가산 투영법을 이용한 고속 Block Matching Algorithm

°이 인 홍 박 래 홍

서 강 대 학 교 전 자 공 학 과

A Fast Block Matching Algorithm Using Integral Projection

In Hong LEE Rae Hong PARK

Dept. of Electronic Eng. Sogang Univ.

ABSTRACT

In this paper a fast block matching algorithm using integral projection is proposed to find the motion vector. In this case we can reduce the computation time greatly. Especially when we apply the integral projection to the OTS method, the average ratio of the computational savings is about 4.

1. 서 론

연속하는 영상 데이타 압축방식으로 화상(frame)간의 중복성을 이용하는 interframe coding방식이 효과적이다. Interframe coding에는 CRC(conditional replenishment coding)[1]와 MCC(motion compensated coding)[2]가 대표적인데 이중 MCC의 데이타 감축능력이 더욱 우수하다. MCC는 연속영상에서 움직이는 부분의 이동벡터를 검출하여 prediction에 이용한다.

MCC에서 이동벡터를 구하는 방식으로는 PRA(pel recursive algorithm)[3]와 BMA(block matching algorithm)가 있는데, 이중 BMA는 계산이 간단하고 하드웨어 구현이 용이한 장점이 있으므로 본 논문에서는 BMA의 개선에 관해 연구하였다.BMA에는 DMD방식과 three step search방식[2], mean vector 방식[4],OTS방식[2,5] 등이 있는데 여기서는 기존의 OTS방식에 가산투영법을 적용시켜 계산량을 줄이는 방법을 제안하였다.

본 논문에서는 두화면의 연속영상을 데이타 베이스로 하여 제안한 방법을 중심으로 기존의 OTS방식과 비교분석 하였다.

2. 기본적인 BMA 방식

연속하는 화상들 사이의 이동벡터를 검출함에 있어 시간축으로 인접한 화상들 내에서 부영상(subblock)사이의 상관값(correlation value)를 비교하여 최대치를 이동보상 위치로 이용하는 방법을 BMA라 한다.(그림1. 참조)

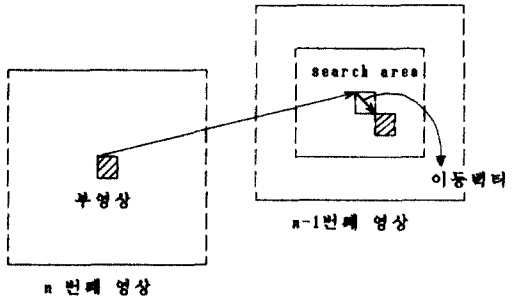


그림1. BMA에서 이동벡터 검출
Fig.1. Detection of the motion vector in BMA.

BMA의 처리과정은 다음과 같다. 우선 화상을 고정된 크기의 부영상으로 나눈다. 이때 이전 화상내의 부영상과의 상관값이 최대로 되는 위치를 구하기 위해 다음과 같은 함수 D(·)를 정의한다.[2]

$$D(u,v)=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N} G(U(m,n)-Ur(m+u,n+v)),\ -p\le u,v\le p \quad (1)$$

여기서 G(·):error power를 구하는 비선형 함수

U : 현 화상내에서 M x N 크기의 부영상

Ur:이전 화상에서 (M+2p)x(N+2p)크기의 search area

p : 최대 이동가능 거리

이다. 이때 이동벡터는 D(u,v)를 최소로 하는 (u,v)로 주어진다. 일반적인 G(·)함수로 MAE(mean of the absolute error)와 NMSE(normalized mean square error)를 이용한다.

여태까지 제안된 BMA중에서 가장 계산량이 작은

OTS방식을 예로 들어보자. 이 방식은 이동벡터를 찾을 경우 그림2에서와 같이 오차가 작은 방향으로 한 점씩 이동해 가며 search를 한다. 여기서 한 점이란 실제 한 부영상을 대표하는 search point를 나타낸 것으로, 각 search point에 대해 식(1)을 이용하여 D(u,v)값을 구하게 된다. 그러므로 부영상의 크기가 8x8이고 G(·)을 MAE라 하면, 한 search point에 대해 64번의 뺄셈,64번의 절대값 계산,63번의 덧셈이 필요하다.(그림3 참조)

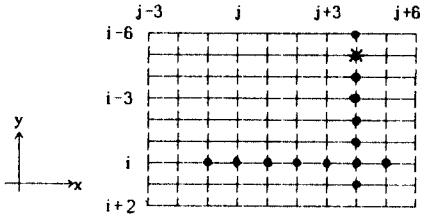


그림2. OTS방식, *:찾고자 하는 위치(i-5,j+4)
Fig.2. OTS method, *:optimal point(i-5,j+4).

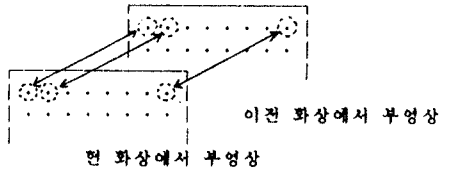


그림3. MAE나 NMSE에의한 cost 함수 계산
Fig.3. Calculation of the cost function by using MAE or NMSE.

3. 가산 투영법을 이용한 계산량 감축

가산 투영법은 화상신호가 있을 때 임의의 방향에서 투영하여 그 투영선상에서 각 화소가 갖고 있는 gray값을 모두 합하는 것이다. 디지탈 화상 f의 열(column)성분화소를 더하여 얻은 벡터를 수직방향의 가산투영, 행(row)성분 화소를 더하여 얻은 벡터를 수평방향의 가산투영이라 한다. 수직(수평) 가산투영은 그 방향의 f의 밝기분포를 나타내므로 가산 투영법은 화상들을 구별하는데 사용할 수 있다. 그림4는 본 논문에서 제안한 수직 가산투영을 나타낸다.

즉 식(1)에서 처럼 화소 개개에 대하여 뺄셈,절대값 계산, 덧셈을 취하는 것이 아니라 일단 기준되는 search point에 대해 수직 가산투영을 통한 값을 구한 다음 이 값들을 가지고 뺄셈, 절대값 계산,덧셈을 하는 것이다. 이것은 원래 MAE나 NMSE로 cost를 계산할 경우 2-D로 해야 하는 것을 가산투영을 이용하여 1-D인 한 방향으로만 고려하는 것이다.

$$D(u) = \frac{1}{M}\sum_{m=1}^{M} G(V(m) - Vr(m+u)) \quad , \quad -p \le u \le p \qquad (2)$$

여기서 V(m)은 현 화상의 부영상에 대한 가산투영값이고 Vr(m+u)는 search area내의 부영상에 대한 가산투영값이다.

그림4(a)는 그림2의 x축 방향으로 search할 경우 사용되는 가산투영 방식이다. 이 경우 바로 앞의 search point에 대해서는 단지 V(9)란 값만 update하면 된다. 그림4(b)는 그림2의 y축 방향으로 search할 경우 사용되는 가산투영 방식이다. 이 경우 바로 위의 search point에 대해서는 맨 아래 행의 값을 빼주고 위 행의 값을 더해주면 되므로 많은 계산상에 이득이 있다.

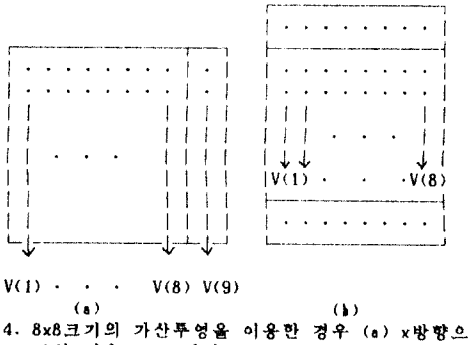


그림4. 8x8크기의 가산투영을 이용한 경우 (a) x방향으로 search할 경우,(b) y방향으로 search할 경우
Fig.4. Integral projection in 8x8 subblock size. (a) search in x-direction,(b)search in y-direction.

예를들어 OTS방식의 경우 한 부영상당 대략 12번의 search point에 대해 계산해야 하므로 64x12 + 64x12 + 63x12 = 2292 번의 계산이 필요한 반면, 가산 투영법을 이용하면 284 + 144 + 96 = 524 번의 계산만 필요하므로 약 77%의 계산량이 감축된다.

4. 실험 결과 및 검토

본 실험에서는 입력으로 크기가 256x256 이고 각 화소당 8 bits인 연속화상을 IBM-PC/XT로 시뮬레이션 하였다. 2가지 종류의 입력을 사용하여 입력1은 움직임이 많은 연속 화상으로, 입력2는 움직임이 적은 연속화상으로 하고 부영상의 크기는 기존의 OTS에서 사용한 것과 같이 8x8로 하였다.[5] 또 성능비교는 계산시간,NMSE, 예측가능화소의

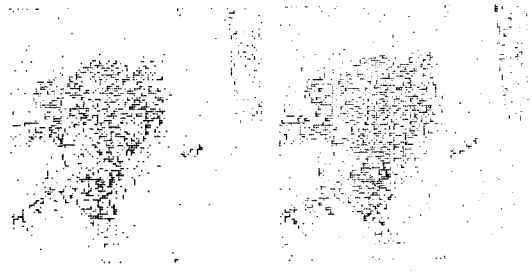
백분율등을 이용하여 OTS방식의 경우 MAE와 가산 투영법을 이용한 결과를 비교하였다. NMSE의 정의는 다음과 같다.

$$NMSE = 10 \times \log \frac{E\{(Si - Ti)^2\}}{E\{Si^2\}} \quad (dB)$$

여기서 Si는 원화상의 gray 값이고 Ti는 운동 보상 결과 얻은 영상의 gray 값이다. 그리고 Ti-Si인 화소를 예측 가능 화소라 정의하여 어느 정도 정확하게 이동벡터를 구했는지 비교하였다.

그림5는 입력1에 대해 기존의 OTS방식과 가산투영을 이용했을 경우 각각에 대한 예측오차를 binary로 보여주는데 여기서 검은 부분이 예측오차가 3이상인 부분을 나타낸다. 그림6은 입력1,2에 대해서 기존의 OTS방식과 가산투영을 사용했을 때 생긴 예측오차의 발생빈도를 비교한 것이다.

그림5와 6으로 부터 BMA에서 상관치를 구하기 위해 기존의 방법에 가산 투영법을 적용해도 오차상에 별 문제가 없음을 알 수 있다.



(a) (b)

그림5. 입력1에서 운동보상결과 생긴 예측오차의 비교 (a)기존의 OTS 경우,(b)가산투영을 이용한 경우

Fig.5. Comparison of the prediction error with MCC for input1.(a)Conventional OTS method,(b)Proposed method using integral projection.

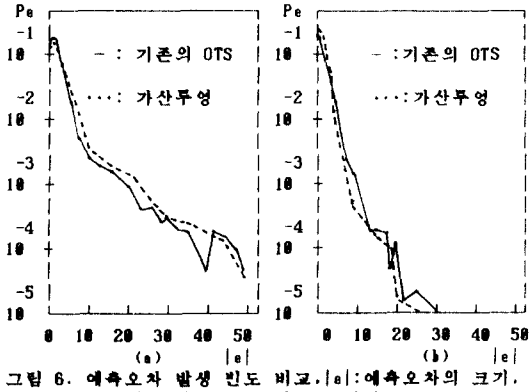


그림 6. 예측오차 발생 빈도 비교,|e|:예측오차의 크기, Pe:|e|의 확률,(a)입력1,(b)입력2

Fig.6. Comparison of the motion tracking performance. (a)input1,(b)input2.

표1은 OTS방식에서 MAE계산시 1:1 subsampling, 2:1 subsampling한 경우와 제안한 가산투영을 사용한 경우 각각에 대해 예측된 영상의 NMSE와 계산시간및 예측가능 화소의 백분율을 비교한 것이다. 이를 보면 가산 투영법이 기존의 OTS방식에 비해 오차상에는 별 차이가 없는 반면 계산시간이 4배 정도 단축됨을 알 수 있다.

표 1.기존의 OTS방식과 가산투영을 이용했을 경우 성능비교
Table 1. Comparison of the performance.

	계산시간 (sec)	예측가능화소의 백분율(%)	예측영상의 NMSE (dB)
MAE 이용 (1:1 sub-sampling)	57	22.70	25.7
MAE 이용 (2:1 sub-sampling)	33	20.40	20.6
가산투영 이용	11	22.90	25.4

5. 결 론

본 논문에서는 기존의 BMA에 가산 투영법을 적용함으로써 약 4배 정도의 계산시간 단축을 보였다. 여기서는 BMA중 OTS방식에 대해서 적용하였지만 이외의 BMA 방식에도 적용이 가능하며 그 경우에도 많은 계산량 감축을 얻을 것이다.

참 고 문 헌

[1] B.G.Haskell, "Frame replenishment coding of TV," in Image Transmission Techniques,W.K.Pratt,Ed., New York:Academic Press, 1979, pp.190-217.

[2] H.G.Musmann et al.,"Advances in picture coding," Proc. IEEE, vol.73, no.4, pp.523-541, Apr.1985.

[3] A.N.Netravali and J.D.Robbins,"Motion-compensated television coding : Part I ," Bell Syst. Tech. J., vol.58, no.3, pp.631-670, Mar.1979.

[4] Y.Ninomiya and Y.Ohtsuka, "A Motion-compensated interframe coding scheme for TV pictures," IEEE Trans. Commun., vol.COM-30, no.1, pp.201-211,Jan. 1981.

[5] R.Srinivasan and K.R.Rao,"Predictive coding based on efficient motion estimation," IEEE Trans. Commun., vol.COM-33, no.8, pp.888-896, Aug. 1985.