적응형 윈도우 크기 기반 NTSS(New Three-Step Search Algorithm) 알고리즘 방법

유종훈 손채봉 오승준 박호종 안창범 *강경옥
광운대학교 VIA 멀티미디어 센터
{ hoon0905[O], cbsohn sjoh, hcpark, cbahn }@media.kw.ac.kr, *kokang@etri.re.kr

An Study Adaptive Window Size based NTSS Algorithm

Jonghoon Yu Chaebong Sohn Seoungjun Oh Hojong Park
Changbum Ahn *Kyeongok Kang

Dept. VIA-Multimedia Center, Kwangwoon University
*Radio and Broadcasting Laboratory, ETRI

요약

NTSS(New Three-Step Search Algorithm)는 대표적인 Fast BMA(Block Matching Algorithm)인 TSS(Three-Step Search Algorithm)에 중앙 편향적(Center-Biased) 특성을 고려하여 향상시킨 방법이다. 그러나 NTSS는 움직임이 작은 영상인 경우에는 TSS보다 개선된 성능을 보여주지만, 움직임이 큰 영상에 대해서는 TSS와 큰 차이가 없으며 탐색영역이 커질수록 오히려 성능이 떨어지는 단점이 있다. 본 논문에서는 움직임 벡터의 특성에 맞는 탐색영역을 적용시킴으로써 탐색영역의 증가로 발생되는 NTSS의 단점을 보완하여 움직임이 큰 영상에 대해서도 향상된 성능을 갖는 방법을 제안한다. 제안된 방법을 적용 하였을 때 움직임이 작은 영상에서는 기존의 방법과 동일한 결과를 얻었으며 움직임이 큰 영상에서는 최고 0.5dB이상 성능이 개선되었다.

1. 서 론

움직임 예측(Motion Estimation : ME)은 영상 정보의 시간적 중복성을 제거하여, 화질의 저하 없이 낮은 비트율로 고속의 영상 부호화가 가능하기 때문에 효과적인 영상 부호화가 가능하다. 이러한 움직임 예측에 있어서 여러 가지 알고리즘(Algorithms)이 적용되고 있는데, 그 중에서도 블록 정합 알고리즘(Block Matching Algorithm : BMA)이 수행에 효과적이며 간단하기 때문에 현재 비디오 코딩 표준에 대표적으로 적용되고 있다[1]. 그 중에서도 가장 높은 성능을 가진 전역 탐색(Full-Search : FS) 블록 정합법은 모든 탐색 점을 검색하기 때문에 최적의 해를 제공하지만 매우 높은 계산량이 요구되기 때문에 실시간 비디오 코딩에는 적합하지 않다. 그러므로 계산량을 줄이면서도 FS에 가까운 성능을 가진 고속의 블록 정합법이 요구 되었고, 많은 고속 블록 정합법이 개발 되었다. 그 중에서도 TSS(Three-Step Search)은 낮은 계산량과 효율적인 측면에서 가장 대표적인 고속 블록 정합법으로 사용되고 있다[2]. 그러나 첫 번째 단계에서 탐색점 간에 거리가 멀어 움직임이 큰 영상에 전역 최소값(Global minimum)을 찾기에는 적합하지만 움직임이 작은 영상에 대해서는 성능이 떨어지는 단점이 있다.

NTSS(New Three-Step Search)은 이러한 TSS의 단점을 보완하기 위해 Center-Biased 특성, 즉 실제 비디오 영상에서 대부분의 움직임 백터(Motion vector)는 중심에서 크게 벗어나지 않는다는 점을 고려함으로써, 움직임이 작은 영상에 대해 TSS의 단점을 보완하기 위해 제안 되었다[3]. 그러나 NTSS는 움직임이 큰 영상에 대한 높은 성능의 개선이 없으며, 또한 움직임이 큰 영상에 적합한 탐색을 위하여 탐색 범위에 크기가 증가 할 경우, 화질은 오히려 감소한다.

따라서 본 논문에서는 영상이 가지는 움직임의 크기를 판단하여 그 특성에 맞게 탐색 범위를 적용함으로써, 움직임이 큰 영상에 대해서도 향상된 성능을 갖는 방법을 제안한다. 탐색 범위 적용에 관한 대표적인 논문으로 Lee와 그의 동료들이 제안한 방법이 있다[4]. 이 알고리즘은 초기 탐색 범위는 변하지 않고 두개의 최소값의 차에 따라 탐색 범위의 크기를 줄여 나가는 방법이다. 그러나 본 논문에서 제안하는 탐색 범위의 크기 조절은 한 매크로 블록(MB)의 움직임 예측을 하기 전에 이전 매크로 블록의 움직임 벡터를 참조하여 탐색 범위를 결정한다.

본 논문의 구성은 2장에서 NTSS 에 관해 간략하게 설명하고 탐색영역의 크기 증가가 어떻게 NTSS의 PSNR(dB)의 감소를 가져오는지에 대해 언급한다. 그리고 3장에서는 영상에 움직임의 크기를 판단하는 두 가지 방법에 대해 설명하고 판단 결과에 따라 다양한 크기에 탐색 범위를 적용하는 알고리즘에 대하여 제안한다. 마지막 4장에서는 제안된 방법을 통한 실험 결과를 보여주며 결론을 맺는다.

2. NTSS(New Three-Step Search Algorithm)

2장에서는, NTSS 알고리즘에 대해 알아본다. NTSS은 TSS에 중앙 편향적인(Center-Biased) 특성을 고려하여 중심점에 이웃하는 8개의 탐색점을 추가한다. 탐색 범위가 W=7인 경우, 만약 최소점이 중심점이라면, 이 탐색은 끝나게 되며, 만약 중심점에 이웃하는 8개의 3x3격자 내에서 최소점을 찾게 되면 오직 3개, 또는 5개의 추가적인 탐색점들이 탐색된다. 만약 그렇지 않을 경우, TSS의 탐색 과정을 수행하게 된다.

그림 1은 5x5 내에서의 움직임 벡터(Motion Vector :MV)를 찾

기 위한 두개의 다른 탐색 경로를 보여준다. NTSS는 정적인 블록(Stationary Blocks)에서 17개의 탐색점을 필요로 하며 중심 5x5 영역 내에서의 작은 MV에 대해 20개 또는 22개의 탐색점을 필요로 한다. 여기서 중요한 것은 탐색 범위가 증가하는 경우, TSS의 첫 번째 단계인 8x8의 탐색점의 거리는 탐색 범위에 크기 증가와 비례하여 증가하는 반면 중심점 주변에8x8내에 위치한 추가된 탐색점은 고정되어 있다는 것이다. 이와 같이 탐색 범위가 증가하는 경우, 중심점에 이웃하는 5x5내의

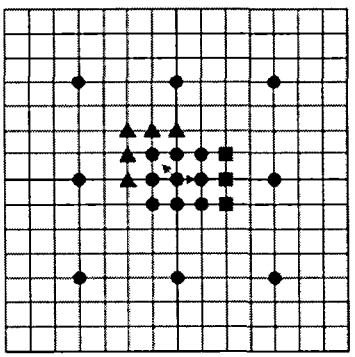

그림 1. NTSS 의 5x5 내에서의 움직임 벡터를 찾기 위한 두개의 다른 탐색 경로

탐색점과 9x9의 탐색점간에 거리는 일정한 비율로 멀어지지게 되며, 만약 전역 최소값이 그 사이에 위치한다면 이러한 탐색은 전역 최소값을 찾지 못하는 경우가 더 증가하게 된다.

그림 2에서 볼 수 있듯이, 실제로 Stefan 영상과 같은 움직임이 큰 영상에서도 위와 같은 이유 때문에 탐색 범위가 클 경우 예를 들어, w=15 일때 w=7에 비해 성능이 떨어지는 것을 알 수 있다.

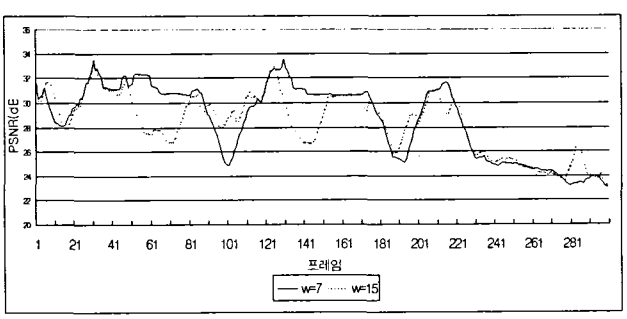

그림 2. NTSS를 두가지 크기의 탐색 범위에 대한 Stefan 영상(CIF(352x288),300프레임)에 대해 수행 하여 얻은 PSNR 비교

이 그림에서는 w=7과 w=15의 PSNR의 차이가 크게 내부분에서 나타나는 것을 볼 수 있다. 먼저 49~81번째와 126~154번째 프레임 내에서는 탐색 범위가 w=7 일때 더 좋은 성능을 보여주며, 90~118번째 그리고 277~292번째 프레임 내에서는 탐색영역이 w=15 일때 상대적으로 좋은 성능을 보여줌을 알 수 있다. 이는 이 영상에서 움직임이 큰 경우에 탐색범위가 큰 것이 더 우수하고 움직임이 작을 때는 탐색 범위가 작은 것이 더 우수하다는 것을 직관적으로 알 수 있다. 따라서 움직임의 특성을 파악하여 탐색 범위의 크기를 조절해 줌으로써 향상된 성능을 얻는 것이 본 논문에 목표이다. 3장에서는 움직임에 크기를 판단하여 탐색 범위를 변화시켜 적용하는 알고리즘에 대하여 소개한다.

3. 영상 특성에 따른 효율적인 탐색 범위 적용

제안된 알고리즘의 탐색은 다른 정합법과 같이 MB단위로 수행 된다. 그리고 탐색을 시작하기 전에 이전 프레임의 움직임 벡터, 또는 그 이전 프레임의 움직임 벡터까지 고려하여 움직임의 크기를 결정한 후, 그 특성에 맞는 탐색 범위 내에서 탐색을 시작한다. 따라서 움직임에 크기를 판단하는 방법은 크게 두 가지로 나뉠 수 있다.

1) 이전 프레임에 MV의 크기를 참조하는 방법

표1은 Stefan 영상에서 탐색 범위의 크기 w=15가 w=7보다 우수한 성능을 보여주는 부분 중 차이가 큰 100번째 프레임의 움직임 벡터의 값과 SAD 값, 그리고 탐색 과정(TSS 또는 NTSS)을 보여주고 있다. 7개의 움직임 벡터는 다른 움직임 벡터들에 비해 탐색 범위의 크기에 따라 비교적 SAD 값의 차이가 큰 움직임을 갖기 때문에 선택 되었다. 이 표에서는 탐색 범위 w=7에 경우 움직임 벡터들이 정확한 전역 최소값을 찾지 못하고 탐색 범위 경계 부분에 걸리면서 w=15에 비해 큰 오류를 갖는 것을 볼 수 있다. 따라서 움직임 벡터가 탐색 범위 경계 부분에 가까울수록 더 큰 탐색 범위가 필요하다는 것을 예상 할 수 있다. 이전 프레임에서 탐색 범위의 경계선상에 위치한 움직임 벡터에 대해 현재 프레임에서 탐색 범위를 조절하는 수식은 식(1)과 같다.

$$\begin{aligned} d_c &= d_p \times 2, \quad if \quad |x| = d_p \ \ or \ \ |y| = d_p \\ d_c &= d_p \times 3, \quad if \quad |x| = 2d_p \ \ or \ \ |y| = 2d_p \end{aligned} \qquad (1)$$

여기서 $MV[x,y,t-1]$ 는 이전 프레임에 움직임 벡터를 나타내며 d_p 와 d_c 는 각각 이전 프레임의 탐색 범위 크기와 현재 프레임의 크기를 나타낸다. 그리고 $x,y \in MV[x,y,t-1]$ 이다.

2) 전 프레임과 두 프레임 전의 움직임 벡터 비교에 의한 방법

1의 방법만으로 움직임이 큰 특성을 갖는다고 단정 할 수는 없다. 그 이유는 전 프레임의 전역 최소값이 탐색 범위 경계선에 걸리지 않더라도 그 이전 프레임의 최소값과 차이가 큰 경우라면 충분히 움직임이 큰 특성을 보여준다고 판단 할 수 있기 때문이다. 움직임의 크기를 판단하기 위한 두 번째 방법으로써, 먼저 움직임 예측의 한 가지 가정인 선형 운동(Linear Motion Hypothesis)을 전제로 한다. 이 가정을 적용하기 위해 이전 프레임과 그전 프레임의 움직임 벡터가 서로 다른 부호를 가졌을 경우를 고려한다. 그리고 두 움직임 벡터가 5x5격자 내에 위치할 때 즉, NTSS의 과정으로 움직임 예측을 했을 경우를 제외하면

$$d_c = d_p \times 2, \quad if \quad |x - x'| \geq 5 \ \ or \ \ |y - y'| \geq 5 \qquad (2)$$

여기서 $x,y \in MV[x,y,t-1] \ \ and \ \ x',y' \in MV[x,y,t-2]$ 이며 $x',y' \in MV[x,y,t-2]$ 는 현재 프레임에서 두 프레임 이전의 움직임 벡터이며, 두 움직임 벡터간의 차가 최소 5

이상일 경우 큰 움직임 특성을 갖는다고 판단할 수 있다. 그 이유는 5x5격자 내에서 두 벡터가 가질 수 있는 최대 차는 4이므로 5이상이 적절하기 때문이다. 따라서 제안된 두 가지 방법으로 영상의 움직임의 크기를 판단하여 적절한 탐색 범위를 적용함으로써 NTSS가 가지는 큰 움직임 특성에 대한 단점을 보완하도록 한다.

표 1. Stefan영상 100번째 프레임의 두개의 탐색 범위에 크기에 따른 움직임 벡터의 값과 SAD값, 그리고 탐색방법

	W=7			W=15		
	SAD	MV	Pattern	SAD	MV	Pattern
(10,8)	3502	(-7,0)	TTS	1485	(-8,0)	TSS
(11,8)	5455	(2,0)	N3SS	2450	(-8,0)	TSS
(12,8)	4399	(-1,-2)	N3SS	2518	(-9,0)	TSS
(2,11)	4118	(5, 0)	TSS	2651	(-8,0)	TSS
(3,11)	3868	(-7, 0)	TSS	2760	(-8,0)	TSS
(4,11)	3972	(-7,-5)	TSS	2034	(-8,0)	TSS
(11,11)	4524	(-2,0)	N3SS	2274	(-8,0)	TSS

4. 실험 과정 및 결과

본 장에서는 제안된 두 가지 방법에 대해 각각의 성능 개선 여부를 알아본 후 두 가지 방법을 조합한 결과를 알아보고 기존의 방법과 비교 분석한다.

성능 실험을 위해 비디오 부호화기 H.263+를 사용하였다[5]. 실험 영상으로는 CIF(352x288)크기의 300프레임 영상Akiyo, Coastguard, Stefan들를 사용하였다. 또한 기본 탐색 범위w는 7, GOP는 첫 번째 프레임만을 인트라(intra)로 부호화 하였고 나머지는 모두 인터(inter)로 부호화하였다. 비트율은 비트율제어 방법을 적용하여 각 영상 특성을 고려하여 10~1024 kbps로 부호화 하였다. 표 2는 NTSS 방법으로 본 논문에서 제안한 첫 번째, 두 번째 방법, 그리고 두 가지를 조합한 방법의 결과를 보여준다. 실험 영상은 움직임이 큰 특성을 가진 대표적인 영상 Stefan을 사용하였다.

표 2. 각각의 제안된 방법을 통한 성능 비교

방법 \ Stefan	PSNR(dB)	
NTSS	28.64(w=7)	28.33(w=15)
(1)의 방법적용	28.80	
(2)의 방법적용	28.82	
조합된 방법적용	28.91	

각각의 방법을 별도로 적용 하였을 때 기존의 NTSS보다 PSNR이 0.16~0.49dB 정도 향상되었으며 조합된 방법을 적용하였을 경우 최대 0.58dB 까지 향상됨을 알 수 있다.

따라서 이 실험을 통하여 움직임이 큰 영상에 경우 제안된 방법이 NTSS의 성능을 개선시킨다는 것을 확인 할 수 있다. 표3 에서는 다른 움직임 특성을 가진 네 개의 영상의 결과를 보여준다. 작은 움직임을 갖는 대표적인 영상 Akiyo의 경우 기존의 NTSS와 제안된 방법은 유사한 성능을 보여준다. 이는 본 논문에서 제시하는 방법이 기본적으로 사용되는 탐색 범위w=7을 사용하기 때문에 움직임이 작은 특성에 대해 성능이 감소하지 않았음을 보여주는 것이다. 그리고 Stefan 과 같은 큰 움직임 특성을 가진 영상뿐만 아니라 Foreman, Coastguard 와 같은 중간 움직임의 특성을 갖는 영상에서도 성능이 향상됨을 알 수 있다. 그림3은 두가지 다른 탐색 범위에 대한 결과를 비교한 그림이다. 제안된 방법의 결과는 탐색 범위가 w=15일 때 감소하는 PSNR을 크게보완

표 3. CIF 영상의 평균 PSNR(dB/Frame)

방법 \ 영상종류	FS	TSS	NTSS	제안방법
Akiyo	35.06	34.80	35.02	35.02
Foreman	31.32	30.60	30.72	30.93
Coastguard	28.11	27.94	28.04	28.06
Stefan	28.90	28.63	28.64	28.91

하였으며 탐색 범위가 w=7인 경우의 감소 또한 상당히 개선된 것을 확인 할 수 있다. 그리고 약 290번째 프레임부터는 영상 내의 움직임이 굉장히 큰 부분으로써 탐색 범위 w=5와 w=7를 사용 하였을 때 PSNR이 크게 감소. 그러나 본 논문에서 제안된 방법의 경우 (w=21)까지 탐색 범위의 크기가 적용 되어 그림과 같이 두 고정된 탐색 범위를 사용한 결과 보다 높은 성능을 보여주고 있다.

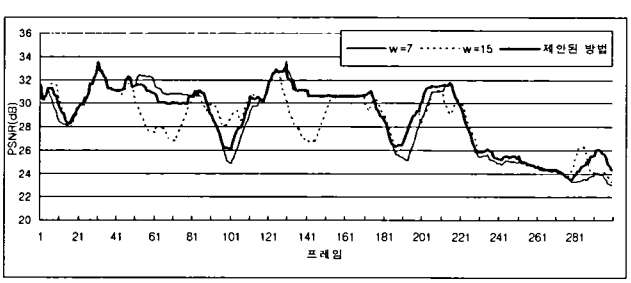

그림 3. 그림 2의 두 가지 결과에 대한 제안된 방법 비교

5. 결론

본 논문에서는 기존의 NTSS가 가지는 문제점을 개선한 적응형 윈도우 기반의 NTSS를 제안하였다. 이 알고리즘은 기존의 NTSS와 비교하였을 때, 움직임이 작은 특성을 갖는 영상에 대해서 PSNR 측면에서 유사한 결과를 보이면서 움직임이 큰 특성을 가진 영상에서는 0.27~0.58dB 증가하였다. 따라서 제안된 방법은 움직임 특성에 좌우되지 않는 화질을 제공할 수 있는 움직임 예측이다.

감사의 글

본 연구는 2005년도 한국소프트웨어진흥원 IT-SoC 핵심설계인력양성사업, 한국전자통신연구원 위탁과제(0201-2005-0080)로 수행되었음.

참고문헌

[1] A.Murat Tekalp,Digital Video Processing,Prentice Hall. p.101-106. 1995.

[2] T. Koga, K. Iilnuma, a Hirano, Y. Iijima, T. Ishiguro, "Motion-compensated interframe coding for video conferencing." in Proc. Nat. Telecommun. Conf., NewOrleans, LA. Nov. 29-Dec. 3 1981. P.G5.3.1-G5.3.5

[3] R. Li, B.Zeng, and M.L.Liou, "A New three-step search algorithm for block motion estimation," *IEEE Trans.CircuitsSyst.VideoTechnol.*,vol.4,pp.438-442,Aug.1994

[4] Liang-Wei Lee,Jhing-Fa Wang,Jau-Yien Lee, and Jung-Dar Shie "The Dynamic Search-Window Algorithm" *IEEE Trans. Circuit and System. Video Tech.* Vol 3. pp. 85-87. February 1993

[5] ITU-T, Video Coding for Low Bitrate mmunication, Draft Recommendation H.263+, July.1998.