Journal of Korea Multimedia Society Vol. 7, No. 8, August 2004(pp. 1025-1034)

움직임의 시간적 연속성을 이용한 적응적 움직임 추정 알고리즘

최정현[†], 이경환[††]

요 약

본 논문에서는 움직임의 시간적 연속성을 이용한 적응적 움직임 추정 알고리즘을 제안한다. 기존의 전역 탐색 방법 (FSA)에서와 같이 정방형의 전역 탐색 영역 (GSR)을 정하고, GSR 안에 시간적으로 인접한 블록들의 움직임 벡터들로 예측한 비정방형의 적응적 국부 탐색 영역 (LSR)들을 정한다. 여기서 시간적으로 인접한 블록은 이전 프레임의 블록 중 움직임 벡터로 볼 때 현재 프레임에 시간적으로 움직임이 연속될 가능성이 있는 블록이다. 그리고 이 LSR들로 만든 영역에 대해서만 움직임 탐색을 행하여 탐색의 정확성을 높이고 탐색 시간을 줄일 수 있다. 모의 실험 결과, 제안한 방법은 기존의 탐색 방법들에 비해 탐색 영역을 줄여 계산량을 현저히 낮추면서도 우수한 화질을 보였다.

Adaptive Motion Estimation Algorithm Using Temporal Continuity of Motion

Jung Hyun Choi[†], Kyeong Hwan Lee[††]

ABSTRACT

This paper proposes an adaptive motion estimation algorithm using the temporal continuity of motion. We set up a squared global search region (GSR), which basically corresponds to the search region of FSA, and non-squared adaptive local search regions (LSRs), the positions for which are predicted by the motion vectors of the temporal neighbor blocks, are constructed in the GSR. The previous frame blocks that possibly have effects on the current block are to be the temporal neighbor blocks. Because motion estimation is only performed in the areas made by LSRs, we can estimate motion more correctly and reduce processing time. Experimental results show that the proposed method can enhance visual qualities with significant reductions of complexity by reducing search regions, when compared to the conventional methods.

Key words: Motion Estimation(움직임 예측), Motion Vector(움직임 벡터), BMA(블록 정합 방법), FSA (전역 탐색 방법), MAD(평균 절대치 오차)

1. 서 론

동영상 신호에는 높은 시간적 및 공간적 중복성이 존재하므로 이를 제거함으로써 높은 부호화 효율을 가질 수 있다. 공간적인 중복성을 제거하기 위해서는 기존의 정지영상 부호화에서 적용된 방법들이 주로 사용되는 반면, 동영상에서는 시간적인 중복성을 효과적으로 제거함으로써 큰 압축효과를 얻는다[1,2]. 이러한 시간적인 중복성을 제거하는 방법으로는 주로 블록 정합 방법인 BMA (block matching algorithm)을 쓰는데, 이는 이전 프레임의 현재 블록의

※ 교신저자(Corresponding Author) : 이경환, 주소 : 경북 경주시 강동면 525번지(780-713), 전화 : 054)760-1712, FAX : 054)760-1719, E-mail : khlee@mail.uiduk.ac.kr
접수일 : 2003년 9월 15일, 완료일 : 2004년 1월 19일
[†] 정회원, 기술신용보증기금
(E-mail : jhchoi@kibo.or.kr)
[††] 정회원, 위덕대학교 컴퓨터멀티미디어공학부

위치를 중심으로 일정한 탐색 영역을 정하여 왜곡을 계산하여 가장 왜곡이 작은 블록과 현재 블록의 위치 차이를 움직임 벡터로 정하여 움직임 보상을 하는 방법으로, 일반적으로 탐색 영역의 전체를 화소단위로 천이시켜 탐색하는 전역 탐색 방법, 즉 FSA (full search algorithm)을 사용한다. 그러나 FSA를 이용하여 정확한 움직임 탐색을 하려면 탐색 영역이 넓어야 하므로 계산량이 증가하는 단점이 있다[3].

그러나 움직임이 보상된 후에도 여전히 시간적 및 공간적 상관성이 잔재하여, 탐색을 통해 구해진 움직임 벡터들 또한 시간적 및 공간적 인접 블록들 간에 비슷하게 나타난다. 이 특성을 이용하여, 인접 블록들의 움직임 벡터들로써 현재 블록의 움직임을 예측하고, 이를 기준으로 움직임을 추정하여 성능을 향상시킨 방법들이 제안되었다[4,5]. 이들 예측 움직임 추정 방법들에서는, 시간적 및 공간적 인접 블록들의 움직임 벡터들에 가중치를 곱한 합 (weighted sum)으로써 현재 블록의 움직임을 예측한다. 또한 현재 프레임에서 미리 움직임을 탐색한 공간적 인접 블록들과 이전 프레임의 현재 블록, 즉 시간적 인접 블록의 움직임 벡터를 모두 이용하고 가중치를 부여하여 문턱값 (threshold)에 의해 움직임을 예측하는 방법이 제안되어 더욱 정확한 움직임을 추정하였다[6]. 그러나 이들 방법은 결국 하나의 움직임만을 예측하여 이를 중심으로 움직임 탐색을 행하므로, 움직임이 복잡하거나 빠른 부분의 경우 인접 블록들의 움직임 벡터들 간에 상관성이 떨어지는 단점이 있다.

움직임 추정 오차는 최소 오차 탐색점으로부터 멀어질수록 단조 증가하는 특성이 있다. 이를 이용하여 움직임 추정 계산량을 감소시키는 단계적 움직임 추정 방법들이 제안되었다[7,8]. 그러나 단계적 탐색으로 인해 국부 최소 (local minimum)에 빠지는 경우가 많이 발생하여, 움직임 추정 오차가 증가되는 단점이 있다. 또한 현재블록의 화소값으로 비트평면 (bit-plane)을 만들어 이를 먼저 비교하여 탐색 범위를 정하고, 탐색범위의 크기에 따라서 움직임의 크기를 세 단계로 나누어 각각에 맞게 적응적으로 움직임 탐색을 행하는 방법이 제안되어 전역탐색에 비해 계산량을 92~95%까지 줄일 수 있었다[9]. 그러나 먼저 화소값 정보를 이용하여 비트평면을 만들어야 하고, 각 과정에 문턱값을 사용하여 비교하여야 하므로 비효율적인 단점이 있다.

BMA 방법으로 구한 블록의 움직임은 프레임의 진행에 따라 시간적으로 계속되는 특성이 있다[10-12]. 이를 이용하면 현재 블록의 움직임을 효과적으로 예측할 수 있는데, 현재 블록에 대한 탐색 영역은 이전 프레임에서 구성되므로 이 탐색 영역에 포함되는 시간적 인접 블록들의 움직임 벡터는 현재 블록의 움직임을 구하기 위한 좋은 근거가 된다. 그러므로 이전 프레임의 시간적 인접 블록들의 움직임을 이용하여 현재 블록의 움직임 특성을 예측할 수 있으며, 이에 따라 탐색 영역을 효율적으로 정할 수 있다. 이를 이용하여 기본적으로 넓은 전역 탐색 영역 (global search region, GSR)을 설정하고, 이 중 시간적 인접 블록들에 의해서 예측된 탐색 영역에 대해서만 탐색을 행한다. 시간적 인접 블록들의 움직임을 이용하여 정방형의 국부 탐색 영역 (local search region, LSR)의 위치를 GSR 범위 안에서 예측하는 움직임의 시간적 연속성을 이용한 움직임 추정 (temporal predictive motion estimation, TPME) 방법을 제안하였다[13].

본 논문에서는 기존에 제안한 TPME 방법을 한 단계 발전시킨 적응적 움직임 추정 방법 (adaptive temporal predictive motion estimation, ATPME) 방법을 제안하였다. TPME 방법에서는 LSR의 크기를 고정하고서 탐색 영역을 예측한다. 그러나 제안한 ATPME 방법에서는 시간적 인접 블록의 움직임 벡터에 따라 탐색 영역의 형태를 적응적으로 변화시켜서 움직임 추정 효율을 더욱 향상시킨다. 시간적 인접 블록의 움직임 벡터의 수직 및 수평 성분에 맞게 예측 탐색 영역 LSR의 세로 및 가로의 크기를 적응적으로 변화시킨다. 따라서, 움직임 벡터의 크기 및 방향이 모두 고려된 적절한 탐색 영역이 구성된다. 제안한 ATPME 방법에서는 움직임 벡터의 특성에 적응적으로 예측하여 탐색 영역을 구성하므로, 화질의 향상 및 계산량의 급격한 감소를 얻을 수 있다.

다양한 움직임 특성을 가지는 실험 영상들에 대한 모의 실험 결과, 제안한 ATPME 방법은 기존의 FSA 및 움직임 벡터를 예측하는 방법들, 그리고 기존에 제안한 TPME 방법들에 비해 우수한 PSNR과 급격한 계산량 감소를 나타냄을 확인할 수 있었다.

2. 기존의 움직임 탐색 방법

2.1 FSA 움직임 탐색방법

블록 단위의 움직임 추정 방법인 블록 정합 방법

(BMA)은, 블록 기반의 변환 부호화와 호환성이 좋고 알고리즘이 간단하므로, 동영상 부호화에서 널리 사용되고 있다. 또한 한 개의 움직임 벡터가 블록의 움직임을 대표하여 구해지기 때문에, 높은 데이터 압축률이 요구되는 동영상 부호화에 적합하다.

BMA로서 가장 기본적인 움직임 추정 방법인 전역 탐색 방법 (FSA)에서는, 이전 프레임에서 현재 블록과 같은 위치를 중심으로 탐색 영역을 정하고, 이 영역 내의 모든 탐색점들에 대한 블록 정합을 행하여, 그 중 최소 오차를 가지는 탐색점의 좌표를 움직임 벡터로 구한다. 블록 정합 방법에서의 움직임 벡터를 그림 1에 나타내었다. 이전 프레임 전체를 탐색 영역으로 둘 수도 있으나 계산량이 방대하므로, 일반적으로 그림 1에서와 같이 현재 블록과 동일한 위치를 중심으로 한 제한된 탐색 영역을 이전 프레임에서 구성하며, 움직임 벡터는 탐색 영역 내에서 현재 블록과 가장 유사한 영역에 대한 탐색점 좌표가 된다.

즉 전역 탐색 방법 (FSA)에서는, 이전 프레임의 정해진 탐색 영역 내의 모든 탐색점들에 대해서 탐색을 행하여, 그 중 현재 블록과 가장 유사한 영역을 찾아서 움직임 벡터를 구한다. 그러므로 탐색 영역을 크게 설정하면 움직임 보상 영상의 화질이 향상되는 반면 이에 따른 급격한 계산량 증가가 큰 문제이므로 움직임 벡터가 될 수 있는 영역을 예측하여 탐색을 행하는 방법이 필요하다.

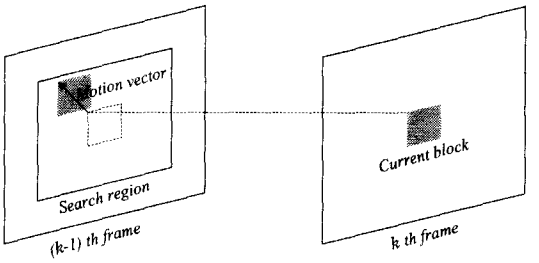


그림 1. 블록 정합 방법에서의 움직임 벡터

2.2 움직임 예측을 이용한 탐색방법

블록 정합 방법으로 움직임을 추정하였을 때, 프레임 내에서 공간적으로 인접한 블록들의 움직임 벡터들은 높은 상관성을 가진다. 따라서 현재 프레임에서 인접 블록의 움직임 벡터들을 이용하여 현재 블록의 움직임 벡터를 예측하고, 이를 기준으로 움직임을 추정하는 방법 (interblock predictive motion estimation, IBPME)이 제안되었다. 이 방법에서는 그림 2에서와 같이 현재 블록의 움직임을 추정하기 위한 탐색 영역을 먼저 탐색한 공간적 인접 블록의 움직임 벡터들로써 예측한 만큼 이동한다. 움직임을 추정할 때에는, 움직임 예측에 의해 이동된 탐색 영역에 대해서 현재 블록과의 블록 정합을 행한다.

한편 현재 블록의 움직임 벡터를 구할 때, 이전 프레임의 같은 위치를 중심으로 인접한 블록들의 움직임 벡터들을 이용하여, 움직임을 예측하고 추정하는 방법 (interframe predictive motion estimation, IFPME)이 제안되었다. 이 방법에서는 그림 3에서와 같이 이전 프레임의 시간적 인접 블록들로써 현재 블록의 움직임 벡터를 예측하고, 예측된 움직임 벡터만큼 현재 블록의 탐색 영역을 이동시켜서 현재 블록의 움직임을 추정한다.

이러한 움직임 예측을 이용한 움직임 추정 방법들에서는, 시간적 및 공간적 인접 블록들의 움직임 벡터들에 가중치를 곱한 합으로써 현재 블록에 대한 예측 움직임 벡터를 구하고, 탐색 영역을 예측 벡터

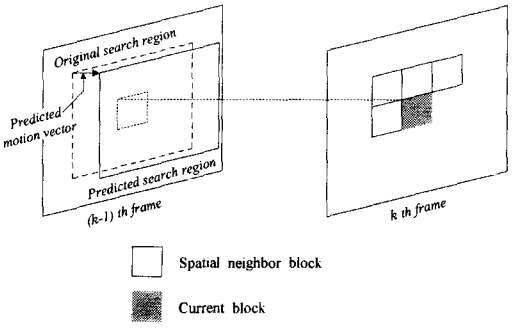


그림 2. 블록 간의 예측을 이용한 움직임 추정 방법

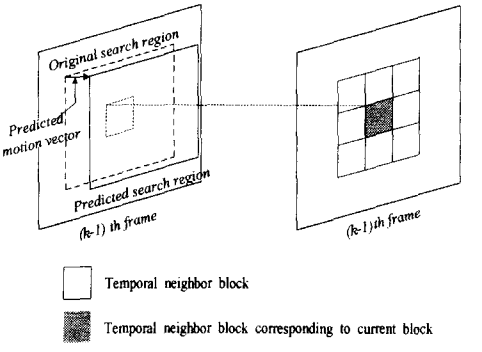


그림 3. 프레임 간의 예측을 이용한 움직임 추정 방법

만큼 이동하여 움직임을 추정 한다. 그러므로 탐색 영역의 크기를 FSA보다 줄이면서 움직임 추정 성능을 향상시킬 수 있었다. 그러나 이 방법에서도 FSA에서와 마찬가지로 탐색 영역의 크기를 적절히 정하기가 어렵다. 또한 인접 블록의 움직임 벡터들이 비슷한 분포를 가지는 영역에 대해서는 이와 같은 움직임 예측이 정확하지만, 인접 블록의 움직임 벡터들의 상관성이 떨어질 때에는, 한 개의 예측 벡터가 모든 인접 블록들의 움직임을 제대로 표현할 수가 없어 예측오차가 커지는 단점이 있다.

3. 움직임의 시간적 연속성을 이용한 탐색 방법

3.1 움직임 벡터의 시간적 연속성을 이용한 탐색 방법 (TPME)

현재 프레임의 움직임 특성을 판단할 수 있는 가장 좋은 근거는 이전 프레임에서의 움직임이며, 시간적 인접 블록들의 움직임을 이용하면 현재 블록의 움직임을 효율적으로 예측할 수 있다. 이전 프레임의 움직임 벡터는 이미 부호화 과정에서 알고 있는 것이므로 부가적인 계산량이 필요 없다.

동영상 신호에서 프레임 간의 시간 간격은 1/30초 정도로 짧기 때문에, 그림 4에서와 같이 어떤 프레임에서 물체 또는 영역의 움직임이 나타났다면 이 움직임은 프레임의 진행에 따라 계속되는 성질이 있다. 즉, 이전 프레임의 어떤 영역이 시간적으로 이동하여 현재 프레임의 한 블록으로 나타났다면, 다음 프레임에서 이 블록은 이전 프레임에서 현재 프레임으로 이동한 만큼 옮겨서 나타날 확률이 높다.

그러나 블록 단위의 움직임 추정 방법인 BMA의 특성상 프레임의 시간적인 진행에 따라 블록 격자가

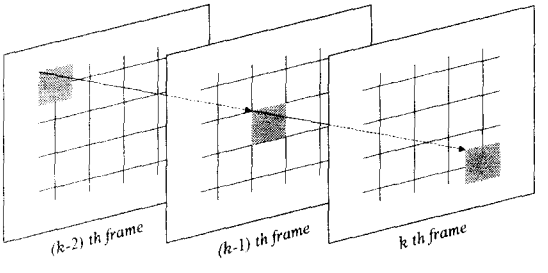


그림 4. 움직임의 시간적 연속성

바뀌므로 현재 블록의 움직임은 이전 블록들의 부분들이 현재 블록을 구성하는 만큼 움직임 특성이 혼합되어서 표현된다. 따라서 현재 블록의 움직임을 예측할 때에는, 한 개의 예측 움직임 벡터로 표현하는 것보다 탐색 영역 전체에 속하는 모든 인접 블록들의 움직임 벡터를 이용하여 예측하여야 한다. 블록의 움직임은 시간적 연속성을 가지며, 현재 블록의 탐색 영역에 속하는 시간적 인접 블록들의 움직임 벡터를 이용하면, 현재 블록의 움직임 특성을 효과적으로 표현할 수 있다. 시간적 인접 블록의 움직임 벡터는 이미 이전 프레임의 부호화시에 구한 것이기 때문에 부가적인 계산이 필요 없다. 또한, 여러 개의 인접 블록들 각각의 움직임 특성을 모두 고려하여 현재 블록의 움직임을 예측하므로, 한 개의 예측 움직임 벡터로 현재 블록의 움직임을 예측하는 기존의 예측 움직임 추정 방법들에 비해서 견실한 예측 (robust prediction)이 가능하다.

TPME 방법에서는, 그림 5에서와 같이 현재 블록 크기의 2배인 넓은 범위의 전역 탐색 영역 (GSR)을 설정한 후 이에 대한 전역탐색을 행하지 않고 다음과 같이 움직임 예측을 통해 탐색영역을 제한해 간다. 먼저 현재 블록 $B_k(m,n)$의 움직임을 예측하기 위한, 이전 프레임의 시간적 인접 블록들의 범위를 나타내는 인덱스 집합을 Θ_w라 하면, 시간적 인접 블록들의 집합 $\Omega_w(m,n)$는,

$$\Omega_w(m,n) = \bigcup_{p,q\in\Theta_w} B_{k-1}(m+p,n+q) \qquad (1)$$

과 같고, 탐색 영역에 포함되는 시간적 인접 블록들이 $\Omega_w(m,n)$에 속하게 된다.

이전 프레임의 시간적 인접 블록들 중에는, 블록의 실제 움직임 (real motion) 방향이 현재 블록 위치로 향하는 것이 있고, 현재 블록과 멀어지는 방향의 것도 있다. TPME 방법에서는 전역 탐색 방법을 기반으로 움직임을 예측하고 추정한다. 따라서 현재 블록의 위치를 기준으로, 시간적 인접 블록들의 실제 움직임이 현재 블록 방향인 것만 고려하여 탐색 영역을 예측해 주기 위해서, 시간적 인접 블록의 움직임이 현재 블록 위치와 상반되는 방향이면 이 블록은 현재 블록의 움직임 예측에서 제외시킨다.

실제 움직임이 현재 블록 방향인 것들의 집합을 $\Omega_c(m,n)$이라 하면, 현재 블록 $B_k(m,n)$의 시간적

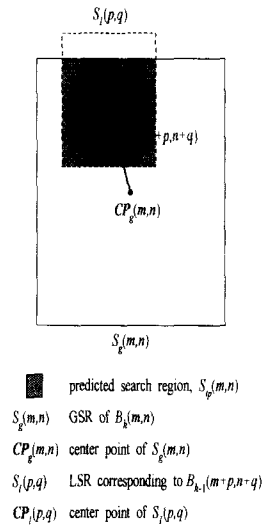


그림 5. 시간적 인접 블록의 움직임 벡터를 이용한 탐색 영역 예측

인접 블록들의 집합 $\Omega_w(m,n)$에 속하는 블록들 중에서, $\Omega_c(m,n)$에 속하는 인접 블록들은,

Procedure *Find* $\Omega_c(m,n)$ *from* $\Omega_w(m,n)$

for ($p,q \in \Theta_w$) *do*
case $p=q=0$:
$B_{k-1}(m+p,n+q) \in \Omega_c(m,n)$
end case
case $p=0$ and $sig(q)=sig(v^h_{p,q})$:
$B_{k-1}(m+p,n+q) \in \Omega_c(m,n)$
end case
case $sig(p)=sig(v^v_{p,q})$ and $q=0$:
$B_{k-1}(m+p,n+q) \in \Omega_c(m,n)$
end case
case $sig(p)=sig(v^v_{p,q})$ and $sig(q)=sig(v^h_{p,q})$:
$B_{k-1}(m+p,n+q) \in \Omega_c(m,n)$
end case

end

end *Find* $\Omega_c(m,n)$ *from* $\Omega_w(m,n)$ (2)

과 같이 구한다. 여기서, $v^v_{p,q}$, $v^h_{p,q}$는 각각 $V_{k-1}(m+p,n+q)$의 수직, 수평 성분, 즉,

$$V_{k-1}(m+p,n+q) = [v^v_{p,q}, v^h_{p,q}]^T \quad (3)$$

와 같다. $S_g(m,n)$의 중심점을 $CP_g(m,n)$이라 하고, 이전 프레임의 $(m+p,n+q)$ 위치의 인접 블록 $B_{k-1}(m+p,n+q)$의 움직임 벡터 $V_{k-1}(m+p,n+q)$로 예측한 현재 블록 크기의 국부 탐색 영역(local search region, LSR)을 $S_l(p,q)$라 하면, $S_l(p,q)$의 중심점 $CP_l(p,q)$는,

$$CP_l(p,q) = CP_g(m,n) + V_{k-1}(m+p,n+q) \quad (4)$$

와 같이 구해지고, 이는 그림 5에서와 같이 표현된다. 이 때에, $S_l(p,q)$의 크기는 $S_g(m,n)$ 크기보다 작다.

TPME 방법에서는, 기존의 탐색 영역이 움직임의 시간적 연속성을 이용하여 구한 새로운 탐색 영역 $S_{tp}(m,n)$, 즉

$$S_{tp}(m,n) = S_g(m,n) \cap S_l(p,q) \quad (5)$$

으로 바뀌게 된다. 이 방법에서는 FSA 기반으로 탐색 영역을 설정하므로, 그림 5에서와 같이 $S_g(m,n)$에 포함되는 예측 탐색 영역에 대해서만 탐색을 행한다.

LSR은 현재 블록의 크기이므로 이전 프레임의 움직임이 연속될 가능성이 있는 블록은

$$B_{k-1}(m+p,n+q),\ -1 \le p \le 1,\ -1 \le q \le 1 \quad (6)$$

의 9개에 국한된다. 예로 그림 6에서와 같이 4개의

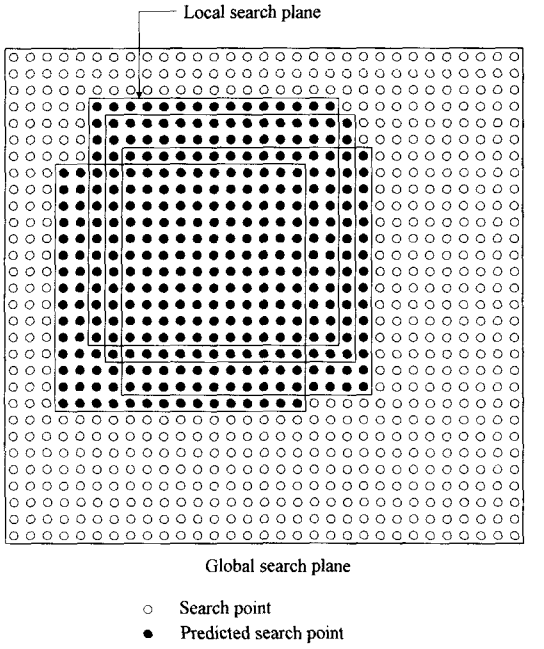


그림 6. TPME 방법에서의 탐색 영역

움직임 연속이 일어났을 경우 실제 움직임 추정은 검은색 탐색점들에 대해서만 행한다. 전역 탐색 영역과 국부 탐색 영역이 겹치는 탐색점에 대해서만 탐색을 행하며, 전역 탐색 범위를 벗어나서 예측되는 탐색점들에 대해서는 탐색을 행하지 않는다. 따라서 TPME 방법은 적응적인 움직임 추정을 행하므로, 일반적으로 예측된 국부 탐색 영역들이 서로 많이 중첩되므로 탐색해야할 탐색점 수가 줄어 계산량을 줄일 수 있고, 시간적 인접 블록들의 움직임이 불규칙할 경우에는 국부 탐색 영역들이 서로 분산되므로 탐색해야 할 탐색점 수가 증가하여 정확한 움직임 추정을 할 수 있게 된다.

3.2 제안한 움직임의 시간적 연속성을 이용한 적응적 탐색 방법 (ATPME)

움직임의 시간적 연속성을 이용하면 현재 프레임에서 현재 블록의 위치는 시간적 인접 블록들의 움직임의 진행선 상에 존재할 확률이 높다는 것을 알 수 있다. 그러므로 현재 블록의 탐색 영역을 구성할 때에, 시간적 인접 블록들의 움직임 벡터로써 예측되는 LSR은 각각의 움직임 벡터의 방향 및 크기에 맞게 설정되어야 함이 타당하다.

그림 7에서는 적응적 탐색 영역의 예측에 대해서 나타내었는데, 이전 프레임에서 블록의 움직임을 알고 있으므로, 움직임의 시간적 연속성을 이용하면 이 블록이 현재 프레임에서 어느 위치로 올 것이라는 것을 예측할 수 있으며, 탐색 영역을 블록의 움직임에 맞추어서 적응적으로 구성하고 탐색을 행하여야 한다.

따라서 본 논문에서는 움직임의 시간적 연속성을

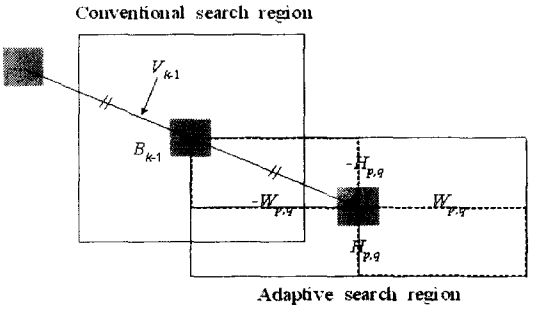


그림 7. 적응적 탐색 영역의 예측

이용한 적응적 움직임 추정 (ATPME)방법을 제안한다. 이 방법에서는 시간적 인접 블록의 움직임 벡터로써 예측하는 LSR을 그 움직임 벡터의 크기에 따라 적응적으로 설정한다.

제안한 ATPME 방법을 설명하면 다음과 같다. 먼저 이 방법에서 현재 블록 $B_k(m,n)$의 전역 탐색 영역을 $S_{ga}(m,n)$라 하면, $S_{ga}(m,n)$의 크기는 TPME 방법에서의 전역 탐색 영역 $S_g(m,n)$와 같다. 그러므로 TPME 방법에서와 같이 시간적 인접 블록들 중 현재 블록 방향의 움직임을 나타내는 것들을 구한다. $(m+p, n+q)$ 위치의 시간적 인접 블록 $B_{k-1}(m+p, n+q)$의 움직임 벡터 $V_{k-1}(m+p, n+q)$를 이용하여 구하는 국부 탐색 영역을 $S_{la}(p,q)$라 하고, $S_{la}(p,q)$의 수직 및 수평 방향의 크기를 움직임 벡터의 크기에 따라 각각 $-H_{p,q} \sim +H_{p,q}$ 및 $-W_{p,q} \sim +W_{p,q}$로 정한다.

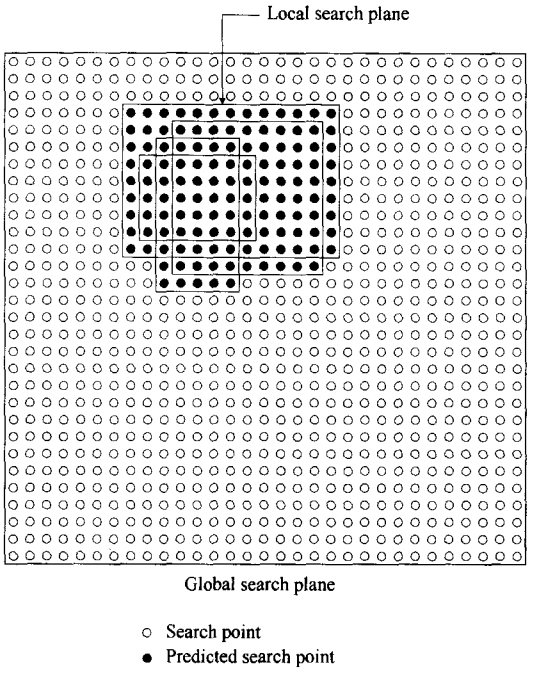
여기서,

$$H_{p,q} = |v^v_{p,q}| + g$$
$$W_{p,q} = |v^h_{p,q}| + g \quad (7)$$

와 같으며, $v^v_{p,q}$, $v^h_{p,q}$는 식 (3)에서와 같이 각각 $V_{k-1}(m+p, n+q)$의 수직, 수평 성분을 나타낸다. 이는 그림 7에서 나타나있다. 그리고 g는 오프셋(offset)으로서, 만약 움직임 벡터가 정 수평 또는 정 수직 방향일 경우 LSR의 크기가 '0' 이 되는 것을 막기 위해서 주어진다.

그림 8에서는 ATPME 방법으로 시간적 인접 블록들의 움직임이 연속된 예를 보여주고 있는데 각각의 움직임 벡터들에 의해 적응적인 탐색영역이 생성됨을 볼 수 있고, 이 중 검은색 점들은 시간적 인접 블록들의 움직임 벡터로 예측된 탐색점을 나타내며, 실제 움직임 추정은 검은색 탐색점들에 대해서만 행한다. 시간적 인접 블록들의 움직임이 균일하게 분포할 경우에는 예측된 국부 탐색 영역들이 서로 많이 중첩되므로 탐색해야할 탐색점 수가 줄어들고, 시간적 인접 블록들의 움직임이 불규칙할 경우에는 국부 탐색 영역들이 서로 분산되므로 탐색해야 할 탐색점 수가 증가한다.

실제 영상에 대해서 블록 정합 방법으로 움직임을 추정해보면, 움직임이 작은 경우가 큰 경우보다 훨씬 많다. 제안한 ATPME 방법에서는 LSR의 크기가 움직임의 크기 및 방향에 따라서 적응적으로 변화되므



(a)

그림 8. 제안한 ATPME 방법에서의 탐색 영역 예측

로, TPME 방법에서보다 더 효율적으로 탐색 영역이 예측되고 탐색점 수도 훨씬 줄어들게 된다.

제안한 적응적 움직임 추정 방법 (ATPME)에서는 탐색 영역의 크기 및 형태를 예측된 움직임 벡터에 맞게 적응적으로 구성함으로써, 효율적인 움직임 추정이 가능하게 하였다. 이 방법은 알고리즘이 간단하며, 영상의 특성에 맞게 계산량과 화질간의 절충(trade off)이 자동으로 이루어지는 장점이 있다.

4. 실험결과 및 고찰

제안한 방법에 대한 성능을 확인하기 위하여 모의실험을 행하였다. 실험 영상으로는, 352×240 크기와 256 밝기의 SIF (source input format) 영상인 FOOTBALL, FLOWER GARDEN, MOBILE, 및 TABLE TENNIS 등의 영상들을 사용하였고, 각 영상의 프레임 수는 60 프레임으로 하였다. FOOTBALL 영상에는 운동 선수의 움직임이 크고 불규칙적인 움직임이 많이 존재하며, FLOWER GARDEN 영상에는 카메라의 패닝 (panning)으로 인해 큰 나무가 일정하게 움직이고 있으며, MOBILE 영상은 고정된 배경에 기차가 움직이는 영상으로서 움직임이 비교적 작고 일정하며, TABLE TENNIS 영상에는 프레임이 진행되면서 카메라의 줌 아웃(zoom out)이 있으며 선수의 팔 부분과 탁구 라켓의 움직임과 탁구공의 빠른 움직임이 존재한다.

움직임 추정을 위한 블록 크기는 16×16으로 두었으며, 제안한 방법과 기존의 방법의 비교를 위한 탐색범위는 표 1에서와 같이 두었다. 그리고 왜곡 척도(distortion measure)로는 계산량이 적으면서 성능이 우수한 MAD를 사용하였으며, 화질의 척도로 PSNR (peak signal-to-noise ratio)을 이용하였다.

$$PSNR = 10\log_{10}\frac{255^2}{\frac{1}{XY}\sum_{x=0}^{X-1}\sum_{y=0}^{Y-1}|I_k(x,y)-\hat{I}_k(x,y)|^2} \quad (8)$$

이때 X, Y는 블록의 가로 및 세로의 크기이다.

또한 제안한 방법과 기존 방법들과의 계산량 비교는 블록 당 평균 탐색점 수로 비교하였는데, −15~+15 탐색 범위의 FSA 1에서 블록 당 평균 탐색점 수는 961 ($=31^2$) 이며, 이를 100 %로 두고서 각 방법들의 상대적인 계산량으로 비교하였다.

제안한 ATPME 방법에 대한 모의실험 결과로 표 2에서와 같이 평균 PSNR을 비교하면, 탐색영역이 GSR과 같은 FSA 1이나 LSR을 고정시켜 계산량이 제안한 방법보다 많은 TPME 방법에 비해서는 움직임이 비교적 큰 FOOTBALL 및 TABLE TENNIS 영상에 대해서 0.1 dB 정도 떨어지지만, FSA 2나 기존의 움직임 예측을 이용한 방법들에 비해서는 계산량이 더 적음에도 불구하고 0.4 dB까지 향상됨을 볼 수 있다. 그리고 움직임이 비교적 적은 영상인 FLOWER GARDEN 및 MOBILE 대해서는 제안한 방법이 기존의 방법들과 유사한 PSNR을 나타냄을 알 수 있다.

표 1. 모의 실험에서 기존 방법 및 제안한 방법의 탐색 범위

Algorithms		Search range
Conventional	FSA 1	−15~+15
	FSA 2	−7~+7
	TPME	GSR : −15~+15 LSR : −7~+7
Proposed ATPME		GSR : −15~+15 LSR : Adaptive range

표 2. 기존 방법과 제안한 방법의 평균 PSNR 비교 Unit(dB)

Sequences \ Algorithms	Conventional					Proposed ATPME
	FSA 1	FSA 2	IBPME	IFPME	TPME	
FOOTBALL	22.59	21.93	22.21	22.20	22.42	22.32
FLOWER GARDEN	25.25	25.24	25.24	25.24	25.24	25.24
MOBILE	23.43	23.43	23.43	23.43	23.43	23.43
TABLE TENNIS	29.48	28.91	29.01	29.03	29.22	29.07

제안한 ATPME 방법의 계산량 감소를 확인하기 위해 표 3에서는 FSA 1 방법의 계산량을 100%로 한 각 방법들의 평균 계산량을 상대적인 값으로 나타내었다. 제안한 방법은 특히 움직임이 작고 일정한 MOBILE 영상에 대해서는 계산량이 3.6 %로 급격한 감소를 나타냄을 볼 수 있다.

FLOWER GARDEN 및 TABLE TENNIS 영상에 대해서도 계산량이 6~7.8 %로 급격히 감소하였다. 또한 움직임이 크고 불규칙적인 FOOTBALL 영상에 대해서도 13.1 %로 많은 계산량 감소를 나타낸다. 이는 움직임 특성에 맞는 적응적인 탐색 영역을 구성하기 때문이다.

그림 9은 움직임이 큰 FOOTBALL 및 TABLE TENNIS 영상 60 프레임에 대한 연속적인 PSNR 비교를 그래프로 나타내고 있다. 제안한 방법은 탐색 영역의 최고 상한 범위에 대해 모두 탐색을 행하는 FSA 1 방법에 근접한 성능을 보였으며, 이는 대부분의 프레임에서 균일한 특성으로 나타남을 알 수 있다.

그림 10에서는, FOOTBALL 영상의 21번째 프레임을 20번째 프레임으로 움직임 추정하여, 움직임 보상 영상을 나타내었다. FOOTBALL 영상에서는 사

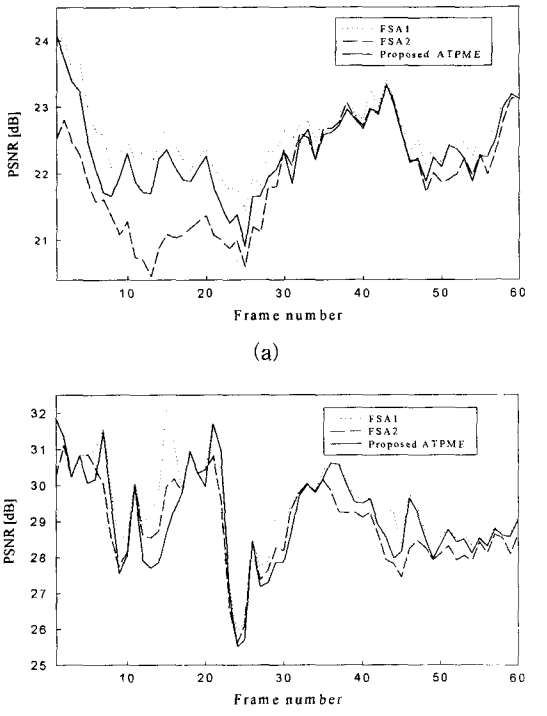


그림 9. (a) FOOTBALL 및 (b) TABLE TENNIS 영상에 대한 FSA와 제안한 ATPME 방법의 PSNR 비교

표 3. 기존 방법과 제안한 방법의 계산량 비교 Unit (%)

Sequences \ Algorithms	Conventional					Proposed ATPME
	FSA 1	FSA 2	IBPME	IFPME	TPME	
FOOTBALL	100.0	23.4	23.4	23.4	32.4	13.1
FLOWER GARDEN	100.0	23.4	23.4	23.4	24.9	6.0
MOBILE	100.0	23.4	23.4	23.4	23.7	3.6
TABLE TENNIS	100.0	23.4	23.4	23.4	23.8	7.8

(a)

(b)

그림 10. (a) FSA 2 및 (b) 제안한 ATPME 방법으로 움직임 보상 된 FOOTBALL 영상

람의 움직임이 아주 크고 불규칙적이다. 그러므로 -7 ~ +7의 탐색 영역으로 움직임을 추정하는 FSA 2 방법인 경우, (a)에서와 같이 화면의 중앙 위쪽에서 빠른 움직임을 나타내고 있는 선수의 발 부분에서 움직임 추정이 제대로 이루어지지 않았음을 알 수 있다. 그러나 제안한 ATPME 방법으로 움직임을 추정하였을 경우에는, (b)에서와 같이 FOOTBALL 영상에서 큰 움직임이 있는 화면 우측의 사람의 등 부분와 선수의 발 부분에서 완벽하게 움직임이 추정됨을 확인할 수 있다.

5. 결 론

본 논문에서는, 움직임의 시간적 연속성을 이용하여 현재 블록의 시간적 인접 블록, 즉 이전 프레임에서 현재 블록으로 움직임이 연속될 가능성이 있는 블록의 움직임 벡터들을 이용하여 탐색 영역을 적응적으로 정하여 효율적인 탐색을 행하는 움직임의 시간적 연속성을 이용한 적응적 움직임 탐색 방법을 제안하였다.

이를 이용하면 움직임 특성에 따라 탐색 영역이 구성되므로, 영상의 특성에 따라 계산량과 화질간의 절충 (trade off)이 적응적으로 이루어진다. 즉, 움직임이 작고 규칙적인 영상에 대해서는 탐색 영역이 작아져서 계산량이 급격히 감소하게 되며, 움직임이 크고 불규칙한 영상에 대해서는 예측 탐색 영역이 자동으로 넓게 구성되므로, 정확한 움직임 추정이 가능하다.

제안한 ATPME 방법은 모든 영상에 대해서 기존 방법들에 비해 우수한 PSNR과 급격한 계산량 감소를 나타냄을 확인할 수 있었다.

참 고 문 헌

[1] B. G. Haskell, A. Puri, and A. N. Netravali, Digital Video: An Introduction to MPEG-2, New York, Chapman and Hall, NY, 1997.

[2] K. R. Rao and J. J. Hwang, Techniques and Standards for Image, Video, and Audio Coding, Upper Saddle River, Prentice Hall, NJ, 1996.

[3] J. Lu and M. L. Liou, "A simple and efficient search algorithm for block-matching motion estimation," IEEE Trans. Circuits Syst. Video Technol., Vol. 7, No. 2, pp. 429-433, 1997.

[4] L. Luo, C. Zou, X. Gao, and Z. He, "A new prediction search algorithm for block motion estimation in video coding," IEEE Trans. Consumer Electronics, Vol. 43, No. 1, 1997.

[5] S. Zafar, Y.-Q. Zhang, and J. S. Baras, "Predictive block-matching motion estimation for TV coding--Part I: Inter-block prediction," IEEE Trans. Broadcasting, Vol. 37, No. 3, pp. 97-101, 1991.

[6] C.-H. Hsieh, P.-C. Lu, J.-S. Shyn, and E.H. Lu, "Motion estimation algorithm using interblock correlation," IEE Electronics Letters, Vol. 26, No. 5, 1990.

[7] R. Li, B. Zeng, and M. L. Liou, "A new

three-step search algorithm for block motion estimation," IEEE Trans. Circuits Syst. Video Technol., Vol. 4, No. 4, pp. 438-442, 1994.

[8] J-H Lim and H-W Choi, "Adaptive motion estimation algorithm using spatial and temporal correlation," IEEE Pacific Rim Conference on Communications, Computers and signal Processing, Vol. 2, pp. 473-476, 2001.

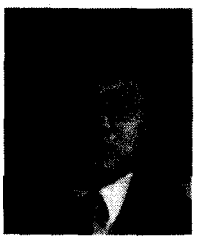
[9] J. Feng, K.-T. Lo, H. Mehrpour, and A. E. Karbowiak, "Adaptive block matching algorithm for video compression," IEE Proc. Vision, Image and Signal Processing, Vol. 145, Iss. 3, pp. 173-178, 1998.

[10] J. Lee and B. W. Dickinson, "Temporally adaptive motion interpolation exploiting temporal masking in visual perception," IEEE Trans. Image Process., Vol. 3, No. 5, 1994.

[11] Y.-Q. Zhang and S. Zafar, "Predictive block-matching motion estimation for TV coding--Part II: Inter-frame prediction," IEEE Trans. Broadcasting, Vol. 37, No. 3, pp. 102-105, 1991.

[12] W-R Sung, E-K Kang, J-S Choi, "Adaptive motion estimation technique for motion compensated interframe interpolation," IEEE Transactions on Consumer Electronics, Vol. 45I, Iss. 3, pp. 753-761, 1999.

[13] 이경환, 류권열, 최정현, "움직임 벡터의 시간적 연속성을 이용한 고속 움직임 추정 알고리즘," 한국멀티미디어학회 논문지, 제6권, 제7호, pp. 1121-1130, 2003.



최 정 현

1991년 2월 경북대학교 전자공학과 졸업
1993년 2월 경북대학교 대학원 전자공학과 졸업(공학석사)
2000년 2월 경북대학교 대학원 전자공학과 졸업(공학박사)
2000년 9월~2001년 3월 기술신용보증기금 부산기술평가센터 차장
2001년 3월~현재 기술신용보증기금 대구기술평가센터 차장
관심분야 : 영상신호처리 및 압축

이 경 환

1994년 2월 경북대학교 전자공학과 졸업(공학사)
1996년 2월 경북대학교 대학원 전자공학과 졸업(공학석사)
2000년 8월 경북대학교 대학원 전자공학과 졸업(공학박사)
2001년 3월~현재 위덕대학교 멀티미디어공학과 조교수
관심분야 : 영상 및 음향신호처리 및 압축, 멀티미디어 프로그래밍