

고속 움직임 탐색을 위한 효율적인 탐색영역 선택 알고리즘

*이원진 **정제창

한양대학교 전자컴퓨터통신공학과

*veronica0083@gmail.com **jjeong@ece.hanyang.ac.kr

The effective search range selection algorithm for fast motion estimation

*Wonjin, Lee **Jechang, Jeong

Dept. Electronics and Computer Eng. Hanyang University

요약

비디오 압축 기법에서 움직임 추정(Motion Estimation)은 매우 중요한 부분을 차지하는데, 그것은 움직임 추정이 화질과 인코딩 시간에 직접적으로 영향을 미치기 때문이다. 가장 기본적인 움직임 추정 기법은 전역 탐색 기법(Full Search)인데, 이는 가장 좋은 화질을 보여주기 하지만 매우 많은 계산량이 필요하다는 단점이 있다. 따라서 좋은 화질을 유지하면서도 계산량을 낮추기 위한 많은 고속 탐색 알고리즘들이 제안되었다. 이 논문에서는 현재 프레임의 매크로블록과 이전프레임의 매크로블록간의 Sum of Absolute Difference를 이용하여 탐색영역을 변경하는 새로운 예측 방법을 제시한다. 실험결과에 따르면 우리가 제안한 알고리즘은 FS와 비슷한 PSNR을 유지하면서 속도가 크게 향상된 것을 볼 수 있었다.

1. 서론

대부분의 비디오 압축 표준들은 프레임 간의 시간적 중복성을 줄이기 위해 블록 정합 알고리즘(Block Matching Algorithm)을 사용한다. 블록정합 알고리즘은 우선 각 프레임을 겹치지 않는 여러 개의 매크로블록으로 분할하고, 참조 프레임의 탐색영역 내에서 현재 매크로블록과 가장 유사한 블록(예측블록)을 찾는다. 현재 매크로블록의 정보를 모두 보내는 대신, 예측블록과의 차이와 예측블록의 상대적 위치를 나타내는 움직임 벡터만을 부호화한다. 하지만 일반적으로 영상 코덱에서 사용 되는 Full Search (FS) algorithm 은 최적의 정합블록을 찾기 위해 탐색 영역 내의 모든 위치에서 SAD 값을 비교하기 때문에 이러한 움직임 추정 및 보상 과정은 전체 부호화 시간에서 가장 많은 부분을 차지하게 된다. 이런 문제를 개선하기 위해, 다양한 고속 움직임 탐색 알고리즘이 개발되었다.

대표적으로 Partial Distortion Search (PDS), Normalized Partial Distortion Search (NPDS)^[1], Adjustable Partial Distortion Search (APDS)^[2] 등과 같이 적은 샘플들을 가지고 최적의 정합블록을 찾아내는 방법들이 개발되었다. 기존의 알고리즘은 FS와 비슷한 PSNR을 유지하면서 속도를 향상시킬 수 있었다. 우리가 제안한 알고리즘은 기존의 알고리즘의 PSNR을 유지시키면서 속도를 개선하기 위하여 현재 프레임의 매크로 블록과 이전 프레임의 매크로 블록 간 SAD를 이용하여 탐색영역을 변경하는 방법을 제시한다.

본 논문의 구성은 다음과 같다. 2장에서 기존의 알고리즘을 간단히 설명하고, 3장에서는 APDS를 기반으로 한 탐색영역 선택 알고리즘에 대하여 설명한다. 4장에서는 기존 기법들과 성능을 비교 평가한 실험결과를 보여주고 마지막으로 5장에서 결론을 맺는다.

2. 기존의 알고리즘

NPDS는 한번에 16×16매크로블록의 모든 픽셀을 계산하여 SAD를 구하는 대신, 하나의 매크로블록의 픽셀들을 그림 1과 같이 16개의 블록으로 나눈다. 그리고 각 블록에서 한 개의 픽셀씩, 16개의 픽셀을 추출하여 하나의 그룹으로 만들고, 현재 매크로 블록과 후보 매크로블록 간 SAD를 구하게 된다.

식(1)은 하나의 그룹에 대한 SAD를 구하는 식이다.

$$d(p)(k,l;u,v) = \sum_{i=0}^3 \sum_{j=0}^3 \left| I_n(k+4i+s(p), l+4j+t(p)) - I_{n-1}(k+4i+s(p)+u, l+4j+t(p)+v) \right| \quad (1)$$

여기서 p는 하나의 그룹을 뜻하고, 총 16개의 그룹에 대한 SAD를 구하게 된다. 위의 식에서 하나의 그룹에 대한 SAD, 즉 d(p)를 구할 때마다 누적시키면서 16×16매크로블록의 모든 픽셀을 계산하여 구한 SAD값과 비교를 하게 된다. 식(2)를 누적시키는 식이다.

$$D(p) = \sum_{i=1}^p d(i) \quad (2)$$

NPDS는 위의 식을 이용하여 16개의 픽셀에 대한 SAD를 누적시키면서 실제 SAD와 비교하게 되는데 빠른 탐색을 위해 D(p)에 16/p를 곱한 값을 실제 SAD와 비교한다. 16/p를 곱하는 이유는 D(p)는 16*p개의 픽셀에 대한 SAD이기 때문이다. 만약 16개 그룹의 SAD가 모두 비슷한 값을 가질 경우, 하나의 그룹의 SAD에 16을 곱하여도 실제 SAD를 계산한 값과 비슷하기 때문에 D(p)에 16/p를 곱하여 고속 탐색이 가능하게 한다. NPDS의 단점은 각 그룹의 SAD가 비슷하지 않을 경우, 잘못된 움직임 벡터를 찾을 확률이 높다.

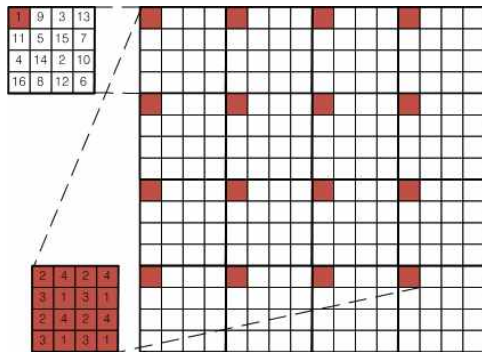


그림 1. partial distortion에 대한 픽셀 그룹 패턴
표 1. p에 대한 픽셀 좌표

p	s(p),t(p)	p	s(p),t(p)
1	(0,0)	9	(1,0)
2	(2,2)	10	(3,2)
3	(2,0)	11	(0,1)
4	(0,2)	12	(2,3)
5	(1,1)	13	(3,0)
6	(3,3)	14	(1,2)
7	(3,1)	15	(2,1)
8	(1,3)	16	(0,3)

그리고 하나의 그룹에서 16개의 픽셀을 계산하게 되므로 FS보다 최대 16배 이상 빨라 질 수 없기 때문에 다른 고속 탐색 알고리즘보다 성능이 떨어진다.

이러한 문제점들을 해결하기 위해서 APDS가 제안 되었다. APDS는 NPDS보다 속도를 향상시키기 위해서 맨 처음 그룹을 4개의 그룹으로 분할하였다. 이로써 FS보다 최대 64배까지 빨라 질수 있게 되었다. 그리고 NPDS처럼 16/p를 곱해서 화질을 떨어뜨리면서 속도를 올리는 대신, 원본 화질과 비슷하게 유지하면서 속도를 향상시킬수 있는 quality factor를 만들었다.

식(3)은 quality factor를 구하는 식이다.

$$f(n,k) = (1 - k)n + kN^2 \quad (3)$$

k는 0~1사이의 값을 가지는데 k가 1이면 PDS와 같은 성능을 가지게 되고, k가 0이면 NPDS와 같은 성능을 가지게 된다.

3. 제안하는 알고리즘

기존의 APDS 알고리즘은 우선 현재 매크로블록과 같은 위치에 있는 이전프레임의 매크로 블록에 대해서 SAD를 구한 뒤, 다음의 후보 블록들에 대해서는 적은 샘플들을 가지고 실제 SAD를 예측하여 탐색영역 전체를 이동하면서 수행하게 된다. 만약 첫 번째 후보의 SAD

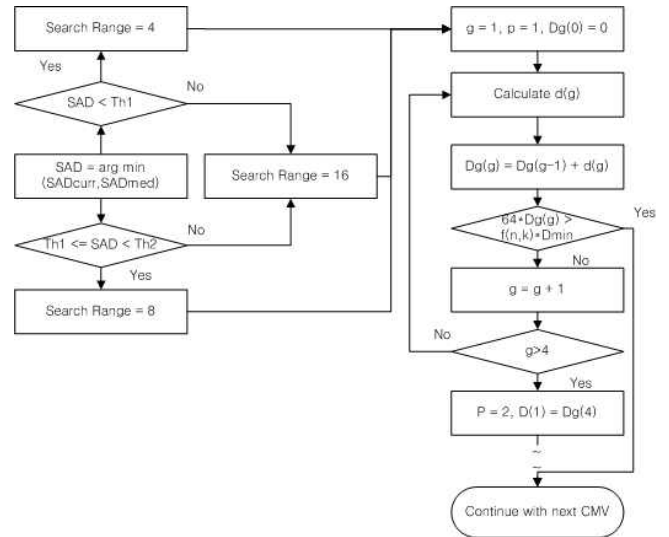


그림 2. 제안하는 알고리즘의 순서도

값이 충분히 작다면, 현재 매크로블록과 이전 매크로블록의 차이가 거의 없다는 의미이므로 실제 움직임 벡터가 근처에 있을 확률이 높다.

이 점에 착안하여, 우리는 전체 탐색영역을 수행하는 것이 아니라 일부 영역만을 탐색하는 새로운 알고리즘을 제안하였다. 그러나 이는 탐색영역의 일부만 검색을 수행하게 되므로 국부 최소점(local minimum point)에 빠지게 되어 부정확한 움직임 벡터를 찾게 될 수도 있다.

이러한 단점을 해결하기 위해서 우리는 탐색 시작점을 원점의 SAD와 현재 매크로블록 주변블록들의 움직임 벡터들을 이용한 중앙값의 SAD를 구한 뒤, 더 적은 값을 가지는 SAD의 위치를 시작점으로 설정하고, SAD값에 따라서 탐색영역의 크기를 조정한다.

위의 그림 2는 제안하는 알고리즘의 순서도이다. 오른쪽의 순서도는 APDS의 순서도와 동일하고, 왼쪽의 순서도는 APDS를 수행하기 전에 탐색영역을 설정하는 부분이다. 예를 들어 원점과 주변 움직임벡터들의 중앙값을 이용해 구한 SAD값이 각각 50, 100일 때, 원점의 SAD값이 작으므로 원점을 탐색 시작점으로 설정한다. 또한 SAD값이 50이므로 이는 현재 매크로 블록과 후보 매크로블록의 차이가 거의 없다고 볼 수 있다. 따라서 우리는 실제 움직임 벡터가 근처에 있다고 판단하여 탐색영역을 작게 설정한다. 만약 SAD값이 너무 크게 되면 실제 움직임 벡터가 근처에 없을 확률이 높기 때문에 탐색영역을 크게 설정한다.

표 2. 기존 방법들과의 평균 PSNR(dB) 비교

	PSNR						
	FS	PDS	NPDS	PPDS	APDS	PSA	PROPOSAL
akiyo	42.94	42.94	42.85	42.75	42.94	42.94	42.94
children	29.79	29.79	29.58	29.30	29.77	29.58	29.77
foreman	31.42	31.42	31.31	31.12	31.41	30.18	31.69
hall	34.83	34.83	34.70	34.59	34.80	34.78	34.79
mobile	24.43	24.43	24.28	24.01	24.42	24.52	24.57
mother	40.44	40.44	40.35	40.20	40.43	40.34	40.44
news	36.90	36.90	36.68	36.44	36.88	36.75	36.88
stefan	24.59	24.59	24.46	24.24	24.59	24.09	24.98
Avg	33.17	33.17	33.03	32.83	33.16	32.90	33.26

표 3. 기존 방법들과의 평균 속도 비교

	speed up						
	FS	PDS	NPDS	PPDS	APDS	PSA	PROPOSAL
akiyo	0.00	10.63	14.57	45.64	34.95	114.33	258.41
children	0.00	8.15	14.38	43.73	29.45	89.94	81.58
foreman	0.00	3.53	13.42	36.06	15.80	50.75	61.11
hall	0.00	3.52	14.14	39.67	16.09	43.41	106.55
mobile	0.00	4.19	14.31	41.97	23.73	76.83	38.52
mother	0.00	3.64	13.66	37.05	14.83	39.05	100.77
news	0.00	7.87	14.45	44.08	28.86	85.83	135.58
stefan	0.00	2.91	13.29	34.09	12.74	41.36	29.41
Avg	0.00	5.55	14.03	40.29	22.06	67.69	101.49

4. 실험 결과 및 분석

실험은 CIF(352×288)영상 “akiyo”, “children”, “foreman”, “hall”, “mobile”, “mother”, “news” and “stefan” 들에 대해 각각 300프레임을 실험하였고, 비교 탐색 알고리즘으로 Full Search(FS), NPDS, Progressive Partial Distortion(PPDS)^[2], APDS, Partial Distortion Search Algorithm Using Predictive Search Area(PSA)^[3], 그리고 제안된 알고리즘을 사용하였다. 움직임 예측에 사용한 매크로 블록의 크기는 16×16픽셀이며, 기본 탐색영역의 범위는 ±16 으로 설정하였고, PSA의 탐색영역의 범위는 ±11로 설정하였다. 그리고 제안하는 알고리즘의 탐색영역은 SAD가 Th1보다 작을때의 탐색영역은 4로, SAD가 Th1보다 크고 Th2보다 작을 때는 8, 그리고 SAD가 Th2보다 클 때는 기존의 탐색영역 크기인 16으로 설정하여 실험을 수행하였다.

이 실험에서 Th1은 1000, Th2는 2000으로 설정하였다.

성능 비교 평가 함수로는 영상 화질의 품질을 평가하기 위해 평균 제곱 오차(MSE: Mean Squared Error)를 이용한 PSNR을 이용하였으며, 제안하는 알고리즘의 속도 향상을 측정하기 위해 블록 당 계산량을 비교하였다.

제안하는 알고리즘은 표. 1, 2에서 볼 수 있듯이 대부분의 영상에 대해서 기존의 고속 탐색 알고리즘과 비슷한 수준의 PSNR을 유지하였고, 빠른 움직임이 있는 영상에 대해서는 FS보다 좀 더 높은 PSNR을 얻을 수 있었다. 그리고 속도면에서 NPDS보다 7배, PSA보다 1.5배, APDS보다 5배 이상 속도가 빨라진 것을 확인할 수 있다. 더욱이 배경화면이 정지된 영상에 대해서는 계산량이 월등히 줄어든 것을 볼 수 있었다.

5. 결 론

본 논문에서는 영상의 압축 효율을 높이는 움직임 추정기법에 있어서 기존에 제안된 블록 정합 방법만큼 좋은 화질을 유지하면서 보다 고속으로 전송하기 위한 효율적인 탐색영역 변경을 제안하였다.

제안한 탐색영역 변경 기법은 현재 매크로블록과 후보 매크로블록 간 SAD값이 작으면 실제 움직임 벡터가 근처에 있을 확률이 높다는 점을 고려하여 탐색영역을 조절할 수 있도록 하였다.

실험 결과에서 보면 알 수 있듯이 제안된 기법은 이전에 알려진 고속 탐색 기법과 비교하여 대부분 영상에서 비슷한 수준의 PSNR을 유지하면서 속도를 향상시킬 수 있었다. 그리고 만약 탐색 시작점을 좀 더 정확하게 결정하는 등의 연구가 이루어진다면 더욱 속도를 향상시킬 수 있을 것이다.

참 고 문 헌

- [1] C. K. Cheung and L. M. Po, "Normalized partial distortion search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 3, pp. 417 - 422, Apr. 2000.
- [2] C. H. Cheung and L. M. Po, "Adjustable partial distortion search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 1, pp. 100 - 110, Jan. 2003.
- [3] X. J. Jing and L. P. Chau, "Partial Distortion Search Algorithm Using Predictive Search Area for Fast Full-Search Motion Estimation," *IEEE Signal Processing Letters*, vol. 14, no. 11, pp. 840 - 843, Nov. 2007.