

이동단말기에서의 영상통신을 위한 움직임추정 알고리즘에 대한 연구

안교주, 문주희
세종대학교 정보통신공학과

A study on the motion estimation algorithm for mobile video communication

Kyo-joo Ahn, Joo-hee Moon

Dept. of Info. & Communications Eng., Sejong University
ahnkj@image.sejong.ac.kr and jhmoon@kunjia.sejong.ac.kr

요약

본 논문에서는 처리 구조, 계산량, 그리고 압축 성능 측면을 동시에 고려하여 head-and-shoulder 영상에 효과적으로 적용될 수 있는 움직임 추정 방법을 제안한다. 제안된 방법은 두 단계로 구성되며, 첫번째 단계에서는 등간격 전역 탐색을 수행하고, 두번째 단계에서는 경계를 갖는 지역 탐색을 수행한다. 제안 방법을 head-and-shoulder 영상에 적용한 결과, 탐색 영역 8 인 경우 전역 탐색과 비교할 때 PSNR 은 0.06 dB 이하로 차이가 나며, 부호화시 발생 데이터양도 103 % 이하로 거의 차이가 나지 않았다 그러나 계산량에 있어서는 전역 탐색과 비교할 때 91% 의 계산량 절감 효과가 있다. 이것은 부호화기 전체 계산량 대비 전역 움직임 추정 계산량이 34.71% 를 차지한다는 사실을 고려해볼 때, 부호화기의 계산량을 31.6% 줄이는 결과를 의미한다.

I. 서론

비디오 부호화에서 시간적인 중복성을 제거하기 위한 움직임 추정은 고압축률을 달성하기 위한 핵심적인 역할을 해왔다. 특히, 여러가지 움직임 추정 기법 중에서 개념 및 구

현이 단순하고 쉽기 때문에 블록 정합 움직임 추정 기법이 대부분 사용되어왔다. 현재의 프레임은 정방형의 블록들로 나누고, 각 블록에 대해 이전 프레임에서 가장 유사한 블록을 탐색하고, 찾아진 이 블록을 기준으로 움직임 벡터를 구하는 것이 블록 정합 움직임 추정의 기본 개념이다. 전역 탐색을 통한 블록 움직임 추정은, 데이터 압축의 관점에서는 최적이지만, 계산량이 너무 많아 영상 통신용 이동 단말기에는 적합하지 않다. 또한 기존에 제안되었던 많은 고속 블록 움직임 추정 알고리즘들은, 계산량 측면에서는 유리하지만, 병렬 처리가 어려운 방법이거나 데이터 압축면에서 불리한 방법들이다. 이동단말기에서 영상통신 서비스를 효과적으로 제공하기 위해서는 적당한 병렬 처리 구조 및 적은 계산량, 그리고 데이터 압축면에서의 효율을 동시에 고려해야만 한다. 전역 탐색을 통한 블록 정합 움직임 추정을 할 경우, 표 1 에서와 같이 탐색 영역 8 에서는 전체 부호화 시간의 34.71 % , 탐색 영역 16 에서는 63.47 %를 차지함을 알 수 있다. 표에 나온 실험 결과는 MPEG-4 부호화기 MoMuSys-FDIS-V1.0-990812[1] 로 foreman qcif 176x144x290 영상을 부호화하였고, 복잡도 측정 도구 gprof[2] 를 사용하였다.

표 1 부호화시 움직임 추정 계산 시간 비교

탐색 영역 모듈	탐색 영역 8 x 8	탐색 영역 16 x 16
부호화 시간	322.74 초	594.53 초
움직임 추정 시간	112.02 초	377.35 초
부호화 시간 대비 움직임 추정 시간	34.71 %	63.47 %

표 1 에서와 같이 부호화기를 최적화 시키는데 있어 움직임 추정 부분의 계산량을 줄이는 것이 효과적이기 때문에 1980 년 초부터 최근까지 블록 정합 알고리즘에 대한 많은 연구가 이루어졌다. 대표적인 블록 정합 알고리즘은 Three Step Search(TSS)[3], New Three Step Search(NTSS)[4], Four Step Search(4SS)[5], Diamond Search(DS)[6] 등이 있다. 본 논문에서는 움직임 벡터의 중앙 집중 특성과 실제 움직임에 가까운 움직임 벡터일수록 보상 오차가 작아진다는 특성을 이용한 2 단계 움직임 추정 기법을 제안한다. II 절에서는 일반적 현상인 움직임 벡터의 중앙 집중 특성을 살펴보고 III 절에서는 제안 알고리즘을 IV 절에서는 실험 결과를 다룬다.

II. 움직임 벡터의 중앙 집중성

움직임 벡터의 중앙 집중 특성이란 움직임 벡터가 0 이 될 확률이 상당히 높다는 것이다. 즉, 움직임이 없는 블록들이 많이 발생한다는 의미이다. 이 특성은 NTSS[4]에서 처음으로 사용되었으며, 이후 많은 블록 정합 알고리즘들이 이 특성을 적용시켰다. 본 논문에서는 이 특성을 검증해 보기 위하여 QCIF head-and-shoulder 영상에서 MPEG-4 부호화기 MoMuSys FDIS V1.0 990812 으로 실험했으며, 실험 결과는 표 2 와 같다. 실험에 사용된 영상은 foreman, carphone, news 를 각각 290 프레임 사용하였다. 중앙 집중 여부를 결정하기 위해 제로 벡터 중심으로 그림 1 과 같이 영역을 나누었으며, 그림에 표시된 숫자는 표 2 에서 r 값에 해당된다. r = 1 인 경우는 검출된 움직임 벡터 크기가 1 이거나 1 보다 작음을 의미한다. 표에서와 같이 움직임 벡터가 r = 2 안으로 들어오는 경우가 70% 이상 나타난다. 이러한 특성은 움직임 추정시 제로 벡터 중심으로 탐색을 함으로써 계산량을 줄일 수 있음을 예상하게 한다.

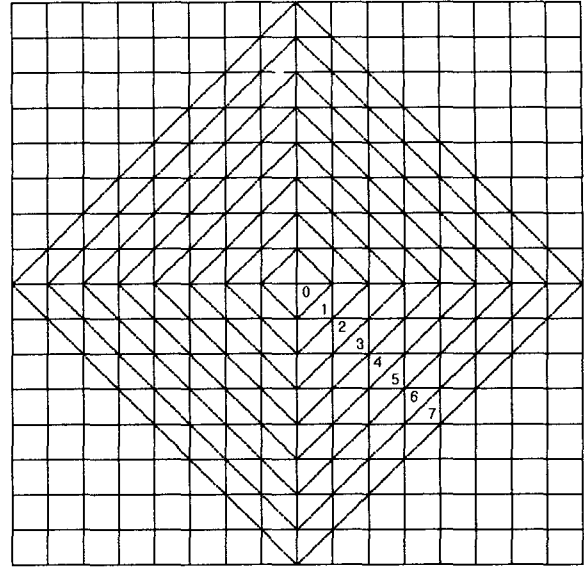


그림 1 움직임 벡터의 중앙 집중성

표 2 움직임 벡터의 중앙 집중성

(단위 %)

탐색 영역 8				
실험 영상	r=0	r=1	r=2	r=3
foreman_qcif	50.06	65.08	73.15	78.52
carphone_qcif	69.11	81.21	88.18	91.19
news_qcif	94.34	97.18	98.43	99.10
탐색 영역 16				
실험 영상	r=0	r=1	r=2	r=3
foreman_qcif	49.69	64.53	72.47	77.72
carphone_qcif	68.85	80.89	87.81	90.76
news_qcif	94.26	97.11	98.35	98.99
탐색 영역 8				
실험 영상	r=4	r=5	r=6	r=7
foreman_qcif	82.10	84.75	87.30	90.02
carphone_qcif	93.07	94.31	95.05	96.09
news_qcif	99.37	99.52	99.63	99.75
탐색 영역 16				
실험 영상	r=4	r=5	r=6	r=7
foreman_qcif	81.10	83.62	86.02	88.09
carphone_qcif	92.51	93.71	94.45	95.23
news_qcif	99.27	99.40	99.50	99.59

III. 제안 알고리즘

본 논문에서 제안하는 움직임 추정 기법이 그림 2 로 설명된다. 첫번째 단계 탐색에서는 움직임 벡터의 중앙 집중성을 고려하여 그림 2 와 같이 제로 벡터 주위로 다이아몬드 형태의 탐색점을 찾는다. 그림 2 에서 작은 원은 1 단계 탐색점을 작은 사각형은 2 단계 탐색점을 나타낸다. 또한 그

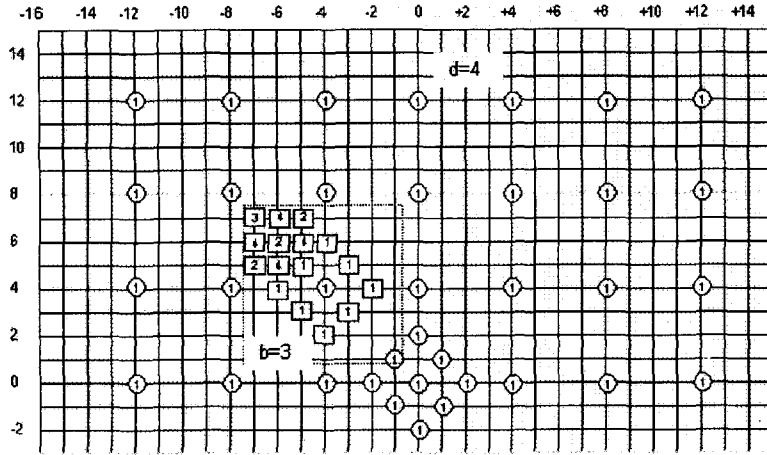


그림 2 제안 알고리즘

탐색 1의 $r > 2$ 인 움직임 벡터가 30% 발생하는 경우를 위해, 제로 벡터를 중심으로 탐색 영역 내에서 d 화소거리의 등간격 탐색점을 동시에 찾는다. 두번째 단계에서는 첫번째 단계에서 찾은 최소 오차 정합점을 중심으로 다이아몬드 탐색, DS 을 수행한다. DS 는 극소수의 특정 블록에 대해 스텝 수가 현저히 많아지는 경우가 있는데, 이것을 대비하기 위해 경계 영역을 두었다. 여기서 b 는 경계 영역의 크기이다. 그림 2 에서 점선은 경계 영역을 의미한다. 제안 알고리즘을 단계별로 살펴보면 다음과 같다.

1 단계 : 탐색 영역의 중심점을 기준으로 d 값에 해당하는 등간격 탐색점들과 다이아몬드 형태의 탐색점을 비교하여 최소 오차 정합점을 찾는다.

2 단계 : 1 단계에서 찾은 최소 오차 정합점을 중심으로 b 값에 해당하는 경계 영역 내에서 DS 를 수행하여 탐색 영역 내의 최소 오차 정합점을 찾는다.

IV 실험 결과

실험 결과는 압축 성능과 계산 복잡도로 나누어서 비교된다. 압축 성능은 전체 부호화기의 데이터 발생량과 휘도 성분에 대한 PSNR 을 사용하여 비교된다. 계산 복잡도는 탐색 영역내 최고 정합점을 찾을 때까지 비교된 탐색점의 수로 비교된다. 실험에 사용된 부호화기는 MoMuSys-FDIS-V1.0-990812 이며 탐색 영역은 $f_code = 0$ 인 $[-8,7]$ 과

$f_code=1$ 인 $[-16,15]$ $[7]$ 로 나누어서 하였다. 양자화 스텝 크기는 I-VOP, P-VOP 에서 각각 10 과 12 로 고정하여 사용하였다. 실험 영상은 각각 290 프레임씩 사용하였다.

표 3 탐색 영역 8에서의 알고리즘 압축 성능 비교

탐색 영역 8		FS		DS	
test seq.	bma	PSNR(Y)	total bits	PSNR(Y)	total bits
foreman_qcif		31.857960	1227200 (100 %)	31.807476	1298488 (105.8 %)
carphone_qcif		32.882061	1024808 (100 %)	32.878101	1044632 (101.9 %)
news_qcif		32.667614	908136 (100 %)	32.669003	907928 (100 %)
bma		제안기법 (d=4,b=3)		제안기법 (d=4,b=4)	
test seq.	bma	PSNR(Y)	total bits	PSNR(Y)	total bits
foreman_qcif		31.800318	1265288 (103.1 %)	31.804510	1263872 (103.0 %)
carphone_qcif		32.872051	1039176 (101.4 %)	32.872513	1038744 (101.3 %)
news_qcif		32.668968	908048 (100 %)	32.668995	908080 (100 %)

표 3 은 탐색 영역 8에서의 알고리즘별로 압축 성능을 보여주고 있으며 FS 는 전역 탐색을 의미하며, total bits 는 실험 영상을 부호화했을 때 발생한 전체 데이터양이다. 제안 알고리즘은 전역 탐색 대비 데이터 발생량이 103% 이내이고 휘도 성분 PSNR 차이도 0.06 dB 이내이다.

표 4 탐색 영역 16에서의 알고리즘 압축 성능 비교

탐색 영역 16				
test seq.	FS		DS	
	PSNR(Y)	total bits	PSNR(Y)	total bits
foreman_qcif	31.872564	1213056 (100 %)	31.811789	1290984 (106.4 %)
carphone_qcif	32.883518	1022920 (100 %)	32.877953	1042552 (101.9 %)
news_qcif	32.671043	908288 (100 %)	32.669544	908088 (100 %)
test seq.	제안기법 (d=4,b=3)		제안기법 (d=4,b=4)	
	PSNR(Y)	total bits	PSNR(Y)	total bits
foreman_qcif	31.801294	1244384 (102.6 %)	31.805416	1243384 (102.5 %)
carphone_qcif	32.871319	1033544 (101.0 %)	32.872253	1033464 (101.0 %)
news_qcif	32.671062	908384 (100 %)	32.671040	908376 (100 %)

표 4는 탐색 영역 16에서의 알고리즘별 압축 성능을 보여 준다. 탐색 영역 8과 동일한 결과를 보여준다.

표 5 알고리즘들의 계산 복잡도

BMA	탐색 영역 : 8 [-8:7]		
	최소값	최대값	평균
기존			
FS	256	256	256
DS	13 (43.89%)	62(0.008 %)	17
제안기법			
d=4,b=2	21(43.72 %)	31(4.28 %)	23
d=4,b=3	21(43.66%)	36(0.015 %)	24
d=4,b=4	21(43.64 %)	39(0.011%)	24
d=5,b=2	21(43.68%)	31(3.34 %)	24
d=5,b=3	21(43.73 %)	35(0.13 %)	24
d=5,b=3	21(43.75 %)	40(0.004%)	24
기존	탐색 영역 : 16 [-16:15]		
	최소값	최대값	평균
FS	1024	1024	1024
DS	13(43.91%)	105(0.004 %)	17
제안기법			
d=4,b=2	61(43.37 %)	71(3.90 %)	64
d=4,b=3	61(43.32 %)	76(0.004 %)	64
d=4,b=4	61(43.28 %)	78(0.011 %)	64
d=5,b=2	61(43.53 %)	71(4.85 %)	64
d=5,b=3	61(43.50 %)	76(0.008 %)	64
d=5,b=3	61(43.54 %)	80(0.03 %)	64

표 5는 알고리즘별 계산 복잡도를 나타낸다. 최소값, 최대값 옆 백분율은 각 값을 가지는 블록의 전체 블록 대비 발생률이다. 제안 알고리즘은 평균 24 번 탐색점을 비교하는

데 이것은 전역 탐색과 비교할 때 계산량을 91% 감소 시키는 것이다.

V 결론

본 논문에서는 이동 단말기에 적합한 움직임 추정 기법에서 head-and-shoulder 영상에 효과적인 알고리즘을 제안하였다. 제안 알고리즘은 탐색 영역 8일 경우 전역 탐색과 비교해서 데이터 발생량은 103% 이내이고 휘도 성분 PSNR 차이는 0.06 dB 이내로 들어오면서 계산량을 91% 감소시켰다.

참고 문헌

- [1] MPEG-4 FDIS reference software
- [2] GNU profiler
- [3] T. Koga, K. Iinuma, A. Hirano, and T. Ishiguro, "Motion-compensated interframe coding for video conference," in Nat. Telecommun. Conf., pp. G5.3.1-G5.3.5, 1981
- [4] 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
- [5] 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
- [6] S. Zhu, K. K. Ma, "A new diamond search algorithm for fast block matching motion estimation," ICICS Conf. Sep. 1997
- [7] ISO/IEC JTC1/SC29/WG11 MPEG099/N2932, "MPEG-4 video verification model version 14.0," October 1999