

특집논문 (Special Paper)

방송공학회논문지 제31권 제2호, 2026년 3월 (JBE Vol.31, No.2, March 2026)

<https://doi.org/10.5909/JBE.2026.31.2.240>

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

## 포인트 클라우드 비디오의 tile 기반 MPEG-DASH 스트리밍

최 문 영<sup>a)</sup>, 임 동 한<sup>a)</sup>, 김 동 호<sup>b)‡</sup>

### Tile-Based MPEG-DASH Point Cloud Video Streaming

Moon Young Choi<sup>a)</sup>, Dong Han Lim<sup>a)</sup>, and Dong Ho Kim<sup>b)‡</sup>

#### 요 약

포인트 클라우드 비디오 스트리밍은 XR, 메타버스, 원격 협업과 같은 차세대 미디어 서비스의 핵심 기술로 주목받고 있다. 그러나 기존 세그먼트 단위 기반 스트리밍은 한 세그먼트 전체가 동일 화질로 스트리밍 되어 불필요한 대역폭 소모, 높은 지연 문제를 야기한다. 본 연구에서는 이러한 문제를 해결하기 위해 타일 기반 MPEG-DASH 스트리밍 시스템을 설계하였다. 포인트 클라우드 오브젝트를 6개의 타일로 분할, MPEG V-PCC를 적용하여 인코딩 작업을 수행하였다. 또한, MPD를 확장하고 각 타일의 좌표, 법선 벡터, 면적 정보를 담은 메타데이터 파일을 생성하여 DASH 스트리밍에 활용하였다. 더불어 개별 타일의 정면성 및 인접 타일과의 겹침 여부를 고려하는 비트레이트 할당 알고리즘을 적용하여 대역폭 예산 내에서 각 타일의 화질을 동적으로 결정하였다.

#### Abstract

Point cloud video streaming has attracted attention as a key technology for next-generation media services such as XR, the metaverse, and remote collaboration. However, conventional segment-based streaming delivers an entire segment at a uniform quality, which leads to unnecessary bandwidth consumption and increased latency. To address these issues, this study designs a tile-based MPEG-DASH streaming system. A point cloud object is partitioned into six tiles, and MPEG V-PCC is applied for encoding. In addition, we extend the MPD and generate a metadata file containing each tile's coordinates, normal vector, and area information for use in DASH streaming. Furthermore, we apply a bitrate allocation algorithm that considers each tile's frontality and overlap with adjacent tiles, dynamically determining the quality of each tile within a given bandwidth budget.

Keyword : Point Cloud Video, MPEG-DASH, Tile-based streaming, Viewport-adaptive streaming, MPEG V-PCC

a) 서울과학기술대학교 전자IT미디어공학과(Seoul National University of Science and Technology)

b) 서울과학기술대학교 스마트ICT융합공학과(Seoul National University of Science and Technology)

‡ Corresponding Author : 김동호(Dong Ho Kim)

E-mail: dongho.kim@seoultech.ac.kr

Tel: +82-2-970-6417

ORCID: <https://orcid.org/0000-0001-9136-8932>

※ 이 논문은 2025년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. RS-2023-00278925, 초실감 메타버스 서비스 위한 통신 기술 연구)

※ 이 논문의 연구 결과 중 일부는 한국방송·미디어공학회 2025년 추계학술대회에서 발표한 바 있음.

· Manuscript January 7, 2026; Revised February 19, 2026; Accepted February 20, 2026.

Copyright © 2026 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.”

## 1. 서론

최근 XR, 메타버스, 원격회의, 디지털 트윈 등 3차원 콘텐츠 서비스의 수요가 급격히 증가하면서 포인트 클라우드(Point Cloud) 기술이 주목받고 있다. 포인트 클라우드는 (x, y, z) 좌표와 색상 정보로 객체를 표현하는 방식으로, 자유로운 시야 이동을 지원하는 6DoF(6 Degrees of Freedom) 환경 구현에 효과적인 기술이다. MPEG(Moving Picture Experts Group)은 포인트 클라우드 압축 표준으로 V-PCC(Video-based Point Cloud Compression)와 G-PCC(Geometry-based Point Cloud Compression)를 제안하였다. 이 중 V-PCC는 3D 데이터를 2D 비디오로 투영 후 압축 과정에서 H.264, HEVC와 같은 기존 비디오 코덱을 활용하기에 압축 효율이 매우 뛰어나 비디오 스트리밍 환경에서 효율적인 압축 기술이다. 이러한 압축 표준을 바탕으로 현재 포인트 클라우드 비디오의 스트리밍에 관한 연구가 활발히 진행 중이다. 그러나 다수의 기존 연구들은 세그먼트 단위 스트리밍에 기반하고 있다. 여기에서의 세그먼트 단위란 개별 포인트 클라우드 오브젝트의 비디오를 시간 단위로 분할한 것으로 기존 2D 비디오의 MPEG-DASH(Dynamic Adaptive Streaming over HTTP) 스트리밍을 위한 세그먼트 분할과 그 개념적 의미가 같다. 이러한 세그먼트 단위 포인트 클라우드 비디오의 스트리밍은 사용자의 FoV(Field of View)와 관계없이 해당 오브젝트의 모든 부분을 동일한 화질로 스트리밍하고, 이는 불필요한 대역폭 소모로 이어지게 되어 전체 비디오의 화질이 낮아지는 등 사용자 QoE(Quality of Experience)가 저하될 수 있다<sup>[1]</sup>. 따라서 본 연구는 타일 단위 스트리밍(Tile-based Streaming) 기법을 적용, 기존 세그먼트 단위 기반 스트리밍에 비해 효율적인 대역폭 사용이 가능하도록 한다. 또한, 타일 분할로 인한 요청-응답 수의 증가에 따른 지연 시간 증가 문제를 최근 발표된 UDP 기반의 HTTP/3 프로토콜을 활용하여 해결하고자 하였다.

## II. 관련 연구 배경

### 1. MPEG V-PCC와 포인트 클라우드 압축 기술

[그림 1]은 원본 포인트 클라우드를 6면 정사영 방식을

통해 6개의 패치로 분할하고, 각 패치를 2D 이미지인 Occupancy Map, Attribute Image, Geometry Image로 변환하여 기존 비디오 코덱(H.264, HEVC 등)으로 인코딩 하는 MPEG 표준 압축 기술 MPEG V-PCC의 패치 생성 및 패키징 과정을 시각화한 것이다. MPEG V-PCC는 압축 효율이 높은 H.264, HEVC 등의 비디오 코덱을 활용하여 전송해야 할 데이터 양을 효과적으로 줄일 수 있다. 또한, V-PCC는 포인트 클라우드를 2D 이미지 형태로 변환하여 압축하므로 결과물이 비디오 세그먼트처럼 처리될 수 있다. 이는 MPEG-DASH의 세그먼트 구조와 호환되어 표준 스트리밍 파이프라인에 통합하기 용이하다. 따라서, 본 연구에서는 MPEG-VPCC를 적용하여 스트리밍 시스템을 구현하였다.

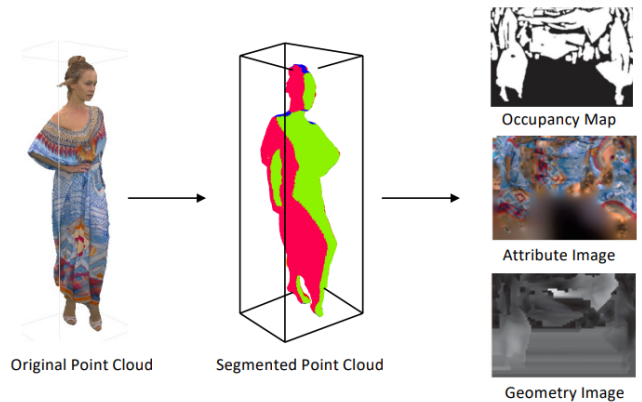


그림 1. MPEG V-PCC의 패치 생성 및 패키징 예시  
Fig. 1. Example of patch generation and packing of MPEG V-PCC

### 2. HTTP/3 QUIC

[그림 2]는 HTTP/2와 HTTP/3에서 발생하는 HoL(Head of Line) blocking 문제를 시각화한 것이다. HoL blocking은 앞서 전송되던 패킷이 손실될 경우 순서 보장 메커니즘으로 인해 해당 패킷의 재전송이 완료될 때까지 다른 패킷의 전송이 일시 중단되는 현상이다. HTTP/2는 한 TCP 연결에서 여러 스트림을 통해 동시에 패킷 송수신이 이뤄진다. 이때, 특정 스트림에서 패킷 전송 중 손실이 발생하여 재전송이 진행될 경우 순서 보장 메커니즘으로 인해 다른 스트림에서의 패킷 전송이 중단되어 전체 지연 시간이 증가하는 문제가 발생할 수 있다. 반면, HTTP/3는 HTTP/2와 달리 HoL blocking 문제로부터 자유로운 장점을 갖는다.

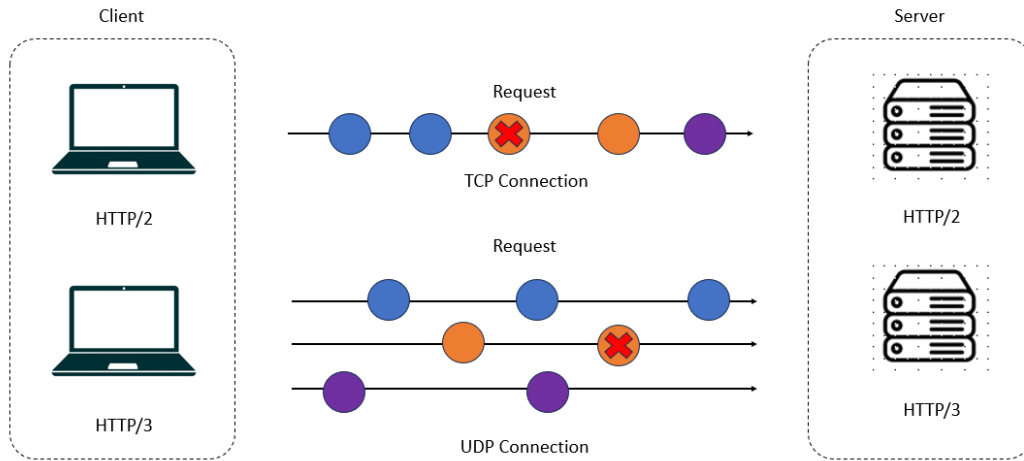


그림 2. HTTP/2와 HTTP/3의 HoL blocking  
 Fig. 2. HoL blocking of HTTP/2 and HTTP/3

HTTP/3는 2021년 표준화 작업이 완료된 3번째 HTTP 프로토콜 버전으로, TCP 대신 UDP 기반의 QUIC(Quick UDP Internet Connections) 위에서 동작하도록 설계되었다. QUIC은 TCP와 달리 TLS 1.3이 통합되어 있어 1-RTT (Round Trip Time)에 보안 연결이 성립되며, 재연결 시 0-RTT로 데이터 전송이 가능하다. 또한, 연결 내의 각 스트림이 독립적으로 순서 보장, 재전송을 처리하도록 설계되어 패킷 손실이 발생한 스트림만 전송이 지연된다. 본 연구에서 도입한 타일 기반 스트리밍은 기존 세그먼트 기반 스트리밍 대비 오브젝트의 수가 증가할수록 요청-응답의 수가 더욱 크게 증가하고, 이는 지연 시간의 증가로 이어져 실시간성이 중요한 포인트 클라우드 비디오의 스트리밍에 있어 큰 제약이 될 수 있다. 따라서, 서버-클라이언트 간 연결 설정이 빠르며, HoL blocking 문제로부터 자유로운 QUIC 기반의 HTTP/3를 스트리밍 시스템에 적용하여 전체 지연 시간 증가 문제를 해결하고자 하였다.

### 3. 관련 선행 연구

타일 기반 적응형 스트리밍 기법이 3DoF 환경의 360° 비디오에서 대역폭 절약을 통한 사용자의 QoE 향상에 효과적임을 입증한 선행 연구를 확인하였다<sup>[2][3][4]</sup>. 해당 연구에서는 360° 비디오의 전체 세그먼트를 타일링하여, 사용

자의 FoV에 따라 특정 부분은 고화질로, 특정 부분은 저화질로 스트리밍하는 타일 기반 적응형 스트리밍 기법이 효율적인 대역폭 사용을 가능하게 하여 FoV 내 비디오를 지속적으로 고화질로 유지하고, 이는 사용자의 QoE 향상에 효과적임을 확인하였다. 한 연구에서는 인간의 시력 특성을 참고한 효율적인 화질 선택 방법을 소개하고 있다<sup>[5]</sup>. 해당 연구에서는 사용자의 카메라와 포인트 클라우드 오브젝트 사이의 거리가 멀수록 화질 저하를 인식하기 어렵다는 특성을 참고하여 FoV 내의 오브젝트에 대해서 사용자와 가까운 오브젝트를 우선적으로 높은 화질로 결정할 필요가 있음을 설명한다. 한 연구에서는 각 오브젝트의 세그먼트에 대한 화질 결정 결과에 따른 비트레이트 할당에 있어 어떠한 방법이 효율적인지에 대해 설명한다<sup>[6]</sup>. 해당 연구에서는 대역폭이 충분하지 않은 상황에서, 고화질로 결정된 세그먼트들만 집중적으로 비트레이트를 할당하는 것이 아닌, FoV 내에 속하는 세그먼트들의 화질을 화질 결정 결과에 따라 설정된 우선순위에 따라 전체적으로 높이는 것이 사용자의 QoE 향상에 있어 효과적임을 입증했다. 타일 기반의 포인트 클라우드 비디오 스트리밍 시스템을 구현한 최근 연구 또한 확인하였다<sup>[7]</sup>. 해당 연구에서는 기존 세그먼트 단위가 아닌 타일 기반 스트리밍을 통해 효율적인 대역폭 사용이 가능하도록 시도했다. 특히, 사용자의 시야 벡터와 타일의 법선 벡터 간의 내적 값을 통해 해당 타일이

정면성을 갖는지를 판단하여 각 타일의 화질을 결정하고자 시도했다. 한 논문에서는 타일링된 포인트 클라우드 비디오를 여러 화질로 인코딩하고, 사용자의 현재 FoV에 따라 MPEG-DASH를 통해 전송하는 시스템을 제시하고 있다<sup>[1]</sup>. 해당 논문에서는 볼류메트릭 비디오가 매우 크고 복잡한 데이터를 갖는 것을 언급하면서, 이에 대한 해결책으로서 타일 기반 MPEG-DASH 스트리밍과 같은 효율적인 스트리밍 시스템이 필수적임을 설명한다. HTTP/3 QUIC을 세그먼트 단위의 포인트 클라우드 비디오 스트리밍에 적용하였을 때, 기존 HTTP/2 프로토콜이 적용된 시스템 대비 유의미한 지연 시간 단축이 발생하였음을 평가한 연구 또한 확인하였다<sup>[8]</sup>. 해당 연구에서는 세그먼트의 길이를 줄일수록, 신뢰도를 낮출수록 HTTP/2 대비 상당한 지연 시간 단축을 이뤄냈음을 입증하여, 실시간 포인트 클라우드 스트리밍에 HTTP/3 프로토콜이 효과적임을 설명한다.

### III. 제안 방법

#### 1. 타일링 및 인코딩

MPEG V-PCC는 3차원 객체의 매 프레임을 패치 단위로 분할한 뒤, 선택된 투영 평면으로 정사영하여 Occupancy Map, Attribute Image, Geometry Image로 구성된 2D 아틀라스를 생성하고, 이를 비디오 코덱으로 압축하는 방식으로 인코딩을 수행한다. 이러한 구조에서 임의로 오브젝트를 타일링하여 타일별로 독립적으로 인코딩하는 경우 렌더링 과정에서 색 불연속, 점 누락, 점 중복에 따른 깜빡임 등의 문제가 발생할 수 있다. 본 연구에서는 이러한 문제를 해결하고자 평균점 기반 축 분할 방법을 타일링 작업에 적용하였다. [그림 3]은 평균점 기반 축 분할 방법을 적용하여 포인트 클라우드 객체를 6면으로 타일링한 예시이다. 평균점 기반 축 분할을 위해, 먼저 전체 포인트 클라우드의 평균 좌표를 계산하여 중심점으로 설정한 뒤, 중심점에서 각 점을 향하는 상대 위치 단위 벡터를 구한다. 이후,  $\pm x$ ,  $\pm y$ ,  $\pm z$  총 6개의 축 단위 벡터와 상대 위치 단위 벡터 간 내적 값을 구하고, 내적 값이 가장 큰 축 방향으로 해당 점을 할당한다. 이 과정이 완료되면 한 객체는 [그림 3]의 오른쪽

그림과 같이 총 6개의 포인트 클라우드 타일로 분할되며, 각 타일에 대해 개별적으로 V-PCC 인코딩 작업이 이뤄진다. 이러한 평균점 기반 축 분할 방법은 별도의 복잡한 투영 혹은 표면 추정 과정 연산 없이 단순 평균점과 벡터 연산만으로 전체 오브젝트를 6개의 타일로 나눌 수 있어 구현이 간단하며 연산 비용이 낮다는 장점을 갖는다. 분할된 타일들은 MPEG V-PCC 인코딩 과정에서 Geometry QP, Attribute QP, Level of Detail(LoD) 조정을 통해 고화질, 중간 화질, 저화질의 3가지 화질로 설정하여 각각 독립적인 비트스트림(.bin)으로 생성하였다. Geometry QP는 고화질, 중간 화질, 저화질별 각각 22 / 28 / 34, Attribute QP는 각각 26 / 32 / 38, LoD는 각각 100 / 50 / 30으로 설정하였다.

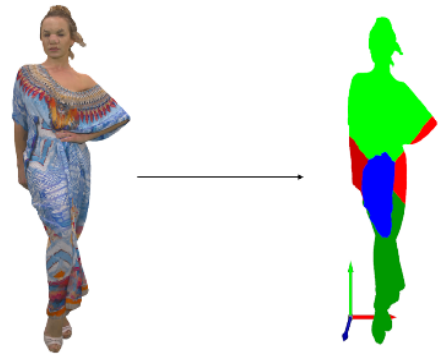


그림 3. 포인트 클라우드 객체의 6면 타일링 시각화  
Fig. 3. Visualization of six-sided tiling for point cloud object

#### 2. MPD 설계 및 타일 메타데이터 생성

관련 선행 연구 중 포인트 클라우드 비디오의 MPEG-DASH 스트리밍을 위한 MPD 구조를 제시한 논문을 확인하였다<sup>[6]</sup>. 해당 논문에서는 클라이언트가 오브젝트별로 다양한 화질의 세그먼트, 즉 비디오 조각을 서버에 요청하여 수신할 수 있도록 하는 새로운 MPD 구조를 제시한다. 하지만, 제시된 MPD 구조는 세그먼트 단위까지만 정의되어 있어 타일 기반 스트리밍을 위해서는 새로운 MPD 구조를 설계할 필요가 있다. 따라서, 세그먼트를 공간 상에서 분할한 타일에 대한 정보를 추가하여 [6]에 제시된 MPD(Media Presentation Description) 구조를 [그림 4]와 같이 수정하였

```

<Period id="object_1" start="PT0.0S" duration="PT10S" xRot="0" yRot="0" zRot="0" xOff="0" yOff="0" zOff="0">
  <AdaptationSet id="object_1_high_quality">
    <Representation id="object_1_high_quality_segment_0">
      <SegmentTemplate timescale="1" duration="1" startNumber="1"
        initialization="object_1/HQ/segment_0/init.bin"
        media="object_1/HQ/segment_0/tile_$.bin" />
    </Representation>
    <Representation id="object_1_high_quality_segment_1">
      <SegmentTemplate timescale="1" duration="1" startNumber="1"
        initialization="object_1/HQ/segment_1/init.bin"
        media="object_1/HQ/segment_1/tile_$.bin" />
    </Representation>
    <Representation id="object_1_high_quality_segment_2">
      <SegmentTemplate timescale="1" duration="1" startNumber="1"
        initialization="object_1/HQ/segment_2/init.bin"
        media="object_1/HQ/segment_2/tile_$.bin" />
    </Representation>
  </AdaptationSet>
</Period>

```

그림 4. 타일 기반 포인트 클라우드 비디오 스트리밍을 위한 MPD 구조  
 Fig. 4. MPD structure for tile-based point cloud streaming

다. 먼저, **Period** 단계에서는 개별 오브젝트를 정의한다. 각 오브젝트는 **id**를 통해 구분되며 3D 공간에서의 **x, y, z** 좌표와 세 축에 대한 회전 정보를 갖는다. **AdaptationSet** 단계에서는 한 오브젝트의 화질을 정의한다. 정의되는 화질 종류는 인코딩 결과에 따라 임의로 설정이 가능하며 **id**를 통해 구분한다. **Representation** 단계에서는 특정 화질에 대한 세그먼트가 정의된다. 여기에서의 세그먼트는 특정 화질로 재생되는 한 오브젝트의 비디오 조각을 의미하며, 각 세그먼트는 **id**를 통해 구분된다. **SegmentTemplate** 단계에서는 한 세그먼트에 속하는 6개의 타일들을 정의한다. 각 타일은 해당 타일이 속하는 세그먼트를 앞서 제시한 평균점 기반 축 분할 방법을 통해 타일링한 결과물이며, 세그먼트와 동일한 재생 시간을 갖는 공간적으로 분할된 비디오 조각이다. 이 MPD는 타일 기반 스트리밍을 위해 [6]과 달리 **SegmentTemplate** 단계에 세그먼트가 아닌 타일을 정의한 구조로서 본 연구에서 새롭게 제시하고 있다. 이러한 새로운 MPD 구조를 통해 클라이언트는 서버로부터 한 오브젝트에 대한 적절한 화질의 타일을 스트리밍 받을 수 있다.

더불어, 각 타일의 **ID**, 중심 좌표, 법선 벡터, 면적, 데이터 크기 등의 정보를 **JSON** 형식의 메타데이터 파일로 별도 생성하였다. 클라이언트는 MPD와 **JSON** 파일을 파싱하고, 현재 시야 정보를 통해 예측한 미래 시야 정보를 바탕으로

각 타일의 화질을 결정, 적절한 타일을 서버에 요청할 수 있다.

### 3. 타일 화질 결정 및 비트레이트 할당 알고리즘

타일 기반 포인트 클라우드 비디오 스트리밍을 위해 개별 타일의 화질 결정 및 비트레이트 할당 알고리즘을 [그림 5]와 같이 설계하였다. 알고리즘 설계를 위해 관련 선행 연구에서 입증한 효율적인 화질 결정 방법을 참고하였다. 먼저, 한 타일의 위치가 사용자의 시선을 기준으로 전체 FoV 내에서 중심, 또는 외곽에 위치하는지 확인한다. 360° 비디오의 타일 기반 적응형 스트리밍 기법을 연구한 선행 논문에서 FoV 내에 속하는 타일들 중 FoV 중심에 위치한 타일들을 고화질로 스트리밍하였을 때 사용자의 QoE 향상에 효과적임을 확인하였다<sup>[2][3][4]</sup>. 따라서 이러한 기존 연구 결과를 반영하여 **Algorithm Example 1**의 5번째 줄과 같이 사용자의 시선 벡터, 즉 사용자로부터 FoV의 중심을 향하는 벡터와 사용자로부터 해당 타일의 중심을 향하는 벡터 사이의 각도의 크기가 전체 FoV 각도의 1/4보다 작거나 같은 경우 해당 타일이 FoV의 중심에 위치한다고 보고 해당 타일을 고화질로 스트리밍하도록 알고리즘을 설계하였다. 만약 두 벡터가 이루는 각도의 크기가 전체 FoV 각도의 1/4보다 크면서 전체 FoV 각도의 1/2보다 작거나 같은 경우에는

| Algorithm 1: Bit-Rate Allocation Example 1   | Algorithm 1: Bit-Rate Allocation Example 2   |
|--|--|
| <p><b>Input:</b> <math>T = \{t_i\}</math>: set of tiles,<br/> <math>TotalArea_i</math>: total tile area,<br/> <math>\theta_{viewport}</math>: angle of viewport,<br/> <math>V_i</math>: vector from the user to the tile center,<br/> <math>\theta_{i-viewport}</math>: angle between view vector and <math>V_i</math>,<br/> <math>\theta_{i-visible}</math>: angle between <math>V_i</math> and normal vector of each tile,<br/> <math>Dist_i</math>: Euclidean distance between user and center of each tile,<br/> <math>\theta_{i,k-overlap}</math>: angle between <math>V_i</math> and a vector that contains two center points of tiles within the viewport,<br/> <math>Budget</math>: total bit-rate budget</p> <pre> 1 <math>qualities \leftarrow [1, \forall t_i \in T]</math>; 2 <math>Usedbudget \leftarrow \sum_{t_i \in T} Size(t_i, 1)</math>; 3 Define Priority levels: <math>Q_1, Q_2, Q_3</math>; 4 for <math>t_i \in T</math> do 5   if <math>\theta_{i-viewport} \leq \frac{\theta_{viewport}}{4}</math> then 6     if <math>-\cos(\theta_{i-visible})</math> is large then 7       if <math>\cos(\theta_{i,k-overlap})</math> is small then 8         if <math>Dist_i</math> is small then 9           add <math>t_i</math> to <math>Q_1</math>; 10        else if <math>Dist_i</math> is medium then 11          add <math>t_i</math> to <math>Q_2</math>; 12        else 13          add <math>t_i</math> to <math>Q_3</math>; </pre> | <pre> 1 while Budget remains do 2   for <math>Q \in \{Q_1, Q_2\}</math> do 3     for tile <math>t_i \in Q</math> do 4       if <math>qualities[t_i] &lt; maxQuality</math> then 5         <math>q \leftarrow qualities[t_i] + 1</math>; 6         <math>cost \leftarrow Size(t_i, q) - Size(t_i, q - 1)</math>; 7         if <math>Usedbudget + cost \leq Budget</math> then 8           <math>Usedbudget \leftarrow Usedbudget + cost</math>; 9           <math>qualities[t_i] \leftarrow q</math>; 10        if no quality improvement in <math>Q_1</math> and <math>Q_2</math> then 11          break; 12 while Budget remains do 13   for <math>Q \in \{Q_3\}</math> do 14     for tile <math>t_i \in Q</math> do 15       if <math>qualities[t_i] &lt; maxQuality</math> then 16         <math>q \leftarrow qualities[t_i] + 1</math>; 17         <math>cost \leftarrow Size(t_i, q) - Size(t_i, q - 1)</math>; 18         if <math>Usedbudget + cost \leq Budget</math> then 19           <math>Usedbudget \leftarrow Usedbudget + cost</math>; 20           <math>qualities[t_i] \leftarrow q</math>; 21        if no quality improvement in <math>Q_3</math> then 22          break; 23 return qualities; </pre> |

그림 5. 타일 비트레이트 할당 알고리즘 예시  
Fig. 5. Example of tile bit-rate allocation algorithm

해당 타일이 FoV의 외곽에 위치, 전체 FoV 각도의 1/2보다 큰 경우에는 FoV 바깥에 위치한다고 보고 이에 따라 타일의 화질을 다르게 결정하도록 알고리즘을 설계하였다. 다른 관련 선행 연구에서는 사용자의 시야 벡터와 타일의 법선 벡터 간의 내적 값을 통해 해당 타일이 정면성을 갖는지를 판단하여 각 타일의 화질을 결정하는 알고리즘을 설계했으며, 이를 통해 사용자의 QoE를 효과적으로 향상시켰음을 입증했다<sup>7)</sup>. 여기에서의 정면성은 포인트 클라우드 오브젝트를 공간 상에서 타일링했을 때 해당 타일이 사용자의 시선을 기준으로 오브젝트의 앞면에 해당하여 시청이 용이한지를 의미한다. 만약 사용자의 시야 벡터와 법선 벡터가 이루는 각도의 크기가 180°가 될 경우 내적 값은 -1이 되며 해당 타일은 정면성을 가장 크게 갖게 된다. 이러한 특성을 반영하여 본 알고리즘에서는 Algorithm Example 1의 6번째 줄과 같이 사용자로부터 한 타일의 중심을 향하는 벡터와 해당 타일의 법선 벡터 사이의 각도를 구하여, 만약 해당 각도의 크기가 최소 90°를 넘어 정면성을 크게 갖는다고 판단되는 경우 해당 타일의 화질을 고화질로 전송하도록 설계하였다. 더불어, 타일 기반 스트리밍의 경우 세그먼

트 기반 스트리밍과 달리 각 타일이 독립적으로 스트리밍 되기 때문에 효율적인 대역폭 사용을 위해 사용자의 시선에 따라 두 타일이 겹치는지 여부까지도 판단하여 적응적으로 화질을 조정하여 스트리밍하는 것이 가능하다. 따라서 Algorithm Example 1의 7번째 줄과 같이 사용자로부터 한 타일의 중심을 향하는 벡터와 해당 타일의 중심을 지나면서 또한 FoV 내 또 다른 타일의 중심을 지나는 한 벡터 사이의 각도를 계산하여 두 벡터의 내적 값이 0에 가까울수록 FoV 내 두 타일 간 겹침 문제가 발생하지 않는다고 보고, 만약 그렇지 않은 경우 두 타일 중 사용자와의 거리가 먼 타일의 화질을 저화질로 스트리밍하도록 알고리즘을 설계하였다. 최종적으로, 인간의 시력 특성으로 인해 사용자와 오브젝트 사이의 거리가 멀수록 오브젝트의 화질 저하를 인식하기 어렵다는 특성을 이용, 사용자와의 거리가 가까운 오브젝트를 고화질로 전송하여 사용자의 QoE를 향상시켰음을 입증한 논문을 참고하여 Algorithm Example 1의 8번째 줄과 같이 사용자와 타일 중심 간의 유클리드 거리를 계산하여 해당 거리가 가까울수록 고화질로 결정하도록 알고리즘을 설계하였다<sup>5)</sup>. 또 다른 연구에서는 대역폭이 충분

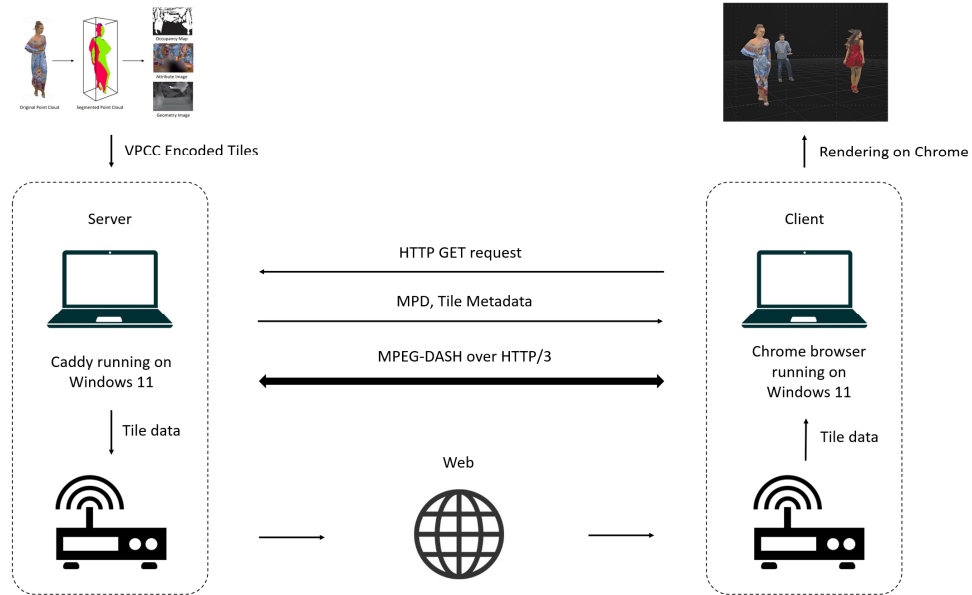


그림 6. 제안 타일 기반 포인트 클라우드 스트리밍 시스템 요약도  
 Fig. 6. Overview of proposed tile-based streaming system

하지 않은 상황에서, 고화질로 결정된 세그먼트들만 집중적으로 비트레이트를 할당하는 것이 아닌, FoV 내에 속하는 세그먼트들의 화질을 화질 결정 결과에 따라 설정된 우선순위에 따라 전체적으로 높이는 것이 사용자의 QoE 향상에 있어 효과적임을 입증한 것을 확인했다<sup>6)</sup>. 따라서 Algorithm Example 2와 같이 고화질에 해당하는 타일들만 우선적으로 비트레이트를 할당하지 않고, 고화질과 중간 화질로 결정된 타일들에 대해 고화질로 결정된 타일들부터 우선적으로 화질을 한 단계 상승시킨 후 중간 화질로 결정된 타일들의 화질을 한 단계 상승시키도록 설계했으며, 만약 고화질과 중간 화질로 결정된 타일들이 모두 최대 화질로 상승되었음에도 비트레이트 예산이 남은 경우 저화질로 결정된 타일들의 화질을 한 단계씩 상승시키도록 알고리즘을 설계하였다. 이와 같은 알고리즘은 기존 연구에서 입증한 QoE 향상에 효과적인 알고리즘을 모두 통합함과 동시에 타일 기반 스트리밍이 갖는 장점인 공간 상 분할의 특징을 활용한 겹침 문제 여부까지 고려하여 효율적인 대역폭 사용 및 사용자의 QoE 향상에 있어 효과적이다. 특히, 기존 연구에서 제시한 효과적인 화질 결정 알고리즘의 통합을 시도한 점과 타일 간 겹침 문제를 고려한 알고리즘 설계는

해당 논문에서 최초로 제시한 것으로 기존 연구, 또는 기술과 차별화되는 점이라고 생각한다.

#### IV. 구현 및 실험 결과

##### 1. 시스템 구현

HTTP/3를 기본 지원하는 Caddy 서버를 기반으로 스트리밍 시스템을 구현하였다. 서버는 인코딩된 타일 비트스트림과 메타데이터를 세그먼트 단위로 Caddy 서버에 저장하고, 이를 QUIC 기반의 HTTP/3 프로토콜을 통해 클라이언트로 전송한다. 클라이언트는 웹 브라우저 환경에서 Three.js 기반 렌더링 모듈과 비트레이트 할당 알고리즘을 통해 대역폭 예산 내에서 필요한 타일을 적응적으로 요청하고 시각화한다.

##### 2. HTTP/3 네트워크 성능 평가

Ubuntu 환경에서 Mininet을 통해 HTTP/3와 HTTP/2를

적용한 시스템 간의 평균 처리량을 분석하였다. 평가 결과, RTT=60ms, Jitter=±5ms, Loss=0.5%의 제약 조건에서 HTTP/3 적용 시 평균 처리량이 HTTP/2 대비 최대 약 40% 증가하여 기존 HTTP/2 적용 대비 한정된 대역폭 조건에서 저지연성을 크게 확보했음을 확인하였다.

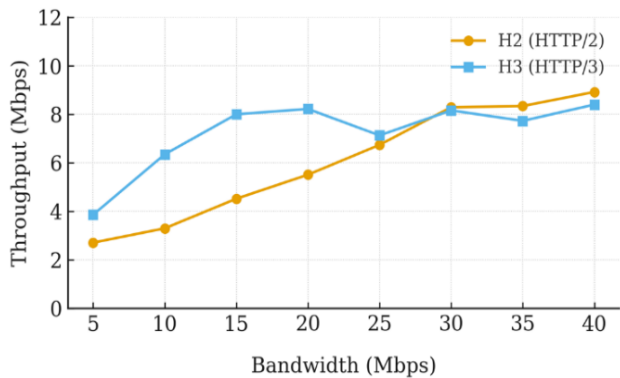


그림 7. HTTP/2와 HTTP/3 간 네트워크 성능 비교 결과  
 Fig. 7. Network performance results of HTTP/3 compared with HTTP/2

### 3. 오프라인 영상 품질 평가

[표 1]과 [그림 8]은 예산에 따른 PSNR(Peak Signal-to-Noise Ratio), SSIM(Structural Similarity Index) 값을 나타낸 표와 예산에 따른 PSNR, SSIM의 개형을 나타낸 것이다. 레이아웃은 일렬(Row), 삼각형(Triangle), 단일(Single) 세 가지로 구성하였다. 다양한 레이아웃(Row, Triangle, Single)에 대해 PSNR과 SSIM을 측정한 결과, 평

균 SSIM이 0.9 이상으로 유지되었으며, 27Mbps 대비 약 60% 낮은 10.5Mbps에서도 시각적 품질 저하가 거의 발생하지 않았다. 이를 통해 제안 시스템이 제한된 대역폭 환경에서도 높은 QoE를 유지함을 확인하였다.

표 1. 예산별 PSNR/SSIM 결과  
 Table 1. PSNR/SSIM Results by Budget

| Layout   | Budget (Mbps) | PSNR (dB) | SSIM   |
|----------|---------------|-----------|--------|
| Row      | 10.5000       | 30.3700   | 0.9809 |
| Row      | 14.4000       | 31.3900   | 0.9834 |
| Row      | 18.9000       | 39.3800   | 0.9876 |
| Triangle | 10.5000       | 33.3500   | 0.9900 |
| Triangle | 14.4000       | 34.3400   | 0.9922 |
| Triangle | 18.9000       | 36.9500   | 0.9948 |
| Single   | 3.0000        | 27.2500   | 0.9686 |
| Single   | 4.5000        | 27.7100   | 0.9703 |
| Single   | 6.3000        | 29.3300   | 0.9821 |

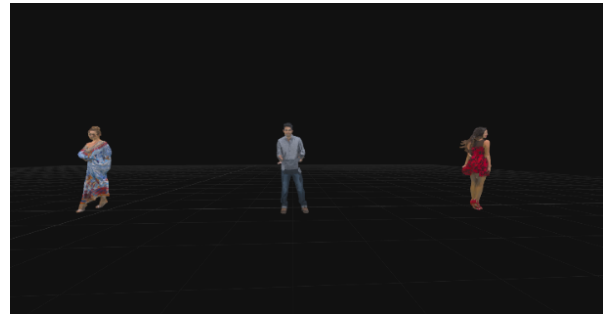


그림 9. Layout Row 렌더링 결과 예시  
 Fig. 9. Rendering result example of Layout Row

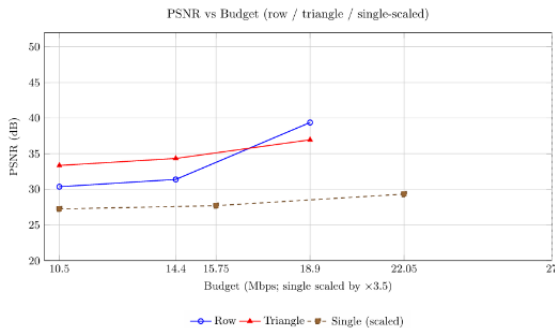
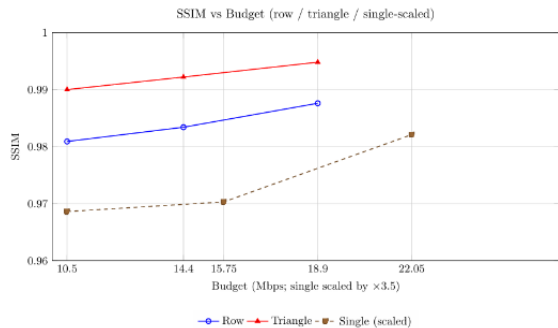


그림 8. 대역폭 예산에 따른 PSNR, SSIM 결과  
 Fig. 8. PSNR and SSIM results by bandwidth budget



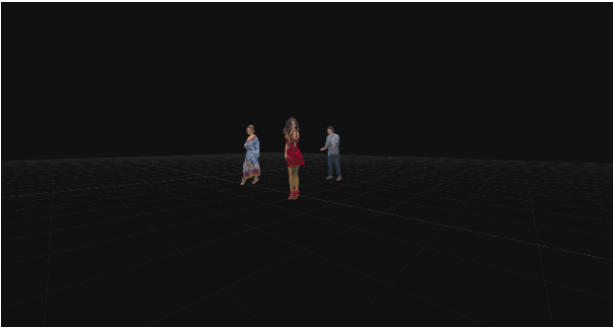


그림 10. Layout Triangle 렌더링 결과 예시  
 Fig. 10. Rendering result example of Layout Triangle

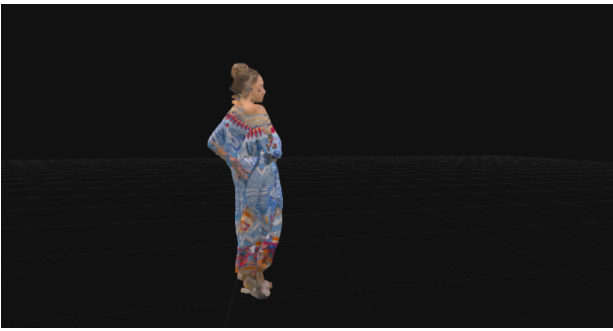


그림 11. Layout Single 렌더링 결과 예시  
 Fig. 11. Rendering result example of Layout Single

#### 4. 타일 비트레이트 할당 알고리즘 지연 시간 평가

[그림 12]는 타일 비트레이트 할당 알고리즘 적용하였을

때 발생하는 스트리밍 지연 시간을 로컬 환경에서 Chrome 브라우저를 통해 분석한 그래프이다. 총 10개의 단일 중간 화질 세그먼트에 대하여 빨간색 선과 파란색 선 각각 타일 비트레이트 할당 알고리즘을 적용하였을 때와 적용하지 않았을 때 발생하는 다운로드 시간을 세그먼트별로 계산하였다. 다운로드 시간은 두 경우 모두 총 10번의 다운로드 시도에 대한 지연 시간의 평균값을 계산하여 구하였다. 평가 결과 타일 비트레이트 할당 알고리즘을 적용한 경우 세그먼트 하나에 대한 다운로드 시간 평균은 약 720.325ms로 알고리즘을 적용하지 않은 경우에 발생하는 다운로드 시간 평균인 575.965ms에 비해 약 144.360ms 증가하는 것으로 확인하였다. 또한, 가장 작은 차이를 보이는 0번 세그먼트에 대한 다운로드 시간의 경우 알고리즘을 적용하였을 때 약 115.888ms의 지연 시간 증가를 확인하였으며, 가장 큰 차이를 보이는 2번 세그먼트에 대한 다운로드 시간의 경우 알고리즘을 적용하였을 때 약 166.797ms의 지연 시간 증가를 확인하였다. 이는 논문에서 제시하는 타일 비트레이트 할당 알고리즘이 한정된 대역폭 조건에서 사용자의 FoV 내에 속하는 타일들의 화질 향상에는 효과적이지만 지연 시간 증가로 인하여 최종 FPS가 하락할 수 있으며, 이는 사용자의 QoE 저하 문제로 이어질 수 있음을 의미한다. 더불어, 본 평가를 통해 확인한 144.360ms의 추가 지연 시간은 스트리밍하는 오브젝트 수의 증가, 즉 전체 타일의 수의 증가에 따라 더욱 커질 수 있다. 따라서 전체 타일의 수가 증가

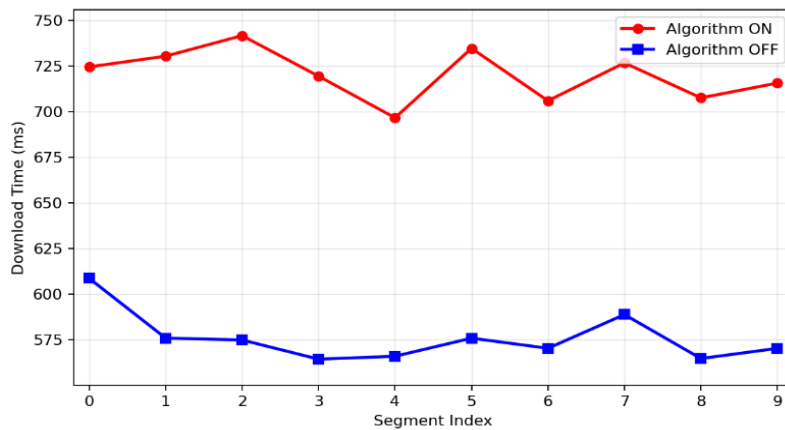


그림 12. 타일 비트레이트 할당 알고리즘 적용 시 발생하는 지연 시간 평가 결과  
 Fig. 12. Evaluation results of latency incurred by applying the tile bitrate allocation algorithm

하는 경우 해당 문제를 해결하기 위한 저지연 코덱의 적용, 세그먼트 당 타일 수 감소, 알고리즘 단순화 등의 시도가 필요할 것으로 생각한다.

## V. 결론

본 논문에서는 타일 기반 MPEG DASH 스트리밍 시스템을 설계·구현하고, 기존 세그먼트 기반 방식에서 발생하는 불필요한 대역폭 사용 문제를 개선할 수 있음을 보였다. 타일 단위로 포인트 클라우드 비디오를 공간적으로 분할하여 여러 화질로 인코딩하고, MPD 확장 및 개별 타일에 대한 메타데이터 파일, 클라이언트 측의 시야 예측 정보를 통해 적응형 스트리밍을 가능하게 하였다. 향후 Draco, WebRTC 기반의 저지연 스트리밍 환경에 본 시스템을 확장할 예정이다.

## 참고 문헌 (References)

- [1] Z. Liu, Q. Li, X. Chen, C. Wu, S. Ishihara, J. Li, and Y. Ji, "Point Cloud Video Streaming: Challenges and Solutions," *IEEE Network*, Vol.35, No.5, pp.202-209, September 2021.  
doi: <https://doi.org/10.1109/MNET.101.2000364>
- [2] A. Yaqoob, M. A. Togou, and G.-M. Muntean, "Dynamic Viewport Selection-Based Prioritized Bitrate Adaptation for Tile-Based 360°

- Video Streaming," *IEEE Access*, Vol.10, pp.29377-29392, March 2022.  
doi: <https://doi.org/10.1109/ACCESS.2022.3157339>
- [3] J. Van der Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. De Turck, "Tile-Based Adaptive Streaming for Virtual Reality Video," *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol.15, No.4, Article 110, pp.1-24, December 2019.  
doi: <https://doi.org/10.1145/3362101>
- [4] J. Van der Hooft, M. T. Vega, S. Petrangeli, T. Wauters, and F. De Turck, "Optimizing Adaptive Tile-Based Virtual Reality Video Streaming," *Proceedings of the 16th IFIP/IEEE International Symposium on Integrated Network Management (IM 2019) - Mini-Conference*, Washington, DC, USA, pp.381-387, April 2019.  
<https://ieeexplore.ieee.org/abstract/document/8717893>
- [5] M. Hosseini and C. Timmerer, "Dynamic Adaptive Point Cloud Streaming," *Proceedings of the 23rd Packet Video Workshop (PV '18)*, Amsterdam, Netherlands, pp.25-30, June 2018.  
doi: <https://doi.org/10.1145/3210424.3210429>
- [6] J. Van der Hooft, T. Wauters, F. De Turck, C. Timmerer, and H. Hellwagner, "Towards 6DoF HTTP Adaptive Streaming Through Point Cloud Compression," *Proceedings of the 27th ACM International Conference on Multimedia (MM '19)*, Nice, France, pp.2405-2413, October 2019.  
doi: <https://doi.org/10.1145/3343031.3350917>
- [7] Y. Shi, B. Clement, and W. T. Ooi, "QV4: QoE-Based Viewpoint-Aware V-PCC-Encoded Volumetric Video Streaming," *Proceedings of the 15th ACM Multimedia Systems Conference (MMSys '24)*, Bari, Italy, pp.144-154, April 2024.  
doi: <https://doi.org/10.1145/3625468.3647619>
- [8] H. K. Ravuri, M. T. Vega, J. Van der Hooft, T. Wauters, and F. De Turck, "Adaptive Partially Reliable Delivery of Immersive Media Over QUIC-HTTP/3," *IEEE Access*, Vol.11, pp.38094-38111, April 2023.  
doi: <https://doi.org/10.1109/ACCESS.2023.3268008>

---

## 저 자 소 개

### 최 문 영



- 2026년 : 서울과학기술대학교 전자미디어공학과 공학사
- 2026년 ~ 현재 : 서울과학기술대학교 스마트ICT융합공학과 석사과정
- ORCID : <https://orcid.org/0009-0001-9933-1656>
- 주관심분야 : 멀티미디어/실감미디어(VR/AR) 전송기술 연구, 차세대이동통신시스템 전송기술 및 시스템 설계 연구

---

저 자 소 개

---



**임 동 한**

- 2026년 : 서울과학기술대학교 전자미디어공학과 공학사
- ORCID : <https://orcid.org/0009-0006-5427-2329>
- 주관심분야 : 멀티미디어/실감미디어(VR/AR) 전송기술 연구, 차세대이동통신시스템 전송기술 및 시스템 설계 연구



**김 동 호**

- 1997년 : 연세대학교 전자공학과 공학사
- 1999년 : KAIST 전기 및 전자공학과 공학석사
- 2004년 : KAIST 전기 및 전자공학과 공학박사
- 2004년 ~ 2006년 : 삼성종합기술원 4G무선기술랩 전문연구원
- 2006년 ~ 2007년 : 삼성전자 통신연구소 책임연구원
- 2007년 ~ 현재 : 서울과학기술대학교 스마트ICT융합공학과 교수
- 2013년 ~ 현재 : IEEE Senior Member
- ORCID : <https://orcid.org/0000-0001-9136-8932>
- 주관심분야 : 차세대이동통신시스템 전송기술 및 시스템 설계 연구, 멀티미디어/실감미디어(VR/AR) 전송기술 연구, 방송통신시스템 전송기술 연구, 무선통신 및 이동통신시스템