

## 육각형 검색 패턴을 위한 효과적인 움직임 벡터 예측 알고리즘

\*이 도경 \*\*정 제창

서울한양대학교 전자컴퓨터통신 대학원

\*dilkusha1006@gmail.com \*\*jjeong@ece.hanyang.ac.kr

## Motion Vector Prediction Algorithm for Enhanced Hexagonal Search

\*Lee, Do-Kyung \*\*Jeong, Je-Chang

Department of Electronics and Computer Engineering, Hanyang University

## 요약

자연현상이나 실제 상황을 담고 있는 비디오 시퀀스는 기본적으로 주위의 픽셀들과 높은 상관도를 가지고 있다. 따라서 움직임 벡터 정보도 주위 블록의 정보와 상관도가 높으며, 움직임 예측을 수행할 때의 계산량을 줄이는데 용이하게 사용되어져 왔다. 본 논문에서도 이를 이용하여 움직임 벡터를 예측하는 방법을 제시한다. 현재 블록의 정확한 움직임 벡터 예측을 위하여, 참조 프레임의 움직임 벡터와 현재 프레임의 움직임 벡터 정보를 참조하여 시작점을 정확하게 재조정하였다. 더 정확해진 시작점 예측으로 인해 움직임 정보의 중앙으로 수렴하는 확률이 더 높아졌으며, 이를 이용하여 enhanced hexagonal search의 첫 단계를 수정함으로써 search point를 줄였다. 실험결과에서는 제안한 알고리즘의 장점과 단점을 분석하고 각 테스트 시퀀스에 따른 성능을 비교하였다.

## 1. 서론

움직임 예측은 MPEG-1/2/3, H.261/263/264과 같은 많은 비디오 코딩에 대한 표준에 필수적인 요소이다. 이것은 블록 매칭 알고리즘(BMA, Block Matching Algorithm)을 통하여 시간적으로 인접한 프레임들 사이의 중복을 줄여주는데 핵심적인 역할을 한다. 블록 매칭 알고리즘이란 먼저 프레임을 매크로 블록(MB, macro block)으로 불리는 다수의 정사각형 블록들로 분할하고, SAD (Sum of Absolute Difference), SSD(Sum of Squared Difference), MSE (Mean Squared Error) 등의 matching criterion에 의하여 움직임 벡터를 결정하는 것이다. 검색은 일반적으로 현재 프레임의 현재 블록과 같은 위치의 참조 프레임에서 시작하고, 현재 블록과 가장 유사한 블록을 찾아 x축 좌표, y축 좌표로 이루어진 움직임 벡터로 표현된다. 이러한 움직임 벡터는 80% 이상이 중앙 집중형태의 분포(center-biased distribution)를 보이고 있어, 이를 이용하면 uniform distribution을 대체하여 더 정확하고 효율적인 예측이 가능해진다. 블록 매칭 알고리즘의 대표적인 방법인 풀서치(full search)는 검색 범위안의 모든 포인트들의 SAD를 비교하여 계산량이 상당히 많으며, 이러한 부담을 줄이기 위하여 지난 20년 동안 많은 PSNR (Peak to Signal-Noise Ratio)의 손실을 최소화하고 계산량은 현저하게 줄이는 고속 움직임 예측 알고리즘이 여러 논문에서 제안되었다. 이들은 다양한 방법을 통하여 움직임 예측의 계산 부담을 줄이고자 노력하였는데, 그 예로 TSS (three-step search) [1], NTSS (new three-step search) [2], 4SS (four-step search) [3], DS (diamond search) [4], NDS (new diamond search) [5], HEXBS (hexagon-based search)[6], EHS (Enhanced Hexagonal Search) [7] 등이 있다.

HEXBS와 비교하여, EHS는 6-side-based fast inner search와 starting point를 예측하는 predictive HEXBS를 채용하여 검색 속도와 PSNR을 향상시켰다. 6-side-based fast inner search는 픽셀사이의 높은 상관도를 이용하여 육각형(hexagonal) 검색 패턴 중, inner search의 속도를 향상시켰고, predictive HEXBS를 통하여 움직임 예측을 하기 전에 시작점을 재설정하였다.

본 논문에서는 predictive HEXBS를 개선하고 검색 패턴에 변화를 주었다. 이해의 편의를 위하여 small vector라는 용어를 정의하였는데, 그 뜻은 hexagonal search의 첫 번째 단계의 inner area에서 정의된 벡터를 의미한다. 움직임 벡터를 좀 더 정확히 예측하여 small vector의 수를 증가시켜 search point의 수를 감소시켰다. 또한 small vector가 증가하는 현상을 활용하여, EHS 검색 패턴, 첫 번째 단계의 과정을 수정하였다. 본 논문에서는 이와 같이 제안한 과정들에 대한 상세한 알고리즘을 명시하고 실험결과를 FS, TSS, NTSS, NDS, HEXBS, EHS와 비교하였다.

## 2. Enhanced Hexagonal Search(EHS)

HEXBS와 EHS에서의 검색 포인트는 그림 1에 명시되어 있다. 1 ~ 6은 큰 육각형(Large hexagonal) 검색 포인트이며, 0번을 중심으로 둘러싸고 있는 a ~ h는 inner search를 위한 구역이다. HEXBS에서는 inner search를 b, d, h, f에 대하여 실시하고 OMS(One-More-Step) HEXBS[6]를 제안하여 좀 더 정확한 움직임 벡터를 찾도록 하였다. OMS HEXBS는 만약 inner search과정에서 b가 최소 distortion을 갖는 포인트로 정해지면,

a와 c에 대해서 추가적으로 SAD를 비교하는 과정이다. 이와 같이 d가 최소 distortion을 갖는 포인트라면 c와 e를, f가 최소 점이라면 g와 e를, h가 최소점이라면 a와 g의 distortion도 확인하여 최종 움직임 벡터를 정의한다.

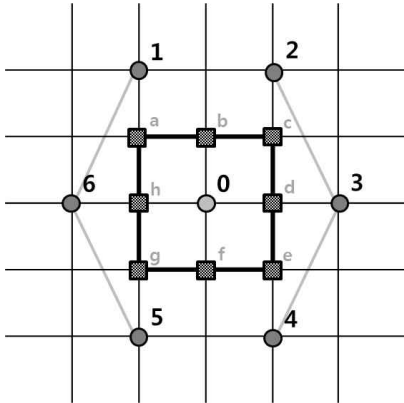


그림 1. HEXBS와 EHS에서 사용하는 기본적인 검색 포인트

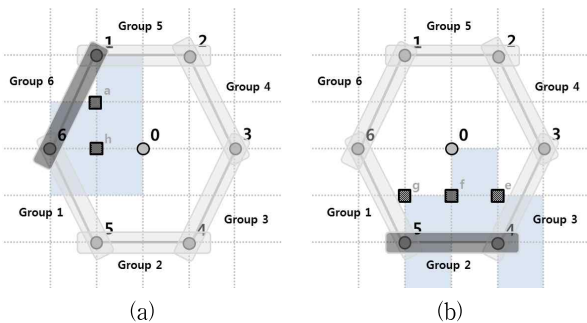


그림 2. 6-side-based fast inner search의 추가적 검색 포인트

하지만 추가적으로 검색 포인트의 distortion을 원점을 제외하고 6개나 구해야하므로 OMS HEXBS의 계산 부담은 더 늘어났다. 그래서 EHS에서는 이러한 inner search의 계산량을 줄이고자 6-side-based fast inner search를 제안하였다. 이 방법은 픽셀의 상관도를 이용하는 것으로 그림 2와 같이 색칠해진 블록들 역시 상관도가 매우 높을 것으로 예측할 수 있다. 이를 이용하여 우선 Group 1 ~ 6을 인접한 포인트를 묶어서 group distortion을 정의한다. 만약 그림 2. (a)와 같이 Group 6의 group distortion이 다른 group들의 distortion 중에서 가장 적다면 추가적인 inner search는 a, h에서만 수행한다. 만약 Group 2가 최소 group distortion을 가진다면 g, f, e 포인트에서 distortion을 구한 뒤 최소 점을 찾아 움직임 벡터를 정의하게 된다.

또한 EHS에서는 predictive HEXBS라는 명칭으로 움직임 예측의 시작점을 정하는 알고리즘을 제안하였다. 움직임 정보도 픽셀의 경우와 같이 이웃해 있는 블록에서 유사한 경향성을 보인다. 그러므로 EHS에서는 현재 블록의 왼쪽과 위쪽 블록의 움직임 정보를 참고하여 움직임 벡터를 예측한다. 이때, 왼쪽 블록과 위쪽 블록의 움직임 벡터가 가르치는 블록과 현

재 블록의 distortion을 계산하여 최소가 되는 지점을 움직임 예측의 시작점으로 제조정하였다. EHS를 비롯한 여럿 서치 패턴 알고리즘은 풀서치에 비해서 손실이 있고 검색 범위안의 모든 포인트를 확인하지 않기 때문에 정확한 움직임 벡터 예측은 PSNR의 향상에 도움을 줄 수 있다.

### 3. 제안하는 알고리즘

움직임 예측을 위해 두 블록 사이의 SAD를 계산하는 것은 상당한 계산량이 들기 때문에, 움직임 예측을 하는 과정에서 검색 포인트의 수를 줄이는 것은 속도개선에 도움은 주지만 PSNR 관점에서 손실이 있을 수 있다. 그러므로 본 논문에서는 움직임 벡터를 정확히 예측하여 PSNR을 높이고 검색 포인트 수 증가를 최소화하는 알고리즘을 제시하고자 한다.

#### 3.1 움직임 벡터 예측 알고리즘

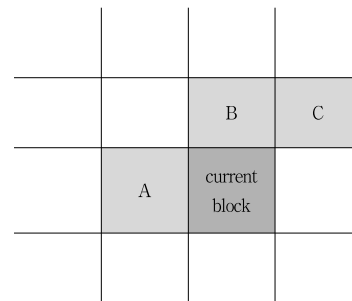


그림 3. 현재 프레임에 있는 참조 블록과 현재 블록의 위치

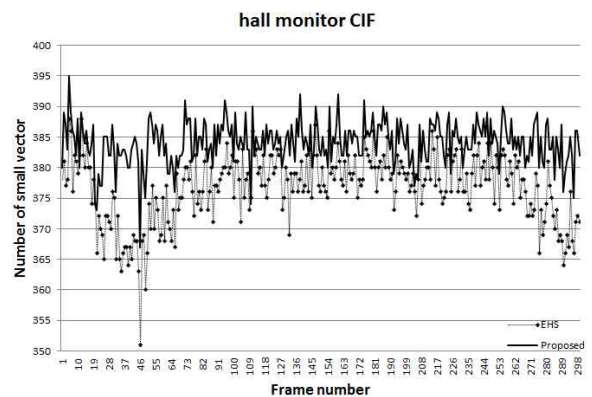


그림 4. 새로운 시작점 예측으로 인한 small vector의 증가량

EHS의 predictive HEXBS에서는 움직임 예측을 위한 시작점을 정하기 위하여 그림 3의 블록 A와 블록 B를 참조하고 현재 블록(current block)의 zero vector 위치에 있는 블록과의 SAD와 비교하여 움직임 벡터 예측을 수행한다. 하지만 이것은 최적의 해답이 될 수 없으며, 좀 더 정확하게 움직임 벡터를 예측하기 위해서는 본 논문에서는 블록 C와 참조프레임의 현재 블록의 위치에 있는 움직임 정보까지 참조하여 시작점

표 1. 다양한 테스트 시퀀스와 비교 알고리즘에 대한 결과

| Sequence     | Frames | Full search |           | Three step search |           | New TSS |           | Diamond Search |           | Hexagonal Search |           | Enhanced HEXBS |           | Proposed method |           |
|--------------|--------|-------------|-----------|-------------------|-----------|---------|-----------|----------------|-----------|------------------|-----------|----------------|-----------|-----------------|-----------|
|              |        | PSNR        | frame/sec | PSNR              | frame/sec | PSNR    | frame/sec | PSNR           | frame/sec | PSNR             | frame/sec | PSNR           | frame/sec | PSNR            | frame/sec |
| akiyo        | 300    | 42.34       | 0.4354    | 42.19             | 0.0366    | 42.33   | 0.0281    | 42.01          | 0.0082    | 41.28            | 0.0078    | 41.92          | 0.0075    | 42.32           | 0.0081    |
| bus          | 150    | 25.59       | 0.4770    | 24.27             | 0.0388    | 24.41   | 0.0414    | 22.24          | 0.0142    | 22.00            | 0.0112    | 23.57          | 0.0091    | 25.13           | 0.0096    |
| football     | 150    | 24.08       | 0.4838    | 23.46             | 0.0400    | 23.29   | 0.0382    | 22.94          | 0.0110    | 22.78            | 0.0094    | 23.27          | 0.0083    | 23.57           | 0.0108    |
| foreman      | 300    | 31.81       | 0.4423    | 30.72             | 0.0380    | 29.63   | 0.0363    | 29.43          | 0.0123    | 29.19            | 0.0093    | 30.99          | 0.0086    | 31.58           | 0.0111    |
| hall_monitor | 300    | 34.63       | 0.4345    | 34.56             | 0.0372    | 34.57   | 0.0293    | 34.50          | 0.0090    | 34.39            | 0.0073    | 34.46          | 0.0078    | 34.55           | 0.0091    |
| mobile       | 300    | 25.04       | 0.5103    | 24.57             | 0.0424    | 24.99   | 0.0435    | 24.24          | 0.0105    | 24.36            | 0.0080    | 24.20          | 0.0079    | 24.96           | 0.0095    |
| stefan       | 300    | 23.90       | 0.4546    | 22.43             | 0.0381    | 23.39   | 0.0386    | 20.96          | 0.0126    | 20.91            | 0.0100    | 23.40          | 0.0084    | 24.95           | 0.0106    |
| table        | 300    | 31.46       | 0.5065    | 30.11             | 0.0422    | 30.25   | 0.0359    | 30.01          | 0.0108    | 29.56            | 0.0084    | 30.41          | 0.0089    | 30.94           | 0.0096    |
| tempete      | 260    | 27.79       | 0.5273    | 27.62             | 0.0445    | 27.68   | 0.0395    | 27.22          | 0.0111    | 27.30            | 0.0088    | 26.75          | 0.0082    | 26.84           | 0.0096    |
| average      |        | 29.63       | 0.4747    | 28.88             | 0.0256    | 28.95   | 0.0236    | 28.17          | 0.0071    | 27.98            | 0.0057    | 28.77          | 0.0083    | 29.43           | 0.0098    |

을 재조정하였다. 그림 4를 참고해보면, 새로운 시작점 예측을 사용하였을 경우, small vector의 수가 증가함을 알 수 있다. 이것은 움직임 벡터의 예측이 더 정확해 졌다는 것을 의미한다.

### 3.2 검색 패턴의 수정

3.1 장의 방법을 통하여 정확한 시작점을 얻을 수 있었기 때문에 움직임 정보가 inner search 영역 안에서 정의 될 가능성이 증가하는, 중앙 집중(center-biased) 현상이 더욱 강화되었음을 알 수 있었다. 따라서 이번 장에서는 이와 같은 현상에 맞추어 검색 패턴을 수정함으로써 검색 포인트의 수를 감소시키고자 한다.

EHS의 검색 패턴은 그림 1의 참고하여 설명하면, 일단 원점을 포함한 육각형 모양의 0, 1, 2, 3, 4, 5, 6번 포인트의 distortion을 비교한 후에 만약 0번, 원점의 포인트가 최소의 distortion을 가진다면 inner search를 수행하게 되고 그렇지 않다면 최소의 distortion을 가지는 점을 원점으로 새로운 육각형 모양의 검색 패턴을 시작하게 된다. 검색을 시작하다가 만약 원점이 최소가 된다면 진행을 멈추고 inner search를 하여 최종 움직임 벡터를 정한다. 하지만 앞서 논의한 바와 같이 중앙 집중 현상이 강해짐에 따라 본 논문에서는 첫 번째 단계의 육각형 검색 패턴에 변화를 주었다. 그 과정은 아래와 같다.

단계 1 : 기존의 inner point 중에 0, a, b, c, d, e, f, g, h를 우선적으로 확인한다. 만약 포인트 0가 최소 distortion을 가지고 있다면 그 위치가 최종 움직임 벡터가 되고 검색을 종료한다. 그렇지 않으면 단계 2로 진행한다.

단계 2 : 포인트 0의 distortion과 육각형 모양의 나머지 포인트들 ( 1, 2, 3, 4, 5, 6 )의 distortion을 비교한다. 만약 포인트 0의 distortion이 가장 작다면 단계 3으로 진행하고 그렇지 않으면 육각형 위에 있는 포인트를 원점으로 하는 새로운 검색을 시작한다. 이 때의 검색 방법은 기존의 EHS에서의 방법과 같다.

단계 3 : 원점 포인트인 0가 최소 distortion을 가지고 있으므로 a, b, c, d, e, f, g, h 중에 최종 움직임 벡터가 존재할 가능성이 높다. 그러므로 이들 중 최소 distortion을

갖는 포인트를 최종 움직임 벡터로 정의한다. 참고로 a, b, c, d, e, f, g, h 점의 distortion은 단계1에서 미리 정의되어 있으므로 다시 계산할 필요는 없다.

### 4. 실험 결과 및 분석

본 논문에서 제안한 알고리즘의 성능을 검증하기 위하여, FS, TSS, NTSS, NDSS, HEXBS 그리고 EHS와 비교를 하였다. 실험 조건은 다음과 같다.

- Matching criterion : SAD
- 검색 범위 :  $\pm 16$  (pixels)
- 블록 크기 :  $16 \times 16$  (pixels)
- 테스트 시퀀스 : Akiyo, Bus, Football, Foreman, Hall monitor, Mobile, Stefan, Table, Tempete.

테스트 시퀀스마다 프레임 수가 다르기 때문에 속도 측정을 frames/second로 하였다. 표 1에서 볼 수 있듯이, 움직임 벡터의 활동이 적은, 즉 움직임이 적은 영상인 Akiyo에서는 전체적으로 좋은 결과가 나타났으며 움직임이 많은 Football, Foreman, Mobile, Stefan등의 시퀀스에서는 움직임 예측이 오래 걸리며 PSNR역시 좋지 않은 것을 관찰 할 수 있다. 제안한 알고리즘은 종래의 다른 알고리즘보다 PSNR관점에서 개선이 되었으며, 그것에 따른 속도 저하가 적은 것으로 보이고 있다.

### 5. 결론

본 논문에서는 움직임 벡터를 예측하는 새로운 방법을 제시 하였으며, 이에 따른 검색 패턴에 변화를 주었다. 종래의 방법은 움직임 벡터를 예측할 때에 왼쪽과 위쪽 블록의 움직임 정보만 참조하지만, 왼쪽, 위쪽 뿐 만아니라 대각선 오른쪽 위 위치해 있는 블록과 참조프레임의 움직임 정보까지 고려함으로써 정확도를 높였다. 움직임 정보의 중앙 집중 현상이 강화됨에 따라 검색 패턴을 변화시켰으며 이에 따라 검색 포인트의 수가 줄어들었다. 따라서 제안한 알고리즘은 움직임

예측을 활용하는 비디오 응용 프로그램들과 하드웨어의 성능을 개선하는 데에 도움이 될 것이다.

### ※ 참고 문헌

- [1] T. Koga, K. Iinuma, A. Hirano, Y. Iijima, and T. Ishiguro, "Motion compensated interframe coding for video conferencing". In Proc. Nat. Telecommun. Conf., New Orleans, L.A., Nov.-Dec. pp. G5.3.1-G5.3.5 1981.
- [2] 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, pp. 438 - 443, Aug 1994.
- [3] L. M. Po and W. C. Ma "A novel four-step search algorithm for fast block motion estimation". IEEE Trans. Circuits Syst. Video Technol., vol. 6, pp. 313-317, Jun 1996.
- [4] J. Y. Tham, S. Ranganath, M. Ranganath, and A. A. Kassim "A novel unrestricted center-biased diamond search algorithm for block motion estimation". IEEE Trans, Circuits Syst. Video Technol., vol. 8, no. 4, pp. 369-377, Aug 1998.
- [5] S. Zhu and K. K. 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.
- [6] C. Zhu, X. Lin, and L. P. Chau.: Hexagon-based search pattern for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol, vol.12, pp. 349 - 355, May.(2002).
- [7] C. Zhu, X. Lin, and L. P. Chau.: Enhanced Hexagonal search for fast block motion estimation. IEEE Trans. Circuits Syst. Video Technol, vol.14, pp. 1210 ~ 1214, October 2004.

### ※ 사사 (Acknowledgement)

이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2011-0011313)