

## 공간적 상관성 기반 고속 참조영상 선택 방법

이상용, 김재곤, \*최해철  
한국항공대학교, \*한밭대학교

{ysyysy1921, jgkim}@kau.ac.kr, \*choihc@hanbat.ac.kr

## Spatial Correlation Based Fast Reference Frame Selection

Sang Yong Lee, Jae-Gon Kim, and \*Haechul Choi  
Korea Aerospace University, \*Hanbat National University

## 요 약

H.264/AVC 는 움직임 예측/보상을 수행할 때, 하나 이상의 다중 참조영상을 허용하여 예측 정확성을 높임으로써 높은 압축효율을 얻을 수 있지만, 부호화기가 높은 복잡도를 갖는 단점이 있다. 본 논문에서는 H.264/AVC 부호화기의 높은 복잡도를 줄이기 위하여 이미 부호화된 주변 블록의 부호화 정보와 현재 부호화하는 매크로블록(MB)의 16x16 화면간 예측 결과를 적용적으로 이용하여 현재 부호화하는 매크로블록의 참조영상 후보의 수를 줄이는 고속 부호화 알고리즘을 제안한다. 모의실험을 통하여 제안한 알고리즘은 JM17.2 에 비해 평균 47% 정도의 부호화 시간을 감소시키며, 이때 평균 비트율은 1.4%로 부호화 효율의 감소가 미미함을 확인 하였다.

## 1. 서론

H.264/AVC [1] [2]는 우수한 부호화 효율을 위하여 기존의 부호화 표준들과는 다른 기법들을 포함하고 있다. 대표적으로 화면간 부호화/복화화의 경우 움직임 예측/보상을 수행할 때, 가변 블록 크기, 화소 보간법을 통한 1/2, 1/4 화소 해상도, 그리고 다중 참조영상을 허용함으로써 부호화 효율을 높인다. 특히, 다중 참조영상 움직임 예측/보상의 경우, 기존과는 달리 하나 이상의 복수개의 참조영상을 이용함으로써 가려진 사물이나 배경 그리고 밝기 변화가 있는 영역에 대해서 보다 정확한 예측을 함으로써 높은 부호화 효율을 얻을 수 있지만, 최적의 참조영상을 선택하기 위한 복잡도 또한 상당히 증가하게 된다. 그리고 정적이거나 평탄한 영역에서의 다중 참조영상 움직임 예측/보상은 부호화 효율보다 복잡도 증가 정도가 훨씬 크기 때문에 이런 경우, 다중 참조영상을 이용하는 것이 오히려 비효율적일 수 있다.

본 논문에서는 화면간 부호화에서 움직임 예측/보상을 수행할 때, 현재블록과 주변블록의 상관성을 이용하여 현재블록의 참조영상 후보의 수를 줄인다. 더불어 제안하는 알고리즘의 적용으로 참조영상의 후보의 수가 계속해서 작게 선택되도록 전파되는 것을 방지하기 위하여 16x16 화면간 예측은 다른 알고리즘을 적용하고, 현재 부호화하는 MB 의 16x16 보다 작은 크기를 가지는 블록의 화면간 예측에서 16x16 화면간 예측 결과를 이용하게 함으로써 성능저하를 최소화 한다.

본 논문의 제 2 장에서는 주변블록과 16x16 화면간 예측 결과를 이용하여 참조영상 후보의 수를 제한하는 알고리즘을

기술하고, 제 3 장에서는 제안한 알고리즘을 적용한 실험결과를 분석하고, 마지막으로 제 4 장에서 결론을 맺는다.

## 2. 고속 참조영상 결정 방법

H.264/AVC 에서는 현재 MB 의 움직임 벡터 부호화를 위하여 예측 움직임 벡터(PMV: Predictive Motion Vector)를 유도할 때, 이미 부호화된 주변블록의 움직임 정보를 이용한다. 이는 현재 블록과 이에 인접한 블록들은 움직임 정보에 대한 상관성이 매우 높기 때문이다. 더 나아가 주변블록의 참조영상이 모두 같으면 현재 MB 도 주변에서 사용하는 참조영상을 사용할 확률이 크다고 볼 수 있다. 따라서 본 논문에서는 주변블록의 움직임 정보를 보고 현재 부호화하고자 하는 MB 의 참조영상 후보의 수를 줄이는 기법을 제안한다.

본 제안방법은 16x16 화면간 예측이나 16x16 보다 작은 크기를 가지는 블록의 화면간 예측이나에 따라 다른 알고리즘을 적용한다. 이는 제안하는 알고리즘을 모든 가변 블록 크기 움직임 예측/보상에 동일하게 적용함으로써 참조영상의 후보가 계속해서 작게 선택되는 것을 방지하기 위함이다. 예를 들어, 참조영상의 수를 5 로 설정을 하고 알고리즘을 적용을 하였을 때, 알고리즘의 적용으로 참조영상 후보의 수가 1 로 제한이 되었을 때, 현재 슬라이스 이하 모든 MB 에 대해서도 참조영상 후보의 수가 1 로 선택되는 전파 문제가 발생할 수 있고 이런 경우, 상당한 성능저하를 초래하게 된다. 이러한 문제를 방지하기 위하여 상기와 같이 블록 크기에 따라 두 가지 알고리즘이 적용된다.

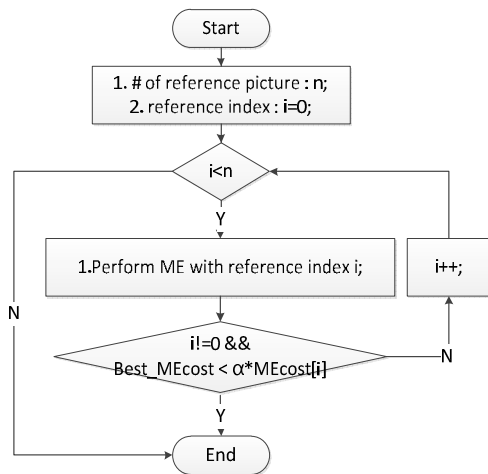


그림 1. 16x16 화면간 예측에서 참조영상 후보 제한 방법

먼저, 16x16 화면간 예측에서 참조영상의 후보의 수를 줄이는 알고리즘은 그림 1 과 같다. 위 그림과 같이, 움직임 예측을 수행하는 전반적인 과정은 H.264/AVC 와 유사하다. 설정된 참조영상의 수만큼 0 의 색인 값을 가지는 참조영상부터 움직임 예측을 수행하면서 가장 좋은 움직임 예측 비용을 저장한다. 그리고 1 이상의 색인 값을 가지는 참조영상에 대한 움직임 예측이 끝나면 현재 움직임 예측 비용과 가장 좋은 움직임 예측 비용을 비교한다. 즉, 현재 움직임 예측 비용의  $\alpha$  ( $0 \sim 1$ ) 배에 해당하는 값이 가장 좋은 움직임 예측 비용보다 작으면, 계속하여 다음 색인 값을 가지는 참조영상에 대해서 움직임 예측을 수행하고 그렇지 않는 경우, 현재 16x16 움직임 예측은 다음 색인 값의 참조영상에 대한 움직임 예측을 수행하지 않고 종료된다. 그러므로 16x16 화면간 예측의 경우 허용되는 참조영상의 수는 움직임 예측 비용에 따라서 최소 1 에서부터 최대 설정된 참조영상의 수의 값을 가질 수 있다.

다음으로 16x16 보다 작은 크기 블록의 화면간 예측에서 참조영상 후보의 수를 줄이는 알고리즘에 대해서 설명한다. 여기서 16x16 보다 작은 크기 블록은 16x8, 8x16, 8x8, 8x4, 4x8, 4x4 의 크기를 가지는 블록들이 해당된다.

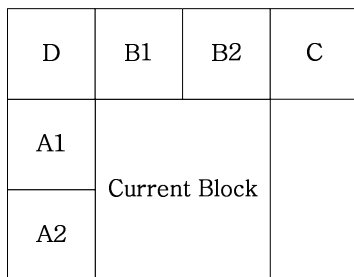


그림 2. 현재 MB 와 인접한 6 개의 8x8 블록

본 제안 방법에서는 PMV 를 유도할 때와는 다르게 그림 2 와 같이 현재 MB 주변의 이미 부호화된 6 개의 8x8 블록들의 움직임 정보를 이용한다. 이는 더 많은 정보에 기반하여 통계적으로 정확도를 높임으로써, 더욱 정확한 참조영상 후보의 수를 결정하기 위함이다. 여기서 현재 MB 의 왼쪽 MB 가 16x16 모드로 부호화 되었을 때, A1 과 A2 블록은 같은

움직임 정보를 가지고 각각 독립적으로 사용된다. 그리고 만약 8x8 의 A1 블록이 4 개의 4x4 블록으로 부호화 되었을 때, A1 블록에서 오른쪽 위쪽에 위치한 4x4 블록의 움직임 정보를 이용한다. 16x16 블록을 제외한 상기 가변 블록들의 화면간 예측에서 참조영상 후보의 수를 줄이는 방법은 그림 3 과 같다.

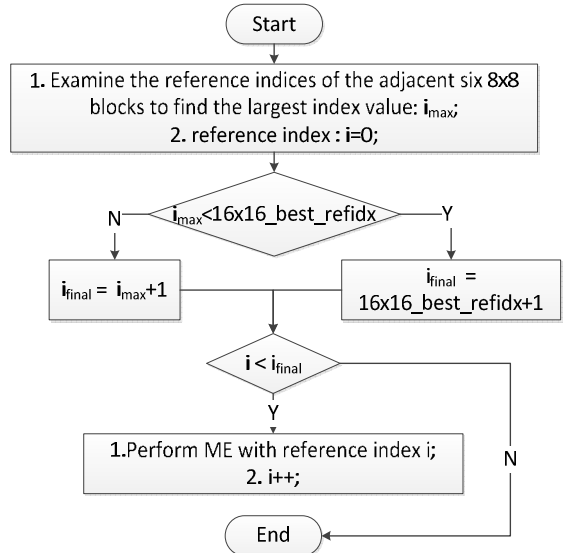


그림 3. 16x16 보다 작은 크기 블록의 화면간 예측에서 참조영상 후보 제한 방법

위의 그림과 같이, 현재 블록의 움직임 예측을 수행하기 전에 이미 부호화된 주변 6 개의 8x8 블록들에서 사용된 참조영상 색인 값의 최대 값을 찾는다. 그리고 그 값을 이전 16x16 화면간 예측에서 선택된 최적의 참조영상의 색인 값과 비교를 한다. 인접한 주변 블록에서 유도한 참조영상 색인 값이 16x16 화면간 예측에서 선택된 참조영상의 색인 값보다 크면, 현재 블록의 움직임 예측에 사용되는 참조영상 후보의 수는 인접한 주변 블록에서 유도한 참조영상 색인 값 + 1 로 제한이 되고 그렇지 않은 경우, 현재 블록의 움직임 예측에 사용되는 참조영상 후보의 수는 16x16 화면간 예측에서 선택된 참조영상의 색인 값 + 1 로 제한된다. 상기와 같은 방법으로 참조영상 후보의 수가 제한되면, 현재 부호화 되는 블록은 0 의 색인 값을 가지는 참조영상부터 제한된 참조영상 후보의 수만큼 움직임 예측을 수행한다.

상기 제안된 2 가지 알고리즘에서 16x16 화면간 예측의 경우, 특정한 조건을 만족하지 않는 경우 최대 설정된 참조영상의 수만큼 움직임 예측을 수행하여 최적의 참조영상을 선택하고, 그 결과를 16x16 보다 작은 크기 블록의 화면간 예측에서 이용함으로써, 제안하는 알고리즘이 적용되어 한 슬라이스의 모든 MB 의 참조영상 후보 수가 1 로 선택되는 전과 문제를 방지함으로써 성능저하를 최소화 할 수 있다.

### 3. 실험결과

제안하는 알고리즘을 H.264/AVC 의 참조 소프트웨어인 JM17.2 에 구현하였다. 실험환경은 표 1 과 같으며, QP(P Slice) 는 QP(I Slice)+1 하였다. 부호화 시간 변화량은 다음 식을 이용하여 백분율로 나타내었다. 여기서  $T_{JM}$  는 JM17.2 로 부호화

했을 때, 전체 부호화 소요시간이고,  $T_{Prop}$  는 제안한 알고리즘을 JM17.2 에 적용하여 부호화 하였을 때, 전체 부호화 소요시간을 나타낸다.

$$\Delta T(\%) = \frac{T_{JM} - T_{Prop}}{T_{JM}} \times 100$$

부호화 성능은 BD-Rate 과 BD-PSNR [3]을 사용하였다.

표 1. 실험환경

| Profile                   | High Profile     |
|---------------------------|------------------|
| Coding Structure          | IPPP             |
| Number of Frame           | 100              |
| QP(I Slice)               | 22, 27, 32, 37   |
| Search Mode               | Fast Full Search |
| Search Range              | ±32              |
| Number of Reference Frame | 5                |

테스트 시퀀스는 4CIF 해상도를 가지는 Soccer, Harbour 와 HEVC 표준화 시퀀스 중 BasketballDrill, BQMall(C Class), BlowingBubbles, RaceHorse(D Class), Johnny, Fourpeople (E Class) [4]를 사용하였다. 표 1 과 표 2 는 16x16 화면간 예측에 사용되는 변수  $\alpha$ 의 값을 각각 0.7, 0.9 로 설정하였을 때, 이들 테스트 시퀀스에 대한 부호화 및 복잡도 성능을 나타낸 것이다.

표 2. 실험결과 ( $\alpha=0.7$ )

| Sequence(Resolution)      | BD-PSNR (dB) | BD-Rate (%) | $\Delta T(\%)$ |
|---------------------------|--------------|-------------|----------------|
| BasketballDrill (832x480) | -0.06        | 0.8%        | 38%            |
| BQMall (832x480))         | -0.05        | 1.7%        | 36%            |
| BlowingBubbles (416x240)  | -0.06        | 1.4%        | 35%            |
| RaceHorse (416x240)       | -0.12        | 2.1%        | 32%            |
| Johnny (720p)             | -0.05        | 1.3%        | 48%            |
| Fourpeople (720p)         | -0.02        | 0.7%        | 46%            |
| Soccer(4CIF)              | -0.06        | 0.9%        | 39%            |
| Harbour(4CIF)             | -0.09        | 1.8%        | 41%            |
| <b>Average</b>            | <b>-0.06</b> | <b>1.3%</b> | <b>39%</b>     |

표 3. 실험결과 ( $\alpha=0.9$ )

| Sequence(Resolution)      | BD-PSNR (dB) | BD-Rate (%) | $\Delta T(\%)$ |
|---------------------------|--------------|-------------|----------------|
| BasketballDrill (832x480) | -0.07        | 0.8%        | 45%            |
| BQMall (832x480))         | -0.06        | 1.8%        | 47%            |
| BlowingBubbles (416x240)  | -0.06        | 1.3%        | 41%            |
| RaceHorse (416x240)       | -0.13        | 2.4%        | 45%            |
| Johnny (720p)             | -0.07        | 1.5%        | 55%            |
| Fourpeople (720p)         | -0.04        | 0.7%        | 52%            |
| Soccer(4CIF)              | -0.05        | 1.0%        | 46%            |
| Harbour(4CIF)             | -0.07        | 2.0%        | 48%            |
| <b>Average</b>            | <b>-0.07</b> | <b>1.4%</b> | <b>47%</b>     |

표 2, 3 을 통해  $\alpha$ 의 값을 0.7 로 설정한 경우, 1.3%의 미미한 성능 저하로 부호화 시간은 39% 감소할 수 있었다. 그리고  $\alpha$ 의 값을 0.9 로 설정한 경우, 부호화 시간을 47% 감소시켰으며, 1.4%의 성능 저하를 보였다. 위의 실험 결과를 통해,  $\alpha$ 의 값을 크게 할 수록 미미한 성능저하로 더 많은 부호화 시간을 감소 시킴을 확인하였다. 또한  $\alpha$ 의 값과 상관없이 E Class 와 같이 움직임이 적고 평탄한 영역이 많이 포함되어 있는 시퀀스의 경우, 미미한 성능저하로 더 많은 복잡도를 감소시킴을 확인하였다. 이를 통해, 제안한 알고리즘은 비디오 부호화기의 부호화 시간을 상당히 감소시키며, 이는 다른 고속 움직임 예측 알고리즘과 함께 사용될 수 있다.

#### 4. 결론

본 논문은 H.264/AVC 부호화기의 화면간 부호화에 있어 이미 부호화된 주변블록들의 움직임 정보와 현재 부호화하는 MB 의 16x6 화면간 예측결과를 적응적으로 이용하여 참조영상 후보의 수를 줄임으로써, 현재 MB 의 움직임 예측 수행 시 복잡도를 감소시키는 방법을 제안하였다. 실험 결과  $\alpha$ 를 0.9 로 설정하였을 때, 부호화 시간을 약 47% 감소할 수 있었으며, 제안방법으로 인한 PSNR 손실과 비트율 증가는 평균 -0.07dB 와 1.4%로 기존 방법에 비해 부호화 효율의 성능저하가 미미함을 확인하였다.

#### ACKNOWLEDGMENT

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터육성 지원사업(NIPA-2012-H0301-12-1006) 및 교육과학기술부의 재원으로 한국연구재단의 기초연구사업(No. 2011-0023182)의 지원으로 수행되었음.

#### 참고 문헌

- [1] ITU-T Recommendation H.264 and ISO/IEC 14496-10, "Advanced video coding for generic audiovisual services", May 2003.
- [2] Thomas Wiegand, Gary J. Sullivan, Gisle Bjntegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., pp. 560-576, July 2003.
- [3] Gisle Bjntegaard, "Improvements of the BD-PSNR Model," ITU-T SG16/Q6, 35th VCEG Meeting, Berlin, Germany, Doc. VCEG-AI11, 16-18 July, 2008.
- [4] JCTVC, "Common test conditions and software reference configurations," JCTVC-H1100, San Jose, CA, USA, 1-10 February, 2012.