

개선된 크로스-육각 패턴을 이용한 고속 블록 정합 움직임 추정 알고리즘

남 현 우*

A Fast Block Matching Motion Estimation Algorithm by using the Enhanced Cross-Hexagonal Search Pattern

Hyeon-Woo Nam*

요 약

동영상 프레임의 블록 정합에 있어서 현재 프레임의 블록들 사이에는 공간적 상관성이 존재한다. 본 논문에서는 움직임 벡터의 중심 지향적 특성과 동영상 내의 공간적 상관성을 이용한 개선된 고속 블록 정합 움직임 추정 알고리즘을 제안한다. 제안하는 알고리즘은 현재 프레임의 인접한 매크로 블록들에서 예측된 움직임 벡터와 크로스-육각 탐색 패턴을 이용하여 정확한 움직임 벡터를 결정한다. 실험 결과를 통해 제안하는 알고리즘이 예측 탐색 알고리즘 (NNS)과 고속 블록 정합 알고리즘 (CHS)에 비해 탐색 속도와 화질 면에서 우수한 성능을 나타내었다. 제안하는 알고리즘을 사용하였을 때, 탐색 속도 측면에서는 약 0.1~38% 정도의 향상을 보였고, 화질 측면에서도 PSNR (Peak Signal Noise Ratio) 값이 약 0.05~2.5 dB 정도 향상되었다.

Abstract

There is the spatial correlation of the video sequence between the motion vector of current blocks. In this paper, we propose the enhanced fast block matching algorithm using the spatial correlation of the video sequence and the center-biased property of motion vectors. The proposed algorithm determines an exact motion vector using the predicted motion vector from the adjacent macro blocks of the current frame and the Cross-Hexagonal search pattern. From the experimental results, we can see that our proposed algorithm outperforms both the prediction search algorithm (NNS) and the fast block matching algorithm (CHS) in terms of the search speed and the coded video's quality. Using our algorithm, we can improve the search speed by up to 0.1~38%, and also diminish the PSNR (Peak Signal Noise Ratio) by at most 0.05~2.5 dB, thereby improving the video quality.

▶ Keyword : 고속 블록 정합 알고리즘(Fast Block Matching Algorithm), 탐색 패턴(Search Pattern), 움직임 추정(Motion Estimation)

• 제1저자 : 남현우

• 접수일 : 2006.08.09, 심사일 : 2006.08.22, 심사완료일 : 2006.09.20

* 동덕여자대학교 컴퓨터학과 전임강사

※ 이 논문은 2005년도 동덕여자대학교 교내 학술연구비 지원에 의해 수행된 것임.

1. 서론

대용량, 고화질 동영상의 응용분야가 확대되면서 동영상 압축을 통한 저장 및 전송에 대한 효율을 높일 수 있는 연구가 활발히 진행되고 있다. 동영상 압축의 성능을 좌우하는 핵심 요소는 연속적인 프레임들 사이에 존재하는 중복성을 움직임 추정(ME: Motion Estimation)을 통해 배제하는 것이다.

일반적으로 움직임 추정에는 계산 복잡도 및 하드웨어 구현이 용이한 블록 단위의 움직임 추정 알고리즘들이 널리 사용된다. 이 방법들은 대부분 동일한 블록 내의 픽셀들은 동일한 움직임을 가지므로 영상의 한 프레임을 동일한 크기의 블록들로 나누고 이들의 각 블록들에 대하여 이전 프레임의 탐색 영역 내에서 정합 오차가 가장 작은 블록간의 위치 차이를 구해서 움직임 벡터(motion vector)로 추정한다.

이와 같이 블록 단위의 대표적인 움직임 추정 알고리즘이 MPEG 이나 H.26x 계열의 동영상 압축 표준에서 사용되고 있는 전역 탐색(FS: Full Search) 알고리즘이다 [1]. 전역 탐색 알고리즘은 탐색 과정이 간단하고 예측 효율과 추정의 정확도를 고려할 때 전체적으로 좋은 특성을 가지며 하드웨어 구현이 용이하고 또한 탐색 영역의 내부 전체를 탐색하면서 가능한 모든 블록들에 대한 정합을 수행하기 때문에 정합 오차가 가장 적은 움직임 벡터를 찾을 수 있지만 많은 계산을 필요하게 되는 단점이 있다. 이러한 전역 탐색 알고리즘의 단점을 극복하기 위해 속도가 개선된 3단계 탐색(TSS: Three Step Search)[2], 4단계 탐색(FSS: Four Step Search)[3], 다이아몬드패턴 탐색(DS: Diamond Search)[4, 5, 6], 육각패턴 탐색(HEXBS: Hexagon-based Search)[7, 8], 크로스-다이아몬드패턴 탐색(CDS: Cross-Diamond Search)[9], 크로스-육각패턴 탐색(CHS: Cross-Hexagonal Search)[10] 등의 다양한 고속 블록 정합 알고리즘(FBMA: Fast Block Matching Algorithm)이 개발되었다.

이들 속도 개선 알고리즘은 블록 정합을 위해 각 탐색 단계에서 정합 오차를 검사하는 블록 내의 탐색점들을 의미하며 탐색 패턴을 이용하여 탐색 영역 내에서 탐색할 위치의 포인터 개수를 감소시켜 계산량의 감소를 유도하게 된다. 탐색 패턴의 탐색점들 중에서 최소 정합 오차를 가지는 위치를 중심으로 다음 단계의 움직임 탐색이 수행되기 때문에 탐색 패턴은 그 모양과 크기에 따라

고속 블록 정합 움직임 추정에서 탐색의 속도와 성능을 좌우하는 중요한 요소가 될 수 있다.

속도를 개선시키기 위한 고속 블록 정합 알고리즘은 계산량을 줄이기 위해 탐색 영역에 포함되는 특정한 패턴에 정의된 일부의 점들만을 움직임 벡터 추정에 사용하기 때문에 국부적인 탐색을 하게 되거나, 해당 블록에 대한 움직임 정보를 가지고 있지 않기 때문에 항상 탐색 영역의 원점에서부터 움직임을 추정하는 과정에서 일부 탐색점들을 블록 정합 대상에서 제외시키거나 부정확한 방향으로 움직임을 추정함으로써 복원된 영상의 화질을 저하시키는 단점이 있다.

이러한 고속 블록 정합 알고리즘의 단점을 개선하기 위해 같은 프레임 내의 인접한 이전 블록들의 움직임 벡터들의 정보를 이용하는 예측 탐색 알고리즘(PSA: Prediction Search Algorithm)들이 제안되었다[11, 12, 13].

이 이외에도 다중해상도를 고려한 고속 블록 정합 알고리즘들이 연구되고 있다[14, 15].

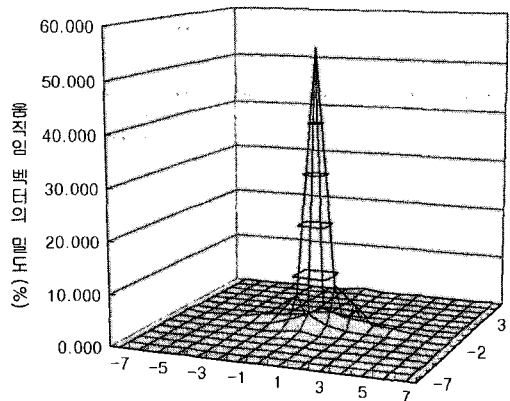


그림 1. 전체 실험동영상의 움직임 벡터 밀도

Fig. 1 Motion vector density of whole experimental videos

일반적으로 움직임이 있는 동영상의 경우라도 프레임 간의 짧은 시간 속에서는 움직임이 미세하거나 카메라의 움직임이나 피사체의 움직임이 동일 방향성을 가지는 경우가 대부분이므로 그림 1과 같이 확률적으로 움직임 벡터가 중심 지향적이고, 현재 매크로 블록(MB: Macro Block)의 움직임 벡터는 인접한 매크로 블록들과 높은 상관관계를 가지게 된다.

본 논문에서는 현재 프레임 내의 인접한 블록의 움직임 벡터 정보를 이용하여 현재 블록의 초기 탐색점들 예

측하고, 그림 1과 같이 탐색점 중심 부근에 확률적으로 많이 분포하는 움직임 벡터를 크로스 패턴을 이용하여 먼저 탐색한 다음 나머지 큰 움직임을 가지고 있는 움직임 벡터를 육각 패턴을 이용해서 찾는 개선된 고속 블록 움직임 추정 알고리즘을 제안하고 실험을 통해 제안된 알고리즘의 우수성을 검증한다.

II. 기존의 움직임 추정 알고리즘

육각패턴 탐색(HEXBS) 알고리즘은 그림 2(a)와 같이 중심점과 수평방향의 거리가 2인 두 점과 중심점으로부터 거리가 $\sqrt{3}$ 인 4개의 점으로 이루어진 큰 육각패턴(LHSP: large hexagon-based search pattern)과 그림 2(b)와 같은 중심점으로부터 거리가 1인 4개의 탐색점을 갖는 작은 육각패턴(SHSP: small hexagon-based search pattern)을 이용함으로써 다이아몬드패턴 탐색 알고리즘보다 현저한 속도 증가를 나타내는 것으로 평가된 탐색 방법이다[7, 8].

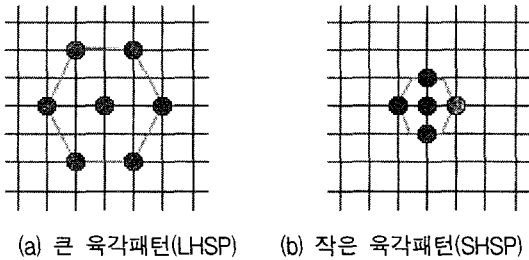


그림 2. 육각패턴 탐색 알고리즘의 탐색 패턴
Fig. 2 Search patterns used in the HEXBS

육각패턴 탐색(HEXBS) 알고리즘은 탐색 영역 내에 미리 정의된 탐색 블록의 중심점 (0,0)을 중심점으로 하는 7개의 탐색점을 가지는 큰 육각패턴을 구성하여 최소 BDM 점을 계산한다. 최소 BDM 점이 큰 육각패턴의 중심점이 될 때까지 이전 단계에서 구해진 최소 BDM 점을 중심으로 하는 큰 육각패턴을 구성하여 탐색을 반복한다. 큰 육각패턴의 중심점이 최소 BDM 점일 때, 중심점 이웃의 4개의 점을 포함하는 작은 육각패턴을 구성하여 최소 BDM 점을 구하고 이 점을 움직임 벡터로 결정한다. 육각패턴을 이용한 탐색 경로는 그림 3과 같다.[7, 8]

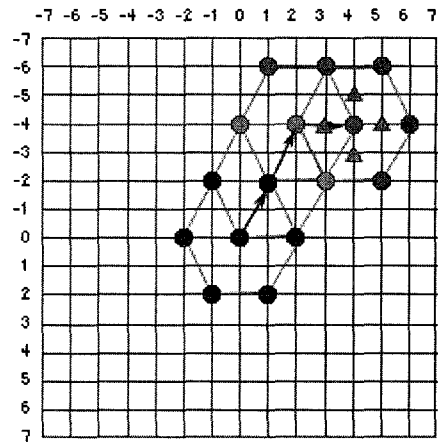


그림 3. 육각패턴 탐색 알고리즘의 탐색 경로
Fig. 3 Search path of the HEXBS

크로스-육각패턴 탐색(CHS) 알고리즘은 그림 1에서 볼 수 있듯이 움직임 벡터의 대부분이 탐색 영역의 중심에 분포하는 특성을 고려하여 그림 4(a)와 같이 중심점을 포함하는 초기 크로스 패턴을 이용한 탐색을 수행한 후 중심점이 최소 BDM 이 아닐 경우 비교적 탐색효율이 높은 것으로 평가된 그림 4(b)와 그림 4(c)의 육각패턴을 이용한 탐색을 수행하는 방법이다[10].

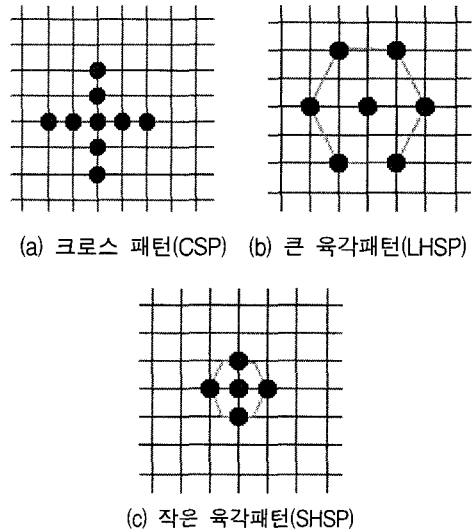


그림 4. 크로스-육각패턴 탐색 알고리즘의 탐색 패턴
Fig. 4 Search patterns used in the CHS

크로스-육각패턴 탐색(CHS) 알고리즘은 그림 4 (a)와

같이 초기 크로스 패턴의 9개의 탐색점을 계산하여 중심점이 최소 BDM 이면 중심점을 움직임 벡터로 결정하고 탐색을 중단한다(첫 번째 종료 조건). 그렇지 않으면 크로스 패턴의 중심점으로부터 각각 $(-1, -1)$, $(1, -1)$, $(1, 1)$, $(-1, 1)$ 에 위치하는 4개의 후보 탐색점 중 크로스 패턴 탐색에서 구해진 최소 BDM 과 가까운 두 개의 탐색점을 추가하여 작은 육각패턴 탐색을 수행한다. 이때 중심점이 최소 BDM 이 되면 작은 육각패턴의 중심점을 움직임 벡터로 결정하고 탐색을 중단한다(두 번째 종료 조건).[10] 두 가지의 종료 조건을 만족하지 못할 때에는 그림 4(b)와 같이 큰 육각패턴을 적용하여 탐색 패턴의 중심점이 최소 BDM이 될 때까지 큰 육각형 형태의 주위 7개의 탐색점을 계산한다. 계산된 최소 BDM 점이 중심점일 때 작은 육각패턴을 구성하고 최종적으로 BDM 계산을 통해 정합 블록의 움직임 벡터를 구한다. 크로스-육각패턴을 이용한 탐색 경로는 그림 5와 같다[10].

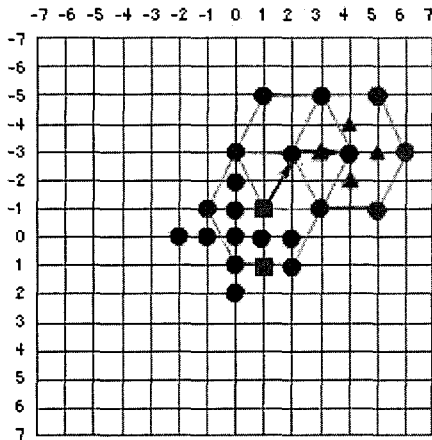


그림 5. 크로스-육각패턴 탐색 알고리즘의 탐색 경로
Fig. 5 Search Path of the CHS

III. 제안하는 움직임 추정 알고리즘

예측 알고리즘에서 예측된 움직임 벡터는 일반적으로 현재 블록과 같은 위치에 있는 이전 시점의 블록이나 현재 프레임에서 추출된 인접 블록들의 움직임 벡터를 이용하여 정의되며, 이렇게 정의된 움직임 벡터를 결정하여 블록 정합을 위한 탐색 패턴으로 TSS 또는 DS 알고리즘을 적절히 적용하고 있다[12, 13].

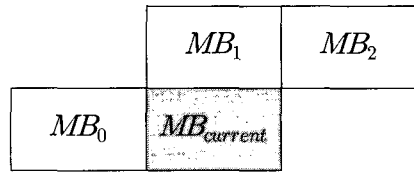


그림 6. 참조되는 인접 매크로 블록
Fig. 6 Referenced Adjacent MB

현재 프레임의 블록들간의 공간적인 상관관계를 반영하여 예측하는 움직임 벡터(SPMV: Spatial Predicted Motion Vector)는 공간적으로 인접한 매크로 블록들은 비슷한 속도로 거의 같은 방향으로 움직인다는 점을 고려하여 그림 6과 같이 왼쪽 매크로 블록(MB_0), 위쪽 매크로 블록(MB_1) 그리고 위쪽의 오른쪽 매크로 블록(MB_2)을 참조한다[10]. 인접 매크로블록 $MB_i (i \in 0, 1, 2)$ 의 움직임 벡터를 $MV_i (i \in 0, 1, 2)$ 라고 한다면, 현재 매크로 블록 $MB_{current}$ 의 예측 움직임 벡터 $MV_{current}$ 는 식 (1)과 같이 이들 인접 매크로 블록 각각의 움직임 벡터 MV_0, MV_1, MV_2 들의 중간 값으로 한다.

$$MV_{current} = \text{median}(MV_0, MV_1, MV_2) \dots\dots\dots (1)$$

이러한 예측 탐색 기법은 공간적인 상관관계를 적용하여 속도뿐만 아니라 화질 측면에서도 좋은 결과를 보인다고 알려져 있다[11]. 그러나 예측된 움직임 벡터를 탐색 원점으로 이용하여 빠르게 탐색을 수행함으로써 속도 측면에서는 좋은 결과를 보일 수 있으나, 움직임 벡터의 상관성이 떨어질 때는 잘못된 움직임을 추정하여 영상의 화질이 크게 저하될 수 있다.

본 논문에서는 기존의 고속 블록 정합 알고리즘이 탐색을 수행함으로써 블록 내에 공존하는 최소 정합 오차를 가지는 탐색점을 제외시켜 잘못된 추정을 하는 문제점을 개선하기 위해 인접 매크로 블록의 움직임 벡터 정보에 의해 예측된 탐색 중심점을 구한 뒤 크로스-육각패턴을 적용하여 가장 작은 정합 오차를 갖는 점을 탐색 원점으로 하여 탐색을 수행하는 알고리즘을 제안한다.

이때 블록 정합 오차를 결정하는 평가 함수로는 식 (2)와 같이 절대값 오차의 합(SAD: Sum of Absolute Difference)을 사용한다[1, 4].

$$SAD(i, j) = \sum_{k=1}^N \sum_{l=1}^N |I_t(k, l) - I_{t-1}(k + i, l + j)| \dots\dots\dots (2)$$

이 때, 식 (2)에서 N 은 영상의 가로와 세로의 각각의 크기이며, $I_t(k,l)$ 은 원영상의 화면을 나타내고, $I_{t-1}(k+i,l+j)$ 은 움직임 추정 화면을 나타낸다.

예측을 통해 구해진 탐색 중심점을 기준으로 우선 크로스 패턴을 이용하여 탐색 중심점 주위에 분포하는 움직임 벡터를 탐색하고 큰 움직임을 가진 움직임 벡터를 빨리 찾아내기 위한 육각패턴을 사용하는 개선된 크로스-육각패턴 탐색을 통해 움직임 벡터의 빠른 탐색과 화질 개선을 도모한다.

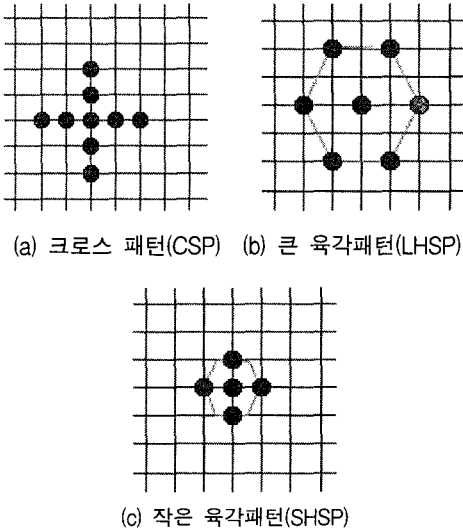


그림 7. 개선된 크로스-육각패턴 탐색 알고리즘의 탐색 패턴
Fig. 7 Search patterns used in the Enhanced CHS

다음은 제안하는 개선된 크로스-육각패턴을 이용하여 움직임 벡터를 추정하는 알고리즘이며, 그림 8은 제안하는 알고리즘의 탐색 흐름도를 나타내었다.

- 1단계: 현재 매크로 블록의 인접 매크로 블록의 평균으로 예측된 움직임 벡터 $MV_{current}$ 를 구한다.
- 2단계: 예측된 움직임 벡터 $MV_{current}$ 를 탐색 중심점으로 그림 7(a)와 같은 크로스 패턴 각 점(9개)의 BDM을 계산하여 최소 BDM 점이 중심점이면 이 점을 움직임 벡터로 결정하고 탐색을 종료한다.
- 3단계: 9개 점 중에서 최소 BDM 을 가지는 점 P 가 중심점이 아니면 $(-1,-1), (1,-1), (1,1), (-1,1)$ 좌표의 4개의 중에서 점 P 에 가까운 2개의 점이 후

보 점으로 추가된다. 여기서, 최소 BDM 점이 $(-1,0), (1,0), (0,-1), (1,0)$ 중의 하나이면 그 점을 움직임 벡터로 결정하고 탐색을 종료한다.

- 4단계: 이전 단계에서 계산된 최소 BDM 점을 중심점으로 하는 그림 7(b)와 같은 큰 육각패턴을 구성한다. 큰 육각패턴의 중심점에서 최소 BDM 이 발견될 때까지 4단계를 반복한다. 큰 육각패턴의 중심점에서 최소 BDM 점이 발견되면 5단계로 진행한다.
- 5단계: 이전 단계에서 구해진 최소 BDM 점을 중심으로 하는 그림 7의 (c)와 같은 작은 육각패턴을 구성한다. 작은 육각패턴에서는 구해진 최소 BDM 점을 움직임 벡터로 결정하고 탐색을 종료한다.

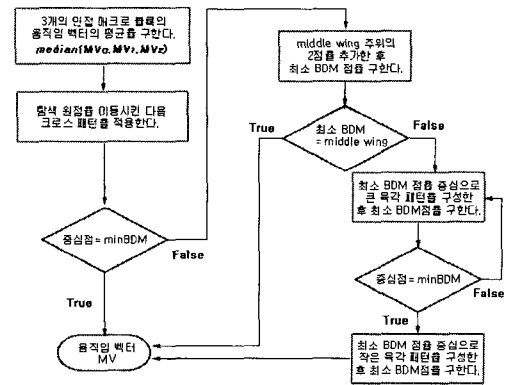


그림 8. 제안하는 알고리즘의 탐색 흐름도
Fig. 8 Search Flow of the proposed Algorithm

IV. 실험 결과

제안하는 개선된 크로스-육각 패턴 알고리즘의 성능을 평가하기 위하여 표 1과 같이 각각 다른 해상도의 실험 동영상 상들을 대상으로 각각 80 프레임씩을 이용하여 실험하였다.

표 1. 실험 동영상
Table. 1 Experimental videos

동영상 형식	해상도	실험 적용 동영상
CIF	352X288	Akiyo, Foreman, Coastguard, Stefan
SIF	352X240	Susie, Flower Garden, Mobile, Popp
QCIF	176X144	Akiyo, Miss America, Coastguard, Stefan

성능 비교에 적용해 본 움직임 추정 기법은 FS, DS, HEXBS, CHS 와 인접 블록을 이용한 예측 탐색 알고리즘인 NNS(Nearest Neighbor Search), 그리고 제안한 움직임 추정 알고리즘을 사용하였다. 예측 탐색 알고리즘인 NNS에는 일반적으로 적용하는 DS 알고리즘을 적용하여 실험하였다. 그리고 움직임 추정에 사용된 매크로 블록의 크기는 16×16 픽셀이며, 탐색 영역의 변위는 ±7을 적용하여 Pentium IV 3.0GHz와 512MB 메모리가 장착된 컴퓨터상에서 실험을 수행하였다.

블록 정합의 정도를 평가하기 위해 대표적인 정합 기준인 평가 함수(cost function)로 영상 화질의 품질을 평가하기 위한 식 (3)의 평균 절대값 오차(MAD: Mean Absolute Difference) 를 이용하였다[4, 8]. 또한 제안하는 기법의 성능 향상을 측정하기 위해 블록 당 탐색점의 개수를 기존 방법들과 비교하였다.

$$MAD(i, j) = \left(\frac{1}{N^2} \right) \sum_{k=1}^N \sum_{l=1}^N |I_i(k, l) - I_{i-1}(k + i, l + j)| \dots\dots (3)$$

그리고 화질의 평가를 위해 평균 제곱 오차(MSE : Mean Square Error)를 이용한 PSNR(Peak Signal to Noise Ratio) 을 구하기 위해 다음 식 (4)를 적용한다.

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \dots\dots\dots (4)$$

$$\{MSE = \left(\frac{1}{M \times N} \right) \sum_{k=1}^M \sum_{l=1}^N [I_i(k, l) - I_{i-1}(k, l)]^2\}$$

제안한 알고리즘의 성능 평가 결과 중 해상도가 서로 다른 실험 동영상 2개씩에 대해 그 실험 결과를 표 2와 표 3으로 나타내었다. 표에 나타낸 실험 동영상은 비교적 움직임이 작은 동영상과 비교적 큰 움직임을 가진 동영상을 각각 선택하였다. 표 2에서는 실험 동영상에서의 각 블록에 대한 평균 탐색 회수를 PN(search Point Number) 과 FS 의 탐색 회수를 기준하여 각 움직임 추정 기법의 상대적인 속도 향상을 의미하는 SR(Speedup Ratio) 로 비교 결과를 나타내었다. 또한 표 3에서는 기존의 움직임 추정 기법과 제안한 알고리즘의 각 블록에 대한 MAD 와 평균 PSNR 값을 비교한 결과를 나타내었다.

실험 결과를 볼 때, 제안한 알고리즘은 전역 탐색의 속도 문제를 개선한 고속 블록 정합 알고리즘이므로 전역 탐색과의 비교는 무의미하다. 실험 결과에 대한 성능 비교는 제안한 알고리즘의 근간이 되는 NNS와 CHS 와

의 비교를 통해 분석해 보기로 하겠다.

표 2에서 나타낸 바와 같이 제안한 알고리즘과 비교 대상인 두 알고리즘과의 탐색 속도 측면의 실험 결과를 보면, 인접한 매크로 블록의 예측 정보를 이용하는 NNS 에 대해서는 평균 탐색점의 수가 약 18~28% 정도 감소하여 약 22~38% 정도의 탐색 속도가 향상되었다. 제안하는 알고리즘의 탐색 패턴 모델인 CHS 에 대해서는 평균 탐색점의 수가 약 0.1~13% 정도 감소하여 약 0.1~14.5% 정도의 탐색 속도가 향상되었다. 두 알고리즘과의 탐색 속도 결과 비교에 의해 제안하는 알고리즘이 평균 탐색점 수의 감소로 인해 탐색 속도의 향상되었음을 볼 수 있었다.

표 3에서 나타낸 제안한 알고리즘과 비교 대상인 두 알고리즘과의 화질 측면의 실험 결과를 보면, 인접한 매크로 블록의 예측 정보를 이용하는 NNS 에 대해서는 MAD 가 일부의 동영상에서 약 0.9~8.6% 정도 증가하여 약 0.1~2.5% 정도의 화질 저하가 발생하였다. 제안하는 알고리즘의 탐색 패턴 모델인 CHS 에 대해서는 MAD 가 약 0.2~39% 정도 감소하여 약 0.1~11.3% 정도의 화질 향상을 보였다. 두 알고리즘과의 화질에 대한 결과 비교에서도 제안하는 알고리즘이 화질의 향상되었음을 볼 수 있었다.

실험에 사용한 동영상별로 실험 결과를 비교해 보면, 움직임이 작은 Akiyo 나 Susie 영상은 인접한 매크로 블록의 예측 정보를 이용하는 NNS 에 대해서는 평균 탐색점의 수가 약 18~28% 정도 감소하여 약 22~38% 정도의 탐색 속도가 향상되었고, 제안하는 알고리즘의 탐색 패턴 모델인 CHS 에 대해서도 평균 탐색점의 수가 약 0.1~5.2% 정도 감소하여 약 0.1~5.5% 정도의 탐색 속도가 향상되었다. 화질 측면의 실험 결과에서는 CHS 에 대해서는 MAD 가 약 0.2~15% 정도 감소하여 약 0.1~11.3% 정도의 화질 향상을 보였지만 NNS 에 대한 실험 결과 중 Susie 영상에서 MAD 가 8.6% 정도 증가하여 약 2.5% 정도의 화질 저하가 발생하였다.

움직임이 큰 Stefan 이나 Flower Garden 영상은 NNS 에 대해서는 평균 탐색점의 수가 약 21.4~25.4% 정도 감소하여 약 27.2~34.1% 정도의 탐색 속도가 향상되었고, CHS 에 대해서는 평균 탐색점의 수가 약 6.6~12.6% 정도 감소하여 약 7.1~14.5% 정도의 탐색 속도가 향상되었다. 제안하는 알고리즘의 탐색 패턴 모델인 CHS 에 대해서는 MAD가 약 15.4~39% 정도 감소하여 약 2.6~11.3% 정도의 화질 향상을 보였지만 NNS 에 대한 실험 결과에서는 미미한 수준이지만 MAD 가 0.9~3.7% 정도 증가하여 약 0.1~1% 정도의 화질 저하가 발생하였다.

표 2. 실험 결과(탐색 횟수 및 속도 향상율)
Table 2 Experimental Result(search Point Number and Speedup Ratio)

Sequence Method	CIF				SIF				QCIF			
	Akiyo		Stefan		Susie		Flower Garden		Akiyo		Stefan	
	PN	SR	PN	SR	PN	SR	PN	SR	PN	SR	PN	SR
FS	204.28	1.000	204.28	1.000	202.05	1.000	202.05	1.000	184.56	1.000	184.56	1.000
DS	12.29	16.621	16.77	12.180	15.38	13.137	16.81	12.018	11.44	16.139	14.27	12.932
HEXABS	10.35	19.740	12.98	15.737	11.92	16.949	13.15	15.365	9.65	19.129	11.36	16.241
CHS	8.89	22.985	10.86	18.809	11.86	17.037	11.98	16.868	8.28	22.290	9.88	18.683
NNS(DS)	12.25	16.672	13.39	15.260	13.72	14.726	13.31	15.181	11.43	16.143	12.37	14.914
Proposed	8.83	23.139	10.13	20.165	11.24	17.969	10.47	19.305	8.27	22.309	9.23	20.005

표 3. 실험 결과(MAD 및 PSNR)
Table 3 Experimental Result(MAD and PSNR)

Sequence Method	CIF				SIF				QCIF			
	Akiyo		Stefan		Susie		Flower Garden		Akiyo		Stefan	
	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR	MAD	PSNR
FS	0.603	42.780	6.715	25.693	2.665	34.722	8.520	23.994	0.488	43.851	6.794	25.716
DS	0.604	42.760	7.717	24.655	2.744	34.420	8.768	23.806	0.488	43.851	6.896	25.622
HEXABS	0.617	42.498	7.929	24.501	2.938	33.754	9.509	23.320	0.491	43.813	7.018	25.531
CHS	0.605	42.682	12.033	22.111	3.495	32.424	11.401	21.567	0.488	43.851	8.106	24.841
NNS(DS)	0.606	42.708	7.077	24.861	2.736	34.165	8.654	23.782	0.488	43.851	6.799	25.511
Proposed	0.604	42.736	7.335	24.607	2.972	33.301	8.780	23.683	0.488	43.851	6.860	25.490

제안하는 개선된 크로스-육각 패턴 탐색 알고리즘과 기존의 두 알고리즘과의 전체적인 성능평가를 하자면 탐색 속도의 측면에서는 제안하는 알고리즘이 평균 탐색점 수의 감소로 인해 비교 대상 알고리즘들에 비해 우수한 탐색 속도의 향상되었음을 볼 수 있었다. 화질의 측면에서도 제안하는 알고리즘의 탐색 패턴 모델인 CHS 에 비해서 화질의 향상을 나타내었고, 인접한 매크로 블록의 예측 정보를 이용하는 NNS 에 대해서는 일부 동영상에서 미미한 수준의 화질 저하가 있었지만 탐색 속도 향상 대비 화질 저하를 평가한다면 우수하다고 평가할 수 있다.

V. 결론

본 논문에서는 기존의 예측 탐색 알고리즘이 예측된 움직임 벡터로 탐색 원점을 이동시켜 탐색을 수행함으로써 화질 저하를 초래하는 문제점과 기존의 고속 블록 정합 알고리즘이 탐색을 수행함으로써 블록 내에 공존하는

최소 정합 오차를 가지는 탐색점을 제외시켜 잘못된 추정을 하는 문제점을 개선하기 위해 인접 매크로 블록의 움직임 벡터 정보에 의해 예측된 탐색 중심점을 구한 뒤 크로스-육각 패턴을 적용하여 가장 작은 정합 오차를 갖는 점을 탐색 원점으로 하여 탐색을 수행하는 알고리즘을 제안하였다.

실험의 결과를 보면 제안한 알고리즘이 기존의 예측 탐색 알고리즘이나 고속 블록 정합 알고리즘에 비해서 탐색 속도의 향상에 있어서 우수한 성능을 보였으며, 화질 측면에서도 화질이 기존에 비해 향상되었고 일부 영상에서도 탐색 속도 대비 화질 저하를 고려한다면 우수한 성능을 나타내었다.

본 논문에서 제안한 예측 탐색을 적용한 고속 블록 정합 움직임 추정 알고리즘을 사용하여 움직임 추정을 한다면 보다 빠르게 움직임 벡터를 찾을 수 있을 것이며, 고속 블록 정합 방법에 적합한 탐색 패턴을 도출하고, 탐색 패턴을 적용적으로 변화시킨다면 다른 고속 블록 정

합 방법들보다 탐색점의 수를 감소시키고, 우수한 화질을 얻을 수 있을 것으로 기대된다.

참고문헌

- [1] Information Technology-Generic Coding of Moving Pictures and Associated Audio Information: Video, ISO/IEC 13818-2(MPEG-2 Video), 2000.
- [2] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion-compensated Interframe Coding for Video Conferencing", In Proc. of the International Telecommunications Conference, New Orleans, LA, pp. G5.3.1-G5.3.5, Nov. 1981.
- [3] L. M. Po, and W. C. Ma, "A Novel Four-Step Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 6, No. 3, pp. 313-317, June 1996.
- [4] Shan Zhu, and Kai-Kuang Ma, "A New Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Image Processing, Vol. 9, No. 2, pp. 287-290, Feb. 2000.
- [5] T. Sappasitwong, S. Aramvith, S. Jitapunkul, A. Tamtrakarn, P. Kitti-punyangam, and H. Kortrakulkij, "Adaptive asymmetric diamond search algorithm for block-based motion estimation", In Proc. of the International Conference on Digital Signal Processing, pp. 563-566, July 2002.
- [6] 홍성용, "구속조건을 적용한 다이아몬드 탐색 알고리즘에 의한 고속블록정합움직임 추정", 한국컴퓨터정보학회 논문지, Vol. 8, No. 4, pp. 13-20, Dec. 2003.
- [7] Ce Zhu, Xiao Lin, Lap-Pui Chau, Keng-Pang Lim, Hock-Ann Ang, and Choo-Yin Ong, "A novel hexagon-based search algorithm for fast block motion estimation", In Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 1593-1596, May 2001.
- [8] Ce Zhu, Xiao Lin, and Lap-Pui Chau, "Hexagon-Based Search Pattern for Fast Block Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 5, pp. 349-355, May 2002.
- [9] C. H. Cheung, and L. M. Po, "A Novel Cross-Diamond Search Algorithm for Fast Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 12, pp. 1168-1177, Dec. 2002.
- [10] 박인영, 남현우, 위영철, 김하진, "십자와 육각패턴을 이용한 고속 블록 정합 동작 예측기법", 한국정보처리학회 논문지 B, Vol. 10, No. 7, pp. 811-814, Dec. 2003.
- [11] Oscar T.-C. Chen, "Motion Estimation Using a One-Dimensional Gradient Descent Search", IEEE Transactions on Circuits & System for Video Technology, Vol. 10, No. 4, pp. 608-616, June 2000.
- [12] A. M. Tourapis, O. C. Au, and M. L. Liou, "Highly efficient predictive zonal algorithms for fast block-matching motion estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 12, No. 10, pp. 934-947, Oct. 2002.
- [13] F. Moschetti, M. Kunt, and E. Debes, "A Statistical Block-Matching Motion Estimation", IEEE Transactions on Circuits & System for Video Technology, Vol. 13, No. 4, pp. 417-431, Apr. 2003.
- [14] Tae-Gyoung Ahn, Yong Ho Moon, and Jae Ho Kim, "Fast full-search motion estimation based on multilevel successive elimination algorithm", IEEE Transactions on Circuits and Systems for Video Technology, VOL. 14, NO. 11, pp. 1265-1269, Nov. 2004.
- [15] G. Calvagno, F. Fantozzi, R. Rinaldo, and A. Viareggio, "Model-based global and local motion estimation for videoconference sequences", IEEE Transactions on Circuits and Systems for Video Technology, VOL. 14, NO. 9, pp. 1156-1161, Sep. 2004.

저 자 소개



남현우

1993년 아주대학교 공과대학
전자계산학과 (공학사)

1996년 아주대학교 대학원
컴퓨터공학과(공학석사)

2004년 아주대학교 대학원
컴퓨터공학과(공학 박사)

2005년~현재: 동덕여자대학교 컴
퓨터학과 전임강사

관심분야: 컴퓨터그래픽스, 영상처
리, 멀티미디어 및 게임