

블록 유형 분류에 의한 움직임 추정

윤 호 순[†] · 유 재 명^{**} · 박 미 선^{***} · 김 미 영^{****} · Nguyen Dinh Toan^{***} · 이 귀 상^{*****}

요 약

움직임 추정은 비디오 영상 압축에서 중요한 역할을 하지만, 최적의 움직임 벡터를 찾기 위해서는 많은 탐색을 필요로 한다. 따라서 탐색 시간을 줄이기 위하여 적절한 탐색 시작점 설정과 효율적인 탐색 패턴을 사용하여야 한다. 만일 움직임 추정을 수행하기 전에 현재 블록의 전체적인 움직임 정보를 예측할 수 있다면 예측된 움직임 정보를 이용하여 보다 효율적으로 움직임 추정을 수행할 수 있다. 본 논문에서는 불필요한 움직임 추정을 생략하고 현재 블록의 움직임 추정을 위한 탐색 시작점 위치와 탐색 패턴을 보다 정확히 결정하기 위해, 후보 벡터들을 이용하여 현재 블록 유형을 분류하고 현재 블록의 움직임을 예측하는 방법을 제시한다. 제안된 기법은 움직임 벡터 추정 속도 면에서 기존의 방법과 비교하여 약 1.5~7배의 속도 향상을 가져왔으며, 움직임이 많은 영상에서의 화질에 있어서 약 0.02~0.2(dB)정도 성능 향상을 보였다.

키워드 : 움직임추정, 디지털 비디오 압축, 움직임 벡터, 탐색 패턴

Motion Estimation by Classification of Block Types

HyoSun Yoon[†] · JaeMyeong Yoo^{**} · MiSeon Park^{***} · Mi-Young Kim^{****} ·
Nguyen Dinh Toan^{***} · GueeSang Lee^{*****}

ABSTRACT

Although motion estimation plays an important role in digital video compression, complex search procedure is required to find an optimal motion vector. To reduce the search time, the search start point should be set up properly and efficient search pattern is needed. If the overall motion of the current block can be predicted, motion estimation can be performed efficiently. In this paper, block types are classified using candidate vectors and the motion activity of the block is predicted, which leads to the search start point close to the optimal motion vector. The proposed method proves to be about 1.5~7 times faster than existing methods with about 0.02~0.2(dB) improvement of picture quality in images with large movements.

Key Words : Motion Estimation, Digital Video Compression, Motion Vector, Search Pattern

1. 서 론

동영상 내에 존재하는 중복된 데이터들은 시간적, 공간적, 통계적 중복성을 이용하여 압축된다. 특히 움직임 추정과 움직임 보상을 수행하여 시간적 중복성을 제거하고 이 때 추정된 움직임 벡터(Motion Vector: MV)에 의해서 보상된 영상과 원 영상과의 차 신호를 부호화함으로써 높은 데이터 압축률을 얻을 수 있다.

현재 많은 비디오 코딩에서는 데이터 흐름의 규칙성, 계산의 복잡도, 하드웨어의 구현을 고려하여 움직임 추정

(Motion Estimation: ME)기법으로 블록 정합 기법을 많이 사용한다. 가장 간단한 블록 정합 기법인 전역 탐색 기법은 움직임 추정 시 탐색범위내의 가능한 모든 블록을 조사하여 움직임 벡터를 찾는다. 전역 탐색 기법은 탐색범위 내에서 가장 적합한 움직임 벡터를 구할 수 있지만 많은 계산량을 요구한다. 전역 탐색 기법의 계산량을 줄이기 위하여 여러 가지 고속 블록 정합 기법(Fast Block Matching Algorithm: FBMA)들이 제안되었는데 대표적인 고속 블록 정합 기법에는 3단계 탐색(Three Step Search:TSS)[1], 새로운 3단계 탐색(New Three Step Search:NTSS)[2], 2차원 로그형 탐색(2 Dimension LOGarithmic search:2DLOG)[3], 4단계 탐색(Four Step Search:4SS)[4], 다이아몬드 탐색(Diamond Search:DS)[5][6], 2단계 탐색(2 Step Search:2SS)[7][8] 등이 있다. 그리고 움직임 벡터간에 존재하는 시간적 상관성 또는 공간적 상관성을 이용하여 현재 블록의 탐색 패턴과 탐색 시작점을 적응적으로 결정하여 움직임 벡터를 추정하는 움직임

* 논문은 2004년도 전남대학교 연구년교수연구비 지원에 의하여 연구되었음.

† 준 회원 : 전남대학교 전산학과 연구원

** 준 회원 : 전남대학교 대학원 전산학과 박사과정

*** 준 회원 : 전남대학교 대학원 전산학과 석사과정

**** 정 회원 : 전남도립남도대학 컴퓨터정보통신과 교수

***** 종신회원 : 전남대 전산학과 교수

논문접수 : 2006년 9월 8일, 심사완료 : 2006년 11월 4일

벡터 필드 적응적 탐색 기법(MVFAST)[9]과 예측된 움직임 벡터 필드 적응적 탐색기법(PMVFAST)[10]들이 있다.

움직임 벡터의 시간적 또는 공간적 상관성을 이용하지 않는 기법들은 움직임이 없는 배경에서부터 움직이는 물체에 이르기까지 동일한 탐색 패턴과 고정된 탐색 시작점을 사용하여 움직임 추정을 수행하기 때문에 움직임 벡터 추정 시 많은 탐색점을 검사하는 문제점을 가지고 있다. 그리고 이러한 문제점은 움직임 벡터 추정 시간에 많은 영향을 미친다. 시간적 또는 공간적 상관성을 이용한 기법들은 현재 블록의 움직임 크기를 예측한 후 예측된 움직임 크기에 따라 적응적으로 탐색 패턴과 탐색 시작점을 선택하여 움직임 추정을 수행하므로 움직임 추정 성능을 향상시켰다[9][10]. 기존 기법들 중 영상 화질면과 움직임 추정 속도 면에서 좋은 성능을 보이는 PMVFAST는 움직임 추정 속도를 빠르게 하기 위하여 다양한 임계치들을 사용한다. 그러나 임계치들은 추가적인 계산량을 요구한다. 이러한 문제점을 해결하고 보다 빠르고 정확하게 움직임 추정을 수행할 수 있도록 본 논문에서는 시·공간적 상관성을 가진 주변 블록들의 움직임 벡터들을 이용하여 현재 블록 유형을 분류하고 현재 블록의 움직임을 예측하여 움직임 추정을 수행하는 기법을 제안한다. 즉, 제안한 기법은 현재 블록 유형에 따라 움직임 추정이 불필요한 블록에서 움직임 벡터 추정을 생략함으로써 움직임 추정 속도를 향상시켰고 예측된 움직임을 크기에 따라 현재 블록의 색 시작점과 탐색 패턴을 적응적으로 선택하여 움직임 추정을 수행한다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 움직임 추정 기법에 대하여 설명하고, 3장에서는 제안한 기법을 기술한다. 그리고 4장에서는 기존 기법들과 성능을 비교한 후, 마지막으로 5장에서는 결론을 맺는다.

2. 기존의 움직임 추정(Motion Estimation)기법

전송매체와 저장매체를 보다 효율적으로 사용할 수 있도록 동영상을 압축하는데 가장 일반적으로 사용되는 방법은 블록 단위의 움직임 추정기법이다. 블록단위의 움직임 추정은 두 가지 전제 조건을 가지고 있는데, 동일한 블록내의 화소들은 동일한 움직임을 갖는다는 것과 블록들은 수평, 수직으로만 움직인다는 것이다. 즉 블록 정합 기법은 영상의 한 프레임을 동일한 크기의 블록들로 나누고 이들의 각 블록에 대하여 참조 프레임(Reference Frame)의 탐색 영역 내에서 정합 오차가 가장 작은 블록을 찾는다. 이때 현재 프레임(Current Frame)의 한 블록과 참조 프레임 내에서 가장 정합이 잘 되는 블록간의 위치차이를 움직임 벡터라 한다. 움직임 벡터를 추정하기 위하여 많은 고속 블록 정합 기법들이 제안되었는데, 3단계 탐색 기법[1]은 넓은 영역에 걸쳐 몇 개의 탐색점을 조사한 후 점차 범위를 좁혀 나가는 방식이고, 새로운 삼단계 탐색 기법[2]은 움직임 벡터가 탐색 영역의 중심에 분포한다는 사실을 이용하여 3단계 탐색 기법을 보완한 기법이며, 4단계 탐색 기법[4]은 새로운 3단

계 탐색 기법의 계산량을 개선한 기법이다. 그리고 2차원 로그형 탐색기법[3]은 대부분의 움직임이 상하좌우 방향으로 일어난다는 사실을 이용한 기법이고, 다이아몬드 탐색 기법[5][6]은 움직임 벡터의 분포의 형태가 다이아몬드라는 사실을 이용한 기법이다.

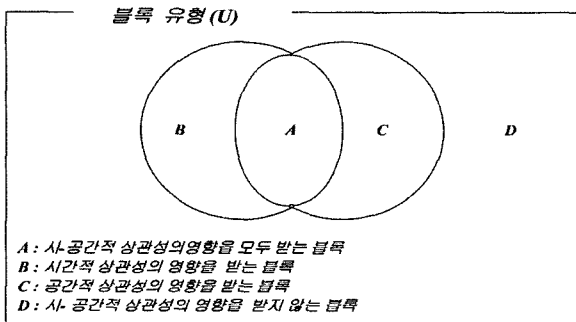
위에 기술한 기법들은 현재 블록의 움직임 크기에 무관하게 고정된 탐색 패턴과 고정된 탐색 시작점을 사용하여 움직임 벡터를 추정하므로 움직임 벡터 추정 시 많은 탐색점들을 검사한다. 이러한 문제점을 해결하기 위하여 움직임 벡터의 시간적 또는 공간적 상관성을 이용하여 현재 블록의 움직임을 예측하고 예측된 움직임에 맞게 탐색 패턴과 탐색 시작점을 적응적으로 선택하여 움직임 벡터를 추정하는 기법들이 제안되었다. 특히, 움직임 벡터 필드 적응적 탐색 기법과 예측된 움직임 벡터 필드 적응적 탐색 기법들은 움직임 추정에 있어서 상당한 성능 향상을 가져왔다.

움직임 벡터 필드 적응적 기법[9]은 탐색 원점에서의 블록 정합 오차가 임계치 보다 적으면 움직임 추정을 종료함으로써 움직임 추정 속도를 향상시켰고 예측된 움직임 벡터 필드 탐색 기법[10]은 초기 단계에서 움직임 추정을 종료시키기 위하여 다양한 임계치를 사용하여 움직임 추정 속도를 향상시켰다. 그러나 임계치들을 계산하는데 있어서 추가적인 계산량을 요구한다. 그리고 움직임 벡터 필드 적응적 탐색 기법과 예측된 움직임 벡터 필드 적응적 탐색 기법들은 초기 단계에서 움직임 벡터 추정을 종료하기 위하여 적어도 하나 이상의 탐색점을 검사하고 움직임이 있는 영상에서 잘못된 움직임 예측으로 인하여 움직임 벡터 추정 시 많은 탐색점들을 사용하는 문제점들을 가지고 있다.

3. 블록 유형 분류에 의한 움직임 추정

동영상에서 인접 프레임간의 시간 간격은 매우 짧기 때문에 단위 프레임의 시간당 움직임 크기 변화량은 적은 범위로 제한된다고 볼 수 있다. 즉 연속하는 두 프레임간의 움직임에 높은 시간적 상관성이 존재하므로 현재 블록의 움직임 벡터는 참조 프레임에서 동일한 위치에 있는 블록의 움직임 벡터와 유사하다. 그리고 프레임내의 인접하는 화소들의 값이 매우 유사하므로 현재 블록의 움직임 벡터는 인접하는 블록의 움직임 벡터들과 유사하다. 즉, 높은 공간적 상관성이 존재한다.

가장 좋은 성능 향상을 보인 움직임 벡터 필드 적응적 기법[9]과 예측된 움직임 벡터 필드 탐색 기법[10]들은 움직임 벡터의 시·공간적 상관성을 이용하여 움직임 추정을 수행한다. 그러나 이 기법들은 움직임이 있는 영상에서 잘못된 움직임 예측으로 인하여 잘못된 탐색 패턴과 탐색 시작점을 사용한다. 잘못된 탐색 패턴과 탐색점 사용은 움직임 벡터 추정 속도와 영상 화질에 영향을 미친다. 그리고 예측된 움직임 벡터 필드 탐색 기법은 초기 단계에서 움직임 추정을 종료하기 위하여 다양한 임계치를 사용함으로써 움직임 추정 속도를 향상시켰지만 임계치들을 계산하는데 있어서 추



(그림 1) 시·공간적 상관성에 따른 블록 유형

가적인 계산량을 요구한다.

본 논문에서는 움직임이 있는 영상에서 보다 빠르고 정확하게 움직임 추정을 수행하기 위하여 블록 유형을 분류하고 블록의 움직임을 예측하여 움직임 추정을 수행하는 기법을 제안한다. 제안한 기법은 분류된 블록 유형에 따라 움직임 추정이 불필요한 블록에서 움직임 벡터 추정을 생략함으로써 초기 단계에서 움직임 추정을 종료한다. 즉 추가적인 계산량을 요구하는 임계치들을 사용하지 않고 초기 단계에서 움직임 추정을 종료할 수 있다. 그리고 예측된 움직임을 크기에 따라 현재 블록의 탐색 시작점과 탐색 패턴을 적응적으로 선택하여 움직임 추정을 수행한다.

제안한 움직임 추정 기법은 움직임이 있는 영상에서 보다 빠르고 정확하게 움직임 추정을 수행할 수 있도록 블록들을 다음과 같이 분류하였다. 일반적으로 블록들을 (그림 1)와 같이 A, B, C, D로 분류할 수 있고 블록은 A, B, C, D 중 하나에 속한다. (그림 1)의 A는 시·공간적 상관성의 영향을 동시에 받는 블록, B는 시간적 상관성의 영향을 받는 블록, C는 공간적 상관성을 받는 블록이고 D는 시·공간적 상관성의 영향을 모두 받지 않는 블록이다.

제안한 기법은 블록 유형을 분류하고 블록의 움직임을 예측하기 위하여 (그림 2)에 있는 움직임 벡터들과 (0,0)을 후보 벡터들로 이용한다. MV_0 과 MV_1 은 현재 블록과 시간적 상관성을 가진 블록들의 움직임 벡터들이고 MV_2 , MV_3 와 MV_4 는 현재 블록과 공간적 상관성을 가진 블록들의 움직임 벡터들이다.

제안 기법은 (그림 2)의 후보 벡터들을 이용하여 블록을 4가지 유형으로 분류한다.

[블록 유형1]: $MV_0 = MV_1 = MV_2 = MV_3 = MV_4$

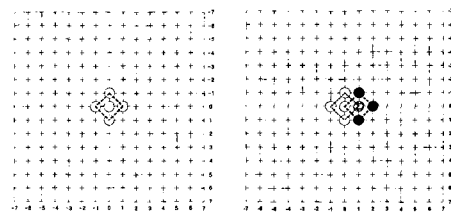
[블록 유형2]: $MV_0 = MV_1$, 블록 유형 1 제외

[블록 유형3]: $MV_2 = MV_3 = MV_4$ 또는 MV_2, MV_3, MV_4 중 2개가 일치한 경우

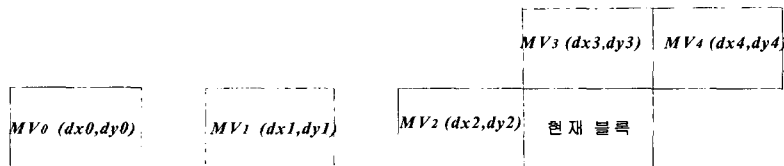
[블록 유형4]: 블록 유형1, 블록 유형2, 블록 유형3을 제외한 경우 제안한 기법은 블록 유형1인 경우, 현재 블록을 시·공간적 상관성의 영향을 모두 받는 블록으로 판단하고 후보 벡터들의 움직임 벡터를 현재 블록의 움직임 벡터로 추정한다. 즉 블록 유형1에 속한 경우 움직임 벡터 추정을 수행하지 않으므로 움직임 추정 속도를 향상시킬 수 있다.

블록 유형2 이거나 블록 유형3인 경우, 현재 블록이 시간적 상관성 또는 공간적 상관성의 영향을 받는 블록이라고 판단한다. 공간적 상관성의 영향을 받는 블록들의 움직임 벡터들 MV_2, MV_3, MV_4 의 중간값 움직임 벡터 MV_{median} 을 구한 후, 시간적 상관성의 영향을 받는 블록의 움직임 벡터 MV_1 과 MV_{median} 에서의 블록 정합 오차를 각각 계산한다. 최소 블록 정합 오차를 가진 점의 근처에 현재 블록의 움직임 벡터가 존재할 가능성이 높으므로 최소 블록 정합 오차를 가진 점을 현재 블록의 예측된 움직임 벡터 (P_x, P_y)로 결정한다. 즉, 최소 블록 정합 오차를 가진 움직임 벡터의 상관성 영향을 더 받는다.

블록 유형4인 경우, 즉 현재 블록이 시간적 상관성과 공간적 상관성의 영향을 받지 않거나 작게 받는 블록인 경우 (1)에 의하여 (P_x, P_y)을 구하고 (0,0)과 (P_x, P_y)에서 블록 정



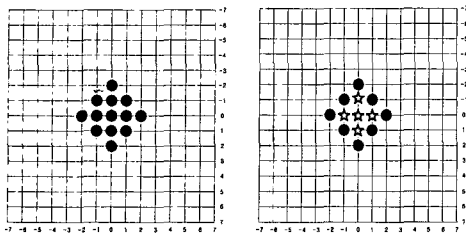
(A) 초기 탐색점 배치 (B) 추가 탐색점
(그림 3) 작은 다이아몬드 탐색 패턴



전 전 프레임 (t-2) 이전 프레임 (t-1) 현재 프레임 (t)

$MV_0(dx_0, dy_0)$: 전 전 프레임에서 현재 블록과의 동일한 위치에 있는 블록의 정 화소 움직임 벡터
 $MV_1(dx_1, dy_1)$: 이전 프레임에서 현재 블록과의 동일한 위치에 있는 블록의 정 화소 움직임 벡터
 $MV_2(dx_2, dy_2)$: 현재 블록의 왼쪽에 있는 블록의 정 화소 움직임 벡터
 $MV_3(dx_3, dy_3)$: 현재 블록의 위쪽에 있는 블록의 정 화소 움직임 벡터
 $MV_4(dx_4, dy_4)$: 현재 블록의 위-오른쪽에 있는 블록의 정 화소 움직임 벡터

(그림 2) 후보 벡터들



(A) 초기 탐색점 (B) 마지막 단계
(C) 최소 블록 오차를 가진 점의 위치가 상,하,좌,우 방향에 있는 경우 추가 탐색점의 위치
(D) 최소 블록 오차를 가진 점의 위치가 대각선 방향에 있는 경우 추가 탐색점의 위치

(그림 4) 수정된 다이아몬드 탐색 패턴

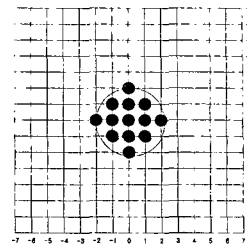
합 오차를 각각 계산한다. 최소 블록 정합 오차를 가진 점을 현재 블록의 예측된 움직임 벡터 (P_x, P_y) 로 결정한다. 즉, 현재 블록이 시간적 상관성과 공간적 상관성을 작게 받는 블록인 경우 탐색 시작점을 예측하기 위하여 MV_1 과 MV_{median} 에 0.5의 가중치를 주어 (P_{x1}, P_{y1}) 을 계산한다. 그리고 현재 블록이 시간적 상관성과 공간적 상관성의 영향을 전혀 받지 않는 블록인 경우 $(0,0)$ 를 탐색 시작점으로 예측한다.

$$P_{x1} = \lfloor dx1 \times 0.5 + median(dx2, dx3, dx4) \times 0.5 \rfloor$$

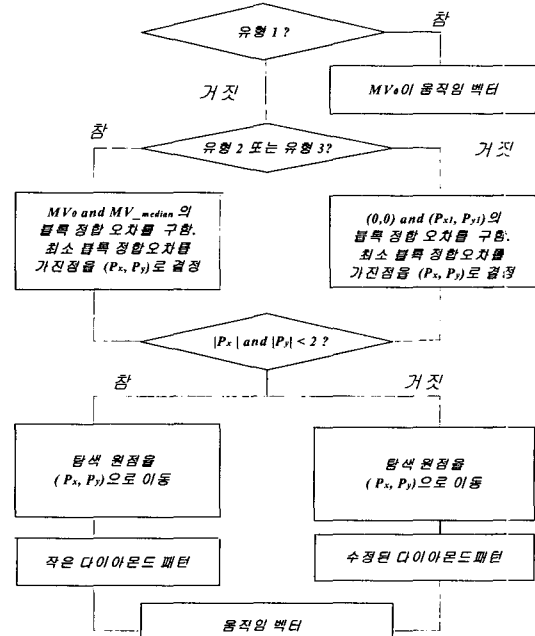
$$P_{y1} = \lfloor dy1 \times 0.5 + median(dy2, dy3, dy4) \times 0.5 \rfloor \quad (식 1)$$

블록 유형2, 블록 유형3, 블록 유형4인 경우, $|P_x| < 2$ 이고 $|P_y| < 2$ 인 경우, (P_x, P_y) 을 탐색 시작점으로 (그림 3)의 작은 다이아몬드 탐색 패턴을 현재 블록의 탐색 패턴으로 결정한 후 움직임 벡터 추정을 수행하고, 그렇지 않는 경우 (P_x, P_y) 을 탐색 시작점으로 (그림 4)의 수정된 다이아몬드 탐색 패턴을 현재 블록의 탐색 패턴으로 결정한 후 움직임 벡터 추정을 수행한다.

다이아몬드 기법은 탐색점 추가 시 탐색 영역의 중심에 있는 일부의 탐색 점들을 정합 에러 계산에서 제외시킴으로 적합한 움직임 벡터를 찾지 못하고 많은 탐색 점들을 사용하는 문제점을 가지고 있다. 그래서 본 논문에서는 다이아몬드 기법을 개선한 수정된 다이아몬드 기법을 사용하여 움직임 추정을 한다. 수정된 다이아몬드 기법은 일반적으로 영상의 움직임 벡터들이 (그림 5)와 같이 탐색 영역 원점을 중심으로 반경 2 pixels 이내에 분포할 확률이 약 56.72%~98.70%[5][6]라는 사실을 이용하여 초기 탐색 점들을 배치하고, 탐색 영역의 중심에서 가장 자리로 점진적으로 움직임



(그림 5) 움직임 벡터 분포도 (움직임 벡터의 분포 확률을 이용한 초기 탐색점 배치)



(그림 6) 제안한 기법의 순서도

추정을 해 나갈 때, 탐색 영역의 중심에 있는 모든 탐색 점들이 블록 정합의 대상이 되도록 추가 탐색 점들을 적절히 배치한다.

(그림 6)은 제안한 기법의 알고리즘을 간략하게 나타낸 것으로 제안한 기법은 후보 벡터들을 사용하여 현재 블록 유형을 결정하고 현재 블록의 움직임을 예측한다. 그리고 예측된 움직임에 따라 탐색 패턴과 탐색 시작점을 적응적으로 결정한다. 제안한 기법은 다음과 같은 알고리즘을 수행하여 움직임 벡터를 추정한다.

<제안 알고리즘 : 블록 유형 분류에 의한 움직임 추정 기법>

- [1단계]: 블록 유형1인 경우, MV_0 을 현재 블록의 움직임 벡터로 결정하고 움직임 추정을 마친다. 즉, 움직임 벡터 추정을 생략한다. 그렇지 않는 경우 2단계를 수행한다.
- [2단계]: 블록 유형2 이거나 블록 유형3인 경우, MV_1 과 MV_{median} 에서의 블록 정합 오차를 구하고 최소 블록 정합 오차를 가진 점을 현재 블록의 예측된 움직임 벡터 (P_x, P_y) 로 결정한다.
- [3단계]: 블록 유형4인 경우, $(0,0)$ 와 (P_{x1}, P_{y1}) 에서의 블록 정합 오차를 구하고 최소 블록 정합 오차를 가진 점을

현재 블록의 예측된 움직임 벡터(P_x, P_y)로 결정한다.

[4단계]: $|P_x| < 2$ 이고 $|P_y| < 2$ 인 경우, 5단계를 수행하고, 그렇지 않는 경우 6단계를 수행한다.

[5단계]: 작은 다이아몬드 탐색

- ① (그림 3)의 (A)와 같이, (P_x, P_y)을 중심으로 5개의 탐색 점들을 배치한 후 각각의 탐색 점들에 대하여 블록 정합을 수행하여 최소 블록 정합 오차를 가진 점을 결정한다. 만약 최소 블록 정합 오차를 가진 점의 위치가 작은 다이아몬드 패턴의 중심에 있는 점이라면 이 점을 움직임 벡터로 추정하고 탐색을 마친다. 그렇지 않는 경우 ②를 수행한다.
- ② 최소 블록 정합 오차를 가진 점의 위치가 작은 다이아몬드 패턴 중심에 있는 점이 아닌 경우, (그림 3)의 (B)와 같이 3개의 추가 탐색 점들을 배치하여 새로운 작은 다이아몬드 패턴을 만든다.
- ③ 최소 블록 정합 오차를 가진 점의 위치가 작은 다이아몬드 패턴의 중심에 있는 점이 될 때까지 위의 단계를 반복 수행하여 움직임 벡터를 추정한다.

[6단계]: 수정된 다이아몬드 탐색

- ① (그림 4)의 (A)와 같이, (P_x, P_y)을 중심으로 13개의 탐색 점들을 배치한 후 각각의 탐색 점들에 대하여 블록 정합을 수행하여 최소 블록 정합 오차를 가진 점을 결정한다. 만약 최소 블록 정합 오차를 가진 점의 위치가 (그림 4)의 (B)와 같이 다이아몬드 패턴 중심에 있는 5개의 점 중 하나이면 이 점을 움직임 벡터로 추정하고 탐색을 마치고, 그렇지 않는 경우 ②를 수행한다.
- ② 최소 블록 정합 오차를 가진 점이 다이아몬드 패턴을 중심으로 상, 하, 좌, 우에 있는 경우, (그림 4)의 (C)와 같이 추가 탐색 점들을 배치한다. 그리고 최소 블록 오차를 가진 점이 다이아몬드 패턴을 중심으로 대각선 방향에 있는 경우, (그림 4)의 (D)와 같이 추가 탐색 점들을 배치하여 새로운 다이아몬드 패턴을 만든다.
- ③ 최소 블록 정합 오차를 가진 점의 위치가 (그림 4)의 (B)와 같이 다이아몬드 패턴 중심에 있는 5개의 점 중 하나가 될 때까지 위의 단계를 반복 수행하여 움직임 벡터를 추정한다.

4. 실험 결과

제안된 기법의 성능을 평가하기 위하여 실험 영상으로 QCIF인 Akiyo, Foreman, Mother and Daughter, Stefan 그리고 Table영상의 각각 150 프레임을 사용하였고, 비교 탐색 기법으로는 전역 탐색 기법(FS), 새로운 삼단계 탐색 기법(NTSS), 4단계 탐색 기법(FSS), 다이아몬드 탐색 기법(DS), 움직임 벡터 필드 적용적 탐색 기법(MVFAST) 그리고 예측된 움직임 벡터 필드 적용적 탐색 기법(PMVFAST)들을 사용하였다. 성능 비교 함수로는 영상 화질의 품질을

평가하기 위하여 PSNR(Peak Signal-to-noise Ratio)을 사용하였는데, PSNR은 다음과 같이 정의된다.

$$MSE = \left(\frac{1}{MN} \right) \sum_{m=1}^M \sum_{n=1}^N [x(m,n) - \hat{x}(m,n)]^2 \quad (\text{식 2})$$

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \quad (\text{식 3})$$

여기에서 (2)에서 M, N은 영상의 가로와 세로의 크기이고, $x(m,n)$ 는 원 영상화면을 나타내고, $\hat{x}(m,n)$ 는 움직임 보상 예측된 화면을 나타낸다. 각 영상에 대한 실험 결과를 표 1, 표 2에 나타냈는데, <표 1>에 각 실험 영상에 대한 PSNR(dB)의 평균을 나타내었고, <표 2>에 각 실험 영상에서 각 블록의 움직임 벡터 추정 시 사용되는 평균 탐색점수를 나타낸다.

<표 1>에 나타나 있는 것과 같이 움직임 보상 예측된 화면의 화질면에 있어서 제안한 기법을 움직임 벡터 필드 적용적 탐색 기법과 예측된 움직임 벡터 필드 적용적 탐색 기법과 비교해 보았을 때, 제안된 기법은 Akiyo, Mother and Daughter처럼 움직임이 작은 영상인 경우에는 0.08~0.3(dB) 정도 화질이 좋지 않다. 왜냐하면 움직임이 작은 영상인 경우에는 시간적 상관성과 공간적 상관성이 매우 높게 존재하므로 제안한 기법을 사용하는 경우 움직임 벡터 추정을 하지 않는 블록이 많아지므로 화질이 저하된다. 그러나 표 2에 나타나 있는 것과 같이 움직임 벡터 추정 속도 면에 있어서 약 5~7배 빠르다.

Foreman, Stefan, Table같이 많은 움직임이 있는 영상인 경우에 제안한 기법을 사용하면 보다 빠르고 정확하게 현재 블록의 움직임을 예측하므로 화질면, 움직임 벡터 추정 속도 면에서 보다 나은 성능을 보였다. 화질 면에서 0.02~0.2(dB)정도 성능이 향상되었고 움직임 벡터 추정 속도 면에 있어서 1.5~2배 성능 향상을 보였다.

<표 1> 각 실험 영상의 PSNR

	FS	NTSS	FSS	DS	MVFAST	PMVFAST	제안기법
Akiyo	34.50	34.48	34.33	34.39	34.39	34.49	34.31
Foreman	29.54	28.19	28.22	28.66	29.00	29.00	29.02
M & D	31.52	31.37	31.34	31.42	31.47	31.47	31.11
Stefan	23.88	22.24	22.62	22.77	23.36	23.45	23.63
Table	26.50	25.60	24.81	25.67	25.59	25.63	25.71

<표 2> 각 실험 영상의 평균 탐색점수

	FS	NTSS	FSS	DS	MVFAST	PMVFAST	제안기법
Akiyo	961	17.0	17.0	13.0	5.05	2.9	0.76
Foreman	961	19.3	18.6	15.4	8.9	7.8	4.53
M & D	961	17.3	17.1	13.2	5.6	4.7	0.88
Stefan	961	20.0	18.9	16.2	10.8	8.9	4.69
Table	961	19.7	18.7	15.5	10.4	8.5	5.41

5. 결론

본 논문에서는 현재 블록과 시간적/공간적 상관성을 가진 블록들의 움직임 벡터들을 이용하여 현재 블록 유형을 분류

하고 현재 블록의 움직임을 예측하였다. 현재 블록 유형에 따라 움직임 추정이 불필요한 블록에서 움직임 벡터 추정을 생략함으로써 움직임 추정 속도를 향상시켰고 예측된 움직임을 크기에 따라 현재 블록의 탐색 시작점과 탐색 패딩을 적응적으로 선택하여 움직임 추정을 수행하는 기법을 제안한다.

실험을 통하여 알 수 있듯이, 제안한 기법은 움직임이 있는 영상인 경우 보다 빠르고 정확한 움직임 예측을 함으로써 움직임 추정 속도 면이나 화질 면에 있어서 성능 향상을 보였다. 움직임 벡터 필드 적응적 탐색 기법과 예측된 움직임 벡터 필드 적응적 탐색 기법과 비교해서 볼 때, 제안한 기법은 움직임 추정 속도 면에서 약 1.5 ~ 7배의 속도 향상을 가져왔으며, 움직임이 있는 영상의 화질 면에 있어서 약 0.02 ~ 0.2 (dB)정도의 성능 향상을 보였다.

참 고 문 헌

- [1] T.Koga, K.Iinuma, A.Hirano, Y.Ishiguro, "Motion compensated interframe coding for video conference", Proc. NTC81, pp.G5.3.1-5.3.5, Nov. 1981.
- [2] R.Li, B.Zeng and M.L.Liou, "A New Three Step Search Algorithm for Block Motion Estimation", IEEE Trans. on Circuits and System for Video Technology Vol. 4 No. 4 pp. 438-441, Aug. 1994.
- [3] J.R.Jain and A.K.Jain, "Displacement measurement and its application in Interframe image Coding", IEEE Trans. on Communications, Vol. 29 No. 12 pp.1779-1808, Dec. 1981.
- [4] L.M.Po and W.C.Ma, "A Novel Four Search Algorithm for Block Motion Estimation". IEEE Trans. on Circuit and Systems for Video Technology Vol. 6. pp.313-317, June 1996.
- [5] J.Y. Tham, S.Ranganath and A.A.Kassim, "A Novel Unrestricted Center-Biased Diamond Search Algorithm for Block Motion Estimation", IEEE Transactions on Circuits and Systems for Video Technology. Vol. 8 pp.369-377, Aug. 1998.
- [6] S.Zhu and K.K.Ma "A New Diamond Search Algorithm for Fast Block Matching Motion", IEEE Transaction on Image Processing Vol. 9 No. 2 pp287-290, Feb. 2000.
- [7] Yuk Ying Chung ,Neil W.Bergmann "Fast Search Block Matching Motion Estimation Algorithm using FPGA", Visual Communications and Image Processing 2000. Proc.SPIE Vol. 4067 pp913-921.
- [8] Danian Gong, Yun He, "Fast Motion Estimation Algorithm using Horizontal and Multi-grid Search Strategy", Picture Coding Symposium 2001. pp362- 365
- [9] Prabhudev Irappa Howur, K. K. Ma, "Report on Performance of Fast Motion Estimation using Motion Vector Field Adaptive Search Technique", ISO/IEC/JCTI/SC29/WG11 M5453, Dec. 1999.
- [10] A. M. Tourapis, O. C. Au, M. L. Liou, "Fast Block Matching Motion Estimation using Predictive Motion Vector Field Adaptive Search Technique", ISO/IEC/JCTI/SC29/WG11 MPEG2000/M5866, Mar. 2000.



윤 효 순

e-mail : estheryoon@hotmail.com
 1993년 호남대학교 전산통계학과 졸업(학사)
 2002년 전남대학교 대학원 전산학과 졸업(석사)
 2005년 전남대학교 대학원 전산학과 졸업(박사)
 2006~현재 전남대학교 전산학과 연구원
 관심분야: 멀티미디어 통신, 영상 압축 및

복원, 에러 은닉



유 재 명

e-Mail : jmyoo@oracle.chonnam.ac.kr
 1992년 전남대학교 응용화학 공학부 졸업(학사)
 2003년 전남대학교 대학원 전산학과 졸업(석사)
 2006 ~ 현재 전남대학교 대학원 전산학과 박사과정

관심분야: 멀티미디어 통신, 영상 처리



박 미 선

e-mail : sunnydix@hanmail.net
 2005년 전남대학교 전산학과 졸업(학사)
 2005 ~ 현재 전남대학교 대학원 전산학과 석사과정

관심분야: 멀티미디어 통신, 영상 처리



김 미 영

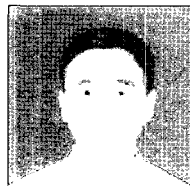
1983년 전남대학교 전산통계학과 이학사
 1985년 이화여자대학교 대학원 이학석사
 1997년 전남대학교 전산통계학과 이학박사
 1986~1997년 목포과학대학 전산계산과 부교수

1998~현재 전남도립남도대학 컴퓨터정보

통신과 교수

관심분야: 멀티미디어

통신, 화상처리, 정보검색, CAD/VLSI



Nguyen Dinh Toan

e-mail : toanhoian@gmail.com
 2005년 University of natural sciences in Ho Chi Minh (학사)
 2006~전남대학교 대학원 전산학과 석사과정
 관심분야: 멀티미디어 통신, 영상 처리



이 귀 상

e-mail : gslee@chonnam.ac.kr
 1980년 서울대 공대 전기공학과 졸업(학사)
 1982년 서울대 대학원 전자계산기공학과 석사
 1982년 금성통신 연구소
 1991년 Pennsylvania 주립대학 전산학과 박사
 1984년~현재 전남대 전산학과 교수

관심분야: 멀티미디어통신, 영상처리 및 복원, 논리합성,

VLSI/CAD