	80% 이상

5. 지능형 브라우징 속도 결정 알고리즘

본 논문에서는 I화면과비교적 추출이 간단한P/B 화면의 모션벡터와 매크로 블록 타입을 이용하여 화면의 복잡도를 결정하고 이를 바탕으로 화면의 복잡도가 높은 구간은 브라우징 속도를 느리게 하고, 화면의 복잡도가 낮은 부분은 브라우징 속도를 빠르게 검색할 수 있게 하였다. 이때 모션 벡터 역시 양자화 후 히스토그램화 하는데 이는 모션 벡터가 절대적인 크기 보다는 해상도에 따른 상대적 크기로 결정되기 때문이고 데이터의 감소를 역시 크기 때문이다.

5.1 화면 복잡도 결정

MPEG-2 비디오 규정에서 모션 벡터는 화면의 상관도를 나타내는 중요한 척도이다. 모션벡터가 클수록 이전화면에서 먼 곳의 매크로 블록을 참조하는 것이며, 그 만큼 화면의 변화가 크고 복잡도가 높다는 사실을 이용한 것이다.

모션벡터는 화면의 해상도에 따라 영향을 많이 받는 f_{code}에 따라서 그 값의 범위가 크게 달라지므로, 크기에 대한 절대적인 양자화로는 비효율적인 작업이 된다. 즉 해상도가 704×480일 때의 모션 벡터 크기와 1920×1088일 때의 모션 벡터 크기가 그 절대적 크기가 같다고 하더라도 같은 벡터 크기라면 실제 복잡도는 해상도 704×480일 때가 더 높은 것이다. 따라서 f_{code}에 따른 상대적인 양자화가 반드시 필요하다. <표 6>과 같이 f_{code}에 따라 서로 다른 양자화 계수로 모션 벡터를 나누면 모션 벡터 값은 수직, 수평 좌표로 각각 8개의 범위만 가지게 되어 쉽게 이용할 수 있게 된다. 예를 들어 f_{code}가 1 인 경우에 실제 나올 수 있는 모션 벡터는 8에서 7 까지이며, 이 때는 각 레벨의 양자화 단위를 2로 하여 8구간으로 나눌 수 있다. 만약 f_{code}가 2가 되면 모션 벡터는 16에서 15까지 32까지 변화할 수 있으므로 양자화 단위를 4로 하여 8구간으로 나누면 상대적인 위치 변화정도에 따라 언제나 일정하게 양자화 할 수 있다.

이렇게 1차적으로 양자화 된 수평 모션 벡터와 수직 모션벡터를 x, y축으로 각각 8단위가 되도록 양자화 하게 되면 64개의 bin을 가진 2차원히스토그램이 <그림 2>와 같이 만들어 지며, 이를 이용하여 해당 P 화면의 복잡도를 결정한다. P 화면의 나머지 매크로 블록 중 인트라 매크로 블록은 이전 화면과 상관성이 없는 것으로 간주하고 가장 높은 복잡도를 부여하며, 생략된

매크로 블록이나 움직임 보상을 사용하지 않은 매크로 블록은 이전 화면과의 상관성이 큰 것으로 간주하여 최소 복잡도를 부여한다. B 화면의 경우 움직임 보상이 사용되는 매크로 블록과 인트라 매크로 블록으로 나누어 P 화면과 같은 기준으로 그 복잡도를 판단한다.

5.2 화면 복잡도 계산

복잡도의 계산은 매크로블록에 따라 복잡도(1~7)를 두어 각 화면마다 정해진다. 공통적으로 인트라 매크로 블록은 복잡도를 7로 두고 반면에 움직임 보상이 일어나지 않았거나 생략된 매크로 블록에 대하여서는 가중치를 1로 둔다. I 화면은 모든 화소가 부호화 되었기 때문에 모든 매크로 블록의 복잡도를 7로 두고, P 화면은 모션 벡터 크기에 따른 복잡도를 <그림 4>와 같이 정한다. 마지막으로 B 화면에서 움직임 보상이 사용된 매크로 블록은 복잡도를 3으로 고정한다.

표 6. f_code의 범위에 따른 MV의 이용

Table 6. MV usage according to the range of f_code.

F_code	모션벡터탐색 범위	허용가능한모션벡터범위	양자화 계수
1	-8~7.5	16	2
2	-16~15.5	32	4
3	-32~31.5	64	8
4	-64~63.5	128	16
5	-128~127.5	256	32
6	-256~255.5	512	64
7	-512~511.5	1024	128
8	-1024~1023.5	2048	256
9	-2048~2047.5	4096	512

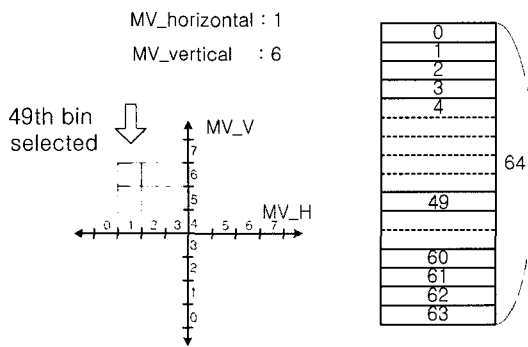


그림 2. 모션 벡터의 히스토그램화
Fig. 2. Histogram of motion vectors.

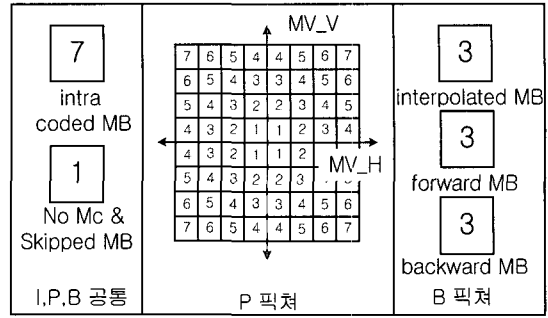


그림 3. 각 화면에서 화면의 복잡도 계수
Fig. 3. Complexity values in each picture

브라우징 속도는 각 씬에 해당하는 화면의 복잡도를 모두 평균하여 정해지게 되는데 이 때 각 씬의 길이 또한 브라우징 속도에 영향을 준다. 같은 복잡도를 가진다 하더라도 씬의 길이가 길수록 속도를 더 빠르게 해주게 된다.

III. 모의 실험

<그림 4>에는 실제 DTV를 통해 수신된 뉴스 프로그램에 대하여 앞에서 설명한 CDSA와 CDSR의 임계치를 이용하여 검출된 샷 단위의 영상을 나타내었다. 최종적으로 5개의 에피소드가 결정되었고, 각 에피소드들은 3~5개의 씬으로 이루어진다. 따라서 본 논문에서 제안하는 샷 검출 알고리즘을 통해 지능형 검색을 위한 샷 단위의 영상 분할이 이루어져 있음을 확인할 수 있다.

<표 7>은 3,800여장으로 구성된 방송된 축구경기에 대한 복잡도를 계산하여 그 브라우징 속도를 4단계로 나누어 결정하는 과정을 나타내었다. 우선 샷 검출 기법을 이용하여 프로그램을 씬 단위로 나눈 후 구간별로 속도를 결정한다. <그림 5>에는 실제 에피소드에 적용한 예로, 이때 브라우징 속도는 고속전진 속도에 대한 상대적인 값을 나타내었다. 그림을 통해 실제 영상의 복잡도와 움직임 정도에 따라 브라우징 속도가 결정되었음을 알 수 있다. 5명을 대상으로 다양한 배속에 대해 일반 선형적인 브라우징과 본 논문에서 제안하는 지능형 브라우징을 비교하는 실험을 하였다. 5명 모두 같은 배속에서 제안하는 지능형 브라우징이 선형 브라우징에 비해 영상의 구체적인 내용을 이해할 수 있었다고 평가하였다. 특히 20 배속 이상에서는 선형 배속에서는 영상의 내용을 거의 이해할 수 없었지만

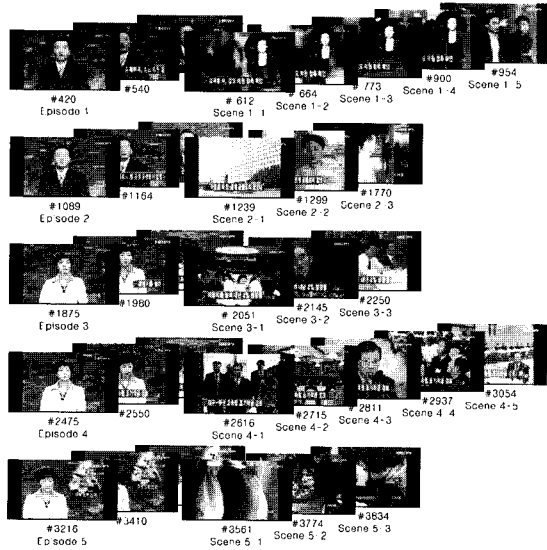


그림 4. 뉴스 프로그램에 대한 에피소드 단위의 결정
Fig. 4. Episode results of a news program.

표 7. 구간별 브라우징 속도 결정
Table 7. Decision of browsing speed for football program.

Scene	구간	구간 길이	평균 복잡도	브라우징 속도	비고
# 1	0~196	197	11.21	× 3	골 다툼
# 2	197~239	43	10.20	× 3	심판에 어필
# 3	240~579	340	10.76	× 3	운동장전경 (정지)
# 4	580~924	345	14.81	× 2	심판에 어필
# 5	925~1065	141	10.74	× 3	로스 타임 개시
# 6	1066~1344	279	8.90	× 3	골키퍼의 골킥
# 7	1345~1530	186	14.55	× 2	운동장전경 (움직임)
# 8	1531~1650	120	9.47	× 3	드로잉
# 9	1651~1809	159	14.77	× 2	운동장전경 (움직임)
# 10	1810~1869	61	10.99	× 3	골키퍼의 골킥
# 11	1870~1948	79	8.94	× 3	심판의 이동
# 12	1949~2451	503	11.09	× 3	운동장전경 (움직임)
# 13	2452~2667	216	11.02	× 3	코너킥 준비
# 14	2668~2907	240	7.89	× 4	운동장전경 (정지)
# 15	2908~3183	276	13.42	× 2	코너킥 동작
# 16	3184~3567	384	11.92	× 2	운동장전경 (움직임)
# 17	3568~3648	81	17.11	× 1	코너킥, 골대 앞흔전
# 18	3649~3816	167	10.59	× 3	운동장전경 (움직임)

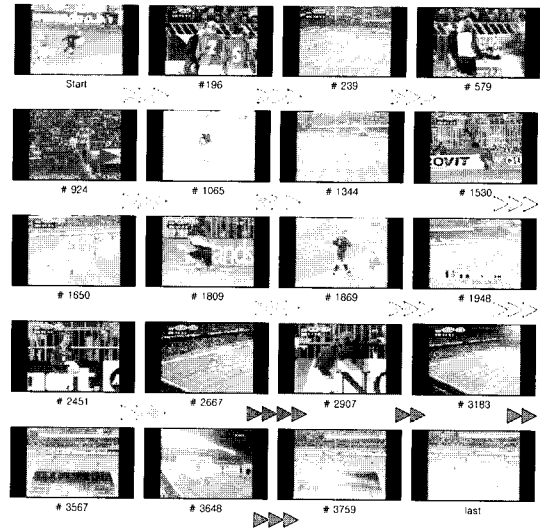


그림 5. 실제 브라우징 속도의 결정
Fig. 5. Browsing speed of each scene for football program.

제안하는 지능형 배속에서는 대부분의 영상 내용을 이해할 수 있었다.

IV. 실시간 구현

앞장에서 소개한 지능형 검색 알고리즘을 실시간으로 구현하기 위하여 본 논문에서는 이를 하드웨어와 소프트웨어로 분리하여 구현하는 혼합 구조를 적용하였다. 하드웨어부에서는 샷을 검출하고 검색 속도를 결정하기 위하여 필요한 각 화면의 히스토그램을 실시간으로 계산하는 기능을 담당하고, 소프트웨어부에서는 이러한 기본 데이터를 입력하여 샷을 검출하여 최종 브라우징을 제어하는 알고리즘을 담당하였다. 이러한 개념은 전체 알고리즘을 구현하는데 있어서 대부분의 연산이 하드웨어가 담당하는 히스토그램을 추출하는데 몰려있고, 이렇게 함으로서 소프트웨어의 수정을 통해 알고리즘의 성능을 개선시킬 여지가 있기 때문이다.

PCI 버스 구조를 갖는 PVR과 인터페이스를 포함하는 하드웨어부의 전체 구조는 <그림 6>과 같다. VSB (Vestigial Side Band) 인터페이스를 통해 직접 DTV 방송을 수신하거나 일반 NTSC방송을 MPEG-2 엔코더를 통해 입력을 받아 사용자가 원하는 프로그램을 PID 필터링을 한 후 PCI 버스를 통해 하드 디스크에 저장하도록 하였다. 또한 하드디스크에 저장된 데이터는 PCI 버스를 통해 다시 MPEG-2 디코더로 출력을 하게

되는데, 이는 외부에서 입력되는 데이터를 저장하지 않고 바로 출력할 수도 있다. <그림 7>과 같이 브라우징에 해당하는 정보를 만드는 부분은 크게 시퀀스와 화면 헤더 정보를 찾아내는 PTD (Picture Type Detection) 블록과 화면 정보를 히스토그램 형태로 만들어 주는 HGD(Histogram Generate Decoder) 블록으로 나누어진다. PTD는 저장 매체에 저장된 비디오 데이터에 대해 화면 단위로 검색을 가능하게 하는 인덱스 정보를 실시간으로 추출하게 되며, 최대 27 Mps의 데이터를 처리할 수 있으며, picture coding type, frame rate 등의 정보를 출력 한다. HGD는 화면 단위로 정보를 히스토그램화 하여 256x16 크기의 메모리 두개를 이용하여 PCI 버스를 통해 CPU로 출력하게 된다. CPU에서는 이러한 데이터를 실시간으로 입력하여 샷을 검출하고 브라우징 속도에 관한 정보를 인덱스에 포함시켜 하드디스크에 저장하게 된다.

HGD의 입력은 최대 27 Mbps 까지의 속도로 들어오나 MPEG-2 디코더의 특성상 27 MHz 클럭으로 동작하고 출력 역시 27MHz에 동기화되어 출력된다. 주요 기능은 MPEG-2 비디오를 화면별로 복호 하여 히스토그램 형태로 내부 메모리에 저장한 후 데이터 전송 준비 신호를 보내면 local controller의 요구에 따라 출력한다. 디코딩의 목적이 완전한 영상복원이 아니고 일부 데이터를 추출하기 위함이기 때문에 버퍼를 이용한 버퍼 컨트롤은 하지 않는다. 대신에 MPEG 데이터의 복호가 가능하도록 레지스터는 27MHz 한 클럭마다 한 비트씩 쉬프트 시키며 최대 코드 길이인 32 비트를 유지하며, 복호 된 비트 만큼 shift count를 주어 항상 다음 부호를 복호 할 수 있는 상태를 유지토록 한다. 각 화면마다 사용되는 정보가 다르기 때문에 슬라이스 단

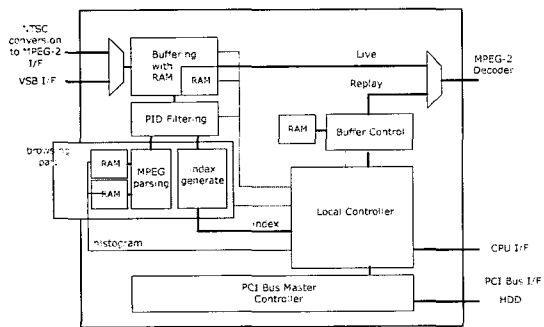


그림 6. PVR에서 브라우징을 포함하는 하드웨어 구조
Fig. 6. Hardware architecture including browsing in PVR.

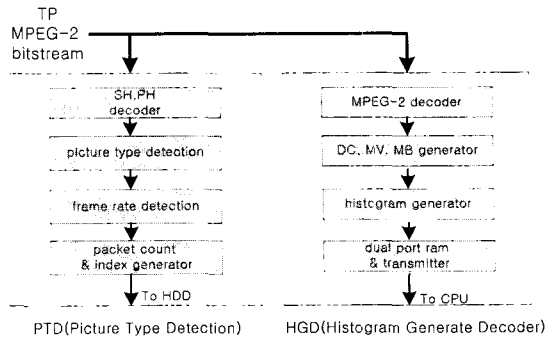


그림 7. 브라우징을 위한 하드웨어 구성도
Fig. 7. Two hardware parts for browsing.

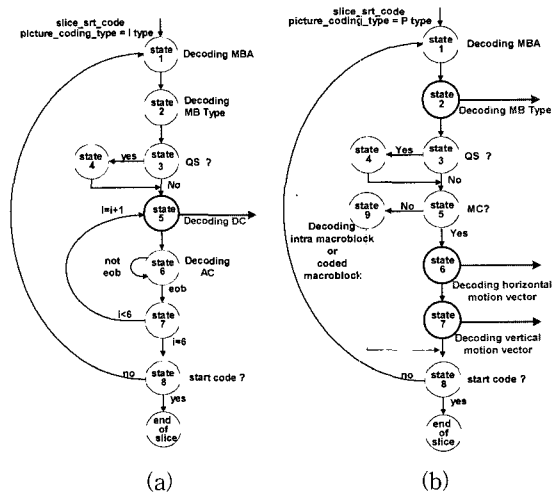


그림 8. MPEG-2 데이터의 디코딩을 위한 상태 천이도: (a) I 화면, (b) P/B 화면
Fig. 8. State diagram for decoding MPEG-2 bitstream: (a) I picture and (b) P/B picture.

의 복호 부분은 로직의 최소화와 속도의 향상을 위하여 각 화면별로 병렬로 동작된다. I 화면에서는 DC 값을 끌어내기 위하여 DC 계수와 AC 계수를 해석하기 위한 부분을 위주로 설계되었고 P/B 화면에서는 모션 벡터 추출을 위하여 매크로 블록 정보를 복호 하는 부분을 위주로 설계하였다. 이러한 하드웨어는 VHDL을 이용하여 설계되었으며, <그림 8>에는 각 화면 별로 주요동작이 다른 주요 상태 천이도를 나타내었다.

<그림 9>에는 메모리와 PCI 버스와 의 구체적인 인터페이스를 나타내었다.

설계 언어로는 IEEE 표준인 VHDL을 이용하였으며, Hynix 0.35 mm, 3.3V 공정을 이용하여 합성하였다. Dynamic Timing analysis는 Synopsys의 Static timing

Analysis tool인 Prime Time을 이용하여 검증하였다. <그림 6>에 나타난 하드웨어를 구현한 칩과 이를 이용하여 실제 DTV PVR에 적용한 PCI 버스 인터페이스를 갖는 보드는 <그림 10>에 나타내었다. 구현된 칩의 패키징은 304핀 PQFP이고 전체 게이트 수는 약 240,000 정도이다.

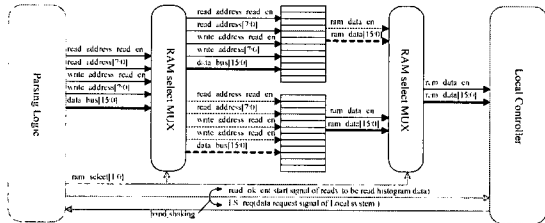


그림 9. 메모리를 통한 디코딩(HDG)과 PCI 버스의 인터페이스도

Fig. 9. Interface between HDG and PCI bus using memory.

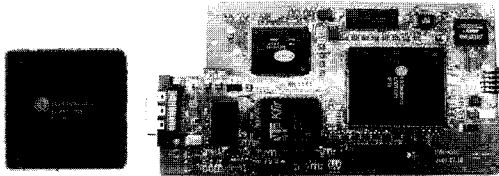


그림 10. 실제 제작된 ASIC 칩과 이를 이용하여 구현된 보드

Fig. 10. Implemented ASIC chip and board.

V. 결론

본 논문에서는 DTV용 PVR에서 지능형 브라우징을 실시간으로 구현하기 위한 저 복잡도의 알고리즘을 제시하고, 하드웨어와 소프트웨어로 나누어 실시간으로 구현하는 혼합 구조를 제시하고 구현하였다. 기존의 선형적인 아날로그 검색 방식의 한계를 해결하기 위하여, 본 논문에서는 카메라의 장면 전환 단위인 샷을 찾고 이를 프로그램의 의미 있는 이야기 구조인 씬으로 클러스터링을 하여 각 화면에 인덱스 정보와 화면의 복잡도에 따른 탐색 속도 정보를 인덱스로 추가시켜 사용자가 하드디스크에 저장된 비디오 데이터를 보다 빠르고 효율적인 검색을 가능하게 하는 방법을 제안하고 모의실험을 통해 성능의 적절성을 확인하였다. 입력되는 MPEG-2 비디오 데이터를 복호하여 필요한 히스토그램 정보를 생성하는 기능을 하드웨어 로직으로 구현

함으로써, CPU의 과도한 부하 문제를 해결하고 HD 급 영상에 대해 실시간 구현이 가능한 구조를 제시하고 실제 ASIC 칩으로 구현하여 DTV PVR에 적용하였다. 또한 하드웨어 로직에서 출력하는 히스토그램 정보를 이용하여 소프트웨어적으로 구현되는 지능형 브라우징에 필요한 알고리즘은 확장이 가능하기 때문에 이에 대한 지속적인 연구가 필요하리 하겠다.

참고 문헌

- [1] A. Hanjalic, R. L. Lagendijk, and J. Biemond, "Automated High-level Movie Segmentation for Advanced Video-Retrieval Systems", *IEEE transaction on circuits and systems for video technology*, vol. 9, no. 4, pp. 580-588, June 1999.
- [2] Minerva M. Yeung, Boon-Lock Yeo, "Video Visualization for Compact Presentation and Fast Browsing of Pictorial Content", *IEEE transaction on circuits and systems for video technology*, VOL.7, No 5, pp. 771-785, Oct. 1997.
- [3] Minerva Yeung, Boon-Lock Yeo, Bede Liu, "Extracting Story Units from Long Programs for Video Browsing and Navigation", *Proceedings of multimedia*, pp. 296305, 1996.
- [4] J.Berens, G.D. Finlayson and G.Qiu, "Image indexing using compressed colour histograms", *IEE Proc.-Vis. Image Signal Process*, vol. 147, no 4, pp. 349-355, August 2000.
- [5] Rick Rickman, Paul Rosin, "Content Based Image Retrieval using Colour N-grams", *The Institution of Electrical Engineers*, pp. 15/1-15/6, 1996.
- [6] W.A.C Fernando, C.N. Cangarajah and D.R. Bull, "Scene change detection algorithms for content-based video indexing and retrieval", *Electronics & Communication Engineering Journal*, pp. 117-126, June 2001.
- [7] Cuneyt Taskiran, Jau-Yuen Chen, Charles A. Bouman and Edward J.Delp, "A compressed Video Database Structured for Active Browsing and Search", *IEEE*, vol. 3, pp. 133-137, 1998.
- [8] AjayDivakaran, Anthony Vetro, Kohtaro Asai,

- Hirofumi Nishikawa, "Video Browsing System Based on Compressed Domain Feature Extraction", IEEE, pp. 637-644, 2000.
- [9] Ullas Gargi, Rangachar Kasturi, and Susan H. Strayer, "Performance Characterization of Video-Shot-Change Detection Method", IEEE Transactions on Circuits & System for video technology, vol. 10, no. 1, pp. 1-13, February 2000.
- [10] Sung-Bae Jun, Kyoungro Yoon, Hee-Youn Lee, "Dissolve Transition Detection Algorithm Using Spatio-Temporal Distribution of MPEG Macro-Block Types", ACM, 2000.
- [11] ISO/IEC 13818-2 MPEG-2, "Generic coding of moving pictures and associated audio information: Video", July 1995.
- [12] ISO/IEC 13818-2 MPEG-2, "Generic coding of moving pictures and associated audio information: System", July 1995.

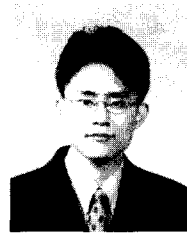
저 자 소 개



鄭壽云(正會員)

1996년 2월 : 한양대학교 공학대학 제어계측공학과 학사. 1998년 2월 : 한양대학교 제어계측공학과 석사. 현재 : 한양대학교 전자전기 제어공학과 박사과정 재학중. <주관심분야 : 멀티미디어 시스템, VLSI

설계, 버스 기반의 디지털 시스템 등.>



張京勳(正會員)

1999년 2월 : 한양대학교 공학대학 제어계측공학과 학사. 2001년 2월 : 한양대학교 제어계측공학과 석사. 현재 : LG전자 DTV 연구소 근무. <주관심분야 : 영상 처리 및 압축, VLSI 설계, DTV 등.>

李東濬(正會員) 第39卷 SP編 第4號 參照