

VC-1용 고속 역양자화 및 역변환 모듈 구현

Implementation of Fast Inverse Quantization and Inverse Transform Module for VC-1

김경현, 송형돈, 손승일
한신대학교 정보통신학과

Kyung Hyun Kim, Hyung Don Song, Seung Il Sohn
Dept. of Information and Communication HanShin
University

요약

Abstract

최근 영상을 중심으로 여러 형태의 정보를 결합하여 저장하거나 전송하는 멀티미디어가 많은 관심을 받고 있다. 현재 카메라와 관련된 동영상 캡처기술은 Motion JPEG이 주류를 이루고 있으며, 텔레비전, DMB 등의 방송 분야 및 DVD, VCR 분야에서는 MPEG-2, MPEG-4, H.264 및 WMV9 등의 압축 코덱이 채용되고 사용되고 있다. 그러나 이러한 다양한 영상 표준 방식은 디코딩시 호환성 문제가 발생하게 되고 이에 따라 통합 코덱 연구가 필요하다. 이에 본 논문은 일반적 스텝 양자화 외에 데드존 양자화를 사용하고 「4×4」, 「4×8」, 「8×4」, 「8×8」의 다양한 블록크기의 변환을 지원하는 VC-1을 기반으로 한 ITIQ C언어를 통해 시뮬레이션하고 최적화된 결과를 VHDL로 구현하여 향후 통합코덱 연구에 응용 가능하도록 연구 및 분석평가 하였다.

설계결과 4:2:0의 YCbCr포맷의 최초 16x16블록을 복원하는데 483~510클럭이 소요되었고 Xilinx XCVPC100 FF1696-6 환경에서 93,128개의 게이트 수와 71.469MHz의 동작속도를 나타내었다. 이는 640*480 크기의 컬러영상을 디코딩 하는데 프레임 당 최대 0.0074초가 소요됨을 의미하며 초당 30프레임의 영상에서도 0.222초면 디코딩이 가능한 결과이다.

I. 서론

최근 영상을 중심으로 여러 형태의 정보를 결합하여 저장하거나 전송하는 멀티미디어가 많은 관심을 받고 있으며 제한된 채널과 데이터양, 처리속도에 대한 문제점들이 제기되어 영상 압축에 대한 관심이 높아졌다. 영상 압축은 디지털 TV, 인터넷 스트림 비디오 그리고 DVD-비디오 같은 영상 매체에서 핵심 요소가 되고 있다.

일반적으로 산업표준 코덱으로 MPEG-2, MPEG-4, H.264/AVC 와 AVS, VC-1이 있고, 현재 개발중인 표준으로는 ITU/MPEG Joint Video Coding(H.264/AVC의 개정안)와 MPEG Multi-view Video Coding이 있다.

VC-1에서 마이크로소프트사 고유의 알고리즘으로 개발된 Windows Media Video9(WMV9)은 2003년 9월에 마이크로소프트사가 SMPTE(Society for Motion Picture and Television Engineers)에 WMV9 비트 스트림 및 구문을 표준화할 것을 제안하였으며, 이 제안이 받아들여져서 VC-1으로 SMPTE에 의해 표준화되었다.

동영상을 압축하는 알고리즘은 현재 지속적으로 발전하고 있다. 새로운 압축 표준들이 발표되고 있으며, 발표된 각각의 압축 코덱은 응용 분야에 따른 장점을 가지고 있는 것으로 판단된다. 현재 카메라와 관련된 동영상 캡처기술은 Motion JPEG이 주류를 이루고 있으며, 텔레비전, DMB 등의 방송 분

야 및 DVD, VCR 분야에서는 MPEG-2, MPEG-4, H.264 및 WMV9 등의 압축 코덱이 채용되고 사용되고 있다. 그러나 이러한 다양한 영상 표준 방식은 디코딩시 호환성 문제가 발생하게 되고 이에 따라 통합 코덱 연구가 필요하다[1].

II. VC-1

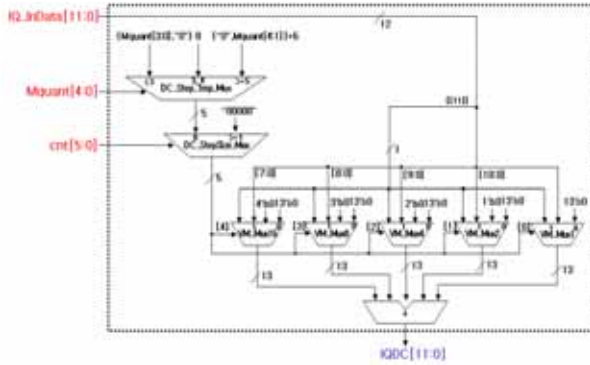
여러 가지 영상 표준 코덱 중 VC-1은 본 논문에서 다루게 될 양자화의 경우 일반적 스텝 크기 기반 양자화 및 데드존 양자화를 모두 이용한다. 데드존 양자화를 이용함으로써 비트율이 낮을 때 상당한 절약이 가능하면 페이딩을 포함하는 장면의 명시적 페이딩 보정을 이용할 수 있는 특징을 통해 화면에서 움직임 보정 품질을 향상시킨다. 또한 「4×4」, 「4×8」, 「8×4」, 「8×8」의 4종류의 다양한 이산 코사인 변환(DCT : Discrete Cosine Transform) 블록사이즈 지원으로 주기적으로 변화하는 입상감을 남겨 필름의 질감을 표현하는데 유리하다. 또한, 비디오 코덱의 낮은 복잡성은 효율적이며 하드웨어로 구현 시 적은 생산 비용과 낮은 소비전력, 적은 발열이 가능하다. VC-1과 H.264/AVC는 MPEG2보다 더 복잡하지만 VC-1의 디코더는 H.264/AVC보다 복잡성이 낮으면서 유사한 압축 효율과 품질을 제공한다.

이러한 이유에서 본 논문에서는 VC-1을 기반으로 통합코덱

에 사용 가능한 ITIQ에 대하여 연구 하였다[2].

III. 역양자화

3.1 DC 역양자화



▶▶ 그림 1. DC 역 양자화

그림 1은 DC 역 양자화를 하는 모듈로 본 논문은 데이터 부호 유지하며 비트확장 및 덧셈만을 통하여 곱셈연산과 같은 결과를 얻을 수 있도록 설계하였다.

$$\begin{aligned}
 DCStepSize &= 2 * MQUANT \quad [MQUANT=1 \text{ or } 2] \\
 &= 8 \quad [MQUANT=3 \text{ or } 4] \\
 &= MQUANT/2 + 6 \quad [MQUANT \geq 5] \quad (1) \\
 DCCoefficient &= DCCoeffQ * DCStepSize \quad (2)
 \end{aligned}$$

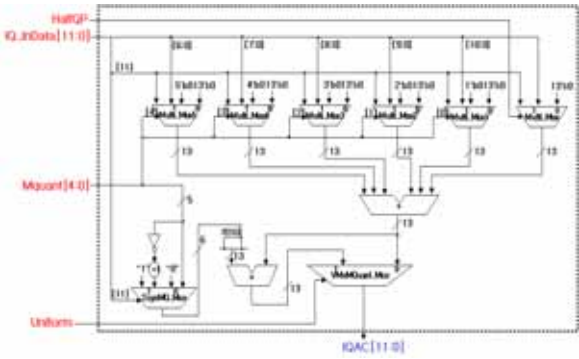
DC 역 양자화는 인트라 모드의 디코딩에서만 사용되는 양자화 방식으로 (1)과 같이 매크로블록의 양자화 스케일 값 MQUANT에 따라서 스텝 사이즈가 결정되고 이 값과 DC Predictor를 통해 연산되어진 DCCoeffQ 값과 곱셈연산을 통하여 (2)와 같은 DC 역 양자화 계수 값이 결정되어 진다 [3][4].

3.2 AC 역양자화

$$\begin{aligned}
 ACCoefficient &= ACCoeffQ * (2 * MQuant + HalpQp) \quad [Uniform] \\
 &= ACCoeffQ * (2 * MQuant + HalpQp) \\
 &\quad + (MQuant * SignMQ) \quad [Nonuniform] \quad (3)
 \end{aligned}$$

AC 역 양자화는 AC Predictor 통해 연산되어진 ACCoeffQ는 인트라 모드의 경우 최초 DC값 이후의 모든 연산 데이터를 「8×8」 블록사이즈에 대하여 Normal, Horizontal, Vertical의 인코딩 방식중 택일하여 에 따라 처리하게 되고, 인트라 방식의 데이터는 Normal 인코딩 방식으로 「4×4」,

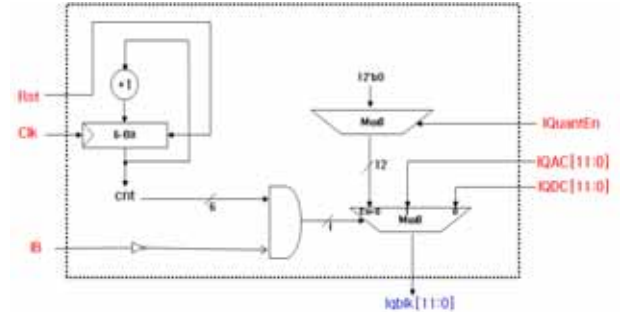
「4×8」, 「8×4」, 「8×8」의 블록 사이즈에 따른 인코딩 방식에 따라 처리하게 된다. 이때 Picture Quantizer Index값에 의해 Uniform / Nonuniform 양자화 방식이 결정되고 (3)과 같이 AC 역 양자화 계수값이 결정되어 진다[5][6][7].



▶▶ 그림 2. AC 역 양자화

그림 2는 AC 역양자화를 하는 모듈로 DC 역양자화에서 추가적으로 MQUANT 및 Uniform의 제어 신호부가 추가 설계 하였다[6][7].

3.3 역양자화 처리 선택기



▶▶ 그림 3. 역양자화 처리 선택기

그림 3은 역양자화 처리 선택기로서 인트라 또는 인트라 블록 정보 및 AC/DC성분을 판단 선택적으로 데이터를 처리 및 출력시킨다[8].

IV. 역변환

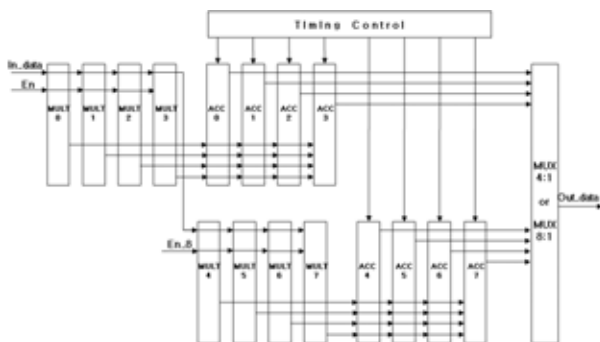
4.1 변형된 역변환

$$\begin{aligned}
 E_{M*N} &= ((D_{M*N} * T_M) + 4) \gg 3 \\
 \Rightarrow R_{M*N} &= ((T'_N * E_{M*N}) + (C_N * 1_M) + 64) \gg 7 \quad (4) \\
 E_{M*N} &= ((D_{M*N} * T_M) / 4 + 1) / 2 \\
 \Rightarrow R_{M*N} &= (((E'_{M*N} * T_N) + (C_N * 1_M)) / 64 + 1) / 2 \quad (5)
 \end{aligned}$$

역변환에서 가장 많은 연산을 필요로 하는 부분은 수평, 수

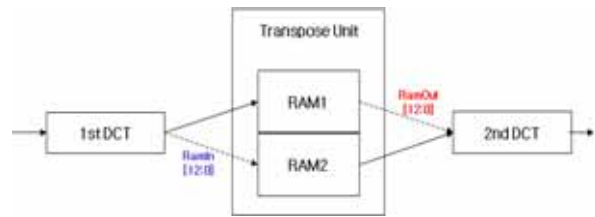
직 방향으로 역변환을 2번 수행하는 단계에 집중되어 있다. 이러한 역변환의 연산 대부분은 행렬곱셈으로 이루어져 있다. 이에 본 논문은 연산량이 집중되어 있는 행렬곱셈을 최적화시킴으로서 고속의 역변환을 수행할 수 있도록 하였다. (4)는 $M \times N$ 사이즈의 입력 블록 크기를 가질때 입력 데이터(D), 변환행렬 계수(T), 열벡터 계수(C)를 가질 때 $E_{M \times N}$ 은 수평방향의 IDCT 결과를 유도해 내고 $R_{M \times N}$ 은 수직방향의 IDCT 결과를 유도하여 최종적인 변환 데이터를 얻게 되는 일반적인 VC-1 역변환 유도방식이다. 이를 소프트웨어로 구현하는데 있어 (4)는 별다른 제약이나 문제점은 없지만 하드웨어로 구현하는데 있어 수평의 IDCT를 구하는 $R_{M \times N}$ 의 행렬곱셈 연산은 $E_{M \times N}$ 에서 연산되어지는 행렬 곱셈연산과 같은 방식임에도 고정된 변환행렬 계수 T 값을 재사용하는데 있어 함수의 규칙성에 대한 문제점이 발생한다. 이에 본 연구에서는 (5)와 같이 변형하여 연산함으로써 $E_{M \times N}$ 과 $R_{M \times N}$ 을 연산하는 행렬곱셈 연산을 공통된 하나의 모듈로 재사용 할 수 있게 함으로써 불필요 설계 면적을 줄인다. 또한 나눗셈연산의 경우 데이터의 쉬프트 연산만을 통해서 얻을 수 있음을 이용 IDCT연산유도 과정에서 부분적 나눗셈을 함으로써 이후 덧셈연산의 값을 모두 '1'로 줄임으로써 불필요 덧셈연산을 줄였다[9][10].

4.2 고속 역변환



▶▶ 그림 4. 고속 행렬 곱셈

그림 4는 행렬 곱셈을 하는데 있어 블록크기가 8×8 일 경우 ($D8 \times 8 \cdot T8$) 또는 ($E'8 \times 8 \cdot T8$)의 전체 행렬 곱셈연산 시 하나의 $D8 \times 8$, $E'8 \times 8$ 데이터가 8개의 T값과 곱셈 연산이 이루어져야 한다. 이를 위하여 본 논문은 그림 4와 같이 기본적으로 하나의 IDCT에 8개의 행렬곱셈 모듈을 병렬 처리함으로써 최초 8클럭의 지연시간 이후에는 순차적으로 매 클럭마다 데이터를 얻음으로써 고속의 행렬 곱셈 연산이 가능하다. 이때 만약 입력 블록이 4×4 크기일 경우 총 8개중 4개의 행렬곱셈기가 불필요 연산을 수행하게 되지 않도록 별도의 제어신호를 통해 최초 4클럭 지연 이후에는 행렬곱셈 결과가 나오는 방식의 고속 역변환 모듈을 설계하였다.

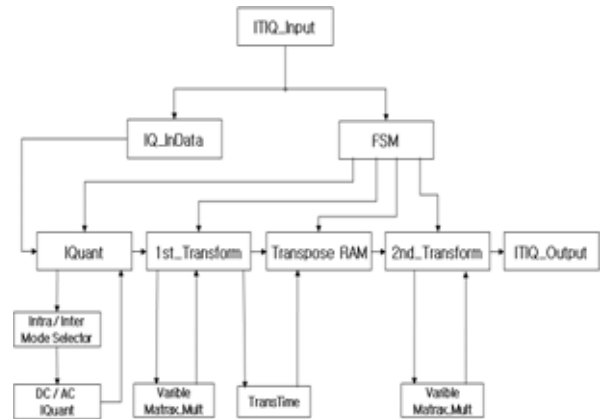


▶▶ 그림 5. 듀얼포트 트랜스포즈 메모리

그림 5는 만약 8×8 크기의 수평방향의 IDCT를 수행한다면 이때 나온 $E_{M \times N}$ 의 결과값은 0, 1, 2, 3, 4, 5, 6, 7, 8, ... 과 같이 순차적인 주소 값과 함께 듀얼포트 전치 메모리에 저장되어 진다. 이후 수직방향의 IDCT를 수행할 경우 0, 8, 16, 24, 32, 40, 48, 56, 1, 9, ... 의 주소 값의 데이터를 순차적으로 얻어옴으로써 입력된 $E_{M \times N}$ 을 $E'_{M \times N}$ 으로 변환하여 곱셈하는 효과를 얻을 수 있다. 이때 사용되어진 메모리는 최대블록 크기인 8×8 블록이 2개 저장되어질 수 있는 크기로 사용함으로써 두 개의 IDCT연산이 하나의 메모리에서 입력 및 출력을 하는데 공간 부족에 따른 대기시간이 없도록 한다[8][9][10].

V. ITIQ 모듈 설계

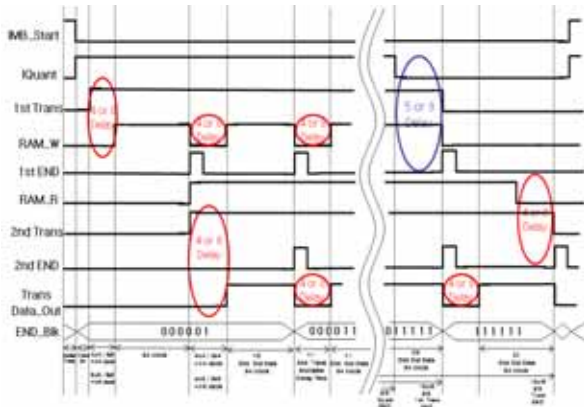
5.1 ITIQ 블록도



▶▶ 그림 6. ITIQ 전체 블록도

그림 6은 ITIQ 전체 블록도로서 모든 모듈은 상태머신에 의해 제어된다. 디코딩 데이터와 블록 정보신호가 인가되면 양자화 모듈에서는 인터 또는 인트라 모드에 따라 구분되어 AC 또는 DC 방식의 역양자화 처리과정을 거친다. 이후 수평방향의 크기에 따라 4 또는 8 블록크기의 1st IDCT를 수행한다. 이 중간 결과는 전치 메모리에 저장되어 정해진 주소값에 따라서 수직방향의 크기에 맞춰 2nd IDCT를 거쳐 최종적인 ITIQ결과 데이터를 얻는다[9][10].

5.2 ITIQ 타이밍



▶▶ 그림 7. ITIQ 타이밍도

그림 7은 VC-1용 표준 영상 포맷에 맞춰 YCbCr의 4:2:0 방식을 기준으로 실험한 타이밍 이다. 블록 크기에 상관없이 모든 변환 과정은 16x16매크로블록 단위로 이루어지도록 설계하였다. 만약 수평방향의 블록 크기가 4이라면 최초 1st IDCT는 4개 블록에 대한 처리를 하는 것이고 블록 크기가 8이라면 2개 블록을 처리하는 방식으로 설계하였다.

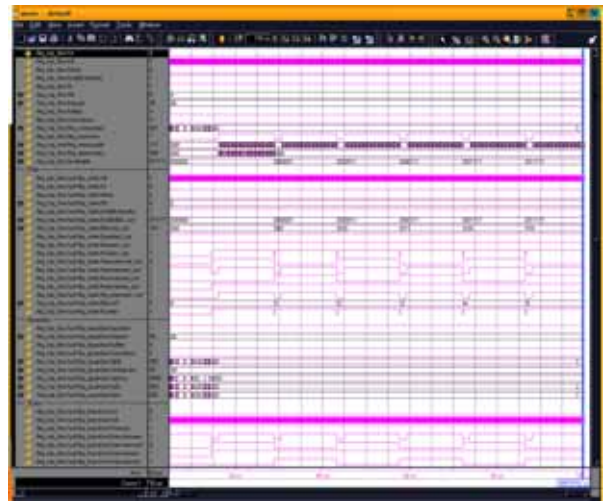
최초의 메모리 초기화 및 역 매크로블록 시작 신호에 따라서 양자화가 수행되며 양자화 결과 데이터가 나오는 2클럭 이후부터 1st IDCT가 수행된다. 이후 수평 방향의 블록 크기에 따라서 4 또는 8클럭의 행렬곱셈 지연시간이 발생하게 되며 이후부터는 전치 메모리에 결과가 64클럭동안 저장되게 된다. 이후 수직방향의 블록 크기에 따라서 4~8클럭의 지연시간 이후에는 최종적인 ITIQ블록 Y0의 출력이 시작된다. 이후 64클럭동안 Y0결과가 출력되고 다음 블록부터는 2nd IDCT의 최초 행렬곱셈 지연시간에 따라서 4~8클럭의 지연이후에 Y1, Y2, Y3, Cb, Cr의 블록 데이터가 각각 출력되어진다 [7][8][9][10].

최초 지연시간부터 첫 출력 데이터가 나오기 시작하는데 79~86클럭의 지연이 생기고 이후부터 64클럭동안 출력되는 블록데이터 6번에 4~8클럭지연 5번을 통해 첫 번째의 16x16 컬러 매크로블록을 처리하게 된다. 이때 총 483~510클럭이 소요된다.

VI. ITIQ 구현 결과

6.1 시뮬레이션 결과

그림 8은 최초 초기화 사이클 이후부터 역양자화 및 역변환 연산을 순차적으로 수행한 결과 파형이다. ModelSim XE II /Starter 5.7g 클럭주기 100ns 환경에서 (640*480)크기의 컬러영상을 디코딩하는데 494471~528070 클럭이 소요되었다.



▶▶ 그림 8. ITIQ 결과파형

6.2 ITIQ 결과 정리

표 1은 Xilinx XC2VP100 FF1696-6 시리즈에서 설계된 프로세서의 결과이다. ITIQ 모듈의 합성결과 설계에 필요한 게이트 수는 총 93,218개였다. 선택한 FPGA에서는 71.469MHz의 동작속도 나타내었다. 이는 4:2:0포맷의 컬러 영상 프레임 한 장을 디코딩하는데 최대 0.0074초면 가능하고 초당 30프레임 영상의 ITIQ 디코딩 연산에 걸리는 총 시간은 0.222초면 가능하다.

[표 1] ITIQ 설계결과

분 류	내 용
FPGA	Xilinx XC2VP100 FF1616-6
Tools	Xilinx ISE 7.1i
Total Gate	93K
Maximum Frequency	71.469MHz
Image Format	4:2:0 YCbCr 16x16 Macro Block
MB Process Cycle	First MB : 483~510 Others MB : 412~440

VII. 결론 및 향후 연구방안

본 논문에서는 복잡한 곱셈연산에 따른 효율성저하를 막기 위하여 비트확장 및 덧셈연산만을 통해 간단히 곱셈연산을 수행 하였다. 역변환시 에는 일반적으로 사용되는 듀얼포트 메모리가 비교적 많은 면적을 차지함에 따라 트랜스포즈 어레이를 사용함으로써 설계면적을 축소시켰다. 또한, 변형된 역변환식을 사용함에 따라 행렬곱셈에서 사용되는 변환계수의 유지를 통하여 1st IDCT와 2nd IDCT 모두 공통된 행렬곱셈 모듈을 사용할 수 있게 하여 설계의 복잡성을 줄였다. 마지막으로 역

변환시 사용되는 블록크기가 가변적임에 따라 각각의 블록 처리 설계가 필요하였지만 본 논문에서는 8x8단위로 고정적으로 처리함으로써 블록사이즈에 따른 불필요 추가 설계를 줄였다.

현재 VC-1에 대한 연구 자료가 미비한 상태에서 비교적 간단한 하드웨어 구조 및 다양한 처리능력을 갖춘 ITIQ 디코더 설계는 향후 연구방향 및 실험에 유용하게 활용될 것이다. 본 연구를 바탕으로 JPEG 및 H.264와 연계 연구를 통해 행렬곱셈 계수 및 양자화 변환식을 추가적으로 접목시킴으로써 부분적인 통합코덱 모듈을 구현할 것이다. 또한 움직임 보상 및 가변길이 코딩, 각종 필터 등의 추가적인 통합 모듈을 접목 설계함에 따라 하나의 완성된 통합 코덱의 연구를 지향함을 본 연구의 목표로 삼는다.

■ 참고 문헌 ■

- [1] 후지와라 히로시, 정제창 역, “최신 MPEG”, 교보문고 1995.
- [2] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, JVT-G050r1, 2003.
- [3] 이호석·김준기, “알기 쉬운 MPEG-2: 소스코드 해설”, 홍릉과학 출판사, 2002.
- [4] Iain E.G. Richardson, “H.264 and MPEG-4”, WILEY, 2003.
- [5] 김영문, “Fast Motion Estimation Algorithm for Multiple Reference Images Based on Elimination Algorithm”, 중앙대 첨단영상대학원, 2004.
- [6] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, JVT-I020, 2003.
- [8] Fernando Pereira·Touradj Ebrahimi, The MPEG-4 Book, 2002.
- [9] John Watkinson, The MPEG Handbook, Focal Press, 2001.
- [10] Hee-Soon Kim, Seunghwan Kim, and Yo-Sung Ho, “Fast Mode Decision Algorithm Using Mode Classification for H.264”, ICT 2004, No. 18-19, 2004