

블록 기반 움직임 추정을 위한 유연한 탐색 알고리즘

정창욱*, 김종호**, 최진구*, 김용득**

*한국산업기술대학교 컴퓨터공학과. **아주대학교 전자공학과

A Flexible Search Algorithm for Block Motion Estimation

Chang-Uk Jeong*, Jong-Ho Kim**, Jin-Ku Choi*, Yong-deak Kim**

*Dept. of Computer Engineering, Korea Polytechnic University

**Dept. of Electronics, Ajou University

E-Mail: *jcu@kpu.ac.kr, *jkchey@kpu.ac.kr,

**jhkim@digitalsis.com, **yongdkim@ajou.ac.kr

요약

블록 정합 기법(block matching algorithm, BMA) 중에서 가장 널리 알려진 3 단계 탐색(three-step search, 3SS) 알고리즘은 큰 움직임 추정에 적합하지만 고정된 탐색 점으로 인하여 작은 움직임 추정에는 계산 면에서 낭비가 심하고 탐색이 잘못될 경우가 대부분이다. 한편, 효율적인 3 단계 탐색(efficient three-step search, E3SS)은 중앙-편중된 움직임 추정을 작은 다이아몬드 탐색(small diamond search, SDS) 알고리즘으로 보완하여 예측성과 탐색 속도를 향상시킨 알고리즘이다. 본 논문에서는 탐색 초기 단계에서 탐색 점을 최적 배치하고 E3SS의 SDS 알고리즘을 변형시킨 탐색 알고리즘을 제안한다. 실험 결과는 제안된 탐색 알고리즘이 E3SS와 비교하여 평균 22% 정도 계산량을 감소시키면서도 MSE(Mean Square Error)의 성능 저하를 거의 보이지 않는 것으로 나타난다.

I. 서론

디지털 동영상은 시간 축 방향으로 많은 중복성(temporal redundancy)이 존재한다. 이것의 효과적인 제거 방법에는 움직임 추정(motion estimation, ME) 기법이 대표적이다. ME는 비디오 인코더 내에서 가장 많은 처리 시간을 요구하므로 ME 알고리즘의 최적화 정도는 동영상 압축 기술의 가장 큰 관건이다. 최근 소프트웨어와

하드웨어 구현에서 주도적으로 사용되고 있는 블록 정합 기법(block matching algorithm, BMA)은 우수한 예측 효율성과 추정의 정확도, 계산상의 복잡도 그리고 구현의 단순성 때문에 가장 광범위하게 사용되고 있는 ME 기법으로서 ITU-T H.261, H.263, MPEG-1, MPEG-2, MPEG-4 [1]와 같은 국제 표준 비디오 압축 기술에 적용된다. 전 탐색(full search, FS) 알고리즘은 탐색 윈도우(search window) 내부의 모든 참조 블록들을 전부 탐색함으로써 움직임 추정의 가장 단순한 해결책을 제시한다. 하지만 이 BMA는 매우 높은 계산 복잡도를 요구하기 때문에 실시간 소프트웨어 기반의 응용프로그램에서 사용하기 어렵다. FS의 이러한 높은 계산 복잡도를 감소시키기 위해 2-D 대수 탐색(2-logarithmic search, LOGS) [2], 3 단계 탐색(three-step search, 3SS) [3], 십자형 탐색(cross search, CS) [4], 새로운 3 단계 탐색(new three-step search, N3SS) [5], 4 단계 탐색(four-step search, 4SS) [6], 블록 기반 경사 하강 탐색(block-based gradient descent search, BBGDS) [7], 다이아몬드 탐색(diamond search, DS) [8], [9] 등 다수의 고속 BMA들이 제안되었다. 이러한 고속 BMA들은 FS 알고리즘보다 탐색 점(search points)을 크게 감소시키면서 각기 다른 탐색 패턴과 탐색 전략을 사용하여 최적의 움직임 벡터를 찾는다.

본 논문에서는 최적의 BMA가 가져야 할 조건들과 관찰한 내용들을 토대로 하여 3SS 알고리즘을 기반으로 하는 유연성이 있는 3 단계 탐색(flexible three-step search, F3SS) 알고리즘을 제안한다.

II. 관찰

실세계 영상들은 움직임 벡터들의 분포가 대부분 중앙에 편중되어있다 [5]. 또한 움직이는 개체 뒤의 배경은 주로 제로 움직임(zero motion) 형태를 취한다. 이것은 알고리즘의 탐색 초기 단계에서 중앙의 탐색 점들에 대한 배치가 알고리즘의 전체적인 성능에 큰 영향을 미친다는 것을 의미한다. 3SS 알고리즘은 큰 움직임 추정 에 적합한 성능을 보이지만 탐색 초기 단계에 탐색 윈도우의 중앙으로부터 단 하나의 탐색 점만 배치시킴으로써 작은 움직임 추정에는 지역적 최소 BDM(local minimum Block Distortion Measure)에 빠질 가능성이 높다. BBGDS 는 3×3 크기를 가진 작은 사각형 패턴을 고정적으로 탐색하는 기법으로서 지나치게 중앙-편중된 탐색만을 고려하기 때문에 이 BMA 의 특징인 비제한적 탐색 단계(unrestricted search step)에도 불구하고 큰 움직임을 보이는 영상에서는 지역적 최소화로 쉽게 빠질 수 있다. 이것은 여러 개의 지역적 최소 BDM 점이 큰 움직임 개체를 담고 있는 이미지 영상들에게서 일반적으로 존재하는 것이기 때문에 발생하는 현상으로, 탐색하려는 점이 전역적 최소 BDM 점(global minimum BDM point)으로부터 멀리 떨어져 있을수록 BDM 이 단조롭게 증가한다는 고속 BMA 들의 기본 가정 [2]에 배치되는 것이다. 그러므로 실세계 영상들의 최적 블록 위치가 어느 방향에서나 존재한다는 것과 움직임 크기의 다양화를 고려한다면 지나치게 중앙-편중된 ME 알고리즘을 일반적인 영상에서 단독으로 사용하기에는 원하는 성능을 낼 수 없다. 한편, N3SS 는 3SS 의 큰 사각형 탐색 패턴과 중앙-편중된 움직임을 고려한 3×3 크기의 작은 사각형 탐색 패턴을 사용하여 비교적 안정된 탐색 성능을 제공한다. 하지만 N3SS 는 중앙 탐색에서 수행되는 절반 탐색-중단 기법(halfway-stop technique)이 5×5 영역을 둘러싸고 있는 전역적 최소 BDM 점을 간과함으로써 발생하는 탐색 정확도 감소의 문제가 있으며 [10], 탐색 윈도우 크기 15×15(W=7)에서, 9×9 크기의 탐색 패턴 위에 8 개와 중앙에 9 개, 전체 17 개의 탐색 점을 탐색 초기 단계부터 필연적으로 가지기 때문에 아무리 빠른 탐색을 수행하더라도 한 블록당 탐색 점은 17 개 미만이 될 수 없다. N3SS 의 중앙 부근에 탐색 문제와 탐색 초기 단계에서의 많은 탐색 점 수는 효율적인 3 단계 탐색(efficient three-step search, E3SS) [10]의 탐색 전략으로 어느 정도 해결된다. E3SS 는 탐색 초기 단계에

서 중앙-편중된 영상을 위한 작은 다이아몬드 패턴을 사용하고 중앙으로부터 탐색 시 비제한적인 탐색 단계를 거침으로써 탐색 점 수의 감소와 더불어 탐색 정확도를 향상시킨다. 결과적으로 중앙 탐색에서의 유연성이 3SS 알고리즘의 전체적인 성능을 끌어올린 것이다.

III. 유연성이 있는 3 단계 탐색 알고리즘

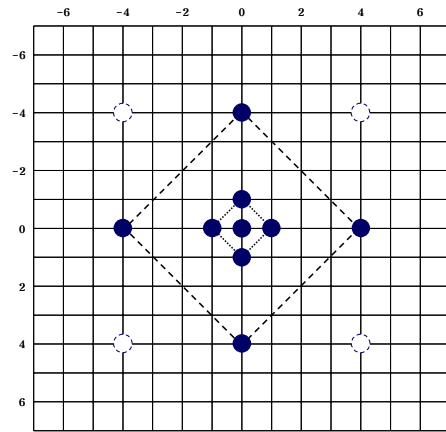


그림 1. F3SS 의 초기 탐색 단계에서 사용된 탐색 패턴.

탐색 윈도우 크기가 W=7 이라고 할 때, 제안된 F3SS 알고리즘은 탐색 초기 단계에서 3SS 처럼 9×9 사각형 패턴 위에 8 개의 탐색 점을 배치하는 대신 상하좌우로 4 개의 탐색 점만을 우선적으로 사용한다. 이것은 대부분의 실세계 영상들이 주로 수평이나 수직 방향으로 움직이는 카메라를 통해 촬영된다는 점 [9]에 착안한 것이다. 그리고 실세계 영상들의 중앙-편중된 움직임 벡터 분포를 고려하여 작은 다이아몬드 패턴을 중앙에 위치시킨다. 그림 1 은 F3SS 의 탐색 초기 단계에서 사용된 9 개의 탐색 점을 보여준다. 만약 제로 움직임 벡터라고 추정되면, 9 개의 점만을 탐색하고 탐색이 중단된다. 최소 BDM 점이 9×9 패턴의 상하좌우 4 개 점들 중에 하나라면, 다음 단계에는 9×9 패턴의 모서리 부분에 해당하는 나머지 4 개의 점들을 탐색한다. 그리고 선택된 최소 BDM 점을 중심으로 하는 작은 다이아몬드-엑스 탐색(small diamond-X search, SDXS)을 수행한다. 여기에서 3SS, N3SS, E3SS 와의 차이점은 작은 다이아몬드-엑스 탐색을 수행하기 전에 9×9 탐색 패턴의 크기를 각 단계마다 절반으로 감소시키지 않는다는 것에 있다. 다만, W=15(탐색 윈도우 크기 31×31)일 경우에는 탐색 초기 단계의 큰 사각형 패턴이 17×17 크기이므로 이것의 절반인 9×9 까지 크기가 감소하게 된다. 3SS 의 탐

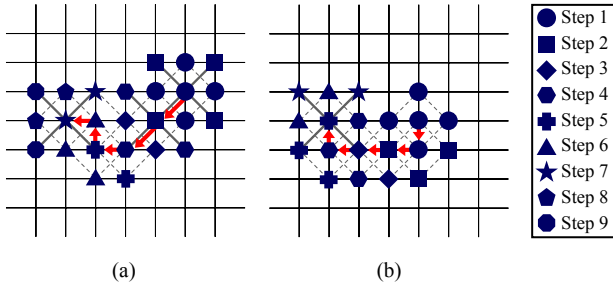


그림 2. SDXS의 비제한적인 탐색 경로 예.

색 단계에 따른 큰 사각형 패턴 크기의 대수적 감소 (logarithmic decreasing)를 제한하는 이유는 작은 다이아몬드-엑스 탐색 기법에 기인한다. F3SS의 작은 다이아몬드-엑스 탐색 방식은 E3SS의 작은 다이아몬드 탐색과 기본적으로 동일하나 패턴의 중앙으로부터 최소 BDM 점이 발견된다면 3×3 크기를 가진 작은 엑스 (small X) 형태의 추가적인 탐색을 시도한다. 작은 다이아몬드 탐색의 작은 엑스 패턴의 추가는 BBGDS보다 더 적은 탐색 수로 비슷한 탐색 정확도를 내기 위한 탐색 전략의 일환이다. 그림 2는 F3SS에서 사용된 작은 다이아몬드-엑스 탐색의 예들을 보여준다. 제안된 F3SS 알고리즘은 W=7에서 다음의 절차로 진행된다.

- 1 단계) 9×9 사각형 패턴 위에 상하좌우로 4개의 탐색 점과 중앙의 1개 그리고 작은 다이아몬드 패턴 위에 4개의 탐색 점을 배치한다. 만약 최소 BDM 점이 탐색 윈도우의 중앙이라면, 탐색을 중단한다. 그렇지 않다면, 2 단계를 진행한다.
- 2 단계) 만약 1 단계의 최소 BDM 점이 9×9 사각형 패턴 위에 상하좌우 4개의 점들 중 하나라면, 사각형 패턴의 모서리 부분에 해당하는 4개의 탐색 점을 추가로 탐색하고 3 단계를 진행한다. 그렇지 않다면, 5 단계를 진행한다.
- 3 단계) 사각형 탐색 패턴의 크기가 9×9 라면, 4 단계를 진행한다. 그렇지 않다면, 사각형 패턴의 크기를 절반으로 감소시키고 그 패턴의 중앙을 이전 단계의 최소 BDM 점에 놓은 다음 축소된 사각형 패턴 위에 8개의 점을 탐색한다. 그리고 다시 3 단계로 간다. 참고로 3 단계의 과정은 W=15 이상일 경우에만 수행된다.
- 4 단계) 이전 단계의 최소 BDM 점을 중앙으로 하는 탐색 점 4개의 작은 다이아몬드 패턴을 배치한다. 만약 작은 다이아몬드 패턴의 중앙이 최소 BDM 점이라면, 6 단계를 진행한다. 그렇지 않다면, 5 단계를 진행한다.
- 5 단계) 이전 단계에서 발견된 BDM 점이 작은 다이아몬드의 중앙이 되도록 하고 작은 다이아몬드 패턴의 나머지 점들을 탐색한다. 만약 최소 BDM 점이 작은 다이아몬드의 중앙에서 발견되면, 6 단계를 진행한다. 그렇지 않다면, 작은 다이아몬드가 탐색 윈도우의 경계에 도달할 때까지 5 단계의 과정을 반복한다. 탐색 윈도우의 경계에 도달하면 탐색은 중단된다.
- 6 단계) 이전 단계의 BDM 점이 3×3 크기를 가진 작은 엑스 패턴의 중앙이 되도록 하고 작은 엑스 패턴의 나머지 점들을 탐색한다. 만약 최소 BDM 점이 작은 엑스의 중앙에서 발견되면, 탐색을 중단한다. 그렇지 않다면, 5 단계로 간다.

W=7 일 때, 제안된 탐색 알고리즘은 큰 움직임 벡터의 추정에서 대체로 9+4+4+4=21 개 이상의 탐색 점이

필요하며 작은 움직임 추정에는 9+3+2=14 개 이상이 요구된다. 반면, E3SS는 최소 13 개의 탐색 점을 필요로 하며 작은 움직임의 추정에 16 개 이상, 최악의 경우에 9+4+8+8=29 개를 탐색한다. F3SS의 작은 다이아몬드-엑스 탐색은 E3SS의 작은 다이아몬드 탐색보다 최소 BDM 점을 확정하는 조건이 까다롭지만 지역적 최소화에 더 잘 대처할 가능성이 크다. 제안된 F3SS 알고리즘의 핵심적인 탐색 전략은 제로 움직임은 빨리 찾아내고 큰 움직임의 추정에는 유연성을 갖도록 하는 것이다.

IV. 실험 결과

본 논문에서 제안된 알고리즘의 성능을 측정하기 위한 실험은 예측된 프레임과 원본 프레임 사이의 MSE, 블록당 평균 탐색 점 개수(search points, SPT), FS 대비 탐색 속도 상승 비율(speedup ratio, SUP) 등 3가지의 평가 항목들을 기준으로 수행된다. 실험에서 사용된 시퀀스들은 “Garden”(CCIR601 720×486, 98 프레임), “Salesman”(CIF 360×288, 299 프레임), “Football”(CIF 360×240, 100 프레임) 등 3개이다. BDM으로는 MAD(Mean Absolute Difference) 왜곡 측정(distortion measurement) 알고리즘이 사용된다. 블록 크기는 16×16을 기본으로 하며 탐색 윈도우 크기는 W=7과 W=15가 각각 사용된다. 표 I-III는 본 논문에서 제안된 F3SS 알고리즘과 FS, 3SS, 4SS, N3SS, DS, E3SS 등 BMA들의 성능을 비교한다. 한 방향으로만 움직이는 경향을 보이는 “Garden” 시퀀스에 대한 표 I에서는 제안된 알고리즘이 다른 알고리즘들에 비해 MSE 수치가 가장 낮으면서도 평균 탐색 점 수 또한 상대적으로 적다는 것을 나타내며, 특히 탐색 윈도우가 W=15일 때 더 뛰어난 효율을 보여준다. “Salesman”과 같은 중앙-편중된 영상에서는 제안된 알고리즘의 탐색 기법이 다른 비교 대상의 알고리즘들보다 우수하다는 것을 표 II에서 확인할 수 있다. 비교적 크고 복잡한 움직임을 보이는 영상에 대한 표 III에서는 제안된 알고리즘이 3SS, 4SS, N3SS, DS, E3SS에 비해 계산량이 많지 않으면서도 오히려 MSE 수치의 향상이 있음을 나타낸다. W=15에서 제안된 알고리즘은 E3SS와 비교하여 MSE의 증가 없이 평균 22%가 더 빠른 탐색 속도를 보이고 있으며 대체로 3개의 탐색 점을 더 적게 탐색한다. 그림 3은 W=15에서 “Football” 시퀀스를 소스로 하는 고속 BMA들의 MSE 성능과 블록당 평균 탐색 수를 프레임 별로 비교한 도표이다.

표 I

“GARDEN” CCIR601 시퀀스에 대한 성능 비교

BMA	Window Size W=7			Window Size W=15		
	MSE	SPT	SUP	MSE	SPT	SUP
FS	131.89	225.00	1.00	99.39	961.00	1.00
3SS	161.35	25.00	9.00	195.08	33.00	29.12
4SS	171.48	21.66	10.39	137.30	30.96	31.04
N3SS	151.52	27.48	8.19	234.23	27.51	34.93
DS	174.05	21.71	10.36	154.95	22.30	43.09
E3SS	150.85	24.49	9.19	131.03	26.14	36.76
F3SS	149.58	21.02	10.70	123.52	22.82	42.11

표 II

“SALESMAN” CIF 시퀀스에 대한 성능 비교

BMA	Window Size W=7			Window Size W=15		
	MSE	SPT	SUP	MSE	SPT	SUP
FS	18.55	225.00	1.00	18.49	961.00	1.00
3SS	19.46	25.00	9.00	19.65	33.00	29.12
4SS	19.23	17.38	12.95	19.42	25.46	37.75
N3SS	18.67	18.13	12.41	18.87	18.10	53.09
DS	19.14	13.81	16.29	19.13	13.81	69.59
E3SS	18.72	14.16	15.89	18.74	14.16	67.87
F3SS	18.69	10.92	20.60	18.69	10.94	87.84

표 III

“FOOTBALL” CIF 시퀀스에 대한 성능 비교

BMA	Window Size W=7			Window Size W=15		
	MSE	SPT	SUP	MSE	SPT	SUP
FS	204.40	225.00	1.00	183.30	961.00	1.00
3SS	226.51	25.00	9.00	233.26	33.00	29.12
4SS	223.98	18.91	11.90	211.77	27.36	35.12
N3SS	217.66	20.89	10.77	227.47	21.25	45.22
DS	225.65	16.50	13.64	213.83	16.89	56.90
E3SS	218.47	17.26	13.04	217.34	18.06	53.21
F3SS	218.26	13.94	16.14	211.90	14.63	65.69

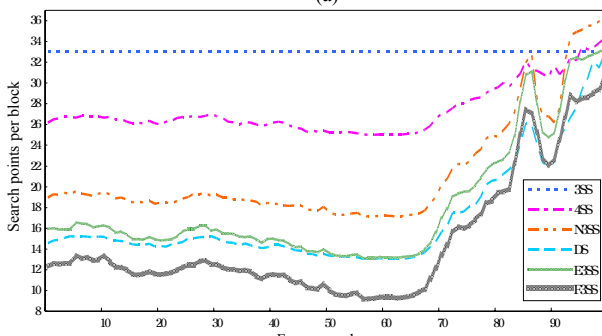
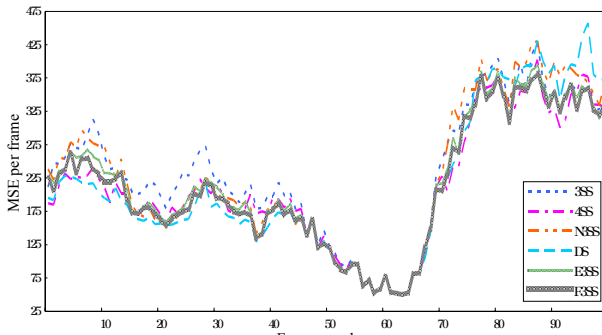


그림 3. W=15 일 때, “Football” 시퀀스에 대한 프레임당 비교. (a) MSE per frame. (b) Average search points per block.

V. 결론

본 논문에서는 E3SS 를 향상시킨 유연성이 있는 3 단계 탐색(F3SS) 알고리즘을 제안한다. 제안된 F3SS 알고리즘은 일반적인 영상의 특성에 근거하여 초기 탐색 점을 최소화하고 큰 사각형 패턴의 단계적인 축소를 제한하며 탐색 마지막에 비제한적인 탐색 단계를 수행하는 작은 다이아몬드-엑스 탐색 기법을 사용한다. 실험 결과는 제안된 알고리즘이 일정한 움직임에 갖는 영상과 작거나 큰 움직임을 포함하는 영상 모두에 적합하다는 것을 나타내며 작은 탐색 윈도우는 물론이고 큰 탐색 윈도우에서도 3SS, 4SS, N3SS, DS, E3SS 보다 성능과 효율성이 상대적으로 더 뛰어나다는 것을 입증한다.

참고문헌

- [1] K. R. Rao and J. J Hwang, *Techniques and Standards for Image, Video and Audio Coding*. Englewood Cliffs, NJ: Prentice Hall, 1996.
- [2] J. R. Jain and A. K. Jain, “Displacement measurement and its application in interframe image coding,” *IEEE Trans. Commun.*, vol. COM-29, pp. 1799-1806, Dec. 1981.
- [3] 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, LA, Nov. 29–Dec. 3 1981, pp. G5.3.1-G5.3.5.
- [4] M. Ghanbari, “The cross-search algorithm for motion estimation,” *IEEE Trans. Commun.*, vol. 38, pp. 950-953, July 1990.
- [5] 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. 438-442, Aug. 1994.
- [6] 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, June 1996.
- [7] L. K. Liu and E. Feig, “A block-based gradient descent search algorithm for block motion estimation in video coding,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 419-423, Aug. 1996.
- [8] 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, pp. 369-377, Aug. 1998.
- [9] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block matching motion estimation,” *IEEE Trans. Image Processing*, vol. 9, pp. 287-290, Feb. 2000.
- [10] Xuan Jing and Lap-Pui Chau, “An efficient three-step search algorithm for block motion estimation,” *IEEE Trans. Multimedia*, vol. 6, pp. 435-438, June. 2004.