

# VVC의 엔트로피 코딩

□ 김대연 / (주)칩스앤미디어

## 요약

VVC(Versatile Video Coding)는 H.264/AVC(Advanced Video Coding)와 H.265/HEVC(High Efficiency Video Coding)의 엔트로피 코딩 기술로 사용되었던 CABAC(Context-based Adaptive Binary Arithmetic Coding)을 기반으로하여 압축률과 처리율을 향상시킬 수 있는 다양한 기술들이 채택되어 현재 CD(Committee Draft)가 완성되었고 참조 모델인 VTM6.0이 정식으로 배포되었다. 본 논문에서는 VVC Draft 6에 채택된 엔트로피 코딩 관련 기술들과 H.265/HEVC의 엔트로피 코딩의 차이점을 설명하고 엔트로피 코딩의 압축 성능과 엔트로피 코딩의 복잡도를 분석한다.

## 1. 서론

ISO/IEC MPEG과 ITU-T VCEG은 2015년 10월 H.265/HEVC의 다음 동영상 압축 표준을 만들기 위한 사전 준비 작업을 위해 Joint Video Exploration Team(JVET)을 설립하였다. 전세계의 동영상 압축

전문가들이 압축 성능을 향상시킬 수 있는 다양한 코딩 툴들을 제안하였으며 제안된 코딩 툴들이 융합되어 H.265/HEVC의 참조 소프트웨어인 HM 대비 약 35%의 압축 효율을 갖는 소프트웨어 JEM을 개발하였다. JEM을 통하여 새로운 표준에 대한 실현 가능성을 확인하였고 2018년 4월 Call for Proposal을 시작으로 H.265/HEVC 다음 버전의 동영상 압축 표준인 VVC에 대한 표준화가 시작되어 2020년 표준화 완료 예정이다. 2019년 7월 스웨덴 예테보리 회의에서 JVET은 Draft 6[1]를 위원회 초안(Committee Draft : CD)으로 발간하기로 결정하였고 현재 ISO/IEC의 CD 승인 절차가 진행 중이다.

VVC의 엔트로피 코딩은 H.265/HEVC의 엔트로피 코딩을 기반으로 개발되었기 때문에 상당히 유사한 면이 있지만 압축 성능면에서의 개선을 위해 context의 확률을 유도하는 방법에 있어서 갱신 속

도가 다르며 정확도가 높은 2개의 확률을 이용하는 방법이 채택되었으며 처리율 향상을 위하여 변환 계수 블록 단위의 context-coded bin의 수를 제한하는 잔여 신호에 대한 엔트로피 코딩 방법 등의 기술들이 채택되었다.

본 고의 II 장에서는 H.265/HEVC 엔트로피 코딩과 비교하여 VVC의 엔트로피 코딩에서 새롭게 추가된 기술들과 변경된 확률 초기화 방법, CABAC range sub-interval 유도 방법 등을 알아본다. 그리고 III장에서는 VVC 표준화 회의에 보고된 자료를 토대로 채택된 기술들로 인한 압축 성능 변화 및 엔트로피 코딩의 처리율 변화를 알아보며 IV장에서는 결론을 맺는다.

## II. VVC의 엔트로피 코딩

### 1. 확률과 엔트로피 코딩 엔진의 초기화

VVC의 CABAC은 H.265/HEVC와 동일하게 context에 대한 확률 초기화가 필요하다. 한 슬라이스 내에서 부호화할 CTU의 개수가 적을 수록 또는 비트율이 낮은 환경일수록 context의 실제 확률과 유사한 초기화 값을 사용하는 것이 압축 효율 측면에서 높은 이득을 얻을 수 있다. VVC 표준화에서는 컨텍스트 확률들의 초기화 값을 결정하기 위해 CTC(Common Test Condition)에서 정의된 환경(영상 및 부호화 예측 구조, QP 범위) 뿐만 아니라 높은 QP(Quantization Parameter)와 다양한 영상들을 이용하여 트레이닝이된 최적의 확률 값을 초기화 값으로 정하였다. context들의 확률은 슬라이스 타입에 따라 정의되어 부호화할 슬라이스 타입에 따라 다른 초기화 값이 사용된다. 6비트의 초기

화 값(initValue)을 이용하여 각 컨텍스트에 대한 2개의 확률 인덱스(pStateIdx0, pStateIdx1)를 유도하며 그 과정은 다음 식 (1)~(4)와 같이 정의되었다. 먼저 i번째 컨텍스트에 대하여 현재 슬라이스 타입(sliceType)에 해당하는 initValue(sliceType, i)를 이용하여 확률이 QP에 영향을 받는 정도를 나타내는 m과 n의 값을 식 (1)과 (2)와 같이 유도한다. 그리고 슬라이스 헤더에 전송되는 휘도 성분에 대한 슬라이스 QP와 식 (3), (4)를 이용하여 최종 2개의 pStateIdx0, pStateIdx1를 유도한다. H.265/HEVC는 LPS의 확률에 대한 인덱스를 유도하는 반면에 pStateIdx0, pStateIdx1은 1의 확률을 인덱스화한 값을 의미하기 때문에 엔트로피 코딩 복호화 시 각 context의 MPS 값을 따로 저장하지 않아도 되는 장점이 있다.

$$\begin{aligned} \text{slopeIdx} &= \text{initValue}(\text{sliceType}, i) \gg 3 \\ \text{offsetIdx} &= \text{initValue}(\text{sliceType}, i) \& 7 \end{aligned} \quad (1)$$

$$\begin{aligned} m &= \text{slopeIdx} - 4 \\ n &= (\text{offsetIdx} * 18) + 1 \end{aligned} \quad (2)$$

$$\text{preCtxState} = \text{Clip3}(1, 127, ((m * (\text{Clip3}(0, 51, \text{SliceQpY}) - 16)) \gg 1) + n) \quad (3)$$

$$\begin{aligned} \text{pStateIdx0} &= \text{preCtxState} \ll 3 \\ \text{pStateIdx1} &= \text{preCtxState} \ll 7 \end{aligned} \quad (4)$$

VVC CABAC의 산술 엔트로피 코딩 엔진은 H.265/HEVC와 동일하다. 따라서, H.265/HEVC와 동일하게 range와 offset은 16 bit 정확도를 갖고 엔진의 초기화가 필요한 경우, range는 510, offset은 비트스트림으로부터 9 bit를 읽은 값으로

초기화된다.

## 2. 확률 초기화 및 엔진 초기화 수행 시점

VVC는 brick 내의 첫번째 CTU의 첫번째 선택스 요소에 대한 복호화 시작 시, 컨텍스트 확률 초기화와 엔진 초기화를 수행한다. 이 때, 초기화 값은 QP에 의존적인 표준에 미리 정의된 값으로 유도된 값이 사용된다. 만약, `entropy_coding_sync_enabled_flag`이 1인 경우, 즉, Wave-front Parallel Processing이 수행되는 경우에는 brick 단위 행의 첫번째 CTU의 복호화 시작시 확률의 동기화가 수행된다. 동기화란 현재 CTU의 context들의 확률값들을 상단 CTU의 모든 context들의 확률값으로 동기화시키고 엔진 초기화를 수행하는 과정이다.

## 3. CABAC range sub-interval 유도 및 확률 갱신

기존 H.264/AVC와 H.265/HEVC는 LPS의 확률에 대한 range 계산 시, 곱셈 연산을 수행하지 않고 발생가능한 range를 4개의 구간으로 샘플링하여 근사화된 range를 LPS 확률 인덱스로 매핑하는 64x4 크기의 테이블을 이용하는 LUT(Look-Up Table) 방식을 사용했다. 하지만 LUT 방식도 확률의 정밀도가 높아져 테이블의 크기가 큰 경우에는 곱셈 연산을 수행하는 것과 비교하여 처리를 측면에서 큰 차이가 나지 않기 때문에 VVC의 산술 엔트로피 복호화는 H.265/HEVC와는 다르게 다음 식 (5), (6)과 같이 곱셈 연산이 포함된 방법으로 LPS 확률에 대한 range(`ivlLpsRange`)를 유도한다. 식 (5)에서 `ivlCurrRange`는 현재 range이며, `valMps`는 현재 context-coded bin의 MPS(Most

Probable Symbol)을 의미한다.

$$qRangeIdx = ivlCurrRange \gg 5 \quad (5)$$

$$\begin{aligned} pState &= pStateIdx1 + 16 * pStateIdx0 \\ valMps &= pState \gg 14 \\ ivlLpsRange &= (qRangeIdx * ((valMps - 32767 \\ &\quad - pState : pState) \gg 9) \gg 1) + 4 \end{aligned} \quad (6)$$

Context-coded bin의 엔트로피 복호화가 수행된 뒤, 해당 context에 대한 확률의 갱신의 수행된다. H.265/HEVC는 확률의 갱신 시, MPS가 발생될수록 확률 변경의 간격이 좁아지는 로그 형태의 규칙을 LUT 방식으로 사용하였다. VVC 엔트로피 코딩의 확률 갱신은 선형 방정식을 이용한 확률 갱신을 사용하며 모든 context 마다 확률 갱신의 속도를 조절하기 위한 상수값(`shiftIdx`)을 사용하여 갱신된 2개의 확률에 대한 인덱스를 유도한다. 식 (7)과 같이 확률 갱신의 속도를 나타내는 상수값(`shiftIdx`)로부터 2개의 확률에 대한 확률 갱신 속도(`shift0`, `shift1`)를 나타내는 값을 유도할 수 있다. 이 값들과 현재 복호화된 bin 값(`binVal`)과 갱신 전 2개의 확률 값에 대한 인덱스를 이용하여 갱신된 확률값에 대한 인덱스(`pStateIdx0`, `pStateIdx1`)들을 식 (8)과 같이 유도할 수 있다.

$$\begin{aligned} shift0 &= (shiftIdx \gg 2) + 2 \\ shift1 &= (shiftIdx \& 3) + 3 + shift0 \end{aligned} \quad (7)$$

$$\begin{aligned} pStateIdx0 &= pStateIdx0 - (pStateIdx0 \gg shift0) \\ &\quad + (1023 * binVal \gg shift0) \\ pStateIdx1 &= pStateIdx1 - (pStateIdx1 \gg shift1) \\ &\quad + (16383 * binVal \gg shift1) \end{aligned} \quad (8)$$

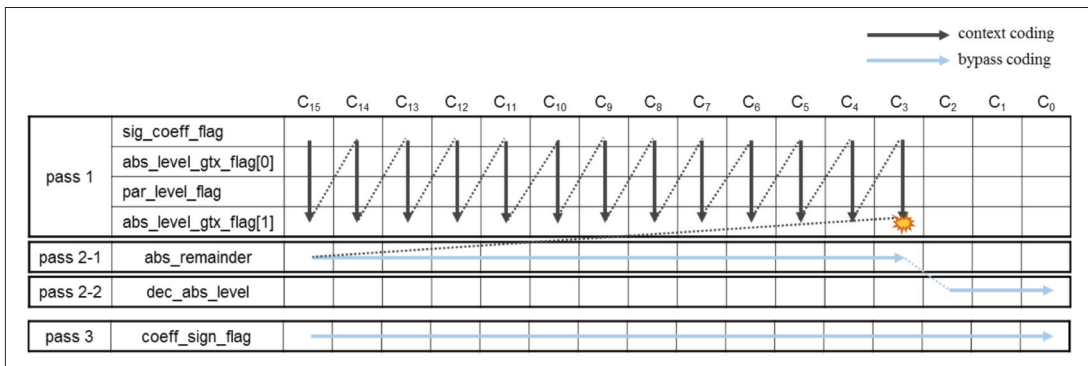
#### 4. 잔여 계수에 대한 부호화

변환 및 양자화 후 발생하는 잔여 계수는 높은 처리율이 요구되는 고비트율 환경에서는 비트 스트림 내 가장 많은 양을 차지하기 때문에 이에 대한 처리율을 높일 수 있는 방법이 필요하다. 특히, context-coded bin에 대한 엔트로피 복호화는 산술 엔트로피 복호화 특성 상, 압축 효율을 높일 수 있는 장점이 있지만 처리율을 감소시키는 문제가 있다. 따라서, VVC에서는 변환 블록 단위(TB) 내에서 발생될 수 있는 context-coded bin의 최대 개수를 제한하여 고비트율 환경에서도 엔트로피 코딩의 처리율을 어느 정도 보장해줄 수 있는 기술이 채택되었다. 식 (9)는 TB 내 최대 허용되는 context-coded bin의 수 MCCB를 계산하는 식이다. VVC는 서브 블록 트랜스폼(SBT) 또는 Low-Frequency Non-Separable Transform(LFNST) 사용시, 변환 영역 내 특정 영역의 잔여 계수들을 강제로 0으로 만드는 방법이 사용된다. 식 (9)에서  $TB\_zoSize$ 는 0으로 만든 영역을 제외한 나머지 영역의 크기이며 상수  $\alpha$ 는 휘도 블록의 경우 1.75, 색차 블록의 경우

1.25로 정의되었다.

$$MCCB = TB\_zoSize \times \alpha \quad (9)$$

〈그림 1〉은 잔여 계수에 대한 신택스들의 부호화 과정을 3개의 pass로 구분한 그림이며 임의의 변환 계수 블록이 16개의 잔여 계수를 갖으며 MCCB를 초과한 경우에 어떤 신택스들이 어떤 순서로 부호화되는지를 나타낸 그림이다. 먼저 부호화에 앞서 pass 1에서 context-coded bin의 개수가 MCCB를 초과하는지 판단하기 위한 변수  $remBinsPass1$ 를 MCCB로 초기화한다. 그리고 각 변환 계수 블록의 잔여 계수 부호화 시, pass 1에서 모든 잔여 계수에 대한 context-coded bin으로 정의된  $sig\_coeff\_flag$ ,  $abs\_level\_gt1\_flag$ ,  $par\_level\_flag$ ,  $abs\_level\_gt3\_flag$ 를 부호화하고 난 뒤, 모든 잔여 계수 중 4 이상의 level을 갖는 계수들에 대하여 pass 2-1에서 bypass bin으로 정의된  $abs\_remainder$ 를 부호화한다. 이 때, pass 1에서 context-coded bin을 부호화할 때마다 해당 변수의 값을 1씩 감소시키며 부호화 도중  $remBinsPass1$ 이 4 미만인 경우에는



〈그림 1〉 VVC의 잔여 계수에 대한 신택스들의 부호화 과정의 예[2]

그 이후의 모든 선택스들은 bypass bin으로 부호화한다. 따라서 그림에서 C2, C1, C0 잔여 계수들은 pass 1을 거치지 않고 bypass bin으로 정의된 dec\_abs\_level와 coeff\_sign\_flag 선택스들로 부호화된다. 이 방법으로 인해 contert coded bin의 최대 개수를 제한할 수 있으며 연속된 bypass bin이 발생하는 경우를 많이 발생하게 함으로써 고비트율 환경에서의 엔트로피 코딩의 처리율의 급격한 하락을 막을 수 있다.

### III. 압축 성능 및 복잡도 분석

앞서 설명한 바와 같이 VVC의 엔트로피 코딩에 채택된 컨텍스트의 확률 유도 방법은 압축 성능 향

상을 위해 각 컨텍스트마다 다른 확률 갱신 선형 모델을 갖을 수 있고 다른 확률 갱신 속도를 갖을 수 있는 2개의 확률을 사용한다. <표 1>과 <표 2>는 이 방법이 채택됨으로써 JVET에서 권고하는 공통 실험 조건(CTC : Common Test Condition)[3] 하에서 H.265/HEVC CABAC 대비 BD-Rate 증가율 및 복잡도를 나타낸 것이다. 표와 같이 H.265/HEVC의 CABAC과 비교하여 복잡도의 증가는 미미한 수준이지만 AI(All Intra)과 RA(Random Access) 환경 모두 약 0.9%의 압축 성능 향상을 보였다[4].

JVET은 VVC 표준화 과정 중, 구현 관련 AHG (Ad hoc)을 결성하여 VTM이 변경됨에 따라 엔트로피 코딩의 객관적인 처리율과 HM(HEVC Reference Model)과의 처리율 비교를 보고하고 있다[5].

<표 1> AI 환경에서의 H.265/HEVC CABAC 대비 VVC 엔트로피 코딩의 BD-Rate 증가율 및 복잡도

|          | Y      | U      | V      | EncT | DecT |
|----------|--------|--------|--------|------|------|
| Class A1 | -0.94% | -1.35% | -1.11% | 109% | 103% |
| Class A2 | -0.97% | -1.33% | -1.27% | 108% | 103% |
| Class B  | -0.92% | -0.97% | -1.16% | 108% | 100% |
| Class C  | -0.90% | -0.95% | -1.17% | 108% | 107% |
| Class E  | -0.93% | -1.21% | -1.37% | 110% | 104% |
| Overall  | -0.93% | -1.13% | -1.21% | 109% | 103% |
| Class D  | -0.85% | -1.08% | -1.07% | 107% | 106% |
| Class F  | -0.76% | -0.82% | -0.81% | 108% | 104% |

<표 2> RA 환경에서의 H.265/HEVC CABAC 대비 VVC 엔트로피 코딩의 BD-Rate 증가율 및 복잡도

|          | Y      | U      | V      | EncT | DecT |
|----------|--------|--------|--------|------|------|
| Class A1 | -0.87% | -1.09% | -1.22% | 106% | 100% |
| Class A2 | -0.90% | -0.74% | -0.82% | 101% | 97%  |
| Class B  | -0.94% | -0.85% | -1.07% | 104% | 101% |
| Class C  | -0.89% | -1.07% | -0.99% | 104% | 103% |
| Overall  | -0.91% | -0.93% | -1.03% | 104% | 101% |
| Class D  | -0.84% | -1.22% | -1.04% | 103% | 102% |
| Class F  | -0.59% | -0.65% | -0.84% | 104% | 102% |

〈표 3〉 CTC 상에서의 VTM6.0과 HM16.19의 최대 Bit 대비 Bin의 비율(Bin2bit ratio)

|     | Bin2bit ratio (peak, unweighted) |        |                    | Bin2bit ratio (peak, weighted) |        |                    |
|-----|----------------------------------|--------|--------------------|--------------------------------|--------|--------------------|
|     | HM16.19                          | VTM6.0 | VTM6.0 vs. HM16.19 | HM16.19                        | VTM6.0 | VTM6.0 vs. HM16.19 |
| AI  | 1.23                             | 1.38   | 12.91%             | 0.95                           | 1.14   | 19.89%             |
| RA  | 1.21                             | 1.39   | 14.36%             | 0.93                           | 1.13   | 21.70%             |
| LDB | 1.24                             | 1.38   | 11.57%             | 0.96                           | 1.13   | 17.14%             |
| LDP | 1.23                             | 1.38   | 11.49%             | 0.96                           | 1.13   | 17.21%             |

〈표 4〉 낮은 QP 범위에서의 VTM6.0과 HM16.19의 최대 Bit 대비 Bin의 비율(Bin2bit ratio)

|     | Bin2bit ratio (peak, unweighted) |        |                    | Bin2bit ratio (peak, weighted) |        |                    |
|-----|----------------------------------|--------|--------------------|--------------------------------|--------|--------------------|
|     | HM16.19                          | VTM6.0 | VTM6.0 vs. HM16.19 | HM16.19                        | VTM6.0 | VTM6.0 vs. HM16.19 |
| AI  | 1.19                             | 1.31   | 9.85%              | 0.79                           | 0.93   | 17.71%             |
| RA  | 1.18                             | 1.27   | 6.87%              | 0.74                           | 0.86   | 15.15%             |
| LDB | 1.18                             | 1.29   | 8.98%              | 0.78                           | 0.92   | 17.73%             |
| LDP | 1.18                             | 1.29   | 8.94%              | 0.78                           | 0.92   | 17.52%             |

〈표 3〉과 〈표 4〉는 다양한 예측 구조 (AI, RA, LDB, LDP)에서 VTM6.0과 HM16.19의 최대 Bit 대비 Bin의 비율(Bin2bit ratio)을 나타낸 표이다. Bin의 종류는 context-coded bin과 bypass bin가 있고 처리율 측면에서 일반적으로 bypass bin은 context-coded bin보다 약 4배 이상의 처리율로 부호화 및 복호화 가능하기 때문에 처리율을 의미할 수 있는 Bin2bit ratio 계산시 1개의 bypass bin을 0.25개로 간주하여 좀 더 정확한 처리율을 예측할 수 있다. 〈표 3〉과 〈표 4〉에서 unweighted는 bypass bin을 1개의 bin으로 간주하여 bin2bit ratio를 계산한 결과이며 weighted는 bypass bin을 0.25개의 bin으로 간주하여 계산한 결과이다. 〈표 3〉은 CTC에서 측정된 결과이기 때문에 일반적인 QP 범위 (22, 27, 32, 37)의 결과이며, VTM6.0이 HM16.19 대비 unweighted 시, 약 12%, weighted 시, 약 20%의 처리율이 감소할 것이라고

예측할 수 있다. 〈표 4〉는 엔트로피 코딩의 처리율이 특히 문제가 되는 낮은 QP 범위 (2, 7, 12, 17)에서 측정된 결과이며, VTM6.0이 HM16.19 대비 unweighted 시, 약 8%, weighted 시, 약 17%의 처리율이 감소할 것이라고 예측할 수 있다. 일반적인 QP 범위에서 측정된 결과와 비교하여 처리율 증가율이 낮은 이유는 낮은 QP 영역에서는 잔여 계수들이 발생하기 때문에 본문에서 설명한 변환 블록 단위의 최대 context-coded bin 제한 방법이 효과적으로 적용되어 처리율을 향상시킨 결과라고 유추할 수 있다.

## IV. 결론 및 CD 이후 채택 가능성 있는 기술

본 고에서는 H.265/HEVC 엔트로피 코딩과 비

교하여 VVC에서 새롭게 추가된 context 확률 예측 방법과 context-coded bin 수 제한을 갖는 잔여 계수 엔트로피 코딩 방법과 변경된 확률 초기화 방법, CABAC range sub-interval 유도 방법 등을 알아 보았다. 이러한 기술들로 인해 VVC의 엔트로피 코딩은 처리율의 감소를 최소화하면서 압축 성능을 향상시킬 수 있었다. CD 이후 엔트로피 코딩 분야에서는 압축 성능 측면에서의 약간의 손해는 있더라도 고비트율에서의 매우 높은 처리율로 엔트로피

부호화 및 복호화를 가능하게 하는 기술들이 제안되고 있다. 파나소닉과 칩스앤미디어는 CABAC을 수행하지 않고 이진열을 그대로 전송하는 CABAC skip mode[6]를 제안하였으며 인텔과 HHI는 CABAC range sub-interval 유도 시, shift 연산만을 이용하는 방법을 제안하였다[7-8]. 이러한 기술들이 산술 엔트로피 코딩의 처리율 문제를 해결할 수 있는 방법이기 때문에 향후 채택 가능성이 높은 기술로 판단된다.

## 참고 문헌

- [1] B. Bross, J. Chen and S. Liu, "Versatile Video Coding (Draft 6)", JVET-O2001-vE, the 15th JVET meeting, July 2019, Gothenburg, SE
- [2] J. Chen, Y. Ye, S. Kim, "Algorithm description for Versatile Video Coding and Test Model 6 (VTM 6)", JVET-O2002, the 15th JVET meeting, July 2019, Gothenburg, SE
- [3] J. Boyce, K. Suehring, X. Li and V. Seregin, "JVET common test conditions and software reference configurations for SDR video", JVET-N1010, the 14th JVET meeting, March 2019, Geneva, CH
- [4] F. Bossen, "CE5 on arithmetic coding: experiments 5.1.1, 5.1.2, 5.1.3, 5.1.4, 5.1.5, 5.1.6, 5.1.7, 5.1.8, 5.1.10, 5.1.11, 5.1.12, 5.1.13, 5.2, and more", JVET-M0453, the 13th JVET meeting, Jan. 2019, Marrakech, MA
- [5] M. Zhou, "AHG16/Non-CE7: A study of bin to bit ratio for VTM6.0", JVET-P0050, the 16 th JVET meeting Oct. 2019, Geneva, CH
- [6] K. Abe, T. Toma and, D.-Y. Kim, "CABAC skip mode", JVET-O0308, the 15th JVET meeting, July 2019, Gothenburg, SE
- [7] A. Alshin, J. Boyce, Pavel Frolov, Vasily Aristarkhov, "Removal of infrequently used context models", JVET-O0554, the 15th JVET meeting, July 2019, Gothenburg, SE
- [8] H. Kirchhoffer, D. Marpe, B. Bross, T. Nguyen, C. Rudat, H. Schwarz, and T. Wiegand, "High throughput CABAC mode for VVC", JVET-P0300, the 16 th JVET meeting Oct. 2019, Geneva, CH

## 필자 소개



김대연

- 2001년 3월 ~ 2005년 2월 : 세종대학교 인터넷학과 학부 졸업
- ~ 2010년 8월 : 세종대학교 컴퓨터공학과 박사 졸업
- ~ 2012년 7월 : ㈜픽스트리 선임연구원
- ~ 현재 : ㈜칩스앤미디어 수석연구원