

확장된 작은 다이아몬드 패턴을 이용한 고효율의 움직임 추정 알고리즘

정창욱^o, 최진구

와세다대학대학원 정보생산시스템 연구과^o, 한국산업기술대학교 컴퓨터공학과
jeongchanguk@yahoo.co.kr^o, jkchey@kpu.ac.kr

An Expanded Small Diamond Search Algorithm for Fast Block Motion Estimation

Chang-Uk Jeong^o, Jin-Ku Choi

Graduate School of Information, Production and Systems, Waseda University^o

Dept. of Computer Engineering, Korea Polytechnic University

요약

본 논문에서는 다이아몬드 탐색(diamond search, DS)과 효율적인 3 단계 탐색(efficient three-step search, E3SS) 등의 블록 정합 기법(block matching algorithm, BMA)들에서 이용된 작은 다이아몬드(small diamond) 패턴을 광역 탐색에 적합하도록 확장시킨 고속의 움직임 추정 알고리즘을 제안한다. 제안된 알고리즘에서는 탐색 윈도우(search window)의 중앙으로부터 설치된 정사각형 패턴의 크기가 내부에서 대수적으로 감소되며 작은 다이아몬드 탐색(small diamond search, SDS) 기법에 의해 탐색이 완료된다. 실험 결과는 제안된 알고리즘이 DS 보다 평균 3개의 탐색 점을 더 적게 사용하고 E3SS에 비하여 약 5개 정도의 탐색 점 수에 대한 이득을 보이지만 움직임 추정상의 정확도는 다른 고속 BMA들과 거의 동일한 수준을 유지하는 것으로 확인된다.

1. 서론

블록 정합 움직임 추정(block matching motion estimation, BMME)은 비디오 코딩 기술과 표준들에서 가장 중요한 부분이다. BMME는 ISO/IEC MPEG-1, MPEG-2, MPEG-4, ITU-T H.261, H.263 [1] 그리고 최근의 H.264 [2], [3]와 같은 국제 표준 비디오 압축 기술에 적용된다. 블록 정합 기법(block matching algorithm, BMA)은 예측 효율성과 추정상의 정확도가 우수하고 구현의 단순성 때문에 폭넓게 사용되고 있다. BMA는 블록내의 모든 화소가 동일한 움직임 벡터(motion vector, MV)를 가지며 이들이 병진 운동을 한다는 가정하에 현재 프레임의 블록과 참조 프레임을 비교하여 가장 최적의 블록을 결정한다. 지난 수년 동안, 높은 계산 복잡도를 가지는 완전 탐색(full search, FS) 알고리즘을 대체하기 위해서 고속의 알고리즘들이 다수 제안되었다. 제안된 BMA들은 움직임 추정의 정확도와 탐색 속도를 동시에 만족시키기 위해 각기 다른 탐색 전략을 사용한다. 특히 탐색 패턴의 크기와 형태는 알고리즘의 효율

에 적지 않은 영향을 미친다. 고속 BMA의 가장 전형적인 알고리즘에는 3 단계 탐색(three-step search, 3SS) [4]이 있다. 3SS는 각 탐색 단계마다 사각형 탐색 패턴의 크기가 대수적으로 감소(logarithmic decreasing)한다. 3SS 알고리즘은 탐색 전략상 광역 탐색에 적합한 성능을 보이지만 고정된 탐색 패턴의 영향 때문에 탐색 점(search point, SPT)의 개수가 항상 일정하여 계산상의 낭비가 심하고 실제 영상들의 MV 분포가 대부분 중앙에 편중되어있다 [5]는 점을 간과하고 있다. 이후에 등장한 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] 알고리즘 등은 중앙으로 편중된 움직임 추정을 고려하고 있기 때문에 실제 영상들에 대한 탐색에 보다 적합하다. 한편, 다이아몬드 탐색(diamond search, DS) [8]-[10] 알고리즘이 제안되기 전까지는 탐색 패턴의 형태가 대부분 사각형으로 한정되었다. DS는 다이아몬드 형태의 탐색 패턴을 비제한적인 탐색 단계(unrestricted search step)로 수행함으로써 더 적은 탐색 점을 사용하면서도 높은

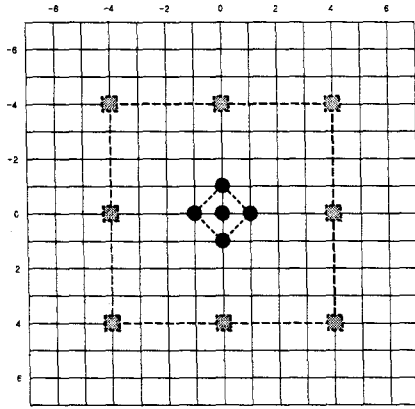


그림 1. ESDS의 초기 탐색 단계에서 사용되는 SDSP와 두 번째 탐색 단계에서의 LRSP. ESDS는 제로 움직임 벡터(ZMV)를 탐색하는데 5개의 SPT만을 사용한다. ZMV가 아니라고 판단되면 다음 단계에서 LRSP의 8개 SPT들을 추가 탐색한다.

예측 품질을 얻는다. 하지만 알고리즘의 특성상 지역적 최소 BDM 점(local minimum Block Distortion Measure point)에 취약한 것은 광역 탐색의 성능을 저하시키는 요인이 된다. 최근 제안된 효율적인 3 단계 탐색(efficient three-step search, E3SS) [11]은 광역 탐색을 위해 3SS의 3 단계 탐색 전략을 사용하고 중앙 탐색 시 작은 다이아몬드 탐색 (small diamond search, SDS)을 수행함으로써 작은 움직임과 큰 움직임의 추정을 모두 고려한다.

본 논문에서는 SDS와 정사각형의 패턴을 이용한 확장된 작은 다이아몬드 탐색(expanded small diamond search, ESDS) 알고리즘을 제안한다. 2장에서는 본 논문에서 제안된 ESDS 알고리즘에 대해서 설명한다. 제안된 알고리즘의 성능은 3장에서 실험 결과로 확인한다. 마지막으로 본 논문에 대한 결론은 4장에서 서술된다.

2. 확장된 작은 다이아몬드 탐색 알고리즘

2.1. ESDS 알고리즘에서 사용된 탐색 패턴

제안된 ESDS 알고리즘은 탐색 초기 단계에서 3x3의 크기를 가지는 작은 다이아몬드 형태의 패턴(small diamond-shaped pattern, SDSP)을 탐색 윈도우(search window)의 중앙에 배치한다. SDSP는 작은 움직임과 제로 움직임 벡터(zero motion vector, ZMV)를 찾아내는데 적합한 최소 크기의 패턴이다. N3SS, DS, E3SS가 ZMV를 찾기 위한 최소 SPT 개수는 각각 17, 13, 13개이지만 ESDS는 단 5개의 SPT만을 필요로 한다. 최소 BDM 점이 SDSP의 중앙이라면 탐색은 중단되며 외곽일 경우에는 다음 두 번째 단계를 진행한다. 15x15 크기의 작은 탐색 윈도우

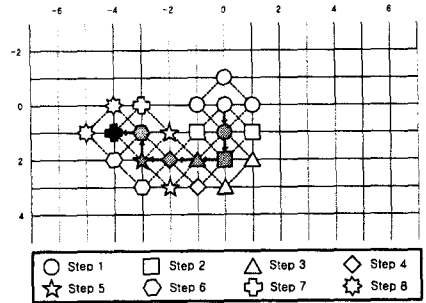


그림 2. SDS 알고리즘의 MV=(-4, 1) 탐색 경로 예시.

(maximum displacement $W=7$)에서는 탐색 윈도우의 절반 크기인 9x9 사각형을 두 번째 단계의 탐색 패턴으로 사용한다. 그림 1에서, SDSP 외부에 점선으로 표시된 8개의 SPT들은 탐색 윈도우의 중앙을 중심으로 설치된 사각형 패턴(large rectangle-shaped pattern, LRSP)의 요소들이다. 특히, $W=15(31 \times 31)$ 이상의 탐색 윈도우에서는 광역 탐색 시 LRSP가 9x9 크기까지 대수적으로 감소된다. 절반으로 크기가 감소된 LRSP는 탐색 윈도우의 중앙을 패턴의 중심으로 하여 배치되며 SPT는 8개로 일정하다. ESDS의 마지막 단계에서는 최소 BDM 점을 중심으로 하는 SDS(그림 2 참고)를 수행한다. 탐색 윈도우 크기 $W=15$ 일 때, 그림 3은 각 탐색 단계에서 사용되는 SDSP와 LRSP를 도식화하여 보여주고 있으며, 그림 4는 ESDS의 탐색 예시를 나타낸다.

2.2. ESDS 알고리즘의 탐색 절차

ESDS 알고리즘은 크게 3단계의 탐색 절차를 가진다. (1) SDSP의 SPT를 탐색하고, (2) LRSP의 SPT를 추가적으로 탐색한 후, (3) SDS 알고리즘으로 탐색을 완료한다. 다음은 ESDS 알고리즘의 각 탐색 단계에 대한 세부적인 설명이다.

(1) 단계: SDSP의 SPT를 탐색

중앙의 SDSP 위에 5개의 탐색 점을 배치한다. 최소 BDM 점이 탐색 윈도우의 중앙이라면 탐색을 완료하며($MV=ZMV$), 그렇지 않다면 (2)단계를 진행한다.

(2) 단계: LRSP의 SPT를 탐색

탐색 윈도우의 절반 크기인 LRSP를 가정한다. LRSP의 중심 점이 탐색 윈도우의 중앙이 되도록 배치하고 그 패턴 위에 놓인 8개의 SPT들을 이전 단계의 최소 BDM 점과 비교 탐색한다. (1) 단계의 SPT가 최소 BDM이라면 (3) 단계를 진행한다. 그렇지 않다면 다음의 부차적인 (2.5) 단계를 진행한다. 참고로, $W=7$ 인 경우 LRSP 크기는 9x9이며, $W=15$ 인 경우 17x17이다.

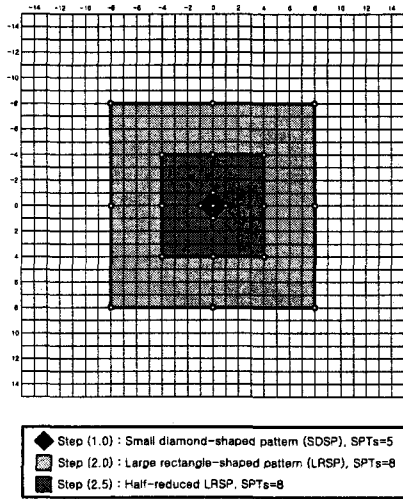


그림 3. 탐색 윈도우 크기 $W=15$ 일 때, ESDS에서 사용되는 탐색 패턴들. 광역 탐색 윈도우에서, (2.5) 탐색 단계에 진입하면 LRSP는 9×9 크기까지 축소가 진행되며 항상 탐색 윈도우의 중앙에 배치된다. 즉, 외부에서 내부로의 탐색이 수행된다.

(2.5) 단계: LRSP의 축소

LRSP의 크기를 절반으로 축소한다. 축소된 LRSP의 크기가 9×9 미만이라면 (3) 단계를 진행한다. 그렇지 않다면 LRSP를 탐색 윈도우의 중앙에 배치하고 그 패턴 위에 놓인 8개의 SPT들을 이전 단계의 최소 BDM 점과 비교 탐색한다. 그리고 (2.5) 단계를 반복한다.

(3) 단계: SDS 알고리즘의 수행

이전 단계의 BDM 점이 3×3 크기를 가진 SDSP의 중앙이 되도록 하고 SDSP의 나머지 SPT들을 탐색한다. 최소 BDM 점이 SDSP의 중앙이라면 탐색을 중단한다. 그렇지 않다면 (3) 단계의 과정을 반복한다.

ESDS 알고리즘은 최소 5개의 SPT만으로 ZMV의 탐색을 완료하며 중앙에 배치된 LRSP를 사용함으로써 지역적 최소 BDM에 적절하게 대응한다. 작은 탐색 윈도우($W=7$)에서는 작은 움직임 추정을 위해 최소 $5+8+3=16$ 개의 SPT를 사용하며 광역 탐색을 위해 최소 $5+8+4=17$ 개의 SPT를 탐색한다. 광역 탐색 윈도우에서는($W \geq 15$) 대수적으로 감소된 LRSP를 이전 LRSP의 내부에 배치시키는 탐색 전략을 사용한다. 이것으로 중간 크기를 가진 MV의 탐색을 강화하는 동시에 지역적 최소 BDM으로 인한 잘못된 방향으로의 탐색을 보완하는 역할을 기대할 수 있다. $W=15$ 에서는 중앙으로 편중된 MV의 탐색을 위해 최소 $5+8+3=16$ 개의 SPT를 필요로 하며 크고 거친 움직임 추정을 위해 최소 $5+8+8+4=25$ 개의 SPT가 요구된다.

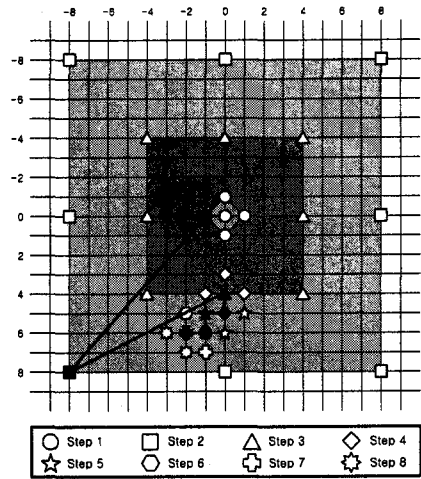


그림 4. $W=15$ 일 때, ESDS의 $MV=(-2, 6)$ 탐색 경로 예시.

3. 실험 결과

제안된 ESDS 알고리즘의 성능 측정 실험은 탐색 속도와 탐색 품질을 평가한다. 탐색 속도는 블록당 평균 탐색 점 개수 (average search points per block, Avg.)와 FS 대비 탐색 속도의 상승 비율로 확인 가능하며(Speedup), 프레임별 블록당 평균 SPT 개수의 최소(Min.) 및 최대(Max.)값을 참고한다. 탐색 품질에서는 예측된 프레임과 원본 프레임 사이의 MSE(Mean Square Error) 측정치뿐만 아니라 FS에서 탐색된 MV를 각 BMA들이 탐색한 MV와 블록당 비교한 최적 MV 적중률(probability of finding true MV per block, Prob. %)도 함께 제시한다. 실험에서 사용된 시퀀스들은 “Claire”(CIF 360×288 , 100 프레임), “Coastguard”(CIF 352×288 , 200 프레임), “Football”(CIF 360×240 , 200 프레임), “Stefan”(CCIR601 720×480 , 200 프레임), “Garden”(CCIR601 720×486 , 90 프레임) 등 5개이다. BDM 측정을 위해 MAD(Mean Absolute Difference) 왜곡 측정(distortion measurement) 알고리즘이 사용된다. 블록 크기는 16×16 을 기본으로 하며 탐색 윈도우 크기는 $W=15$ 를 사용한다. 표 1 5는 본 논문에서 제안된 ESDS 알고리즘과 FS, 3SS, 4SS, N3SS, DS, E3SS 등 BMA들과의 성능을 비교한다. 표 1에서 볼 때, 정적인 움직임 경향을 보이는 “Claire” 시퀀스에서 ESDS는 DS와 E3SS보다 7개 정도의 SPT를 더 적게 탐색함으로써 약 120% 이상 탐색 속도가 향상된 반면에 추정의 정확도는 다른 알고리즘과 비슷하게 유지되고 있는 것으로 확인된다. 거친 움직임이 담긴 “Coastguard”과 “Football” CIF 시퀀스에서는 표 2와 표 3에서 확인할 수 있듯이 MSE가 가장 낮으면서도 탐색 속도는 가장 우수한 것으로 나타난다. 표 4의 “Stefan” 시퀀스는

표 1
"CLAIRE" CIF 시퀀스에 대한 성능 비교

BMA	Search Points				Quality	
	Min.	Max.	Avg.	Speedup	MSE	Prob.
FS	961.00	961.00	961.00	1.00	4.95	100.00
3SS	33.00	33.00	33.00	29.12	5.09	98.20
4SS	25.02	25.81	25.16	38.20	5.06	98.18
N3SS	17.02	18.05	17.40	55.23	5.01	97.86
DS	13.03	13.96	13.23	72.64	4.98	89.07
E3SS	13.03	14.42	13.42	71.61	5.01	97.71
ESDS	5.09	8.41	6.02	159.63	5.01	97.52

표 2
"COASTGUARD" CIF 시퀀스에 대한 성능 비교

BMA	Search Points				Quality	
	Min.	Max.	Avg.	Speedup	MSE	Prob.
FS	961.00	961.00	961.00	1.00	67.92	100.00
3SS	33.00	33.00	33.00	29.12	82.75	95.53
4SS	25.42	35.63	28.64	33.55	75.38	97.00
N3SS	17.73	40.36	20.64	46.56	82.81	86.22
DS	13.64	31.24	17.69	54.32	73.24	98.17
E3SS	14.09	36.52	18.76	51.23	72.37	97.75
ESDS	8.03	31.78	17.56	54.73	71.06	98.15

표 3
"FOOTBALL" CIF 시퀀스에 대한 성능 비교

BMA	Search Points				Quality	
	Min.	Max.	Avg.	Speedup	MSE	Prob.
FS	961.00	961.00	961.00	1.00	190.44	100.00
3SS	33.00	33.00	33.00	29.12	239.52	70.58
4SS	25.02	37.35	30.75	31.25	230.80	72.12
N3SS	17.10	39.61	28.23	34.04	243.66	65.02
DS	13.05	41.67	23.90	40.21	244.90	69.22
E3SS	13.08	35.94	25.48	37.72	230.82	70.94
ESDS	5.28	36.45	21.10	45.55	229.78	68.55

표 4
"STEFAN" CCIR601 시퀀스에 대한 성능 비교

BMA	Search Points				Quality	
	Min.	Max.	Avg.	Speedup	MSE	Prob.
FS	961.00	961.00	961.00	1.00	234.25	100.00
3SS	33.00	33.00	33.00	29.12	305.78	79.76
4SS	25.27	33.11	29.61	32.46	338.85	77.50
N3SS	17.53	34.53	27.10	35.46	333.45	71.46
DS	13.48	27.87	21.50	44.70	415.65	70.97
E3SS	13.68	30.71	24.46	39.29	304.92	79.41
ESDS	6.05	23.65	18.19	52.83	344.65	73.47

표 5
"GARDEN" CCIR601 시퀀스에 대한 성능 비교

BMA	Search Points				Quality	
	Min.	Max.	Avg.	Speedup	MSE	Prob.
FS	961.00	961.00	961.00	1.00	99.08	100.00
3SS	33.00	33.00	33.00	29.12	194.12	70.78
4SS	27.79	32.24	30.99	31.01	137.22	80.13
N3SS	21.12	33.71	27.49	34.96	231.47	62.79
DS	17.20	24.90	22.33	43.04	154.13	84.51
E3SS	18.61	31.88	26.10	36.82	130.26	82.13
ESDS	14.48	25.63	23.02	41.75	116.13	83.16

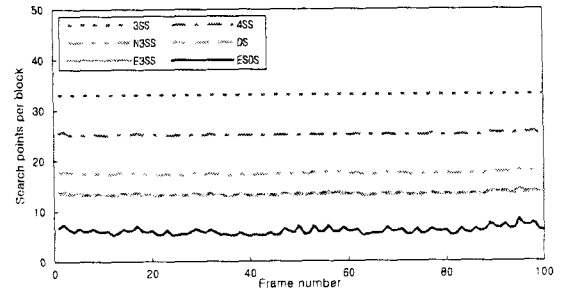
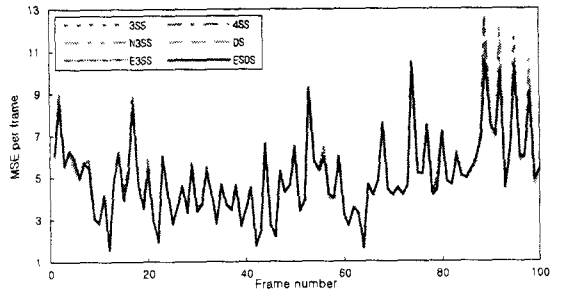


그림 5. "Claire" 시퀀스에 대한 프레임당 비교.
(a) 프레임당 MSE. (b) 블록당 평균 탐색 점 개수.

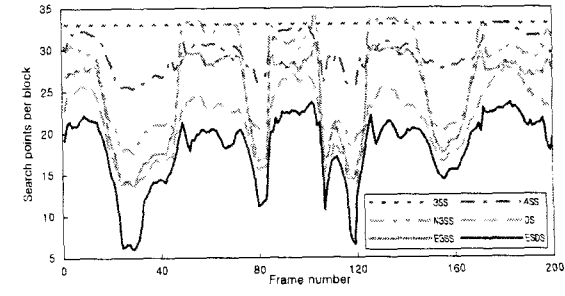
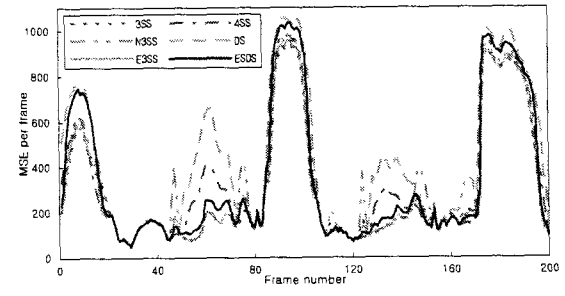


그림 6. "Stefan" 시퀀스에 대한 프레임당 비교.
(a) 프레임당 MSE. (b) 블록당 평균 탐색 점 개수.

720×480의 고해상도이며 거친 스포츠 경기 장면을 포함한다. 표 4의 MSE와 Prob. 항목에서는 DS가 가장 낮은 품질을 보여 주고 있는 것으로 나타나며 3SS, N3SS, E3SS 등의 광역 탐색에 적합한 알고리즘들의 추정 성능이 비교적 좋은 것을 알 수 있다. 제안된 ESDS는 “Stefan” 시퀀스에서 DS와 E3SS의 중간 정도의 탐색 품질을 보여주지만 DS와 E3SS에 비하여 3 6개의 SPT가 이득인 것으로 나타난다. 화면 전체가 수평의 일정한 방향으로만 움직이는 경향을 보이는 표 5의 “Garden” 시퀀스에 대해서도 제안된 알고리즘이 다른 알고리즘들에 비해 MSE 수치가 가장 낮으면서도 탐색 속도 역시 우수하다는 것을 알 수 있다. $W=15$ 에서 실시된 실험 결과는 제안된 알고리즘이 DS와 E3SS보다 평균 3 5개의 SPT를 더 적게 사용하기도 탐색 품질은 거의 동일한 수준으로 유지된다는 것을 보여준다. 그림 5와 그림 6은 각각 “Claire” 시퀀스와 “Stefan” 시퀀스를 소스로 하는 고속 BMA들의 MSE 측정치와 블록당 평균 탐색 점 개수를 프레임별로 비교한 도표이다. 두 시퀀스들은 서로 상반된 움직임 경향을 가지는 영상들이다. 도표의 결과는 제안된 ESDS가 영상의 종류에 관계없이 대부분의 프레임 구간에서 걸쳐서 가장 빠른 탐색 속도를 보이며 탐색 품질도 또한 우수하여 특정 영상에 부분 최적화(sub-optimal solution)되지 않은 것으로 확인된다.

4. 결론

본 논문에서는 SDSP와 LRSP를 이용한 확장된 작은 다이아몬드 탐색(expanded small diamond search, ESDS) 알고리즘을 제안한다. 제안된 ESDS 알고리즘은 광역 탐색 윈도우에서 LRSP의 크기가 내부로부터 대수적으로 감소되며 SDS 기법에 의해 탐색을 완료하는 탐색 전략을 사용한다. 광역 탐색 윈도우에서 실시된 실험 결과는 제안된 알고리즘이 DS와 E3SS보다 평균 3 5개의 탐색 점에 대한 이득을 보이면서도 탐색의 정확도는 다른 고속 BMA들과 대체로 비슷한 수준을 유지하는 것으로 나타난다. ESDS 알고리즘은 탐색의 고효율과 구현상의 용이함으로 하드웨어나 소프트웨어 기반의 비디오 응용에서도 적합하다 판단된다.

참고 문헌

[1] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video and Audio Coding*. Englewood Cliffs, NJ: Prentice Hall, 1996.
 [2] H. Schwarz and T. Wiegand, “The emerging JVT/H.26L video coding standard,” in *Proc. of IBC 2002*, Amsterdam, The

Netherlands, Sep. 2002.

[3] “Draft ITU-T Rec. and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264-ISO/IEC 14 496-10 AVC),” Joint Video Team (JVT) of ITU-T and ISO/IEC JTC1, Geneva, JVT of ISO/IEC MPEG and ITU-T VCEG, JVT-G050r1, 2003.
 [4] 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.
 [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] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block-matching motion estimation,” in *Proc. 1997 Int. Conf. Information, Communications and Signal Processing (ICICS)*, vol. 1, Sep. 9 12, 1997, pp. 292 296.
 [9] 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.
 [10] S. Zhu and K. K. Ma, “A new diamond search algorithm for fast block matching motion estimation,” *IEEE Trans. Image Process.*, vol. 9, no. 2, pp. 287 290, Feb. 2000.
 [11] 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.