

# HEVC의 계층적 부호화 블록 구조

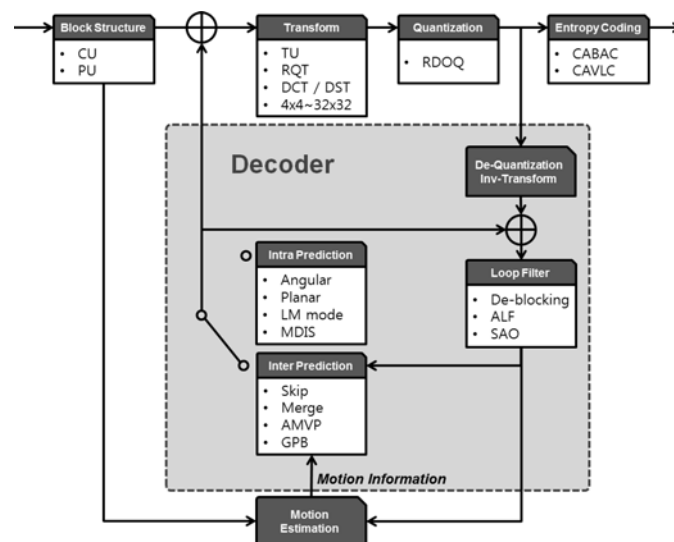
서찬원 · 한종기 (세종대학교)

## I. 서론

MPEG(Moving Picture Experts Group)과 VCEG(Video Coding Experts Group)은 H.264/AVC의 표준화 이후, 고화질 및 고해상도 영상 압축을 위한 코덱의 필요성이 증가함에 따라 이를 만족시킬 수 있는 비디오 코덱의 표준화를 시작하였다. 그 첫 번째 과정으로, 2010년 1월 일본 Kyoto에서 열린 표준화 회의에서 JCT-VC (Joint Collaborative Team on Video Coding)을 설립하였다. 그리고 새로운 비디오 코덱 표준화 프로젝트인 HEVC (High Efficiency Video Coding)를 본격적으로 진행하기 시작하였으며, 차세대 코덱 표준화를 위한 CiP (Call for Proposal)을 발표하였다<sup>[1]</sup>. 국내외 27개 기업, 연구소, 그리고 대학들은 CiP를 만족하는 비디오 코덱들을 설계 및 구현하여 제출하였으며, 2010년 4월 독일 Dresden에서 열린 첫 번째 JCT-VC 회의에 이 코덱들을 평

가하고 논의하였다<sup>[3]</sup>. 그 후 2010년 10월 JCT-VC 중국 Guangzhou 회의에서는 참여기관들이 발표한 기술들을 기반으로 HM (HEVC Test Model)이 결정되었다<sup>[2]</sup>.

MPEG-2를 비롯하여 MPEG-4, H.263 H.264/AVC 등의 기존 비디오 코덱들에서는 하나의 픽처(Picture)를 16×16 크기 블록 단위인 MB(Macro Block)으로 분할하여 부호화하였으나, HEVC에서는 부호화 기본단위인 CU (Coding Unit)을 사용한다. CU는 SPS(Sequence Parameter Set)에 저장된 파라미터값이 지정하는 최대 및 최소 크기까지 분할될 수 있다<sup>[5]</sup>. HM3.0에서는 최대와 최소 CU 크기를 각각 64×64와 8×8로 설정하여 부호화 및 복호화를 수행한다. 예측 부호화의 기본단위는 PU(Prediction Unit)로 정의되며, 하나의 CU는 다수개의 PU로 분할되어 예측된다. 예측 방법은 크게 인트라(Intra) 예측과 인터(Inter) 예측으로 분류할 수 있다. HM은 부호화 효율을 향상시키기 위하여 <그림 1>과 같이 다



<그림 1> HEVC 부호화기 구조

양한 기술들을 사용한다. 인트라 예측에서는 최대 34가지 방향성을 고려한 인트라 예측과 평탄한 영역을 효율적으로 부호화하기 위한 Planar 예측이 사용되고 있으며, 인터 예측에서는 Merge 모드 및 AMVP(Advanced Motion Vector Prediction) 등의 기술들이 사용된다. 또한, 변환 부호화의 효율을 극대화하기 위하여 정수화된 DCT (Discrete Cosine Transform) 및 DST (Discrete Sine Transform)를 적응적으로 사용한다. 그 밖에 복원 영상의 화질 향상을 위하여 ALF (Adaptive Loop Filter) 및 SAO (Sample Adaptive Offset) 등의 기술들도 사용된다<sup>[5]</sup>. 이와 같이 JCT-VC에서는 다양한 기술을 채택하여 HEVC 코덱의 부호화 효율을 향상시키기 위한 노력을 하고 있다. 이 과정에서, 부호화 효율뿐만 아니라 코덱의 복잡도도 동시에 고려하여 효율적인 Test Model을 정립하기 위한 다양한 실험과 분석을 수행하고 있다.

본 문서에서는 HM3.0의 계층적 부호화 구조와 변환 블록 구조를 구체적으로 서술하고, 최근 표준화 회의를 통해 채택된 기술 중 부호화 블록 구조와 관련된 내용을 간략하게 소개한다. 본 문서의 제 II 장에서는 HM3.0에서 사용되는 계층적 부호화 구조에 대하여 설명하고, 제 III 장에서는 예측 및 변환 블록 구조의 특징들을 서술한다. 제 IV 장에서는 최근 표준화 동향을 반영한 HM4.0의 부호화 구조에 대하여 설명한다. 그리고 본 문서의 마지막 제 V 장에서 본 문서의 결론을 맺는다.

## II. 계층적 부호화 구조

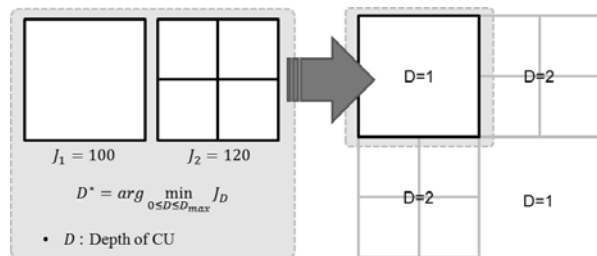
기존 비디오 코덱에서는 기본 부호화 단위를 16×16 크기의 블록으로 사용하였으며, 이를 MB 로 정의하였다. HEVC에서는 기본 부호화 단위를 CU로 정의하며, CU의 최소 크기는  $\log_2\_min\_coding\_block\_size\_minus3$ 을 이용하여 표현하며, 최대 CU 크기는  $\log_2\_diff\_max\_min\_coding\_block\_size$ 을 이용하여 계산할 수 있도록 한다. 해당 선택스(Syntax)들은 SPS에 포함되어 전송된다<sup>[5]</sup>. 현재 HM3.0에서는 기본적으로 최대 CU 크기를 64×64로, 최소 CU 크기를 8×8로 설정하여 부호화 및 복호화를 수행한다. 부호화기에서는 최대 크기의 CU부터 최소 크기의 CU까지 모두 사용하여 부호화 효율이 가장 좋은 크기의 CU를 사용하여 부호화를 수행한다. 부호화 효율은 수식 (1)의 R-D (Rate-Distortion) Cost 최소화 관점에서 최적의 CU 크기를 결정한다.

$$J = D_Q + \lambda \times (R_H + R_C) \quad (1)$$

(1)에서  $D_Q$ 은 양자화로 인하여 발생하는 에러를 나타내

며,  $R_H$ 은 예측 모드 및 움직임 정보 등을 부호화하기 위하여 사용되는 헤더 비트량을 의미한다.  $R_C$ 은 양자화 된 변환 계수의 부호화에 사용되는 비트량을 나타낸다.  $\lambda$ 은 라그랑지 (Lagrange) 곱수이며, 양자화 에러와 사용된 비트량간의 상대적인 중요도를 표현하는 상수이다. 부호화기는 가장 작은 값의  $J$ 를 발생시키는 CU 크기를 최적의 CU 크기로 결정한다. <그림 2>는 최적의 CU크기를 결정하는 과정을 보여준다. CU의 분할 구조를 표현하기 위하여 엔트로피 코더에 따라  $cu\_split\_pred\_part\_mode$  또는  $split\_coding\_unit\_flag$ 을 사용한다. <표 1>은 엔트로피 코더에 따른 CU 분할에 사용되는 선택스를 보여준다.

HEVC에서는 다양한 크기의 CU를 사용함으로써, 영상의 공간 해상도 및 블록 특성을 효과적으로 고려하여 부호화할 수 있다. 일반적으로 영상의 해상도가 작거나 화소값들이 국지적으로 크게 변화는 경우에는 작은 크기의 CU들을 이용하여 인트라 및 인터 예측을 수행하는 것이 효율적이다. 작은 크기의 CU를 이용하게 되면 부호화에 필요한 헤더 비트량  $R_H$ 는 증가하지만, 상대적으로 예측이 정밀하게 이루어져 양자화 에러  $D_Q$ 와 변환 계수의 부호화에 필요한 비트량  $R_C$ 이 감소하는 장점이 있다. 반대로 영상의 공간해상도가 크거나 화소값들의 변화가 적은 영역에서는 큰 CU를 사용하는 것이 부호화 효율을 높일 수 있다. 이 경우에는 큰 CU를 사용하여도 작은 CU를 사용하여 예측하는 경우에 비하여 예측오차가 크게 증가하지 않는 경향이 있으므로 이러한 블록들을 부호화할 경우, 큰 CU를 사용하여 헤더 비트량  $R_H$ 을 절약하는 것이 효율적이다. <표 2>는 HM3.0을 이용하여 부호화한 영상별 최적의 CU 깊이의 평균을 보여준다. 값이 0이면 64×64 CU를 의미하고 3이면 8×8 CU를 나타낸다. <표 2>와 같이 영상의 해상도가 높을수록 큰 CU가 많이 사용되고, 해상도가 낮을수록 작은 CU가 많이 선택됨을 알 수 있다. 또



<그림 2> 최적의 CU 크기 결정 과정의 예

<표 1> 엔트로피 코더에 따른 CU 분할 선택스

Entropy Coder	Syntax
CAVLC	$cu\_split\_pred\_part\_mode$
CABAC	$split\_coding\_unit\_flag$

〈표 2〉 HM3.0에서 영상별 최적의 평균 CU 깊이

해상도	영상	AI-HE	AI-LC	RA-HE	RA-LC	LB-HE	LB-LC	LP-HE	LP-LC
Class A 2560x1600	Traffic	1.95	1.91	0.67	0.61				
	PeopleOnStreet	2.16	2.06	1.61	1.53				
	Nebuta	1.28	1.36	0.77	0.83				
	SteamLocomotive	0.83	0.82	0.34	0.33				
Class B 1920x1080	Kimono	1.15	1.07	0.65	0.60	0.82	0.79	0.84	0.81
	ParkScene	1.91	1.89	0.81	0.75	1.06	0.99	1.12	1.04
	Cactus	1.87	1.87	0.77	0.72	0.90	0.84	0.97	0.91
	BasketballDrive	1.66	1.74	0.72	0.71	0.84	0.86	0.89	0.91
Class C 832x480	BQTerrace	2.08	2.08	0.76	0.73	0.90	0.87	1.10	1.06
	BasketballDrill	2.32	2.35	0.97	0.94	1.04	1.02	1.13	1.09
	BQMall	2.23	2.28	0.95	0.91	1.15	1.12	1.23	1.19
	PartyScene	2.75	2.73	1.24	1.19	1.55	1.50	1.70	1.63
Class D 416x240	RaceHorses	2.03	2.05	1.45	1.41	1.59	1.55	1.64	1.60
	BasketballPass	2.13	2.27	1.47	1.45	1.61	1.60	1.63	1.61
	BQSquare	2.54	2.59	1.14	1.14	1.57	1.54	1.74	1.68
	BlowingBubbles	2.78	2.78	1.34	1.31	1.71	1.67	1.81	1.75
Class E 1280x720	RaceHorses	2.33	2.36	1.81	1.77	1.96	1.93	2.00	1.95
	Vidyo1	1.66	1.66			0.49	0.45	0.52	0.47
	Vidyo3	1.55	1.58			0.45	0.42	0.53	0.48
평균	Vidyo4	1.57	1.56			0.40	0.38	0.45	0.41
	Class A	1.55	1.54	0.85	0.83				
	Class B	1.74	1.73	0.74	0.70	0.90	0.87	0.98	0.95
	Class C	2.33	2.35	1.15	1.11	1.33	1.30	1.42	1.38
평균	Class D	2.45	2.50	1.44	1.42	1.71	1.68	1.80	1.75
	Class E	1.59	1.60			0.44	0.42	0.50	0.46

한 영상의 특성에 따라 복잡한 영상들은 작은 CU로 부호화하는 것이 효율적이고, 움직임이 별로 없거나 복잡하지 않은 영상들에서는 큰 CU로 부호화하는 것이 효율적임을 알 수 있다.

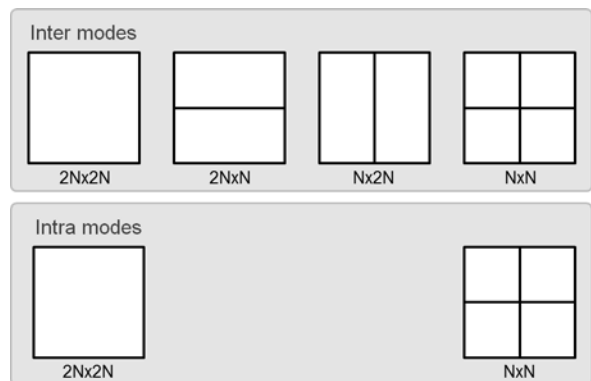
### Ⅲ. 예측 및 변환 블록 구조

#### 1. 예측 블록 구조

제 Ⅱ 장에서 서술한 바와 같이, CU는 HEVC의 기본 부호화 블록이다. 하나의 CU는 그 특징에 따라, 인터 예측에서는 2N×2N 또는 2N×N, N×2N, N×N PU 모드가 적용되며, 인트라 예측에서는 2N×2N 또는 N×N PU모드가 사용된다. 여기서 2N×2N PU 모드는 현재 부호화 하고자 하는 CU와 동일한 크기의 PU를 의미하며, N×N PU 모드는 현재 CU의 1/4 크기로 분할한 모드를 나타낸다. HEVC에서는 다양한 크기의 CU와 PU 모드를 사용함에 따라 발생하는 중복되는 모드 사용을 피하기 위하여 N×N 예측 모드는 현재 CU가 최소 크기

의 CU인 경우에만 적용한다. 〈그림 3〉은 인터 또는 인트라 CU에 적용 가능한 PU 모드들을 보여준다.

HEVC는 기존의 인터 모드 및 스킵(Skip) 모드 뿐만 아니라 머지(Merge) 모드도 사용한다. 머지 모드는 현재 PU 주변에 존재하는 PU들의 움직임 정보들 중 하나를 현재 PU에 사용하는 모드이다. 이에 따라 어느 PU의 움직임 정보를 현재 PU에서 사용했는지를 나타내는 색인 정보를 추가로 부호화한다.



〈그림 3〉 PU 분할 모드

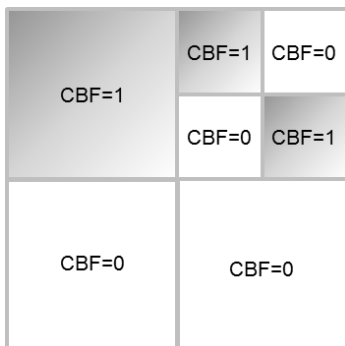
## 2. 변환 블록 구조

기존의 비디오 코덱에서는 하나의 기본 부호화 블록을 다수개의 세부 블록으로 분할하여 예측을 수행하고, 해당 예측 블록 단위로 인트라 및 인터 예측을 수행한 후, 변환 및 양자화를 수행한다. 특히, 최신 코덱인 H.264/AVC에서는 8×8와 4×4 블록 단위로 변환을 수행하며, 분할된 세부 예측 블록의 크기가 8×8 보다 작은 경우에는 4×4 변환만 수행한다<sup>[6]</sup>. 이는 하나의 기본 부호화 블록이 다수 개의 세부 예측 블록으로 분할되는 경우, 각 세부 예측 블록의 오차 신호들 사이의 연관성이 떨어지는 것을 반영한 제한이다.

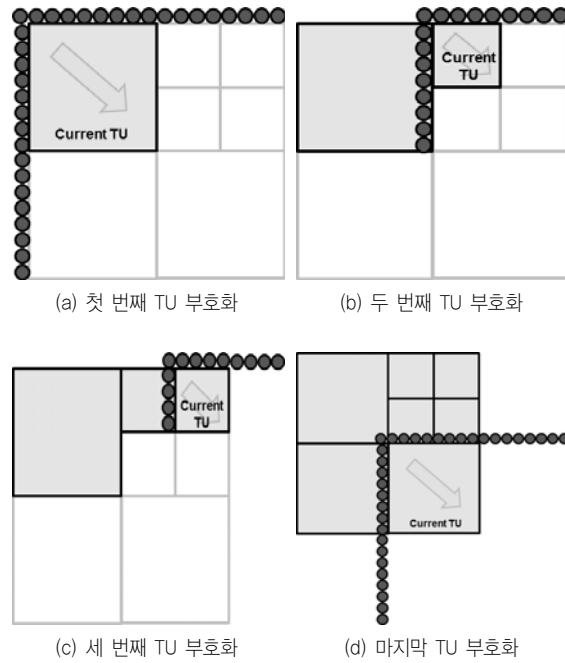
HEVC는 기존 변환 블록의 구조와 달리 RQT(Residual Quadtree) 개념을 도입하여 현재 CU의 크기에 따라 다양한 변환 블록의 크기를 사용할 수 있다<sup>[5]</sup>. 최소 변환 블록 크기는  $\log_2_{min\_transform\_block\_size\_minus2}$ 를 사용하여 표현하고, 최대 변환 블록 크기는  $\log_2_{diff\_max\_min\_transform\_block\_size}$ 를 사용하여 최소 변환 블록 크기로부터 계산한다. 두 선택스 모두 SPS를 통해 전송된다. 각 CU에서 계층적으로 분할되는 변환 블록을 표현하기 위해서 *split\_transform\_flag*를 사용한다. <그림 5>와 같이 RQT 구조를 사용하여 예측 오차 신호를 변환 및 양자화 함으로써 예측 오차 신호의 특성에 따라 적응적으로 변환 블록의 크기가 결정된다.

CU가 다수개의 PU로 분할되더라도 각 PU에서 발생하는 예측 오차 신호들이 비슷한 특성을 가지는 경우가 발생한다. 이 경우에는 각 PU의 잔차 신호들을 하나의 TU로 설정한 후, 변환 및 양자화를 적용하고 엔트로피 부호화를 수행한다. 이 과정을 통해 부호화 효율을 향상시킬 수 있다. 다른 한편, 하나의 예측 블록에 존재하는 예측 오차신호들이 화소 위치별로 서로 다른 특성을 가지는 경우에는 다수개의 TU로 분할하여 변환 및 양자화를 수행함으로써 전체적인 부호화 효율의 향상을 얻게 된다. <그림 4>는 인터 PU의 잔차 신호를 다양한 크기의 TU로 분할하여 변환 및 양자화, 엔트로피 부호화를 수행한 결과를 보여준다.

<그림 4>와 같이, 인터 PU의 잔차 신호는 각 TU의 특성에



<그림 4> 인터 PU의 TU 구조의 예



<그림 5> 인트라 PU의 TU 구조의 예

따라 다수개의 TU로 분할되어 부호화될 수 있다. 이에 따라 부호화할 변환 계수가 존재하는 TU에 대해서만 엔트로피 부호화를 수행함으로써 부호화를 효율적으로 수행할 수 있다.

인트라 PU 내부의 화소값들은 하나의 같은 예측 모드를 사용하여 예측을 수행한다. 그리고 하나의 인트라 PU는 인터 PU의 경우와 마찬가지로 다수개의 TU로 분할되어 변환 및 양자화를 수행하게 된다. 그러나 인트라 PU의 경우에는 부호화 효율을 최대화하기 위하여 TU 단위로 부호화 및 복호화를 수행하여 복원 화소값을 생성하게 된다. 이에 따라 하나의 TU에서 먼저 복호화된 복원 화소값을 사용하여 다음 TU의 화소값을 예측하는데 이용할 수 있다. 따라서 예측 블록 단위로 인트라 예측을 수행하는 경우와 비교하여 보다 정밀한 인트라 예측이 가능하다. <그림 5>는 인트라 PU의 TU 구조와 부호화 과정의 예를 보여준다. 첫 번째 TU는 <그림 5>의 (a)와 같이 현재 PU의 주변 복원 화소들을 이용하여 예측을 수행하고, 변환 및 양자화, 역양자화, 역변환을 차례로 수행하여 첫 번째 TU의 화소값을 복원한다. 이처럼 이전 TU들에서 복원된 화소들은 <그림 5>의 (b)~(d)와 같이 다음 TU들을 예측할 때 사용된다. 이에 따라 각 TU들의 가장 가까운 거리의 복원 화소들을 이용할 예측을 수행함으로써 보다 정확한 예측 결과를 얻을 수 있다.

## IV. HM4.0의 부호화 구조

최근 이탈리아 토리노에서 개최된 6차 JCT-VC 회의에서

는 부호화 구조에 대한 다양한 의견들이 제시되었다. 그 중에서도 작은 크기의 PU에 대한 논의가 활발히 진행되었다. 그 결과로 인터 예측에서는 4×4 크기의 PU를 사용하지 않아도 부호화 효율의 저하가 크지 않고 부호화 복잡도가 상대적으로 많이 감소하는 것을 알 수 있었다<sup>[7]</sup>. 이에 따라 SPS에 1비트 선택스를 삽입하여 인터 4×4 PU의 사용여부를 조절할 수 있도록 하였다. 또한 부호화 효율의 향상을 위하여 N×2N과 2N×N PU들의 첫 번째 블록을 강제로 머지 모드로 부호화하는, 일명 ‘Partial Merge’ 방법을 사용하지 않기로 결정하였다<sup>[8]</sup>. 이 같은 내용을 반영한 참조 소프트웨어로서 HM4.0이 2011년 8월 중에 공개될 예정이다.

## V. 결론

본 문서에서는 현재 HEVC의 최근 동향을 반영한 부호화 및 예측, 변환 블록 구조에 대하여 설명하였으며, 이와 관련된 선택스를 소개하였다. 그리고 추후 공개될 HM4.0에서 블록 구조와 관련된 내용을 간략하게 서술하였다.

### 참고문헌

- [1] “Joint Call for Proposals on Video Compression Technology,” w11113, ITU-T Q6/16 Visual Coding and ISO/IEC JTC1/SC29/WG11 Coding of Moving Pictures and Audio, Jan., 2010.
- [2] “Summary of HEVC working draft 1 and HEVC test model (HM),” JCTVC-C405, 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 Oct., 2010.
- [3] G. Sullivan and J. R. Ohm, “Meeting report of the first meeting of the Joint Collaborative Team on Video Coding (JCT-VC),” Dresden, DE, 15-23 April, 2010, JCTVC-A200, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 1st Meeting: Dresden, DE, 15-23 April, 2010.
- [4] “Meeting report of the third meeting of the Joint Collaborative Team on Video Coding (JCT-VC),” JCTVC-C400, 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 Oct., 2010.
- [5] “High Efficiency Video Coding (HEVC) text specification Working Draft 3,” JCTVC-E603, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 5th Meeting, Geneva, CH, 7-15 Mar., 2011.
- [6] “Advanced Video Coding for Generic Audiovisual Services,” ITU-T Rec. H.264 and ISO/IEC 14496-10, Doc. E32768, Nov., 2007.
- [7] B. Bross, “BoG Report on MV Coding and Throughput/Parsing Robustness,” JCTVC-F744, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting, Torino, IT, 14-22 Jul., 2011.
- [8] I. K. Kim, E. Alshina, et. al. “Experiments on tools in Working Draft (WD) and HEVC Test Model (HM3.0),” JCTVC-F465, Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 6th Meeting, Torino, IT, 14-22 Jul., 2011.



서 찬 원

2007년 2월 세종대학교 정보통신공학과 학사.  
2009년 2월 세종대학교 정보통신공학과 석사.  
2011년 8월 현재 세종대학교 정보통신공학과 박사과정.  
(관심분야) HEVC, H.264/AVC, SVC



한 종 기

1992년 2월 KAIST 전기및전자공학과 학사.  
1994년 2월 KAIST 전기및전자공학과 석사.  
1999년 2월 KAIST 전기및전자공학과 박사.  
1999년 3월~2001년 8월 삼성전자 책임연구원.  
2001년 9월~2006년 2월 세종대학교 정보통신공학과 조교수.  
2006년 3월~현재 세종대학교 정보통신공학과 부교수.  
(관심분야) 영상처리, 동영상 부호화, 최적화이론