

일반논문-11-16-4-06

FIR필터와 선형필터를 이용한 색차 보간법

김 정 필^{a)}, 이 영 렬^{a)‡}

Chroma Interpolation using FIR Filter and Linear Filter

Jeong-Pil Kim^{a)} and Yung-Lyul Lee^{a)‡}

요 약

최근 차세대 비디오 코딩 표준화를 위해 JCT-VC에서 HEVC라 불리는 새로운 비디오 압축 표준 기술을 개발하고 있다. HEVC는 H.264/AVC보다 높은 성능을 갖는 많은 부호화 기술을 채택하였다. 그중 색차 신호를 보간할 때 H.264/AVC에서 사용된 선형필터보다 좋은 성능을 가지는 DCT 기반으로 한 보간 필터를 사용하고 있다. 본 논문에서는 H.264/AVC에서 사용된 FIR필터와 선형필터를 통합한 필터를 제안하여 부호화 효율을 높이는 방법을 제안한다. 제안하는 방법과 DCT 기반으로 한 보간 필터와 비교하였을 때 실험 결과로 제안한 방법은 random access구조의 high efficiency 경우 색차성분 U,V에서 각각 평균 0.9%, 1.1%의 BD-rate가 감소하였고, random access의 low complexity 구조인 경우 색차성분 U,V에서 각각 평균 1.1%, 1.1%의 BD-rate가 감소하였고, low delay의 high efficiency 구조인 경우 색차성분 U,V에서 각각 평균 0.9%, 1.4%의 BD-rate가 감소하였고, low delay 구조의 low complexity인 경우 색차성분 U,V에서 각각 평균 1.8%, 1.8%의 BD-rate가 감소하였다.

Abstract

Recently, the JCT-VC is developing the next generation video coding standard that is called HEVC. HEVC has adopted many coding technologies increasing coding efficiency. For chroma interpolation, DCT-based interpolation filter showing better performance than the linear filter in H.264/AVC was adopted in HEVC. In this paper, a combined filter that utilizes the FIR filter and the linear filter in H.264/AVC is proposed to increase coding efficiency. When the proposed method is compared with DCT-based interpolation filter, the experimental results for various sequences show that the average BD-rate improvements on chroma U and V components are 0.9% and 1.1%, respectively, in the high efficiency case of random access structure, those on U and V components are 1.1% and 1.1%, respectively, in the low complexity case of random access structure, those on U and V components are 0.9% and 1.4%, respectively, in the high efficiency case of low delay structure, and those on U and V components are 1.8% and 1.8%, respectively, in the low complexity case of low delay structure.

Keyword : HEVC, H.264/AVC, Interpolation, FIR filter, Linear filter

1. 서 론

a) 세종대학교 컴퓨터공학과 DMS 연구실
DMS Lab. Dept. of ComputerEngineering, Sejong University
‡ 교신저자 : 이영렬(ylylee@sejong.ac.kr)
※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2011-0002990).
· 접수일(2011년3월17일), 수정일(2011년5월23일), 게재확정일(2011년6월1일)

차세대 비디오 압축 표준 기술인 HEVC(High Efficiency Video Coding)^[1]의 개발을 위하여 ITU-T VCEG(Video Coding Experts Group)과 ISO/IEC MPEG(Moving Picture Experts Group)에서 JCT-VC(Joint Collaborative Team on

Video Coding)을 결성하여 표준화 회의중에 있다. 이번 표준화 회의에서는 기존 동영상 압축 표준인 H.264/AVC (Advanced Video Coding)^{[2],[3]}보다 2배정도 향상된 성능을 목표로 많은 부호화 기술이 기고되고 있다. 현재까지 H.264/AVC에서 사용된 최대 16×16에서 확장된 64×64 블록 구조를 가지며 인트라 모드에서 확장된 방향성을 가지는 AIP(Angular intra prediction)^[4], 개선된 움직임 벡터 예측방법^[5], 향상된 디블리킹 필터^[6], 8bit 영상을 코덱 내에서 10bit로 증가시켜 코딩되는 IBDI(Internal Bit-Depth Increase)^{[7],[8],[9]} 기술등 향상된 부호화 기술이 채택되었다. 또한 H.264/AVC에서 사용된 부호화 기술도 채택 되었는데 그중 휘도 신호에서 움직임 벡터를 정수화소 사이의 1/4 서브샘플 위치까지 보간하여 움직임 벡터를 보다 정밀하게 찾는 부호화 기술도 채택 되었다. 4:2:0인 컬러 포맷일 경우 색차 신호의 정수 화소는 휘도신호의 1/2배이다. 그러므로 색차 신호의 서브샘플은 휘도 신호의 1/4 서브샘플의 1/2배인 1/8 서브샘플 위치까지 보간해야된다. 이때 사용되는 보간법으로 DCT-IF(DCT based Interpolation Filter)^[10]가 채택되어 H.264/AVC에서 사용된 선형필터와 비교하였을 때 많은 성능 향상을 가져왔다. 차세대 비디오 압축 표준인 만큼 더 높은 성능 향상을 위하여 본 논문에서는 FIR (Finite Impulse Response)필터와 선형필터를 통합한 필터 계수를 사용하여 부호화 성능을 높이는 방법을 제안한다.

제 2장에서는 현재 HEVC에서 사용중인 DCT-IF에 대해

설명하고, 제 3장에서는 제안하는 방법에 대해 설명하고 제 4장에서는 HEVC의 참조 소프트웨어인 HM2.0(HEVC Test Model)^[11]의 DCT-IF를 기준으로 제안된 방법과 선형 필터, 방향성 필터의 성능 결과를 비교하고, 제 5장에서 결론을 맺는다.

II. DCT-IF를 이용한 색차보간법

현재 HEVC에서 색차신호에 대한 서브샘플 값을 보간할 때 DCT-IF를 이용하여 보간한다. DCT-IF는 DCT를 기반으로 여러 테스트영상을 통해 추출한 최적의 필터 계수값을 사용하고 있다. 채택된 필터 계수는 표 1과 같다. 표 1의 α 는 보간할 서브샘플 위치를 나타낸다.

표 1. 각 서브 샘플 위치의 DCT-IF 계수

Table 1. DCT-IF coefficients at each sub-sample position

α	$filter_m(\alpha)$
1/8	{ -3, 60, 8, -1}
1/4	{ -4, 54, 16, -2}
3/8	{ -5, 46, 27, -4}
1/2	{ -4, 36, 36, -4}
5/8	{ -4, 27, 46, -5}
3/4	{ -2, 16, 54, -4}
7/8	{ -1, 8, 60, -3}

그림 1의 알파벳 대문자(A, B, C, D, E, F, G, H, I, J, M, N, O, P)

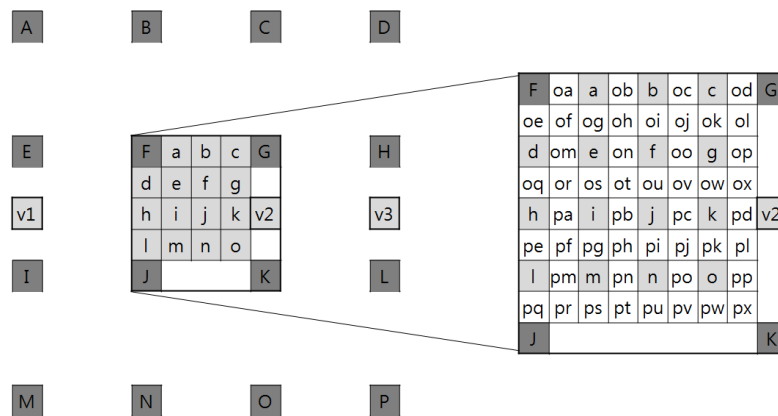


그림 1. 1/8 서브샘플 위치

Fig. 1. 1/8 sub-sample positions

K, L, M, N, O, P)는 정수화소 값을 나타낸다. 그림 1의 알파벳 소문자 (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, v1, v2, v3)는 1/2, 1/4 화소정밀도의 서브샘플 값을 나타낸다. 그림 1의 나머지 알파벳 (oa, ob, oc, od, oe, of, og, oh, oi, oj, ok, ol, om, on, oo, op, oq, or, os, ot, ou, ov, ow, ox, pa, pb, pc, pd, pe, pf, pg, ph, pi, pj, pk, pl, pm, pn, po, pp, pq, pr, ps, pt, pu, pv, pw, px)은 1/8화소정밀도의 서브샘플 값을 나타낸다.

DCT-IF를 이용한 색차 보간 방법은 다음과 같다. 정수화소 사이에 수평으로 위치한 서브샘플(oa, a, ob, b, oc, c, od)은 표 1의 계수 값을 이용하여 수평으로 필터를 적용하여 보간한다. 정수화소 사이에 수직으로 위치한 서브샘플(oe, d, oq, h, pe, l, pq)은 수직으로 필터를 적용하여 보간한다. 나머지 정수화소 안에 위치한 서브샘플은 수직으로 필터를 적용하여 4개의 서브샘플 값을 보간하고, 그 4개의 서브샘플 값을 수평으로 필터를 적용하여 보간한다. 수평, 수직으로 한번 적용되는 필터식은 다음 식 1과 같다.

$$X = \left(\sum_{m=-1}^2 Ref_m \times filter_m(\alpha) + 32 \right) \div 64 \quad (1)$$

식 1에서 Ref_m 은 정수화소 값을 나타내고, $filter_m$ 은 표 1의 필터 계수를 나타내고, α 는 서브샘플 위치를 나타내고, X 는 보간 된 값을 나타낸다.

예를 들어 정수화소 사이에 수평으로 위치한 1/8 서브샘플 a는 다음 식 2와 같이 한 번의 연산을 통해 구해진다.

$$a = (-3 \times E + 60 \times F + 8 \times G - 1 \times H + 32) \div 64 \quad (2)$$

정수화소 안에 위치한 서브샘플은 식 1을 통해 수직으로 필터를 적용해 4개의 서브샘플 값을 보간하고, 보간 된 4개의 서브샘플 값을 수평으로 필터를 적용해 구한다.

예를 들어 정수화소 내에 위치한 서브샘플 j는 다음 식 3과 같이 구한다.

$$\begin{aligned} v1' &= (-4 \times A + 36 \times E + 36 \times I - 4 \times M) \\ h' &= (-4 \times B + 36 \times F + 36 \times J - 4 \times N) \\ v2' &= (-4 \times C + 36 \times G + 36 \times K - 4 \times O) \\ v3' &= (-4 \times D + 36 \times H + 36 \times L - 4 \times P) \end{aligned} \quad (3)$$

$$j = (-4 \times v1' + 36 \times h' + 36 \times v2' - 4 \times v3' + 2048) \div 4096$$

III. 제안하는 방법

1. 제안한 방법을 이용한 색차보간법

본 논문에서 제안하는 방법은 기존 H.264/AVC에서 사용된 FIR필터 계수의 low-pass filter의 특징을 유지하면서 각 서브샘플위치의 거리에 따라 선형 필터를 통합하므로 분리되어 있을 때 보다 계산량과 나눗셈연산으로 인한 에러 값을 감소시키도록 생성하였다. 그림 2는 제안된 필터계수가 low-pass filter임을 보여주고, 표2는 제안된 방법의 필터계수를 나타낸다.

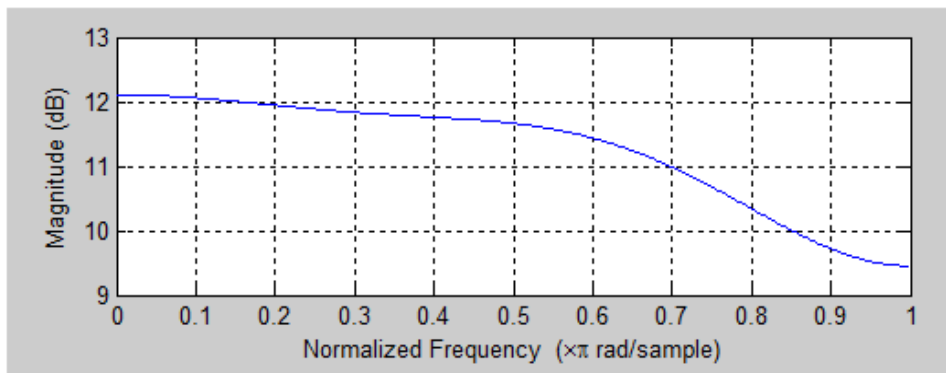


그림 2. 주파수 영역에서의 1/8 서브샘플 필터 계수
Fig. 2. 1/8 sub-sample filter coefficients at frequency domain

표 2. 제안된 방법의 서브샘플 위치에서 필터 계수
Table 2. Filter coefficients at each sub-sample position of the proposed method

α	$filter_m(\alpha)$
1/8	{1, -5, 116, 20, -5, 1}
1/4	{2, -10, 104, 40, -10, 2}
3/8	{3, -15, 92, 60, -15, 3}
1/2	{4, -20, 80, 80, -20, 4}
5/8	{3, -15, 60, 92, -15, 3}
3/4	{2, -10, 40, 104, -10, 2}
7/8	{1, -5, 20, 116, -5, 1}

FIR 필터와 선형필터를 통합하는 방법은 다음과 같다. 우선 제안하는 방법의 동기가 된 H.264/AVC의 휘도 신호의 보간 방법은 다음과 같다. H.264/AVC에서 휘도신호는 1/4 서브샘플 위치까지 보간한다. 첫 번째 단계로 1/2 서브샘플은 FIR 필터 계수값 (1, -5, 20, 20, -5, 1)을 사용하여 보간하고, 두 번째 단계로 1/4 서브샘플은 정수화소와 보간된 1/2 서브샘플간의 평균값을 취하는 선형필터를 이용하여

보간한다. 제안하는 방법은 1/4 서브샘플을 보간할 때 1/2 서브샘플의 계산과정과 선형필터의 계산과정을 통합하여 1/4 서브샘플 위치에 독립적인 필터계수를 생성하였다. 우선 1/2 서브샘플의 필터계수는 H.264/AVC에서 휘도 신호에 대한 FIR 필터계수와 동일한 값을 사용하였고, 예를 들어 1/4 서브샘플 위치인 a와 c를 다음 식 4에서 설명하였다.

$$\begin{aligned}
 b &= (E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J) \div 32 \\
 b' &= E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J \\
 a &= (32 \times G + b') \div 64 \\
 a &= (32 \times G + (E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J)) \div 64 \quad (4) \\
 a &= (E - 5F + 52G + 20H - 5I + J) \div 64 \\
 c &= ((E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J) + 32 \times H) \div 64 \\
 c &= (E - 5F + 20G + 52H - 5I + J) \div 64
 \end{aligned}$$

a의 필터 계수를 만들기 위하여 b를 32로 나누지 않은 b'를 이용하고, 정수 화소 G에 32를 곱함으로써 나눗셈 에러를 줄였다. 1/8 서브샘플 위치인 oa, ob, oc, od 또한 앞서

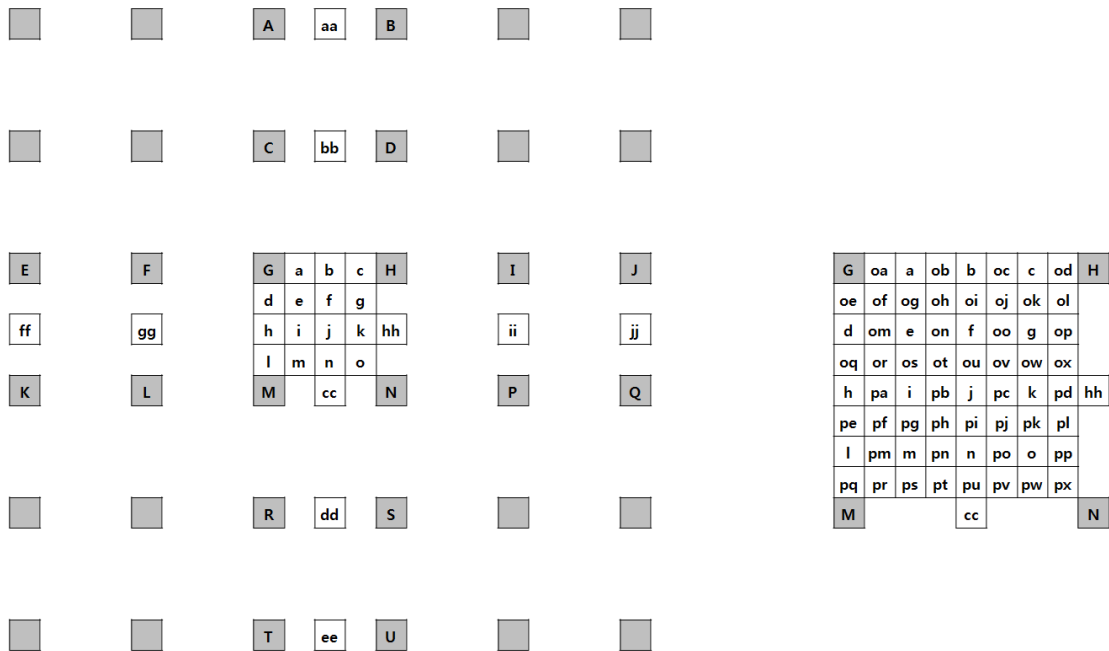


그림 3. 1/8 서브샘플 위치
Fig. 3. 1/8 sub-sample positions

$$\begin{aligned}
 oa &= (3 \times 32 \times G + b') \div 128 \\
 oa &= (3 \times 32 \times G + (E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J)) \div 128 \\
 oa &= (E - 5F + 116G + 20H - 5I + J) \div 128 \\
 ob &= (32 \times G + 3 \times b') \div 128 \\
 ob &= (32 \times G + 3 \times (E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J)) \div 128 \\
 ob &= (3E - 15F + 92G + 60H - 15I + 3J) \div 128
 \end{aligned}
 \tag{5}$$

1/4 서브샘플 필터계수 생성 방법과 동일하게 1/2 서브샘플 보간 과정과 정수화소와 보간 된 1/2 서브샘플간의 거리에 따른 가중치를 적용한 선형필터 계산과정을 통합하여 생성하였다.

예를 들어 서브샘플 oa와 ob는 다음 식 5와 동일하게 생성한다.

1/4 서브샘플 위치 필터 계수에 2를 곱하고 1/2 서브샘플 위치 필터 계수에 4를 곱하여 7개의 필터 비트수를 동일하게 하였다.

$$\begin{aligned}
 oa &= (E - 5F + 116G + 20H - 5I + J) \div 128 \\
 a &= (2E - 10F + 104G + 40H - 10I + 2J) \div 128 \\
 b &= (4E - 20F + 80G + 80H - 20I + 4J) \div 128
 \end{aligned}
 \tag{6}$$

제안된 방법의 필터계수를 이용한 색차 보간 방법은 이전 장에 설명된 DCT-IF필터를 이용한 방법과 동일하다. 정수화소 사이에 수평으로 위치한 서브샘플은 수평으로 필터를 적용해 보간하고, 수직으로 위치한 서브샘플은 수직으로 필터를 적용한다. 나머지 정수화소 안에 위치한 서브샘플은 수직으로 6개의 서브샘플 값을 우선 보간하고, 그 6개의 서브샘플 값을 수평으로 필터를 적용하여 보간한다.

2. 계산 복잡도

필터를 적용할 때 곱셈과 나눗셈 연산은 비트 시프트 연산 (\ll, \gg)을 통해 처리 속도를 향상 시킬 수 있다.

예를 들어 DCT-IF 계산식인 수학식 2를 비트 시프트 연산으로 다음 식 7처럼 바꿀 수 있다.

$$a = ((F \ll 6) - ((F + E) \ll 2) + E + (G \ll 3) - H + 32) \gg 6 \tag{7}$$

예를 들어 제안된 방법의 필터 계산식인 수학식 5의 oa는 다음 식 8처럼 바꿀 수 있다.

$$\begin{aligned}
 x &= F + I \\
 oa &= (E + J - x - ((x - (G + H)) \ll 2) - ((G - H) \ll 4) + (G \ll 7) + 64) \gg 7
 \end{aligned}
 \tag{8}$$

다음 표 3은 DCT-IF와 제안된 방법의 필터 계수를 비트 시프트 연산으로 수행하였을 경우의 계산 복잡도를 나타낸다.

표 3. DCT-IF와 제안된 방법의 각 서브픽셀 위치에서 계산 복잡도
Table 3. computational complexity of DCT-IF and the proposed method at each sub-sample position

	DCT-IF		Proposed method	
	\ll	\pm	\ll	\pm
1/8	3	5	3	9
1/4	5	5	4	9
3/8	4	7	5	10
1/2	2	4	3	7
5/8	4	7	5	10
3/4	5	5	4	9
7/8	3	5	3	9

표 3에서 반올림 연산과 나눗셈 연산은 중복되는 연산이기 때문에 제외하였다. 제안된 방법은 기존 4개의 정수화소를 참조하는 DCT-IF보다 더 많은 정수화소를 참조하기 때문에 약간의 계산 복잡도가 증가하였지만, 향상된 성능 결과를 가져왔다.

IV. 실험결과

1. 실험조건

제안된 방법은 HEVC의 참조 소프트웨어인 HM2.0에 구현하였고, 실험조건은 JCTVC-D600^[12]과 동일하게 하였다.

Random access 구조는 GOP(Group of pictures)크기가 8

인 계층적(hierarchical) B픽처(IBBBBBBBP)를 갖고 1초마다 I픽처(IDR, Instantaneous Decoder Refresh)를 넣는 구조이며, low delay 구조는 첫 장면이 I픽처를 갖는 IPPPPPPP... 구조이다. 여기서 P픽처는 GPB(Generalized P and B picture)이다. 표 4는 High efficiency 와 low complexity에서 사용되는 부호화 기술에 대한 표이다. 실험 영상으로는 HEVC 실험영상을 사용하였고, 표 5와 같다^[13]. 표 5에서 fps는 frames per second의 약자이다.

2. 실험결과

제안된 방법과 비교하기 위하여 H.264/AVC에서 사용된 선형필터와 방향성 필터^{[14][15]}를 사용하였다.

선형필터는 4개의 정수화소와 서브샘플간의 거리에 반 비례하는 가중치 값을 곱함으로써 보간하는 방법이다.

$$x = [(8 - d_x)(8 - d_y)G + d_x(8 - d_y)H + (8 - d_x)d_yM + d_xd_yN] \div 64 \quad (9)$$

표 4. High efficiency와 low complexity 방식의 코딩 방법 비교
Table 4. Comparison of coding tools in high efficiency and low complexity modes

High Efficiency	Low Complexity
Coding Unit 8x8 up to 64x64 in tree structure	
Prediction Units	
Transform unit tree(3 level max)	Transform unit tree(2 level max)
Transform block size of 4x4 to 32x32 samples	
Angular Intra Prediction (Max. 34 directions)	
Luma: 8-tap separable DCT-based interpolation filter with 6 bit precision Chroma: 4-tap separable DCT-based interpolation filter with 6 bit precision	
Advanced motion vector prediction	
CABAC entropy coding	LCEC Phase 2
Internal bit-depth increase (2 bits)	X
Deblocking filter	

표 5. HEVC의 실험 영상
Table 5. Test sequences of HEVC

Class	Sequence name	Frame size	Frame rate	Random access	Low delay
A	Traffic	2560x1600	30fps	O	
A	PeopleOnStreet	2560x1600	30fps	O	
B	Kimono	1920x1080	24fps	O	O
B	ParkScene	1920x1080	24fps	O	O
B	Cactus	1920x1080	50fps	O	O
B	BasketballDrive	1920x1080	50fps	O	O
B	BQTerrace	1920x1080	60fps	O	O
C	BasketballDrill	832x480	50fps	O	O
C	BQMall	832x480	60fps	O	O
C	PartyScene	832x480	50fps	O	O
C	RaceHorses	832x480	30fps	O	O
D	BasketballPass	416x240	50fps	O	O
D	BQSquare	416x240	60fps	O	O
D	BlowingBubbles	416x240	50fps	O	O
D	RaceHorses	416x240	30fps	O	O
E	Vidyo1	1280x720	60fps		O
E	Vidyo3	1280x720	60fps		O
E	Vidyo4	1280x720	60fps		O

G, H, M, N은 그림 3의 정수화소를 나타내고 x 는 G, H, M, N 안에 있는 보간 하려는 서브샘플을 나타낸다. d_x 는 x 축으로 정수화소와 서브샘플 간의 거리를 나타내고, d_y 는 y 축으로 정수화소와 서브샘플 간의 거리를 나타낸다.

방향성 필터는 서브샘플 위치에 따라 방향성을 고려하여 필터를 적용하는 방법이다. 그림 4에서 서브샘플 oa, a, ob, b, oc, c, od는 정수화소 E, F, G, H, I, J를 참조하여 수평으로 필터를 적용하고, 서브샘플 oe, d, oq, h, pe, l, pq는 정수화소 A, C, G, M, R, T를 참조하여 수직으로 필터를 적용하고, 서브샘플 of, e, ot, pj, o, px와 ol, g, ov, ph, m, pr은 각각 정수화소 AA, CC, G, N, FF, HH와 BB, DD, H, M, EE, GG를 참조하여 대각선으로 필터를 적용하여 보간한다. 서브샘플 j는 정수화소 C, D, F, G, H, I, L, M, N, P, R, S를 참조하여 보간한다. 나머지 서브샘플에 대해서는 복잡도 문제로 이전에 설명한 선형필터를 사용하여 보간한다. 필터계수는 참조문서 [15]를 바탕으로 표 6과 같다.

표 6. 방향성 필터계수
Table 6. Directional filter coefficients

α	$filter_m(\alpha)$
1/8	{5, -22, 247, 35, -13, 4}
1/4	{6, -30, 222, 74, -20, 4}
3/8	{11, -43, 196, 118, -36, 10}
1/2	{6, -34, 156, 156, -34, 6}
5/8	{10, -36, 118, 196, -43, 11}
3/4	{4, -20, 74, 222, -30, 6}
7/8	{4, -13, 35, 247, -22, 5}
j	{5, 5, 5, 22, 22, 5, 5, 22, 22, 5, 5, 5}

표 7, 8, 9, 10에 나타난 실험결과는 DCT-IF를 기준으로 제안하는 필터와 방향성 필터, 선형필터의 BD-rate이고, BD-rate는 VCEG-M33^[16]에 정의 되어 있으며, 울-왜곡 곡선 상에서의 평균 비트율 증가량을 나타낸다. 따라서 BD-rate가 음수일 때 압축률이 향상되었다고 평가할 수 있다. 실험 결과로 제안된 방법은 DCT-IF와 비교하였을 때

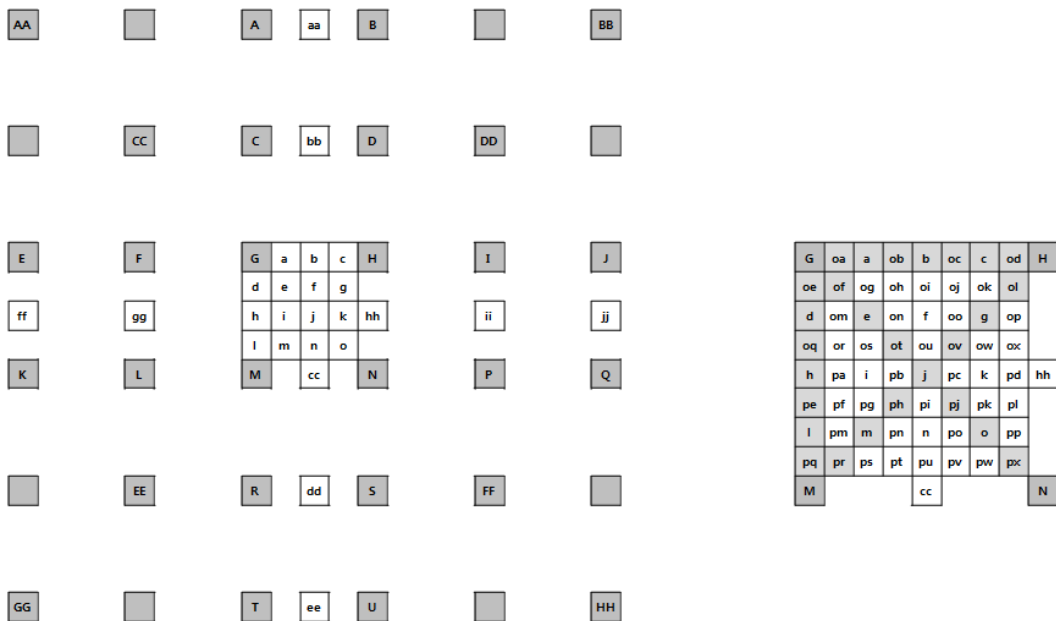


그림 4. 1/8 서브샘플 위치
Fig. 4. 1/8 sub-sample positions

표 7. Random access (high efficiency) 구조에서 실험결과
 Table 7. Experimental results in random access structure (high efficiency)

	Proposed method			Directional filter			Linear filter		
	Y	U	V	Y	U	V	Y	U	V
Traffic	0.0	0.2	-0.3	0.1	1.2	1.7	0.1	1.8	3.3
PeopleOnStreet	0.0	-0.1	-0.2	0.0	1.3	1.0	0.0	1.9	1.5
Average	0.0	0.1	-0.3	0.1	1.2	1.4	0.1	1.8	2.4
Kimono	0.0	-0.1	-0.4	0.0	0.4	1.3	0.0	0.7	2.3
ParkScene	0.0	-0.8	-1.3	0.1	1.8	2.3	0.1	3.1	4.2
Cactus	0.0	-0.5	-0.8	0.1	1.8	3.1	0.2	3.7	6.5
BasketballDrive	-0.1	-1.0	-1.5	0.1	1.8	2.0	0.1	4.0	4.6
BQTerrace	-0.1	-1.5	-1.9	0.0	3.9	2.9	0.1	6.2	6.1
Average	0.0	-0.8	-1.2	0.1	1.9	2.3	0.1	3.5	4.8
BasketballDrill	-0.1	0.0	-0.3	0.0	0.6	1.2	0.1	2.5	2.7
BQMall	-0.1	-0.6	-0.5	0.0	0.6	1.1	0.1	2.8	3.3
PartyScene	-0.1	-1.8	-1.6	0.3	2.6	3.2	0.3	5.5	5.5
RaceHorses	0.0	-1.2	-1.2	0.1	2.4	2.2	0.0	4.0	3.5
Average	-0.1	-0.9	-0.9	0.1	1.6	1.9	0.1	3.7	3.7
BasketballPass	0.0	-0.7	-1.3	0.1	2.3	3.4	0.2	4.6	6.5
BQSquare	0.0	-2.3	-2.9	0.1	4.6	8.2	0.2	5.3	9.2
BlowingBubbles	-0.1	-1.4	-1.3	0.3	1.8	2.3	0.4	4.5	5.1
RaceHorses	0.1	-1.8	-1.6	0.3	3.5	3.6	0.3	6.6	6.0
Average	0.0	-1.6	-1.8	0.2	3.0	4.4	0.3	5.2	6.7
Average	0.0	-0.9	-1.1	0.1	2.0	2.6	0.1	3.8	4.7

표 8. Random access (low Complexity) 구조에서 실험결과
 Table 8. Experimental results in random access structure (low complexity)

	Proposed method			Directional filter			Linear filter		
	Y	U	V	Y	U	V	Y	U	V
Traffic	0.1	0.4	-0.2	0.1	1.3	2.4	0.2	3.5	5.7
PeopleOnStreet	0.0	0.2	-0.1	0.0	1.3	0.9	0.0	1.5	1.1
Average	0.0	0.3	-0.1	0.1	1.3	1.6	0.1	2.5	3.4
Kimono	0.0	0.2	-0.6	0.0	0.2	1.8	0.1	0.6	3.8
ParkScene	0.0	-0.4	-1.6	0.1	4.5	4.5	0.2	8.9	9.2
Cactus	0.0	-1.1	-0.9	0.2	2.4	4.3	0.2	2.2	7.5
BasketballDrive	0.0	-2.0	-2.2	0.1	2.7	3.6	0.2	6.7	8.1
BQTerrace	0.0	-4.7	-2.6	0.2	9.6	6.2	0.3	22.0	14.5
Average	0.0	-1.6	-1.6	0.1	3.9	4.1	0.2	8.1	8.6
BasketballDrill	-0.1	0.5	0.4	0.1	1.3	1.5	0.3	7.3	7.9
BQMall	0.0	-0.3	-0.2	0.1	1.7	1.9	0.1	5.6	5.6
PartyScene	-0.1	-2.2	-1.8	0.3	4.7	5.2	0.4	13.3	12.5
RaceHorses	0.0	-0.4	-0.1	0.1	4.6	4.2	0.1	7.2	7.0
Average	0.0	-0.6	-0.4	0.1	3.1	3.2	0.2	8.4	8.2
BasketballPass	-0.1	-1.1	-1.4	0.2	2.6	4.4	0.3	5.9	8.5
BQSquare	-0.1	-2.2	-3.3	0.2	7.5	12.2	0.3	13.2	20.0
BlowingBubbles	-0.1	-1.4	-1.4	0.2	3.4	4.2	0.4	10.1	10.7
RaceHorses	0.0	-1.3	-0.8	0.3	7.0	6.8	0.3	11.9	11.5
Average	-0.1	-1.5	-1.7	0.2	5.1	6.9	0.3	10.3	12.7
Average	0.0	-1.1	-1.1	0.1	3.6	4.3	0.2	8.0	8.9

표 9. Low delay (high efficiency) 구조에서 실험 결과
 Table 9. Experimental results in low delay structure (high efficiency)

	Proposed method			Directional filter			Linear filter		
	Y	U	V	Y	U	V	Y	U	V
Kimono	-0.1	-0.3	-0.7	0.1	0.7	0.6	0.1	1.0	1.7
ParkScene	-0.1	-1.3	-2.3	0.3	0.9	1.2	0.5	2.7	3.4
Cactus	-0.1	-0.9	-1.0	0.5	0.7	2.1	0.9	2.6	5.2
BasketballDrive	-0.1	-1.2	-1.3	0.2	1.1	0.8	0.3	2.7	2.6
BQTerrace	0.0	1.4	-1.2	0.2	6.6	7.3	0.3	11.8	14.3
Average	-0.1	-0.5	-1.3	0.2	2.0	2.4	0.4	4.2	5.4
BasketballDrill	0.1	0.3	-1.0	0.2	1.6	2.6	0.5	4.2	3.9
BQMall	-0.1	-0.7	-1.2	0.1	0.3	0.8	0.3	2.9	3.0
PartyScene	-0.2	-2.3	-2.9	0.5	2.1	2.5	0.9	6.4	6.8
RaceHorses	-0.1	-1.4	-1.4	0.2	1.6	0.9	0.3	2.9	2.4
Average	-0.1	-1.0	-1.6	0.3	1.4	1.7	0.5	4.1	4.0
BasketballPass	-0.2	-1.6	-2.1	0.4	2.3	2.4	0.7	3.9	3.9
BQSquare	-0.2	-1.2	-4.4	0.3	18.1	21.4	0.3	18.5	22.5
BlowingBubbles	-0.2	-2.6	-2.0	0.2	-0.2	1.6	0.8	5.9	8.2
RaceHorses	-0.3	-2.1	-2.0	0.5	2.4	1.8	0.8	4.4	2.9
Average	-0.2	-1.9	-2.6	0.3	5.7	6.8	0.7	8.2	9.4
Vidyo1	-0.5	1.2	0.6	-0.5	3.1	2.5	0.1	7.9	6.6
Vidyo3	0.1	-1.6	1.5	0.2	-1.3	-1.2	0.2	1.4	9.6
Vidyo4	0.2	-0.1	-0.9	0.1	-1.8	-1.8	0.0	-1.8	-0.4
Average	-0.1	-0.2	0.4	-0.1	0.0	-0.2	0.1	2.5	5.3
Average	-0.1	-0.9	-1.4	0.2	2.4	2.9	0.4	4.8	6.0

표 10. Low delay (low complexity) 구조에서 실험 결과
 Table 10. Experimental results in low delay structure (low complexity)

	Proposed method			Directional filter			Linear filter		
	Y	U	V	Y	U	V	Y	U	V
Kimono	-0.1	0.0	-1.5	0.2	-1.4	0.6	0.3	-0.4	3.1
ParkScene	-0.2	-1.7	-3.8	0.7	6.5	4.5	1.5	12.7	10.4
Cactus	-0.2	-1.9	-0.7	0.5	1.4	4.3	1.0	3.6	11.2
BasketballDrive	-0.2	-3.1	-2.8	0.5	5.0	4.1	0.8	9.4	8.2
BQTerrace	-0.1	-6.0	-6.5	0.6	14.6	13.8	1.0	28.2	26.4
Average	-0.2	-2.5	-3.0	0.5	5.2	5.5	0.9	10.7	11.8
BasketballDrill	0.0	2.7	2.2	0.3	1.9	2.8	0.8	13.4	13.5
BQMall	0.0	-1.0	-1.1	0.3	1.4	1.3	1.0	6.4	6.2
PartyScene	-0.3	-2.6	-2.3	0.6	3.2	4.8	1.2	10.9	12.2
RaceHorses	-0.2	-1.4	-0.9	0.5	7.1	4.2	1.1	11.0	7.6
Average	-0.1	-0.6	-0.5	0.4	3.4	3.3	1.0	10.4	9.9
BasketballPass	-0.2	-0.7	-1.3	0.5	3.9	3.8	1.1	8.5	7.4
BQSquare	0.0	-4.9	-5.9	0.7	16.0	24.2	0.8	22.8	32.9
BlowingBubbles	-0.1	-1.8	-1.7	0.5	0.7	2.1	1.3	7.8	10.9
RaceHorses	-0.1	-1.5	-1.6	1.2	7.7	6.5	2.1	13.7	11.7
Average	-0.1	-2.2	-2.6	0.7	7.1	9.1	1.3	13.2	15.7
Vidyo1	-0.1	-1.6	-0.7	0.3	-0.6	1.1	0.5	-0.3	1.8
Vidyo3	-0.1	-2.0	0.0	0.2	-6.6	-1.1	0.2	-8.6	0.6
Vidyo4	0.2	-0.7	-0.9	0.2	-4.5	-1.5	0.4	-6.1	-4.5
Average	0.0	-1.4	-0.5	0.2	-3.9	-0.5	0.4	-5.0	-0.7
Average	-0.1	-1.8	-1.8	0.5	3.5	4.7	0.9	8.3	10.0

표 11. 부호화 시간 비교

Table 11. Encoding time comparison

	Random access, high efficiency	Random access, low complexity	Low delay, high efficiency	Low delay, low complexity
ClassA	100%	101%		
ClassB	101%	101%	100%	100%
ClassC	100%	101%	100%	101%
ClassD	100%	101%	100%	100%
ClassE			100%	100%
Average	100%	101%	100%	100%

표 12. 복호화 시간 비교

Table 12. Decoding time comparison

	Random access, high efficiency	Random access, low complexity	Low delay, high efficiency	Low delay, low complexity
ClassA	102%	102%		
ClassB	102%	102%	101%	100%
ClassC	101%	99%	100%	101%
ClassD	101%	104%	106%	103%
ClassE			98%	101%
Average	101%	102%	101%	101%

회도성분(Y)의 BD-rate 변화 없이 random access 구조의 high efficiency 경우 색차성분 U,V에서 각각 0.9%, 1.1%의 BD-rate가 감소하였고, random access 구조의 low complexity인 경우 1.1%, 1.1%의 BD-rate가 감소하였고, low delay 구조의 high efficiency인 경우 색차성분 U,V에서 각각 0.9%, 1.4%의 BD-rate가 감소하였고, low delay 구조의 low complexity인 경우 1.8%, 1.8%의 BD-rate가 감소하였다. 반면, 방향성 필터와 선형 필터는 BD-rate가 전체적으로 증가되는 것을 볼 수 있다.

표 11과 12은 인코딩, 디코딩 시간을 제안한 방법과 비교한 표이다. 기존 보간 방법보다 전체적인 부호화, 복호화 시간이 1~2% 증가하였지만, 전체적인 BD-rate를 감소시키며 부호화 성능을 향상시켰다.

V. 결론

제안된 FIR 필터와 선형필터의 통합필터는 기존 HM2.0

에 구현된 DCT-IF보다 계산 복잡도가 증가하면서 부호화, 복호화 시간이 증가하였지만, BD-rate를 감소시키므로 향상된 부호화 성능 결과를 가져왔다. 따라서 제안된 방법은 고화질 영상을 필요로 하는 동영상 부호화에 유용하게 사용되어질 것으로 전망된다.

참고 문헌

- [1] Thomas Wiegand, Woo-Jin Han, Benjamin Bross, Jens-Rainer Ohm, Gary J. Sullivan, "WD1:Working Draft 1 of High-Efficiency Video Coding", Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 3rd Meeting, Guangzhou, CN, 7-15 October, 2010.
- [2] ITU-T Recommendation H.264 and ISO/IEC 14496-10 "Advanced video coding for generic audiovisual services", May 2003.
- [3] Thomas Wiegand, Gary J. Sullivan, Gisle Bjntegaard, and Ajay Luthra, "Overview of the H.264/AVC video coding standard", IEEE Trans. Circuits Syst. Video Technol., pp. 560-576, July 2003.
- [4] Frank Bossen, TK Tan, Junya Takiue, "Simplified angular intra prediction", Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 2nd Meeting,

Geneva, CH, 21-28 July, 2010.

[5] Ken McCann, Woo-Jin Han, Il-Koo Kim, "Samsung's Response to the Call for Proposals on Video Compression Technology", Joint Collaborative Team of Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 1st Meeting, Dresden, DE, 15-23, April, 2010.

[6] Kemal Ugur, Kenneth R. Andersson, Arild Fuldseth, "Description of video coding technology proposal by Tandberg, Nokia, Ericsson", Joint Collaborative Team of Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 1st Meeting, Dresden, DE, 15-23, April, 2010.

[7] T. Chujoh, R. Noda, "Internal bit depth increase for coding efficiency", ITU-T SG16 Q.6 Document, VCEG-AE13, Marrakech, Jan, 2007.

[8] T. Chujoh, R. Noda, "Internal bit depth increase except frame memory", ITU-T SG16 Q.6 Document, VCEG-AF07, San Jose, April, 2007.

[9] Minhua Zhou, "Evaluation results on IBD1", Joint Collaborative Team of Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting, Daegu, KR, 20-28, January, 2011.

[10] Jianle Chen, Elena Alshina, Woo-Jin Han, "CE4:Experimental results of DCTIF application for Chroma MC by Samsung", Joint Collaborative Team of Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting, Daegu, KR, 20-28, January, 2011.

[11] <http://hevc.kw.bbc.co.uk/trac/browser/tags/HM-2.0>.

[12] Frank Bossen, "Common test conditions and software reference configurations", Joint Collaborative Team on Video Coding(JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, 4th Meeting, Daegu, KR, 20-28, January, 2011.

[13] <ftp://hevc@ftp.tnt.uni-hannover.de/testsequences/>

[14] Dmytro Rusanovskyy, Kemal Ugur, Antti Hallapuro, Jani Lainema, and Moncef Gabbouj. "Video Coding With Low-Complexity Directional Adaptive Interpolation Filters", IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 8, pp. 1239-1243 August 2009.

[15] Kemal Ugur, Jani Lainema, "CE4:Results for chroma interpolation filter tests by Nokia", Joint Collaborative Team of Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 4th Meeting, Daegu, KR, 20-28, January, 2011.

[16] Gisle Bjontegarrd, "Calculation of Average PSNR Differences between RD curves", ITU-T SC16/Q6, 13th VCEG Meeting, Austin, Texas, USA, Apr. 2001.

저 자 소 개



김 정 필

- 2010년 2월 : 세종대학교 컴퓨터공학과 (학사)
- 현재 : 세종대학교 컴퓨터공학과 (석사과정)
- 주관심분야 : 영상압축, H.264/AVC, HEVC



이 영 렬

- 1985년 2월 : 서강대학교 전자공학과 (학사)
- 1987년 2월 : 서강대학교 전자공학과 (석사)
- 1999년 2월 : 한국과학기술원 전기 및 전자공학과 (박사)
- 1987년 1월 ~ 1994년 2월 : 삼성전자 중앙연구소 DMS Lab. 선임연구원
- 1999년 3월 ~ 2001년 8월 : 삼성전자 중앙연구소 DMS Lab. 수석연구원
- 2001년 9월 ~ 현재 : 세종대학교 컴퓨터공학부 컴퓨터공학과 부교수
- 주관심분야 : 영상처리(압축, 복원), 영상전송, 멀티미디어 시스템, 3차원 비디오코딩, 트랜스코딩, Scalable 비디오 코딩