<http://dx.doi.org/10.7236/IIBC.2014.14.6.99>

IIBC 2014-6-15

H.264에서 고속 매크로 블록 분할 결정 방법에 관한 연구

A Study on Fast Macroblock Partition Decision Method at H264

송대건*

Dae-Geon Song*

요 약 H.264/MPEG-4 AVC는 높은 부호화 효율을 재현하는 동영상 부호화 방식이나 부호화 처리가 많다는 문제점이 있다. 또한 많은 부호화 처리에서는 Inter예측에 관련된 처리가 대부분을 차지하고 있으며 특히 H.264/MPEG-4 AVC에서는 그 비율이 높게 나타나있다. 기존의 동화상 부호화 방식도 Inter예측이 차지하는 비율이 높으나 H.264/MPEG-4 AVC에서는 가변 블록 크기 움직임 보상, 2개 이상의 복수 참조 프레임에 더하여 1/4 영상 정도의 움직임 보상을 이용 할 수 있도록 하고 있으며 부호화 처리량 증가가 커다란 하나의 요인이 되어 있다. 따라서 본 논문에서는 부호화 대상 매크로 블록 크기와 탐색 중에 참조하는 매크로 블록의 pixel 수의 차분을 이용함으로써 움직임 보상 블록크기의 탐색을 유효하게 감소시키는 방법에 대해 제안하였다.

Abstract The performance improvement in MPEG-4 AVC is provided at the expense for higher computational complexity. Most of the complexity is caused by Inter prediction. To improve coding efficiency, some functions are added in H.264/MPEG-4 AVC, such as variable block size motion compensation, multi reference frame and quarter-pel motion compensation. A fast macroblock partition decision method is proposed in this paper. The macroblock size is efficiently determined by using the pixel value difference between encoding and the referred macroblock.

Key Words : PSNR, MPEG-4 AVC, RD Curve, SAD, Inter prediction

1. 서 론

2000년 초에 ITU-T와 MPEG에 의해 동영상 부호화 방식 H.264/MPEG-4 AVC(이후 H.264로 표시)가 국제 표준화^[1-5]되었다. H.264는 MPEG-2나 MPEG-4에 비교해서 1.5배에서 2배 높은 압축률을 실현할 수 있으나 부호화 처리량이 많다는 문제점을 안고 있다. 부호화 효율을 높이기 위해 도입한 복수 참조 프레임, 가변 블록 크기 움직임 보상, 1/4 pixel 정도 움직임 보상과 같은 처리

기능 등에 의해 Inter 예측이 점유하는 비율이 높아지고 있다. 또, 움직임 보상 탐색 자체에 관한 고속화는 이미 많은 검토가 진행되고 있으나 H.264에서 새롭게 추가된 기능에 관한 고속화 방법에는 아직까지 충분한 검토가 이루어지지 않고 있다. 따라서 본 논문은 움직임 보상 시, 가변 블록크기 결정방법을 착안하여 부호화 대상 매크로 블록과 탐색 중에 참조하는 매크로 블록의 pixel값의 차분을 이용함으로써 움직임 보상 블록 크기의 탐색을 효율 높게 줄이는 방법에 대하여 제안하고자 한다.

*정회원, 가천대학교 IT대학 전자공학과
접수일자: 2014년 11월 15일, 수정일자: 2014년 12월 10일
게재확정일자: 2014년 12월 12일

Received: 15 November, 2014 / Revised: 10 December, 2014

Accepted: 12 December, 2014

*Corresponding Author: dgsong@gachon.ac.kr

Dept. of Electronic Engineering, Gachon University, Korea

II. 본 론

1. H.264에서의 Inter 예측

H.264에 새롭게 추가된 Inter예측에 관련된 기능을 보면 우선 참조 프레임 수가 증가되었고, MPEG-2나 MPEG-4에서는 예측할 때에 참조 프레임을 1개 밖에 이용할 수가 없었으나 H.264에서는 2개 이상의 복수 참조 프레임을 참조 할 수 있도록 되어 있어 전 프레임에 대하여 탐색 처리를 하면 참조 프레임 수에 비례하여 움직임 보상을 요구하는 시간이 늘어난다.

또한 가변 크기 변화에 대응하여 블록크기 방법을 도입하였다. 이제까지의 움직임 보상의 단위는 16×16 pixel 단위 밖에 안되었으나 H.264에서는 다양한 단위구조를 만들어 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4 등 7가지의 블록크기를 이용하여 처리토록 하였다.

계층구조에서는 우선 16×16, 16×8, 8×16, 8×8 중 어느 크기를 사용 할 것인지를 먼저 결정해야만 하고 특히 8×8 pixel 단위 분할을 선택하였을 때는 8×4, 4×8, 4×4의 작은 블록크기를 이용 할 수 있도록 하였다.

그러나 이용 가능한 블록 크기의 종류가 늘어나서 전체를 조합하여 탐색 할 때에는 처리시간이 늘어난다는 문제도 있으나 최종 움직임 보상 정도가 기존의 1/2 pixel 정도에서 1/4 pixel 정도로 향상되었다.

표 1. Inter 예측 경향을 측정하는 실험 조건
Table 1. Experiment conditions which measure the tendency of Inter Prediction

JM Version	9.6
Profile	Baseline Profile
Input Sequence	Foreman, New(QCIF) Mobile&Calender, Tempete(CIF)
QP	I, P:22~28
Frame Nimber	150
Frame Structure	N=15, M=1
Frame rate	30fps
Reference Frame Number	5
Search rate	16
Optimization	RD-Optimization

1/2 pixel 정도의 pixel값은 6-tab의 FIR필터에 의해 계산되며 1/4 pixel 정도의 pixel값은 정수정도의 pixel값과 1/2 pixel 정도의 pixel를 이용하여 평균값 필터에 의

해 작성된다. 정도의 향상에 따라 기존의 탐색 범위와 같은 장소를 탐색하는 데는 보다 넓은 범위를 탐색하여야만 한다. 따라서 이러한 기능적 추가에 따라 정도가 높은 움직임 보상을 적용하도록 되어있어 Inter 예측이 차지하는 부호화 처리량은 증가하게 된다는 문제점도 안고 있다. 그러므로 이런 Inter 예측에 추가된 기능에 의한 부호화 효율의 차이를 확인하기 위한 표준 동화인 QCIF 크기의 Foreman, News, CIF 크기의 Mobile&Calendar, Tempete를 H.264의 참조 소프트웨어인 JM^[6]으로 부호화 처리 하였다. 표 1에 실제 조건을 나타냈으며 QP를 28로 설정하였을 때 처리시간을 그림 1에 나타내었고 움직임 보상 블록크기의 내역을 그림 2에 나타내었다.

또, 이용 가능한 블록크기를 제한하였을 때의 Mobile&Calendar의 RD 곡선을 그림 3에 나타내었다.

그림 1. 처리시간의 백분율
Fig. 1. Percentage of Processing Time

처리시간은 Intra 예측이 전체의 10% 전후를 차지하고 있음에 대해 16×16~8×8 pixel 단위까지의 블록크기에 대한 Inter 예측처리가 20% 전후, 8×8부터 4×4 pixel 단위까지의 서브블록크기에 대한 Inter 예측 처리가 60% 전후를 차지하고 있다.

그러므로 세밀한 블록크기에서의 탐색처리를 높이는 쪽이 전체에 대한 효과를 증대하는 것이라 생각된다.

블록크기 이용 비율은 움직임이 적은 News에서는 전 프레임에 대응하는 매크로 블록을 복사하여 이용하며 그 이외의 pixel에서는 16×16~8×8 pixel 단위 블록크기가 70%, 8×8~4×4 pixel에서는 10% 전후가 되도록 하였다.

그림 2. 블록크기의 백분율
Fig. 2. Percentage of Blocksize

8x8~4x4 pixel 단위의 블록크기를 이용함에 따른 PSNR의 향상은 적으나 16x16~8x8 pixel 단위로 분할함으로써 1[dB]정도 PSNR이 향상되고 있음을 확인할 수 있었다. 이상으로부터 블록크기는 8x8 크기까지의 분할은 PSNR의 관점에서는 대단히 유효하며 또한 세밀한 분할은 PSNR이 향상하는 것으로 모든 매크로 블록에서 탐색을 하면 대단히 많은 처리량을 차지하고 있음을 확인할 수 있었다.

그림 3. 블록크기 제한과 RD 곡선
Fig. 3. RD Curve with BlockSize Limit

2. 기존 방법

Intra 예측의 고속화 관점은 크게 3가지로 분류할 수 있다. 즉, (1)복수 참조 프레임 선택, (2)움직임 보상 블록크기 결정 (3)움직임 벡터 탐색 등이다.

움직임 벡터 탐색은 MPEG-2에 대해 검토된 방법을 그대로 이용 가능하나 복수참조 프레임 선택과 움직임 보상 블록 크기 결정에 관해서는 몇가지 실행되어 왔으나 다른 크기의 예측 결과를 이용하여 블록을 통합/분할하는 방법은 다양하게 표현되어 있다^[7-10].

문헌^[7,8]에서는 4x4나 8x8과 같은 작은 예측단위에서 움직임 보상을 처리한다. 움직임 벡터의 크기와 방향이 갖추어진 블록을 통합함으로써 탐색처리량을 줄이는 방법을 제안하고 있다. 그러나 이방법의 문제점으로는 처리량이 많은 작은 크기 예측을 함에도 불구하고 작은 예측단위로 얻을 수 있는 움직임 벡터의 쪽이 신뢰성이 낮다는 점을 들 수 있다.

문헌^[9,10]에서는 8x8 pixel 단위로 시작으로 움직임 보상을 구현하고 나서 블록을 통합/분할하는 방법을 제안하고 있다. 탐색 처리량이 증가, 움직임 벡터의 신뢰성이라는 2가지 문제점을 해결 할 수는 있으나 블록크기 이용을 제한하였을 때 성능이 떨어지는 문제를 고려해야만 한다. 큰 단위로부터 탐색을 시작하여 블록크기를 분할해 가면서 보다 작은 블록의 움직임 보상을 하는 방법은 처리량을 최대로 적게 할 수 있으며 정도가 높은 움직임 벡터를 얻을 수 있다. 그러나 분할을 할 것인가 말 것인가하는 판단이나 처리를 중지할지 말지를 판단하는 Threshold값에 따라 성능이 크게 좌우되어 Threshold값을 적절하게 결정하지 않으면 처리량을 통합하는 접근 방법보다 감쇄되지 않은 다는 문제점이 있다. 큰 크기로부터 분할하여 탐색하는 접근 방법은 최대로 처리량을 필요로 하는 8x8 크기 이하의 작은 블록크기 탐색을 최대로 억제할 수가 있으며 앞으로 새롭게 등장하는 동영상 부호화 방법에서 움직임 보상에 이용되는 블록크기의 종류가 더욱더 증가 하였을 때에 가장 성능을 발휘할 수 있다고 생각된다.

따라서 본 논문에서는 큰 크기로부터 탐색을 시작하여 분할해 가면서 탐색을 하는 방법에 대하여 검토하고자 한다.

III. 제안 방법

우선 먼저 분할 형태의 판단 방법에 대해 검토하고자 한다. 움직임 보상에서 분할 상태를 결정하는 방법에 맞추어 몇가지 정보가 필요하다.

예를 들어 분할은 동영상 중의 이동물체의 움직임에 따라 처리한다고 가정하며 에지 성분에 따라 분할하는 방법을 생각해 볼 수가 있다.

감각적으로 동물체의 경계에 따라 블록크기를 분할하는 방법이 자연스러우나 실제로 최적의 블록크기 분할은 에지 근접에 작은 블록크기를 집중시키는 것이 보편적이다. 그래서 탐색하고 있는 블록크기를 4분할하여 한 번에 작은 예측 단위로 SAD(Sum of Absolute Difference)를 취하고 판단의 기준으로 하는 방법을 제안하고자 한다. SAD의 크기에 따라 어느 위치가 효율 좋게 움직임 보상이 가능한지를 알 수 있으며 실제로 움직임 보상이 효율 좋게 처리되지 않는 장소에 초점을 맞추어 분할 처리를 함으로서 양질의 성능을 얻을 수 있다고 생각된다. 제안 방법을 그림 4에 나타내었다.

1. Search with Large Size
Calculate each SAD with Quarter Size
Update the position in which SAD is minimized.
2. Check MV position.
2-1 : 4 position is equal quit processing.
2-2 : 3 Position is equal search 16x8, 8x16 and 8x8
2-3 : each 2 position is equal search 16x8 or 8x16
2-4 : else search 8x8
3. Search with Small size

그림 4. 제안 방법

Fig. 4. Proposed Method

큰 크기로부터 탐색을 시작하여 4분할한 작은 단위 (16x16크기로 부터 8x8 pixel 단위, 8x8크기로 부터 4x4

크기 pixel 단위)로 SAD를 얻으면서 각각의 장소에 대하여 SAD값과 움직임 벡터 길이로부터 코스트의 값이 최소가 되는 위치를 기록하도록 한다.

어느 정도 탐색이 진행된 후, 기록된 위치가 모인 정도를 파악하고 처리의 중단 여부를 부가하도록 한다. 4 장소가 같은 위치를 가리키고 있을 때는 큰 크기를 선택하고 중첩 어딘가 두장소씩 모여 있는 경우에 있어서는 방향에 맞추어 2개씩 분할하여 처리한다.

큰 크기에서는 탐색을 끝내 처리가 중단된 경우라도 3개 또는 2개가 모여 분포되어 있는 경우와 4개가 각각 떨어져 있는 경우 등 3가지 경우를 생각할 수 있다.

3 또는 2개의 블록이 모여 있는 경우는 모든 분할 패턴을 탐색하도록 하여 가리키고 있는 위치가 4개가 서로 떨어져 있는 경우에는 블록을 4개로 분할하여 패턴만 탐색하도록 처리하도록 한다.

16x16으로부터 8x8로의 분할, 8x8로부터 4x4로의 분할은 같은 방법을 이용하도록 한다.

또, 분할 후의 각 블록에 대하여 탐색은 분할 전의 SAD가 최소가 되는 위치를 중심으로 처리하도록 함으로서 탐색 범위를 좁힌다. 이렇게 함으로서 예측 크기가 작게 되면 될수록 탐색점을 줄일 수 있도록 한다.

다음은 분할의 필요성을 판단에 사용하는 Threshold 값에 대하여 검토하고자 한다.

처리를 도중에 중단하는 방법으로서 문헌^[11]에서 사용하고 있는 완료된 부호화 주변 매크로 블록으로부터 목표 SAD를 예측하는 방법을 고려한다.

그러나 블록크기결정에 이용되는 SAD와 움직임 벡터를 합친 코스트 값은 근소한 값으로 정해지는 경우가 많다. 그래서 SAD가 최소가 되는 위치의 장소 방향을 첨가하여 어느 정도 간격에서 값이 변경되지 않는가를 판단하도록 하였다. 본 논문에서는 과거 100회 탐색하여 갱신이 되지 않는 경우에 중단 처리하는 대상으로 하였다. 이 제안 방법은 작은 블록 크기의 예측 결과로부터 통합하는 방법과 비교하여 먼저 “탐색회수가 적음”, 움직임 벡터 정도가 높음“을 부가하여”전체 코스트가 최소가 되는 탐색점을 얻음“과 같은 이점이 있다. 작은 크기에 대한 예측 결과로부터 통합을 하는 경우, 큰 크기를 이용하는 것은 통합 전에 대응하는 모든 위치의 코스트가 최소가 되는 경우이다.

표 2. 실험 조건

Table 2. Experiment Conditions

JM Version	9.6
Profile	Baseline Profile
Input Sequence	Foreman, New(QCIF) Mobile&Calender, Tempete(CIF)
QP	I, P:22~28
Frame Number	150
Frame Structure	N=15, M=1
Frame rate	30fps
MC Block Size	16×16~8×8
Reference Frame Number	1
Search rate	16
Optimization	RD-Optimization
MC Accuracy	Full Pel

그러나 실제로는 각각 위치의 코스트가 최소가 안 되더라도 전체 코스트는 최소가 되는 것이며 통합하는 접근에서는 대응 불가능하다. 문헌^[12-14]와 같이 블록크기마다 최적한 탐색 방법을 채용하는 방법이나 Diamond Search^[13]등과 같은 방법과 조합하였을 때, 큰 크기로부터 탐색을 하는 제안 방법에서는 큰 크기의 코스트가 최소가 되는 점을 탐색함으로써 탐색점을 이동함으로써 전체 코스트가 최소가 되는 점을 억제 할 수 있다.

IV. 실험 및 결과

1. 실험

참조 소프트웨어인 JM9.6에 제안 방법을 탑재하여 성능을 확인하였으며 실험 조건을 표2에 나타내었다.

블록 크기 분할의 타당성을 확인하기 위하여 참조 프레임은 1로 하고 이용 할 수 있는 블록 크기도 8×8까지로 제한하였다. 탐색 블록크기의 제한만으로 처리한 것을 Prop_1, 분할 후 탐색 범위를 제한한 것을 Prop_2로 하여 나타내었다.

단, JM의 탑재 문제로부터 Prop_2에 대해서는 16×16 탐색에 의해 얻어진 SAD의 최소 위치를 그대로 작은 블록에서의 움직임 벡터로 하였다.

2. 처리량의 감쇄도

제안 방법에 의한 처리량의 감쇄도를 표 3에 나타내었

다. 현재 JM에서는 각 블록마다 탐색 범위를 전부 위치에 두고 SAD를 미리 계산한 뒤에 중단 처리를 도중에 설정하여도 처리시간에는 그다지 차이가 나지 않기 때문에 처리량의 감쇄는 탐색점의 감쇄도로 나타낸다.

탐색은 Search Range가 16인 경우, $((16 \times 2) + 1) \times ((16 \times 2) + 1) = 1089$ [회]의 탐색점이 하나의 블록에 대하여 조사한다.

표 3. 실험 조건

Table 3. Search Point Reduction

	Forman		
	PSNR[dB]	Bitrate[Kbps]	Complexity[%]
Origin	35.29	262.26	-
16×16	35.15	288.26	-
Prop_1	35.25	267.04	42.54
Prop_2	35.21	279.57	7.17

	News		
	PSNR[dB]	Bitrate[Kbps]	Complexity[%]
Origin	36.74	144.64	-
16×16	36.70	150.42	-
Prop_1	36.74	145.59	11.87
Prop_2	36.72	147.50	2.91

	Mobile&Calendar		
	PSNR[dB]	Bitrate[Kbps]	Complexity[%]
Origin	33.48	3914.38	-
16×16	33.43	4036.08	-
Prop_1	33.48	3934.29	41.67
Prop_2	33.46	3978.52	6.90

	Tempete		
	PSNR[dB]	Bitrate[Kbps]	Complexity[%]
Origin	34.19	2438.21	-
16×16	34.15	2501.36	-
Prop_1	34.18	2455.12	42.27
Prop_2	34.17	2475.00	7.00

그림 5. Foreman RD 곡선
Fig. 5. Foreman RD Curve

그림 6. News RD 곡선
Fig. 6. News RD Curve

그림 7. Mobile RD 곡선
Fig. 7. Mobile RD Curve

그림 8. Tempete RD 곡선
Fig. 8. Tempete RD Curve

16×8, 8×16은 16×16의 2배, 8×8은 16×16의 4배의 탐색 처리를 보였다.

탐색 블록 크기 제한 만으로 탐색을 40%까지 감쇄 가능함을 Prop_1의 결과로부터 볼 수가 있다.

이 경우 부호량의 증가량은 적으며 탐색범위를 제한한 Prop_2에서는 탐색을 7%까지 감쇄 할 수가 있었으나 부호량도 동반 증가됨을 알 수 있었다.

그러므로 큰 크기에서 구하는 움직임 벡터에서는 작은 블록크기에 대하여 보정 할 필요가 있음을 알 수 있었다. 어느 경우에도 이용 가능한 블록크기를 16×16만으로 제한한 경우보다도 PSNR, 부호량에 대해서 좋은 성능을 얻을 수가 있었다.

3. PSNR과 부호량

표 3에서는 PSNR, 부호량이 각각 다르기 때문에 QP를 변화 시켜서 얻은 데이터에 의한 RD곡선을 그림 5부터 그림 8까지 나타내었다. 그림으로부터 제안 방법이 모든 부호량에서 16×16만을 이용한 경우보다도 효율 좋은 성능을 얻을 수 있음을 확인 할 수 있었다.

V. 결 론

H.264에서의 블록크기 분할 방법에 대하여 검토하였고 탐색 블록크기의 1/4 단위 마다 SAD를 구하여 분할 형상을 결정하는 방법을 제안하였다. 제안 방법에 의해 Inter 예측의 탐색점을 7% 정도로 줄일 수가 있었다.

이번 실험에서는 8×8 크기까지로 제한하였기 때문에 4×4까지 블록 크기를 이용 하여 처리 하였을 때의 성능 확인과 가로, 세로로 분할하였을 경우의 판단 정도의 향상 및 참조 프레임이 증가하였을 때의 감쇄 방법에 대하여는 앞으로의 풀어야할 과제라 할 수 있다.

References

- [1] ISO/IEC 14496-10, "Advanced Video Coding," ITU-T Rec. H.264, 2003
- [2] B.S. Min, S.J. Kim and D.K.Lim, "Enhancement of Wavelet-coded Image by Directional Filtering", Journal of the Korea Academia-Industrial cooperation Society, Vol. 9, No. 2, pp. 257-266, Apr. 2007.
- [3] C.S. Kang, "Method for Improvement of Coding Efficiency in Scalability Extension of H.264/AVC", The Journal of the Institute of Electronics Engineers of Korea, Vol. 47, IE, No.2, pp. 21-26, Jun.. 2010.
- [4] K.S.Park, M.J. Kim, J.H.Jun, S.R. Ryu and S.H. Kim, "Enhanced Mode Estimation Method for Intra/Inter Prediction in H.264/AVC," Journal of the Korea Academia-Industrial cooperation Society, Vol. 13, No. 4, pp. 1830-1838, Apr. 2012.
- [5] M.T. Kim and I.H. Jee, "H.264 Encoding Technique of Multi-view Image expressed by Layered Depth Image," The Journal of the Institute of Webcasting, Internet and Telecommunication, Vol. 10, No. 1, pp. 81-90, Feb. 2010.
- [6] <http://iphome.hhi.de/suehring/tmpl/>, "JVC Reference Software," version 9.6, May 2005
- [7] Y.K.Tu, J.F.Yang, M.T. Sun, "Fast Variable-Size Block Motion Estimation Using Merging Procedure With an Adaptive Threshold," Proc. IEEE International Conference on Multimedia & Expo, Bltmore, MD, USA, July 2003
- [8] M.Kucukgoz, M.T.Sun, "Early-stop and Motion Vector Reusing for MPEG-2 to H.264 Transcoding," SPIE Proc. Visual Comm. Image Proc. Sn Jose, CA, USA, Jan.2004.
- [9] Zhi Zhou, Ming-Ting Sun, Yuh-Feng Hu, "Fast Variable Block-size Motion Estimation Algorithms Based On Merge And Split Procedures For H.264/MPEG-4 AVC," IEEE IS-CAS 2004
- [10] X.Li, G.Wu, "Fast Interger Pixel Motion Estimation," JVT-F011, 6th Meeting, Awaji Island, Japan, Dec, 2002
- [11] Z.Chen, P.Zhou, Y.He, "Fast Motion Estimation for JVC," JVT-G016, 7th Meeting, Pattaya, Thailand, Mar. 2003
- [12] K.K.Ma, G.Qiu, "An Improved Adaptive Rood Pattern Search For Fast Block- matching Motion Estimation in JVC/H.26L," Proc. IEEE International Symposium on Circuits and Systems, Bangkok, Thailand, May 2003
- [13] JY.Tham, S.Ranganath, M.Ranganath, and A.AKassim, "Anovel unrestricted center-based diamond serch algorithm for block motion estimation," IEEE Trans, Circuits System Video Technol, vol 8, pp. 369-377, Aug 1998.
- [14] D.G.Song, "Coast Evaluation Techniques for Mode Selection in Video Coding", The Institute of Internet, Broadcasting and Communication, Vol 13, No.6, pp.275-280, Dec 2013.

저자 소개

송 대 건(정회원)

- 1982년 : 경희대 전자공학과 졸
- 1987년 : 와세다대학 대학원 졸(공학 석사)
- 1998년 : 경희대 대학원 전자공학과 졸(공학박)
- 1989년 ~ 2014년 현재 : 가천대학교 전자공학과 교수

<주관심분야 : 신호 및 영상처리, 통신시스템, 임베디드시스템 설계>