

H.264/AVC를 위한 통계 특성 기반 정수 화소 단위 고속 움직임 예측 기법

노진영*, 홍민철^o

Fast Integer-Pel Motion Estimation Based on Statistical Property for H.264/AVC

Jin-young Noh*, Min-cheol Hong^o

요 약

본 논문에서는 H.264/AVC 동영상 부호화 방식을 위한 국부 움직임 벡터 통계 특성을 활용한 정수 화소 단위의 고속 움직임 벡터 예측 기법에 대해 제안한다. 인접 블록들의 움직임 벡터에 의한 통계 특성을 정의하고, 이에 따라 현재 부호화 블록의 움직임 검색 영역의 모드를 결정하는 방식에 대해 기술한다. 더불어, 통계 기반의 움직임 영역 예측에 의한 문제점을 해결하기 위하여 현재 블록과 이전 부호화 블록들의 동일 움직임 검색 영역 모드에 대한 누적 통계를 기반으로 하는 움직임 영역 적응적으로 보정 기법에 대해 제안한다. 실험 결과를 통해 기존 방식과 비교하여 제안 방식의 연산량 절감의 우수성을 확인할 수 있었다.

Key Words : H.264/AVC, 움직임 벡터, 인접 블록, 움직임 검색 영역 모드, 누적 통계

ABSTRACT

In this paper, we propose an efficient fast integer-pel motion estimation for H.264/AVC using local statistics of local motion vectors. Using neighboring motion vectors, we define a new statistical property that is used to determine a mode of motion search range of current block. In addition, an adaptive motion search range compensated method that is based on cumulative statistics of previous coded blocks is addressed to solve the problem of the statistical motion search range decision method. Experimental results show that proposed algorithm has the capability to reduce the computational cost over the other methods.

I. 서 론

ITU와 ISO/IEC가 공동으로 진행한 H.264/AVC 동영상 표준 부호화 방식은 다양한 예측 부호화 기법과 엔트로피 부호화 방법들을 통해 기존 비동영상 부호화 방식들과 비교하여 최대 50%까지의 비트율을 절감할 수 있는 효과적인 방식으로 평가받고 있다^[1-8].

기존 동영상 표준 부호화 방식과 비교하여 H.264/AVC는 가변 블록 크기 기반의 움직임 예측 기법을 사용하고 있으며, 이를 통해 영상의 시간적 중복성을 제거를 극대화하고자 하였다. 가변 블록 단위의 움직임 예측 기법은 영상 간 일정한 검색영역 내에서 다양한 블록 크기에 대해 적합성을 비교하여 가장 유사한 블록을 예측한다. 이와 같은 가변 블록 단위의 움직임 예측 과정은 기존 방

※ 본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT 연구센터 육성지원 사업의 연구결과로 수행되었음 (NIPA-2012-H0301-12-2006)

♦ 주저자 : 송실대학교 대학원 정보통신공학과 영상처리연구실, bo2sto.ssu.ac.kr, 학생회원

o 교신저자 : 송실대학교 정보통신전자공학부, mhong@ssu.ac.kr, 종신회원

논문번호 : KICS2012-07-304, 접수일자 : 2012년 7월 5일, 최종논문접수일자 : 2012년 7월 27일

식과 비교하여 많은 연산량을 필요로 하게 되어 다양한 응용 서비스로의 활용에 한계성을 갖게 된다. 그러므로 H.264/AVC 동영상 표준 부호화 과정에서 움직임 예측 과정의 계산량 감소 기법에 대한 연구는 매우 중요한 내용으로 인식되고 있다.

움직임 예측의 연산량을 개선하기 위한 많은 고속 움직임 예측 방식이 개발되어 왔으며 전통적인 고속 움직임 예측 기법들은 인접 블록의 움직임 벡터나 이전 영상의 동일 위치 블록의 움직임 벡터를 고려하여 검색영역을 적응적으로 조절하여 연산량을 절감하고자 하였다⁹⁻¹²⁾. 인접 블록의 움직임 벡터 정보만을 고려하여 검색영역을 결정하는 방식은 시간에 따른 움직임 벡터의 변화를 반영할 수 없게 된다. 이와 같은 문제점은 정교하지 못한 검색영역 설정으로 불필요한 검색영역이 설정되어 계산량을 증대시키거나 필요한 검색영역을 검색하지 못하게 되어 압축효율을 떨어뜨리는 단점을 갖고 있다. 반면에 이전 영상의 동일 위치 블록의 움직임 벡터를 참조하는 경우 이전 영상의 모든 움직임 벡터를 저장해야하므로 많은 메모리를 요구하게 된다.

본 논문에서는 상기와 같은 문제점을 해결하기 위하여 인접 블록의 움직임 벡터의 통계 특성을 세분화하고 이를 이용한 고속 정수-화소 움직임 예측 기법을 제안한다. 특히 인접 블록의 움직임 벡터 특성에 따라 해당 움직임 벡터 예측 블록의 특성을 예측하고, 예측된 블록의 특성 및 움직임 벡터의 누적 통계에 따라 검색영역을 설정한다. 설정된 검색영역에서 선택적인 가변 단계 검색 기법과 초기 종료 기법을 이용하여 기존의 기법들과 비교하여 계산량이 절감된 화소 단위의 움직임 예측을 수행하게 된다. 움직임 예측을 절감하기 위한 고속 부화소(Sub-pixel) 움직임 예측 기법이나 모드 결정 기법 등이 있으나¹³⁻¹⁶⁾, 본 논문에서는 화소 단위의 움직임 예측을 기반으로 한 고속 움직임 예측 방법을 다루고자 하였다.

본 논문의 구성은 다음과 같다. II장에서는 기존 움직임 예측 기법의 배경 이론에 대해 소개하고, III장에서는 움직임 벡터의 통계 특성 및 영상의 특성을 반영한 효과적인 고속 화소 단위의 움직임 예측 기법을 기술한다. IV장에서는 제안한 고속 움직임 예측 기법의 실험 결과에 대해 비교 분석하며, 마지막으로 V장에서는 본 논문의 결론을 맺는다.

II. 배경 이론

H.264/AVC을 포함한 대부분의 동영상 표준 부호화 방식은 블록 매칭 (Block Matching) 기반으로 일정 움직임 검색 영역 내에서 움직임 벡터 예측을 수행하게 된다. 일반적으로 움직임 벡터 예측 과정은 다음과 같이 수행된다.

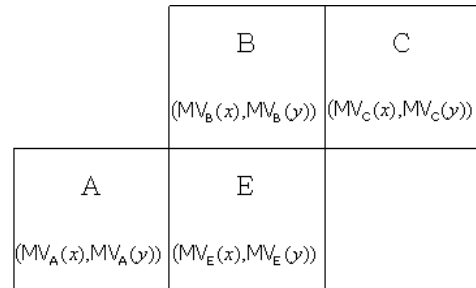


그림 1. 현재 블록과 인접블록의 위치 관계
Fig. 1. Position of current and neighbor blocks

그림 1에 도시된 바와 같이 블록 E가 해당 부호화 블록인 경우 A, B 및 C는 이미 움직임 벡터가 예측된 인접 블록에 해당되며 블록 E의 움직임 벡터 예측 원점은 예측 움직임 벡터 (PMV: Predictive Motion Vector)로 정의되며 다음과 같이 결정된다.

$$PMV(i) = median(MV_A(i), MV_B(i), MV_C(i)) \quad (1)$$

위 식에서 $i = x, y$ 로 정의되며, x, y 는 움직임 벡터의 수평, 수직방향 성분을 나타낸다. 더불어 median(.)은 입력 값 중에서 중앙값을 출력하는 함수를 의미한다.

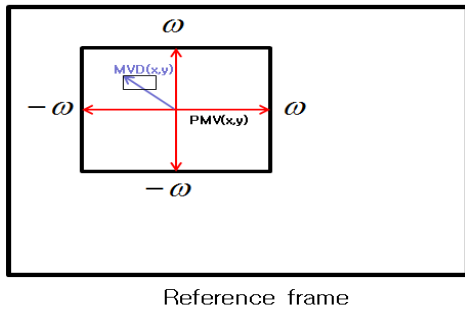
해당 블록의 움직임 벡터 예측은 PMV를 중심으로 수직, 수평 방향으로 사전에 설정된 검색 영역(ω)내에서 그림 2와 같이 수행되게 된다.

$$\begin{aligned} -\omega \leq dx \leq \omega \\ -\omega \leq dy \leq \omega \end{aligned} \quad (2)$$

식 (2)에서 dx 및 dy 는 움직임 예측을 위한 해당 블록의 수평 및 수직 방향의 움직임 검색 영역을 의미한다. 식 (2)와 같이 설정된 검색 영역 내에서 (U,V)번째 $M \times N$ 크기 블록의 움직임 벡터는 다음과 같이 정의된 SAD(Sum of Absolute Difference)의 최소값을 갖는 위치에 의해 결정된다.

이 논문은 RDO(Rate Distortion Optimization) off 환경에서만 실험하였기 때문에, RDO on 환경에서 사용하는 Rate Distortion cost (RDcost)에 대

한 기술은 생략하고 SAD에 대해서만 기술한다.



Reference frame

그림 2. 참조프레임 움직임 검색 영역 설정 예
Fig. 2. Example of motion search range of reference frame

$$SAD_{dx,dy} = \min_{dx,dy} \sum_{j=0}^{N-1} \sum_{i=0}^{M-1} |C(k,l) - R(k+m,l+n)| \quad (3)$$

$$k = U \times M + i, l = V \times N + j$$

$$m = dx + PMV(x), n = dy + PMV(y)$$

식 (3)에서 C 및 R 은 움직임 예측을 위한 현재 영상 및 해당 참조 영상을 의미한다. 움직임 검색 영역 내에서 SAD값이 가장 작은 (dx, dy) 를 결정하여 움직임 벡터 차이 (MVD : Motion Vector Difference)를 생성하여 전송하게 된다. 움직임 벡터 차이 값은 다음과 같이 결정된다.

$$MVD(i) = MV(i) - PMV(i) \text{ for } i = x, y \quad (4)$$

III. 제안하는 고속 움직임 예측 기법

기존 움직임 예측 기법들의 연산량 문제점을 해결하기 위해 본 논문에서는 인접 블록의 움직임 벡터 뿐 아니라 영상의 특성을 반영하여 저연산량의 고속 움직임 예측을 할 수 있는 기법을 제안한다. 더불어 고속 움직임 예측을 위하여 통계 특성에 따른 검색 영역 설정 방식 및 조기 종료 기법에 대해 기술한다. 제안하는 방식은 4개의 주요 처리 단계로 이루어져 있다. (1) 인접 블록을 이용한 통계적 특성 분류, (2) 통계적 특성 분류에 따른 현재 블록의 검색 영역 결정, (3) 선택적 단계별 검색 및 (4) 조기 종료 과정으로 이루어진다.

3.1. 인접 블록을 이용한 통계적 특성 분류

고속 움직임 예측을 위하여 인접 블록의 움직임 벡터값에 따라 현재 블록의 움직임 검색 영역 결정을 위한 통계적 특성을 결정한다. 참고문헌 [9]에

기술된 기법은 인접 블록의 값이 매우 작을 때와 방향성이 같을 때, 그 이외의 경우로 나누어 검색영역을 적응적으로 설정하였다. 그러나 기존 방식은 검색 영역의 세분화가 이루어지지 않아서 인접 블록 움직임 벡터 특성에 따른 해당 블록의 움직임 검색 영역을 결정하는데 한계성이 존재하고 이로 인해 불필요한 영역을 검색하는 문제점이 있다. 이와 같은 문제점을 해결하기 위해 본 연구에서는 인접 블록의 통계적 특성을 세분화하여 분류된 특성에 따라 검색 영역을 제어하는 방식에 대해 기술하고자 한다.

그림 3은 인접 블록의 움직임벡터에 따른 현재 블록 MVD의 절대 값을 나타낸 누적 분포함수 (CDF : Cumulative Distribution Function)의 예를 나타내고 있다. 위 그림에서 확인할 수 있듯이 인접 블록의 움직임 벡터가 모두 '0' 인 경우 ($MV_A = MV_B = MV_C = 0$) 해당 부호화 블록의 움직임 벡터 값이 상대적으로 작은 값을 가질 확률이 매우 높은 것을 알 수 있으며, 인접 블록의 움직임 벡터가 동일한 방향성을 갖는 경우 ($\text{sgn}(MV_A) = \text{sgn}(MV_B) = \text{sgn}(MV_C)$)도 다른 경우와 비교하여 해당 부호화 블록의 움직임 벡터 값이 작게 나타날 확률이 높은 것으로 예측할 수 있다. 이와 같은 특성은 인접 블록 움직임 벡터들의 특성에 따라 움직임 검색 영역을 가변적으로 설정하여 불필요한 영역의 검색 영역 배제하고, 이를 통해 연산량 감소를 유도할 수 있음을 알 수 있다.

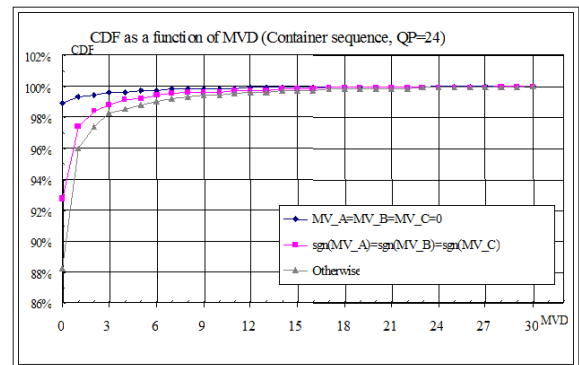


그림 3. 인접 블록 MVD특성에 따른 CDF
Fig. 3. CDF according to MVD of neighbor blocks

상기 분석 결과로부터 인접 블록의 움직임 벡터 값들과 PMV와의 차이를 움직임 벡터의 균일 정도의 표현 및 움직임 검색 영역을 결정하는 통계 특성으로 활용할 수 있음을 확인할 수 있다. 본 논문에서는 움직임 검색 영역을 결정하는 특성으로

SDMV (Sum of absolute Difference Motion Vector)를 사용하였으며 다음과 같이 정의된다.

$$SDMV(i) = \sum_{k=A}^C |PMV(i) - MV_k(i)| \quad (5)$$

식 (5)에서 $MV_k(i)$ 에 대하여 $k = A, B, C$ 에 대한 움직임 벡터를 나타내며, i 는 수평 또는 수직 방향성을 의미한다. 인접 블록 움직임 벡터의 밀집 정도를 나타내는 SDMV는 PMV를 기준으로 인접 블록의 움직임이 균일한 경우 매우 작은 값을 갖게 되며 인접 블록 움직임 벡터 방향성이 다르거나 불균일한 경우 매우 큰 값을 갖는 특성을 나타낸다. 상기와 같은 특성을 이용하여 최초에 설정된 움직임 검색 영역 ω 에 대하여 SDMV에 따른 보정된 검색영역을 설정하여 해당 부호화 블록의 움직임 검색 영역을 표 1과 같이 통계 특성에 따라 가변적으로 설정한다.

표 1에서 SR_{min} 및 SR_{max} 는 통계 특성에 따라 정의된 최소 및 최대 검색 영역을 의미하며 수평 및 수직 방향의 검색 영역은 독립적으로 결정된다.

위와 같이 인접 블록의 통계 특성을 이용한 검색 영역 설정은 영상의 움직임 특성을 반영하기에 한계가 있으며, 이로 인해 움직임 예측 오류 등의 문제점을 갖고 있다. 본 논문에서는 해당 부호화 영상의 누적 통계 특성을 결합하여 위와 같은 문제점을 해결하고자 하였다.

표 1. SDMV에 따른 모드 구분 및 움직임 검색 영역
Table 1. Mode classification and motion search range in accordance with SDMV

모드 (k)	상태	SR_{min}^k	SR_{max}^k
1	$SDMV = 0$	0	$\omega/4$
2	움직임벡터 동일 방향 이고, $SDMV \leq 1$	$\omega/16$	$\omega/4$
3	움직임벡터 동일 방향이고, $SDMV > 1$	$\omega/8$	$\omega/2$
4	otherwise	$\omega/8$	ω

3.2. 누적 통계 특성을 이용한 검색 영역 보정

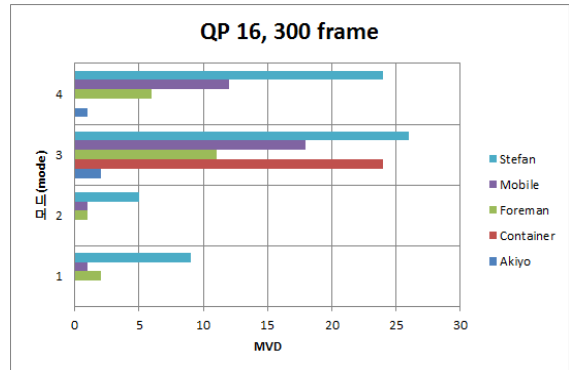


그림 4. 모드에 따른 MVD분포 예
Fig. 4. Example of MVD distribution according to mode classification

그림 4에 도시된 인접 블록 통계 특성에 의해 결정된 모드에 따른 MVD 분포도의 예로부터 Stefan, Mobile과 같이 움직임이 큰 영상은 각 모드에 대한 MVD 변화가 크며 Akiyo, Container와 같이 움직임이 작은 영상에서는 MVD 변화가 작음을 알 수 있다. 이와 같은 분석 결과로부터 영상의 움직임 국부 통계 및 MVD 통계 분포도 움직임 검색 영역을 결정하기 위해 효과적으로 활용될 수 있음을 알 수 있다.

그림 5는 동일 모드로 결정된 블록들의 MVD 분포 누적 통계 예로서, 그림 5의 점들은 현재 움직임 예측 블록과 동일 모드로 부호화된 이전 블록들의 MVD 값의 분포를 나타낸다. MVD 분포가 원점에 밀집될수록 동일한 모드의 이전 블록들의 MVD의 상관성이 높은 것으로 판단할 수 있으며, 밀집 정도가 작을수록 블록들 간의 MVD 상관성이 낮은 것을 의미한다. 움직임 벡터 예측은 PMV를 기준으로 수행되는 관계로 위와 같은 특성은 움직임 검색 영역을 적응적으로 결정하기 위해 효과적으로 활용될 수 있다.

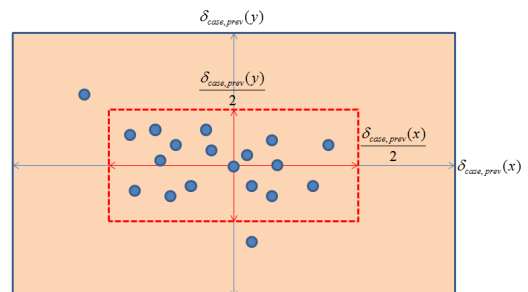


그림 5. 모드별 MVD 누적 통계 예
Fig. 5. Example of MVD CDF of a mode

상기와 같은 특성을 기반으로 k 모드로 결정된

현재 블록의 최종 검색 영역 $\delta_{cur}^k(i) (i=x,y)$ 은 표 1에 기술된 각 모드별 검색 영역과 식 (6)에 기술된 k 모드로 결정된 M 개의 이미 부호화된 이전 블록의 MVD 누적 통계 특성에 의해 결정된다. MVD 누적 통계 특성에 의한 검색 영역 결정은 다음과 같이 이루어진다.

$$\delta_{cur,CDF}^k(i) = \begin{cases} \frac{\delta_{prev}^k(i)}{2} & \text{if } \Phi(MVD_j^k(i), \delta_{prev}^k(i)/2) \geq T_1 \\ 2 \times \delta_{prev}^k(i) & \text{if } \Phi(MVD_j^k(i), \delta_{prev}^k(i)/2) \leq T_2 \\ \delta_{prev}^k(i) & \text{otherwise} \end{cases} \quad (6)$$

for $j = (1, \dots, M)$.

식 (6)에서 $\delta_{cur,CDF}^k(i)$ 는 현재 블록의 MVD 분포 누적통계를 이용한 검색영역이다. $\Phi(MVD_j^k(i), \delta_{prev}^k(i)/2)$ 는 이전 블록들 중 통계 특성 모드가 k 인 M 개의 블록들 중, 해당 블록의 $MVD_j^k(i)$ 가 $\delta_{prev}^k(i)/2$ 이하를 만족하는 블록의 수를 의미하며, T_1 및 T_2 는 각각 임계값을 의미한다. 더불어 $\delta_{prev}^k(i)$ 는 k 모드로 결정된 이전블록의 최종 검색영역을 의미하며 초기값은 표 1의 SR_{max}^k 값이 된다. 예를 들어, 현재 블록이 표 1에 의해 모드가 1로 결정이 된 경우, $\delta_{prev}^k(i)$ 는 $\omega/4$ 가 된다. 식 (6)은 현재 블록과 동일한 국부 특성 모드로 부호화된 블록들의 MVD 밀집도를 평가하기 위한 식으로서, 일정 영역 내에서 동일한 모드로 결정된 블록 수가 많고 해당 블록의 MVD가 작은 값으로 밀집되어 있을수록 현재 블록의 MVD가 이미 부호화된 이전 블록들과 유사한 움직임 벡터 특성을 가질 확률이 높게 된다.

표 1에 정의된 특성 특성에 따른 모드 분류에서 모드 1이고 MVD 누적 통계 분포에 의한 $\delta_{cur,CDF}^k(i)$ 도 '0'인 경우, 식 (10)에서 기술된 최종 검색 영역이 '0'으로 결정되어 현재 블록 이후 블록에 대해 식 (6)의 조건 검색 변수인 $\delta_{prev}^k(i)/2$ 가 '0'으로 설정되어 누적 분포 특성에 의한 움직임 검색 영역 설정에 오류를 발생하게 된다. 이와 같은 문제점을 해결하기 위해 $\delta_{cur,CDF}^k(i)$ 가 '0'인 경우에 대해 다음 조건을 만족하는 경우 $\delta_{cur,CDF}^k(i)$ 를 보정하게 된다. 보정 결과는 k 모드로 결정된 현재 블록과 이전 블록의 SAD값을 비교하여 다음과 같이 이루어지게 된다.

$$\delta_{cur,CDF}^k(i) = \begin{cases} 1 & \text{if } 2 \times SAD_{prev}^k \leq SAD_{cur}^k \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

표 1에 기술된 인접 블록의 국부 통계 특성에 의해 움직임 검색 영역과 식 (6) 및 (7)에 기술된 누적 MVD 특성에 의한 움직임 검색 영역 보정 결과에 의해 현재 블록을 위해 움직임 검색 영역은 다음과 같이 설정된다.

$$\delta_{cur}^k(i) = \min(\max(\delta_{cur,CDF}^k(i), SR_{min}^k(i)), SR_{max}^k(i)) \quad (8)$$

위 식에서 $\min(\cdot)$ 및 $\max(\cdot)$ 는 각기 최소 및 최대값을 나타내는 연산자이다. $\delta_{cur}^k(i)$ 는 다음 모드가 k 인 블록에서 식 (6)을 이용하여야 하므로, $\delta_{prev}^k(i)$ 에 $\delta_{cur}^k(i)$ 을 저장한다.

$$\delta_{prev}^k(i) = \delta_{cur}^k(i) \quad (9)$$

식 (8)에서 설정된 검색 영역을 이용하여 수평 및 수직 방향의 움직임 벡터 결정은 다음과 같은 검색 영역에서 이루어지게 된다.

$$\begin{aligned} -\delta_{cur}^k(x) &\leq dx \leq \delta_{cur}^k(x) \\ -\delta_{cur}^k(y) &\leq dy \leq \delta_{cur}^k(y) \end{aligned} \quad (10)$$

3.3. 선택적 단계별 검색

III장 1절 및 2절에서 기술한 움직임 검색 영역 결정 방식을 통해 연산량 절감을 할 수 있으나 결정된 검색 영역 내의 모든 화소에 대해서 움직임 벡터 결정을 위해 SAD연산을 수행하기에는 연산량의 절감에 한계가 따른다. 이와 같은 문제점을 해결하기 위해 참고문헌 [9]에 기술된 가변 단계별 검색 기법(VSS: Variable Step Search)을 활용하였으며 십자형을 추가하여 선택적인 가변 단계별 검색기법을 적용하여 연산량을 급격히 줄이고자 하였다. 가변 단계별 검색 기법은 다음과 같이 수행된다.

과정1: 최종 결정된 검색 영역($\delta_{cur}^k(i)$)에 대하여 최대 검색 단계와 스텝 크기($Step_Size(i)$)를 다음과 같이 결정한다.

$$\begin{aligned} \max_{step}(i) &= \max_{M(i)} [2^{M(i)} < \delta_{cur}^k(i)] + 1, \\ Step_Size(i) &= 2^{M(i)}. \end{aligned} \quad (11)$$

식 (10)에서 $\lfloor A \rfloor$ 는 A 보다 크지 않은 최대 정수값을 표현한다.

과정2: 검색 원점을 중심으로 $Step_Size(i)$ 거리에 위치한 8개의 지점을 선택하여 검색 원점 및 8개 선택 지점에 대한 SAD 연산을 수행 후 SAD 값이 최소가 되는 지점을 다음 단계의 중심으로 결정한다.

과정3: $Step_Size(i)$ 를 반으로 줄인 후 2단계 과정을 $Step_Size(i)$ 가 수평 및 수직 방향에 대해 각각 '1'일 될 때까지 과정2를 반복 수행한다.

$\delta_{cur}^k(x) = 17, \delta_{cur}^k(y) = 11$ 인 경우에 대한 가변 단계별 검색 기법의 예를 그림 6에 나타내었다. 그림 6의 경우에 최대 검색 단계는 $\max_{step}(x) = 5, \max_{step}(y) = 4$ 가 되며 수평 및 수평 방향에 대한 초기 스텝 크기는 $Step_Size(x) = 16$ 이고, $Step_Size(y) = 8$ 이 된다.

1절에서 기술된 국부 통계 특성에 의한 모드 및 검색 영역 설정에서 모드1에 해당되는 블록들은 검색 원점을 기준으로 움직임 벡터가 작은 영역에서 결정될 확률이 매우 높을 것으로 예측된 블록이다. 움직임이 작은 영상인 경우 많은 블록이 모드1에 해당되어 검색 영역 내에서 연산량 절감을 위해 모드1의 블록에 대해 8개의 정방향 검색이 아닌 그림 7에 도시된 십자형 검색 방식을 사용하였다. 그림 2의 예에서는 $Step_Size(i) = 2$ 인 경우의 정방향과 십자형을 나타내었다.

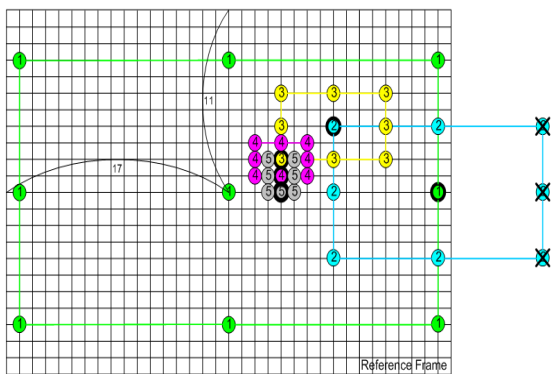


그림 6. 가변 단계별 검색 기법의 예
Fig. 6. Example of variable step search motion estimation

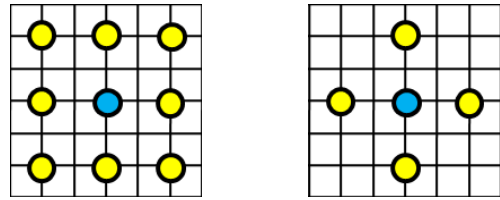


그림 7. Step size=2의 정방향과 십자형 검색 예
Fig. 7. Example of square and cross search with Step size=2

3.4. 조기 종료 조건

고속 움직임 벡터 예측을 위해 검색 영역내에서 조기 종료 조건을 이용한 방식에 대한 연구가 진행되어 왔다. 대표적으로 EPZS에서는 인접 블록의 SAD 값과 최소, 최대 경계 값을 이용하여 조기 종료 기준을 설정하고, 설정된 기준 값을 만족하면 움직임 예측을 종료하도록 되어있다[12]. EPZS 종료 조건은 다음과 같다.

$$SAD_{cur} \leq Stop\ Criterion \quad (12)$$

위 식의 SAD_{cur} 은 현재블록의 SAD값을 의미한다. 본 연구에서는 식 (9)에 의해 검색 영역을 결정한 후, 3절에서 기술된 단계별 검색 과정에서 식 (14)의 조기 종료 조건을 이용하였다.

본 논문에서 제안하는 고속 움직임 예측 기법에 대한 순서도를 그림 8에 나타내었다.

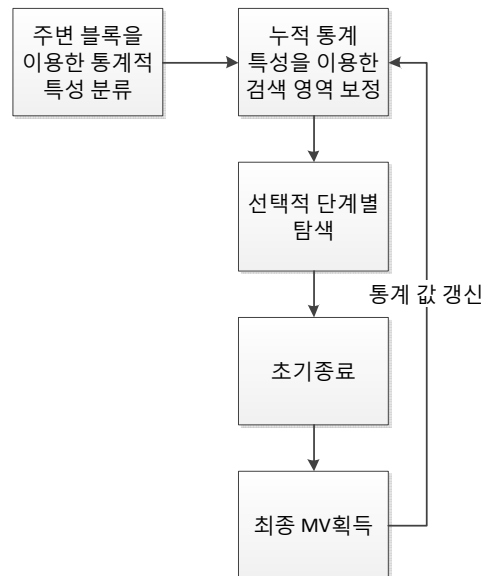


그림 8. 움직임 예측 순서도
Fig. 8. Flow of proposed motion estimation

IV. 실험 및 결과

본 논문에서 제안한 기법의 성능 비교를 위해 H.264/AVC JM 17.2 환경에서 고속 전역 검색(Full Search), 참고문헌 [12]의 EPZS와 비교하였다. 본 논문이 제안하는 기법은 고속 정수-화소 예측 기법 이므로서 EPZS와 비교시에 EPZS의 고속 부-화소 예측 기법을 off 시킨 상태에서 비교하였다. 더불어 움직임 예측 과정에서 단일 참조 프레임을 사용하여 수행하였으며, 양자화 파라미터 값(QP)을 각각 16, 24, 32, 40에 대하여 window 7, Quad Core CPU 2.67GHz, 4G RAM 환경에서 실험하였다. 실험 영상으로는 CIF 크기의 "Akiyo", "Container", "Foreman", "Mobile" 및 "Stefan"을 사용하였으며, JM 17.2 Baseline Profile Level 4.0에서 RD Optimization off 환경에서 실험하였다. 각 동영상에 대해 총 300 frame에 대해 30 frame/sec 환경에서 실험하였으며 초기 검색 영역은 32로 설정하였으며, 제안 방식의 식 (7)의 임계값은 각각 $T_1 = 90$ 및 $T_2 = 70$, $M = 100$ 으로 설정하여 모든 영상에 동일하게 적용하였다.

표 3에 FS 대비 EPZS 및 제안 방식의 BDPSNR(Bjonteggard Delta PSNR) 및 BDBR(Bjonteggard Delta Bit Rate)^[17] 측면에서 성능 비교를 나타내었다. 상기 결과에서 알 수 있듯이 움직임이 적은 영상인 경우 EPZS 및 제안 방식의 성능이 FS와 비교하여 성능 차이가 거의 없음을 알 수 있다. 반면에 "Foreman" 및 "Stefan"과 같이 객체 및 카메라의 움직임이 큰 영상은 0.1 dB 내외의 성능 차이가 있음을 확인할 수 있었다. 평균적으로 FS 방식과 비교하여 EPZS는 0.04 dB의 성능 향상이 있으며, 제안방식은 0.07 dB의 성능 저하가 있음을 확인할 수 있었다.

표 4 및 5에 각 방식에 대한 움직임 예측 시간 및 검색 지점 수의 비교 결과를 나타내었다. 연산량 비교는 총 움직임 예측 시간을 의미하는 Total MET (Motion Estimation Time) 및 블록당 평균 검색 지점 수로 구분하여 지표로 사용하였으며 MET 및 검색 지점 절감량은 다음과 같이 정의된다.

$$\text{절감량} = \frac{\text{제안방식} - \text{비교방식}}{\text{비교방식}} \times 100(\%) \quad (13)$$

표 4 및 5의 결과에 정리된 내용에서 알 수 있

듯이 MET 측면에서 제안 방식은 FS와 비교하여 평균 96(%) 이상의 절감됨을 확인할 수 있었으며, EPZS와 비교하여 11(%) 이상 절감됨을 알 수 있다. 더불어, 움직임 검색 지점 수 측면에서는 FS 및 EPZS와 비교하여 각기 98(%) 및 28(%) 이상의 절감 효과가 있음을 알 수 있다. 식 (9)와 같이 움직임 검색 영역 결정에 추가적인 연산량이 필요하여 MET의 절감율이 검색 지점 수 절감율에 비해 작은 결과로 나타났으며, 영상 간의 움직임이 클수록 연산량의 절감율은 크게 됨을 확인할 수 있었다. 더불어 본 논문에서는 자세히 기술되지 않았지만 제안 방식을 HD급 영상에 적용한 경우에도 유사한 결과를 얻을 수 있었다.

EPZS 방식은 이전 프레임의 블록 모드별 MVD 값을 저장해야하며, 제안 기법은 표 1에 기술된 블록의 구분에 따른 MVD 값을 저장해서 활용해야 한다. EPZS 및 제안 기법에서 필요한 MVD 수는 표 6과 같다.

표 6에서 알 수 있듯이 동영상의 공간 해상도가 커질수록 EPZS 방식은 저장해야 할 MVD 수가 선형적으로 증가하는 반면에 제안 방식은 활용하는 이전 부호화 블록 MVD 수가 고정되어 있는 관계로 동영상의 공간 해상도에 무관하게 일정함을 알 수 있다.

상기 결과로부터 제안된 고속 움직임 벡터 예측 방식은 FS 및 EPZS 방식과 비교하여 압축 효율 측면에서는 성능 저하가 다소 있었으나 검색지점 수 측면에서는 큰 절감 효과가 있음을 확인할 수 있었다. 제안 방식은 저연산을 필요로 하는 다양한 응용 서비스 분야로 활용 가능할 것으로 예측된다.

V. 결 론

본 논문에서는 H.264/AVC를 위한 통계 특성 기반 고속 정수-화소 움직임 예측 기법을 제안하였다. 특히 인접 블록들의 움직임 벡터 상관성을 표현하기 위한 SDMV를 정의하여 해당 부호화 블록의 움직임 검색 영역을 위한 모드 설정 및 움직임 벡터의 누적 통계를 기반으로 하는 움직임 검색 영역 보정 기법에 대해 제안하였다.

실험 결과를 통해 본 논문에서 제안한 방식이 FS 및 EPZS 기법과 비교하여 압축 효율 측면에서 0.07 (dB) 및 0.1 (dB) 정도의 성능 저하가 있었으나, 움직임 예측을 위한 검색 지점 수에서는 96(%) 및 28 (%) 이상의 절감 효과가 발생함을 확인할 수

표 3. CIF 영상별 BDPSNR 및 BDBR 비교
Table 3. BDPSNR and BDBR comparisons of CIF sequences

Sequence	QP	PSNR(dB)			Bitrate(kbit/sec)			BDPSNR(dB)		BDBR(%)	
		FS	EPZS	제안 방식	FS	EPZS	제안 방식	EPZS	제안 방식	EPZS	제안 방식
Akiyo	16	47.28	47.28	47.28	852.13	847.07	851.35	0.01	-0.03	-0.20	0.68
	24	42.60	42.58	42.59	189.10	187.17	189.50				
	32	37.44	37.38	37.43	50.40	49.92	50.80				
	40	32.42	32.38	32.41	17.66	17.45	17.75				
Container	16	46.10	46.09	46.08	2,578.71	2,576.29	2,580.23	0.00	-0.03	0.03	0.64
	24	39.11	39.11	39.09	566.29	565.68	566.96				
	32	33.54	33.50	33.50	116.33	115.62	116.34				
	40	28.72	28.64	28.67	30.75	29.80	30.31				
Foreman	16	46.40	46.40	46.40	4,437.69	4,399.62	4,434.83	0.03	-0.16	-0.64	3.39
	24	39.72	39.70	39.71	1,167.94	1,157.21	1,180.73				
	32	34.16	34.08	34.10	284.18	279.66	295.95				
	40	29.50	29.41	29.37	102.01	97.10	111.16				
Mobile	16	46.54	46.54	46.54	8,488.15	8,390.06	8,409.90	0.06	0.02	-0.97	-0.30
	24	38.75	38.76	38.75	3,792.29	3,747.19	3,765.65				
	32	31.23	31.22	31.23	1,166.32	1,155.51	1,165.06				
	40	24.99	24.95	24.95	263.23	259.67	263.73				
Stefan	16	46.78	46.78	46.79	7,178.17	7,069.61	7,147.07	0.09	-0.13	-1.46	2.12
	24	39.90	39.91	39.91	2,896.46	2,842.08	2,900.37				
	32	33.07	33.08	33.08	854.22	845.05	886.48				
	40	27.07	27.06	27.07	250.47	249.37	272.55				
평균								0.04	-0.07	-0.65	1.31

표 4. CIF 영상별 움직임 예측 시간 비교
Table 4. Motion estimation time comparisons of CIF sequences

Sequence	QP	Total MET(sec)			FS 대비 MET 절감량(%)	EPZS 대비 MET 절감량(%)
		FS	EPZS	제안방식		
Akiyo	16	5,339.45	224.15	219.96	-95.9	-1.9
	24	4,533.18	231.04	226.26	-95.0	-2.1
	32	3,202.68	218.77	212.72	-93.4	-2.8
	40	2,366.09	186.19	176.46	-92.5	-5.2
Container	16	7,728.51	216.96	210.61	-97.3	-2.9
	24	7,340.95	232.24	226.76	-96.9	-2.4
	32	5,985.96	252.49	241.10	-96.0	-4.5
	40	4,054.31	244.21	221.32	-94.5	-9.4
Foreman	16	10,870.29	290.19	259.36	-97.6	-10.6
	24	8,573.69	296.53	263.89	-96.9	-11.0
	32	6,316.93	319.12	271.93	-95.7	-14.8
	40	4,590.48	305.39	256.63	-94.4	-16.0
Mobile	16	12,781.44	286.86	247.71	-98.1	-13.6
	24	11,651.01	295.08	251.74	-97.8	-14.7
	32	9,585.08	339.36	265.07	-97.2	-21.9
	40	7,619.31	376.14	276.80	-96.4	-26.4
Stefan	16	12,363.95	319.32	266.47	-97.8	-16.6
	24	10,659.47	317.54	265.11	-97.5	-16.5
	32	8,152.62	325.79	262.25	-96.8	-19.5
	40	6,165.61	336.84	257.46	-95.8	-23.6
평균					-96.18	-11.82

표 5. CIF 영상 블록별 평균 검색 지점 수 비교
Table 5. Average search points comparisons per block of CIF sequences

Sequence	QP	블록당 평균 검색 지점 수			FS 대비 검색 지점 절감량(%)	EPZS 대비 검색 지점 절감량(%)
		FS	EPZS	제안방식		
Akiyo	16	421.20	2.29	1.93	-99.54	-15.69
	24	309.97	2.73	2.56	-99.17	-6.38
	32	157.75	3.60	3.43	-97.83	-4.51
	40	76.81	3.84	3.77	-95.09	-1.70
Container	16	840.18	3.30	3.04	-99.64	-7.89
	24	586.58	3.34	3.57	-99.39	7.05
	32	320.65	5.83	5.31	-98.34	-8.87
	40	165.05	6.57	5.19	-96.86	-21.09
Foreman	16	1011.13	9.47	6.91	-99.32	-27.03
	24	752.85	9.38	6.83	-99.09	-27.18
	32	446.06	11.78	7.63	-98.29	-35.21
	40	235.97	10.91	7.56	-96.80	-30.69
Mobile	16	1080.13	16.22	8.44	-99.22	-47.93
	24	912.36	16.14	8.31	-99.09	-48.51
	32	702.5	20.12	8.90	-98.73	-55.77
	40	431.39	21.24	8.96	-97.92	-57.80
Stefan	16	1089.91	16.09	9.03	-99.17	-43.90
	24	940.42	15.69	8.70	-99.07	-44.54
	32	685.30	17.01	8.51	-98.76	-49.96
	40	414.43	17.55	8.51	-97.95	-51.53
평균					-98.46	-28.96

표 6. 각 방식별 필요한 MVD 수 비교
Table 6. Required MVD of each method

Size	EPZS	제안방식
CIF(352x288)	16,236	2,800
HD(1280x720)	147,600	2,800

있었다. 더불어 영상의 해상도가 커질수록, 영상 내의 객체의 움직임 정도가 커질수록 연산량의 절감 효과는 커짐을 확인할 수 있었다. 제안된 방식은 저 연산량을 필요로 하는 고해상도 동영상 응용서비스에 활용 가능할 것으로 예측된다.

References

[1] ITU-T, Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264|ISO/IEC 14496-10 AVC) JVT-G050, May 2003.

[2] T. Wiegand, G. Sullivan, G. Bjontegaard, and A. Luthar, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuit and Systems for Video Tech., 13(7), pp.560-576, July 2003.

[3] I. Richardson, H.264 and MPEG-4 Video Compression, Wiley, 2003.

[4] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, "Video coding with H.264/AVC: tools, performance, and complexity," IEEE Circuits and Systems Magazine, 4(1), pp. 7-28, First Quarter 2004.

[5] T. Wiegand, X. Zhang, and B. Girod, "Motion-compensating long-term memory prediction," IEEE Int. Conf. Image Processing, pp.53-56, Oct. 1997.

[6] T. Wiegand, X. Zhang, and B. Girod, "Block-based hybrid video coding using motion-compensated long-term memory prediction," in Picture Coding Symp. Proceedings, Sep. 1997.

[7] T. Wiegand, X. Zhang, and B. Girod, "Long-term memory motion-compensated prediction," IEEE Trans. Circuit and Systems for Video Tech., 9(1), pp. 70-84, Feb. 1999.

[8] T. Wiegand and B. Girod, Multi-Frame Motion-Compensated Prediction for Video Transmission. Norwell, MA: Kluwer, 2001.

[9] K. B. Kim, Y. G. Jeon, and M.-C. Hong, "Variable step search fast motion estimation for H.264/AVC video coder," IEEE Trans. Consumer Electronics, 54(3), pp. 1281 - 1286, Aug. 2008.

[10] H. Gu, S. Chen, S. Sun, S. Chen, X. Chen, "Multiple search centers based fast motion estimation algorithm for H.264/AVC," Int. Conf. Parallel and Distributed Computing, Applications and Technologies, pp.346-350, Dec. 2010.

[11] J. W. Lee, M. S. Choi, J. S. Kim, W. K. Cho, "Fast H.264/AVC motion estimation algorithm using adaptive search range." Int. Symposium Integrated Circuits, pp.336-339, Dec. 2009.

[12] A. M. Tourapis, "Enhanced predictive zonal search for single and multiple frame motion estimation," Proc. Visual Communications and Image Processing, pp. 1069-1079. Jan. 2002.

[13] Z. Zhao, P. Liang, "A Statistical Analysis of H.264/AVC FME mode deduction," IEEE Trans. Circuits and Systems for Video Tech., 21(1), pp.53-61, Jan. 2011.

[14] M. Knesebeck, P. Nasiopoulos, "An efficient early-termination mode decision algorithm for H.264," IEEE Trans. Consumer Electronics, 55(3), pp.1501-1510, August 2009.

[15] H. Zeng, C. Cai, K. Ma, "Fast mode decision for H.264/AVC based on macroblock motion activity," IEEE Trans. Circuits and Systems for Video Tech, 19(4), pp.491-499, April 2009.

[16] W. Lin, D. Baylon, K. Panusopone, M. Sun, "Fast sub-pixel motion estimation and mode decision for H.264," IEEE Int. Symp. Circuits and Systems, pp.3482-3485, May 2008.

[17] G. Bjontegaarda, "Calculation of average psnr differences between rd-curves," ITU-T VCEG meeting, VCEG-m33, April. 2001

노진영 (Jin-young Noh)



2011년 2월 숭실대학교 정보통신전자공학부 학사
 2011년 3월~현재 숭실대학교 정보통신공학과 석사과정
 <관심분야> 움직임 추정, 객체 추적

홍민철 (Min-cheol Hong)



1988년 2월 연세대학교 전자공학과 학사
 1990년 8월 연세대학교 전자공학과 석사
 1997년 8월 Northwestern대학교 전기및컴퓨터공학과 박사
 1997년 9월~1998년 8월 Northwestern University 전기및컴퓨터공학과 박사후 과정
 1998년 9월~2000년 2월 LG전자 선임연구원
 2000년 3월~현재 숭실대학교 정보통신전자공학부 부교수
 <관심분야> 영상 복원 및 enhancement, Blind image deconvolution, 동영상 부호화, 3D 영상신호처리, 비선형 영상 신호처리, 움직임 추정 및 모델링