

# 시·공간적 상관도를 이용한 고속 H.264/AVC 전 영역 탐색 방법

문지희, 호요성

광주과학기술원 정보기전공학부

{jhmoon, hoyo}@gist.ac.kr

## Fast H.264/AVC Full Search Algorithm using Spatial and Temporal Correlation

Ji-Hee Moon and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)

### 요약

H.264/AVC 동영상 표준은 기존의 동영상 표준과 비교했을 때 뛰어난 압축률을 보인다. 특히 세밀한 움직임 예측을 통해 영상을 효율적으로 압축하지만 움직임 예측은 H.264/AVC 동영상 표준의 높은 복잡도의 원인 중 하나이다. 따라서 H.264/AVC의 부호화 시간을 단축하기 위해서는 고속 움직임 추정 기법이 필수적이다. 일반적으로 영상 신호는 인접한 화면과 매크로블록 사이에서 상관관계가 높고 부호화하고자 하는 매크로블록의 움직임벡터는 인접한 매크로블록에서 결정된 최적의 움직임 벡터와 유사한 방향성을 가진다. 그러므로 고정된 탐색 영역의 크기를 이용하면 불필요한 영역까지 움직임 예측 과정이 수행되어 계산량이 증가한다. 본 논문에서는 영상의 공간적, 시간적 상관도를 이용하여 탐색 영역의 크기를 결정하는 방법을 제안한다. 인접하는 블록들의 움직임 벡터의 표준편차를 이용하여 움직임이 작은 영역에서는 작은 탐색 영역을 이용하여 움직임 예측을 수행하고 반대로 움직임이 큰 영역에서는 큰 탐색 영역을 이용하여 움직임 예측을 수행한다. 또한 현재 화면과 참조 화면의 거리차가 클수록 참조 화면으로 선택되는 확률이 낮다는 사실을 이용하여 적응적으로 탐색 영역의 크기를 조절한다. 제안한 방법은 기존의 전 영역 탐색 방법과 유사한 부호화 성능을 보이면서 움직임 예측 시간이 평균 약 58.93% 감소하는 것을 확인할 수 있다.

### 1. 서론

H.264/AVC 표준은 현재 전 세계에서 차세대 동영상 압축 기술로 높이 평가되고 있다. 특히, 위성 DMB를 포함한 차세대 멀티미디어 서비스와 접목되면서 고품질의 영상 압축, 인터넷이나 케이블 모델에서의 영상 전달, 디지털 데이터 방송, 차세대 휴대전화를 통한 동영상 교환 등에 많은 응용이 기대되고 있다.

H.264/AVC 표준[1]은 기존의 표준에서 사용한 움직임 예측 및 보상과 DCT 변환 방법을 채택하였다. 하지만 H.264/AVC은 압축 효율을 높이기 위해 기존의 표준과 다른 새로운 기술들을 사용하였다. 블록의 크기를  $4 \times 4$  단위까지 줄여 움직임 보상을 수행하며  $1/4$  화소 단위의 움직임 예측을 사용하고, CAVLC와 CABAC과 같은 향상된 엔트로피 부호화, 그리고 새로운 비용함수를 이용한 R-D 최적화 방법을 등을 사용하였다. 또한 주관적 화질을 향상시키기 위해 디블록킹 필터 추가하고, 정수 기반의  $4 \times 4$  DCT를 사용하여 기존의 비디오 부호화 표준 방식의 부호기와 복호기 사이에 발생했던 DCT 계수 불일치 문제를 해결하였다. 이러한 기술들을 통해 H.264/AVC 표준은 기존의 비디오 부호화 표준에 비해 압축성능이 2배 정도 좋다[2].

H.264/AVC의 움직임 예측 및 보상은 부호화 효율을 증가시키지만, 동시에 높은 복잡도를 요구하고 전체 부호화에 필요한 계산량을 증가시키므로 실시간 구현에 어려움이 있다. 실제 H.264/AVC에서 부호화 시간의 50~60%는 움직임 예측 및 모드 결정에서 사용한다. 따라서 부호기의 복잡도를 줄이기 위한 여러 가지 고속 움직임 탐색 방법들이

제안되어 왔다. H.264/AVC에서 사용되는 대표적인 고속 움직임 탐색 방법으로는 정화소 예측을 위한 UMHexagonS(unsymmetrical multi-hexagon search)와 부화소 예측을 위하여 CBFPS(Center Biased Fractional Pel Search)가 있다[3]. UMHexagonS 고속 움직임 예측 방법은 계층적인 움직임 예측 방법의 한 가지이다. 이 방법은 초기 탐색점 예측(initial search point prediction), Unsymmetrical-cross search, Uneven Multi-Hexagon-grid search, Extended hexagon based search의 네 단계로 이루어진다. UMHexagons 방법은 지정된 탐색영역 안에서 탐색지점의 수를 효율적으로 줄일 수 있다. 정화소 움직임 예측 단계를 결정한 후, 부화소 움직임 예측 과정이 수행된다.  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$  블록에 대해서는 전역 부화소 움직임 예측이 수행되고 그 이하 모드에 대해서는 고속 부화소 움직임 예측 과정, 즉 CBFPS가 수행된다. CBFPS는 초기 탐색점 결정과 다이아몬드 탐색의 두 단계로 이루어져 있다. 이와 같은 고속 움직임 예측 방법을 비롯하여 매크로블록의 최적 모드 결정을 빠르게 하기 위한 방법들도 제안되어 왔다.

본 논문에서는 움직임 예측에서 소요되는 시간을 줄이기 위해 탐색 영역을 적응적으로 결정하여 불필요한 계산량을 줄임으로써 기존의 방법보다 향상된 전 영역 탐색 방법을 제안한다. 영상의 공간적 상관도를 이용하기 위하여 부호화하고자 하는 블록에 인접하는 블록들의 움직임 벡터의 통계적인 특성을 비선형 함수를 통해 탐색 영역을 결정한다. 영상의 공간적인 특성을 이용하여 결정된 탐색 영역은 시간적으로 연속된 참조 화면의 특성을 이용하여 다시 한 번 적응적으로 조절된다.

## 2. H.264/AVC 전 영역 탐색 방법

본 논문에서 제안하는 방법은 기본적으로 H.264/AVC의 전 영역 탐색 방법을 사용한다. H.264/AVC에서 제공하는 전 영역 움직임 탐색 방법은 그림 1과 같이 이루어진다.

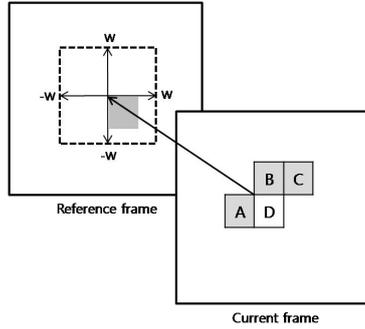


그림 1. 기존의 H.264/AVC 전 영역 움직임 탐색 방법

현재 부호화하고자 하는 매크로블록의 움직임 벡터를 예측하기 위해 움직임 벡터 간의 공간적 상관도를 이용한다. 인접하는 블록에서 정해진 움직임 벡터들의 중간 값을 수평 방향과 수직 방향으로 독립적으로 구한다. 예측된 움직임 벡터는 다음 식 (1)과 같이 표현할 수 있다.

$$\begin{aligned} MV_{Px} &= \text{median}\{MV_{Ax}, MV_{Bx}, MV_{Cx}\} \\ MV_{Py} &= \text{median}\{MV_{Ay}, MV_{By}, MV_{Cy}\} \end{aligned} \quad (1)$$

식 (1)을 이용하여 계산된 움직임 벡터는 현재 부호화하고자 하는 매크로블록의 움직임 벡터 예측이 되면서 참조 화면에서 움직임 예측 시 초기 시작점으로 사용된다. 탐색 영역을  $w$ 로 정했다면 예측된 초기 시작점을 기준으로 하여 상하좌우로  $w$ 길이만큼을 탐색 영역이 정해진다. 정화소 단위로 탐색 영역의 크기는  $(2w + 1)^2$ 이 된다. 정화소 단위에서 움직임 예측 과정이 진행되어 정해진 정화소 주위로 부화소 단위로 움직임 예측 과정이 수행된다. 이와 같은 방법으로 최적의 움직임 벡터가 결정된다.

## 2. 제안하는 방법

본 논문에서 제안하는 알고리즘은 크게 두 가지 방법으로 나눌 수 있다. 공간적 상관도를 이용하여 탐색 영역을 결정하는 방법과 시간적 상관도를 이용하여 참조 화면 사이의 탐색 영역을 적응적으로 결정하는 방법으로 나눌 수 있다. 이번 장에서는 본 논문에서 제안하는 알고리즘에 대해 설명한다.

### 가. 공간적 상관도를 이용하는 방법

한 화면에 위치하는 인접한 화소 사이에는 높은 상관도가 존재하는데, 이는 움직임 벡터에서도 마찬가지이다. 그러므로 인접하는 매크로블록의 움직임 크기에도 공간적 상관도가 높다. 기존의 움직임 예측 방식은 고정된 정사각형 모양의 탐색 영역 내에서 수행하므로 불필요한 위치를 탐색하여 계산량이 증가한다. 이러한 불필요한 연산을 줄이기 위해 인접하는 블록 사이의 움직임 벡터의 통계적 특성을 이용하여 수평, 수직 방향으로 각각의 움직임 예측 탐색 영역을 가변적으로 결정한다.

움직임 벡터의 통계적 특성을 이용하고자 인접하는 블록의 움직임 벡터들의 표준편차를 계산한다. 표준편차는 움직임 벡터의 수직방향과 수평방향의 움직임의 크기 정도를 의미한다. 예를 들어 수평방향의 표준편차가 수직방향의 표준편차보다 크면 현재 부호화하고자 하는 매크로블록 주변에 수평방향의 움직임이 크다는 것을 알 수 있다. 이를 이용하여 수평방향, 수직 방향에 대해 각각 서로 다른 크기의 탐색 영역을 결정하게 된다. 수직방향과 수평방향 움직임 벡터의 표준편차를 이용하여 적절한 영역 내에서 움직임 예측을 수행함으로써 계산량 절감이 가능하다. 인접하는 블록들의 움직임 벡터의 표준편차를 이용하여 탐색 영역이 기존의 탐색 영역보다 줄어들 것을 그림 2에서 확인할 수 있다.

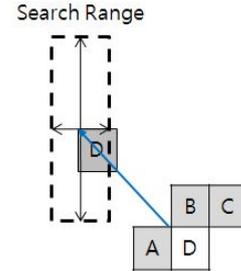


그림 2. 표준편차에 의해 감소된 탐색 영역

그러나 최적의 움직임 벡터가 나올 가능성이 있는 지점은 탐색 영역에 포함시켜 기존의 방법과 비교했을 때 성능 면에서 차이가 없어야 한다. 본 논문에서는 기존의 전 영역 탐색방법의 성능을 보장하기 위하여 비선형 함수를 이용하여 탐색 영역을 결정한다. 움직임 벡터의 표준편차 값이 지정된 탐색 영역의 크기를 넘어갔을 때 비선형 함수를 사용함으로써 최대 탐색 영역의 값을 가지도록 할 수 있다. 또한 단순히 표준편차 값을 탐색 영역의 크기로 결정하는 방법보다 비선형적인 값으로 탐색 영역의 크기를 조정함으로써 움직임 벡터의 표준편차 값보다 더 큰 값을 탐색 영역의 크기로 결정하여 성능 면에서 보장할 수 있다. 식 (2)는 표준편차를 이용하여 수평방향과 수직방향의 탐색 영역이 결정되는 식을 나타낸다.  $\alpha$ 는 표준편차를 이용하여 결정된 탐색 영역이 최대 탐색 영역을 넘어가는 것을 막기 위한 Scaling factor이다.

$$\begin{aligned} R_x &= \max\left\{1, \alpha \sqrt{\text{standard deviation}(MV_{s_x})}\right\} \\ R_y &= \max\left\{1, \alpha \sqrt{\text{standard deviation}(MV_{s_y})}\right\} \\ \alpha &= \text{search\_range} / 2 \sqrt{\text{search\_range}} \end{aligned} \quad (2)$$

### 나. 시간적 상관도를 이용하는 방법

공간적인 특성을 이용하여 결정된 탐색 영역은 모든 참조 화면에서 동일한 크기로 적용된다. 그러나 실제 최적의 참조 화면은 현재 부호화하고자 하는 매크로블록에서 가장 가까운 화면이 선택된다. 최적의 참조 화면이 선택되는 빈도를 그림 3에서 나타내고 있다. 그림 3을 보면, 현재 부호화하고자 하는 매크로블록에서 가장 가까운 참조 화면부터 세 번째 참조 화면이 최적의 참조 화면으로 선택될 확률이 95% 이상임을 확인할 수 있다. 부호화하고자 하는 매크로블록에서 가장 멀리 있는 참조 화면에서 움직임 예측 과정을 수행하기 위해 불필요한 계산량이 증가한다. 가장 멀리 있는 참조 화면에서 움직임 예측에 소모되는 시간을 효과적으로 줄이기 위해서 시간적 상관도를 이용한다. 부호화하고자 하는 화면의 시간적 상관도를 이용하여 결정된 탐색 영역을 참

조 화면에 따라 적응적으로 줄인다. 이 때, 탐색 영역은 실제 결정된 움직임벡터와 예측된 움직임 벡터의 차이값을 이용하여 결정된다.

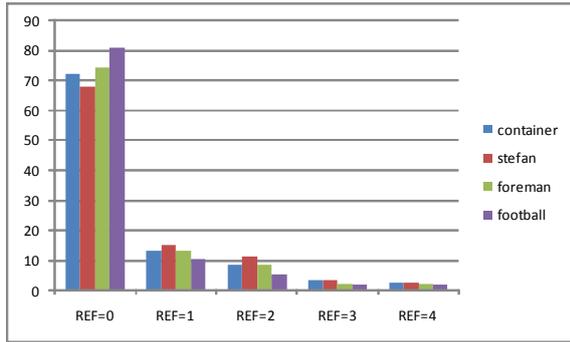


그림 3. 최적의 참조 화면이 선택되는 빈도

식 (3)은 실제 움직임 벡터와 예측된 움직임 벡터의 차이를 이용하여 네 번째 참조 화면과 마지막 참조 화면에서 업데이트된 탐색 영역을 나타낸다. 부호화하고자 하는 매크로블록에서 가장 가까운 첫 번째 참조 화면에서부터 세 번째 참조 화면까지 결정된 움직임 벡터와 예측된 벡터의 차이값들의 중간값을 이용하여 탐색영역을 업데이트 한다.

$$\begin{aligned}
 MVD_{x_{ref}} &= Predicted\_MV_{x_{curr}} - Best\_MV_{x_{ref}}, \quad 0 \leq ref \leq 2 \\
 MVD_{y_{ref}} &= Predicted\_MV_{y_{curr}} - Best\_MV_{y_{ref}}, \quad 0 \leq ref \leq 2 \\
 R'_x &= median\{MVDs_{x_{ref}}\}, \quad 0 \leq ref \leq 2 \\
 R'_y &= median\{MVDs_{y_{ref}}\}, \quad 0 \leq ref \leq 2
 \end{aligned} \quad (3)$$

작은 차이값은 현재 부호화하고자 하는 매크로블록의 움직임이 작다는 것과 초기 시작점으로부터 멀리 않은 곳에 최적의 움직임 벡터가 존재한다는 것을 의미한다. 반대로 큰 차이값은 현재 매크로블록의 움직임이 화면 사이에서 크다는 것과 초기 시작점으로부터 멀리 떨어진 곳에서 최적의 움직임 벡터가 존재하는 것을 나타낸다. 그러므로 작은 차이값을 보이는 영역, 즉 움직임 작은 영역에 대해서는 탐색 영역의 크기를 줄인다. 이 전의 참조 화면으로부터 결정된 움직임 벡터와 예측된 움직임 벡터의 차이값이 이 전에 공간적 상관도를 이용하여 결정된 탐색 영역보다 작으면 네 번째 참조 화면과 마지막 참조 화면에서는 탐색 영역의 크기를 줄인다. 반대로 차이값이 이 전에 결정된 탐색 영역보다 크면 탐색 영역을 바꾸지 않고 원래 방법대로 움직임 예측을 수행한다. 그림 4는 시간적 상관도를 이용하여 줄어진 탐색 영역을 보여준다. 네 번째 화면과 마지막 화면에서만 탐색 영역의 크기를 추가적으로 줄이는 이유는 참조 화면이 부호화할 매크로블록 멀어질 수록 오류가 누적될 수 있기 때문에 가까운 세 개의 참조 화면에 대해서는 성능을 보장하기 위해 위의 방법을 수행하지 않는다.

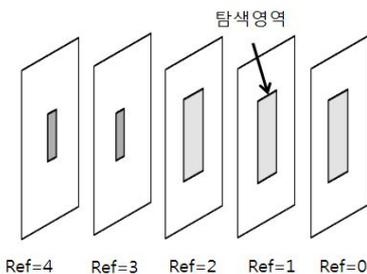


그림 4. 각 참조 화면에서 줄어진 탐색 영역

#### 4. 실험 결과 및 분석

본 논문에서 실험은 H.264/AVC 표준 소프트웨어 JM 12.4[4]를 이용하여 실험했다. 움직임이 다양한 영상을 네 개의 양자화계수(24, 28, 32, 36)와 두 개의 탐색 영역(16, 32)에 대해서 제안한 방법의 성능을 평가하였다. 실험에 쓰인 영상은 CIF(352×288) 해상도의 Foreman, Stefan, Football, Container이다. 표 1은 H.264/AVC를 이용한 시간적, 공간적 상관도를 이용한 적응적 탐색 영역 결정방법의 실험 환경을 나타낸다.

표 1. 실험환경

실험 변수	값
ProfileIDC	77
LevelIDC	44
SearchMode	-1
SymbolMode	1

표 2는 공간적 상관도만 고려한 방법과 공간적, 시간적 상관도 모두 고려한 방법의 비교를 나타낸다. 표 2를 통해서 공간적, 시간적 상관도를 이용하여 적응적으로 탐색 영역을 결정하는 방법이 공간적 상관도만 이용하여 탐색 영역을 결정하는 것보다 평균 8.12%의 속도 향상을 보임을 알 수 있다.

표 2. 제안한 알고리즘의 성능 비교

	QP	Spatial	Proposed	△ Time(%)
container	24	1454.748	1315.777	-9.55293
	28	1150.302	1047.906	-8.90166
	32	963.381	859.869	-10.7447
	36	815.748	736.365	-9.73131
stefan	24	2287.78	2074.764	-9.31104
	28	2091.106	1920.211	-8.17247
	32	1892.92	1739.744	-8.09205
	36	1653.128	1541.554	-6.74927
football	24	2790.75	2553.775	-8.49144
	28	2480.791	2298.531	-7.34685
	32	2160.455	1995.79	-7.62176
	36	1843.956	1697.332	-7.9516
foreman	24	1942.177	1813.646	-6.61788
	28	1592.923	1491.437	-6.37105
	32	1322.034	1229.042	-7.03401
	36	1098.911	1020.278	-7.15554
평균				-8.11535

H.264/AVC JM12.4에서 구현된 전 영역 탐색 방법과 비교 실험된 비트율-왜곡 곡선을 그림 5에 나타내었다. 가장 낮은 성능을 보인 container 영상과 높은 성능을 보인 football 영상을 비교하였다. 그림 5에서 알 수 있듯이 제안한 방법과 기존의 방법이 모든 비트율에서 PSNR이 동일함을 알 수 있다. 평균적으로 PSNR은 0.2dB 감소하였고 비트율은 1.10% 증가하였다.

표 3은 제안한 방법의 성능을 비교하기 위해 탐색 영역이 16인 경우에 대해서 움직임 예측 시간을 비교하였고 표 4는 탐색 영역이 32인 경우에 대해서 움직임 예측 시간을 비교하였다. 탐색 영역이 16인 경우의 움직임 예측 소요 시간은 약 56.12% 감소하였고 탐색 영역이 32인 경우의 움직임 예측 소요시간은 약 61.73%이 감소하였음을 확인할 수 있다.

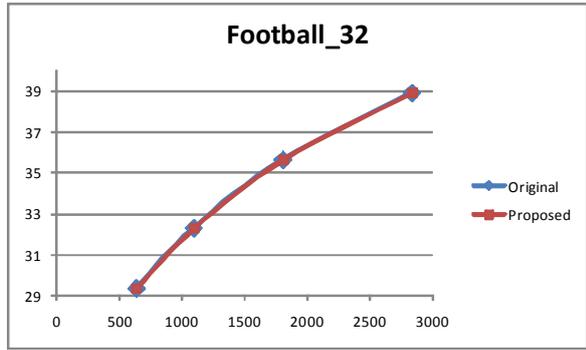
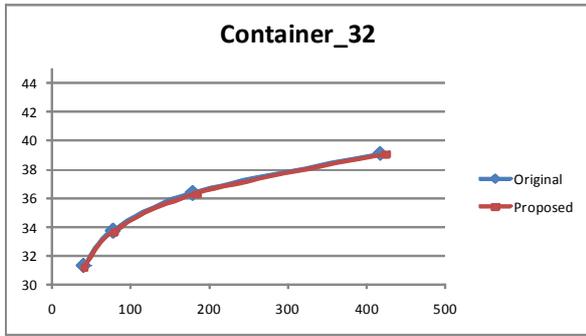


그림 5. 비트율-왜곡 곡선

탐색영역이 16인 경우와 탐색영역이 32인 경우를 비교했을 때, 탐색영역이 32인 경우의 움직임 예측에 소요되는 시간이 더 많이 감소했다는 것을 확인할 수 있다. 이는 탐색영역이 32인 경우가 탐색영역이 16인 경우보다 탐색영역의 크기가 감소되는 비율이 더 크기 때문이다. 예를 들어, 동일한 매크로블록의 표준편차가 12 일 때, 탐색영역이 16인 경우는 탐색영역이 14로 결정된다. 감소되는 비율이 20%이다. 탐색영역이 32인 경우에는 20으로 결정된다. 이 경우에는 감소되는 비율이 40%이다. 동일한 표준편차라도 탐색영역이 32인 경우가 감소되는 비율이 상대적으로 더 크기 때문에 전체 움직임 예측에 소요되는 시간도 감소하는 것이다.

표 3. 탐색 영역이 16인 경우의 움직임 예측 소요 시간 비교

	QP	Original	Proposed	$\Delta$ Time(%)
container	24	1138.858	461.409	-59.4849
	28	969.827	392.331	-59.5463
	32	819.944	336.94	-58.9069
	36	682.85	295.345	-56.7482
stefan	24	1464.027	640.032	-56.2828
	28	1363.314	603.125	-55.7604
	32	1260.631	565.363	-55.1524
	36	1150.004	524.07	-54.4289
football	24	1783.945	771.312	-56.7637
	28	1668.524	730.678	-56.2081
	32	1553.338	674.029	-56.6077
	36	1413.678	604.997	-57.204
foreman	24	1322.049	607.609	-54.0404
	28	1153.749	530.524	-54.0174
	32	999.98	462.196	-53.7795
	36	861.273	404.292	-53.0588
Average				-56.1244

표 4. 탐색 영역이 32인 경우의 움직임 예측 소요 시간 비교

	QP	Original	Proposed	$\Delta$ Time(%)
container	24	3765.369	1315.777	-65.0583
	28	3060.382	1047.906	-65.75898
	32	2527.185	859.869	-65.97523
	36	2052.031	736.365	-64.11531
stefan	24	5354.61	2074.764	-61.25275
	28	4825.752	1920.211	-60.20908
	32	4266.714	1739.744	-59.2252
	36	3746.529	1541.554	-58.85381
football	24	6641.756	2553.775	-61.5497
	28	5956.775	2298.531	-61.41316
	32	5304.639	1995.79	-62.37652
	36	4616.05	1697.332	-63.22977
foreman	24	4466.948	1813.646	-59.39854
	28	3696.351	1491.437	-59.6511
	32	3071.855	1229.042	-59.99023
	36	2529.531	1020.278	-59.66533
평균				-61.7325

### 5. 결론

본 논문에서는 영상의 시간적, 공간적 상관도를 이용하여 적응적으로 탐색 영역을 결정하는 방법을 제안하였다. 현재 부호화하고자 하는 블록의 인접하는 블록들 사이의 움직임 벡터의 표준편차 값을 이용하여 탐색 영역을 비선형적으로 줄였고, 가까운 참조 화면이 최적의 참조 화면으로 결정된다는 특성을 이용하여 이전 참조 화면에서 결정된 움직임 벡터와 현재 매크로블록의 예측된 움직임 벡터의 차를 이용하여 탐색 영역을 추가적으로 줄였다. 시간적 상관도를 추가적으로 사용함으로써 움직임 예측 시간이 평균 8.12% 감소함을 확인할 수 있었다. 제안한 방법은 PSNR과 비트율 측면에서 기존의 방법과 매우 유사한 성능을 유지하면서 움직임 예측 시간을 평균 약 58.93% 감소시켰음을 확인할 수 있다.

### 감사의 글

본 연구는 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업의 연구결과로 수행되었음. (NIPA-2009-(C1090-0902-0017))

### 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] Z. Chen, J. Xu, Y. He and J. Zheng "Fast integer-Pel and fractional-pel motion estimation for H.264/AVC," Journal of Visual Communication and Image Representation, vol. 17, no. 2, pp.264-290, 2006.
- [4] [http://iphome.hhi.de/suehring/tml/download/old\\_jm/jm12.4.zip](http://iphome.hhi.de/suehring/tml/download/old_jm/jm12.4.zip), Joint Video Team, Reference Software Version 12.4.