

동영상 압축 방식을 위한 최소 자승 기반 적응 움직임 벡터 예측 알고리즘

준회원 김 지 회*, 학생회원 정 종 우**, 정회원 홍 민 철**

Least Squares Based Adaptive Motion Vector Prediction Algorithm for Video Coding

Ji-hee Kim* Associate Member, Jong-woo Jeong** Student Member,
Min-Cheol Hong** Regular Member

요 약

본 논문에서는 동영상 부호화 기의 성능을 개선하기 위한 최소 자승 기반 적응 움직임 벡터 예측 알고리즘을 제안하고자 한다. 적응 움직임 벡터 예측 방식은 동영상 움직임 벡터의 국부 통계적인 특성의 돌연한 변화로 특징지어 진다는 것을 바탕으로 최소 자승(Least Squared) 기반의 선형 움직임의 계수들을 통계적 특성에 따라 최적화하는 방식이지만 상기 방식은 매우 높은 계산량을 요구하는 단점을 지니고 있다. 본 논문에서는 공간적인 움직임 변화 방향성을 가지는 최소 자승 최적화를 기반으로 움직임 예측기의 계수를 적응적으로 조절하여 움직임 예측 오류뿐만 아니라 계산량도 감소시키는 방식에 대해 기술한다. 실험을 통해 제안된 방식의 성능을 확인할 수 있었다.

Key Words : 최소 자승, 적응 움직임 벡터 예측, 국부 통계 특성, 공간적인 움직임 변화 방향성

ABSTRACT

This paper addresses an adaptive motion vector prediction algorithm to improve the performance of video encoder. The block-based motion vector is characterized by non-stationary local statistics so that the coefficients of LS (Least Squares) based linear motion can be optimized. However, it requires very expensive computational cost. The proposed algorithm using LS approach with spatially varying motion-directed property adaptively controls the coefficients of the motion predictor and reduces the computational cost as well as the motion prediction error. Experimental results show the capability of the proposed algorithm

I. 서 론

동영상 부호화기에서 사용되는 움직임 벡터는 인접된 블록 사이에 높은 상관관계를 갖고 있으며, 이와 같은 통계적 특성을 이용하여 움직임 벡터 예측기는 움직임 벡터와의 오차 값에 대한 통계적 특성을 변형시켜 동영상 부호화기의 압축 효율을 개선하는 목적으로 한다¹⁾.

현존하는 동영상 압축표준을 포함한 기존의 동영

상 표준 부호화 방식은 움직임 벡터를 부호화 하는 과정에서 움직임 벡터 예측 방식으로 미디언 필터를 사용하고 있다²⁾. 미디언 예측 기법은 인접한 블록의 움직임 벡터를 이용하는 문맥비나의 비선형 예측 방식으로서 움직임 에지를 보존하는 특성이 있으며, 부호기에서 전송된 예측 오류와 이미 처리된 인접 블록의 움직임 벡터만으로 움직임 벡터를 재구성하는 이점을 갖는다. 그러나 미디언 움직임 벡터 예측기는 제한된 지역적 정보를 이용하므로

* (주)엠크브 워크스 (jihee@mcubeworks.com), ** 숭실대학교 정보통신전자 공학부 (nobreaky@vip1.ssu.ac.kr, mhong@e.ssu.ac.kr)
논문번호 : #040016-0110, 접수일자 : 2004년 1월 10일

※본 연구는 한국 과학재단 목적 기초연구 (R01-2002-000-00073-0)지원으로 수행되었음

인접한 블록의 움직임 벡터의 지역적 통계적 특성이 비안정적인 (non-stationary) 경우, 큰 예측 오류와 분산값을 발생시킬 수 있다. 이와 같은 경우 미디언 움직임 벡터 예측기는 최적의 움직임 벡터 예측 오류를 구할 수 없고 결과적으로 압축 효율의 저하를 가져오게 된다^{3,4,5)}.

최소 자승 (Least Squares) 기반의 예측은 신호의 통계적 특성에 대응하는 지역적으로 최적의 계수를 얻을 수 있으나 많은 계산량의 연산을 요구하기 때문에 실제 응용에 쓰일 수 없는 단점을 가지고 있다. 그러므로 본 논문에서는 최소 자승 예측 기법을 움직임 벡터 예측에 적용하며, 움직임 방향성을 기반으로 재해석하여 예측 오류의 최소화는 물론 최소 자승 예측 방식이 갖는 계산량을 절감하는 적응 움직임 벡터 예측 방식을 제안한다. 본 논문에서 제안된 방식을 이용하여 움직임 예측기의 계수는 움직임 벡터의 공간적 통계특성의 변화의 결합에 의해 적응적으로 결정된다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 최소 자승기반의 움직임 예측기를 설명하고 3장에서는 최소 자승 기반 예측의 움직임 방향성과 새로운 적응 움직임 벡터 예측 방식을 소개하고 최종저공로 4장과 5장에서는 실험 결과와 결론에 대해 기술한다.

II. 최소 자승 기반 움직임 벡터 예측 알고리즘

동영상 부호화 방식에서 인접된 블록 사이의 움직임 벡터는 높은 상관관계를 갖고 있으며, 이와 같은 특성을 이용하여 압축 효율을 향상시키기 위해 실제 움직임 벡터와 예측 움직임 벡터 사이의 오차 값인 MVD(Motion Vector Difference)를 전송하게 된다. 2차원 영상의 (i, j) 번째 블록의 MVD는 다음과 같이 정의 된다.

$$MVD(i, j) = mv(i, j) - \widehat{mv}(i, j) \quad (1)$$

식 (1)에서 mv 및 \widehat{mv} 는 실제 움직임 벡터 및 예측 움직임 벡터를 각각 의미한다.

그림 1과 같이 영상 내의 움직임 벡터가 스택 순서로 저장되어 있다고 가정하여 동영상 내의 한 영상의 공간좌표를 하나의 인덱스인 n 으로 기술한다. 여기서 n 번째 움직임 벡터 $mv(n)$ 은 수평과 수직 성분으로 구성되어 있다. 제안하는 알고리즘은 수평과 수직의 움직임 벡터 요소들을 각각 독립적

으로 예측한다. 선형 예측을 기반으로 하는 최소 자승 움직임 벡터 예측 방식은 N 차 마코프 특성의 시계방향 순서에 의한 N 개의 가장 인접한 움직임 벡터만을 고려하여 다음과 같이 표현할 수 있다.

$$\widehat{mv}(n) = \sum_{k=1}^N a(k)mv(n-k) \quad (2)$$

식 (2)에서 $\widehat{mv}(n)$ 은 예측된 움직임 벡터를, $mv(n-k)$ 는 예측에 사용되는 k 번째 인접한 움직임 벡터를 나타내며 가중치 $a(k)$ 는 선형 예측기의 k 번째 계수를 나타낸다.

mv(n-11)	mv(n-8)	mv(n-6)	mv(n-9)	mv(n-12)
mv(n-7)	mv(n-3)	mv(n-2)	mv(n-4)	mv(n-10)
mv(n-5)	mv(n-1)			

그림 1. 학습창 내의 움직임 벡터 순서

일반적으로 대부분의 움직임 벡터는 정적인 가우시안 분포를 갖는다는 가정에 위반되며 이와 같은 문제를 해결하기 위해 움직임 벡터를 예측하는 과정에서 국부 통계적 특성을 이용하여 예측기를 설계한다. 그러므로 그림 2에서와 같이 움직임 벡터 예측 계수를 최적화하기 위해 설정된 학습창 (training window)은 두 개의 직사각형 안에 $M = 2T(T + 1)$ 개의 인접한 처리된 움직임 벡터들로 구성된다.

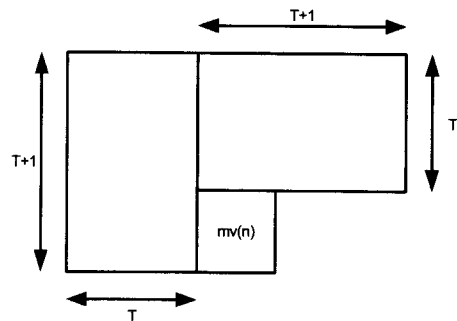


그림 2. 학습창

학습창을 $M \times 1$ 열 벡터 $\overrightarrow{mv} = [mv(n-1), mv(n-2), \dots, mv(n-M)]^T$ 로 명시하면 \overrightarrow{mv} 를 예측하기 위한 인접한 움직임 벡터들은 $M \times N$ 형태인 다음과 같은 식 (3)의 행렬로 주어진다. 여기서 T 는 전치를 의미한다.

$$C = \begin{bmatrix} mv(n-1-1) & mv(n-1-2) & \dots & mv(n-1-N) \\ mv(n-2-1) & mv(n-2-2) & \dots & mv(n-2-N) \\ \vdots & \vdots & \ddots & \vdots \\ mv(n-M-1) & mv(n-M-2) & \dots & mv(n-M-N) \end{bmatrix} \quad (3)$$

식 (3)에서 $mv(n-j-k)$ 는 $mv(n-j)$ 를 예측하기 위한 k 번째 이웃하는 움직임 벡터이다. 학습창 내의 움직임 벡터 예측 계수는 다음과 같은 부가 함수를 최소화시켜 얻을 수 있다.

$$M(\vec{a}) = \| \overrightarrow{mv} - C \vec{a} \|^2 \quad (4)$$

식 (4)의 $\| \cdot \|$ 는 유클리디안 norm을 나타내며 최적 필터 계수는 식 (4)에 gradient를 취하여 얻게 된다.

$$\nabla_{\vec{a}} M(\vec{a}) = -C^T(\overrightarrow{mv} - C \vec{a}) = 0 \quad (5)$$

식 (5)에서 0는 $N \times 1$ 크기의 벡터를 의미하며, $\vec{a} = [a(1), a(2), \dots, a(N)]^T$ 는 최적화된 움직임 예측 계수를 나타낸다^{6,7,8,9,10}. 식 (5)를 \vec{a} 에 대해 다음과 같이 표현할 수 있다.

$$\vec{a} = (C^T C)^{-1} (C^T \overrightarrow{mv}) \quad (6)$$

III. 움직임 방향성을 이용한 계산량 절감 방식

적응 움직임 벡터 예측 방식의 유용성은 인접 벡터들 사이의 상관관계가 매우 높은 움직임 배경영역과 상관관계가 상대적으로 낮은 움직임 에지 영역으로의 적응 능력에 의해 결정된다. 움직임 에지는 불규칙한 특성으로 인해 정확한 움직임 벡터를 예측하기는 용이하지 않다. 최소 자승 기반 움직임 예측 방식은 적응적으로 최적의 움직임 벡터 방향성에 대한 예측을 근사화 함에 따라 최적화에 근접된 움직임 벡터 예측 방식을 제공하게 된다.

최소 자승 최적화 과정에서 설정된 학습창 내의

인접한 움직임 벡터들은 움직임 영역 주변의 에지 움직임 벡터들과 움직임 영역에서 떨어져 있는 비 에지 움직임 벡터들로 분류된다. 비 에지 움직임 벡터들에 대해서는 C 는 정칙 (full-rank) 행렬이 되지 못해 최소 자승 최적화 과정은 유일한 해를 작지 못할 수 있다. 실제로 움직임 에지에서 떨어져 있는 움직임 벡터들에서의 최적 예측 집합, $S_{non-edge}$

은 N 차 다중 평면 $S^N = \sum_{k=1}^N a(k)$ 안에 존재한다. 반면에 움직임 에지 주변의 움직임 벡터들에서는 C 가 정칙 행렬이 되어 해가 유일하게 존재하므로 움직임 에지 주변의 움직임 벡터들의 최적 예측 집합 S_{edge} 은 다음과 같이 기술될 수 있다.

$$S_{edge} \subset S^N \quad (7)$$

그러므로 움직임 벡터 예측 과정에서 움직임 에지 주변의 움직임 벡터들이 주된 역할을 담당하게 되어 위에서 설명된 움직임 에지 방향성으로 인해 학습창 내의 최소 자승 최적화 과정은 움직임 에지의 검출 없이도 움직임 에지 주변의 최적 예측 계수를 얻을 수 있는 유용한 방법을 제공한다.

최소 자승 최적화 방식의 가장 큰 문제점은 식 (6)의 공분산 행렬 $C^T C$ 의 연산에 필요한 계산량이다. 본 논문에서는 이러한 문제를 해결하기 위해 일정한 성능을 유지 하면서 움직임 벡터들중 일부만을 최소 자승 최적화 과정에서 사용함으로써 계산량을 줄이는 방식에 대해 제안한다. 움직임 벡터 예측은 다음 두 가지 면을 바탕으로 하는 움직임 에지 방향성을 사용한다. 첫째, 움직임 에지 주변의 움직임 벡터들을 최적화하는 예측 계수는 같은 움직임 에지를 따르는 인접한 움직임 벡터에 대해서도 적용가능하다. 둘째, 임의의 움직임 에지에서 최적화된 예측 집합은 배경영역의 최적화된 예측 집합의 부분 집합이다. 이와 같은 특성은 움직임 에지 주변의 움직임 벡터에서 최적화된 예측 계수를 저장하여 다음 움직임 에지가 존재할 때 까지 반복적으로 사용하는 것을 가능하게 한다. 다시 말하면 최소 자승 최적화 과정을 움직임 벡터단위가 아닌 움직임 에지 단위로 수행함으로써 계산량을 줄일 수 있다. 본 논문에서는 움직임 에지 기반의 최소 자승 최적화를 수행하기 위해 다음과 같은 예측 계수 전환 방법을 정의 하였으며 예측 계수 전환의 기준의

되는 예측 오류는 다음과 같다.

$$e(n) = mv(n) - \widehat{mv}(n-1) \quad (8)$$

식 (8)의 움직임 예측 오류가 큰 경우는 해당 블록이 움직임 에지 영역을 의미하며 예측 오류가 작은 블록은 움직임 비에지 영역을 의미한다. 이미 기술한 바와 같이 움직임 비 에지 영역에서는 필터 계수에 의해 영향을 크게 받지 않으므로 움직임 예측 오류에 따라 필터 계수를 다음과 같이 정의한다.

$$a(n) = \begin{cases} (C^T C)^{-1} C^T \widehat{mv} & \text{if } |e(n)| > TH \\ a(n-1) & \text{otherwise} \end{cases} \quad (9)$$

또한 식 (2)에서의 예측된 움직임 벡터 $\widehat{mv}(n)$ 는 아래 식 (10)과 같이 예측 오류 값에 따라 구하게 된다.

$$\widehat{mv}(n) = \begin{cases} \sum_{k=1}^N a(k)mv(n-k) & \text{if } |e(n)| > TH \\ \widehat{mv}(n-1) & \text{otherwise} \end{cases} \quad (10)$$

IV. 실험 결과

제안된 알고리즘을 다양한 영상, 해상도, 비트율로 실험하여 ITU-T 및 ISO/IEC에서 공동으로 개발하고 있는 차세대 동영상 부호화 방식인 H.264/AVC 방식의 JM2 (joint model Number2) 미디언 예측과 비교하였다. 본 실험에서는 4개의 인접한 움직임 벡터를 가지는 학습창과 다양한 임계치에 적용하였다. 알고리즘의 객관적인 성능을 측정하기 위해 PSNR(Peak Signal to Noise Ratio)를 사용하였으며, $M \times N$ 의 크기의 8비트 이미지에 대해서 PSNR은 다음과 같이 정의된다.

$$PSNR = 10 \log_{10} \frac{MN \times 255^2}{\|f - \tilde{f}\|^2} \quad (11)$$

식 (11)에서 f 는 원영상을, \tilde{f} 는 재구성된 영상을 각각 나타낸다. 더불어 움직임 벡터 예측기의 성능을 비교하기 위해 MPEPB (Motion Prediction Error Per Block)를 식 (12)와 같이 정의 하였다.

$$MPEPB = \frac{1}{K} \sum_{i=1}^K (|mv_{i,j}(t) - \widehat{mv}_{i,j}(t)| + |mv_{i,j}(t) - \widehat{mv}_{i,j}(t)|) \quad (12)$$

식 (12)에서 K 는 영상 내의 총 블록수이며, $mv_{i,j}(t)$ 와 $\widehat{mv}_{i,j}(t)$ 는 i 번째 블록의 j 방향 움직임 벡터 값과 움직임 벡터 예측값을 나타낸다. 또한 미디언 방식과 제안된 알고리즘의 계산량을 비교하기 위해 ETSPF(Encoding Time Saving per Frame)을 아래 식 (13)과 같이 정의 하였다.

$$ETSPF = \frac{ET(JM2) - ET(JM2 + LS)}{ET(JM2)} \times 100(\%) \quad (13)$$

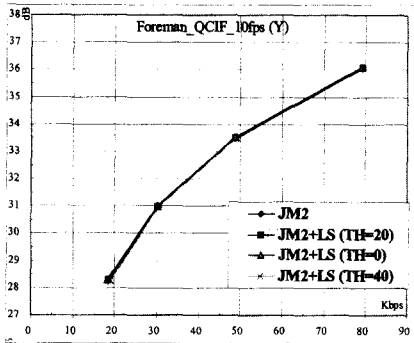
본 논문의 실험을 위해 256MB 램의 펜티엄 III 700MHz PC를 사용하였으며, 움직임 탐색 영역은 수평 수직 방향 32를 사용하였다. 실험 결과는 10 frame/s 로 부호화된 QCIF Foreman, 10 frame/s로 부호화된 QCIF Container, 10 frame/s로 부호화된 QCIF News, 그리고 30 frame/s로 부호화된 CIF Mobile 영상에 대해서 기술하였다.

그림 3은 식 (8) 및 식 (9)의 임계치 TH 가 '0', '20', 및 '40'에 대해 비트율에 따른 PSNR 비교를 보여준다. 실험 결과를 통해 움직임 벡터 예측 개선에 의해 감소된 정보량 보다 양자화된 계수의 비중이 큰 이유로 제안된 방식에 의한 PSNR 개선의 미지함을 확인할 수 있었다. 더불어 영상에 따라 차이는 있으나 임계치가 작을 수록 PSNR의 성능이 있음을 확인할 수 있었다. 이와 같은 결과는 최소 자승 기법을 움직임 벡터 예측기에 적용함이 상당

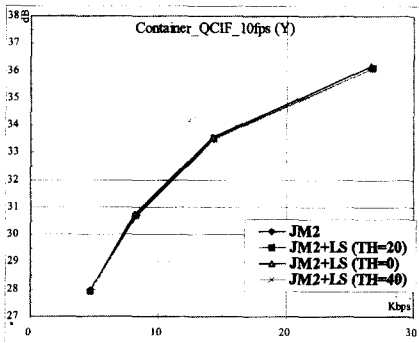
효과적임을 나타내고 있다. 그림 4는 비트율에 따른 움직임 벡터 예측 오류를 나타내고 있다. 위의 결과는 최소 자승 기법이 미디언 필터 기법 보다 성능면에서 우월하며 예측 오류는 임계치 값이 클 수록 크며 이와 같은 결과는 식 (8)에서 정의한 임계치가 큰 경우 이전 블록의 움직임 벡터 예측시에 사용되었던 필터 계수가 움직임 에지 영역에서도 동일하게 사용된 경우에 기인한 것임을 확인할 수 있었다. 흥미로운 사항은 H.264/AVC 부호화 방식이 비트율 제어를 사용하기 때문에 움직임 벡터는 비트율에 따라 제한되어 비트율이 증가함에 따라 움직임 벡터 예측 오류가 증가 했다는 점이다. 그림 5에 다양한 임계치에 대해 비트율에 따른 ETSPF 비교를 나타내었다. 그림 4와 상반되게 임계치 값이 클수록 계산량의 이득이 커짐을 알 수 있었으며 더불어 이미 기술한 바와 같이 예측 오류에 따라 예측 계수 전환 방식을 사용하지 않은 경우 ($TH=0$) 과다한 계산량이 필요로 됨을 확인할 수 있었다. 위

의 결과로부터 임계치 값이 '20' 부근에서 움직임 벡터 오류 감소 및 계산량의 절감을 동시에 얻을 수 있었다.

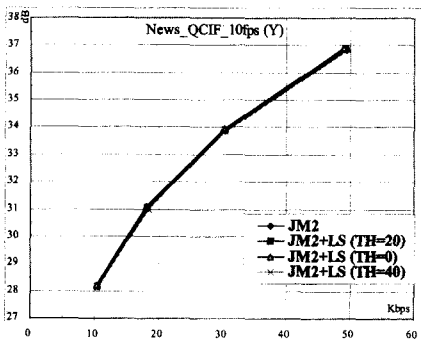
제안된 알고리즘의 독창성은 움직임 에지 영역에서의 예측 계수의 집합이 움직임 배경영역에 대한 최적 움직임 예측 집합의 부분 집합이라는 것으로 주변 움직임 벡터의 움직임 에지 방향성에 의해 움직임 벡터의 예측 계수가 적응적으로 조절된다는 것이다. 이를 통해 움직임 벡터 예측 오류의 감소 및 계산량의 이득이 있음을 확인할 수 있었다.



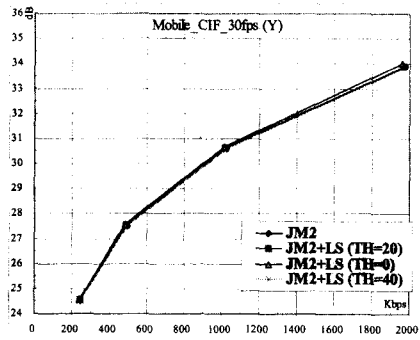
(a)



(b)

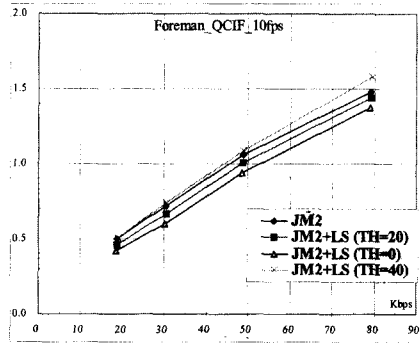


(c)

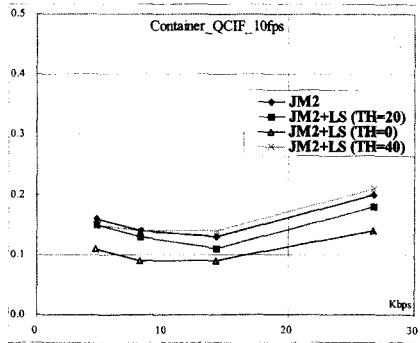


(d)

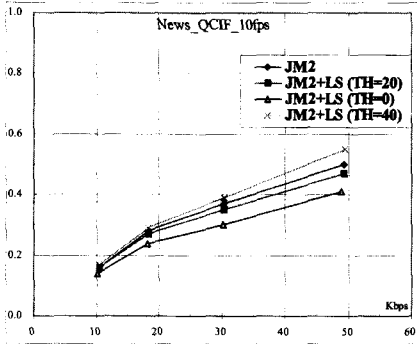
그림 3. 비트율에 따른 PSNR 비교: (a) QCIF Foreman 동영상 (10 frames/sec), (b) QCIF Container 동영상 (10 frames/sec), (c) QCIF News 동영상 (10 frames/sec), (d) CIF Mobile 동영상 (30 frames/sec)



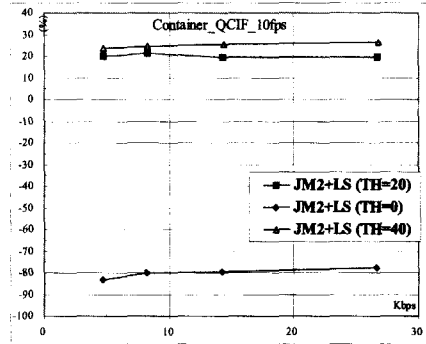
(a)



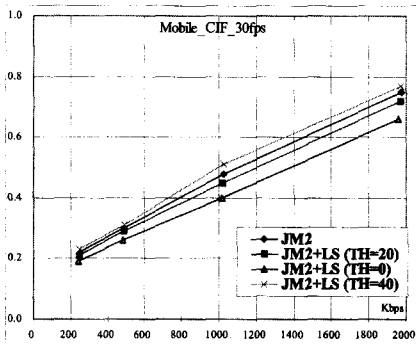
(b)



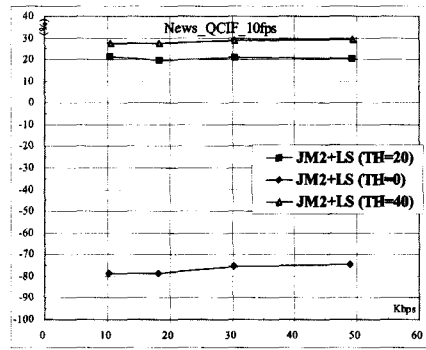
(c)



(b)

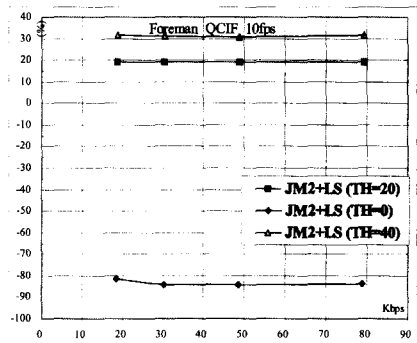


(d)

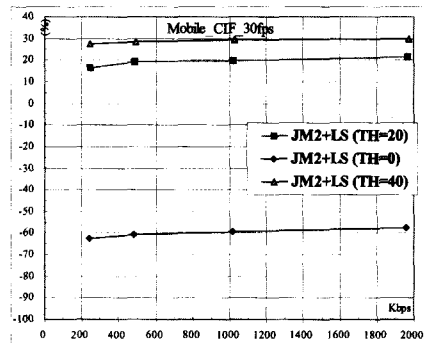


(c)

그림 4. 비트율에 따른 MPEPB 비교: (a) QCIF Foreman 동영상 (10 frames/sec), (b) QCIF Container 동영상 (10 frames/sec), (c) QCIF News 동영상 (10 frames/sec), (d) CIF Mobile 동영상 (30 frames/sec)



(a)



(d)

그림 5. 이비트율에 따른 MTSFP 비교: (a) QCIF Foreman 동영상 (10 frames/sec), (b) QCIF Container 동영상 (10 frames/sec), (c) QCIF News 동영상 (10 frames/sec), (d) CIF Mobile 동영상 (30 frames/sec)

