

서브 블록 분할을 이용한 효율적인 근접 무손실 화면내 부호화

최정아 호요성

광주과학기술원 정보기전공학부

{jachoi, hoyo}@gist.ac.kr

Efficient Near Lossless Intra Coding using Sub-block Partitioning

Jung-Ah Choi Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)

요약

본 논문은 H.264/AVC를 이용한 근접 무손실 화면내 부호화에서 기존의 라인 기반 예측 방법에서 발생할 수 있는 오류 전파 문제를 해결하기 위한 방법을 제안한다. H.264/AVC의 화면내 예측 부호화는 주변 블록의 복원 화소들을 현재 부호화하려는 블록의 예측값으로 사용하여 공간적 상관성을 제거하고 부호화 효율을 향상시킨다. 기존의 라인 기반 예측 방법에서는 일반적으로 화소 간 거리가 멀어질수록 화소 간 상관성이 떨어지므로 라인 단위로 예측을 수행하여 예측의 정확도를 높였다. 하지만, 이 경우 오류 전파 문제에 취약하다는 단점이 있다. 본 논문에서는 화면내 16x16 수직 및 수평 예측 모드에 대해 매크로블록을 라인 단위로 예측하면서 정해진 서브 블록마다 참조 화소를 업데이트하는 방법을 제안한다. 따라서, 라인 기반 예측 방법의 예측 정확도는 유지하면서 오류 전파 문제를 예방할 수 있다. 실험을 통해, 제안하는 방법이 High 프로파일에서 H.264/AVC 표준 소프트웨어 JM 12.2에 비해 평균 약 5.8%의 비트율을 감소시킬 수 있음을 보였다.

1. 서론

최근 기술의 급속한 발전에 따라 우수한 품질의 HD 콘텐츠가 등장하고 관련 산업의 부흥기를 맞고 있다. 뿐만 아니라 영화계를 중심으로 활성화되고 있는 3차원 영상 분야에서도 높은 몰입감을 위해서는 HD 이상의 입력 영상이 필요하므로 고화질 콘텐츠를 다루는 기술의 중요성이 높아지고 있다. 관련 기술로는 콘텐츠 획득, 압축, 시스템 송수신, 콘텐츠 재생 및 저장, 품질 측정 등의 기술이 있다. HD 콘텐츠는 기존의 영상에 비해 많은 양의 정보를 담고 있어 데이터의 양이 매우 크다. 따라서 이를 효율적으로 압축할 수 있는 효율적인 부호화 기술이 필요하다.

하지만 HD 콘텐츠의 압축은 단순히 압축양만 따지는 것이 아니라 시청자의 고품질 방송 서비스에 대한 욕구를 만족시킬 수 있을 만큼 복호된 화질이 선명함을 유지하고 있는지가 중요하다. 그 결과, 복호 후에도 고화질의 영상을 얻을 수 있는 무손실 또는 근접 무손실 압축 환경에서의 부호화 기술이 새롭게 떠오르고 있다. 기존의 H.264/AVC[1][2][3]는 기존의 비디오 압축 표준보다 높은 부호화 성능을 제공하지만, 손실 압축에 우선한 코덱이므로 근접 무손실 부호화에서는 성능이 높지 않다.

H.264/AVC의 인트라 16x16 예측은 현재 매크로블록에 인접한 최대 33개의 주변 화소를 이용하여 현재 매크로블록에 속한 256개의 화소 값을 한 번에 예측한다. 특히, 전체 예측 모드 중 수직과 수평 예측 모드에서는 16개의 수직 또는 수평 위치에 위치한 주변 화소로 256개의 화소 값을 예측하므로, 참조 화소와 거리가 멀어질수록 예측의 정확도가 떨어져 부호화 비트가 증가한다.

기존 H.264/AVC의 예측 정확도를 높이기 위해 인트라 16x16 블록의 수직과 수평 예측에 대해 16개 화소 한 줄을 단위로 예측을 수행

하는 라인 기반 인트라 16x16 예측 방법을 제안했다 [4]. 하지만 이러한 라인 기반 예측 방법은 복원된 화소에 오류가 생기는 경우 오류 전파(error propagation)를 막지 못한다는 단점이 있다. 본 논문에서는 이러한 오류 전파를 막기 위해 기존의 라인 단위가 아닌 서브 블록 단위의 효율적인 근접 무손실 화면내 부호화 방법을 제안한다.

2. 기존의 화면내 16x16 예측 방법

가. H.264/AVC 화면내 16x16 예측

H.264/AVC의 화면내 예측 부호화는 주변 블록의 복원 화소들을 현재 부호화하려는 블록을 예측하기 위한 참조 블록으로 활용하여 공간적 상관성을 제거하고 부호화 효율을 향상시킨다. 현재 H.264/AVC 예측에서 사용하는 화면내 예측에는 화면내 16x16, 화면내 8x8, 화면내 4x4 모드가 있다. 이 중 화면내 16x16 예측은 그림 1과 같은 네 가지 모드를 제공한다. 수직 (모드 0), 수평 (모드 1), 평균치 (모드 2), 평면 예측(모드 3)으로 정의된 각 모드는 정해진 방향성을 가진다. 이 중 수직, 수평, 평균치 모드는 크기가 커졌을 뿐 화면내 4x4 모드에서의 방법과 같고, 평면 예측 모드는 정수 연산만을 가지는 이중 선형 변환과 유사하다.

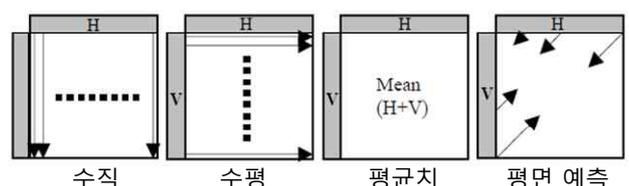


그림 1. 인트라 16x16 블록의 네 가지 예측 모드

나. 라인 기반 화면내 16×16 예측

예측의 정확도는 참조 화소와의 거리에 영향을 많이 받는다. 현재 예측할 화소를 좀 더 가까운 참조 화소를 이용하여 예측하기 위해, 제안하는 알고리즘에서는 화면내 16×16 블록 예측을 한번에 수행하지 않고, 라인 단위로 나누어 예측한다.

매크로블록에 대해 먼저 기존의 H.264/AVC의 화면내 16×16 예측을 수행하여 네 개의 모드에 대한 예측 블록을 만든다. 만들어진 네 개의 예측 블록에 대해 SATD 비용 값을 계산하여, 가장 작은 SATD 값을 가지는 예측 모드를 최적 모드로 결정한다. 결정된 최적 모드가 2보다 작은 경우, 즉 수직 또는 수평 모드인 경우, 제안한 라인 기반 화면내 16×16 예측을 수행하고 그렇지 않은 경우, 즉 평균치 또는 평면 예측인 경우에는 기존의 인트라 16×16 예측을 그대로 수행한다.

라인 기반 화면내 16×16 예측은 최적 모드의 방향성은 유지하면서 한 라인에 대해서만 예측을 수행한다. 기존의 인트라 예측에서는 한 매크로블록 전체에 대해서 예측을 수행하므로, 한 매크로블록 전체에 대해서 참조 화소의 값을 복사하여 예측하였으나, 제안하는 예측 방법에서는 수직 모드일 경우 1×16을 한 라인으로 보고 한 라인에 대해서만 참조 화소의 값을 복사하여 예측한다. 그 다음, 예측된 라인과 원 영상에서의 라인에 해당하는 화소 값의 차를 구해 잔여 영상을 구하고, DCT(Discrete Cosine Transform)를 수행한다. H.264/AVC 표준에서 제공하는 DCT는 4×4 정수 DCT이므로 호환성을 위해 잔여 영상의 한 라인을 래스터 주사(raster scan)하여 4×4 블록으로 만든다.

3. 제안하는 화면내 16×16 부호화 방법

화면내 4×4 블록은 9가지의 예측 모드를 제공하는 반면, 화면내 16×16 블록은 수직, 수평, 평균치, 평면 예측의 네 가지 모드만을 제공한다. 따라서, 복잡한 영역에서는 화면내 4×4 모드가 화면내 16×16 모드에 비해 더 나은 예측 정확도를 보이고, 화면내 16×16 모드는 비교적 복잡하지 않은 평탄한 영역에서 사용된다. 하지만, 근접 무손실 부호화의 경우, 평탄한 영역이 많지 않기 때문에 대부분의 영역에서 화면내 4×4가 최적 모드로 선택된다.

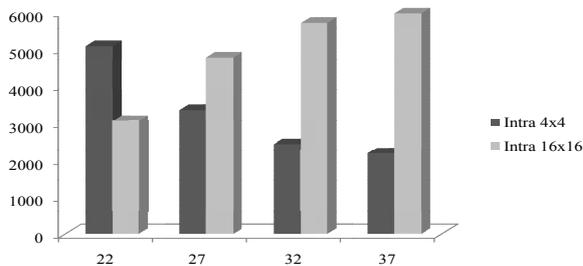


그림 2. 손실 화면내 부호화 후의 최적 모드 분포

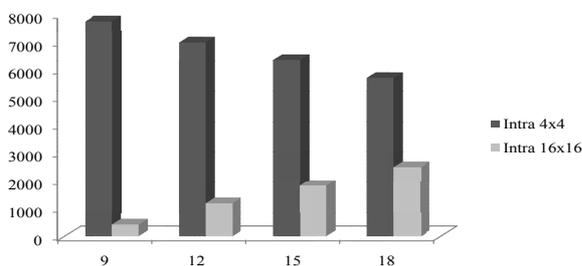


그림 3. 근접 무손실 화면내 부호화 후의 최적 모드 분포

그림 2와 그림 3은 BasketballDrive(1920×1080) 영상에 대해 H.264/AVC 표준으로 손실 및 근접 무손실 환경에서 각각 화면내 부호화를 수행한 후의 최적 모드 분포를 보여준다. 실험을 통해 일반적으로 높은 양자화 계수(QP = 22, 27, 32, 37)를 사용하는 손실 부호화에서는 화면내 16×16 모드의 비율이 높지만, 그에 비해 낮은 양자화 계수(QP = 9, 12, 15, 18)를 사용하는 근접 무손실 부호화에서는 화면내 4×4 모드의 비율이 훨씬 높음을 확인할 수 있었다.

화면내 16×16 예측 모드 중 방향성을 가지는 수직 및 수평 예측 모드는 16개의 이미 부호화된 화소로 현재 매크로블록 내의 256개의 화소값을 예측하기 때문에 매크로블록 내의 화소값에 증감값 (gradient)이 있는 경우 이를 제대로 예측하지 못한다. 이 경우, 참조 화소와 인접한 화소에서는 예측값이 비교적 정확하지만, 참조 화소와 멀어질수록 예측의 오차가 커진다. 따라서, 윌-왜곡 최적화(R-D optimization) 후에 이러한 블록들은 화면내 4×4 모드가 최적 모드로 결정된다. 화면내 4×4 모드는 작은 블록으로 나뉘어져 상대적으로 정확한 예측값을 제공하지만, 각 블록에 대한 예측 모드를 전송해야 하므로 더 많은 부호화 비트를 사용해야 하는 단점이 있다.

화면내 16×16 블록에서의 예측 정확도를 향상시키기 위해 라인 기반 예측을 제안했다 [4]. 라인 기반 예측은 기존의 화면내 예측 방법보다 조밀한 라인 단위 예측을 사용하여 근접 무손실 부호화에서 H.264/AVC보다 평균 약 6.92%의 부호화 효율을 향상시켰다.

인터넷을 통해 동영상을 전송할 때, 네트워크의 상태에 따라 패킷 손실이 발생할 수 있다. 만약 라인 기반 예측의 경우 복원된 화소값에 패킷 손실이 발생하면 해당 값 뿐 아니라 그 화소를 참조하는 다른 화소, 즉 매크로블록 내의 한 열에 해당하는 화소들에 영향을 미치는 오류 전파가 발생한다. 이를 막기 위해, 본 논문에서는 라인 기반 예측을 수행할 때 매크로블록을 정해진 서브 블록 단위로 나누어 복원 화소를 생성하여 오류 전파 현상을 막는 새로운 화면내 16×16 부호화 방법을 제안한다.

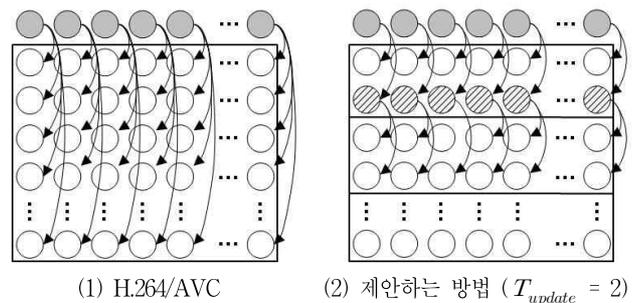


그림 4. H.264/AVC와 제안한 방법의 수직 예측 방법 비교

그림 4는 화면내 16×16 블록의 일부를 나타낸 것으로 기존의 H.264/AVC 화면내 16×16 수직 예측 방법과 본 논문에서 제안하는 서브 블록 분할을 사용한 인트라 수직 예측 방법을 보여준다. 여기에서 원은 영상의 각 화소를 나타내며, 회색으로 표시된 원은 이미 부호화된 현재 매크로블록의 주변 화소들을 나타낸다. 기존의 H.264/AVC 표준의 수직 모드는 현재 매크로블록 위의 16개의 참조 화소의 값을 그대로 복사하여 예측값을 생성하는 반면, 라인 기반 예측 방법은 매크로블록에 인접한 기존의 참조 화소들로 맨 위 한 라인에 대한 예측을 수행한 후 변환, 양자화, 역변환, 그리고 역양자화를 거쳐 복원 화소를 생성한다. 복원된 화소는 다음 라인에 해당하는 화소들을 예측하는 참조 화소로 사용된다. 이 과정을 매크로블록의 끝까지 수행한다.

매 라인마다 참조 화소를 현재 복원된 화소로 업데이트하지 않고, 설정한 값마다 업데이트를 하는 방식을 사용하면 오류 전파를 줄일 수 있다. 참조 화소값을 업데이트하는 주기를 T_{update} 라 정의하면 총 16 개의 라인에 대해 T_{update} 는 1, 2, 4, 8의 값을 가질 수 있는데, $T_{update} = 1$ 은 매 라인마다 참조 화소 업데이트를 수행하는 기존의 라인 기반 예측 방식과 동일하므로 제외한다.

즉, 제안하는 알고리즘은 매번 참조 화소를 업데이트하는 기존의 방식과 달리, 오류 전파를 방지하기 위해 업데이트 주기일 때에만 복원된 화소의 값을 업데이트한다. 현재 부호화하려는 매크로블록의 줄의 위치를 나타내는 정보를 $line_idx$ 라 하고, 이것이 1부터 16의 값을 가질 때, 업데이트 여부를 결정하는 판단식은 식 (1)과 같다.

$$\text{mod}(line_idx, T_{update}) = 0 \quad (1)$$

여기서, $\text{mod}(x, y)$ 는 x 에서 y 를 나눈 후의 나머지를 취하는 모듈로(modulo) 연산이다. 부호화가 끝난 후, 이 판단식의 결과가 0인 경우에는 다음 라인의 예측을 위해 복원된 화소값을 저장한다. 즉, $line_idx$ 가 T_{update} 의 배수인 경우에는 복원된 화소값을 저장하고, 이를 다음 화소의 참조값으로 사용한다.

주기 값 T_{update} 가 2, 4, 8인 경우에 대해 최적의 값을 결정하기 위해 다양한 Full HD (1920×1080) 영상에 대해 실험을 수행한 후, 각각 기존의 H.264/AVC와 부호화 성능을 비교하였다. 그 결과는 표 1에 나타나 있다.

부호화 성능을 평가하기 위한 객관적 지표로, BDBR(Bjonteggard Delta Bitrate)와 BDPSNR(Bjonteggard Delta PSNR)을 이용했다 [6]. 이는 각각 비교하고자 하는 두 방법 간의 비트율과 PSNR 차이의 평균을 의미한다. BDBR에서의 (-) 부호와 BDPSNR에서의 (+) 부호는 부호화 성능의 이득을 나타낸다.

표 1의 실험 결과를 통해, 모든 실험 영상에 대해 T_{update} 이 2인 경우 가장 높은 성능을 제공하는 것을 확인할 수 있다. 따라서, 제안하는 방법에서 Full HD 영상을 위한 T_{update} 의 값을 2로 결정했다.

표 1. T_{update} 값의 결정

실험영상	T_{update}	BDPSNR (dB)	BDBR (%)
Basketball Drive	2	+1.583	-17.092
	4	+0.600	-6.631
	8	+0.327	-3.590
Christmas Tree	2	+0.128	-1.183
	4	-0.018	+0.128
	8	-0.023	+0.168
Cactus	2	+0.451	-3.782
	4	+0.110	-0.894
	8	+0.031	-0.260
BQTerrace	2	+0.145	-1.146
	4	+0.010	-0.083
	8	0.013	-0.112

그림 5는 제안하는 서브 블록 분할을 이용한 화면내 부호화의 블록도를 보여준다. 기존의 라인 기반 예측과 부호화 순서는 동일하지만, 부호화 후에 업데이트 여부를 결정하는 판단식을 이용해 참조 화소 업데이트를 제어하는 피드백 부분이 추가됐다.

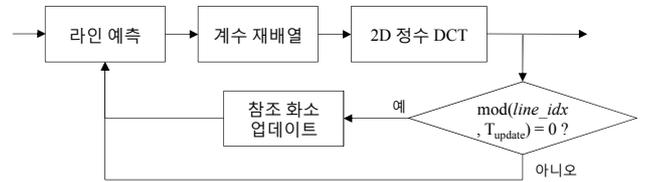


그림 5. H.264/AVC와 제안한 방법의 수직 예측 방법 비교

4. 실험 결과 및 분석

본 논문에서는 H.264/AVC 기준 소프트웨어 JM 12.2에서 제안한 알고리즘을 구현했다 [5]. 이 실험에서는 1920×1080 크기를 가지는 4 개의 4:2:0 포맷의 실험 영상 (BasketballDrive, ChristmasTree, Cactus, BQTerrace) 첫 프레임을 사용했다. 실험에서 사용한 부호화 변수는 표 2와 같다.

표 2. 부호화 변수

<i>ProfileIDC</i>	100
<i>LevelIDC</i>	51
<i>QPISlice</i>	9, 12, 15, 18
<i>Transform8x8Mode</i>	0
<i>SymbolMode</i>	1

표 3은 H.264/AVC 표준과 제안한 알고리즘으로 부호화한 결과를 비교한 것이다. 표 3에서 보는 바와 같이, 제안한 알고리즘은 H.264/AVC에 비해 BDBR로 평균 약 5.80%, BDPSNR로 평균 약 0.577dB의 부호화 이득을 제공한다.

표 3. H.264/AVC와 제안한 알고리즘의 성능 비교

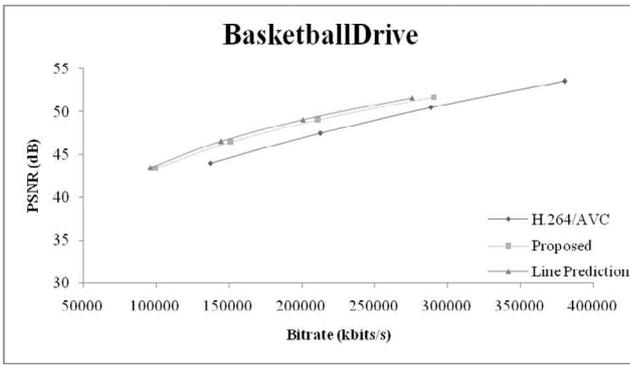
	BDPSNR (dB)	BDBR (%)
BasketballDrive	+1.583	-17.092
ChristmasTree	+0.128	-1.183
Cactus	+0.451	-3.782
BQTerrace	+0.145	-1.146
Average	+0.577	-5.801

표 4. 라인 기반 예측과 제안한 알고리즘의 성능 비교

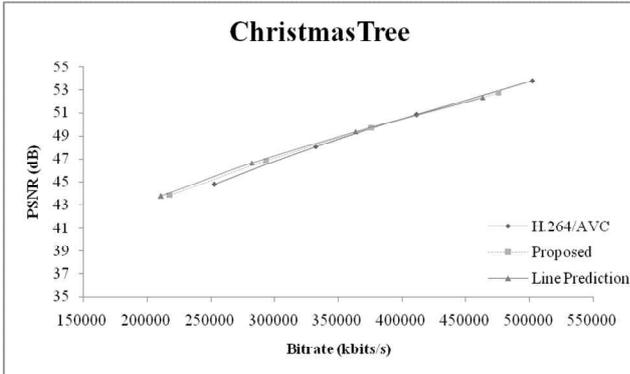
	BDPSNR (dB)	BDBR (%)
BasketballDrive	-0.410	+5.406
ChristmasTree	-0.154	+1.405
Cactus	-0.116	+1.041
BQTerrace	-0.019	+0.135
Average	-0.175	+1.997

표 4는 기존에 제안했던 라인 기반 예측과 제안한 알고리즘으로 부호화한 결과를 비교한 것이다. 표 4에서 보는 바와 같이, 제안한 알고리즘은 기존의 라인 기반 예측에 비해 BDBR로 평균 약 1.997%, BDPSNR로 평균 약 0.175dB의 부호화 성능 차이를 보인다. 제안한 방법은 기존의 라인 기반 예측과 성능 차이가 크지 않으면서 오류 전파 문제를 해결할 수 있다는 점에서 장점을 가진다.

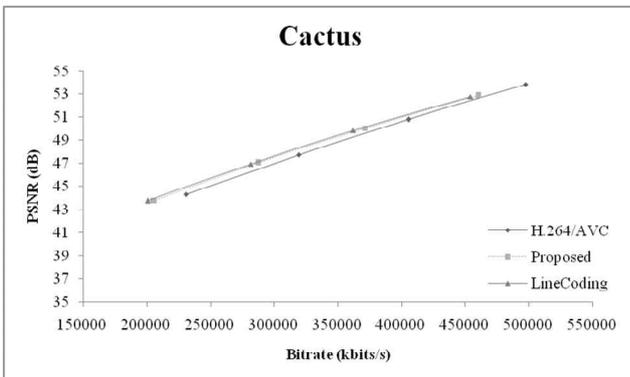
그림 6은 표 3과 표 4를 바탕으로 도시한 비트율-왜곡 곡선이다. 오류 전파 문제를 해결하기 위해 제안한 화면내 부호화 방법은 H.264/AVC보다 더 나은 부호화 효율을 보이고, 기존의 라인 기반 예측과 비슷한 성능을 보인다.



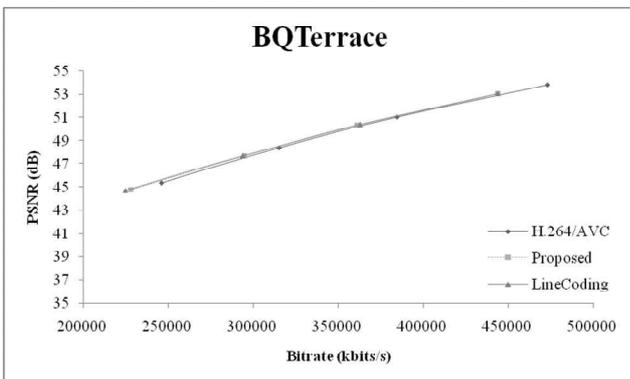
(a) BasketballDrive, 50fps



(b) ChristmasTree, 50fps



(c) Cactus, 50fps



(d) BQTerrace, 60fps

그림 6. 비트율-왜곡 곡선

있는 오류 전파 현상을 고찰했고, 그 문제점을 해결하기 위한 서브 블록 단위의 예측 방법을 제안했다. 제안한 알고리즘은 예측은 라인 단위로 수행하고 참조 화소의 업데이트는 정해진 서브 블록 단위로 수행한다. 그 결과, 콘텐츠 전송 시 발생할 수 있는 오류 전파 현상을 해결하면서 기존의 라인 기반 예측의 장점인 높은 예측 정확도를 유지할 수 있었다. 실험 결과로부터 제안한 서브 블록 기반 화면내 16×16 예측 방법이 H.264/AVC의 High 프로파일에 비해 동일 화질을 유지하면서 평균 약 5.8%의 비트 수를 감소시키는 것을 확인했다. 제안하는 방법은 동영상 부호화에서 보다 높은 압축 효율을 제공함과 동시에 오류 전파를 방지하기 위해서는 반드시 필요한 기술이므로, 동영상 부호화에서 유용하게 활용될 것으로 전망된다.

감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음 (NIPA-2010-(C1090-1011-0003))

참고 문헌

- [1] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC Video Coding Standard," IEEE Trans. Circuits System Video Technology, vol. 13, no. 7, pp. 560-576, July 2003.
- [2] I. Richardson, H.264 and MPEG-4 Video Compression. Wiley, 2003.
- [3] G. Sullivan, T. McMahon, T. Wiegand, and A. Luthra, Eds., Draft Text of H.264/AVC Fidelity Range Extensions Amendment to ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, ISO/IEC JTC1/SC29/WG11 and ITU-T Q.6/SG16 Joint Video Team Document JVT-L047, July 2004.
- [4] J.A. Choi and Y.S. Ho, "Line-by-line Intra 16x16 Prediction for High Quality Video Coding," IEEE International Conference on Multimedia and Expo (ICME) 2010, to be published.
- [5] http://iphome.hhi.de/suehring/tml/download/old_jm/jm12.2.zip, Joint Video Team, Reference Software Version 12.2.
- [6] G. Bjontegaard, "Improvements of the BD-PSNR model," ITU-T SG16/Q6, 35th VCEG Meeting, Berlin, Germany, 16th-18th, doc. VCEG-A111, July 2008.

5. 결론

본 논문에서는 H.264/AVC의 인트라 16×16 예측보다 좀 더 정확한 라인 기반 인트라 16×16 예측 알고리즘에서 발생할 수