

일반논문 (Regular Paper)

방송공학회논문지 제23권 제3호, 2018년 5월 (JBE Vol. 23, No. 3, May 2018)

<https://doi.org/10.5909/JBE.2018.23.3.431>

ISSN 2287-9137 (Online) ISSN 1226-7953 (Print)

HEVC에서 부분복호화를 통한 썸네일 추출 알고리즘

이 원 진^{a)}, 정 제 창^{a)*}

Fast Thumbnail Extraction Algorithm with Partial Decoding for HEVC

Wonjin Lee^{a)} and Jechang Jeong^{a)*}

요 약

본 논문에서는 aliasing artifact 없이 영상 품질을 유지하고, 썸네일 생성에 필요한 계산 복잡도를 줄이는 알고리즘을 제안한다. 제안하는 알고리즘은 고속으로 복호화를 진행하기 위해서 TU(Transform Unit)에서는 4x4크기마다 경계부분만을 부분 복호화를 수행하고, PU(Prediction Unit)에서는 TU경계부분만을 부분 복호화 한다. 그리고 화면내 예측 모드 방향에 따른 가중치 값을 구하고, 그 값을 이용해서 실제 썸네일 화소를 예측한다. 제안하는 방법은 기존 방법들과 썸네일 추출시간을 비슷하게 유지하면서 썸네일의 품질을 향상시킨다.

Abstract

In this paper, a simple but effective algorithm to reduce the computational complexity of thumbnail generation and to improve image quality without aliasing artifacts is proposed. For the high speed decoding, the proposed algorithm performs partial decoding per 4x4 boundary in TU(Transform Unit), and preforms TU boundary in PU(Prediction Unit). The proposed method defines the weights based on intra prediction directions and estimates the thumbnail pixel by using that weights. this method remains thumbnail extraction time and improves thumbnail image quality compared with conventional algorithms.

Keyword : Thumbnail, partial decoding, HEVC, down sampling, reduced size image

a) 한양대학교 전자컴퓨터통신공학과(Department of Electronic and Computer Engineering, Hanyang University)

* Corresponding Author : 정제창(Jechang Jeong)

E-mail: jjjeong@hanyang.ac.kr

Tel: +82-2-2220-4370

ORCID: <https://orcid.org/0000-0002-3759-3116>

※ 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신·방송 연구개발사업의 일환으로 수행하였음.[2014-0-00670, ICT 장비용 SW 플랫폼 구축]

※ This work was supported by the ICT R&D program of MSIP/IITP [2014-0-00670, Software Platform for ICT Equipment]

· Manuscript received March 19, 2018; Revised April 23, 2018; Accepted April 23, 2018.

1. 서 론

썸네일 이미지는 기존 이미지의 축소판이며, 이를 표시하여 사용자가 쉽고 빠르게 이미지나 동영상을 검색 할 수 있도록 돕는다.

부호화 된 비디오 비트 스트림으로부터 썸네일 이미지를 추출하기 위해서는 전체 복호화 과정 후에 다운샘플링 방법을 사용하여 썸네일을 추출해야하기 때문에 높은 복잡도를 요구한다. 이 문제를 해결하기 위해 많은 썸네일 추출

Copyright © 2016 Korean Institute of Broadcast and Media Engineers. All rights reserved.

“This is an Open-Access article distributed under the terms of the Creative Commons BY-NC-ND (<http://creativecommons.org/licenses/by-nc-nd/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited and not altered.”

알고리즘이 제안되었다. MPEG-2^[1]에서는 DCT영역에서의 DC값은 화소영역에서의 평균값이기 때문에 썸네일 이미지는 DCT 영역에서 매크로 블록 당 DC값을 추출함으로써 바로 생성할 수 있다. H.264 / AVC에서는 높은 압축률을 얻기 위해 화면 내 예측이 제안되었다^[2]. DCT 영역의 DC 값은 화면 내 예측 방법으로 인해 발생하는 잔차값들에 대한 DC 값이므로 MPEG-2의 축소 영상 추출 알고리즘을 사용할 수 없다. 그렇기 때문에 H.264 / AVC를 위한 썸네일 추출 알고리즘이 제안되었다^[3,4]. 이러한 알고리즘은 DCT 영역에서 일반화 된 수식을 정의하고, 이를 LUT (Look-Up Table)로 미리 정의한 뒤, 썸네일 이미지를 추출한다.

최근에 표준화 된 HEVC^[5]은 매우 높은 압축률을 제공하고 H.264 / AVC 보다 복잡도가 굉장히 높기 때문에 HEVC로 부호화된 비트스트림에서 썸네일을 추출하기 위해서는 복잡도가 낮을 필요가 있다. 그러나 H.264 / AVC의 썸네일 추출 방법은 높은 압축비를 달성하기 위해 새로 추가된 화면내 예측 모드와 다양한 블록 크기로 인해 HEVC에 적용될 수 없다.

HEVC에서 복잡도가 낮은 썸네일 추출 알고리즘이 연구되었지만, 이러한 방법은 aliasing artifact에 의해서 썸네일 품질 저하가 발생한다^[6,7]. 최근에 제안된 알고리즘은 미리 정의된 가중치 값을 이용하여 실제 썸네일 화소값과 유사한 값을 구한다^[8]. 이렇게 구해진 썸네일은 aliasing artifact를 크게 줄이지만, TU, PU에 대해서 4x4크기마다 경계부분을 복원하고, 다시 4x4크기마다 예측평균값을 구하기 때문에 하드웨어 구현 측면이나 구현 복잡도 면에서 상당히 불리하다는 단점이 있다. 또한 TU와 PU에서 구한 예측 평균값의 오차값 때문에 화질열화가 발생 할 수 있다.

따라서 본 논문에서는 낮은 복잡도를 유지하면서 HEVC에서 썸네일의 화질을 향상시키는 새로운 방법을 제안한다. 본 논문에서는 미리 정의 된 가중치 테이블과 최적의 PU부분 복호화를 통해 썸네일 추출을 위한 불필요한 계산을 줄이고, PU의 예측 평균값의 오차를 완전히 없애서 썸네일 이미지의 화질을 향상시키는 방법을 제안한다. 본 논문에서는 4x4, 16x16, 32x32 TU크기들도 동일한 방식으로 구현될 수 있기 때문에 8x8 TU 크기에 대해서만 설명한다.

II. 제안하는 방법

1. TU와 PU에서의 부분 복호화

HEVC의 8×8 invers discrete cosine transform (IDCT)은 다음과 같이 기술 될 수 있다.

$$r_{8 \times 8} = H_{8 \times 8}^T \times R_{8 \times 8} \times H_{8 \times 8}. \quad (1)$$

여기에서 H는 변환행렬을 의미하고, r과 R은 각각 화소영역과 변환영역의 잔여블록을 의미한다. HEVC로 부호화된 비트스트림에서 썸네일 영상을 만들어내기 위해서는 식(1)을 이용한 IDCT가 반드시 수행되어야 한다. 만약 하나의 화면내 예측 프레임을 만들어 내기 위해서는 잔여블록 내의 모든 잔여값들은 반드시 복원되어야만 한다. 하지만 썸네일 영상을 만들기 위해서는 모든 잔여값들이 필요하지 않다. 그렇기 때문에 저복잡도로 썸네일을 추출하기 위해서 부분 역변환을 제안한다. 부분 역변환은 다음 수식과 같이 잔여블록의 4x4크기마다 경계 부분만 복원을 수행한다.

$$r_{8 \times 8}^{hor} = \begin{cases} H_{8 \times 1, k}^T \times R_{8 \times 8} \times H_{8 \times 8}, & \text{if } k \in \{3, 7\} \\ 0, & \text{otherwise} \end{cases},$$

$$r_{8 \times 8}^{ver} = \begin{cases} H_{8 \times 8}^T \times R_{8 \times 8} \times H_{8 \times 1, k}, & \text{if } k \in \{3, 7\} \\ 0, & \text{otherwise} \end{cases}, \quad (2)$$

$$r_{8 \times 8} = r_{8 \times 8}^{hor} + (r_{8 \times 8}^{hor} \oplus r_{8 \times 8}^{ver}) \otimes r_{8 \times 8}^{ver}.$$

위 수식에서 k는 열 벡터의 위치, $\otimes$ 와 $\oplus$ 은 element-wise 곱셈과 XOR 연산을 각각 나타낸다. 여기서 k는 몇번째 열벡터인지를 표시하는 값이고 4x4단위의 경계마다 복원을 수행해야 하기 때문에 4씩 증가하게 된다. 만약 TU크기가 16x16이면 k는 3,7,11,15가 된다. DST-7변환에 대해서도 동일한 수식을 적용하여 부분복호화를 수행한다. 위에서 언급했듯이, 제안하는 알고리즘은 화면내 예측을 통해 만들어진 예측블록내에서도 전체 예측값들이 필요하지

않다. 기존의 알고리즘들은 4x4크기당 썸네일 화소를 생성하기 위해서 화면내 예측에서도 4x4크기마다 경계부분을 복원했다. 제안하는 부분 화면내 예측은 복잡도를 크게 낮추기 위해서 TU의 경계부분만을 복원한다. 기존의 방법들 [6,7,8]은 4x4단위 주변에 복원된 샘플들(4x4 경계부분)이 예측평균값을 구하기 위해 사용되기 때문에 반드시 구해야 하지만, 제안하는 방법은 아래에서 제안한 새로운 예측평균값을 구하는 방법으로 인해 4x4경계부분을 복원하지 않고 오직 TU 경계부분만을 복원한다. 만약 TU가 8x8일 때, 부분 역변환은 4x4단위의 경계마다 잔여값들을 복원시키기 때문에 총 28개의 잔여값들을 복원시키고, 부분 화면내 예측은 8x8 경계부분에 있는 15개의 예측값만을 복원한다.

2. 썸네일 생성을 위한 down sampling 방법

썸네일은 축소영상이기 때문에 썸네일 영상을 만들기 위해서 원본영상에서 다운샘플링을 반드시 수행해야 한다. 일반적으로, 썸네일은 가로 세로 각각 1/4크기로 줄어든 영상이기 때문에 4x4영역당 한 화소를 추출하는 방법이 제안되었다. 하지만 이러한 방법은 심각한 aliasing artifact를 발생시킨다^[6]. 썸네일 영상은 4x4블록내에 있는 모든 화소들의 평균값을 계산해서 만들 수 있는데, 이 방법은 aliasing artifact를 줄일 수 있지만 4x4당 한 화소를 추출하는 것보다 더 높은 계산 복잡도를 가진다. 이전에 제안된 방법 중 aliasing을 줄이고 계산 복잡도를 낮추는 방법으로서 4x4 경계부분에 있는 7개의 화소에 대한 평균값을 구하여 간단하게 썸네일을 만들어 내는 방법이 제안되었다^[7]. 우리가 제안하는 방법은 4x4 블록내의 모든 화소값들의 평균값을 미리 정의된 가중치 표를 이용해서 근사치를 구한다. 썸네일 화소는 예측 블록과 잔여블록의 가중 평균값들을 더해서 만든다.

<표 1>은 잔여값들의 평균값을 구하기 위해서 화면내 예

측모드에 따라서 미리 가중치값들을 정의했다. 제안하는 방법은 4x4크기마다 경계부분의 잔여값만 존재하기 때문에 존재하는 잔여값에 가중치 값을 곱해서 4x4크기내에 있는 잔여값들의 평균을 예측한다. 만약 10번 모드의 경우 화소값들이 10번 모드의 방향성을 가지고 있다는 뜻이고 이는 잔여값들 또한 같은 방향성을 가지고 있다는 의미이다. 그렇기 때문에 <표 1>과 같이 간단한 가중치 값을 곱해서 4x4크기내에 복원하지 않은 잔여값들을 예측한다.

0번과 1번모드는 4x4경계부분에 있는 7개의 잔여값들의 평균을 사용한다. 6번 모드에 대한 가중치는 2번모드와 10번모드의 평균으로 정의했고, 4번모드에 대한 가중치는 2번모드와 6번모드의 평균으로 정의했다. 나머지 모드들에 대한 가중치는 위에 설명한 것과 동일한 방법으로 정의했다. 잔여값들의 가중 평균값에 대한 수식은 미리 정의한 <표1>을 이용하여 다음과 같이 정의했다.

$$AR_{2 \times 2}(i, j) = \frac{\sum_{y=3}^3 \sum_{x=0}^3 rw_{x+3}^M \cdot r_{8 \times 8}(x+4i, y+4j)}{\sum_{k=0}^6 rw_k^M} + \frac{\sum_{x=3}^3 \sum_{y=0}^2 rw_y^M \cdot r_{8 \times 8}(x+4i, y+4j) + \frac{1}{2} \cdot \sum_{k=0}^6 rw_k^M}{\sum_{k=0}^6 rw_k^M}. \quad (3)$$

여기서 rw 은 잔여값들의 평균값을 구하기 위해서 <표1>에서 미리 정의한 가중치를 의미한다.

부분 화면내 예측을 통해 복원된 예측값들은 잔여값들과는 달리 TU 경계부분에 있는 값들만 복원된다. 그렇기 때문에 잔여값들의 평균을 구할 때와는 달리 현재 복원하려고 하는 블록의 참조화소들을 이용해서 4x4당 하나의 예측 평균값을 구한다. 기존의 화면내 예측은 현재 코딩하려고

표 1. 잔여값들의 평균을 위한 대표 가중치값

Table 1. Representative weights for average residual value

Mode (M)	rw_0^M	rw_1^M	rw_2^M	rw_3^M	rw_4^M	rw_5^M	rw_6^M
2	4	3	1	4	3	1	1
10	4	4	4	1	1	1	1
18	1	2	3	1	2	3	4
26	1	1	1	4	4	4	1
34	4	3	1	4	3	1	1

하는 블록주위에 있는 참조화소들을 이용해서 현재 블록과 유사한 예측블록을 생성한다. 이 예측블록을 구한 뒤 4x4크기마다 16개의 예측값들에 대한 평균값을 구하면 예측블록에 대한 평균값을 구할 수 있다. 이 과정을 단축 시키기 위해서 우리는 예측블록을 만든 뒤 평균값을 구하지 않고, 참조화소에 가중치 값을 곱해서 예측블록에 대한 평균값들을 바로 구한다. <표2>는 화면내 예측모드가 3일 때 대표 가중치 값들이다. 나머지 모드들에 대한 가중치 값들도 직접 예측블록 복호화와 평균값을 구하는 식을 풀어 쓰면 쉽게 구할 수 있다. 예측값들의 가중평균값에 대한 수식은 미리 정의한 <표2>를 이용하여 다음과 같이 정의했다.

표 2. 예측값들의 평균을 위한 대표 가중치값 (화면내 예측모드가 3일때), $pw_j^3(h)$
Table 2. Representative predefined weight, $pw_j^3(h)$, of intra prediction mode 3

	$h=0$	$h=1$	$h=2$	$h=3$	$h=4$	$h=5$	$h=6$	$h=7$
$j=0$	3	22	41	60	61	42	23	4
$j=1$	17	37	56	64	47	27	8	0

$$(l, k) = \begin{cases} (4i-1, 4j+3) & \text{if } 11 \leq \text{intra mode} \leq 25 \\ (4i, 4j) & \text{otherwise} \end{cases},$$

$$z = l + ((1+k) \cdot d) \ggg 5 + 1,$$

$$tw = \sum_{r=0}^7 pw_j^m(r),$$

$$AP_{2 \times 2}(i, j) = \frac{\sum_{r=z}^{z+7} pw_j^m(r-z) \cdot ref_{r,0} + \frac{tw}{2}}{tw},$$

$$AP_{2 \times 2}(i, j) = \left(\sum_{r=z}^{z+7} pw_j^m(r-z) \cdot ref_{r,0} + \frac{tw}{2} \right) \ggg \log_2(tw).$$

$$THB_{2 \times 2}(i, j) = AR_{2 \times 2}(i, j) + AP_{2 \times 2}(i, j), \quad (5)$$

표 3. 하나의 썸네일 화소를 구하기 위한 연산량 비교

Table 3. Computational complexity for one thumbnail pixel

Sequence	FD		Method ^[6]		Method ^[7]		Method ^[8]		Proposed method	
	덧셈	곱셈	덧셈	곱셈	덧셈	곱셈	덧셈	곱셈	덧셈	곱셈
PeopleOnTheStreet	482	215	207	119	213	121	215	134	180	115
Traffic	535	257	261	161	267	164	269	177	220	148
BQTerrace	461	212	199	119	206	122	208	134	172	113
Cactus	507	245	243	151	248	153	251	165	205	137
Kimono	572	317	333	224	338	226	340	240	269	193
ParkScene	512	243	252	153	258	155	260	168	213	140

III. 실험 및 결과

제안하는 알고리즘은 HEVC 참조 소프트웨어인 HM 13에서 실험했다^[9]. 실험에는 HEVC 비디오 시퀀스를 이용했고, 모든 프레임들은 I 프레임으로 코딩되었다.

<표 3>는 기존 방법들과 하나의 썸네일 화소를 구하기 위한 연산량을 비교하였다. <표 3>에서 Full Decoding method (FD)는 완전한 복호화 후 4x4크기마다 평균값을 썸네일 화소로 하는 방법이다. 기존의 방법들[6,7,8]은 편의상 [6], [7], [8]방법으로 표기했다. 제안하는 알고리즘은 FD와 비교했을 때 덧셈과 곱셈 각각 약 59%, 43%의 연산량 감소를 보였고, [6], [7], [8]방법들과 비교했을 때 가장 적은 연산만을 이용해서 썸네일 화소를 추출하는 것을 확인할 수 있다. <표4>에서는 썸네일 화소의 휘도성분으로 PSNR을 비교했다. 제안하는 알고리즘은 FD와 비교했을 때 평균적으로 약 6dB 정도의 PSNR 감소를 보였고, [6]방법보다 약 10dB, [7]보다 약 3dB, [8]보다 약 0.1dB 높았다.

[6]방법은 4 to 1 다운샘플링으로 인해 PSNR이 낮았고, [7]방법은 복원된 7개의 화소를 단순히 평균값을 구했기 때문에 썸네일 화소를 정확하게 예측하지 못했다.

<그림1>은 BQTerrace 시퀀스의 썸네일 추출 결과를 보여준다. 여기서 [8]방법은 가중치를 이용해서 4x4마다 예측 평균값을 구하는데 FD결과와 비교했을 때 aliasing이 크게 줄어들었지만 TU에서 구한 예측평균값과 실제 평균값에 대한 오차와 PU에서 구한 예측평균값과 실제 평균값과의 오차 때문에 <그림 1e>의 오른쪽부분에서 화질열화가 발생한 것을 확인할 수 있다. 제안하는 방법은 PU에서 구한 예측평균값이 실제 평균값과 정확하게 일치하기 때문에 실제 썸네일화소와의 오차가 적어지면서 [8]방법보다 aliasing이 줄어든 것을 확인할 수 있었다.

표 4. 썸네일 영상의 PSNR (dB) 비교
Table 4. PSNR (dB) of luminance component

Sequence	# of frames	FD	Method ^[6]	Method ^[7]	Method ^[8]	Proposed method
PeopleOnTheStreet	150	42.68	23.18	30.40	34.81	34.99
Traffic	150	42.95	25.79	32.85	36.28	36.39
BQTerrace	600	43.07	22.21	29.58	32.43	32.51
Cactus	500	43.32	26.75	33.84	37.29	37.40
Kimono	240	43.83	30.95	37.44	39.47	39.55
ParkScene	240	42.87	27.91	34.86	36.17	36.28

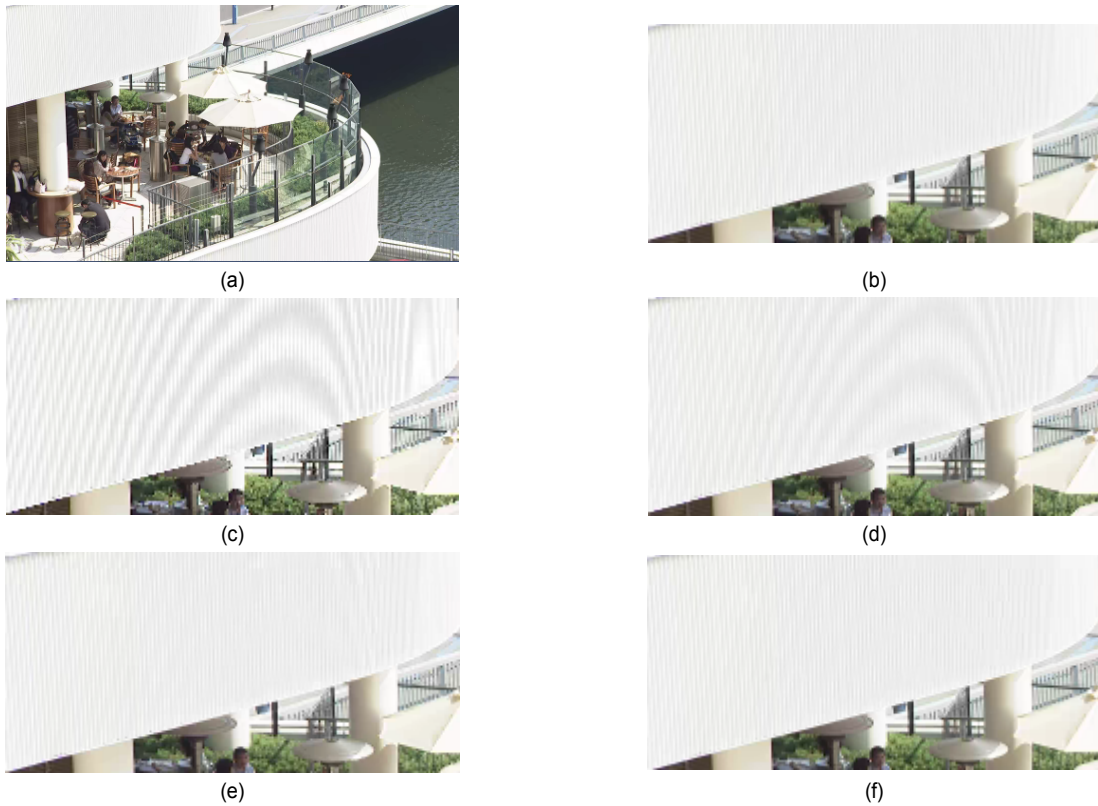


그림 1. BQTerrace 시퀀스에 대한 썸네일 생성 결과 (a) 원본시퀀스의 첫번째 프레임, (b) FD의 확대결과, (c) [6]방법의 확대결과, (d) [7]방법의 확대결과, (e) [8]방법의 확대결과, (f) 제안하는 방법의 확대결과

Fig. 1. Thumbnail results for the "BQTerrace" sequence (a) Original sequence (1st frame), (b) Zoomed result of FD, (c) Zoomed result of method [6], (d) Zoomed result of method [7], (e) Zoomed result of method [8], (f) Zoomed result of proposed algorithm

IV. 결 론

우리는 썸네일 생성에 필요한 계산 복잡도를 줄이기 위한 부분 복호화와 썸네일 영상의 시각적 품질 향상을 위해서 미리 정의한 가중치 표를 제안했다. 제안하는 알고리즘은 화면내 예측 방향을 고려한 가중치 값들을 이용하여 썸네일 영상의 시각적 품질을 향상시켰다. 또한, 제안하는 방

법은 썸네일 생성을 위한 계산 복잡도를 크게 증가시키지 않고 aliasing artifact를 효과적으로 제거했다. 제안하는 방법에서, PU에서 간단한 부분복호화를 통해 TU 경계부분만을 복원하는 방식과 참조화소만을 이용한 PU의 예측 평균 값을 구하는 방식은 실제 하드웨어 구현시 메모리 측면이나 구현 복잡도 측면에서 기존 방법들보다 큰 장점이 될 수 있을 것이다.

참 고 문 헌 (References)

- [1] ITU-T and ISO/IEC JTC 1, ITU-T Recommendation H.262 and ISO/IEC 13818-2 MPEG-2: "Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video," 1994
- [2] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560-576, July 2003.
- [3] E.-S. Kim, T.-W. Um, and S.-J. Oh, "A fast thumbnail extraction method in H.264/AVC video streams," IEEE Trans. Consumer Electron., vol. 55, no. 3, pp. 1424-1430, Aug. 2009.
- [4] M. Kim, H. Lee, and S. Sull, "Fast thumbnail generation in integer DCT domain for H.264/AVC," IEEE Trans. Consumer Electron., vol. 57, no. 2, pp. 589-596, May 2011.
- [5] G.J. Sullivan, J.R. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp.1649-1668, Dec. 2012.
- [6] W. Lee, G. Jeon, and J. Jeong, "Thumbnail extraction for HEVC using prediction modes," IET Electron. Lett., vol. 51, no. 6, pp. 480-482, Mar. 2015.
- [7] W. Lee, G. Jeon, and J. Jeong, "Efficient fast thumbnail extraction algorithm for HEVC," Proceeding of Real-Time Image and Video Process., vol. 9400, San Francisco, USA, pp. 94000H1-8, 2015.
- [8] W. Lee, G. Jeon, and J. Jeong, "Video thumbnail extraction for HEVC," Signal Process.: Image Commun., vol. 58, pp. 282-286, Oct. 2017.
- [9] D. Flynn, H.265/HEVC Reference Software (online), <https://hevc.hhi.fraunhofer.de/trac/hevc/browser/branches/HM-13.0-dev>, Mar. 2013

저 자 소 개



이 원 진

- 2010년 02월 : 명지대학교 정보공학과 학사
- 2010년 03월 ~ 현재 : 한양대학교 전자컴퓨터통신공학과 석박사
- ORCID : <https://orcid.org/0000-0001-8505-0290>
- 관심분야 : 영상처리, 영상압축



정 제 창

- 1980년 02월 : 서울대학교 전자공학과 학사
- 1982년 02월 : KAIST 전기전자공학과 석사
- 1990년 : 미국 미시간대학 전기공학과 공학박사
- 1980년 ~ 1986년 : KBS 기술연구소 연구원(디지털 TV 및 뉴미디어 연구)
- 1990년 ~ 1991년 : 미국 미시간대학 전기공학과 연구 교수(영상 및 신호처리 연구)
- 1991년 ~ 1995년 : 삼성전자 멀티미디어 연구소(MPEG, HDTV, 멀티미디어 연구)
- 1995년 ~ 현재 : 한양대학교 전자통신컴퓨터공학과 교수(영상통신 및 신호처리 연구실)
- 1990년 12월 : 정보통신부장관상 수상
- 1998년 11월 : 과학기술자상 수상
- 2007년 : IEEE Chester Sall Award 수상
- 2008년 : ETRI Journal Paper Award 수상
- 2011년 5월 : 제 46회 발명의 날 녹조근정훈장 수훈
- ORCID : <http://orcid.org/0000-0002-3759-3116>
- 관심분야 : 영상처리, 영상압축, 3DTV, UHD TV