

H.264/AVC 인트라 예측모드용 디코더 설계

정덕영, 손승일

한신대학교 정보통신학과

Design of Decoder for H.264/AVC Intra Prediction Mode

Duck Young Jung, Seung Il Sonh

Dept. of Information and Communication HanShin University

E-mail : mykor01@naver.com

요 약

영상 정보의 발전으로 다양한 멀티미디어 서비스를 가능하게 하였고 네트워크와 IT의 발전으로 사용자가 풍부한 정보를 접할 수 있는 기회를 제공하였다. 이러한 동영상과 정지영상의 많은 정보를 압축하는 여러 방식 중에서 디지털 비디오 압축 관련 국제 표준안 중 MPEG-4와 H.264가 발표되었다. 유연성이 좋은 MPEG-4와 달리 H.264는 비디오 프레임의 효율적인 압축과 신뢰성을 강조 한다. 특히 H.264의 압축 기술은 HDTV처럼 큰 영상 뿐 아니라 카메라폰이나 DMB등의 특히 작은 크기의 영상에서 고품질의 영상을 보다 효율적으로 제공 한다.

본 논문은 기존의 동영상 압축 표준에 비하여 높은 압축성과 유연성의 장점을 가지고 있고 표준 H.264/AVC에서 공간적 예측을 사용하여 비디오 프레임을 압축하는 방법인 Intra coding 에서 사용하는 여러 모드 중 4*4 예측모드를 연구하여 C언어를 이용한 최적화된 시뮬레이션과 Intra coding decoder의 성능평가를 통한 최적화를 실시하였고, 최적화된 예측 정보를 바탕으로 Intra coding decoder를 VHDL언어를 이용하여 하드웨어로 구현하였다.

1. 서론

요즘의 영상정보를 처리하고 전송하는 방식의 발전은 음성 정보로 국한된 서비스로부터 정지 영상 및 동영상등의 정보를 포함한 다양한 멀티 미디어 서비스를 제공하고 네트워크망의 발전과 대중매체를 통하여 사용자가 풍부한 정보를 접할 수 있는 기회를 제공하였고, 동영상 및 정지영상의 방대한 정보를 저장하고 전송하기 위해 여러 가지 압축 방식이 연구 되어 왔으며 1988년 표준 규격이 대두된 이후 많은 표준 규격이 제정 되었는데, 그 중 동영상 표준 규격인 MPEG-1, MPEG-2, MPEG-4, JPEG, JPEG2000, H.263 등이 동영상뿐 아니라 정지 영상 압축을 사용하는데 최근 MPEG-4와 AVC(Advanced Video Coding)으로 발표된 H.264/AVC의 인트라 코딩은 기존의 압축 방식 보다 고품질의 영상을 효과적으로 보여주고 있다[1][2].

본 논문에서는 한 화면을 4*4의 일정한 블록으로 나누어 첫 블록은 그대로 전송하고 나머지

블록은 바로 전 블록이나 이웃하는 윗부분의 블록을 사용하여 9가지 인트라 예측을 수행하는 인트라 코딩 중 3가지 모드만 사용하여 기존의 압축 방식보다 고품질의 영상을 효과적으로 보여주고 있는 H.264/AVC 인트라 코딩용 디코더를 C언어를 사용하여 성능평가를 하고 최적화된 결과를 이용하여 VHDL언어로 설계하였다.

2. H.264의 개요

H.264는 비디오 프레임의 효율적인 압축과 신뢰성을 강조 하고 MPEG(Moving Picture Experts Group)과 유사한 방법으로 활동하며, 일련의 비디오 통신 표준안 시리즈를 개발했던 국제 통신 위원회(ITU-T : International Telecommunication Union)의 연구 그룹인 VCEG(Video Coding Experts Group)에 의해 시작되었다. H.264 표준안 개발의 최종 단계는 Joint Video Team에 의해 수행되고 있으며, VCEG와 MPEG의 협동적인 노력으로 ISO/IEC(MPEG-4 Part 10)과 ITU-T

(Recommendation H.264)의 공동 후원 하에 2003년 공동 발표 되었다[3][4][5].

H.264는 기본적으로 직사각형 비디오 프레임의 효과적이고 강력한 압축 및 전송을 지원하기 위한 것으로 코딩된 픽처를 생성하기 위해 필드 또는 순차주사(또는 비월주사 비디오)의 프레임을 인코딩 한다. 코딩된 프레임은 프레임 번호를 가지는데, 프레임 번호는 디코딩 순서와 반드시 관련된 것은 아니며, 순차주사 프레임 또는 비월주사 프레임의 각각의 코딩된 필드는 필드의 디코딩 순서를 정의하고 있는 픽처 순서 번호를 가지고 있다. 이전에 코딩된 픽처(참조 픽처)는 이후의 인트라 예측을 위해 사용한다. 참조 픽처는 list 0과 list 1로 표현하며 하나 또는 두 개의 목록인 참조 픽처에 해당하는 숫자들의 집합으로 구성된다.

코딩된 픽처는 여러 개의 매크로블럭으로 구성되는데, 각 매크로블럭은 16*16 휘도 샘플 및 이와 관련된 색차 샘플인 8*8 Cb와 8*8 Cr 샘플을 포함한다. 각 픽처 내의 매크로 블럭들은 슬라이스에 배열되는데, 슬라이스에는 매크로 블럭들이 순차적인 스캔 순서로 존재한다. I 슬라이스는 I 매크로블럭만을 포함할 수 있다. I 매크로블럭은 현재 슬라이스 내의 디코딩된 샘플로부터 인트라 예측을 사용하여 예측한다.

예측은 전체 매크로블럭의 휘도샘플과, 관련된 색차 샘플 또는 매크로블럭 내의 각각의 4*4, 8*8, 16*16 블록의 휘도샘플에 대해 수행된다. 예측코딩이란 이전프레임을 기반으로 현재 프레임의 영역에 대한 예측 데이터를 생성하고, 현재 영역으로부터 예측영역을 빼서 오차 데이터를 만들어 내는 것을 의미한다. 예측이 성공적으로 수행되면, 오차 데이터의 에너지는 원본 프레임의 에너지보다 적게 되고, 오차 데이터는 원본 프레임보다 적은 비트로 표현될 수 있다. 이전에 전송된 동일한 이미지 또는 프레임의 샘플로부터 예측 이미지 샘플 또는 영역을 만들어 낼 수 있다. 예측 코딩은 초기 이미지 압축 알고리즘의 기초로 사용되었고, H.264 인트라 코딩의 중요한 요소인 변환과정에 적용한다[1][5].

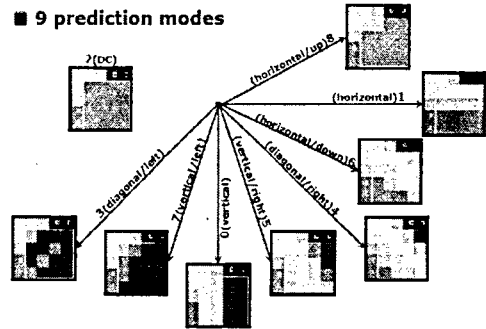
3. 인트라 예측

인트라 예측은 공간적인 상관관계를 이용한 것으로써 영상 내에 현재 압축하고자 하는 블록이나 화소는 이전에 이웃하는 블록이나 화소와

깊은 상관관계가 있다는 것이다. 즉 현재 압축하고자 하는 블록이나 화소는 이전의 이웃하는 블록이나 화소와의 차이 값만을 이용하여 압축할 수 있으며 또한 복호화도 할 수 있다.[1][6][7].

3.1 4*4 예측모드

그림1은 4*4 인트라 예측 모드를 보여 주고 있다.



[그림 1] 4*4 블록 인트라 예측모드

M	A	B	C	D	E	F	G	H
I	a	b	c	D				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	P				

[그림 2] Labeling of prediction samples 4*4

그림2에서 위쪽과 왼쪽의 샘플들(A~M)은 이전에 인코딩되고 복원되었으므로, 인코더와 디코더에서 예측의 참조 데이터로 사용될 수 있다. 예측 블록 P의 샘플 a,b,c ... p는 다음에 따라 샘플 A~M으로부터 9개의 모드로 계산되는데 각 모드의 대한 설명은 표1에 나타나 있다 [1][5][6].

[표 1] 4*4 블록의 9가지 인트라 예측모드

모드 0 (Vertical)	위쪽 샘플 A,B,C,D가 수직으로 확장
모드 1 (Horizontal)	왼쪽 샘플 I,J,K,L이 수평으로 확장
모드 2 (DC)	P의 모든 샘플은 샘플 A...D 그리고 I...L의 평균으로 예측
모드 3 (Diagonal Down-Left)	샘플은 좌측 하단과 우측 상단 사이의 45° 각도로 보간
모드 4 (Diagonal Down-Right)	샘플은 우측 하단으로 45°각도로 확장
모드 5 (Vertical-Right)	수직의 오른쪽 약 2.66°각도로 확장(넓이/높이=1/2)
모드 6 (Horizontal-Down)	수평의 아래 약 2.66°각도로 확장
모드 7 (Vertical-Left)	수직의 왼쪽 약 2.66°각도로 확장(또는 보간)
모드 8 (Horizontal-Up)	수평의 위쪽 약 2.66°각도로 보간

3.2 8*8 예측모드

인트라 코딩된 8*8 색차 성분은 이전에 인코딩된 위쪽 또는 왼쪽의 색차 샘플로부터 예측되며, 두 개의 색차 성분은 항상 같은 예측 모드를 사용한다. 8*8에서 사용되는 4가지의 예측 모드는 16*16 예측 모드와 매우 유사하다. 모드에는 모드 0 (DC), 모드 1 (Horizontal), 모드 2 (Vertical) 그리고 모드 3 (Plane)이 있다[4][5].

3.3 16*16 예측모드

표2는 16*16 예측 모드를 보여 주고 있다. 4*4 예측 모드와는 다르게 표2와 같은 4가지의 모드로 16*16 위도 성분이 한번의 동작으로 예측될 수 있다[4][5].

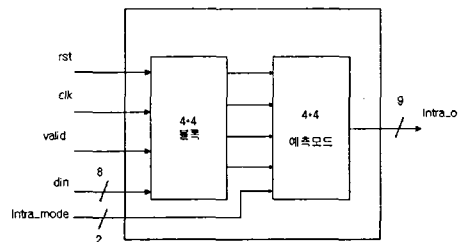
[표2] 16*16의 4가지 예측 모드

모드 0 (Vertical)	위쪽 샘플로부터 확장(H)
모드 1 (Horizontal)	왼쪽 샘플로부터 확장(v)
모드 2 (DC)	위쪽과 왼쪽 샘플의 평균(H+V)
모드 3 (Plane)	선형 'plane' 함수가 왼쪽과 위쪽 샘플 H와V에 사용부드럽게 변화하는 휘도 영역에서 잘 동작

4. 인트라 디코더 설계

4.1 인트라 디코더 블록도

그림3은 인트라 디코더의 블록도를 보여 주고 있다. din으로 들어오는 8비트의 영상정보를 valid 신호가 활성화 될 때 입력받아 4*4 블록에서 4*4 예측모드로 연산하기 위한 4*4 크기의 블록으로 처리하여 전송하고 전송된 데이터를 이용하여 9개의 모드 중 본 논문에서 사용한 3가지의 모드의 역연산하고 입력된 모드의 연산된 값을 선택하여 연산된 데이터를 처음 입력받고 16clock 후에 데이터를 보낸다.

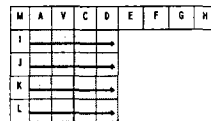


[그림 3] 인트라 디코더 블록도

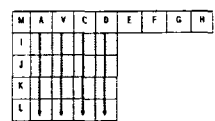
4.2 3개의 인트라 예측모드

그림4는 9개의 인트라 예측모드 중 본 논문에서 사용한 3가지의 예측모드를 보여 주고 있다.

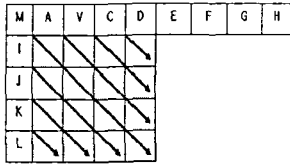
9개의 예측모드를 모두 사용하여 연산을 수행하는 하드웨어로 설계하게 되면 많은 연산처리로 인하여 처리속도가 느려지게 되고 설계한 하드웨어가 복잡해지는 문제가 발생하게 된다. 사람의 시각은 9개 모드로 할 경우와 3가지 모드만으로 수행할 경우 별 차이를 느끼지 못하기 때문에 9가지 예측모드로 하드웨어를 설계할 필요 없이 본 논문에서는 그림4와 같이 모드0(Vertical), 모드1(Horizontal) 그리고 모드4(Diagonal Down-Right)의 3가지 모드만을 사용하여 전송된 데이터를 인트라 예측모드의 역예측으로 복잡하고 처리속도가 느린 단점을 보완하여 복원하였다.



모드 0(Vertical)

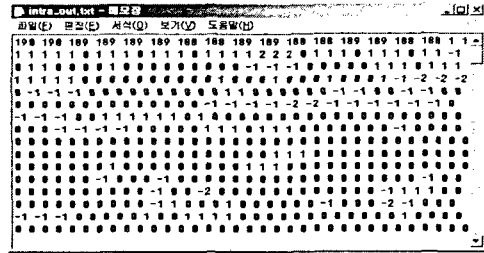


모드 1(Horizontal)



모드 4(Diagonal Down-Right)

[그림 4] 3개의 인트라 예측모드

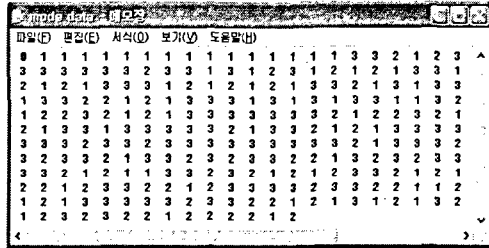


[그림 6] C에서 사용된 data

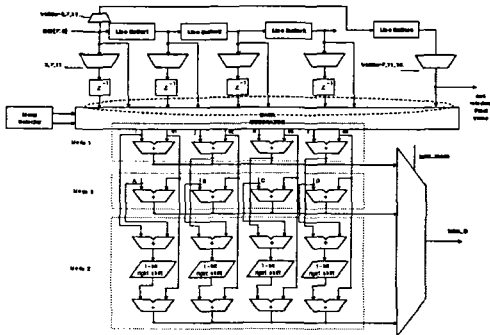
4.3 인트라 디코더 내부 블록도

그림5는 인트라 내부 블록도를 보여주고 있다. 8비트의 영상정보를 입력 받아 4*4 크기의 블록으로 만들기 위해 3개의 버퍼를 사용하여 4*4 예측 모드의 맞게 데이터를 저장하고 16개의 데이터를 입력 받은 후 4*4 예측 모드 연산을 수행하기 위해 line buffer를 사용하여 한번에 4개의 데이터를 전송한다.

전송된 데이터를 입력 받아 9개의 4*4 모드 중 본 논문에서 선택한 3가지의 모드의 역연산을 수행하고 각 모드의 연산 결과 중 intra_mode의 data에 따라 연산된 결과를 출력한다.



[그림 7] C에서 사용한 mode input



[그림 5] 인트라 디코더 내부 블록도

그림8은 4*4 크기의 블록을 예측 모드를 통해서 데이터를 처리하여 복원된 데이터를 보여주고 있다.



[그림 8] C를 이용한 시뮬레이션 intra output

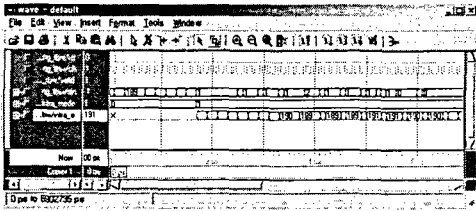
5. 시뮬레이션 및 고찰

5.1 C를 이용한 시뮬레이션

아래의 그림들은 C로 시뮬레이션 하여 최적화한 결과를 보여 주고 있다. 그림6은 C로 시뮬레이션 할 때 사용된 intra 인코더의 출력값을 입력으로 입력받은 data를 나타내고, 그림7은 4*4 예측모드에서 사용하는 9개 모드 중 본 논문에서 사용한 3가지 모드로 intra 인코더에서 연산했을 때 얻어진 모드를 입력 받은 data를 보여 주고 있다.

5.2 시뮬레이션

그림9는 인트라 예측을 통한 인트라 디코더의 출력과형을 보여주고 있다. 본 논문에서 설명한 바와 같이 8비트의 영상 data를 입력받아 4*4 크기의 블록으로 만들어서 3가지의 4*4의 예측 모드를 이용하여 연산된 결과 중 가장 작은 연산 결과를 선택하는 방식으로 처음 data를 받고 16 clock 후에 연산하여 복원된 data를 전송한다.



[그림 9] 인트라 디코더의 출력파형

5.2 합성 결과

[표3] 설계 합성 결과

NO. of Slices	No. of Gates	Timing	Target Device
887	28,727	Minimum Period: 5.958ns	XC2V1000 -6bg256

본 논문에서 설계한 인트라 디코더의 각 등가 게이트와 사용 슬라이스 수 그리고 타이밍 결과가 표3과 같이 나타났다.

6. 결론

영상 정보가 발전함에 따라 방대한 정보를 한 채널에 보다 많은 데이터를 전송하고, 빠른 전송 속도에 신뢰성 높은 데이터를 전송하여 고품질의 영상을 제공하기 위한 압축 방법이 활발하게 연구되고 있다. 그중 하나로 최근 H.264/AVC는 특히 비디오 프레임의 효율적인 압축에 집중하고 있는데 JPEG과 JPEG2000 그리고 유연성이 좋은 MPEG-4 보다 높은 압축률로 좋은 정지 영상 압축 표준안을 제시하고 있다. 이런 H.264/AVC에서 사용되는 인트라 예측 모드를 본 논문에서는 C언어를 이용한 최적화된 시뮬레이션과 Intra coding 성능평가를 통한 최적화를 실시하였고, 최적화된 예측 정보를 바탕으로 인트라 예측모드용 디코더를 VHDL언어를 이용하여 하드웨어로 구현하였다. 설계된 모듈을 이용하여 기존의 정지된 영상 압축 방법인 JPEG이 1/16로 압축했을 때 화면이 깨지는 것에 비해 Intra coding은 원 화소의 데이터와 같은 효과를 나타내고 부호화기의 복잡도가 기존의 표준에 비해 훨씬 증가하는 단점이 있지만 H.264/AVC의 원리 및 기술을 잘 이해하고 활용하면 카메라폰이나 DMB 그리고 DVD등에서 사용하는 H.264/AVC에 응용이 가능하다.

7. 참고문헌

- [1] 박경세, "디지털 멀티미디어 방송 기술 및 서비스" 커뮤니케이션선북스
- [2] H. S. Malvareral., "Low-Complexity Transform and Quantization in H.264/AVC", IEEE Trans. on Circ & Sys for Video Technology, Col. 13, No.7, pp.598-603, July 2003.
- [3] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, "JVT-G050r1", 2003
- [4] McubeWorks, Inc., "H.264와MPEG-4 PART10 AVC Video coding", 2004
- [5] Lain E. G. Richardson, "H.264 and MPEG-4", 홍릉과학출판사, 2004.9
- [6] Detlev Marpe, Valeri George, HansL.Cycon, and Kai U. Barthel,"Performance evaluation of Motion_JPEG2000 in comparison with H.264/AVC operated in pure intra coding mode", 2003.
- [7] Video Team, R&D Center, "H.264 Overview", 2004.12