

# 분산 기반의 무손실 부분 정합 오차 탐색

\*정동진 \*\*김용훈 \*\*\*정제창

한양대학교 전자컴퓨터통신공학과

\*dlying@naver.com

## Variance based Lossless Partial Distortion Search

\*Jung, Dong-Jin \*\*Kim, Yong-Hoon \*\*\*Jeong, Je-Chang

Electronics and Computer Eng. Dept., Hanyang University

### 요약

국내 IT시장에서 비디오 분야가 차지하는 분야는 매우 크다. 스마트폰, 테블릿 PC, 3D, UHDTV 등 간접적으로나 직접적으로 비디오 분야와 밀접한 관계를 맺고 있다. 이런 추세에 따라서 멀티미디어를 무선 인터넷 환경에서 실시간으로 감상하기 위해 비디오 분야에서는 높은 압축률의 코덱이 선결 과제이다. 이에 따라 기존의 H.264에서 좀 더 발전한 HEVC 차세대 비디오 코덱의 표준화가 진행 중이다.

이에 본 논문은 비디오 분야에서 가장 큰 복잡도를 갖는 움직임 추정 부분과 관련해서 새로운 알고리즘을 제안한다. 제안하는 알고리즘을 사용해 복잡도를 줄이고 기존 알고리즘과 비교해 화질 열화를 개선함으로써 차세대 코덱에 필요한 데이터 감소에 크게 이바지 할 것으로 보인다.

### 1. 서론

현재 방송계나 통신계에서도 방송융합이 제일 큰 화두이다. 이런 걸음에 맞추어서 비디오 분야에서는 3D, 차세대 코덱 등 다양한 표준화가 진행되고 있다. 이 중에서 차세대 코덱 HEVC는 기존의 H.264에서 포화상태에 있었던 압축효율을 좀 더 늘려서 MMS나 UHDTV 등 차세대 방송통신을 도입할 시 부족한 대역폭의 문제점을 해결하는데 크게 이바지 할 것으로 보인다.

비디오 분야 중에서 움직임 추정 분야는 높은 복잡도를 가지고 있어서 부호화기의 전체 복잡도에 50%이상을 점유하는 분야이다. 움직임 추정 분야에서 가장 기본이 되는 알고리즘은 전역 탐색(Full Search) 알고리즘으로 현재 블록에 해당하는 모든 탐색지점의 정합오차들을 모두 탐색하기 때문에 계산량이 매우 크지만 가장 이상적인 움직임 정보를 획득할 수 있다. 이 높은 복잡도를 줄이기 위해서 여러 가지 알고리즘들이 제안이 되었다

첫 번째는 Successive Elimination Algorithm(SEA)[1] 계열로 대표적인 알고리즘들로는 SEA로 Multi-level Successive elimination Algorithm(MSEA)[2]가 있는데 이 알고리즘들은 현재 블록에 해당하는 모든 탐색지점에서 정합오차를 구하기 때문에 무손실 알고리즘이다

두 번째는 Partial Distortion Search(PDS) 계열 알고리즘으로 Normalized Partial Distortion Search(NPDS)[3]와 Adjustable Partial Distortion Search(APDS)[4] 알고리즘이 대표적이다. 이 알고리즘들은 정합오차를 구하는 과정에서 조기 종료 조건을 사용하기 때문에 전역 탐색과 비교했을 때, 손실 알고리즘이다.

마지막은 이진 변환 블록 알고리즘으로 대표적인 알고리즘들로는 1BT[5], 2BT[6], Truncated gray code[7]가 있다. 이 알고리즘들의 특징은 8bit로 나타내지는 화소 값을 8bit보다 적은 bit로 변환 후, 하드웨

어 친화적인 블록 정합 오차 방법을 사용해서 계산 속도를 줄인다.

본 논문에 2장에는 앞에서 언급한 알고리즘의 간단한 설명과 제안하는 알고리즘을 설명할 것이다. 3장에서는 기존 알고리즘들과 제안하는 알고리즘의 실험결과와 분석을 할 것이다. 마지막으로 4장에서는 결론을 맺는다.

### 2. 기존 알고리즘과 제안하는 알고리즘

#### 2.1 연속 제거 알고리즘(SEA)

식 1은 수학적 부등식으로

$$\left| \sum_{i=1}^N \sum_{j=1}^N |f(i,j,t) - f(i-x,j-y,t-1)| \right| \tag{1}$$
$$\leq \sum_{i=1}^N \sum_{j=1}^N |f(i,j,t) - f(i-x,j-y,t-1)|.$$

명확하게 참이 되는 것을 알 수 있다. SEA 알고리즘에서는 식 1을 변형하여 다음과 같은 식 2와 3을 만들었다.

$$R - M(x,y) \leq SAD(x,y) \tag{2}$$

$$M(x,y) - R \leq SAD(x,y). \tag{3}$$
$$R - SAD(m,n) \leq M(x,y) \leq R + SAD(m,n).$$

여기서, R은 현재블록 화소 값들의 합이고, M은 이전 프레임의 정합 후보 블록 화소 값들의 합이다. SAD(m,n)은 현재블록과 이전 프레임의 정합 후보 블록 화소 값들 차이의 합이다. SEA는 식 3의 조건을 사용해서 이 조건을 충족하면, 블록정합 오차를 구하는 과정을 거치고, 충족하지 않으면, 다음 탐색지점으로 넘어간다. 결과적으로 식 3 조건을 사용하여 탐색 지점을 줄일 수 있고 무손실로 압축이 가능하다.

## 2.2 NPDS

NPDS는 조기 종료 알고리즘으로 PDS계열 알고리즘에서 대표적인 알고리즘 중에 하나다. NPDS는 현재블록과 참조블록의 정합오차가 전체 블록에 균일하게 분포 되어있다는 가정에서 출발한다. 그래서 다음 그림 1과 같이 16x16 블록을 16개의 지역으로 나눈 다음, 각 지역에서 하나씩의 샘플들을 채취해서 총 16개 화소의 참조블록과의 정합오차를 얻는다.

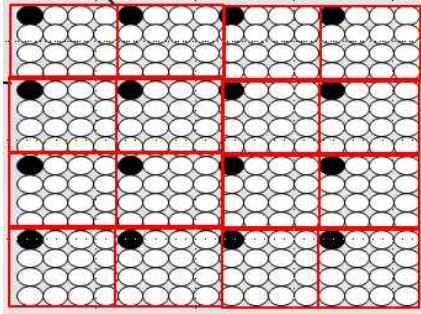


그림 1. NPDS에서 16x16블록을 16개로 나눈 그림

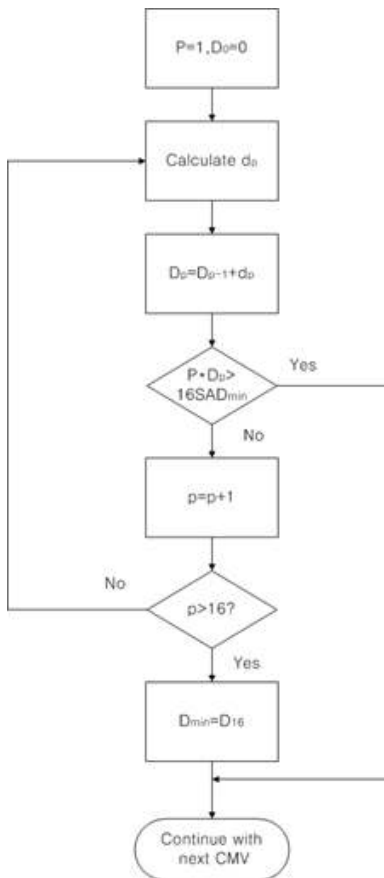


그림 2. NPDS 알고리즘의 흐름도

그림 2의 흐름 도를 보면  $d_p$ 는 각  $p$ 에 대한 부분 정합오차 값이고  $D_p$ 는  $p$ 가 1부터 16까지 변화면서 누적된 부분 정합 오차의 값이다.  $p$ 가 1부터 시작해서 16까지 변화하는데 변할 때마다 다음 조건인

$$16D_p > pD_{min} \quad (4)$$

을 만족하면 조기 종료를 하고, 만족하지 않으면  $p$ 에 1을 더해서 반복과정을 거친다.

## 2.3 제안하는 알고리즘

제안하는 알고리즘은 다음과 같은 가정에서 출발한다. 서브블록이 높은 디테일을 가질수록 정합오차 값이 커질 것이다. 그러므로 높은 디테일을 가진 서브블록부터 부분 정합오차를 구하면 좀 더 빨리 조기 종료를 시킬 수 있다. 여기서 중요한 점은 '높은 디테일을 가지는 것을 어떤 식으로 표현할 것이냐'이다. 그런 점에서 분산이 좋은 term이라 생각되므로 분산을 가지고 알고리즘을 전개해 나가겠다.

서브블록의 분산은 다음 식을 이용해서 구하였다.

$$Var(X) = E(X^2) - [E(X)]^2 \quad (5)$$

이 식을 이용해서 서브블록의 분산을 구한 다음, 그림 3과 같이 서브블록의 분산이 큰 순서대로 번호를 매겼다

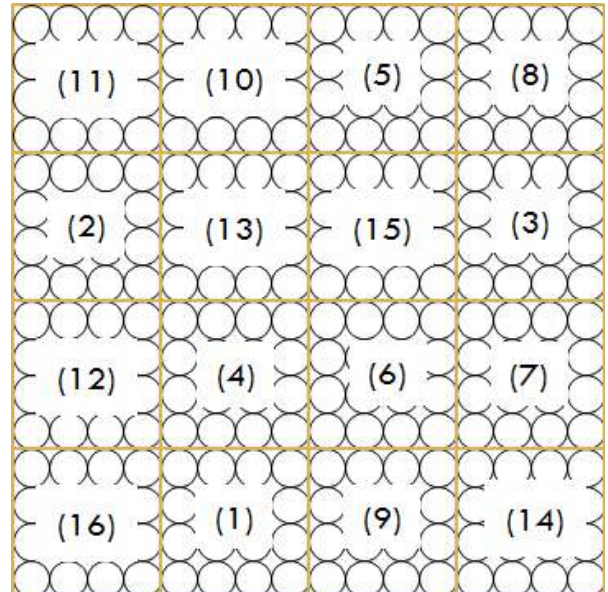


그림 3. 제안하는 알고리즘의 부분 정합오차의 순서를 나타낸 그림

제안하는 알고리즘은 최소 정합오차 값 대신 정합오차 문턱 값을 사용하는데 다음 식과 같다.

$$SAD_{th(p)} = SAD_{th(p-1)} + W_p * SAD_{min} \quad (6)$$

for = 1,2,3...16 and  $SAD_{th(0)} = 0$ .

$$W_p = V_p / VS, \quad VS = \sum_{p=1}^{16} V_p. \quad (7)$$

그림 4를 보면 전체적인 흐름은 NPDS와 같은데 조건문  $D_p > SAD_{th}$ 이 다른 것을 알 수 있다.

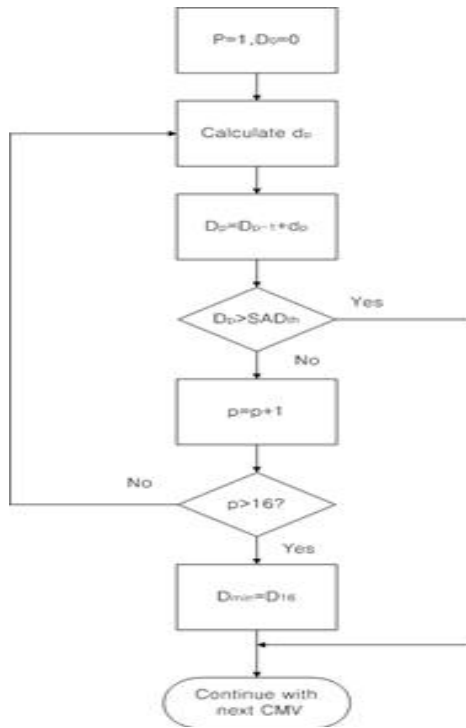


그림 4. 제안하는 알고리즘의 흐름도

추가적으로 제안하는 알고리즘에 SEA를 적용하였다. SEA 알고리즘의 조건문 식 3을 이용해서 실제 움직임 벡터가 될 수 없는 탐색지점들은 없애서 알고리즘의 전체적인 복잡도를 낮추었다.

### 3. 실험결과

실험은 352x288 영상 CIF를 가지고 실험을 하였고 각 영상마다 가지고 있는 최대 프레임에 대해서 코딩을 하였고 FS, SEA, NPDS, TS-EPDS(Efficient Two Step Edge based Partial Distortion Search)[8], 제안하는 알고리즘에 대해서 실험을 하였다. Speed-up은 복잡도를 측정하는 방법으로

$$Speed-up = \frac{\text{operation's FS}}{\text{operation's BMA}} \quad (6)$$

측정을 하였다. 그리고 Operation은 다음과 같이 구하였다.

$$Operation : add + comparison + mul * 8 + abs. \quad (7)$$

여기서 곱셈 연산자는 다른 연산자보다 복잡도가 높기 때문에 가중치 8을 두어서 연산량을 구하였다.

PSNR	Silent	Children	Bus	Mobile	Average
FS	35.98	29.83	25.65	25.18	29.16
SEA	0	0	0	0	0
NPDS	-0.17	-0.21	-0.2	-0.18	-0.19
TS-EPDS	-0.11	-0.09	-0.19	-0.11	-0.12
Proposed	0	0	0	0	0

표 1. PSNR(dB)차분 값(FS) 성능 비교

표 1은 PSNR 실험결과로 FS는 본래의 PSNR 값을 나타내었고 다른 알고리즘들은 FS와 비교한 차분 값 PSNR을 나타내었다. 여기서 (-)는 FS와 비교할 때, PSNR 감소가 있다는 것을 뜻한다. 표 2를 보면 제안하는 알고리즘은 FS와 비교했을 때 PSNR 감

소가 거의 없는 것을 확인 할 수 있다. 그러나 TS-EPDS 나 NPDS는 대략 0.1~0.2dB의 감소를 확인 할 수 있다.

Speed-up	Silent	Children	Bus	Mobile	Average
FS	1	1	1	1	1
SEA	8.25	10.64	2.46	2.80	6.04
NPDS	11.51	11.52	10.69	11.45	11.29
TS-EPDS	38.97	39.61	10.75	25.58	28.73
Proposed	22.14	25.05	8.0	16.86	18.01

표 2 Speed-up 성능 비교

표 2는 Speed-up을 측정한 표로 숫자가 높아질수록 FS와 비교했을 때, 몇 배 빠른지를 나타낸다. 여기서 NPDS는 보통 FS와 비교했을 때 11배 정도 빨라지고 TS-EPDS는 30배 정도 빨라지며, 제안하는 알고리즘은 약 18배 정도 빨라지는 것을 확인 할 수 있다.

### 4. 결론 및 향후 연구 방향

제안하는 알고리즘은 FS와 비교했을 때 PSNR 감소가 없는 장점이 있지만 TS-EPDS와 같은 고속 알고리즘과 비교했을 때 속도가 느린 점이 단점이다. 속도를 좀 더 높이기 위해서 SEA 알고리즘을 다른 것으로 대체 할 계획이고, 추가적으로 코딩 시에 복잡도를 줄이는 방법을 연구할 계획이다.

### 사사(Acknowledgement)

- 국문 : “이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(2011-0011313)”
- 영문 : “This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(2011-0011313)”

### 5. 참고 문헌

- [1] Y. Wang and G. Tu, “Successive elimination algorithm for binary block matching motion estimation,” *Electronics Letters*, vol. 36, pp. 2007-2008, Nov. 2003.
- [2] X. Q. Gao and C. J. Duanmu, C. R. Zou, “A Multilevel Successive Elimination Algorithm for Block Matching Motion Estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 3, pp. 501-504, Mar. 2000.
- [3] C. K. Cheung and L. M. Po, “Normalized partial distortion search algorithm for block motion estimation,” *IEEE Trans. CSVT*, vol. 10, pp. 417-422, Apr. 2000.
- [4] C. H. Cheung and L. M. Po, “Adjustable partial distortion search algorithm for fast block motion estimation,” *IEEE Trans. CSVT*, vol. 13, pp. 100-110, Jan. 2003.
- [5] B. Natarajan, V. Bhaskaran, and K. Konstantinides, “Low-Complexity Block-based Motion Estimation via One-Bit Transforms,” *IEEE Trans. Circuits Syst Video Technol.*, vol. 7, no. 5, pp. 702-706, Aug. 1997.
- [6] A. Ertuk and S. Ertuk, “Two-Bit Transform for Binary Block Motion Estimation,” *IEEE Trans. Circuits Syst Video Technol.*, vol. 15, no. 7, pp. 938-946, Jul. 2005.
- [7] A. Celebi, O. Akbulut, O. Urhan, S. Ertuk, “Truncated Gray-Coded Bit-Plane Matching based Motion Estimation and its Hardware Architecture,” *IEEE Trans. Consum. Electron.*, vol. 55, no. 3, Aug. 2009.
- [8] M. G. Sarwer and Q. M. Jonathan Wu, “Efficient Two Step Edge based Partial Distortion Search for Fast Block Motion

Estimation," *IEEE Trans Consum Electron*, vol. 55, no. 3,  
Nov. 2009.