

# FGK 알고리즘을 이용한 VLC의 효율성 증대 방법

김정우

LG전자 Digital TV 연구소

## An Effective VLC method using FGK Algorithm

Jeongwoo Kim

Digital TV Research Lab., LG Electronics Inc.

E-mail : myimage@lge.com

### Abstract

본 논문에서는 FGK 알고리즘을 동영상 압축 표준 안에 적용하였을 때 발생하는 문제점들을 해결하기 위한 방법을 제안한다. 먼저, 계산량을 줄이기 위해 트리의 갱신 주기를 가변적으로 적용하고 심벌 당 발생한 가중치를 한번에 처리하기 위하여 FGK 알고리즘을 수정하여 사용한다. 또한, 확률분포가 현재의 영상 시퀀스를 잘 반영할 수 있도록 모집단의 수를 적절하게 유지하는 방법을 제시한다.

제안 방법을 동영상 시퀀스에 적용하였을 경우, 영상의 종류와 비트 율에 따라 다르지만 최대 0.33dB 정도의 PSNR 이득을 얻을 수 있었다.

### I. 서론

FGK 알고리즘은 동적 Huffman Coding으로도 불리며 정적 Huffman 알고리즘을 개선하기 위하여 Faller[1], Gallager[2], Knuth[3]에 의해 개발되고 개선되었다. 이 알고리즘은 매 심벌이 발생할 때마다 Huffman 조건을 만족하도록 트리를 수정하여 각 심벌에 대한 최적의 코드를 발생시킨다. 하지만, 트리의 수정을 위한 계산량이 증가하여 실시간 전송 시스템과 같은 응용에서는 큰 오

버헤드로 작용한다. 이를 극복하기 위하여 트리 자체의 갱신 대신에 심벌의 코드들을 교환하여 계산량을 줄이는 논문들[4,5]도 발표되어 있지만 심벌의 라플라시안 분포가 변화함에 따라 할당되는 코드의 길이 자체가 변화하는 경우에는 적합하지 않다. 그러므로, 본 논문에서는 영상의 특성과 파라미터들로부터 트리의 수정 시점을 조절하여 계산량을 줄일 수 있는 방법과 FGK 알고리즘을 수정하는 방법, 그리고 HDTV와 같이 무한히 반복되는 영상 시퀀스에 적합하도록 심벌의 모집단 수를 적절히 유지하는 방법을 제시한다.

2장에서는 FGK 알고리즘에 관하여 설명하고 3장에서는 제안한 방법에 관한 설명과 적합성 여부를 살펴보고, 4장에서는 실험결과에 대하여 고찰한다.

### II. FGK 알고리즘

임의 심벌들의 가중치 집합을 이용하여 만든 이진 트리가 Huffman 트리가 되기 위한 조건은 다음과 같다.

- (1)  $n$ 개의 외부 노드로 이루어진 가중치 집합을 일정한 순서로 놓았을 때, 각 내부 노드는 두 자식 노드 가중치의 합으로 이루어진다.
- (2)  $2n-1$ 개의 전체 노드를 나열한 집합을  $(y_1, y_2, \dots,$

$2_{n-1}$ 이라 할 때,  $y_{2j-1}$ 과  $y_{2j}$ 는 형제 노드이며, 이들의 부모 노드는  $y_{2j-1}$ 과  $y_{2j}$ 의 앞에 위치하지 않는다.

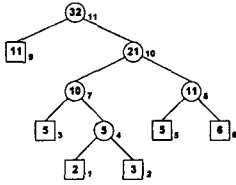


그림 1. Huffman 트리의 예

그림 1에서 2번 노드의 가중치가 1만큼 증가하여 4가 되면 부모 노드의 값이 6이 되어 Huffman 조건(2)를 만족하지 못하게 된다. FGK 알고리즘은 위와 같이 가중치가 변화하는 경우 트리를 수정하여 항상 Huffman 트리의 조건을 만족하도록 유지하는 방법이다. FGK 알고리즘을 간단하게 기술하면 다음과 같다.

- (1) 가중치를 증가하고자 하는 외부 노드에서 시작
- (2) 이 노드의 부모 노드와 같은 가중치를 갖는 노드들 중 다시 그 부모 노드가 최대치인 노드를 찾아 부모 노드들끼리 서로 교환
- (3) (2)의 과정을 루트에 이를 때까지 반복
- (4) 가중치를 증가

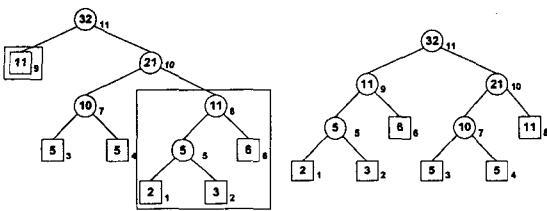


그림 2. FGK 알고리즘을 이용한 트리 갱신

FGK 알고리즘에 의해 각 노드들을 교환하는 과정을 그림 2에 나타내었으며 이제 2번 노드의 가중치가 4로 증가하여도 Huffman 트리의 조건을 만족한다.

### III. 제안한 FGK 알고리즘 적용 방법

표 1에는 FGK 알고리즘의 효율성을 설명하기 위하

여 MPEG-4 part.2[6]에서 제시한 I-VOP의 MCBPC에 관한 심벌과 코드 길이를 나타내었다. 또한, Foreman 영상(CIF, 100frames)을 이용하여 비트 율의 변화에 따른 심벌의 가중치와 FGK 알고리즘에 의해 할당된 코드 길이를 조사하여 함께 표시하였다.

표 1. MCBPC의 발생 빈도수와 코드 할당 수 비교

심벌	Code 수	64Kbps		512Kbps		1Mbps	
		가중치	Code수	가중치	Code수	가중치	Code수
0	1	2597	1	618	2	639	2
1	3	78	2	21	7	57	6
2	3	25	4	56	5	191	3
3	3	22	5	45	6	172	4
4	4	46	3	1497	1	771	2
5	6	1	7	123	4	123	5
6	6	0	8	186	4	420	3
7	6	3	6	226	4	399	3
8	9	0	8	0	7	0	6

표준 안에서는 비트 율에 관계없이 항상 심벌 0의 가중치를 가장 높게 보고 Huffman 알고리즘에 따라 1비트의 코드를 할당하였다. 하지만, 결과를 살펴보면 높은 비트 율에서는 심벌 4의 가중치가 가장 높아서 표준 안의 분포가 낮은 비트 율에 한정되어 있음을 의미한다. FGK 알고리즘은 현재 입력된 영상의 심벌 가중치를 이용하므로 비트 율에 상관없이 잘 표현하고 있음을 알 수 있다.

### 3.1 트리의 갱신주기 설정

각 심벌이 발생할 때마다 트리를 재구성하는 것은 계산량의 측면에서 상당히 비효율적이다. 만약, 일정한 주기 후에 한꺼번에 갱신하게 된다면 비록 다음 갱신 때까지 최근 심벌의 발생분포를 적용하지 못한다는 단점이 있지만 계산량의 측면에서는 상당한 이득이 있을 것이다.

본 논문에서는 계산량을 줄이기 위하여 VOP 주기마다 트리를 갱신하며 다음의 조건에 해당하는 경우에만 MB 단위의 트리 갱신을 시도한다.

- (1) 첫번째 I-VOP 혹은 P-VOP인 경우
- (2) MB별 QP(dquant)의 절대값이 최대값인 경우
- (3) 현재까지의 가중치 순서 중 1, 2위가 이전 MB의 순서와 다른 경우 (단,  $abs((N_{1st} - N_{2nd}) / N_{1st}) > \alpha$ )

### 3.2 수정된 FGK 알고리즘

FGK 알고리즘에서는 가중치가 항상  $w$ 에서  $w+1$ 로 변하는 경우만 고려하므로 비록 MB 혹은 VOP 단위로 갱신을 하더라도 심벌의 전체 발생 수만큼 갱신 동작을 하게 되어 계산량의 변화는 없다. 그러므로, 본 논문에서는 다음과 같이 누적된 가중치에 대하여 한번에 처리할 수 있는 알고리즘을 제안하여 사용하였다.

- (1) 가중치가 증가한 경우 외부 노드들의 순서를 재배치
- (2) 증가한 외부노드의 가중치가 자신의 직계 부모를 제외한 나머지 부모노드들보다 크면 교환
- (3) 증가한 외부노드의 부모노드 값이 그 상위 레벨의 내부노드 값보다 크면 교환
- (4) 교환된 부모노드의 하위 노드들에 대하여 (1)~(3)의 방법을 반복
- (5) 루트에 이를 때까지 (2), (3)을 반복

(1)의 경우는 Huffman 트리의 조건 (1)을 만족하기 위한 방법이며, (2)와 (3)은 조건 (2)를 만족시키기 위하여 재귀적으로 적용하게 된다.

그림 3에서는 그림 1의 Huffman 트리에서 2번 노드의 가중치가 3에서 70으로 증가한 경우의 예이다.

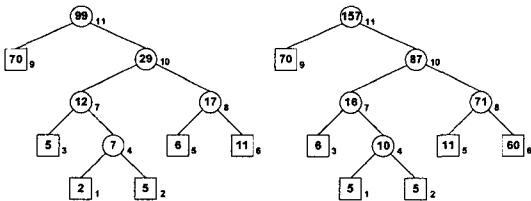


그림 3. 수정된 FGK 알고리즘의 예

그림의 두 번째 트리는 2번 노드의 가중치가 다시 60으로 변화한 경우의 예이며, 노드 6번과 7번이 제안

조건 (2)에 의해 다시 교환된다.

표 2에서는 제안한 방법의 효율성을 실험하기 위하여 MB, VOP 단위로 FGK 알고리즘을 적용한 경우와 본 논문에서 제안한 방법을 비교하여 나타내었다.

표 2. 제안한 MB 단위의 갱신

항목	VOP	MB	제안한 방법	
			$\alpha = 0.0$	$\alpha = 0.1$
PSNR(dB)	32.60	32.84	32.79	32.79
계산MB수	×	396	277 <sup>1</sup>	101 <sup>1</sup>

<sup>1</sup> VLC for Transform Coefficients

Conditions : Foreman CIF, 100 frames, 512Kbps

제안한 방법이 MB 단위의 갱신에 비하여 PSNR은 거의 차이가 없으면서도 전체적인 계산량은  $\alpha=0.1$ 의 경우 약 1/4로 줄일 수 있음을 보여준다.

### 3.3 FIFO를 이용한 가중치 유지방법

가중치 누적을 위한 VOP의 개수가 무한대가 되면 입력되는 영상의 특성을 충분히 이용하고자 하는 취지에 어긋나게 되며 반대로 적은 양의 심벌 수는 확률적으로 우세한 심벌의 출현 기회를 줄일 가능성이 있다. 심벌의 라플라시안 분포가 뾰족하고 값이 클수록 (leptokurtic) 확률이 높은 심벌에 보다 더 짧은 비트가 할당되며 비슷한 확률이 많은 반대의 경우(platykurtic)에서는 비트수가 여러 심벌에 분산되어 할당된다. 결국, 적당한 수의 최근 VOP에서 발생한 심벌의 가중치를 이용하여 영상의 특성도 적용하며 우세한 심벌의 분포 또한 충분히 발생될 수 있도록 만드는 것이 필요하다. 이를 위하여 본 논문에서는 각 심벌의 가중치를 VOP별로 저장할 수 있는 FIFO를 구성하였다. 고정된 FIFO의 수는 급격하게 변화하는 영상의 특성을 반영하지 못하므로 초기에 5개의 버퍼에서 시작하여 QP의 변화에 따라 다음과 같이 FIFO의 수를 능동적으로 변화시키는 방법을 사용하였다.

$$FIFO수 = f(15 - |QP_{current} - QP_{prev}|)$$

단,  $f(\cdot)$ 는 [1,15] Clip 함수이다.

표 3. FIFO 수의 변화에 의한 PSNR

Unit	Category		
	표준안	VOP	제안방법
PSNR(dB)	29.78	29.84	30.09

Conditions : Foreman CIF size, 100frames, 256Kbps

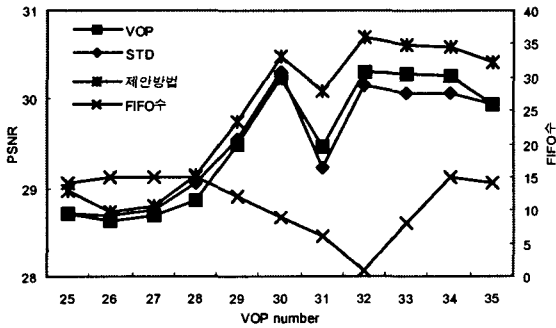


그림 4. FIFO 수의 변화에 의한 PSNR 분포

그림 4를 살펴보면 약 27프레임정도에서부터 시작하여 QP의 변화에 따라 FIFO 수를 줄임으로써 31 프레임 정도에서 나타난 급격한 변화에도 충분히 대처할 수 있음을 알 수 있다.

#### IV. 실험 결과 및 고찰

본 논문에서는 Class A, B, C에 해당하는 3개의 CIF 시퀀스, Akiyo, Foreman, Mobile&Calendar 150프레임을 사용하여 제안한 방법을 실험하였다. 실험에서 사용된 VLC 테이블은 Intra / Inter MCBPC, Intra / Inter CBPY, Intra / Inter TCOEF, MVD로써 MPEG-4 simple 프로파일에서 사용된 테이블 중 Huffman 알고리즘을 사용하지 않았거나 지나치게 심벌의 수가 적은 경우는 제외하였다.

각 영상의 복잡도로 인하여 비트 율을 만족하는 최소단위에서부터 실험을 하여 결과를 표 4에 나타내었다. 앞에서 설명되었던 것처럼 낮은 비트 율에서는 차이가 무시할만하며 비트 율이 높아질수록 제안한 방법의 결과가 점점 나아진다. 이는 표준 안의 VLC 테이블이 낮은 비트 율에서 최적화된 결과로 생각된다.

표 4. 각 영상별 평균 PSNR(dB),  $\alpha = 0.1$

Sequence	Method	Bit rate(bps)		
		40K/ 128K/ 256K	64K/ 192K/ 512K	128K/ 256K/ 1M
Akiyo	표준안	31.13	31.24	34.84
	제안한 방법	31.14	31.35	35.09
	PSNR Gain	0.01	0.11	0.25
Foreman	표준안	28.63	29.09	30.01
	제안한 방법	28.64	29.30	30.34
	PSNR Gain	0.01	0.21	0.33
Calendar	표준안	21.90	22.12	24.00
	제안한 방법	21.90	22.25	24.21
	PSNR Gain	0.00	0.13	0.21

본 논문에서는 FGK 알고리즘을 동영상 압축에 적용하여 압축 효율을 높일 수 있는 방법을 제시하였다. 결과를 살펴보면, 영상에 따라 차이는 있지만 제안한 방법이 MPEG-4 표준 안에 비하여 높은 비트 율에서 최대 0.33dB 정도의 PSNR 이득을 얻을 수 있었다.

#### V. References

- [1] N. Faller, "An adaptive system for data compression," Record of the 7<sup>th</sup> Asilomar Conference on Circuits, Systems, and Computers, pp.593-597, IEEE Press, 1973
- [2] R.G. Gallager, "Variations on a theme by Huffman," IEEE Trans. on Information Theory, pp.668-674, Vol. IT-24, No.6, Nov. 1978
- [3] D.E. Knuth, "Dynamic Huffman coding," Journal of Algorithms 6, pp.163-180, 1985
- [4] B. Jeon, J. Park, J. Jeong, "Huffman coding of DCT coefficients using dynamic codeword assignment and adaptive codebook selection," Signal Processing: Image Communication 12 (1998) 253-262
- [5] Y. Okada, N. Sato, K. Murashita, S. Yoshida, "Self-organized dynamic Huffman coding without frequency counts," Data Compression Conference '95, Proceedings, pp.473, Mar. 1995
- [6] ISO/IEC, MPEG-4 Video VM 16.0, N3312, Mar. 2000